XXXIV JORNADAS DE AUTOMÁTICA

Terrassa, 4-6 de septiembre 2013



Actas

Actas de las

XXXIV Jornadas de Automática

Terrassa, 4 al 6 de Septiembre de 2013

Organizan:





Actas de las XXXIV Jornadas de Automática

Editado por:

Ramon Sarrate Estruch

Joseba Quevedo Casín

ISBN: 978-84-616-5063-7

Depósito Legal: B. 17365-2013

Presentación

Las XXXIV Jornadas de Automática (JA2013), a realizar del 4 al 6 de Septiembre del 2013 en el Campus de Terrassa de la Universitat Politècnica de Catalunya (UPC), han sido organizadas por el Departamento de Ingeniería de Sistemas, Automática e Informática Industrial (ESAII) de la UPC.

Las Jornadas vienen realizándose anualmente desde el año 1977, promovidas por el Comité Español de Automática (CEA), y organizadas por distintas universidades o centros de investigación.

Al evento se han inscrito aproximadamente 200 expertos en esta área, principalmente miembros del Comité Español de Automática (CEA), que está organizado en 9 grupos temáticos: Automática Marina, Bioingeniería, Control Inteligente, Educación en Automática, Ingeniería de Control, Modelado y Simulación, Robótica, Sistemas de Tiempo Real y Visión por Computador.

La conferencia inaugural será impartida por Mª Luisa Castaño Directora General de Innovación y Competitividad del Ministerio de Economía y Competitividad (MINECO). El programa técnico incluye dos conferencias plenarias (una a cargo de Manfred Morari sobre "The role of Theory in Control Practice" y la otra a cargo de Ramon Vilanova sobre "El PID en el Tercer Milenio"), una reseña sobre la historia de la Automática por parte de Sebastián Dormido, una presentación de las oportunidades que ofrece Universia por parte de Jaume Pagés, una mesa redonda docente sobre los Másteres en Automática, las reuniones técnicas de los diferentes grupos temáticos y sesiones de presentación de comunicaciones expuestas, como es costumbre en las Jornadas, en un espacio con pósteres. Y este año como novedad se llevará a cabo un interesante curso sobre robótica asistencial y cuatro presentaciones prácticas de empresas muy interesantes. Además se ha previsto en el día central de las Jornadas realizar un entrañable reconocimiento a los impulsores de la Automática en la UPC.

El programa técnico viene acompañado, como es habitual, con un interesante programa social y cultural, como la visita del Museo Nacional de la Ciencia y la Tecnología de Catalunya (MNATEC) y la copa de bienvenida en la sala Jazz Cava de Terrassa así como la visita románica de Sant Benet del Bages, visita al Instituto gastronómico Alicia y cena de gala en el restaurante de ese entorno.

Además, y como viene siendo habitual en las Jornadas precedentes, el programa técnico y el programa social vienen acompañados de un programa de ocio y de actividades para acompañantes.

Y sin olvidar que como en ediciones pasadas, este año también se llevarán a cabo los concursos de robots humanoides y de vehículos cuatrirrotores o "drones" con una fantástica respuesta de participación de equipos investigadores nacionales e internacionales.

Por último, queremos expresar nuestro más sincero agradecimiento a todas las personas y entidades que han prestado su colaboración a que la presente edición de las Jornadas haya sido posible. A todos los miembros del comité organizador, pieza indispensable para el buen funcionamiento del evento, y a todos los miembros del comité científico. También queremos agradecer a las empresas colaboradoras por los diferentes premios ofrecidos durante las XXXIV Jornadas de Automática.

Os damos la bienvenida a todos los participantes y quedamos a vuestra disposición para garantizar una estancia grata y fructífera durante estos días.

Más información en la Web de las Jornadas en http://ja2013.upc.edu/

Terrassa, 20 de Julio de 2013

Joseba Quevedo, Teresa Escobet y Vicenç Puig

Co-responsables del Comité Organizador

Comités Organizador y Científico

Comité organizador

Joseba Quevedo
Vicenç Puig
Teresa Escobet
Fatiha Nejjari
Andreu Quesada
Sebastián Tornil
Josep Contreras
Rosa Ginesta
Albert Márquez
Albert Masip
Judit Esteve
Ramon Sarrate
Quim Blesa
Rita Planas
Pep Cugueró
Miquel Àngel Cugueró
Ramon Comasòlivas
Josep Mª Fuertes
Pere Caminal
Enric Fossas
Maria Serra
Robert Griñó
Beatriz F. Giraldo
Jose María Huerta
Cecilio Angulo
Pere Ponsa
Ramon Costa
Diego García
Ramon Pérez
Damiano Rotondo
Juan C. Hernández
Carlos Ocampo
Josep Pascual

Comité científico

Presidente CEA: César de Prada (UVA)

Vicepresidente CEA: Joseba Quevedo Casín (UPC)

Secretario CEA: Miguel Ángel Mañanas Villanueva (UPC)

Automar: Pere Ridao Rodríguez (UdG)

Bioingeniería: Javier Pérez Turiel (UVA)

Control Inteligente: José Manuel Andujar Márquez (UHU)

Educación en Automática: Fabio Gómez Estern (US)

Ingeniería de Control: Francesc Xavier Blasco Ferragud (UPV)

Modelado y Simulación de sistemas dinámicos: Miguel Ángel Piera Eroles (UAB)

Robótica: Pedro J Sanz Valero (UJI)

Sistemas de Tiempo Real: Alejandro Alonso Muñoz (UPM)

Visión por Computador: Enrique Alegre Gutiérrez (UL)

Comunicaciones

Automar

GRASPER: Un Proyecto Dirigido a Incrementar la Autonomía de la Manipulación Submarina
P. J. Sanz, J. J. Fernández, J. Pérez, A. Peñalver, J. C. García, D. Fornas, J. Sales, J. Bernabé, R. Marín

DISEÑO MECATRÓNICO DE UN ROBOT SUBMARINO DE BAJO COSTE PARA USO DOCENTE
Manuel Godoy, Ramón González, Francisco Rodríguez

SENSORES CIENTÍFICOS PARA UN VEHÍCULO AUTÓNOMO SUBMARINO
Narváez, F., González, H., García, E.

SISTEMA DE COMUNICACIONES PARA UN PLANEADOR AUTÓNOMO SUBMARINO
González, H., Narváez, F., García, E.

Automatización de vehículo marino no tripulado en superficie para guiado autónomo
Carlos Cano Espinosa, Santiago T. Puente Méndez, Fernando Torres Medina

Realtime AUV Terrain Based Navigation with Octomap
Guillem Vallicrosa, Albert Palomer, David Ribas, Pere Ridao

Bioingeniería

Asistencia de Robots Colaborativos para Procedimientos de Sutura vía Cirugía Mínimamente Invasiva
Enrique Bauzano Núñez, María Belén Estebanez Campos, Isabel García Morales, Víctor Fernando Muñoz Martínez

INCORPORACIÓN DE UN SISTEMA DE MINI-ROBOTS A LA CIRUGÍA LAPAROSCÓPICA DE INCISIÓN ÚNICA
María Cuevas Rodríguez, Irene Rivas Blanco, Enrique Bauzano, Jesús Gómez deGabriel y Víctor Fernando Muñoz

CONTROL FUERZA-POSICIÓN DE UNA CÁMARA ROBÓTICA PARA TÉCNICAS DE CIRUGÍA DE PUERTO ÚNICO
I. Rivas-Blanco, E. Bauzano, M. Cuevas-Rodriguez, P. del Saz-Orozco, V.F. Muñoz

Desarrollo de un Sistema Multimodal de Rehabilitación asistida por Robots
R. Morales, F. Badesa, N. Garcia-Aracil, J.M. Sabater, Miguel Almonacid

Aspectos de diseño y evaluación preliminar de la plataforma robotizada de neuro-rehabilitación PHYSIOBOT
Juan-Carlos Fraile Marinero, Javier Pérez Turiel, Carlos Rodríguez Guerrero, Rubén Alonso Alonso, Dra. Paulina Oliva Navarrete

Sistema Robótico Planar para Neuro-rehabilitación: Primeros Resultados con pacientes
F. Badesa, R. Morales, A.Llinares, N. Garcia-Aracil, Maria Garcia-Manzanares, Daniel Tornero, M. Oliva

CLASIFICACIÓN DE SEÑALES DE POTENCIAL DE ERROR A TRAVÉS DE UNA INTERFAZ GRÁFICA CON REALIMENTACIÓN DE FUERZAS
Joaquín López, Andrés Úbeda, Eduardo Iáñez, D. Planelles, José M. Azorín, Javier Gimeno, José A. Flores, José M. Climent

MOVIMIENTO BIDIMENSIONAL DE UN CURSOR MEDIANTE EL USO DE ARTEFACTOS EN SEÑALES ELECTROENCEFALOGRÁFICAS
Álvaro Costa, Eduardo Iáñez, Enrique Hortal, José M. Azorín, Alberto Rodríguez, Daniel Tornero, José A. Berná, José M. Cano

MRSeg - Herramienta interactiva para generar segmentaciones de referencia de imágenes médicas
F. Fumero, S. García, O. Núñez, J. Sigut, S. Alayón

ANÁLISIS ESPECTRAL NO LINEAL DEL EEG DE NIÑOS CON EPILEPSIA INTRATABLE
O. Portolés, R. Schroeder, M. Vallverdú, A. Voss, P. Caminal

Evaluación de métodos de segmentación de propósito general sobre imágenes de fondo de ojo
Omar Núñez, Francisco Fumero, Jose Sigut, Silvia Alayón.

Control Inteligente

REDES NEURONALES ARTIFICIALES EN UN REGULADOR PID PARA CONTROL AUTOMÁTICO DEL RUMBO DE UN BUQUE
José Luis Casteleiro-Roca, José Luis Calvo-Rolle, Matilde Santos

CONTROL HIBRIDO DIFUSO-DESLIZANTE PARA CONVERTIDOR DC/DC TIPO BOOST
Mónica L. Vásquez F., Edwar Jacinto G., Fredy H. Martínez S.

Social and Smart SANDS
Raúl Feliz Alonso, Eduardo Zalama Casanova, Jaime Gómez García-Bermejo, Bruno Apolloni

DESARROLLO DE UN CONTROLADOR BORROSO PARA EL CONTROL DE LA PROFUNDIDAD ANESTÉSICA MEDIANTE ARDUINO
Israel Riveron, Juan A. Mendez, A. Marrero, Ana Leon, Isabel Martin

Modelado borroso de una Pila de Combustible NexaTM 1.2 kW Ballard Power System
Antonio Javier Barragán Piña, Francisca Segura Manzano, José Manuel Andújar Márquez, Miguel Ángel Martínez Bohórquez

CONTROL DE ANESTESIA MEDIANTE LÓGICA DIFUSA CON DINÁMICA INVERSA
A. Marrero Ramos, J.A. Méndez Pérez, J.A. Reboso Morales, A. M. León Fragoso, I. Martín Mateos, E. Morell González

CONTROL AUTÓNOMO DEL SEGUIMIENTO DE TRAYECTORIAS DE UN VEHÍCULO CUATRIRROTOR
Tur Ortega, José Manuel; Santos Peñas, Matilde

MODELADO DE UN VEHICULO AEREO NO TRIPULADO MEDIANTE APLICACIÓN CONJUNTA DE TECNICAS PARAMETRICAS Y NEURONALES
Jesús Enrique Sierra García, Matilde Santos

Comparación de técnicas de optimización multi-objetivo clásicas y estocásticas para el ajuste de controladores PI
Helem Sabina Sánchez, Gilberto Reynozo-Meza, Ramon Vilanova, Xavier Blasco

Educación en Automática

Utilización del software AnyLogic en la enseñanza del Control Automático
José María González de Durana y Oscar Barambones

SISTEMAS DE DOCUMENTACIÓN Y ELABORACIÓN DE TEXTOS CIENTÍFICOS: UNA ASIGNATURA PARA APRENDER A PUBLICAR EN INGENIERÍA
Jorge L. Martínez, Anthony Mandow y Alfonso García-Cerezo

simLab: Laboratorio de simulación de la planta Festo
Silvia Alayón, Norena Martín, Omar Núñez, Francisco Fumero

Simuladores 3D y evaluación automática para prácticas de Automatización Industrial
Adolfo J. Sánchez del Pozo, David Muñoz de la Peña, Fabio Gómez-Estern

HERRAMIENTAS DE HARDWARE Y SOFTWARE LIBRE PARA LA IDENTIFICACIÓN EXPERIMENTAL, EL DISEÑO Y LA IMPLEMENTACIÓN DE CONTROLADORES PID
Roberto Sanchis Llopis, Silvia Estupiña Ariño

Laboratorio Virtual y Remoto de robots paralelos
Arturo Gil, Adrián Peidró, José María Marín, Óscar Reinoso, David Valiente, Luis Miguel Jiménez, Miguel Juliá

Benchmark de Control y Supervisión de Redes de Distribución de Agua
Gerard Sanz, Ramon Pérez

Aprendizaje colaborativo e interinstitucional en entornos virtuales
Antonio Javier Barragán Piña , José Manuel Andújar Márquez , Yolanda Ceada Garrido, Eloy Irigoyen Gordo , Fernando Artaza Fano , Vicente Gómez Garay

HERRAMIENTA DE SIMULACIÓN DEL COMPORTAMIENTO DE UN BIORREACTOR DISCONTINUO
Adrián Casimiro Álvarez, Javier de Pedro López, Antonio Coca Riega, Diego García-Ordás

PLATAFORMA ROBÓTICA DE BAJO COSTE Y RECURSOS LIMITADOS BASADA EN ARDUINO Y DISPOSITIVOS MÓVILES
A. Soriano, L. Marín, R. Juan, J. Cazalilla, A. Valera, M. Vallés, P. Albertos

Desarrollo de plantas industriales virtuales para el aprendizaje de la automatización
Rodríguez Díaz, Francisco; Moreno Úbeda, José Carlos; Pawlowski, Andrzej; Sánchez Molina, Jorge Antonio; López, Antonio

Herramienta basada en SCORM para la integración automática de Laboratorios Online en LMS
Ildefonso Ruano Ruano, Juan Gómez Ortega, Javier Gámez García, Elisabet Estévez Estévez

DESARROLLO DE UN EQUIPO DIDACTICO PARA LAS MATERIAS DE CONTROL DE PROCESOS DE LOS GRADOS DE INGENIERIA DE LA RAMA INDUSTRIAL Y DE SISTEMAS TIC
Teresa Escobet Canal, Antonio Escobet Canal, Juan Martínez Domence

APLICACIÓN DE CONTROL CON LABVIEW PARA EL SIEMENS S7-1200, EN RED LOCAL O INTERNET
Eduardo J. Moya de la Torre, Francisco J. García Ruíz, Israel Surribas Planas

SISTEMAS DE CONTROL POR COMPUTADOR - UNIFICACIÓN DE CRITERIOS
José Luis Casteleiro-Roca, José Luis Calvo-Rolle, Ramón Ferreiro García

Ingeniería de Control

Control Multivariable QFT para la Dinámica Longitudinal de un Vehículo Aéreo
Javier Joglar Alcubilla
Joaquín Aranda Almansa

Control de Velocidad mediante Paso de Pala para un Aerogenerador Experimental de Pequeña Potencia
Sergio Fragoso Herrera, Francisco Vázquez Serrano, Fernando Morilla García

Control Multivariable Descentralizado para un Aerogenerador Experimental de Pequeña Potencia
Sergio Fragoso Herrera, Francisco Vázquez Serrano, Fernando Morilla García

Modelado y control de un vehículo eléctrico mediante una estrategia de control predictivo basado en modelo
Agustín Pérez Castro, Julio Elias Normey Rico, José Luis Guzmán Sánchez, Manuel Berenguel Soria

Estrategia para la corrección de modelos no lineales variantes en el tiempo mediante estimación de parámetros: Aplicación al control predictivo y a la diagnosis de sistemas
Agustín Pérez Castro, José Sánchez Moreno, José Luis Guzmán Sánchez

Sistema experto para la determinación de referencias en el proceso de elaboración de aceite de oliva virgen
P. Cano Marchal, D. Martínez Gila, J. Gámez García, J. Gómez Ortega

DISEÑO DE LOS CONTROLADORES DE LOS SISTEMAS DE VACÍO Y TEMPERATURA DE UN REACTOR AGITADO
Ferran Babot Branzuela, Ramon Costa Castelló

LA ADMINISTRACIÓN DE RECURSOS APLICADA AL MANTENIMIENTO DE DISPOSITIVOS DE CAMPO
Ramón Ferreiro García, José Luis Calvo-Rolle, José Luis Casteleiro-Roca, Manuel Romero Gómez

Estimación Garantista de la Posición de un Quadrotor con GPS
Ramón A. García, Manuel G. Ortega, Francisco R. Rubio y Guilherme V. Raffo

DISEÑO DE CONTROLADORES POR ADELANTO PARA INVERSIÓN DE RETARDO NO REALIZABLE
C. Rodríguez, J.L. Guzmán, M. Berenguel, T. Hägglund, J.E. Normey-Rico

Restricciones en el Valor de los Enlaces de Comunicación en un Sistema de Control Coalicional
Francisco Javier Muros Ponce, Jose María Maestre Torreblanca, Encarnación Algaba Durán, Eduardo Fernández Camacho

ADMINISTRACIÓN DE RECURSOS REFORZADA MEDIANTE SUPERVISIÓN DE ELEMENTOS FINALES DE CONTROL
Ramón Ferreiro García, José Luis Calvo-Rolle, José Luis Casteleiro-Roca, Manuel Romero Gómez, Alberto Demiguel Catoria

Control No Lineal Iterativo de Modelos de Maniobra de Vehículos Marinos
Elías Revestido, M. Tomás-Rodríguez (b), Francisco J. Velasco

A ROBUST GRID SIDE CONVERTER CONTROL FOR WIND TURBINE SYSTEM BASED ON DOUBLE FEED INDUCTION GENERATOR
Oscar Barambones, José María González de Durana and Patxi Alkorta

Plataforma para monitorización y control distribuidos basada en 802.15.4
Conrado Arquer, Luis Orihuela, Francisco R. Rubio

DETECCION DE FALLOS CON VALIDACION PROBABILISTICA
Joaquim Blesa, Amalia Luque, Teodoro Alamo, Fabrizio Dabbene

CONTROL DE ACTITUD DE UN COHETE DE SONDEO ATMOSFERICO
Pau Manent, Joseba Quevedo, Bernardo Morcego

Desarrollo de una herramienta para el análisis de datos multi-criterio. Aplicación en el ajuste de controladores del tipo PID.
Alberto Pajares Ferrando, Francesc Xavier Blasco Ferragud, Gilberto Reynoso-Meza, Juan Manuel Herrero Dura

CONTROL DE POSICIÓN Y ORIENTACIÓN DE UNA PLATAFORMA OMNIDIRECCIONAL NO-HOLONOMA
Pablo Bengoa Ganado, Josu Larrañaga Leturia, Eloy Irigoyen

Validación experimental de estrategias de reducción del consumo de sensores inalámbricos en sistemas de control en red.
Ignacio Peñarrocha, Adrian Dinu, Roberto Sanchis

SISTEMA DE DETECCIÓN DE FALLOS BASADO EN TÉCNICAS DE ERROR ACOTADO Y UMBRAL DINÁMICO
José Manuel Bravo Caro

Diseño Óptimo Multiobjetivo de PIDs para el Control de Temperatura en Cavidades Microondas
S. Garcia-Nieto, G. Reynoso-Meza, F. Peñaranda-Foix, A. Borrell

Diseño e implementación en UAVs de un sistema de control de formación de vuelo basado en SMRC
Ignacio Ribelles, Alejandro Vignoni, Sergio García-Nieto, Jesús Picó

OBSERVATION OF THE INTERNAL STATES OF A PEMFC ANODE GAS CHANNEL
J. Luna, C. Batlle, C. Kunusch, J. Riera, M.L. Sarmiento-Carnevali, M. Serra

Modelado y Simulación de Sistemas Dinámicos

Modelo para la generación de datos de consumo energético residencial basado en un enfoque probabilístico
J.K. Gruber, M. Prodanovic

PROGRAMACIÓN DE CONTROLADOR PREDICTIVO EN ECOSIMPRO E IMPLEMENTACIÓN EN PLANTA DE DISTRIBUCIÓN DE OXÍGENO
Juan David Tejerina, Rubén Martí, Daniel Navia, César de Prada

PREDICTOR NO PARAMÉTRICO BASADO EN TÉCNICAS DE ERROR ACOTADO
José Manuel Bravo Caro

Modelo simplificado y orientado al control de sistemas de refrigeración
Guillermo Bejarano, Manuel G. Ortega, Francisco R. Rubio, Fernando Morilla

Estudio de la interacción del fuselaje y el movimiento de aleteo de las palas del rotor principal en un helicóptero
Salvador Castillo-Rivera, María Tomás-Rodríguez, G. Nicolás Marichal Plasencia, Ángela Hernández López

TENSIONES INDUCIDAS EN CABLES DE ALTA TENSIÓN
Miguel Angel Calafat Torres, Matilde Santos Peñas

SIMULACIÓN Y CONTROL DE COMPRESORES CENTRÍFUGOS USANDO ECOSIMPRO
Tania Rodríguez Blanco, Daniel Sarabia Ortiz, César de Prada Moraga, José Luis Morales

CONCURRENT AND DISTRIBUTED SYSTEMS ANALYSIS USING COLORED PETRI NETS
Jenaro Nosedal Sánchez, Olatunde Baruwa, Miquel Ángel Piera Eroles

DESARROLLO Y VALIDACIÓN EXPERIMENTAL DE UN MODELO DINÁMICO PARA UN VEHÍCULO ELÉCTRICO CON MOTORES EN LAS RUEDAS
David Marcos, Carlos Bordons, Johan Wideberg, Daniel A. Mántaras, Pablo Luque.

MODELO NARMAX DE PH EN UN FOTOBIORREACTOR TUBULAR DE MICROALGAS
I. Fernández, G.A. Andrade, J.L. Guzmán, M. Berenguel, D.J. Pagano

Identificación de un modelo pasivo para la grúa Inteco 3D Crane e implementación en EJS
Pablo Falcón, Antonio Barreiro, Miguel D. Cacho, Emma Delgado

Intercambio de Modelos entre Herramientas de Simulación con Functional Mock-up Interface (FMI)
Javier Bonilla, José Domingo Álvarez, Lidia Roca, Alberto de la Calle, Luis José Yebra, Francisco Rodríguez

Modelado del reactor químico del proceso BTO mediante redes neuronales artificiales
Gorka Sorrosal, Eloy Irigoyen, Cruz E. Borges, Ainhoa Alonso

ANÁLISIS DEL PROBLEMA DE EXPLOSIÓN DE ESTADOS EN SISTEMS DICRETOS BASADO EN EL CUBO RUBIK
Emilio Jimenez Macias, Francisco Javier Leiva Lazaro, Juan Ignacio Latorre Biel, Mercedes Perez de la Parte

MODELADO DE UN PROCESO DE EXTRUSIÓN EN ALIMENTACIÓN
Eduardo J. Moya de la Torre, F. Javier Antolín Jiménez, Alfonso Poncela Méndez,Oscar F. Calvo Martínez

Reducción de modelos en Biología Sintética: Modelado y simulación de ruido y variabilidad en expresión genética.
Yadira Boada, Alejandro Vignoni, Jesús Picó

Robótica

Desarrollo de una clase en Matlab para la simulación realista de robots seriales
R. Morales, A. Sanchez, F. Badesa, N. Garcia-Aracil, C. Perez, J.M. Sabater

GEOMETRIC HYBRID PATH PLANNING FROM THE ARTIFICIAL POTENTIAL FIELD METHOD
Cristian C. Rincón, Edwar Jacinto G. y Fredy H. Martínez S.

DISEÑO Y DESARROLLO DE ROBOT MÓVIL PARA TRANSPORTE Y DISPENSACIÓN DE MEDICAMENTOS
Roberto Pinillos, Salvador Domínguez, Eduardo Zalama, Jaime Gómez García-Bermejo, Ángel de Miguel Navares

Manipulación de Objetos con dos Dedos Usando Información Táctil
Andrés Montaño, Raúl Suárez

Búsqueda de prensiones con force-closure de objetos 2D articulados con 2 eslabones
Noé Alvarado Tovar, Raúl Suárez

SOLUCIÓN AL PROBLEMA DE SLAM EMPLEANDO SGD CON IMÁGENES OMNIDIRECCIONALES
David Valiente, Arturo Gil, Miguel Juliá, Lorenzo Fernández, Óscar Reinoso

CONTROL VISUAL EMBEBIDO PARA EL POSICIONAMIENTO AUTÓNOMO DE UN HEXACÓPTERO
Javier Pérez, Jorge Pomares, Gabriel J. García, F. Torres

LOCALIZACIÓN ASISTIDA POR GPS PARA ROBOTS MÓVILES EN CONFIGURACIÓN ACKERMANN DE RECURSOS LIMITADOS
Leonardo Marín, Ángel Soriano, Vicent Mayans, Marina Vallés, Ángel Valera, Pedro Albertos

IMPLEMENTACIÓN DE CONTROLADORES FUERZA Y POSICIÓN PARA UN MANIPULADOR PARALELO DE 3DOF
Jose I. Cazalilla, Miguel Corberán, M. Díaz-Rodríguez, Ángel Valera, Vicente Mata, Marina Vallés

Fusión sensorial de visión, fuerza y aceleración para el control de robots manipuladores con movimientos restringidos.
Alejandro Sánchez García, Javier Gámez García, Juan Gómez Ortega

Mejora de las Capacidades de un Robot Mini-Humanoide en la Tarea de Subir y Bajar Escaleras
Wilmer Cardenas Pilatuña, Félix Rodríguez Cañadillas, Alberto Jardón Huete

Control de posición/fuerza de un robot antropomórfico basado en control predictivo
J. de la Casa Cárdenas, J. Gámez García, J. Gómez Ortega

Hacia una arquitectura de exploración multirobot distribuida para entornos no controlados
Miguel Juliá, Oscar Reinoso, Luis Payá, Francisco Amorós, David Úbeda

DISEÑO DE UN DISPOSITIVO HÁPTICO DE BAJA RESOLUCIÓN BASADO EN SMA PARA APLICACIÓN EN TELEROBÓTICA
Francisco Javier Tormo, Jaime Masiá, Juan Ramón Rufino, José Ferri, Borja Mansilla

Cálculo automático de secuencias de ensamblado basado en una técnica de agrupación para la construcción de estructuras mediante equipos de robots
Álvaro Sempere, Ivan Maza y Aníbal Ollero

Sistemas de Tiempo Real

ESTUDIO DE FIABILIDAD DE UNA PLATAFORMA DE COMUNICACIÓN BASADA EN EL ESTÁNDAR 802.11
Francisco Cárdenas Capitán, José M. Fornés Rumbao, Francisco Rodríguez Rubio, Rafael Estepa Alonso

La recta de Chevysev en la aproximación automática a crítico de los reactores nucleares
J. L. Fernández Marrón,Juan Blázquez Martínez, José Miguel Barcala Riveira

MONITORIZACIÓN Y CONTROL REMOTO DE CONSUMO ENERGÉTICO MEDIANTE COMUNICACIÓN WIFI
Isidre Ortega Gómez, Sergio Zarza Sánchez, José Antonio Lázaro Villa, Rosa M. Fernández-Cantí

Plataforma de gestión de aplicaciones distribuidas de seguridad funcional con tolerancia a fallos
Aitor Agirre, Rafael Priego, Dario Orive, Elisabet Estévez, Jon Pérez, Marga Marcos

Diseño Basado en Componentes y generación de aplicaciones robóticas manipuladoras bajo el middleware de OROCOS
A. Sánchez García, E. Estévez, J Gómez Ortega, J. Gámez García

EVENT DRIVEN MIDDLEWARE FOR DISTRIBUTED SYSTEM CONTROL
Manuel Muñoz, Eduardo Munera, J. Francisco Blanes, Jose E. Simo, Gines Benet

Herramienta basada en MDE para sistemas orientados a servicio dinámicamente reconfigurables
Aintzane Armentia, Marga Marcos, Elisabet Estévez, Francisco Gómez Molinero

Reconfigurando aplicaciones de automatización industrial con FTT-MA
Mikel González, Adrián Noguero, Federico Pérez, Isidro Calvo

Implementación de un sistema Hardware-In-the-Loop para la simulación en tiempo real de pilotos automáticos para UAVs
J. Velasco, S. García-Nieto, G. Reynoso-Meza, J. Sanchis

Arquitectura de tiempo real para el control de actitud en un satélite experimental
Juan Zamorano, Ángel Esquinas, Peter Bradley, Daniel Brosnan, Jorge Garrido, Emilio Salazar, Alejandro Alonso y Juan A. de la Puente

Visión por Computador

EVALUATION OF SHAPE AND COLOR DESCRIPTORS BY USING BAG OF WORD TECHNIQUES WITH ONE VS ALL CLASSIFICATION
Fidalgo E., Fernández-Robles L., García-Ordás M., García-Olalla O., Alegre E.

Comparison of different Local Binary Pattern variants for material recognition using KTH-TIPS 2a dataset
Oscar García-Olalla, María Teresa García-Ordás, Laura Fernández-Robles, Diego García-Ordás, Enrique Alegre

Rotation Invariant Contour Points Descriptor Histogram for shape based image retrieval
María Teresa García-Ordás, Oscar García-Olalla, Laura Fernández-Robles, Diego García-Ordás, Enrique Alegre

ROBUSTNESS TO ROTATION IN PERCEPTUAL HASHING METHODS VIA DOMINANT ORIENTATION
Diego García-Ordás, Enrique Alegre, María Teresa García-Ordás, Oscar García-Olalla, Laura Fernández

COMBINING SURF WITH GLOBAL TEXTURE DESCRIPTORS FOR CLASSIFYING BOAR SPERM
Fernández-Robles, Laura; Alegre, Enrique; García-Ordás, María Teresa; García-Olalla, Oscar; García-Ordás, Diego; Fidalgo Fernández, Eduardo

Detección y estimación de la orientación del rostro del conductor basada en modelos 3D
Gustavo Peláez, Fernando García, José María Armingol, Arturo de la Escalera

Determinación del estado de batido de la pasta de aceituna empleando visión por computador
Diego Manuel Martínez Gila, Pablo Cano Marchal, Javier Gámez García, Juan Gómez Ortega

Construcción de mapas topológicos y estimación de trayectorias usando descriptores de apariencia visual global
Francisco Amorós, Luis Payá, Lorenzo Fernández, Óscar Reinoso, Mónica Ballesta, Miguel Juliá

OBJECT RETRIEVAL UNDER DIFFERENT ILLUMINATION DIRECTIONS USING INVARIANT LOCAL FEATURES
Gorgojo, Nuria; Fernández-Robles, Laura; Alegre, Enrique

Seguimiento de objetos sin marcas fiduciales con cámaras ToF
S. Satorres Martínez, J. Gómez Ortega, J. Gámez García, A. Sánchez García, J. de la Casa Cárdenas

SISTEMA DE ALERTA Y REGISTRO DE INFRACCIONES DE TRÁFICO
Nourdine Aliane, Javier Fernandez y Sergio Bemposta.

Análisis del descriptor VFH para el reconocimiento 3D de múltiples vistas de objetos domésticos
Francisco Martínez, Carlos M. Mateo, Pablo Gil, Fernando Torres

Control visuo-inercial de 6 grados de libertad para el posicionamiento aéreo de una cámara
Jesús González Villagómez, Manuel Vargas Villanueva, Francisco Rodríguez Rubio

GRASPER: Un Proyecto Dirigido a Incrementar la Autonomía de la Manipulación Submarina

P. J. Sanz, J. J. Fernández, J. Pérez, A. Peñalver, J. C. García, D. Fornas, J. Sales, J. Bernabé, R. Marín
Departamento de Ingeniería y Ciencia de Computadores
Av. Sos Baynat, s/n, 12071 Castellón de la Plana (España)
Universitat Jaume I
{sanzp, fernandj, japerez, penalvea, garciaju, dfornas, salesj, jbernabe, rmarin}@uji.es

Resumen

Este trabajo presenta la investigación en marcha del proyecto GRASPER, que representa uno de los tres sub-proyectos del proyecto nacional coordinado TRITON. El proyecto TRITON se enmarca en el contexto de investigación en robótica submarina de intervención, y en particular, el sub-proyecto GRASPER se centra en el desarrollo de las habilidades de manipulación necesarias para ello. Este trabajo presenta los desarrollos de investigación recientes para aumentar los niveles de autonomía de un sistema de manipulación submarina, ofreciendo también algunos resultados preliminares, desde el punto de vista de la mecatrónica, tanto de los sistemas de agarre como de las herramientas software implementadas al efecto. Así, se presenta la implementación de una nueva pinza que incluye sensores táctiles, y conexión a un sensor de fuerza que permitirá, junto con sistemas de visión y láser, la ejecución autónoma de operaciones de aproximación agarre y recuperación en el entorno submarino, incrementando la robustez y fiabilidad de estas acciones. Por otro lado, se muestra el funcionamiento del simulador 3D, llamado UWSim, desarrollado al efecto, que permite monitorizar las acciones en un mundo virtual, y que facilita la interacción humano-robot para especificar las acciones de manipulación a ejecutar.

Palabras Clave: Intervención autónoma submarina, simulación 3D, agarre guiado por información multisensorial, mecatrónica de garras robóticas.

1 INTRODUCCIÓN

El objetivo principal del proyecto TRITON es el uso de vehículos autónomos para la ejecución de tareas complejas de intervención submarina. Se trata de un proyecto nacional coordinado que, debido a su complejidad, requiere de la cooperación activa de tres grupos de investigación expertos en distintas áreas, y cada uno responsable del correspondiente sub-proyecto:

- "COMAROB", Robótica Cooperativa, (UdG).
- "VISUAL2", Percepción Multisensorial, (UIB).
- "GRASPER", Manipulación Autónoma, (UJI).

El proyecto se centrará en el uso de varios vehículos cooperando de una forma coordinada durante la ejecución de una misión, así como en la mejora de las capacidades de manipulación de un brazo robot (sub-proyecto GRASPER) que se ensambla en el vehículo autónomo de intervención Girona500 (sub-proyecto COMAROB) (véase la Figura 1).

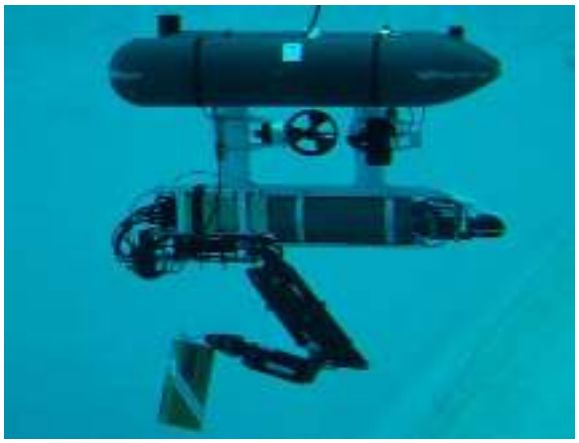


Figura 1. El AUV Girona500 (responsable UdG) equipado con el brazo robot (responsable UJI) en el área de carga. Se muestra la recuperación de una caja negra (escenario de validación 1 de TRITON).

Es importante señalar que TRITON (2012-2014) es una extensión directa del proyecto RAUVI (2009-2011) ya finalizado, coordinado también por las tres universidades citadas. Los escenarios elegidos para el desarrollo y validación experimental del proyecto son: (1) "Búsqueda y Recuperación" y (2) "Observatorios Permanentes". En (1) se abordará la recuperación de una caja negra típica de las aeronaves actuales y en (2) se realizarán operaciones de mantenimiento propias de las instalaciones submarinas permanentes.
El proyecto TRITON sería inviable de no ser, entre otras cosas, por el desarrollo tecnológico alcanzado en el anterior proyecto nacional coordinado (DPI2008-06548-C03) RAUVI [10] [3] y por el proyecto europeo (FP7-ICT-248497) TRIDENT [11].

Gracias a RAUVI se dispone de un prototipo de I-AUV, denominado *Girona 500*, a partir del cual, se pueden realizar mejoras funcionales y operativas sustanciales con un esfuerzo investigador asumible y realista.

En síntesis, los principales retos científico-tecnológicos asumidos por GRASPER son:

- (O1) Desarrollo e implementación de una interfaz de usuario y un simulador para facilitar la especificación de la intervención, incluyendo la caracterización 3D del agarre.
- (O2) Mejoras funcionales de la capacidad de agarre y manipulación, incluyendo la mecatrónica y la integración sensorial necesarias para ello.
- (O3) El control específico denominado "*visual free-floating manipulation*". Si en el proyecto RAUVI, se atacaba el problema de la intervención simplificadamente, desacoplando el control del brazo y del AUV, ahora se asume un nuevo reto en el que nuevas técnicas de control permitirán utilizar simultáneamente los grados de libertad del vehículo, además de los propios del sistema brazo-mano, facilitando así la intervención.

En definitiva, gracias entre otros factores al trabajo conjunto realizado por las tres universidades en los proyectos previos RAUVI y FP7-TRIDENT, se dispone del "*know-how*" necesario para afrontar lo que actualmente se considera por la comunidad científica un auténtico desafío tecnológico: Ejecutar de manera autónoma operaciones reales de intervención submarina, mediante robótica cooperativa, percepción multisensorial y capacidad autónoma de manipulación.

En síntesis, este trabajo presenta el estado actual de los avances hacia la manipulación autónoma submarina, que actualmente se desarrollan por el IRS-Lab de la UJI, mediante el simulador 3D UWSim [6] (relacionado con O1), las mejoras en la capacidad de acción-reacción de la manipulación combinando información sensorial basada en láser, visión y sensores táctiles (relacionado con O2), así como los progresos realizados en la implementación mecatrónica de los elementos de agarre (relacionado con O2). El objetivo O3 no se contempla en el presente trabajo.

1.1 TRABAJO RELACIONADO

En el contexto de la intervención submarina enfocada a la manipulación, proyectos anteriores como ALIVE [2] y SAUVIM [4] han representado importantes hitos para la Comunidad Científica, incrementando de manera significativa la autonomía en las operaciones de intervención, y reduciendo así de manera drástica los costes derivados de la logística en superficie a la que obligan los sistemas tele-operados. Es reconocido por la comunidad científica actual que el problema de la manipulación robótica en general, es uno de los puntos más críticos y a la vez más necesarios a la hora de abordar operaciones cualesquiera de intervención. Y, si este problema que encapsula otros muchos (e.g. actuadores, mecatrónica de las garras/manos, sensores, algoritmos de planificación y control, etc.) dista de estar bien resuelto en el contexto terrestre, debajo del agua y buscando su autonomía se convierte en un auténtico desafío. La tendencia actual es combinar información multisensorial para facilitar de manera autónoma el guiado de las acciones del robot en el agarre y manipulación [8], que ha permitido grandes progresos en el contexto del agarre submarino incrementando su nivel de autonomía [11].

Si proyectos anteriores como ALIVE, se centraban en la manipulación en paneles de la industria off-shore, o SAUVIM en la recuperación de objetos, GRASPER irá un paso más allá, facilitando a través del proyecto coordinado TRITON, la posibilidad de intervenir indistintamente en cualquiera de estos escenarios, promoviendo así la capacidad autónoma y multipropósito de las operaciones de intervención submarinas.

En la actualidad, la mayoría de las intervenciones de manipulación submarina son realizadas mediante ROVs (Remote Operated Vehicles), tele-operados por un piloto experto que controla el manipulador del ROV mediante algún dispositivo tipo "*joystick*". Casi todos los mecanismos de agarre de dichos manipuladores son pinzas de dos dedos acoplados, diseñadas en diferentes configuraciones (paralelas, garra, etc.) dependiendo del propósito de la intervención. Normalmente, la visión es la única realimentación sensorial para el operador, ya que en la actualidad apenas hay disponibles comercialmente dispositivos hápticos con realimentación de fuerza, que ayudan a incrementar la percepción del piloto durante la intervención facilitando así el control de la manipulación. En particular, uno de los dispositivos hápticos más conocidos es PREDATOR [9].

2 PROGRESOS EN LA MECATRÓNICA DE MANIPULACIÓN

En GRASPER, centrado en el punto de vista de la manipulación, el objetivo es aumentar el nivel de autonomía de las intervenciones autónomas submarinas intentando sustituir, o en algunos casos simplificar, las acciones llevadas a cabo por el operador. Para ello, es necesario dotar al sistema de manipulación de una mayor sensorización (véase la Figura 2, 3, 4).

El sistema de manipulación de partida de GRASPER es el manipulador submarino *Light-Weight ARM 5E* [1], uno de los resultados del proyecto antecesor RAUVI [10] [3], fruto del esfuerzo investigador con la empresa británica CSIP (ahora ECA Robotics). Este brazo posee 4 GDL, todos ellos de rotación, más la apertura/cierre de su pinza.

Tabla 1: Características de la UJIOne.

ESPECIFICACIONES	COMPONENTES PRINCIPALES
• 1 GDL (apertura/cierre) • Dimensiones: ○ Cerrada: 479.47mm (l.) x 130mm (h.) x 104.36mm (w.) ○ Abierta: 276.33mm (l.) x 130mm (h.) x 561.13mm (w.) • Peso en el agua: 2Kg • Peso en el aire: 3.93Kg • Capacidad de carga máxima: 2Kg • Profundidad máxima: 100m (50m con el sensor JR3) • Alimentación: 9-36 Vdc • Consumo eléctrico: 6W (250mA/24Vdc) • Material: Aluminio y acero inoxidable • Sensors: ○ 3-ejes-6-GDL Fuerza-Par (JR3) ○ 4 Flexiforce Táctiles, 1 en cada garra ○ 4 Galgas extensiométricas, 1 en cada garra • Controles: ○ Velocidad, posición y par-fuerza	• Cilindro de encapsulamiento • Garras • Engranaje de 40 dientes, modulo 1 • Sinfín, modulo 1 • Motor Dynamixel AX-F18 • Arduino Mega 2560 • Traco Power: 9-36Vdc/+-12Vdc • Traco Power: 9-36Vdc/+12Vdc • Sensor JR3 • JR3 PCB • Dynamixel PCB • Conectores submarinos Seacon • Juntas tóricas y retén • Sensores táctiles: ○ Flexiforce ○ Galgas extensiométricas • Interfaz con ROS: ○ UJIone_control_node ○ UJIone_sensor_node

La configuración de la pinza, en el que su apertura y cierre se realiza mediante un empujador axial, dificulta la medida de la fuerza y el par ejercidos sobre el efector final en cualquier eje al no ser fácil situar los sensores necesarios en esta configuración. Para resolver este problema se han explorado dos estrategias: (1) añadir cuatro garras a la configuración original (véase la Figura 2) que permiten disponer sobre ellas los sensores táctiles (Flexiforce) [5] y de fuerza (galgas extensiométricas), además de favorecer el agarre de objetos de mayor volumen como la caja negra considerada en los escenarios de validación (véase la Figura 1); (2) realizar una nueva pinza autocontenida, la *UJIOne*, que permite disponer en su base un sensor comercial de fuerza/par como el JR3, además de disponer de cuatro garras con el mismo propósito que en el primer caso (véase la Figura 3 y 4).

A continuación se describen los componentes principales de la *UJIOne*, donde se subsume la utilización y sensorización de las garras correspondientes a la primera aproximación comentada anteriormente, ya que ésta es directamente aplicable a la *UJIOne*.

2.1 DISEÑO MECÁNICO

La pinza *UJIOne* (véase la Figura 3) fue diseñada teniendo en cuenta los siguientes requisitos: robustez, fiabilidad y simplicidad (ver características en la Tabla 1). El cuerpo principal de la *UJIOne* está formado por un cilindro de encapsulamiento donde va alojada la electrónica de control y alimentación de la misma. Su forma cilíndrica le proporciona una elevada resistencia al colapso debido a la presión ejercida por el agua. El cilindro se cierra con dos tapas: la inferior, que sirve de acoplamiento para el sensor de fuerza/par JR3 y la superior, donde se alojan los conectores, soportando el peso del mecanismo de transmisión y las garras. Tanto el cilindro como las tapas están fabricados en aluminio para reducir su peso. La estanqueidad del conjunto se consigue mediante sendas juntas tóricas alojadas en cada una de las tapas y un retén alojado en la tapa superior por donde pasa el eje de transmisión del mecanismo de actuación. El sistema de actuación está formado por un motor que mueve un eje de transmisión con un sinfín en su extremo, que a su vez mueve dos engranajes que hacen girar sendos ejes sobre los que están fijadas las garras. La relación de reducción del motor con los engranajes permite actuar la *UJIOne* con un motor pequeño y económico. El mecanismo de agarre está formado por cuatro garras, que se pueden cruzar para hacer más estable el agarre de objetos de menor tamaño.

2.2 DISEÑO ELÉCTRICO

La *UJIOne* se alimenta externamente con un rango de tensión continua que puede ir de 9 a 36 voltios. Internamente posee dos convertidores DC/DC que adaptan dicha tensión de entrada a la adecuada para el motor y los sensores. La placa (i.e. circuito Arduino) se alimenta también de forma externa a través de su conexión USB. La *UJIOne* posee conectores sumergibles y conectables en mojado para el paso de las señales de alimentación, sensores y comunicación.



Figura 2. (izquierda) Sensor táctil *FlexiForce* adaptado a ambientes submarinos, (centro), medidores de tensión instalados en las garras, y (derecha) experimento de retroalimentación del control de la fuerza utilizando Galgas, integrado en el brazo robótico *ARM5E* (http://youtu.be/ng5ESy6Q68o).

2.3 SENSORES TÁCTILES

Con el objetivo de proporcionar a la UJIone una adecuada capacidad táctil, se ha realizado una primera validación experimental montando cuatro garras sobre la pinza original del *Light-Weight ARM 5E* con sensores táctiles tipo *FlexiForce* (véase la Figura 2). Hay que tener en cuenta, que los objetos de intereses suelen ser de un tamaño considerable con respecto al tamaño de la pinza, con lo que la zona final de la garra es la que posee mayor superficie de contacto. Estos sensores han sido adaptados para su uso en entornos bajo el agua. Ha sido necesario también desarrollar un sistema para leer los valores proporcionados por los sensores, de modo que el controlador de brazo robótico y los algoritmos de manipulación puedan ser conscientes del posible contacto de los dedos con cualquier objeto [5].

2.4 SENSORES DE FUERZA

Al igual que en caso de los sensores táctiles, con el objetivo de proveer a la *UJIOne* con percepción de fuerza, se ha realizado un primer experimento montando galgas extensiométricas sobre las garras diseñadas para el *Light-Weight ARM5E*. Tras varias pruebas, se comprobó que la zona más sensible para la medida de la fuerza era la más cercana al final de la curvatura de la garra (véase la Figura 2, centro), coincidiendo con su máxima deformación durante un agarre. En la Figura 2 (derecha) puede observarse el uso de estos sensores para el control de fuerza (http://youtu.be/ng5ESy6Q68o). Además, se muestra su uso durante aplicaciones de agarre de diferentes objetos, con la finalidad de mantener una fuerza de agarre constante y detectar un posible deslizamiento o fallo durante el agarre (http://youtu.be/VOLNBWfeoLs, http://youtube/c62FTTycxsQ), y para la detección de una posible colisión con el fondo y reaccionar ante tal evento (http://youtu.be/42ZklVwNaqcx).

Por otra parte, la *UJIOne* está concebida para poder ensamblar en su tapa inferior un sensor de fuerza/par JR3 comercial. Esta ubicación del sensor JR3 permite percibir las fuerzas y los pares ejercidos sobre el efector final en cualquier dirección.

2.5 ARQUITECTURA DE CONTROL Y COMUNICACIÓN

El control y comunicación de la *UJIOne* está implementado sobre una plataforma *Arduino*, la cual recibe los comandos de apertura/cierra de la pinza del PC de control principal a través de su puerto USB y envía las acciones de control a la placa de control del motor. El control del motor se realiza mediante una señal PWM y se puede controlar en posición, velocidad y fuerza. Además recibe las señales de realimentación procedentes de los diferentes sensores. La implementación de los diferentes algoritmos de medida y actuación se basan en una arquitectura modular mediante comunicación con ROS.



Figura 3. La pinza UJIOne. (izquierda) Diseño CAD. (centro) UJIOne despiezada (derecha) UJIOne ensamblada y test de estanqueidad en tanque de agua.



Figura 4: La pinza UJIOne: Construcción e integración mecatrónica de sus componentes.

3 DESARROLLO SOFTWARE

3.1 EL SIMULADOR UWSim

El simulador 3D submarino *UWSim* [6] es una herramienta "*open source*" para la simulación y supervisión de intervenciones submarinas. El *software* permite visualizar un escenario submarino virtual que puede ser configurado usando herramientas de modelado estándar. De este modo, se pueden añadir a la escena vehículos submarinos controlables, embarcaciones de superficie, manipuladores robóticos e incluso sensores simulados, siendo todos ellos accesibles externamente a través de interfaces de red. Esto permite integrar fácilmente la herramienta de simulación, incluida la visualización, con arquitecturas de control ya existentes, permitiendo simulaciones denominadas "*hardware-in-the-loop*".

Las principales características de *UWSim* son:

- Entorno configurable: La escena completa puede ser modificada y adaptada utilizando ficheros XML.
- Soporte para múltiples robots: Puede incluirse cualquier tipo de robot utilizando el formato estándar URDF.
- Sensores simulados: Existen más de 10 tipos de sensores diferentes simulados incluyendo cámaras, DVL, presión, rango, IMU, GPS o sónares *multibeam*. Además nuevos sensores están siendo desarrollados para explorar nuevas posibilidades en las intervenciones submarinas como proyector de luz estructurada, láser, o sensores USBL.
- Interfaces de red: Todos los sensores son accesibles mediante ROS. Además, existe la posibilidad de utilizar el uso de aplicaciones como Matlab y Simulink para el desarrollo de soluciones, a través de los denominados *ipc-bridge ROS stack o S-functions,* respectivamente. (Ver http://youtu.be/kEOanX2zL-A y http://youtu.be/LXEzUcLwMqk).
- Física: La física de contactos está soportada por *osgBullet*, y la física del agua está en estado experimental.
- Dinámica: *UWSim* permite la simulación de movimientos de objetos rígidos utilizando un modelo dinámico de estado-espacio en términos de variables de estado, representando velocidades lineales y angulares, así como las posiciones de los cuerpos.
- Widgets: Se pueden añadir *widgets* personalizables a la ventana principal.

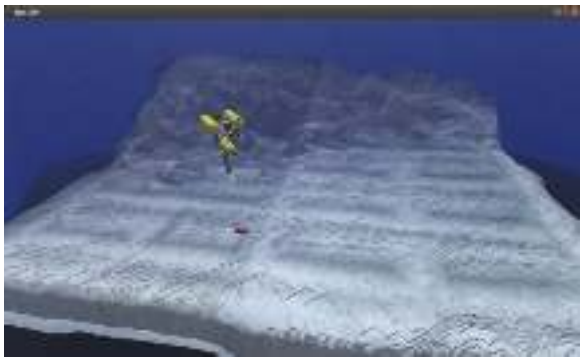


Figura 5. Reconstrucción de suelo submarino real en UWSim.

- Terrenos multi-resolución: Los modelos de los terrenos pueden ser cargados como objetos estándar, incluyendo aquellos compuestos por mallas complejas con texturas multi-resolución generados externamente con datos de batimetría e imágenes (véase la Figura 5).

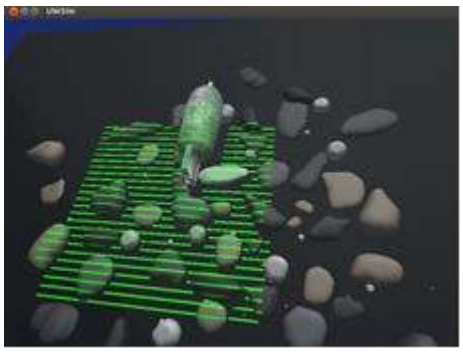


Figura 6. Proyector de luz estructurada en desarrollo para posterior reconstrucción 3D.

UWSim ha sido utilizado con éxito para simular la lógica de intervenciones submarinas y para reproducir experimentos reales a partir de los datos

guardados en los denominados *logs*. Además de esto, se ha desarrollado un módulo de benchmarking para UWSim, que permite comparar y medir resultados en todo tipo de condiciones. Este módulo ha sido utilizado para evaluar y comparar algoritmos de visión en casos de baja visibilidad y de control en diferentes condiciones de corrientes submarinas.
Para el proyecto GRASPER se planea aumentar la funcionalidad del simulador en la dirección de los objetivos del proyecto y mejorar la interacción con el usuario. Para ello se ha procedido al desarrollo del proyector de luz estructurada mencionado anteriormente para estudiar soluciones al problema de la falta de textura a la hora de recuperar 3D con cámaras estéreo (véase la Figura 6).
Esta característica permitirá comparar diversos patrones de luz estructurada y combinaciones de posición de cámara y proyector. Ofreciendo de esta forma, una herramienta con la que poder hacer benchmarking de reconstrucción del entorno de una forma económica y sencilla.
Además, se han dado pasos para mejorar la física del simulador y ofrecer respuestas a los contactos en cadenas cinemáticas, como el brazo del robot. Así mismo, se ha añadido la posibilidad de incluir objetos manipulables que formen las piezas básicas de un panel de intervención, como son válvulas, botones, y conectores, permitiendo la simulación de escenarios de la industria *off-shore*.
Este software se ofrece como *open source*, proporcionando una herramienta única dentro de la comunidad de robótica submarina, donde predominan los simuladores comerciales orientados a pilotos de ROV.

3.2 ESPECIFICACIONES DE LA INTERFAZ DE USUARIO

Con el objetivo de mejorar la experiencia de uso del simulador UWSim, éste se ha integrado en una interfaz gráfica. De este modo, muchas de las opciones que se establecen en los ficheros de configuración, pueden modificarse mediante el uso de diferentes pantallas y paneles o de un menú específico. Además, el uso de la interfaz de usuario permite especificar la tarea a realizar por el robot.
Inicialmente, la interfaz reconoce dos tipos generales de operación: "reconocimiento del área" e "intervención". Dentro del tipo "intervención", definimos las tareas que se definen actualmente en el proyecto al que está circunscrito la interfaz. Estas tareas son, por el momento: el agarre de un objeto (e.g. la caja negra de un avión comercial) y la operación sobre un panel de intervención submarina (e.g. abrir-cerrar una válvula o conectar un cable). En un futuro, la interfaz podría implementar nuevas tareas.
Si el usuario selecciona "reconocimiento de área", la interfaz muestra en los paneles sólo las opciones relacionadas al respecto: especificar los puntos intermedios (*waypoints*) del recorrido a realizar, la altura y la profundidad, etc. Una vez se han especificado todas las opciones, se genera un archivo con todos los parámetros, que puede validarse en el simulador o puede descargarse al robot, utilizando el módulo de comunicación ROS (esto es, utilizando paso de mensajes dentro de la red) [12].
Dado que asumimos la existencia de una operación de "reconocimiento de área" previa a la "intervención", suponemos que tenemos un conjunto de parámetros (i.e. corrientes, visibilidad, etc.) que se pueden englobar dentro de un módulo que hemos denominado "configuración de la escena".



Figura 7. La interfaz de usuario permite cargar tanto mosaicos como escenas completas.

Este módulo se ejecuta siempre, independientemente del tipo de "intervención" a realizar, y permite seleccionar la configuración específica del vehículo (i.e. el vehículo utilizado en la primera fase puede que sea distinto al segundo), y carga el foto mosaico obtenido en el "reconocimiento del área" (véase Figura 7).
Así pues, si el usuario selecciona una "intervención sobre panel", y tras la carga de la configuración de la escena, el usuario puede navegar a través de la escena en busca de su objetivo (i.e. el panel). Cuando el vehículo se encuentra cerca del panel, y asumiendo que tenemos el archivo CAD del panel, la interfaz aplicará técnicas de Realidad Virtual y Realidad Aumentada. De este modo, conseguiremos mostrar todas las características del panel con mayor detalle del que se podría obtener de forma habitual, pues las condiciones de visibilidad no suelen ser las más idóneas. Finalmente, el usuario podrá seleccionar la acción predefinida más conveniente: abrir-cerrar una válvula o conectar un cable. A pesar de que sólo existan estas acciones predefinidas, éstas pueden modificarse o crear nuevas desde un menú específico. De un modo similar, cuando el usuario selecciona "recuperación de caja negra", la interfaz mostrará en los módulos de la interfaz, las opciones

relativas a este tipo de misión. En este caso, los parámetros más importantes son los relativos a la especificación del objeto de interés y los relacionados con el agarre.
Actualmente la interfaz de usuario se encuentra en proceso de implementación, en conexión con el simulador UWSim presentado anteriormente. No obstante, se puede observar parte del trabajo en progreso en la figura 9 del siguiente apartado.

4 RESULTADOS PRELIMINARES: RECONSTRUCCIÓN 3D Y AGARRE

El sistema se compone de un brazo robótico al que se le ha agregado un láser emisor de luz estructurada en el antebrazo, así como una cámara que se encuentra enfocando a la escena [7] (véase la figura 8). Para la reconstrucción del objeto a agarrar, se mueve el codo del brazo robótico a una velocidad constante, hasta que se ha escaneado la escena completa. Durante este proceso, se ejecuta en paralelo un algoritmo de procesado visual. En las imágenes proporcionadas por la cámara, el algoritmo se encarga de segmentar el láser del resto de la imagen. Utilizando las detecciones, se construye una nube de puntos 3D de la escena.
Una vez reconstruida la escena, utilizando el simulador *UWSim*, se muestra la nube de puntos resultante así como una instantánea tomada por la cámara. En este momento, el usuario se encarga de seleccionar sobre esta instantánea dos puntos de agarre antípodos.



Figura 8. Escenario de validación: tanque de agua con el sistema mecatrónico formado por un brazo robot (*Light-Weight ARM5E*), pinza y distintos sensores.

A continuación, un algoritmo se encarga de buscar cuál es la mejor posición y orientación para realizar el agarre, teniendo en cuenta tanto los puntos seleccionados por el usuario, como las restricciones del brazo y las características del objeto a agarrar. Un modelo CAD del efector final del brazo robótico se muestra en la posición y orientación seleccionada sobre la nube de puntos construida anteriormente. Utilizando el simulador, el usuario puede moverse por la nube de puntos con el efector final sobre ella, y decidir el agarre más conveniente o, si se estima oportuno, incluso reiniciar el proceso seleccionando dos nuevos puntos de agarre.

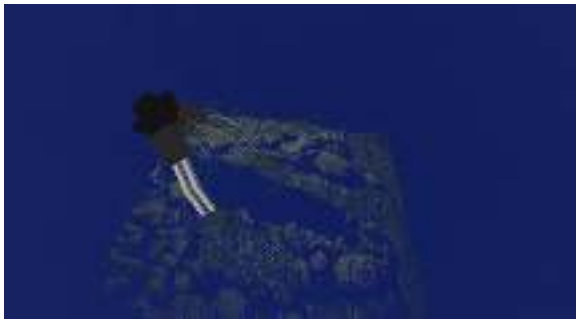


Figura 9. Mejor determinación del agarre. La representación virtual 3D del efector final del robot se muestra sobre la nube de puntos 3D.

Finalmente, una vez el agarre ha sido correctamente especificado, comienza la ejecución del mismo de manera autónoma. El brazo alcanzará la posición y orientación especificadas, y en ese momento la pinza comenzará a cerrarse, hasta que los sensores táctiles que se encuentran en ella detecten que el objeto ha sido agarrado con suficiente fuerza. A continuación, el brazo vuelve a una posición segura predefinida con el objeto ya recuperado.

5 CONCLUSIONES

El proyecto TRITON se enmarca en el contexto de investigación en robótica submarina de intervención, donde se pretende alcanzar un desarrollo tecnológico muy cercano a las necesidades reales del usuario final, facilitando así la potencial transferencia tecnológica que pudiera derivarse de él. En concreto, el sub-proyecto GRASPER, aquí presentado, se centra en el desarrollo de las habilidades de manipulación necesarias para ello. En la actualidad, existe una gran actividad investigadora en este contexto, buscando incorporar mayores niveles de autonomía en las misiones de intervención, que incluyan algún tipo de interacción física con el entorno del robot. Sin embargo, si la manipulación robótica autónoma en tierra sigue siendo una línea abierta de investigación con muchos problemas complejos por resolver, la situación es mucho más crítica debajo del agua donde actualmente los sistemas robóticos de manipulación son tele-operados por un usuario experto desde la superficie, utilizando algún tipo de cable umbilical. Sólo unos pocos sistemas submarinos han demostrado hasta la fecha capacidades de manipulación sin cable umbilical alguno. Entre estos sistemas se encuentran los proyectos, recientemente finalizados, RAUVI y FP7-TRIDENT, y gracias al "*know-how*" generado en

ellos se está progresando actualmente con GRASPER (véase TRITON). En particular, se han presentado avances recientes relacionados con la mecatrónica de una garra sensorizada, nuevas estrategias para la planificación del agarre 3D, incluyendo además el desarrollo de un simulador 3D (UWSim) que facilita enormemente las tareas de identificación y especificación del agarre, así como la monitorización de las ejecuciones de las distintas acciones de manipulación, entre otras funcionalidades.

Puntualicemos para terminar, que actualmente se está investigando un método alternativo de especificación del agarre. Este enfoque utiliza la reconstrucción del escenario para poder extraer la forma primitiva aproximada del objeto con técnicas de procesamiento de nubes de puntos 3D y algoritmos de aproximación RANSAC (RANdom SAmple Consensus). Con este enfoque no es necesario introducir un par de puntos de agarre, aunque la intervención del usuario sigue siendo necesaria para validar el agarre obtenido de forma autónoma. Estos experimentos actualmente están limitados a objetos que tienen una determinada geometría (e.g. en los experimentos actuales se utiliza una vasija con forma cilíndrica) y se deben ajustar ciertos parámetros que limitan su autonomía. La idea es avanzar hacia mayores niveles de autonomía en el agarre y la manipulación.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad, DPI2011-27977-C03 (proyecto TRITON), por el Séptimo Programa Marco de la Comisión Europea bajo el acuerdo de subvención ICT-248497 (Proyecto TRIDENT) y por la Fundación Caixa Castelló-Bancaixa, proyecto PI·1B2011-17.

Referencias

[1] Light-weight ARM 5E Manipulator ARM, ECA ROBOTICS S.A.S. http://www.eca-robotics.com/ftp/ecatalogue/539/ARM_5E_Light-Weight.pdf.

[2] J. Evans, P. Redmond, C. Plakas, K. Hamilton, and D. Lane, "Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation," en OCEANS 2003. Proceedings, vol. 4, 2003, pp. 2201–2210 Vol.4.

[3] J. J. Fernández et al., "Proyecto RAUVI: Historia y Resultados", XXXIII Jornadas de Automática, Vigo, Spain, 2012.

[4] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions AUVs," Ocean Engineering, vol. 36, no. 1, pp. 15 – 23, 2009, autonomous Underwater Vehicles. Disponible en línea: http://www.sciencedirect.com/science/article/pii/S002980180800173X.

[5] Peñalver, A, Prats, M, Fernández, JJ, Marín, R, Sanz, P.J., "Puesta a punto de sensores táctiles y adaptación a una mano robótica sumergible", XXXIII Jornadas de Automática, Vigo, Spain, 2012.

[6] Prats, M.; Pérez, J.; Fernandez, J.J.; Sanz, P.J., "An open source tool for simulation and supervision of underwater intervention missions," Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pp.2577-2582, 7-12 Oct. 2012, Vilamoura, Algarve, Portugal.

[7] Prats, M.; Fernandez, J.J.; Sanz, P.J., "Combining template tracking and laser peak detection for 3D reconstruction and grasping in underwater environments," Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pp.106,112, 7-12 Oct. 2012, Vilamoura, Algarve, Portugal.

[8] Mario Prats, Ángel P. del Pobil, Pedro J. Sanz. "Robot Physical Interaction through the combination of Vision, Tactile and Force Feedback". Springer Tracts in Advanced Robotics, Volume 84, 2013.

[9] PREDATOR Force Feedback Manipulator, Kraft TeleRobotics, Disponible on-line: http://www.krafttelerobotics.com/products/predator.htm.

[10] P.J. Sanz, M. Prats, P. Ridao, D. Ribas, G. Oliver and A. Orti, "Recent progress in the RAUVI Project. A reconfigurable Autonomous Underwater Vehicle for Intervention", 52- th International Symposium ELMAR-2010, pp. 471-474, Zadar, Croatia, Sept-2010.

[11] P. J. Sanz, R. Marin, J. Sales, G. Oliver, and P. Ridao, Editors. "Recent advances in underwater robotics for intervention missions: Sóller harbor experiments". Spain, Low Cost Books, 2012.

[12] www.ros.org

DISEÑO MECATRÓNICO DE UN ROBOT SUBMARINO DE BAJO COSTE PARA USO DOCENTE

Manuel Godoy, Ramón González, Francisco Rodríguez
Grupo de Automática, Electrónica y Robótica, Campus de Excelencia Internacional Agroalimentario (CEIA3), Universidad de Almería, Ctra. de Sacramento s/n, 04120, Almería
Email: magodma@gmail.com, rgonzalez@ual.es, frrodrig@ual.es

Resumen

Este artículo se enmarca en el contexto de la enseñanza de materias de posgrado relacionadas con la robótica. En particular, siguiendo un enfoque mecatrónico se ha diseñado, ensamblado y programado un robot submarino para explicar a los alumnos la cinemática, la dinámica, los grados de libertad, y la programación de este tipo de robots. Nótese que este tipo de robots no son nada habituales en los cursos de robótica actuales, pero en cambio cada vez son más utilizados en aplicaciones reales. Se muestran pruebas realizadas con éxito en condiciones reales.

Palabras Clave: Robot móvil, Teleoperación, Arduino, Educación.

1 INTRODUCCIÓN

Tradicionalmente las materias relacionadas con la robótica abordan desde un punto de vista práctico (sesiones de laboratorio) la utilización de robots manipuladores y robots móviles desplazándose a nivel del suelo mediante ruedas u orugas [1, 2]. Sin embargo, sistemas robotizados tales como robots aéreos, robots marinos y submarinos están menos presentes en dichas materias.

Este trabajo se enmarca en el contexto de la enseñanza de materias de posgrado relacionadas con la robótica, en particular, se centra en los aspectos prácticos de dichas asignaturas y en la utilización de robots submarinos para mostrar a los alumnos esta aplicación de los robots móviles. En este caso, el robot presentado aquí se ha utilizado en la asignatura de Robótica Industrial del Máster en Informática Industrial de la Universidad de Almería, para explicar a los alumnos las etapas de un proyecto siguiendo el paradigma de la mecatrónica, la cinemática y dinámica de los robots submarinos, así como la forma de controlar dichos robots.

Los vehículos submarinos no tripulados constituye un campo de investigación bastante activo [3]. El interés por conocer las profundidades de lagos, ríos, mares y océanos ha sido muy elevado a lo largo de la evolución humana. Los primeros vehículos sumergibles eran meras herramientas para exploraciones subacuáticas. Posteriormente, los vehículos submarinos se han ido desarrollando hasta tener submarinos de mayor autonomía, mayor seguridad y mayor número de tripulantes. Principalmente se ha investigado en vehículos submarinos militares, pero también se han desarrollado vehículos submarinos para uso civil, para exploraciones e investigaciones o de ocio [3]. Un claro ejemplo es el vehículo *Odyssey* utilizado, entre otras aplicaciones, para investigar los restos del Titanic [4] (Figura 1). A nivel nacional especial mención es para el Grupo Automar de CEA [5] y la Plataforma Tecnológica Marítima [6].



Figura 1. Vehículo submarino *Odyssey* [4]

El resto del artículo se ha estructurado de la siguiente forma. En la Sección 2 se detalla el desarrollo del robot submarino. En la Sección 3, se discuten las pruebas reales realizadas. Finalmente se presentan las conclusiones.

2 DESARROLLO DEL ROBOT SUBMARINO

En el contexto de este trabajo se ha seguido un paradigma mecatrónico, esto es, se ha enfocado el desarrollo del robot submarino siguiendo los principios de la mecánica, electrónica y control automático de forma coordinada [7]. En la Figura 2 se muestra la interacción entre las diversas fases de desarrollo del robot móvil.

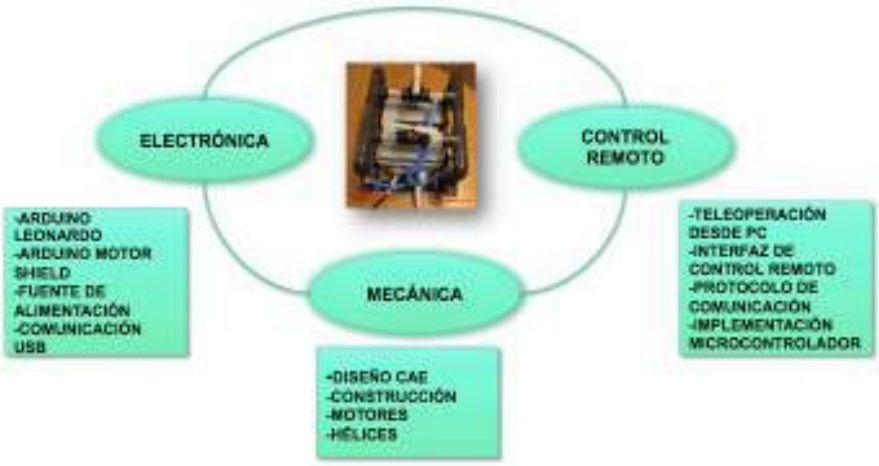


Figura 2. Estrategia seguida para desarrollar el robot submarino propuesto.

2.1 MECÁNICA

A la hora de realizar el diseño de un robot hay que tener en cuenta la funcionalidad y el medio en el que va a operar. En el caso concreto del medio acuático hay que considerar aspectos tales como la presión, la densidad del agua, la corrosión y la estanqueidad, entre otros.

En este punto se describe en líneas generales las características de diseño del robot submarino. El primer concepto importante a la hora de diseñar un robot submarino es que el peso del mismo tiene que estar cerca del elemento neutro o cerca de la flotabilidad, por lo tanto, la densidad del vehículo debe ser ligeramente superior a la del agua. En particular, el robot debe de cumplir una serie de especificaciones:

- Bajo coste (< 200€).
- Peso reducido (< 1 kg).
- Tamaño reducido (< 250 x 150 x 100 mm).

Nótese que estas especificaciones fueron seleccionadas fundamentalmente con el propósito de desarrollar varios robots y aplicarlos en labores docentes. En este sentido, se estableció un coste asequible con unas dimensiones reducidas para poder utilizar el robot en un entorno tipo laboratorio sin requerir una gran "pecera" donde navegue el robot.

Como se puede observar en la Figura 3, se ha desarrollado una estructura típica para un robot submarino, con dos motores de empuje y dos motores para el ascenso/descenso. Configuraciones similares se pueden encontrar en [3, 8] y en las referencias que ahí aparecen.

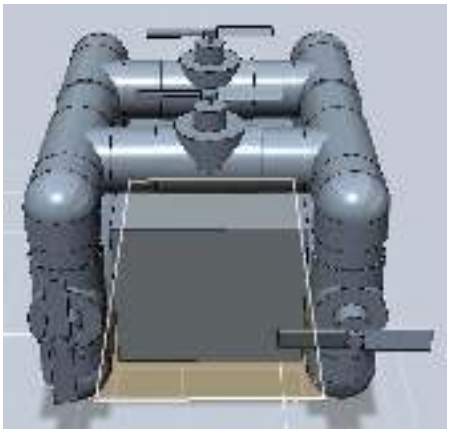


Figura 3. Diseño mecánico del robot submarino. Nótese los cuatro propulsores.

Una vez realizado el diseño del robot, se procedió a la construcción del mismo. Para cumplir con los requisitos iniciales, la estructura se ha construido con tuberías de PVC de 16 mm de diámetro, cumpliendo así los requisitos de bajo coste y peso.

En la base del robot se ha instalado una caja estanca en la cual poder introducir el microcontrolador y las tarjetas controladoras. También se han añadido 4 tuberías de PVC de 3 vías dedicadas a albergar los 4 motores. En la Figura 4 se puede observar además las hélices seleccionadas. Inicialmente se probaron hélices de metal de un diámetro de 1.5 centímetros y 3 aspas, pero no se obtuvo el resultado esperado (propulsión insuficiente), seleccionado finalmente hélices de PVC con un diámetro de 4 centímetros y 2 aspas.

Respecto a la flotabilidad, se optó por colocar una serie de placas de acero en el interior de la caja contenedora y placas de aluminio en el exterior para aportar peso extra al robot. Un aspecto importante fue el instalar dichas placas

manteniendo el centro de gravedad en el centro geométrico del robot para evitar posibles vuelcos u otras situaciones indeseadas.

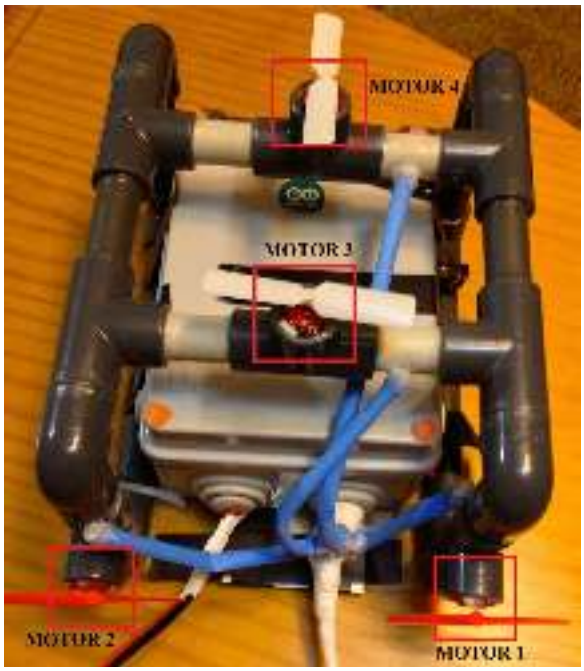


Figura 4. Estructura final del robot y situación e identificación de los motores.

2.2. ELECTRÓNICA

En esta sección se analizan los componentes hardware y electrónicos utilizados para controlar los motores del robot submarino. Tras analizar los dispositivos que ofrece el marcado (PC104 [11], Raspberry Pi [12], etc.) se ha optado por utilizar la tecnología Arduino debido principalmente a las siguientes características [9]:

- Código abierto (*Open-hardware*),
- Bajo coste (< 25€),
- Modularidad,
- Gran comunidad de usuarios,
- Tamaño reducido (7.5x5.5 cm),
- Comunicación USB,
- Peso reducido,
- Señales PWM (para controlar sistemas analógicos con señales digitales).

Para poder realizar el control de los motores se han adquirido dos tarjetas controladoras de motores, en este caso, *Arduino Motor Shield REV3* (Figura 5).

Los motores adquiridos son los motores A-max 12 de Maxon [10], que constituyen un solución compacta y con baja inercia. Algunas de las características son: peso de 12 gramos, 12 milímetros de diámetro de motor con escobillas, produciendo una potencia de 0.5 Watios, velocidad de 12100rpm y un voltaje nominal de 12 Voltios, características suficientes para las especificaciones pretendidas.

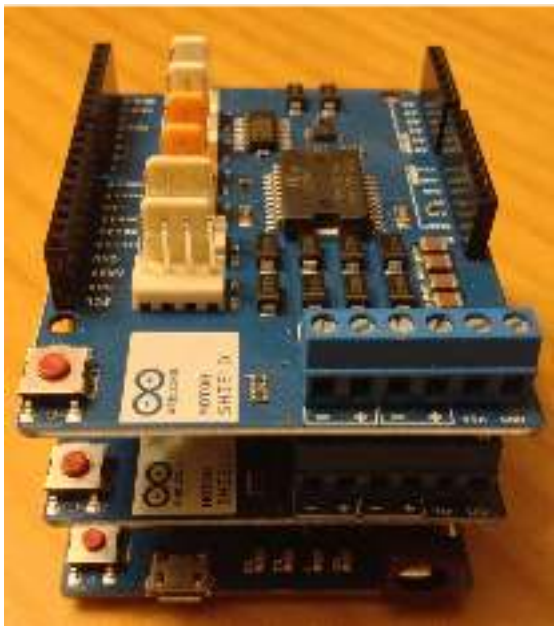


Figura 5. Ensamblaje de las dos controladoras Arduino Motor Shield y Arduino Leonardo.

Para poder operar en el agua se han llevado a cabo una serie de tratamientos para que los componentes electrónicos no se vean afectados. En primer lugar los motores se han tratado con resina epoxi. Los cables de las conexiones se han introducido en aislantes rellenados con silicona para evitar la acumulación de agua. Finalmente la caja contenedora del microcontrolador se ha tratado para poder conectar el microcontrolador con el exterior evitando la entrada de agua en el interior mediante el uso de silicona, material aislante y pegamento de PVC (Figura 6).



Figura 6. Orificio de comunicación de los componentes eléctricos con el exterior.

2.3 TELECONTROL DEL ROBOT

En este sección se detalla la estrategia de control remoto implementada para dirigir el robot submarino.

Para el correcto control del robot submarino es necesaria la implementación de un protocolo de comunicación, ya que el objetivo del proyecto es poder controlar los motores instalados en el robot, pudiendo modificar la velocidad de cada motor en el momento deseado como se puede comprobar en la Figura 7.

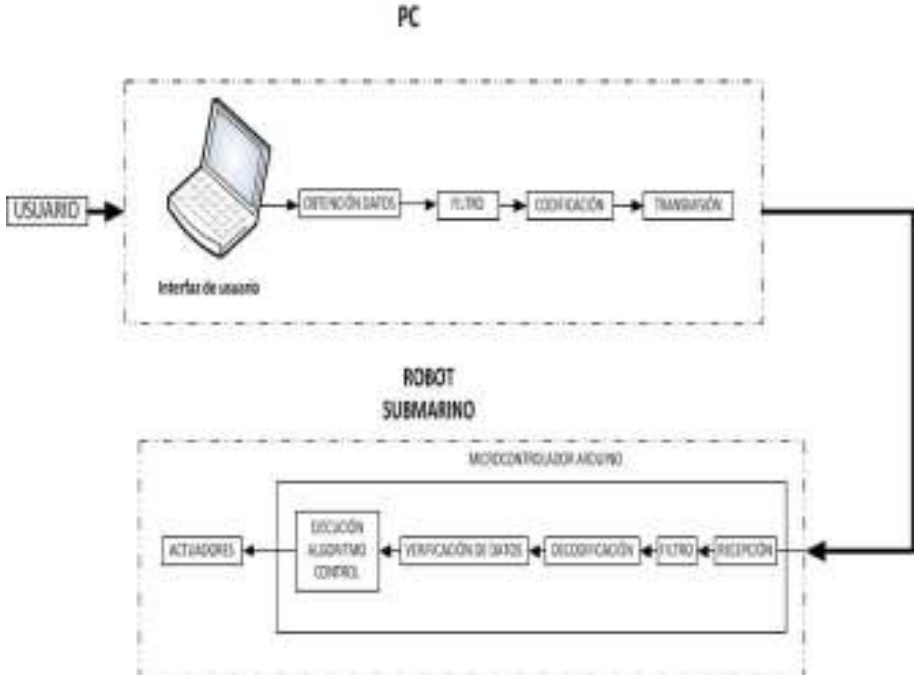


Figura 7. Estrategia de control remoto, comunicación PC-Robot submarino (Arduino).

En el proyecto desarrollado se ha optado por el uso de un protocolo de comunicación síncrono orientado a bit, para comunicar la información desde el PC al hardware empotrado en el robot submarino (Sistema Arduino). Los mensajes inherentes a una sesión de comunicación se envían en línea hacia la estación destinataria con un formato denominado marco. Cada marco está constituido los siguientes campos: flag de inicio, campo de información y flag de fin.

En la Figura 8 se puede observar las estructura del marco implementado.

Flag Inicio	Campo de información	Flag Fin

Figura 8. Estructura del marco.

Este marco se envía cada vez que se realiza una modificación de la información, en primer lugar, el microcontrolador Arduino recibe el marco completo comenzando a interpretarlo. El primer campo que se encuentra es el flag de inicio (I), indicando que a partir de dicho campo comienza la información. En el campo de información se encuentran los datos para los cuatro motores (Figura 9).

Tras el carácter M se encuentra el indicador de motor, a continuación el sentido de giro del motor y finalmente la velocidad del motor comprendida entre 000 y 255 (Figura 10).

CAMPO DE INFORMACIÓN

M	1	M	2	M	3	M	4

Figura 9. Campo de información.

INFORMACIÓN DE MOTOR

M	i	R	V	V	V

Figura 10. Información de motor.

Ésta sería la información correspondiente a un motor, donde: Mi (i= A,B,C,D) es el indicador del motor, R = Reverse, es un valor binario (0 ó 1) con el cual se indica el giro del motor y VVV, a campo que contiene la información de la velocidad del motor, comprendida entre 000 y 255.

Una vez procesada la información del marco, dicha

información se procesa y se envía la correspondiente a los motores. El marco completo (para los 4 motores) se muestra en la Figura 11.

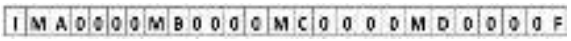


Figura 11. Marco completo para los 4 motores

En la Figura 12 se presenta el algoritmo implementado en el microcontrolador Arduino, programado en entorno de desarrollo Arduino (basado en *Processing*) [9]. Este programa recibe el marco de información y posteriormente realiza las acciones pertinentes. La comunicación PC-Arduino se realiza mediante puerto serie (USB).

En el PC utilizado para controlar de forma remota al robot se ha implementado un programa en *Microsoft Visual Studio 2010* © en el lenguaje de programación *Visual Basic .NET* (Figura 13).

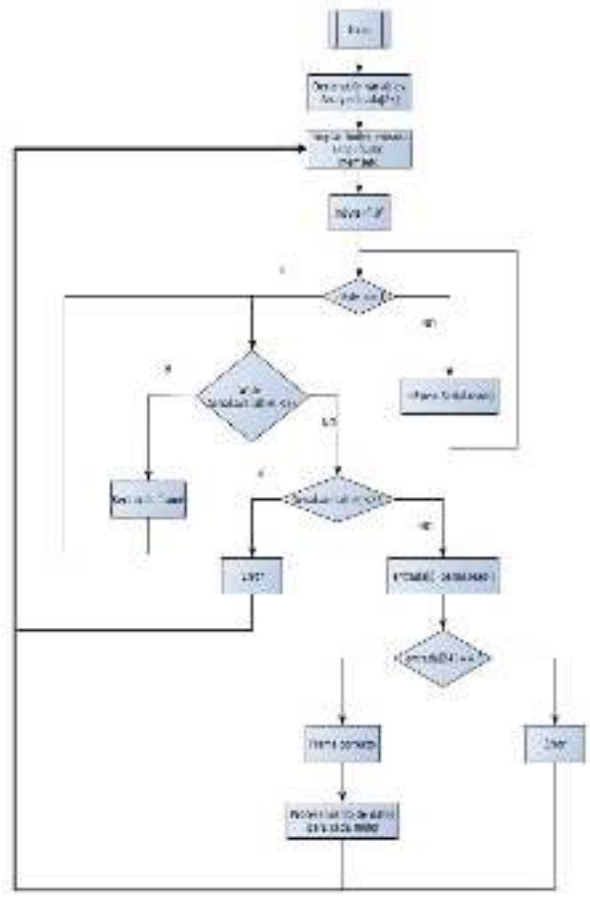


Figura 12. Algoritmo implementado en Arduino.

En la Figura 14 se puede comprobar el algoritmo implementado en la interfaz de control remoto. En primer lugar se comprueba la conexión puerto serie, una vez realizada la conexión se puede interactuar con la interfaz de control a través de los dos submódulos desarrollados (Control de movimientos y control individual) accionando los diferentes botones de cada módulo o a través de teclado, creando en cada ocasión un marco que se enviará a la unidad de control Arduino.

Toda la información obtenida durante la ejecución se muestra en la interfaz gráfica y al finalizar se almacena en disco para su análisis posterior.



Figura 13. Interfaz gráfica para control remoto (Control individual de cada motor, control de movimiento de la plataforma completa e imágenes de cámara web).

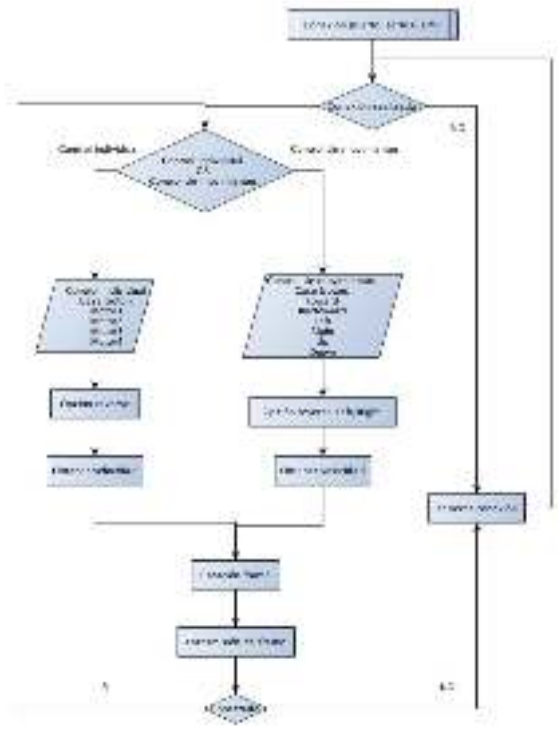


Figura 14. Algoritmo para la interfaz de control.

3 RESULTADOS

Se han realizado diversas pruebas tanto de algunos componentes por separado como del sistema completo del robot submarino para comprobar el correcto funcionamiento de éste.

3.1 FLOTABILIDAD

Las pruebas de flotabilidad se han llevado a cabo para encontrar el punto neutro de flotabilidad del robot submarino en el agua (elevación vs hundimiento).

Se han realizado pruebas con materiales como poliexpan, boyas náuticas, pero el resultado obtenido fue negativo ya que el robot flotaba en exceso por lo que se retiraron dichos materiales. Para solucionar este problema y tal y como se ha comentado anteriormente, se situaron diferentes placas de acero en el interior de la caja donde se encuentra el hardware de control y placas de aluminio en la propia estructura. Como se puede comprobar en la Figura 15 el robot se encuentra en una posición neutra, pudiéndose sumergir o ascender con los motores superiores.

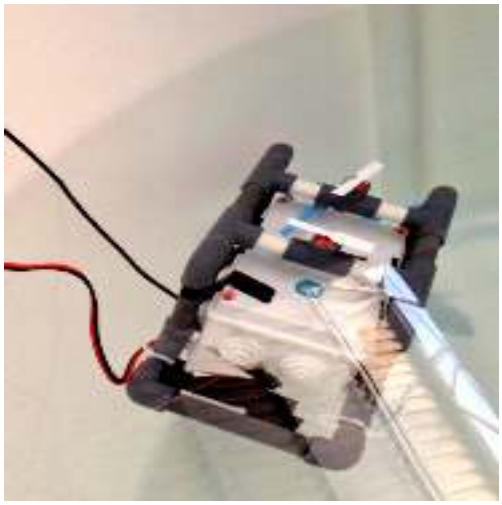


Figura 15. Prueba de flotabilidad del robot submarino.

3.2 MOTORES

Se han llevado a cabo diferentes pruebas con los motores de propulsión. En primer lugar, una vez tratados los motores con resina epoxi, se introdujeron varios motores en un recipiente con agua conectados a una fuente de alimentación y se comprobó que éstos funcionaban correctamente. Tras realizar las pruebas los motores fueron tratados con productos de limpieza y lubricación. En todas las pruebas se observó un comportamiento adecuado de los motores.

3.3 TELEOPERACIÓN

Una vez realizadas las pruebas de los diferentes componentes individuales se llevo a cabo la prueba del sistema completo, para ello se hizo uso de un depósito de agua (1.5 m de diámetro x 1.2 m de altura) en el cual se coloco el robot conectado a la fuente de alimentación y el PC como se puede observar en la Figura 16.

Se comprobaron todas las acciones implementadas: movimientos de avance y retroceso (Figura 17), ascenso, descenso, giros y movimiento de cada motor individual. Todas las pruebas realizadas resultaron positivas ya que el robot realizaba todos los movimientos que se pretendían.

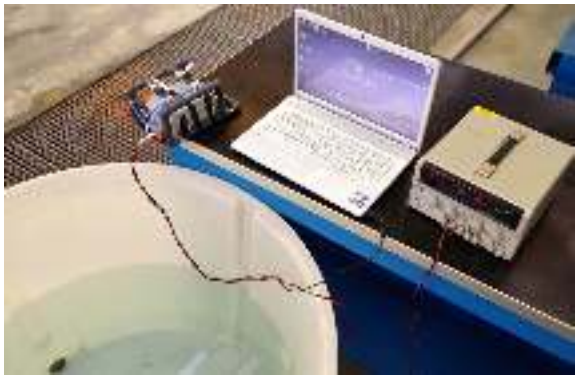


Figura 16. Pruebas del sistema completo desarrollado.

5 CONCLUSIONES Y TRABAJOS FUTUROS

Este artículo ha presentado el desarrollo de un robot submarino de bajo coste y dimensiones reducidas para su aplicación en labores docentes. Se ha buscado un sistema económico para poder equipar un laboratorio de robótica con varias unidades, además se ha pretendido desarrollar un sistema de reducidas dimensiones para no requiera de una gran infraestructura para su uso. En particular, con una depósito de dimensiones reducidas (1.5 m x 1.2 m) ha sido suficiente para comprobar todos los grados de libertad del robot submarino.

Este robot submarino se ha utilizado en un curso de robótica para estudiantes de posgrado de la Universidad de Almería para explicar los conceptos básicos asociados a los robots submarinos. Los alumnos han mostrado una gran acogida de la idea y se pretende en el futuro desarrollar varios robots

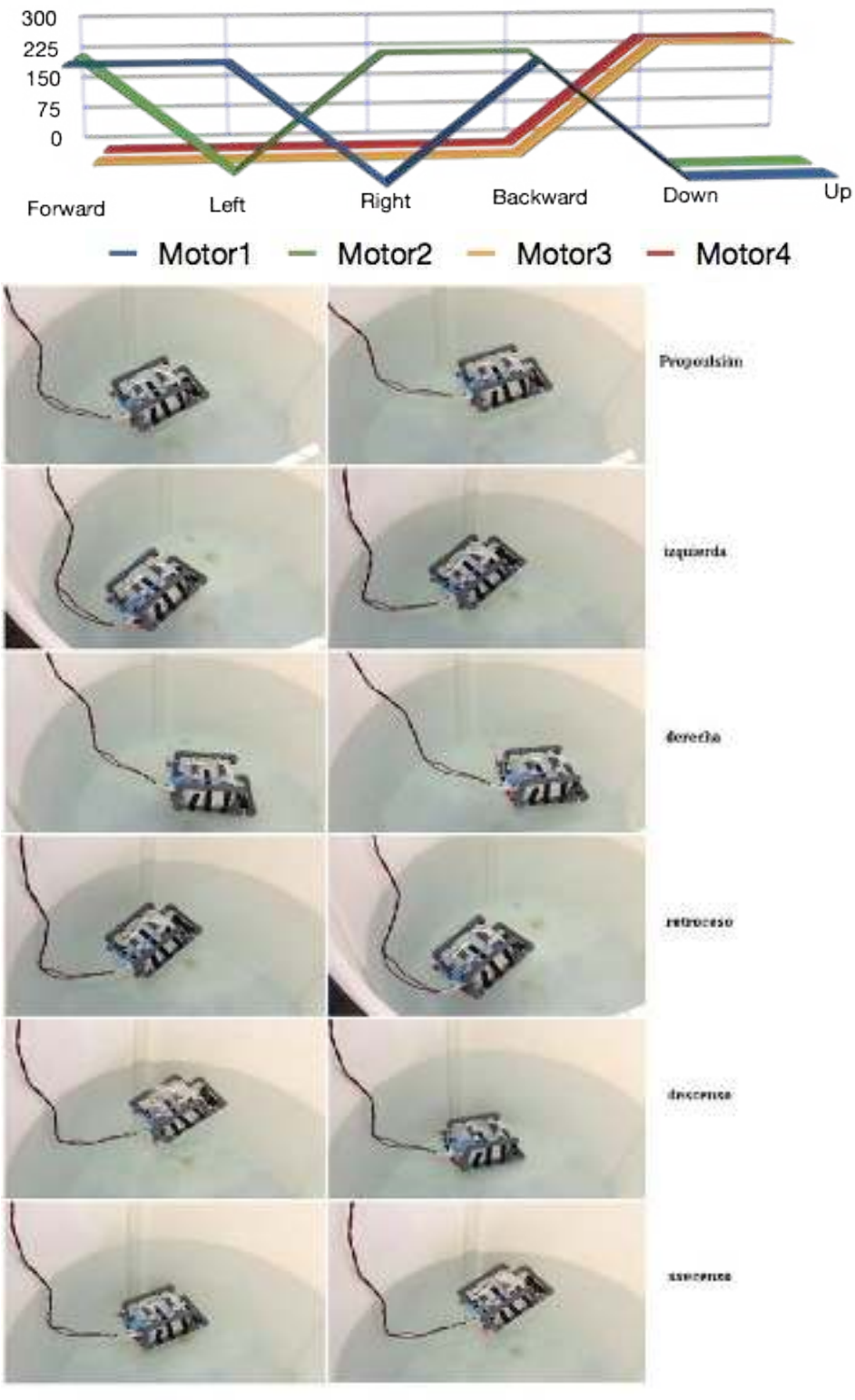


Figura 17. Pruebas de movimientos de retroceso y avance del robot submarino desarrollado

más para que ellos mismos puedan interactuar y programar el robot.

Especial mención es para el sistema de control seleccionado para el robot submarino basado en tecnología Arduino, la cual constituye una solución de bajo coste pero de grande posibilidades. Por lo que en el futuro los alumnos además de programar el robot submarino podrán utilizar nuevos sensores y realizar tareas más complejas como control de posición, seguimiento de trayectorias, y adquisición y transmisión de información del medio submarino, entre otras.

6 BIBLIOGRAFÍA

[1] C. Buiu. Hybrid Educational Strategy for a Laboratory Course on Cognitive Robotics. *IEEE Transactions on Education*, 51(1):100–107, 2008.

[2] D.P. Miller, I.R. Nourbakhsh, and R. Siegwart. Robots for Education. En *Handbook of Robotics*, Pp. 1283–1301. Springer, 2008.

[3] G. Antonelli, T. I. Fossen, D.R. Yoerger, (2008). "Underwater robotics", Springer Handbook of Robotics, pp. 987-1008.

[4] Odyssey. (2009). Odyseey`s Remotely Operated Vehicle(ROV). http://shipwreck.net/g2/gallery2/v/Vesselsandtechnology/rov_zeus/ (Accedido Junio 2013).

[5] Comité Español de Automática, Grupo Automar http://www.ceautomatica.es/og/automar (Accedido Junio 2013).

[6] Plataforma Tecnológica Marítima Española www.ptmaritima.com (Accedido Junio 2013).

[7] R. Isermann. 2005. "*Mechatronic Systems. Fundamentals"*. Springer. 624pp.

[8] H. Bohm y V. Jensen. 2007. "*Build your own underwater robot and other web projects"*. Westcoast words, Canada. 150pp.

[9] Arduino. (2005). Arduino. http://www.arduino.cc (Accedido Junio 2013).

[10] Maxon Motors, http://www.maxonmotor.es/ (Accedido Junio 2013).

[11] PC104 Consortium, www.pc104.org (Accedido Junio 2013).

[12] Comunidad Raspberry Pi, www.raspberrypi.org (Accedido Junio 2013).

SENSORES CIENTÍFICOS PARA UN PLANEADOR AUTÓNOMO SUBMARINO

Narváez, F.** , González, H.** , García, E.*

*Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia, Camino de Vera s/n, 46071 Valencia, (España)
E-mail : egarciam@isa.upv.es

**Escuela Técnica Superior de Ingeniería del Diseño, Universidad Politécnica de Valencia, Camino de Vera s/n, 46071 Valencia, (España)
E-mail : {jesbarch, hecparib}@etsid.upv.es

Resumen

En este trabajo se realizará el análisis de un bloque especial de sensores a utilizar en un vehículo autónomo submarino que permitirá una integración sensorial para la realización de observaciones científicas del entorno marino. El análisis está dedicado al estudio de aplicaciones científicas dejando de lado otras aplicaciones. Esto debido a que los vehículos autónomos submarinos tiene un amplio uso y por ende, un sin número de aplicaciones. La potencia de consumo, dimensiones, peso entre otros factores importantes serán considerados factores claves para la selección adecuada de los sensores y se dará una idea básica del uso de cada sensor.

Palabras Clave: Vehículo autónomo submarino, sensores científicos, carga útil.

1 INTRODUCCION

Los Planeadores Autónomos Submarinos (PAS) siguen siendo una tecnología en desarrollo y un punto importante a tratar son los sensores a utilizar, ya que estos varían en función de la misión a desarrollar. Existen investigaciones cuyo enfoque se centra en algoritmos de control para la navegación, en los cuales se mencionan los sensores a utilizar. Lo que se busca en este artículo es profundizar en ese tipo de subsistema denominado de sensores científicos.

El estudio y las mejoras que están siendo realizadas en vehículos autónomos submarinos se enfocan en gran parte en consideraciones que tienen que ver con el tamaño, peso y requerimientos de consumo de energético del sistema. Esta cuestión es fuertemente argumentada en el estudio realizado en [1].

Dentro de las configuraciones básicas en cuanto a sensores en vehículos submarinos, están el conjunto de sensores CTD (conductividad, temperatura y profundidad), que se aplican en aplicaciones para la navegación como para mediciones científicas del entorno marino. Si lo que se desea es tener un vehículo submarino para otras aplicaciones se pueden incluir sensores biológicos, sensores ópticos considerados como sensores no acústicos en el estudio realizado[1]. Es importante notar que todos los artículos mencionan la necesidad que se tiene de incorporar a los vehículos submarinos sensores de bajo consumo y ligeros, ya que esto puede afectar a la estabilidad del sistema de navegación al cambiar el centro de gravedad de manera importante para maniobrar en los desplazamientos del PAS.

La carga útil y los sensores pueden tratarse de dos formas:

1) Una configuración simple de la carga útil conformada por un solo tipo de sensor.

2) Una carga útil conformada por múltiples sensores, la cual puede ser dedicada a una misión específica siempre que se utilice un sistema modular que permita desmontarlos y colocar un nuevo sensor [1]. Actualmente existen casas comerciales que poseen instrumentos de medida con este fin, como son AML con su módulo Smart-X que incorpora 3 sensores y por medio de un solo canal de comunicación envía los datos.

El artículo constará de especificaciones técnicas de sensores comerciales que serán integrados de manera tal que se adapten a su aplicación, está claro que en este artículo no se mencionará el control de navegación, que sería realizado una vez se integren todas las señales de los sensores.

Tomando en consideración [2], se presentará una dirección clara de la selección de sensores en diversas aplicaciones y en base a los estudios de [3], se seleccionará ciertas variables a medir de acuerdo a diferentes planeadores comerciales, y [4] para argumentar la utilización de sensores químicos y biológicos, los cuales siguen en desarrollo en búsqueda de mejoras de sus requerimientos tanto de

peso como de potencia consumida y sobre todo para la aplicación futura de tareas de mantenimiento que se hacen necesarias al utilizar este tipo de sensores, incrementando así el costo de los vehículos.

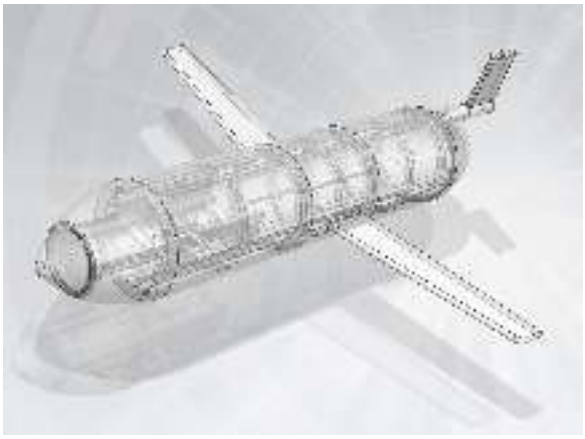


Figura 1.- Casco del prototipo SGU-005.

2 REQUERIMIENTOS DEL SISTEMA

Los sensores dentro de un vehículo autónomo submarino tienen dos funciones globales, por una parte aquellos utilizados para el sistema de navegación y todas las funciones que están relacionadas con el movimiento y posicionamiento del navegador. Los sensores usados para la navegación y posicionamiento serán tratados considerando los requerimientos de exactitud en el posicionamiento deseables en aplicaciones navales. En la Figura 2 se puede ver un diagrama de bloques que ayudará a entender el desarrollo del presente artículo.

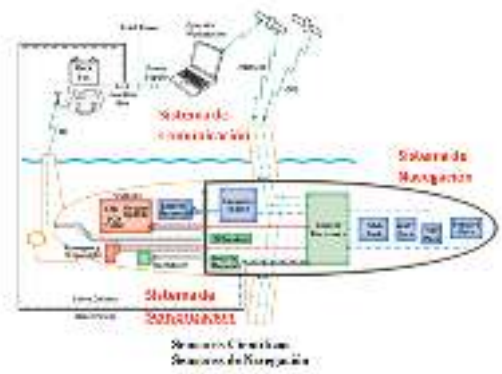


Figura 2 Esquemático de los tres sistemas principales de un planeador submarino.

Los sensores recomendados como base por todos aquellos que han implementado un planeador submarino son los citados a continuación.
En cuanto a medidas científicas:

- Conductividad
- Temperatura

En cuanto a medidas necesarias para el sistema de navegación:

- Profundidad
- Altitud
- Compass
- GPS

En la Figura 3 se muestra el modelo de estados de un PAS.

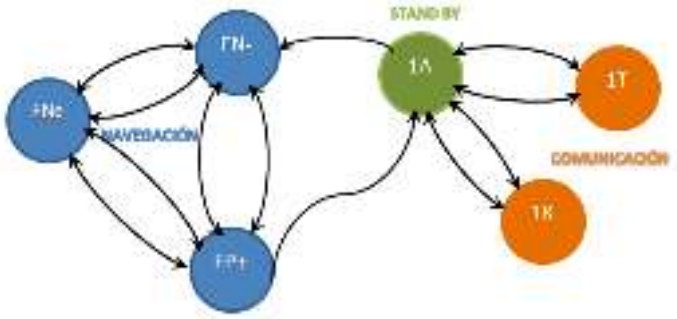


Figura 3 Diagrama de estados del sistema de control

El estado inicial 1A del planeador es aquel en que encontrándose en superficie, está la espera, recibiendo instrucciones o en la tarea de envío de datos. Por lo tanto, se establecerá una comunicación de transmisión o recepción con la central en este estado. El segundo estado FN- es el que se encontrará en navegación iniciando con flotabilidad negativa la inmersión, una vez que llega a su profundidad máxima, cambia a una flotabilidad positiva al estado FP+ para el ascenso, dando inicio a la trayectoria en forma de diente de sierra desarrollando un movimiento derivado de avance. Eventualmente, desde cualquiera de los 2 estados anteriores, puede permanecer en un estado de flotabilidad neutra (FN), suspendido entre dos aguas, realizando mediciones paramétricas controladas susceptibles de realizar a esas profundidades, durante periodos determinados. Cabe aclarar, que inicialmente el rumbo es determinado por medio del GPS y una vez sumergido la trayectoria estará controlada por el compás, altímetro y ecosonda, es decir navegando aplicando técnicas de estimación, que dependiendo de las perturbaciones existentes en inmersión, corrientes, colisiones, etc., pueden resultar con errores considerables.

3 SENSORES PARA APLICACIONES CIENTÍFICAS

Los planeadores submarinos tiene aplicaciones muy diversas, de estudios oceanográficos, de seguridad, comunicaciones, militares y permite la integración de todo tipo de sensores. Ejemplos de sensores de alta

prioridad para observaciones en el mar incluyen: conductividad, temperatura, profundidad, oxigeno, nitrato, amonio, urea, silica, fosfato, Ph, alcalinidad, metano, alógenos disueltos, materiales metálicos, magnesio, hidrógeno y actividad microbiana. Los espectrómetros son sensores altamente deseados debido a que ellos pueden detectar una amplia variedad de especies. Sensores acústicos y ópticos son necesarios para medir una variedad de fenómenos físicos, químicos y biológicos, así como el fitoplankton y zooplankton [4].

3.1 SENSORES CTD COMERCIALES

Salinidad, temperatura, y presión son las tres variables de estado básico que permite el cálculo de la densidad y las asociadas propiedades físicas del agua del mar y su influencia en las condiciones meteorológicas de manera especial. A breves rasgos podemos decir que en aplicaciones militares su uso se basa en conocer SVP (Sound Velocity Profiles), esto con el fin de conocer la velocidad del sonido en función de la profundidad en el océano, la cual es una herramienta útil para tener una amplia fidelidad en la entrada de datos del Sonar, útil en comunicaciones, detección de objetivos, navegación entre otras. Así como en aplicaciones científicas o estudios del comportamiento marino, el CTD es utilizado, por citar algunos ejemplos, en la navegación del vehículo, como para medir parámetros marinos que permiten determinar el cambio que sufre la temperatura a diferentes profundidades así como la salinidad que permita asistir a las predicciones meteorológicas ambientales.

Fabricante	Peso aire/agua	Tamaño	Potencia	Voltaje requerido [VDC]
Sea-Bird Electronics GCTD	2/0.98	-------	2,9 watts hora	8-20 VDC
Sea-Bird Electronics GCTD SBE 49	2,7/1,4	Ø83mm 620mm	3,42 watts hora	9-24 VDC
Valeport miniCTD	1/---	Ø54mm 370mm	2,9 watts hora	9-28 VDC

Tabla 1.- Parámetros de eléctricos y físicos de CTD analizados.

Como se puede observar en la tabla 1, el sensor más atractivo por su peso y consumo es el de Valeport miniCTD; sin embargo, es necesario comparar otras especificaciones en cuanto a rango y salida de datos, esto para equilibrar las opciones, y asegurarnos de una mejor elección. Claro está, que todos ellos son utilizados en aplicaciones con vehículos submarinos, por citar un ejemplo en el planeador éFolaga [5] es utilizado el conjunto de sensores CTD.

Los datos de la profundidad máxima que soporta cada sensor dependerán de los requerimientos y del diseño del vehículo submarino, ya que se tiene la facilidad de adquirirlo en dos tipos de materiales de la coraza de protección, que variará en función de las aplicaciones. En la Tabla 2 se mostrarán los rangos de medida de cada una de las variables y la profundidad máxima que pueden soportar.

Fabricante	C [S/m]	T [° C]	P [dbar]	Profundidad [m]	Salida de datos
Sea-Bird Electronics GCTD	0 to 9	-5 to +42	0 a 2000 configurable	2000	RS-232C
Sea-Bird Electronics GCTD SBE 49	0 to 9	-5 to +35	0 a 7000 configurable	7000	RS-232C
Valeport miniCTD	0 to 8	-5 to +35	0 a 6000 Configurable	6000	RS-232C RS-485 Bluetooth adapter

Tabla 2.- Datos técnicos de CTDs comerciales.

Para el desarrollo de este artículo se seleccionará al miniCTD de Valeport, que tienen mejores características técnicas; sin embargo, es importante aclarar que habrá que considerar el costo ya que debido a sus características sus precios son bastante elevados, 5.150€.



(a) MiniCTD Valeport

(b) Sea-Bird Electronics GCTD

(c) Sea-Bird Electronics GCTD SBE 49

Figura 4. CTDs de diferentes casas comerciales.

3.2 SENSOR CORRENTIMÉTRICO

La medida de las corrientes marinas es un factor a considerar para el conocimiento de los modelos dinámicos del desplazamiento de masas de aguas. Estas corrientes son ocasionadas por la rotación de la

tierra y por vientos constantes o planetarios, así como la posición de la superficie terrestre o diferencia de temperaturas y densidades del agua marina.
La corriente que nos encontremos será diferente según la temperatura y la profundidad en la que estemos. Normalmente, las corrientes superficiales, no son coincidentes con las corrientes submarinas. De las corrientes marinas nos interesan 2 características principales, la velocidad y el rumbo de propagación.
Para medir las corrientes marinas existen dos opciones: los correntímetros electromagnéticos o los correntímetros basados en el efecto doppler [6].
Los sensores a utilizar son los correntímetros electromagnéticos que utilizan el principio de un conductor eléctrico que se mueve a través de un campo magnético e induce una corriente eléctrica. El agua de mar es un conductor muy bueno, y cuando se mueve entre dos electrodos, la corriente eléctrica inducida es proporcional a la rapidez de la corriente oceánica entre los electrodos. Un correntímetro electromagnético tiene una bobina para producir un campo magnético y dos sistemas de electrodos, colocados en ángulo recto el uno del otro. Combinando la rapidez medida por los dos sistemas, el instrumento determina la rapidez y dirección de la corriente oceánica [7].
El sensor a utilizar es el medidor de corrientes Model ROV 803 de Valeport , que proporciona datos sobre la velocidad relativa del agua en tiempo real. Está especialmente diseñado para ser instalado en vehículos teledirigidos para proporcionar velocidades del agua, o equipados con sistemas de gestión de sujeción para dar una medida de las condiciones de flujo local. La selección de las opciones de salida ofrecen una interfaz sencilla RS-232 y RS-485, y los datos se pueden mostrar mediante el software suministrado. Disponible en versiones para 500m acetal y 3000m titanio, el Modelo 803 es un dispositivo fiable para medir las velocidades de corrientes en una amplia variedad de aplicaciones en vehículos subacuáticos. El costo de este sensor es de 5200€.

3.3 SENSOR DE OXÍGENO

La concentración de oxígeno en sistemas marinos depende esencialmente de tres principales procesos: intercambio de gases con la atmósfera, mezclas de aguas que provienen de la superficie, y de la producción de oxígeno en la fotosíntesis. Por lo que se requiere de un sensor de alta calidad para el estudio in situ de estos procesos. Idealmente se requiere que estos sensores tengan una respuesta rápida, una alta sensibilidad a la variación de concentración de oxígeno y una estabilidad adecuada en amplios períodos de tiempo [4]. A partir de estos requerimientos en [8] se realiza una comparación de diferentes sensores de oxígeno comerciales, llegando a la conclusión de que el modelo Aanderaa 4330F tiene un mejor comportamiento en misiones de vehículos sumergibles en largos periodos. El Aanderaa 3830 tiene el mejor tiempo de respuesta, para este artículo se tratará del Aanderaa 4330F.
Los medidores de oxígeno AADI Aanderaa están basados en su habilidad de seleccionar substancias que actúan como atenuadores dinámicos de la fluorescencia, por lo tanto su principio de medida es óptica, puede estar más de un año sin calibración, con bajos niveles de mantenimiento, y su uso es amigable. Este sensor ofrece una salida en formato AiCaP CANbus y RS-232.

3.4 FLUORÓMETRO Y TURBIDEZ

Son usados en la toma de medidas de variables bioquímicas en el océano. El ECO Puck, puede ser configurado como fluorómetro y como sensor de turbiedad. Este sensor puede ser integrado dentro del cuerpo del navegador, su diseño es ampliamente recomendado para uso en vehículos autónomos submarinos, y tiene una salida de comunicación RS-232.
Las características de todos los sensores para una aplicación científica mencionados se encuentran en la Tabla 6 y la Figura 5 muestra los sensores comerciales descritos.

Modelo	Rango	Potencia	Peso aire/agua
Valeport ROV 803	± 10kts	7 – 19 VDC, 3 W	---/Acetal 0.5kg ---/Titanio 3.5kg
Aandaraa 4330F	0 - 500µM	8 – 15 VDC, 100 mA máximo	0.175 Kg / ----
ECO Puck	Dependiente del parámetro a medir	7 – 15VDC, 80mA	0.28 Kg /0.02 Kg

Tabla 6.- Especificaciones de los sensores comerciales para aplicaciones científicas



a) b)



c)

Figura 5.- a) Sensor Correntimétrico ROV 803 ; b) Sensor de Oxígeno Aanderaa 4330F ; c) Conjunto de fluorómetro y sensor de turbiedad ECO Puck.
En resumen el peso total y la potencia consumida se encuentra resumida en la tabla 7 seleccionando el sensor multiparámetros CTD.

Sensor	Consumo [w]	Peso[kg]
Valeport miniCTD	2.9	1
Valeport Acelerometer	3.5	1
Wilcoxon Research model 757	0.3	0.11
Valeport ROV 803	3	1.81
Aandaraa 4330F	1.5	0.175
ECO Puck	1.2	0.28
Total	12.4	4.375

Tabla 7.- Resumen de peso y potencia consumido de sensores para las dos aplicaciones.

Como se mencionó anteriormente se ha hecho una diferencia entre los sensores para navegación y sensores para aplicaciones correspondientes a cada una de las misiones científicas del vehículo submarino, por lo tanto el altímetro como el compás no han sido incluidos en el presente análisis.

4 SENSORES DE NAVEGACIÓN

Debido a que se pretende dejar abierta la posibilidad de emplear al planeador autónomo submarino en aplicaciones navales militares, es imprescindible una alta precisión del sistema de navegación y se requiere un control más estricto que los que se desarrollan en las configuraciones básicas.
El objetivo de los planeadores en aplicaciones navales militares está basada en detección de objetivos, vigilancia y localización; sin mencionar que también se puede neutralizar minas y torpedos [2]. Para ello se emplearán Altímetros acústicos (Valeport Acelerometer), Sensor Acústico Omni-Direccional (Patentado sin fabricación comercial), Sensor Acústico Direccional (Wilcoxon Research model 757) cuyas características se encuentran tabuladas en la Tabla 5.

Modelo	Rango	Potencia	Peso	Profundidad [m]	Salida de datos
Valeport Acelerometer	0.1m – 100m	9-28VDC 125 mA	< 1Kg	6000	RS-232C RS-485
Wilcoxon Research model 757	50 g máximo	18 - 30 VDC 2 - 10 mA	0.11 kg	6500	0-12VDC

Tabla 8. Características sensors ANT para aplicaciones militares

El sensor acústico Omni-Direccional se encuentra patentado, sin embargo podría emplearse otro tipo de sensor que cumpla el mismo objetivo, como pueden ser los sensores ópticos, una cámara CCD para estas aplicaciones, como las tratadas en el artículo de Scripps Intitution of Oceanography. En la Figura 4 se muestras los sensores a utilizar. El altímetro tiene un costo de 2600€.



Figura 6 Valeport Acelerometer, Wilcoxon Research model 757

5 CONCLUSIONES

El tipo de sensores utilizados en vehículos submarinos autónomos es bastante amplio; sin embargo, las limitaciones que se tienen en cuanto a potencia, peso y dimensiones son claves en el momento de seleccionar los sensores, como se ha visto a lo largo del artículo la meta es encontrar sensores de bajo consumo y ligeros. Una opción es elegir sensores modulares y cambiarlos en cada misión. Para que la integración de sensores sea

adecuada existe documentación que permite realizar un estudio para planificar las misiones[9].

RECONOCIMIENTOS

El presente proyecto de investigación ha sido financiado como Proyecto de Investigación ENE2010-21711-C02-01 por el MICINN español.
Las empresas ValenciaPort (Autoridad Portuaria de Valencia), Ingeteam Service S.A., Real Club Naútico de Valencia están colaborando activamente en el desarrollo del mismo.

REFERENCIAS

[1] S.A. Jenkins, << Underwater Glider System Study UC San diego, 203.
[2] W. P.Barker, «An analysis of undersea glider architectures and an assesment of undersea glider integration into undersea applications» Naval Postgraduate School, Monterey, California, 2012.
[3] D. W. French, «Análisis de Unmanned Undersea Vehicle(UVV) Architectures and an Assessment of UUV,» Naval Postgraduated School, Monterey, California, 2010.
[4] L. Daly, «Chemical and Biological Sensors for Time-Series Research:Current Status and New Directions,» College of Marine Sciencie University of South Florida, South Florida, 2004.
[5] A. Alvarez, «Folaga: A low cost autonomous underwater vehicle combining glider and AUV capabilities,» Elsevier Ltd, Genova, Italy, 2008.
[6] M. C. Sanchez, «Sistema de Monitorización y Supervisión de una Boya para Generación de Energía Undimotriz,» Escuela Politécnica de Cataluña, Barcelona, 2011.
[7] F. M. d. Chile, «Portal de Los Siete Mares,» 22 Septiembre 2005. [En línea]. Available: http://www.mardechile.cl/index.php?option=com_content&task=view&id=380&Itemid=31. [Último acceso: 29 Mayo 2013].
[8] K. Parker, «Oxygen Sensor Evaluation for the Tethys Long Range AUV,» Moss Landing Marine Labs, Monterey Bay, 2009.
[9] E. R. Cabrera, «Sistema Integrado de Planificación y Control de Misiones con Vehículos y Control de Misiones con Vehículos Submarinos Autónomos,» Facultad de Informática Universidad de Las Palmas de Gran Canaria , Las Palmas de Gran Canaria, 2010.
[10] D. Rudnick, «Underwater Gliders for Ocean Research,» Scripps Institution of Oceanography, California, 2004.

SISTEMA DE COMUNICACIONES PARA UN PLANEADOR AUTÓNOMO SUBMARINO

González, H.** , Narváez, F.** , García, E.*

*Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia, Camino de Vera s/n, 46071 Valencia, (España)
E-mail : egarciam@isa.upv.es

**Escuela Técnica Superior de Ingeniería del Diseño, Universidad Politécnica de Valencia, Camino de Vera s/n, 46071 Valencia, (España)
E-mail : hecgonmo@etsid.upv.es, fernarch@upv.es

Resumen

El objeto de este trabajo, es el estudio de un subsistema para la obtención de una cobertura global de comunicaciones por satélite para un planeador autónomo submarino.
De manera especial, se describen los módulos necesarios para llevar a cabo esta comunicación tales como: un microcontrolador, un datalogger, una antena y un módulo GPS y de comunicación por satélite dónde se detallará la comunicación por satélite del sistema IRIDIUM.

Palabras Clave: Vehículos Autónomos Submarinos, Comunicación, Satelite.

1 INTRODUCCION

Un Glider (inglés) o Planeador Autónomo Submarino (PAS) es un tipo de vehículo submarino que consigue un movimiento de desplazamiento debido al deslizamiento submarino producido por los cambios programados de su flotabilidad, y eventualmente en desplazamientos horizontales debido a la utilización de la energía cinética obtenida de los flujos existentes en el entorno marino, es decir, por medio de las corrientes marinas. Este tipo de vehículos son reconocidos como de bajo coste porque no disponen de un sistema de propulsión convencional interno, lo que posibilita mayor duración de las baterías, menor tamaño y peso, etc. Lo anterior ha hecho posible su utilización como dispositivos móviles con periodos largos de permanencia en el lecho marino, en funciones de observación y medición de parámetros de interés en dicho entorno.
El control de la navegación para las operaciones de cambio de rumbo, inmersión etc., las realiza mediante cambios controlados de posición de su centro de gravedad provocado por desplazamientos de masas internas y de la variación de su condición de flotabilidad que puede ser positiva, negativa o neutra.

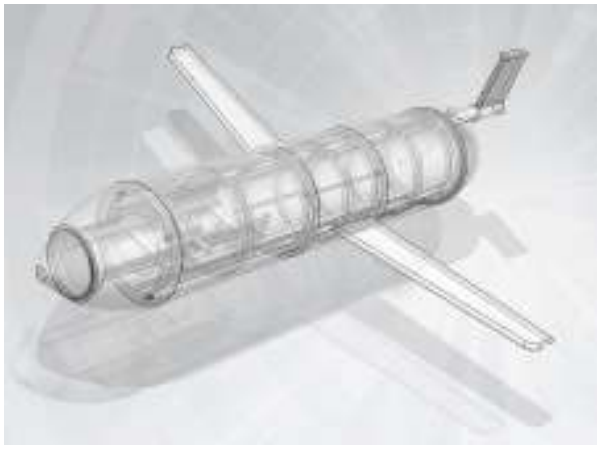


Figura 1. Planeador Autónomo Submarino (PAS)

El sistema IRIDIUM es una red de comunicaciones móviles personales basada en satélites de baja órbita. Está diseñado para permitir que cualquier tipo de transmisión (voz, datos, fax o mensajería) llegue a cualquier parte de la Tierra.
IRIDIUM es el nombre de una constelación de 66 satélites de comunicaciones que giran alrededor de la Tierra en 6 orbitas bajas LEO a una altura aproximada de 780 Km de la tierra. Cada una de las 6 órbitas consta de 11 satélites equidistantes entre sí. Los satélites tardan 100 minutos en dar la vuelta al mundo de polo a polo.
La cobertura global del sistema junto con su órbita baja, lo hacen muy atractivo para comunicaciones móviles en zonas muy despobladas donde la aparición de operadores de telefonía móvil terrestre sea muy difícil, así como en situaciones de emergencia como terremotos, inundaciones, ya que no necesita infraestructura terrena cercana.
Cabe destacar que mientras un terminal INMARSAT cuesta $45.000, un terminal IRIDUM cuesta $3.000, además de tener un tamaño mucho menor y menores requerimientos de potencia.

Otro campo de aplicación donde el sistema IRIDIUM resulta muy adecuado es en las comunicaciones marítimas a realizar a través de sistemas planeadores autónomos submarinos, dónde conocer la posición del PAS y el envío de nuevos parámetros para posibles cambios de ruta lo hacen más atractivo y más versátil en su funcionalidad. La siguiente figura nº2 muestra un diagrama de bloques de un sistema autónomo submarino que integra un módulo de comunicación por satélite.

2 REQUERIMIENTOS DEL SISTEMA

Los PAS fueron utilizados inicialmente como dispositivos más avanzados y con mayores prestaciones dinámicas que los antiguos perfiladores flotantes utilizados por los meteorólogos para la obtención de mediciones de la temperatura y salinidad de los océanos. Posteriormente estas mediciones eran utilizadas en modelos de predicción meteorológica.

En estas funciones, los PAS permanecían durante meses obteniendo datos del entorno marino recorriendo amplias zonas oceánicas, para retransmitir los datos obtenidos a una estación base terrestre. Actualmente las aplicaciones de los PAS, se han diversificado considerablemente en aplicaciones científicas, de comunicaciones, militares, etc., en el entorno marino.

Para ello, se requiere que el sistema garantice una comunicación con un sistema remoto a nivel mundial, incluyendo la posibilidad de tener una comunicación en tiempo real. De la misma manera surge la necesidad de tener un registrador y acumulador de datos donde se almacenen las mediciones de magnitudes diversas ordenadas en el tiempo. En el diagrama de bloques se aprecia los módulos integrantes del sistema.

3 DIAGRAMA DE BLOQUES

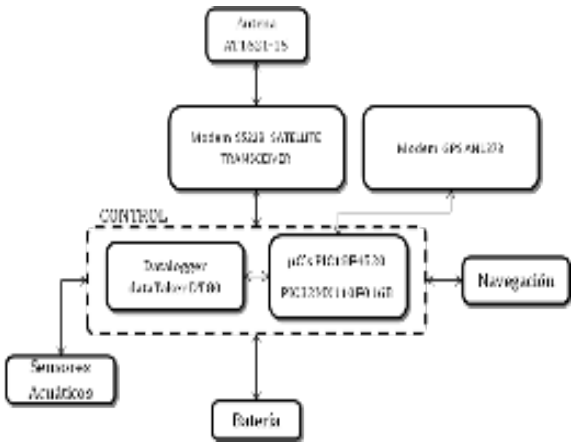


Figura 2.Diagrama de bloques

4 DESCRIPCIÓN DETALLADA

4.1 Microcontrolador 'PIC18F4520 y PIC32MX110F016B'

Es un microprocesador optimizado para controlar equipos electrónicos o interactuar con éstos. Posee un número importante de entradas y salidas que permite interactuar con diferentes elementos activos o pasivos (transistores, amplificadores operacionales, Leds, circuitos y módulos integrados, etc.).

Las ventajas que ofrecen estos sistemas son:

- Tamaño reducido.
- Su funcionamiento proporciona un elevado número de posibilidades, habiendo configurado previamente una serie de instrucciones en su interior.
- Su precio es realmente económico.
- El mercado actual ofrece un número mayor de estos componentes, los cuales son usados en, prácticamente, casi cualquier sistema electrónico actual.

Para esta aplicación nos interesa que sea de dimensiones reducidas dado el poco espacio disponible en un PAS. Cabe mencionar que servirá de apoyo para la gestión de algunos procesos con nuestro modulo GPS y a nuestro datalogger por lo gran funcionalidad que ofrece. Se selecciona el PIC18F4520 y PIC32MX110F016B por la ergonomía de trabajo que ofrece la compañía Microchip.

4.2 Módulo GPS AN1373

Tiene una aplicación cada vez más popular entre la tecnología M2M (machine to machine) para la navegación GPS. Con esta tecnología, se dispone de un sistema completo GPS de navegación para distinto tipo de vehículos. Esta tecnología puede ser utilizada para rastrear la ubicación actual de un vehículo, o proporcionar un mapa de direcciones. Además, el consumidor puede hacer una llamada de emergencia desde el mismo dispositivo que está haciendo el seguimiento. Las empresas de estacionamiento pueden utilizar esta tecnología para saber cuánto tiempo ha estado aparcado un vehículo y cobrar en consecuencia.

El modulo GPS AN1373 se conecta a la interfaz de la tarjeta de expansión Multimedia (MEB), pero también permite la implementación de conectividad GSM/GPRS/GPS mediante los respectivos comandos utilizando un microcontrolador (MCU) PIC32.

La serie de LS20030-3 son receptores GPS completos de antena inteligente low-cost, incluyendo una antena integrada, y circuitos del receptor GPS, diseñada para un amplio espectro de aplicaciones en el sistema OEM. La antena inteligente GPS adquirirá

hasta 66 satélites a la vez. Se le puede proporcionar con una sensibilidad superior y dispone de un buen rendimiento incluso en casco urbano y el medio ambiente con denso follaje. Su capacidad de largo alcance cumple con los requisitos de sensibilidad para la navegación.

4.3 Datalogger 'dataTaker DT80'

Es equipo muy versátil, con un enorme repertorio de posibilidades en cuanto a comunicaciones, tipos de sensores a controlar, posibilidades de cálculos a realizar con las mediciones y con una enorme capacidad de almacenamiento.
Los tipos de sensores analógicos que es capaz de controlar son tensión continua, corriente, bucles de corriente 4-20 mA, puentes y de frecuencia. Además permite realizar cálculos con los valores medidos antes de almacenarlos, transformando así el valor al formato y unidades deseadas.
Tiene 5 canales analógicos, que dependiendo de cómo se conecten los sensores, permite la conexión desde 5 hasta 15 sensores analógicos. Si en cada canal se utilizan sensores analógicos de 3 o 4 hilos independientes, entonces se pueden conectar 5 sensores como máximo, uno por canal. No obstante puede expandirse con unidades adicionales CEM20 la cantidad de sensores analógicos a controlar hasta un total de 100.

4.4 Antena 'AT1621-15'

La antena helicoidal estilo látigo AT1621-15 Iridium está diseñada para aplicaciones basadas en vehículos. Está diseñado para trabajar con todos los dispositivos de comunicaciones de datos y voz de Iridium y es excelente para aplicaciones de vehículos terrestres remotas donde se requiere una antena robusta. Se pueden montar al descubierto y puede resistir el contacto con obstrucciones.

4.5 Módulo de comunicación IRIDIUM 9522B SATELLITE TRANSCEIVER

El 9522B es la segunda generación del receptor satelital global de Iridium para comunicaciones de voz y datos. El 9522B se utiliza para aplicaciones registradas que necesitan un transceptor vía satélite para una aplicación específica, tal como un terminal de teléfono de voz marítimo o una solución de seguimiento de vehículos.
El transmisor-receptor satelital Iridium 9522B es ideal para el envío y recepción de voz y datos de los equipos en todo el planeta. Es funcionalmente compatible con todos los servicios de voz y de datos de Iridium y se integra fácilmente en una amplia variedad de aplicaciones a través de una interfaz serie RS232 y conjunto de comandos AT.

Es un módulo que presenta un peso de 420 g. y un tamaño reducido (162x81x28 mm) por lo que también nos resulta interesante dada la poca disponibilidad de espacio que tiene el sistema.

5 ALGORITMO DE CONTROL

El siguiente diagrama de estados muestra los 3 macroestados de funcionamiento, espera, comunicación y navegación del glider y cómo están interrelacionados.

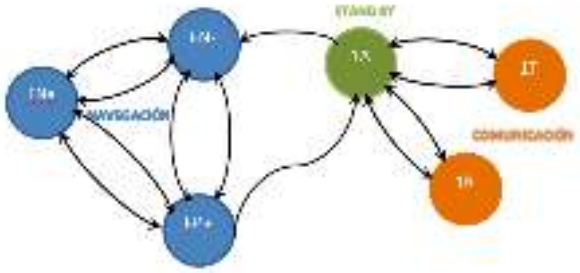


Figura 3.Diagrama de estados.

En su funcionamiento, el PAS es depositado inicialmente en el agua, considerando a esta situación, el estado de inicialización. Seguidamente se le transmiten la configuración de los parámetros necesarios para iniciar una ruta hacia un waypoint, para realizar eventualmente durante el trayecto, una serie de mediciones programadas. En el comportamiento dinámico del PAS, se consideran 3 macroestados diferentes:

1) SITUACIÓN ESTÁTICA EN SUPERCIFICE,
2) MOVIMIENTO DINÁMICO EN DIENTE DE SIERRA EN INMERSIÓN, EVENTUALMENTE DESARROLLA MEDICIONES.
3) SITUACIÓN ESTÁTICA ENTRE 2 AGUAS, EVENTUALMENTE REALIZA MEDICIONES .

La idea principal que se desarrolla en este tipo de vehículos es conseguir que tengan una flotabilidad positiva, pero muy pequeña, para de este modo, con un pequeño tanque de lastre poder sumergir el vehículo variando la flotabilidad del mismo, es decir de positiva a negativa con estados eventuales de flotabilidad neutra. Esto se hace en ciclos programados por lo que el vehículo esta funcionando en fases continuas de inmersión y emersión describiendo una trayectoria en forma de diente de sierra.
Después de realizar el tipo de ciclo anterior el planeador emergerá a superficie, de manera programada, o también determinado por la capacidad de memoria disponible en el sistema. Una vez en superficie se quedará flotando efectuando un despliegue de su antena para comenzar con el procedimiento de transmisión de datos y recepción de nuevas órdenes.

En esta fase el PAS también comunica las coordenadas de la posición en la que encuentra a través del GPS.
También se comunica con el satélite Iridium para enviar todos los parámetros registrados hasta el momento vaciando su buffer, consiguiendo así la máxima capacidad de almacenamiento de datos para la realización de algún estudio posterior. Aquí también le podemos enviar los datos que se deseen como los parámetros correspondientes para una nueva ruta de navegación cambiando su destino.
De ésta misma manera, se aprovechará la situación en superficie para hacer correcciones en la trayectoria derivadas de la aplicación de las técnicas de navegación por estima realizadas en las fases de inmersión, en la que no existe posibilidad de disponer de datos de posicionamiento GPS.
Finalmente una vez acabado el proceso de comunicación, el planeador se colocará en su posición correcta de navegación, comenzando nuevamente la inmersión y posicionándose corrigiendo sus desvíos respecto a los datos obtenidos por el GPS.

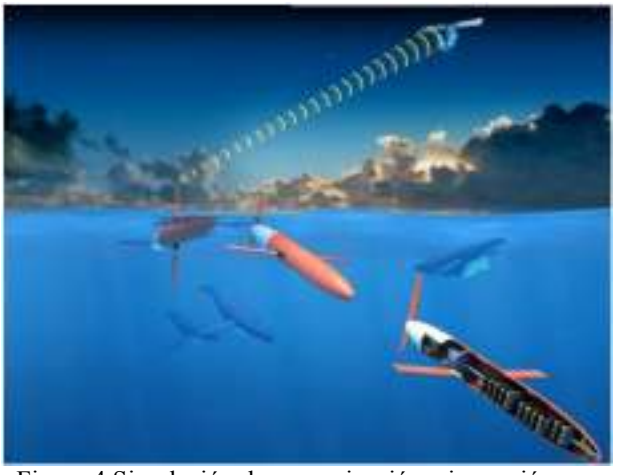


Figura 4.Simulación de comunicación e inmersión.

6 SERVICIO IRIDIUM

Iridium será capaz de soportar millones de usuarios (10 veces la capacidad actual de los sistemas geoestacionarios). El acceso al sistema Iridium se realizará mediante un prefijo universal. La facilidad ICRS (Iridium Cellular Roaming Service) enlazará las diferentes tecnologías celulares con los servicios satélite permitiendo al usuario tener un único terminal con número de teléfono, contrato y factura únicos para ambos sistemas terreno y satélite.
Iridium dispone de una tarifa correspondiente a las necesidades del usuario.
El coste estimado entre un sistema Iridium a Iridium estará sobre los 0.92$/min. Mientras que la comunicación entre Iridium y cualquier otro sistema que funcione por satélite el coste es de 11.79$/min.

7 CONCLUSIONES

En este trabajo, como aportación a un proyecto más amplio de diseño de un Planeador Autónomo Submarino, se ha realizado un trabajo de busqueda de los dispositivos necesarios para la implementación de un subsistema de comunicaciones, posicionamiento, y almacenamiento y tratamiento de datos para la localización de recursos marinos para la generación de energía renovable.
La busqueda de estos dispositivos se ha realizado teniendo en cuenta las restricciones en cuanto a consumo, tamaño, peso y costo de los mismos, considerando las dimensiones típicas de un PAS..

RECONOCIMIENTOS

El presente proyecto de investigación ha sido financiado como Proyecto de Investigación ENE2010-21711-C02-01 por el MICINN español.
Las empresas ValenciaPort (Autoridad Portuaria de Valencia), Ingeteam Service S.A., Real Club Naútico de Valencia están colaborando activamente en el desarrollo del mismo.

REFERENCIAS

[1] Physical Oceanography Physics 3300. (2005)
[2] Chunzhao, G., Mini Underwater Glider (MUG) for Education.
[3] Elvander, J., (EGO 2011) The Bluefin Spray Glider.
[4] Eriksen, C., (2003) Autonomous Underwater Gliders, Prepared of the Autonomous and Lagrangian Platforms and Sensors (ALPS) Workshop.
[5] Rudnick, D.L., (2004) Underwater Gliders for Ocean Research.
[6] Wood, S., (2010) Autonomous Underwater Gliders.
[7] Romiswati, Mohd-Mokhtar., Muhammad Hilmi R. A. Aziz., M. R. Arshad & Nur Afande Ali Hussain. Data compression for underwater glider system using frecuency sampling filters.
[8] Underwater System Study , Scripps Institution of Oceanography Technical Report.
[9] Sistemas de comunicación remota empleados en dispositivos y plantas offshore
[10] http://www.iridium.com/
[11] http://www.bluefinrobotics.com/
[12] http://www.erziasat.com/index.php
[13] http://www.irobot.com/en/us/robots.aspx
[14] http://www.valeport.co.uk/

Automatización de vehículo marino no tripulado en superficie para guiado autónomo

Carlos Cano Espinosa, Santiago T. Puente Méndez, Fernando Torres Medina
Grupo de Automática, Robótica y Visión Artificial.
Dpto. Física, Ingeniería de Sistemas y Teoría de la Señal,
Universidad de Alicante, España
cce1@alu.ua.es, {santiago.puente, fernando.torres}@ua.es

Resumen

Este artículo desarrolla una primera aproximación al diseño y construcción de un vehículo marino no tripulado en superficie. Para ello se emplea un barco radiocontrol como base, que ha sido modificado para adaptarse a las necesidades del diseño. El objetivo es conseguir que sea capaz de navegar de forma autónoma, localizarse en el entorno y trazar trayectorias simples. El artículo se centra en el proceso de construcción, las decisiones de diseño tomadas y la problemática asociada a los diferentes aspectos de implementación, encontrada en las sucesivas pruebas realizadas.

Palabras Clave: Barco, autónomo, prototipo

1 INTRODUCCIÓN

Es obvio que la robótica autónoma tiende a imponerse como solución ideal a la hora de realizar tareas complejas o aquellas que requieren un nivel de atención alto durante un largo periodo de tiempo. Los campos de aplicación engloban todos los escenarios posibles, desde vehículos de conducción autónoma hasta aviones de vigilancia.

Por supuesto, en robótica marítima también se ha empezado a investigar al respecto [5], pudiendo encontrar proyectos de automatización como el desarrollado en el MIT [4] para el mapeado de entornos marítimos, montado sobre un kayak al que se ha añadido la sensorización y los motores.

Por otra parte, están surgiendo empresas dedicadas en exclusiva al desarrollo de sistemas autónomos marítimos[3], como ASV [2], la cual ofrece soluciones tanto de carácter militar como civil, o "Maritime Robotics" [8] con un propósito similar.

La mayoría de estos sistemas son adaptaciones de vehículos marítimos existentes a los que se les añade la electrónica necesaria[7]. Esto, por un lado tiene sus ventajas, puesto que se tratan de elementos ampliamente probados y fiables. Pero, por otro lado, integrar el nuevo hardware al ya existente, modificando incluso la funcionalidad de alguno de los elementos propios del mismo para adaptarlos, puede ocasionar problemas posteriores. Además, el agua es un entorno dinámico, con corrientes sin un patrón claramente, a priori, predecible, lo que dificulta aún más el control.

Teniendo en cuenta esto, se trata de diseñar un prototipo a partir de un barco radiocontrol que únicamente posee los motores. Además, surge la necesidad de establecer unos requisitos principales que aseguren la escalabilidad del sistema para futuras ampliaciones funcionales del mismo.

De entrada, dichos requisitos se centran en un diseño modular y sencillo que, además, está restringido por el espacio útil del barco radiocontrol empleado. Por otro lado, aun tratándose de un sistema autónomo, debe existir comunicación para el seguimiento y captura de datos o, en un momento dado, el control manual del mismo.

En esta primera versión, se emplea un barco utilizado para desplegar líneas de pesca. Se trata de un modelo con un amplio espacio en su interior y con un par de motores de control diferencial, los cuales habrá que adaptar a las necesidades del diseño para poder realizar un control de la velocidad adecuado, que permita variar el ángulo de los giros a fin de orientar el barco correctamente.

2 OBJETIVOS

Al tratarse de una primera aproximación, los objetivos tratan, sobre todo, de definir las bases e identificar los problemas para obtener una visión general de por donde hay que seguir investigando y que partes son las más críticas. Por lo tanto, podemos englobar los objetivos en dos tipos:

Objetivos de diseño:

- Sistema modular para facilitar la ampliación del mismo.
- Elección e integración de la sensorización necesaria.
- Modificar el barco original para adaptarlo a las

nuevas necesidades.

Objetivos funcionales:
- Obtener telemetría en línea.
- Control de la velocidad y dirección.
- Establecer un protocolo de comunicación.
- Localización relativa del barco dentro de la zona de actuación del mismo.

3 DISEÑO

A la hora de diseñar el sistema, hay que lidiar con una serie de restricciones asociadas a la propia estructura del modelo elegido como base (Figura 1). La principal restricción es el espacio útil aprovechable. Aunque este modelo posee un interior amplio, hay que tener en cuenta su morfología a la hora de plantearse la forma de agregarle los nuevos componentes.



Figura 1: Barco radiocontrol utilizado.

Por otra parte, debido al entorno en donde deberá actuar, hay que asegurarse que la electrónica no sufre daños debido a la humedad o a la entrada de agua. En este aspecto, el barco viene preparado con un sellado hermético en sus juntas de goma a las que solo habrá que añadir grasa de silicona para mantener la impermeabilidad necesaria.

El modelo base integra, de serie, un par de motores eléctricos de corriente continua encapsulados en una estructura con los engranajes necesarios para la transmisión del movimiento a las turbinas (Figura 2).



Figura 2: Bloque de motores.

El control de la dirección es diferencial y no posee contramarcha debido al propio diseño de las aspas. Además, los motores son controlados por un sistema electrónico a partir de un mando a distancia, diseñado para una función de “todo o nada” sobre los mismos, por lo que, de entrada, no es posible controlar la velocidad de los motores.

3.1 SENSORIZACIÓN

Una primera necesidad, clave para conseguir los objetivos propuestos, es poder controlar la velocidad de los motores por separado. Para conseguir esto, previamente, hay que obtener información de la velocidad actual y diseñar a posteriori un controlador que compense el error existente entre la misma y la velocidad deseada.

La solución propuesta es la creación de un encoder magnético que mida el número de revoluciones por minuto de cada turbina (Figura 3). Este sensor consta de un detector Hall, el cual reacciona al ser expuesto a una determinada magnitud de un campo magnético. El modelo utilizado es el AH182/AH183 debido a que permite altas frecuencias, lo cual es necesario si se quiere medir velocidades altas.



Figura 3: Encoder magnético.

Para la localización se emplea el sistema GPS, por lo que se usa un módulo que integra antena y comunicación serie. Esta es una solución parcial al problema de la localización, ya que el error inherente del GPS, unido al pequeño tamaño del barco, aumenta la incertidumbre de la posición real, por lo que será necesario tenerlo en cuenta a la hora de diseñar el controlador de guiado. Por otro lado, si el barco se encuentra detenido durante un periodo de tiempo suficiente, debido a la dinámica del agua, su orientación puede cambiar, y el GPS no podrá medir la nueva orientación hasta que no se ponga en marcha. Para solventar este problema, se utiliza una unidad inercial, modelo CHR-UM6 de CH Robotics [6], la cual incluye un magnetómetro, con el que se mantendrá información útil sobre la orientación en estados de reposo.

3.2 INTEGRACIÓN

El núcleo de procesamiento principal elegido es un microcontrolador Arduino, modelo Due [1]. Se encarga de procesar los datos desde los sensores e implementar los controladores.

Los encoders magnéticos se conectan a un par de pines digitales del Arduino y, mediante el uso de interrupciones sobre esos pines, se realiza la estimación de las revoluciones por minuto. Hay que destacar que existen varios métodos para realizar este cálculo, siempre buscando una relación adecuada entre el número de actualizaciones de la información por unidad de tiempo y la precisión de dicha información. Por ejemplo, en este caso concreto, debido a la baja velocidad punta de los motores, no es necesario obtener actualizaciones con una frecuencia elevada, sino que se ha optado por realizar el cálculo una sola vez por segundo. Esto condiciona al sistema completo a adaptarse a esta frecuencia de actualizaciones, ya que es directamente responsable de la velocidad de reacción ante los eventos. Como se establece una restricción temporal, será esta la que actúe como desencadenante del evento, por lo que la manera de calcular las revoluciones por minuto será contar el número de revoluciones por segundo (tiempo de acumulación) y estimar (1).

$$RPM = RPS \cdot 60 \qquad (1)$$

De esta forma se puede configurar la velocidad de actualización fácilmente, variando el tiempo de acumulación al deseado y realizando la conversión necesaria (2). Aunque hay que tener presente que reducir el tiempo de acumulación, hace aumentar el error. En concreto, para T_{acum} = 60s (situación ideal) el error sería de una revolución por minuto (e = 1), ya que, como mucho, el motor podría haberse quedado justo antes de la posición del sensor, con lo que realmente sí que habría dado la vuelta, pero no haberla contabilizado. Si ahora se reduce el tiempo a la mitad, se puede deducir que el error sería el doble. En general, reducir el tiempo, aumenta el error proporcionalmente, efecto que hay que tener en cuenta.

$$RPM = \frac{60 \cdot n}{T_{acum}} \quad (2)$$

Otra manera de realizar este cálculo trataría de usar, como evento desencadenante, cada una de las revoluciones del motor. Esto tiene ciertos inconvenientes. Por un lado, la frecuencia de actualización dependería de la velocidad actual del motor, lo cual implica que, a velocidades bajas, el sistema respondería más lentamente. Por otro lado, como ahora el tiempo de acumulación es variable, el error también lo es, siendo además muy alto. Por lo tanto, este segundo método ha sido descartado.

El GPS se conecta mediante uno de los pines serie del Arduino. Aunque el baudrate es configurable, viene preconfigurado a 9600 baudios, cantidad suficiente para recibir las tramas NMEA correctamente.

La precisión es variable, pudiendo tener errores de varios metros, lo cual es un grave problema al tratarse de un barco de menos de un metro de longitud. Además, localizarlo correctamente es crucial a la hora de trazar trayectorias o esquivar obstáculos. Se requiere de otros métodos de apoyo para evitar colisiones y aumentar la precisión.

Lo primero que se ha hecho es conseguir una imagen o mapa aéreo de la zona de pruebas (Figura 4). Para poder posicionar el barco dentro del mismo, se han medido las posiciones en varios puntos concretos invariantes, tras lo cual se obtiene, mediante triangulación, la posición relativa del barco en dicho mapa. Cuantos más puntos de referencia se tengan, más precisa será la localización.



Figura 4: Imagen aérea de la zona de pruebas.

Por otro lado, para evitar colisiones, se crea una máscara binaria (Figura 5) que, superpuesta al mapa, delimita las zonas de agua seguras, definiendo un margen de anchura proporcional al máximo error posible del GPS. El sistema de trazado de trayectorias evitará estas zonas, centrándose únicamente en las áreas permitidas del mapa. Esta técnica también se puede emplear para delimitar zonas de agua a las que no se quiera que el barco acceda por cualquier motivo.



Figura 5: Mascara binaria que delimita las zonas accesibles por el barco.

Además de para el posicionamiento, el GPS se emplea para obtener la orientación, necesaria si se quiere ajustar la trayectoria. Esto tiene un problema, ya que, si el barco está en reposo sobre el agua, no se puede calcular su orientación al no haber movimiento. Por lo tanto, partiendo de esta situación, debe pasar un tiempo antes de empezar a tener en cuenta la información sobre la orientación que da el GPS. La manera de subsanar esto es empleando un magnetómetro, que sea capaz de dar una buena medición de la orientación en todo momento.

Aunque la precisión del magnetómetro es variable, ya que no en todos los puntos el campo magnético está orientado de la misma forma ni tiene la misma dirección, es una buena medida inicial para que el controlador empiece a mover el barco hacia la dirección correcta.

En cuanto al cálculo de las trayectorias, en este primer prototipo únicamente se contempla la línea recta entre la posición actual y la destino.

3.3 CONTROL

El sistema se compone de tres controladores, el primero se encarga de ajustar los motores a la velocidad, medida en revoluciones por minuto, que se desee. Puesto que, como se ha nombrado anteriormente, la frecuencia de actualización, y por tanto la velocidad de reacción, se ha fijado a una vez por segundo y como a velocidades altas obtener una gran precisión no es necesario, basta con implementar un controlador proporcional (3).

$$V_S = K_P \cdot E_a \quad (3)$$

El cálculo de la constante K_P se realiza mediante experimentación, variando su valor y observando las características del sistema, como el tiempo de subida y la sobre-oscilación (Figura 6). Al final se optó por un valor KP = 0.05, que es el que mejor relación tiempo-oscilación tenía.

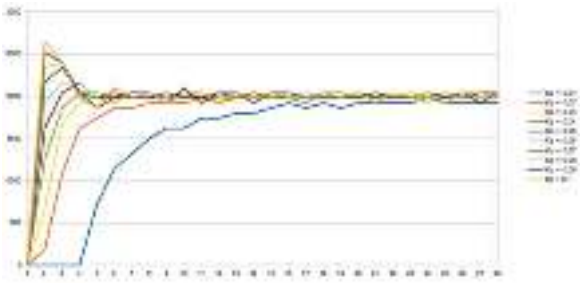


Figura 6: Resultados del controlador.

Midiendo como ajustaba el controlador las revoluciones por minuto utilizando este valor y variando las rpm deseadas (Figura 7) se obtiene buenos resultados.

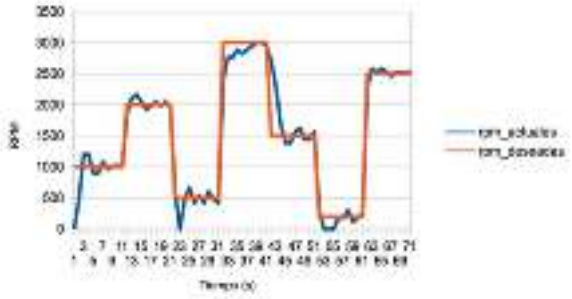


Figura 7: Control a diferentes RPM.

Lo siguiente es controlar la orientación del barco para encararlo al punto de destino. En este punto hay dos situaciones. Por un lado, cuando el barco se encuentra en reposo, es preciso emplear la información proveniente del magnetómetro hasta conseguir una orientación adecuada. Aunque no es preciso esperar a que esté completamente alineado para comenzar a moverse linealmente, si que habrá que establecer un umbral, a partir del cual comenzar a desplazarse.

Por otro lado, una vez se encuentra en movimiento, hay que mantener la orientación adecuada durante todo el trayecto, para lo cual se puede utilizar una combinación de la información dada por el magnetómetro junto con la recibida del GPS, lo que reduce el error de orientación.

Por último, es necesario controlar la velocidad lineal desde el momento en que empieza a desplazarse hasta que llega al punto destino, reduciéndola gradualmente a partir de un punto determinado con el objetivo de detener el barco justamente sobre dicho punto, evitando tener que hacer maniobras

posteriores o cometiendo errores en la precisión.

La combinación de estos controladores y el ajuste de los umbrales (Figura 8) permiten una navegación más fluida y eficiente.

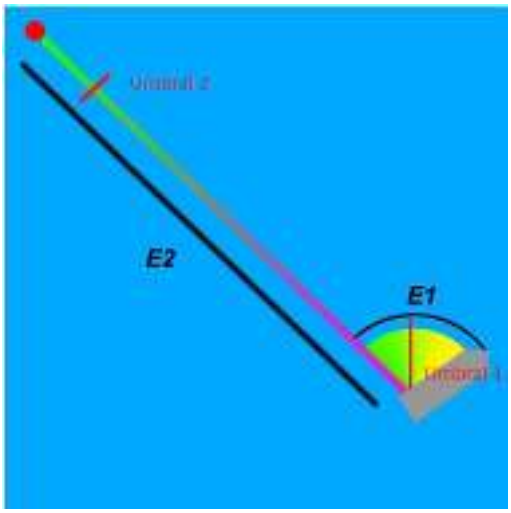


Figura 8: Fases de Control, Errores y Umbrales.

3.4 COMUNICACIÓN

El sistema elegido para comunicar el barco con el software de monitorización es un protocolo simple síncrono sobre Xbee [10].

El protocolo consta de una trama de datos encapsulada en un paquete con una cabecera indicando el tipo de dato que contiene y el inicio de la trama (Tabla 1). El barco hace de servidor, recibiendo la solicitud de acciones desde tierra. Estas acciones son las de escribir o leer sobre registros, ver la tabla 2. Cuando se solicita una acción de escritura se espera una trama de confirmación, las cuales son tramas con la sección de datos vacía. Si por el contrario, la acción es de lectura, la trama recibida contiene los datos solicitados.

Tabla 1: Trama básica.

Sección	Inicio trama	Acción	Datos	Fin
Bytes	3	2	n	1
Valor (Hex)	73 6E 70	varía	varía	40

El inicio de la trama sirve para empezar a capturarla cuando se recibe la secuencia correcta de bytes. Poner tres bytes de cabecera disminuye la probabilidad de que, por errores en la transmisión, el sistema crea que ha encontrado una trama cuando en realidad no lo es. El fin de la trama tiene un propósito similar, y no es tan crítico que sea detectado, puesto que, de no hacerlo y no ser enviada la confirmación o los datos solicitados, se puede volver a enviar la trama nuevamente. Los datos varían dependiendo de lo que se solicite, pudiendo estar vacío en caso de una trama de confirmación.

Tabla 2: Acciones sobre registros.

Acción	Descripción
0x0000	Leer rpm motor 1 y 2 (simultaneo).
0x0001	Leer rpm motor 1.
0x0002	Leer rpm motor 2.
0x0003	Leer datos GPS (posición, velocidad y orientación).
0x0004	Leer datos del magnetómetro.
0x0005	Leer datos del acelerómetro.
0x0100	Escribir rpm motor 1 y 2 (simultaneo).
0x0101	Escribir rpm motor 1.
0x0102	Escribir rpm motor 2
0x0103	Escribir posición destino.
0x01F0	Pasar a control manual.
0x010F	Pasar a control autónomo.
0x0200	Trama de vinculación.

Existe la posibilidad de poner el barco en modo manual, con lo que se le podrá variar la velocidad de los motores. Cuando se encuentre en este modo, el barco ha de recibir constantemente una trama de vinculación o de lo contrario se detendrá por completo hasta que vuelva a recibir dicha trama. Esto es un sistema de seguridad en casos extremos.

4 PRUEBAS

El lugar elegido es un escenario de aguas tranquilas, sin corrientes y con poco viento para comprobar sin interferencia externa el funcionamiento integral del sistema. Las pruebas se han organizado de forma gradual, comenzando por tratar de mover el barco de manera manual, recibir telemetría y por último darle posiciones a las que debe moverse de manera autónoma.

Durante el transcurso de dichas pruebas se observó un comportamiento inadecuado desde un primer momento. El ajuste proporcional de los motores para hacerlo virar no respondía a lo esperado debido a la poca velocidad de los mismos y a su diseño original de “todo o nada” que, aunque se modificaron para permitir ajustar la velocidad, seguían necesitando un pulso demasiado alto para vencer la inercia inicial, lo que hacía que los giros fueran demasiado bruscos e

impedía un ajuste fino de la orientación.

Por otro lado, aunque la comunicación funcionó perfectamente, la distancia efectiva de la misma resultó ser muy reducida, apenas unos 5 metros. Atenuada por la propia carcasa del barco y la baja altura de la antera emisora.

En la figura 9 se puede ver el barco realizando las pruebas iniciales. Se probó a distintos valores de revoluciones por minuto, realizando giros con distintos ángulos de apertura.



Figura 9: Prueba del barco.

5 CONCLUSIONES Y TRABAJO FUTURO

Aunque el sistema resultó ser robusto y funcional, el problema de los motores hace que sea inviable para el objetivo propuesto, por lo que habrá que plantearse su sustitución por otros de mayores prestaciones.

En cuanto a la comunicación, es preciso buscar alternativas que permita un mayor alcance. Pudiendo empezar por montar una antena externa en la carcasa del barco y aumentar la altura del módulo emisor en tierra.

Una vez solventados los problemas de este primer prototipo, se podrá trabajar en ampliar la funcionalidad del mismo:

- Creación de un software de monitorización más avanzado y definitivo que permita generar informes de actividad y alertas.
- Estudiar la posibilidad de integración de un sistema de visión que ayude al trazado de trayectorias y evite obstáculos.
- Reducir el error de localización, combinando los valores inerciales de la IMU con la información del GPS [9].
- Mejorar los sistemas de control para aumentar la velocidad de los motores.

Agradecimientos

Esta trabajo ha sido financiado por el Ministerio de Economía Español, por fondos FEDER Europeos y por fondos de la Generalitat Valenciana por medio de los proyectos de investigación DPI2012-32390 y PROMETEO/2013/085.

Referencias

[1] Arduino Due: http://arduino.cc/en/Main/ArduinoBoardDue

[2] ASV Unmanned marine systems: http://www.asvglobal.com/

[3] Bibuli, M., Caccia, M., Lapierre L., Bruzzone G. (2012). Guidance of Unmanned Surface Vehicles- Experiments in Vehicle Following. IEEE Robotics & Automation Magazine. pp. 92-102.

[4] Chadwick Leedekerken, Jacques (2011) Mapping of Complex Marine Environments using an Unmanned Surface Craft. *Massachusetts Institute of Technology*. http://hdl.handle.net/1721.1/68492

[5] De la Cruz García, J.M., Aranda Almansa J., Girón Sierra, J.M. (2012). Automática marina-una revisión desde el punto de vista del control. *Revista Iberoamericana de Automática e Informática industrial,* no. 9. pp. 205–218.

[6] IMU CHR-UM6: http://www.chrobotics.com/

[7] Larson, J., Bruch, M., Ebken, J. (2006). Autonomous navigation and obstacle avoidance for Unmanned surface vehicles. *SPIE Unmanned Systems Technology VIII*. pp. 17-20

[8] Maritime Robotics: http://maritimerobotics.com

[9] Pifu Zhang, Jason Gu, Evangelos E. Milios, Peter Huynh (2005) Navigation with IMU/GPS/Digital Compass with Unscented Kalman Filter. *IEEE International Conference Mechatronics and Automation*. Vol. 3. pp. 1497-1502.

[10] Xbee Technology: http://www.digi.com/xbee/

Realtime AUV Terrain Based Navigation with Octomap

Guillem Vallicrosa[1], Albert Palomer[1], David Ribas[1] and Pere Ridao[1]

[1]Department of Computer Engineering, Universitat de Girona, 17071 Girona, Spain,
gvallicrosa—apalomer—dribas—pere@eia.udg.edu

Abstract

This paper address the problems of Terrain Based Navigation and Occupancy Grid Mapping. Both attacked separately, but with same tools, in order that is feasible in future works to implement a Simultaneous Localization and Mapping (SLAM). Realtime implementation on the real vehicle Girona500 AUV is achieved by means of a parallelized execution of a Particle Filter and making use of the Octomap library for Occupancy Grid Mapping. Particle filter is implemented as a single node in the vehicle's software architecture in ROS. A motion model based on DVL-AHRS navigation and a measurement model for the sonar multibeam are developed. Design specifications are discussed and tests are carried out in a water tank. Several parameters involving the Particle Filter (number of particles, number of beams, uncertainty of measurements) are studied and compared. Experiments are validated with a ground truth obtained by means of an accurate vision based method. Realtime navigation in a know map is successful with small errors on the xy-plane and a good accuracy on depth.

Keywords: Terrain Based Navigation, Octomap, AUV, Particle Filter, Realtime.

1 INTRODUCTION

The Terrain Based Navigation (TBN) problem, is the name used by the marine robotics community to refer to the more general problem of mobile robot localization with a known map. In particular, TBN solves for the robot pose given an a priori known map, fusing information from dead reckoning navigation with map referenced observations.

TBN has been mainly applied to aerial and underwater vehicles. During the last years the accuracy and extension of the maps has been increased considerably, and TBN has been adopted as a method to complement Inertial Navigation Systems (INS), as an alternative when GPS is not available. Underwater, TBN is still at a research stage, not having evolved into a commercial product. During the last two decades, the scientific community working with mobile robots, has pushed forward the boundaries of the knowledge facing an even more challenging problem: the Simultaneous Localization and Mapping[4]. Recently, these techniques have started to be slowly adapted to underwater environments[10, 12, 16]. Both SLAM and TBN have a great potential to improve the autonomy of the underwater vehicles, allowing AUVs to freely move abroad the areas of coverage of the acoustic transponder networks.

Using navigation sensors it is possible to measure the robot velocity, acceleration and attitude, to be used as the input for the dead reckoning equations needed to compute the robot pose. Due to the noisy measurements, the position estimate will rapidly grow without bounds. In domains like underwater, covert operations, or out space applications the GPS is not available to bound the drift. Absolute positioning fixes can be provided underwater by acoustic positioning like the Long Baseline System (LBL), the Short Baseline (SBL), the Ultra Short Baseline (USBL) or the GPS eqquiped intelligent Buoys (GIB). However, these systems require time for deployment and constrain the vehicle to a certain area of coverage. TBN has a potential to become an alternative to satellite navigation and acoustic transporter networks. TBN takes advantage of existing digital terrain maps of target area, where the vehicle shall navigate. Conventional dead reckoning navigation methods provide a prior estimate of the robot pose within the map. Then, using exteropceptive sensors, terrain observations are obtained to be correlated to the a priori known map in order to compute the robot pose.

1.1 Problem definition

Let:

- $\{\mathbb{E}\}$ be a inertial earth fixed frame.
- $\{\mathbb{B}\}$ be the vehicle fixed frame.
- $x_t = [x\, y\, z\, \phi\, \theta\, \psi]$ be the vehicle pose at time t referenced to $\{\mathbb{E}\}$.

- $\upsilon_t = [u\, v\, w\, p\, q\, r]$ be the vehicle velocity at time t referenced to $\{\mathbb{B}\}$.
- $\mathbb{M}(x_t)$ be a digital elevation map of the environment, assumed to be known a priori.
- z_t be the vector of observations coming from exteroceptive sensors (altimeter, pressure or multibeam sensor).

TBN consist on estimating the vehicle velocity υ_t, referenced to the vehicle $\mathbb{B}$ frame, and/or the vehicle pose η_t, referenced to the inertial $\mathbb{E}$ frame, by matching observations z_t of the measured depth (altitude for UAVs) and ranges with the terrain elevation map stored in $\mathbb{M}$. For AUVs, the linear velocity is provided by a Doppler Velocity Log (DVL) sensor, while the angular velocity is commonly provided from Fiber Optical (FOG) or Ring Laser Gyros (RLG).

For an AUV the measurement model is given by:

$$r_{t,i} = \mathbb{M}_i(\eta_t) - d_t + w_t, \qquad \forall 0 \leq i \leq N$$

where w_t is the measurement noise, d_t the vehicle depth, N the number of beams, and $r_{t,i}$ the projection of range measurements to the vertical axis.

1.2 Bayesian estimation techniques for TBN

Let us assume x_t to be a Markovian state and:

- $p(x_{t-1})$ be the probability density function (pdf) describing the probability of the robot of being at a certain pose at time $t-1$,
- $p(x_t|x_{t-1}, u_t)$ be the state transition probability, also known as motion model, which allows to predict the robot pose after certain input u_t,
- $p(z_t|x_t)$ be the measurement probability which given a map $\mathbb{M}()$ provides the probability of observing z_t when being at state x_t,

then, the TBN problem consists on solving for $p(x_t)$, given $p(x_{t-1})$, $p(x_t|x_{t-1}, u_t)$ and $p(z_t|x_t)$. In the context of bayesian estimation this can be done through the Bayes filter (BF):

$$p(\overline{x}_t) = \int p(x_t|x_{t-1}, u_t)p(x_{t-1})dx_t \qquad (1)$$

$$p(x_t) = \eta p(z_t|x_t)p(\overline{x_t}) \qquad (2)$$

In the most general case, the BF cannot be implemented because it relies on the close form solution of the integral shown in (1). Under some conditions, the BF can be implemented using parametric filters like the Kalman Filter (KF), the Information Filter (IF) or their non-linear counter parts like the Extended Kalman Filter (EKF), the Extended Information Filter (EIF) or the Unscented Kalman Filter (UKF); or non parametrics filters like the Point Mass Filter (PMF), the Particle Filter (PF) or the Rao-blackwellised Particle Filter (RBPF) [15]. During the last years, these probabilistic approaches have become dominant in TBN applications, because they explicitly deal with the uncertainties in both, the motion and the measurements models. It is well known that these uncertainties play an important role to solve the robot navigation and mapping problems.

Solutions to the TBN can be optimally implemented as a Kalman Filter [2] when pdfs are Gaussian and state transition function is a linear stochastic equations. For more realistic and complex scenarios, when the motion and/or the measurement model are nonlinear, Extended Kalman Filter [14] should be used.

Particle Filters (PF), also called sequential Monte-Carlo (SMC) methods, can deal with nonlinear motion and/or measurement models without relying on linearization techniques. PFs use a point mass representation of the density function to approximate the robot pose posterior [1, 3]. Hence a set of random samples, "particles", are used to represent the pdf, being able to represent a much broader space of distributions than Gaussians, like multi modal pdfs. A widely used variant of the PF is the *Sampling Importance Resampling* PF (SIR-PF) [5, 9, 11, 15], designed to solve the PF degeneracy problem where after a while all but one particle will have negligible weight, by introducing a resampling step to eliminate particles with small weight while duplicating particles with high weight.

The goal of this paper is to implement a realtime TBN based on a PF in the Girona500 AUV with an AHRS-DVL motion model and corrections provided by a multibeam sonar profiler and a known map.

Organization of this paper is on Section 2 general idea of the PF used is explained. Section 3 introduces Octomap for Occupancy grid Mapping, followed by Section 4 with the detailed implementation of the PF. Finally Section 5 & 6, present the results and conclusions of the experiments.

2 PARTICLE FILTER MCL

The Particle Filter (PF), is a non-parametric solution to the Bayes Filter. The PF discretizes the

pdf by a set of samples or "particles". Each one of this particles can be seen as the possibility of the vehicle being at that position and attitude. PF is divided in three main steps. Prediction or state transition model, where particles position and attitude is predicted according to proprioceptive sensors. Weighting or measurement model, where particles are weighted according to their likelihood to data provided by exteroceptive sensors. And resample, where particles with small weights are discarded. A constant velocity model

Algorithm 1 Motion Model

function motion_model (x_{t-1}, u_t) **:**

$$\bar{\phi}_t = \phi_t + \mathcal{N}(0, \sigma_\phi) \qquad \bar{u}_{t-1} = u_{t-1} + \mathcal{N}(0, \sigma_u)$$
$$\bar{\theta}_t = \theta_t + \mathcal{N}(0, \sigma_\theta) \qquad \bar{v}_{t-1} = v_{t-1} + \mathcal{N}(0, \sigma_v)$$
$$\bar{\psi}_t = \psi_t + \mathcal{N}(0, \sigma_\psi) \qquad \bar{w}_{t-1} = w_{t-1} + \mathcal{N}(0, \sigma_w)$$
$$\begin{bmatrix} \bar{x}_t \\ \bar{y}_t \\ \bar{z}_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} + Rot_{\bar{\phi}_t, \bar{\theta}_t, \bar{\psi}_t} \cdot \begin{bmatrix} \bar{u}_{t-1} \\ \bar{v}_{t-1} \\ \bar{w}_{t-1} \end{bmatrix} \cdot t$$

return $\left[\bar{x}_t,\ \bar{y}_t,\ \bar{z}_t,\ \bar{\phi}_t,\ \bar{\theta}_t,\ \bar{\psi}_t\right]$

on 6 DoF (Alg. 1) is used for the motion model. The inputs are the linear velocities measured by the DVL and the attitude measured by the AHRS, with their corresponding variances. For the measurement model (Alg. 2), the likelihood of the complete multibeam scan can be measured by individual likelihoods of each beam. Each beam range is compared with the range that would see each particle in its current position. The difference between the real measurement and the expected one is weighted by a Gaussian function. Resampling is done by systematically resampling, using a random seed for the first choosing and constant increments on the seed for all other choosings.

Algorithm 2 Measurement Model

function measurement_model (x_t, z_t) **:**
Require: σ_z
  $w = 0$
  for $n = 1$ **to** N **do**
    if $z_t^{[n]} > 0$ **then**

$$w = w + \frac{1}{\sigma_z \sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{z_t^{[n]} - d_{cast}}{\sigma_z}\right)^2\right)$$

    end if
  end for
  return w

3 OCTOMAP

Octomap [8] is a library for occupancy grid mapping. Occupancy grid mapping of a 3D environment is highly memory consuming. This library uses octrees to represent occupancy, making mapping memory efficient and providing easy methods to insert and query maps. With this kind of representation, maps can be used for online path planning as well as for localization. This library is already integrated in ROS, providing a node to create maps online. They can also be created from existing 3D models to be used as a ground truth (Fig. 1), or by using the library at a lower level (Alg. 3). For each ray inserted on the octree

Algorithm 3 Mapping

function Mapping (x_t, z_t, o_t) **:**
Project z_t (range, bearing) to (x, y, z).
Transform sensor origin o_t from $\mathbb{B}$ to $\mathbb{E}$ using x_t.
Transform scan z_t from $\mathbb{B}$ to $\mathbb{E}$ using x_t.
for $i = 1$ **to** N **do**
  insertRay(o_t, z_t^i)
end for

map, the library will take care of determining all nodes traversed by the ray (ray tracing) and making them free by lowering their probability by a specified value. Finally, it will mark the last node as occupied by another specified value. When lots of scans are inserted, it is useful to call `prune()` method in order to fuse the nodes with same value and save memory.

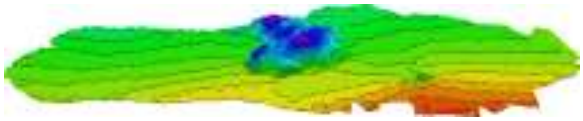


Figure 1: Example of underwater map obtained from a survey near Sant Feliu's coast ($850 \times 280 \times 30\,m$).

4 IMPLEMENTATION

Realtime TBN underwater is implemented in a single C++ class (Fig. 2). The choosing of C++ coding over simpler Python implementation stands to the following reasons. Octomap library can only be used in C++. OpenMP library allows easy parallelization of loops in C++ code. Performance on computations is better to achieve realtime. The main loop which corresponds to the Particle Filter, is triggered when a message from multibeam sonar arrives (red arrow). There are other small callback functions that are triggered when the other sensor messages arrive (green arrows). At the *UnderwaterLocalization* class instantiation, the octree map is loaded from a bonsai tree `.bt` file, the ROS parameters are loaded, and particles are initialized at random positions covering the whole CIRS water tank volume, which

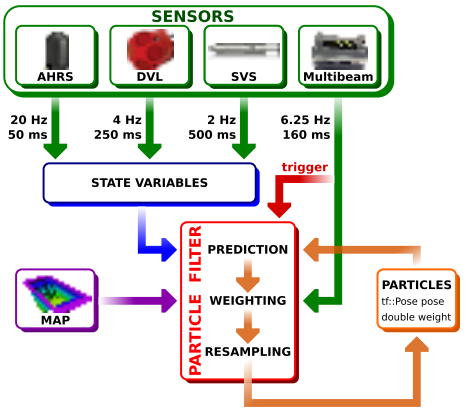


Figure 2: Block diagram of the *UnderwaterLocalization* class.

is the testing scenario. The attitude is initialized by the first reading of the AHRS plus a Gaussian noise. Each particle is composed of a `tf::Pose` which provides position and attitude, and a corresponding weight initialized to zero.

4.1 Sensor callbacks

When a message arrives to the node, a callback automatically saves its information in a variable if it is one of the navigation sensors (AHRS, DVL, SVS). On multibeam sonar, its callback triggers the whole particle filter. One might thinks that for each message of each sensor that arrives, a prediction of the PF could be run. In fact this should be the way. But looking at the statistics of frequency of each kind of message generated by the `navigator` node and the sonar driver (Table 1), this simpler solution was chosen. This solution of

Table 1: Frequencies and periods of each message.

	Freq (Hz)	T (ms)
AHRS	20	50
DVL	4	250
SVS	2	500
Multibeam	6.25	160

triggering the whole particle filter with the multibeam message, is feasible because the multibeam operates at a higher frequency than the DVL. The most important information on the prediction step is the velocities of the vehicle. Working at a higher frequency means that no DVL message will be lost. Because AHRS is fast, when the prediction is computed, the newest information of attitude is introduced in the filter. Finally the biggest concern is to be able to write a particle filter loop that takes less time than the period between multibeam sonar messages (160 ms).

4.2 Map queries

The octree map is used on the weighting step of the particle filter. The ranges read from the multibeam sensor have to be compared with the ranges that the particle itself sees from its position on the map. To query the ranges with the `rayCast()` function, all coordinates have to be given regarding to the map coordinate frame. Origin and endpoints of the multibeam sensor have to be transformed from the sensor frame to the center of the vehicle, and then with each particle information, to the frame of the map. The ray casting query returns the octree node that was hit. To obtain the range one must compute the norm of the difference between the origin of the multibeam on the particle o_i and the endpoint of each beam $e_{i,j}$. The range $r_{i,j}$ for the particle i and beam j is:

$$r_{i,j} = ||o_i - e_{i,j}||$$

Running the `octomap_server` node provided by the Octomap library and using the odometry provided by the EKF dead reckoning, a map was built to observe how similar was to the one created from the blueprint of the CIRS water tank (Fig. 3). As



Figure 3: Comparison between Octomap obtained with dead reckoning and blueprint.

can be observed on the bottom-right part of the real data, some of the beams that should see the vertical wall (dark blue) in fact they see a flat ground beyond the wall. This is due to the high reflective nature of the CIRS water tank. Actually because of the echoes and reflection, some beams see the ground instead of the wall. This problem is not likely to happen in natural environments, so to overcome it easily, and without having to calculate the new beam direction after hitting the vertical wall, a simple solution was implemented by adding an artificial ground behind the vertical wall. Then in the implementation, when a ray casting hits a node, this node is taken as the new origin of the beam and the direction continues to be the same. If a second node is hit, then the probabilities are shared half-half. If the second node is not hit, the probabilities remain the same as not having any reflection.

4.3 Particle filter loop and parallelization

Because particle filter has complexity $O(n)$, being n the number of particles, and the computations for each particle are independent to the other particles, to speed up the PF some parallelization is

introduced with OpenMP library. On the prediction step, a single for loop over all the particles runs the motion model. On the weighting step a double for loop, one for each particle and one for each beam is used. A small speedup can be obtained by precomputing a part of the exponential weighting outside $\left(\frac{1}{2\sigma_z^2}\right)$, which will be the same value for all particles and beams. For parallelization, only the outermost loop is used, the one over all particles.

In the measurement model, instead of the normalized Gaussian the non-normalized Gaussian $\left(\exp\left(-\frac{1}{2}\frac{z^2}{\sigma_z^2}\right)\right)$ is used. This means that when a beam of a particle is in complete agreement ($z = 0$), its weight is one. To normalize the particle weights between zero and one, it is divided by the number of valid beams (those with a valid range). Particles are also checked first if they are inside the water tank volume. If they are not inside, their weight is set directly to zero and no measurement likelihood is checked. Tests were made on the first 15 seconds of a dataset (Table 2). The best improvement between both parallelizations is given by parallelizing the update step on the Particle Filter. In the prediction step, the improvement is not appreciable because the operations inside the loop are simple. The results are even worse for a small number of particles. It takes more time to parallelize, than the time saved using it. In the update step, more complex computations and map queries are performed. The speedup obtained by parallelizing this loop is around 4x on an Intel Core i7-2600 CPU @ 3.40 GHz x 8, 64bit OS and 8 GB of RAM. This means that parallelizing the update step is mandatory, while the prediction step can be useful if the number of particles is big enough. Results of parallelizing PF are almost half of the available time 160 ms.

5 RESULTS & DISCUSSION

The CIRS building has a $16 \times 8 \times 5\, m^3$ water tank (Fig. 4) to do small experiments with underwater vehicles. The real measurements have never been calculated. The only source for building a map is from the CAD drawings. The water tank was first modeled with Blender and then exported by voxelization to an octree representation compatible with Octomap.

5.1 Girona500 AUV

Girona500 (Fig. 5) is an Autonomous Underwater Vehicle (AUV) designed as a research platform with capacity to be reconfigurable for many different applications [13], ranging from the classical sonar and video imaging surveys to inspection and the challenging autonomous intervention tasks. Although many sensors are available, for this paper, only Attitude and Heading Reference System (AHRS), Doppler Velocity Logger (DVL), and a multibeam sonar profiler, will be used. A Imagenex DeltaT multibeam sonar, provides the profile of the seabed by means of echosounders and interferometry. This sensor is specifically designed to produce bathymetric maps of large areas of the seabed. The Girona500 software architecture COLA2 (Component Oriented Layer-based Architecture for Autonomy) is responsible for the vehicle navigation, guidance and control. It is completely written in ROS, currently at *Fuerte* version. Each node in the architecture is designed to be small and easy to understand, so modifications and improvements are easily done. Inputs for TBN implementation are taken from the `navigator_node` which provides an abstraction of all the navigation sensors of the vehicle. Multibeam sonar did not have an online driver compatible with ROS, so it had to be coded and properly tested before starting this work.

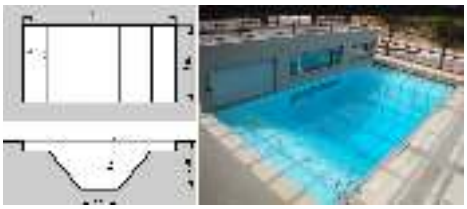


Figure 4: CIRS water tank dimensions and picture.

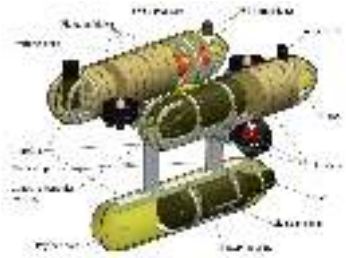


Figure 5: Girona500 AUV schematic.

5.2 Experimental setup

The robot was teleoperated through a network connection. All the messages were recorded into a ROS bagfile. This file keeps all the messages with their corresponding timestamps. Using ROS provided tools, it is easy to replay all the data as if the robot was running. This method is specially useful to compare algorithms as if they were running on the robot, and being able to change parameters and observe how they affect at the output for the

Table 2: Time to execute the particle filter loop (no visualizations active).

	500 particles			1000 particles		
	Mean (ms)	Min (ms)	Max (ms)	Mean (ms)	Min (ms)	Max (ms)
No parallelization	178.2	115.4	243.6	367.1	252.3	518.0
Par. prediction	181.6	119.1	248.9	351.6	244.1	463.5
Par. update	46.3	33.4	75.7	86.9	61.9	114.8
Par. pred. and update	46.3	28.4	64.7	86.6	62.2	118.3

same dataset.

The Imagenex DeltaT multibeam sonar, was mounted on the payload volume of the Girona500 AUV, along with the Bumblebee stereo camera. AHRS and DVL were used for the prediction step of the Particle Filter, while the multibeam was used on the weighting step. The bumblebee stereo camera was mounted in order to compute the ground truth based on the photomosaic located at the bottom of the water tank.

5.3 Ground truth

An absolute ground truth is available for validating the experimental results carried out on the CIRS water tank through the photomosaic located at the bottom of the water tank. Using the available camera on the AUV, it is possible to obtain estimates of absolute camera poses, together with their uncertainty, by registering camera images to the original image of the poster.

The estimation method uses a parameterization for the pose comprising the 6 DoF, rotations and translations referenced to a local frame. The method relies on the identification of point correspondences using feature-based robust matching[6]. The outcome of the matching are two lists of correspondences of points in the camera image and the poster image respectively. These lists are related by an observation function that takes into account the pose, the camera calibration and the scale of the mosaic poster. The uncertainty in the pose is computed following the approach described in [7].

Absolute positioning of the photomosaic on the bottom of the water tank is not known. The world origin for the outputs of the particle filter is situated at the center of the tank on the surface of the water, with the axis being north on x axis, east on y axis and depth on z axis. Ground truth was aligned empirically on all axis by using the mean difference of all the results obtained by the particle filter. Rotations were not taken into account. Thereby the results can differ from the real ones, depending on this small offset.

5.4 The experiments

Different parameters of the PF were evaluated, having a total of $5 \times 5 \times 3 = 75$ experiments. Five different number of particles (250, 500, 1000, 2000, 3000), five different number of beams (240, 48, 24, 12, 8), and three different measurement uncertainties (0.07, 0.12, 0.17). Different number of particles and beams can affect significantly to the realtime processing of the particle filter. If less particles are used realtime its achievable, but representation of pdf may not be accurate enough to localize correctly. If less beams are used, from the total of 240 beams provided by the multibeam, computations will also be faster, but only some reading are going to be used and localization can be affected.

Because PF is stochastic, each experiment was repeated 16 times using different particle initializations. A initialization with the maximum number of particles (3000) was created, and all experiments were carried out using a subset of this initial set of particles. This was repeated for the 16 different random initializations. Runtime of each experiment was of 100 s.

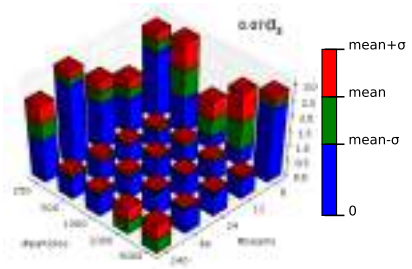
5.5 Results of Terrain Based Navigation

Results were evaluated by taking the mean error through time for each experiment in each iteration. This means of error were later used to calculate the mean and standard deviation (std) between same experiments with the different initializations. Results can be observed in Table 3 and Figure 6.

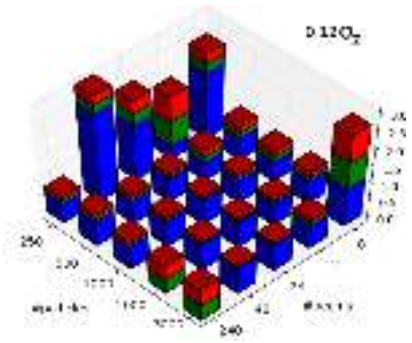
Results on the three different measurement uncertainty does not show any appreciable improvement in localization error, they follow similar patterns. According to number of particles, 250 particles is not enough to correctly localize the vehicle in the given scenario. For number of beams, 8 beams are not enough to achieve correct localization. All other combinations of parameters give similar errors (around 0.7 m error in xy-plane) with a small std (less than 0.10 m), which means the filter is robust and is not being affected by different initializations. Finally note that when using 240

Table 3: Localization mean error with standard deviation ($\bar{e} \pm \sigma_e$).

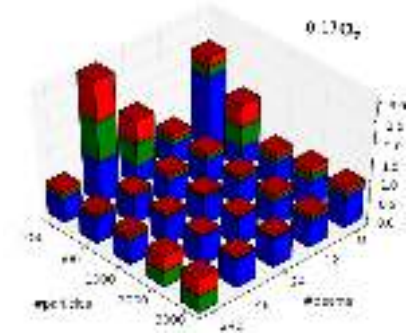
		#beams				
σ_z	#part	240	48	24	12	8
0.07	250	1.981 ± 0.554	2.888 ± 0.215	1.926 ± 0.349	1.506 ± 0.320	2.541 ± 0.317
	500	0.788 ± 0.078	0.768 ± 0.085	0.741 ± 0.040	0.663 ± 0.088	2.112 ± 0.912
	1000	0.771 ± 0.060	0.804 ± 0.065	0.808 ± 0.060	0.731 ± 0.109	1.038 ± 0.584
	2000	0.410 ± 0.425	0.829 ± 0.041	0.805 ± 0.050	0.699 ± 0.070	1.491 ± 0.959
	3000	0.402 ± 0.471	0.833 ± 0.025	0.805 ± 0.033	0.711 ± 0.055	2.665 ± 0.194
0.12	250	0.660 ± 0.103	2.760 ± 0.245	1.928 ± 0.339	1.095 ± 0.692	2.148 ± 0.266
	500	0.828 ± 0.102	0.758 ± 0.077	0.765 ± 0.078	0.719 ± 0.169	0.808 ± 0.169
	1000	0.744 ± 0.073	0.805 ± 0.061	0.777 ± 0.051	0.755 ± 0.087	0.748 ± 0.163
	2000	0.396 ± 0.410	0.809 ± 0.053	0.800 ± 0.051	0.704 ± 0.068	0.800 ± 0.122
	3000	0.351 ± 0.469	0.799 ± 0.029	0.804 ± 0.029	0.758 ± 0.102	1.874 ± 0.766
0.17	250	0.730 ± 0.101	2.122 ± 1.013	1.094 ± 0.602	0.706 ± 0.139	2.122 ± 0.241
	500	0.727 ± 0.053	0.737 ± 0.058	0.743 ± 0.085	0.766 ± 0.112	1.088 ± 0.566
	1000	0.703 ± 0.067	0.755 ± 0.049	0.765 ± 0.049	0.729 ± 0.064	0.764 ± 0.133
	2000	0.391 ± 0.405	0.799 ± 0.021	0.768 ± 0.027	0.755 ± 0.099	0.849 ± 0.170
	3000	0.335 ± 0.448	0.774 ± 0.027	0.790 ± 0.029	0.784 ± 0.082	0.939 ± 0.113



(a) Localization error with $\sigma_z = 0.07$.



(b) Localization error with $\sigma_z = 0.12$.



(c) Localization error with $\sigma_z = 0.17$.

Figure 6: Localization error and std (top of blue: mean-std, top of green: mean, top of red: mean+std).

beams and 2000 or 3000 particles, a high std is observed. This is due that computation time is expensive with big number of particles and beams, and adding the operation to write vehicle position on disk after each PF iteration (to compute error statistics) is to much expensive to still run in realtime, so some multibeam data is lost.

6 CONCLUSIONS

In this paper we have proposed and implemented an AUV navigation and mapping method based on the Octomap library available in ROS. To this aim, several developments were carried out. First, a motion model based on DVL-AHRS navigation and a measurement model for the sonar multibeam were developed. Next, after evaluating the conventional techniques used for terrain based navigation, a PF was selected as the preferred Bayesian estimation technique for solving the TBN problem in realtime. The system was implemented over the COLA2 software architecture and tested with the Girona500 AUV doing experiments in CIRS' water tank. Several filter configurations (number of particles, number of beams and measurement uncertainty) were tested. The results of the particle filter were compared against a ground truth estimated by means of an accurate vision based localization method. The vehicle altitude was very accurately estimated (around $0.2m$ error) and $0.7m$ accuracy for the horizontal (XY) position was also easily achieved.

Acknowledgements

This work was supported by the Spanish project DPI2011-27977-C03-02 (COMAROB) and two European Commission's Seventh Framework Program 2011-2015 projects: FP7-ICT-2011-7-288704 (MORPH) and FP7-INFRASTRUCTURES-2012-312762 (EUROFLEETS2).

References

[1] K. Anonsen and O. Hallingstad. Terrain aided underwater navigation using point mass and particle filters. In *Proceedings of the IEEE/ION Position Location and Navigation Symposium*, 2006.

[2] O. Bergem. Bathymetric navigation of autonomous underwater vehicles using a multibeam sonar and a Kalman filter with relative measurement covariance matrices. *Dr. Scient thesis, University of Trondheim, Norway*, 1993.

[3] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[4] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.

[5] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings*, volume 140, pages 107–113, 1993.

[6] N. Gracias. *Mosaic–based Visual Navigation for Autonomous Underwater Vehicles*. PhD thesis, Instituto Superior Técnico, Lisbon, Portugal, June 2003.

[7] R. Haralick. Propagating covariance in computer vision. In *Proc. of the Workshop on Performance Characteristics of Vision Algorithms*, Cambridge, UK, April 1996.

[8] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, Feb. 2013.

[9] R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filtering and Cramer-Rao lower bound for underwater navigation. *In 2003 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, 2003.

[10] A. Mallios, D. Ribas, and P. Ridao. Localization Advances in the Unstructured Underwater Environment. *Proceedings of the 9th Hellenic Symposium of Oceanography and Fishery*, 1:111 – 116, 2009.

[11] F. Maurelli, S. Krupinski, Y. Petillot, and J. Salvi. A Particle Filter Approach for AUV Localization. *OCEANS MTS/IEEE*, 2008.

[12] D. Ribas, P. Ridao, J. D. Domingo, and J. Neira. Underwater SLAM in Man-Made Structured Environments. *Journal of Field Robotics*, 25:898–921, 2008.

[13] D. Ribas, P. Ridao, L. Magi, N. Palomeras, and M. Carreras. The Girona 500, a multipurpose autonomous underwater vehicle. In *OCEANS 2011 IEEE - Spain*, pages 1–5. IEEE, June 2011.

[14] M. Sistiaga, J. Opderbecke, M. Aldon, and V. Rigaud. Map based underwater navigation using a multibeam echosounder. In *OCEANS'98 Conference Proceedings*, volume 2, 1998.

[15] S. Thrun. *Probabilistic robotics*, volume 45. ACM, 2002.

[16] S. Zandara, P. Ridao, A. Mallios, and D. Ribas. MBpIC-SLAM: Probabilistic Surface Matching for Bathymetry Based SLAM. *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV'2012)*, 2012.

Asistencia de Robots Colaborativos para Procedimientos de Sutura vía Cirugía Mínimamente Invasiva

E. Bauzano, B. Estebanez, I. García-Morales y V.F. Muñoz-Martínez
Severo Ochoa 4, 29590 Málaga, ebauzano@uma.es

Resumen

Los robots quirúrgicos permiten el manejo de instrumental quirúrgico mediante el empleo de sistemas teleoperados o la automatización de ciertas tareas de una intervención. Sin embargo, el objetivo final de estos sistemas siempre ha consistido en permitir al cirujano desarrollar una intervención sin la necesidad de asistencia humana. En este sentido, este artículo se centra en la implementación de un robot quirúrgico semi-autónomo capaz de asistir al cirujano durante un procedimiento de sutura dentro de un escenario de Asistencia Manual en Cirugía Laparoscópica (HALS). La metodología propuesta se basa en un sistema encargado de gestionar las actuaciones del robot dependiendo del gesto quirúrgico reconocido que haya realizado la herramienta del cirujano. El rendimiento de toda la arquitectura ha sido testeado por medio de experimentos in vitro.

Palabras Clave: asistente robótico; interfaz multimodal; movimientos automáticos; modelado quirúrgico.

1 INTRODUCCIÓN

La Cirugía Mínimamente Invasiva (CMI) se ha convertido en una de las técnicas quirúrgicas más importantes de los últimos años gracias a su capacidad para reducir la dimensión de las incisiones, acortar el tiempo de convalecencia posoperatoria y otras complicaciones. Estas mejoras tienen consecuencias tanto sociales como económicas, por ejemplo al minimizar el tiempo que el paciente debe permanecer en el hospital.

En este sentido, una de las tendencias actuales de la CMI consiste en la Asistencia Manual en Cirugía Laparoscópica (HALS según el acrónimo anglosajón). Esta técnica permite al cirujano insertar una mano a través de una pequeña incisión mediante un guante presurizado. Así, el cirujano puede usar su mano no dominante de forma intra-abdominal mientras maneja un instrumento laparoscópico estándar con la otra mano. Con esta técnica el cirujano gana control sobre la operación, sensación de profundidad y realimentación táctil en comparación con una técnica laparoscópica convencional, a la par que se sigue aplicando CMI. Esta estrategia ha sido probada como una opción válida en cierto tipo de intervenciones, por ejemplo la cirugía de cáncer colo-rectal [17] donde los pacientes presentaron una pérdida de sangre, tiempo de intervención y de estancia en la unidad de cuidados intensivos similares a las intervenciones CMI. También se han obtenido resultados parecidos para intervenciones de cáncer de colon [18].

Sin embargo, la metodología HALS sólo permite al cirujano manipular una herramienta laparoscópica, así que resulta necesaria asistencia adicional para manejar la cámara o cualquier otro instrumental quirúrgico. Esta técnica también requiere que el cirujano permanezca en contacto directo con el paciente. Sistemas convencionales, como ZEUS de Computer Motion y DaVinci de Intuitive Surgical, no están diseñados para ofrecer tales requisitos. Por lo tanto, la sustitución de un asistente humano necesita de un sistema robótico que sea capaz de tomar decisiones de forma autónoma así como de comunicarse a través de una Interfaz Persona-Máquina (HMI según el acrónimo anglosajón) para una colaboración más natural.

La automatización de maniobras quirúrgicas mediante el uso del concepto de robot cooperativo permite al cirujano concentrarse en los procedimientos quirúrgicos mientras el robot asistente lo ayuda de forma colaborativa. Las técnicas de Guiado Visual se perfilan como la metodología más común para diseñar tareas automáticas, y se emplea en aplicaciones quirúrgicas como los movimientos del endoscopio en cirugía cardiaca [5]. Otros trabajos basados en la automatización de maniobras quirúrgicas se dedican a asistir de forma activa al cirujano durante la intervención, por ejemplo: desarrollando procedimientos de hilado y anudado [12]; ofreciendo una conversión automática de navegación laparoscópica a movimientos de cirugía abierta [14]; o navegando en intervenciones de colecistectomía sin información preoperatoria [21].

Una interacción natural y colaborativa entre el cirujano y el robot asistente requiere un HMI

avanzado capaz de imitar la comunicación con un humano durante la intervención [22],[10]. En este sentido, algunos estudios se centran en localizar los brazos del cirujano [4], mientras que otros se dedican al seguimiento de las herramientas del cirujano [11] o incluso la detección de gestos faciales [16]. Sin embargo, estos HMIs añaden nuevas tareas no naturales para el cirujano. Por este motivo, una nueva generación de interfaces debe dotar al robot de conocimiento sobre el protocolo quirúrgico [13],[20], de manera que el estado actual de la intervención pueda ser conocido y el robot pueda reaccionar de forma autónoma para ayudar al cirujano sin emplear comandos directos.

Este artículo propone una asistencia robótica diseñada para un escenario HALS, con capacidad para automatizar las maniobras quirúrgicas y asistir de forma colaborativa gracias a un HMI mejorado. De este modo, en primer lugar la sección 2 introducirá el escenario HALS para un procedimiento de sutura, así como la arquitectura de control propuesta en este trabajo para resolver la asistencia automática al cirujano. El principal elemento de esta arquitectura consiste en el asistente de estados, el cual se ocupa de seguir cada paso del procedimiento de sutura y será explicado en la sección 3. La evolución de dicho asistente se produce siempre que el robot finalice una tarea o el reconocedor de los movimientos de la herramienta del cirujano detecte que se ha realizado un gesto, todo lo cual será abordado en la sección 4. La implantación y experimentos que validan la metodología se exponen en la sección 5. Finalmente, la sección 6 presentará algunas conclusiones y líneas de investigación futuras.

2 ESCENARIO QUIRÚRGICO Y ARQUITECTURA DE CONTROL

El desarrollo de la asistencia robótica previamente establecida requiere de una descripción inicial de su entorno y las restricciones que limitarán sus movimientos. Por lo tanto, en primer lugar esta sección presenta el escenario HALS así como el método de sutura seleccionado para la automatización de una tarea quirúrgica.

Por otro lado, el análisis de un protocolo quirúrgico debe tener en cuenta la coordinación entre todas las herramientas quirúrgicas involucradas en la intervención. El conocimiento de dicha coordinación permite distribuir las tareas consideradas entre el cirujano y el asistente robot. De este modo, esta sección muestra en segundo lugar la arquitectura de control propuesta en este trabajo para ayudar al cirujano sin asistencia humana adicional.

2.1 ESCENARIO HALS Y EL PROTOCOLO QUIRÚRGICO

La Figura 1 muestra un escenario HALS que considera la asistencia del robot al cirujano. En este sentido, el robot interacciona con el cuerpo del paciente y el cirujano en un espacio restringido dentro de la Cavidad Abdominal. Una de las manos del cirujano se inserta a través de una incisión especial para manipular de forma directa la estructura anatómica de interés. Por ello, únicamente una mano permanece libre para usar otro instrumento, el cual se ha denotado como Herramienta del Cirujano. Por lo tanto, el robot asistente debería ser capaz manipular al menos la Cámara laparoscópica y una herramienta adicional etiquetada como Herramienta del Robot. Esta limitación de libertad de movimientos para el cirujano puede ser particularmente relevante en ciertos procedimientos quirúrgicos que requieren el uso de instrumental adicional.

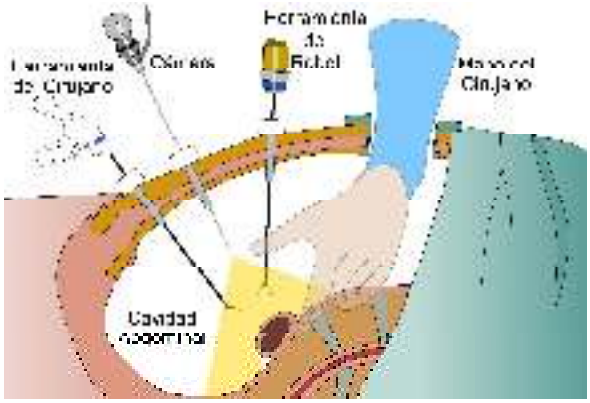


Figura 1: Problema de navegación para un robot asistente en un escenario HALS.

En concreto, la sutura y anudado tradicional consisten en un proceso que involucra a dos herramientas laparoscópicas, el cual puede ser realizado a través de diversas técnicas. Este trabajo se centra en el método de Rosser porque separa las tareas de ambas herramientas en dos: la principal y la de apoyo [8]. La primera realiza la sutura y anudado sobre el tejido, mientras que la herramienta de apoyo se emplea para asistir el proceso sujetando la aguja o tensando el hilo. Así pues, un robot quirúrgico puede facilitar la asistencia automática mediante el manejo de la herramienta de apoyo.

2.2 ARQUITECTURA DE CONTROL

La descripción anterior del procedimiento de sutura y anudado para un escenario HALS establece las tres principales características para que pueda realizarse mediante un robot asistente:

- La navegación de la herramienta de apoyo con múltiples restricciones al movimiento.

- La detección de los gestos quirúrgicos realizados por el cirujano.
- La monitorización de la evolución de la sutura para conocer el estado actual de la intervención.

De esta manera, la Figura 2 presenta la arquitectura de control propuesta para asistir un procedimiento de sutura de forma semi-autónoma. El sistema global, denominado *Asistente de Estados*, consiste en tres elementos que resuelven cada una de los problemas enumerados previamente. El *Asistente de Sutura* modela el flujo de trabajo del procedimiento de sutura a través de un diagrama de estados que será detallado en la sección 3. Dicha evolución puede ser modificada mediante Comandos de Voz del cirujano procesados por el dispositivo *Decodificador de Voz*. Dependiendo de la tarea actual, el diagrama de estados se ocupa de mandar la detección del gesto quirúrgico actual, o bien la actuación del robot asistente mediante un movimiento automático. Estas tareas se realizan mediante dos posibles *Manejadores de Estados*: el *Reconocedor de Gestos* o la *Interacción Auto-Guiada*. Ambos se comunican con el asistente de sutura enviando una activación de Transición y recibiendo el Estado Actual del procedimiento de sutura. El reconocedor de gestos necesita la Localización de las Herramientas medidas por un sensor para el *Tracking de las Herramientas* tanto del cirujano como del robot. Por otro lado, la interacción auto-guiada recibe las Fuerzas/Pares ejercidos y la localización actual del robot para calcular una Trayectoria Planificada adecuada. La explicación general de estas entradas así como las metodologías propuestas para resolver estos manejadores de estados se expondrán en la sección 4.



Figura 2: Arquitectura de control para la asistencia automática del cirujano.

3 EL ASISTENTE DE LA SUTURA

La misión principal del asistente de sutura consiste en seguir dicho procedimiento en línea. En concreto, una sutura realizada mediante el método de Rosser puede dividirse en cuatro pasos denominados maniobras. Cada maniobra se subdivide a su vez en varios movimientos elementales, también llamados gestos. Este trabajo ha considerado dos simplificaciones para adaptar la técnica original de Rosser a una asistencia automatizada:

i) Todos los gestos consecutivos realizados por la misma mano se agrupan en un solo gesto.
ii) Sólo una mano puede realizar un gesto al mismo tiempo y de forma estrictamente secuencial (no hay posibilidad de concurrencia).

La descripción del procedimiento de sutura comienza por la *Preparación de la Aguja*. Esta maniobra se centra en insertar la aguja en la cavidad abdominal, de manera que el cirujano pueda localizarla en el centro de la pantalla. En segundo lugar, la maniobra denominada *Punción de la Aguja* hilvana el tejido con la aguja quirúrgica. A continuación, el *Anudado* consiste en tres nudos que deben asegurar y cerrar el tejido, por lo que esta maniobra se repite tres veces. Por último, el *Corte de Hilo* retira el hilo extra y finaliza el procedimiento de sutura.

La Tabla 1 engloba la secuencia de gestos adaptada para una asistencia automatizada incluyendo la aplicación de las simplificaciones previamente comentadas. Cada gesto ha sido etiquetado como S_k dependiendo de su posición en la secuencia y se incluye dentro de su maniobra correspondiente. A su vez, esta tabla también muestra qué sujeto (cirujano o robot) realiza el gesto correspondiente.

Tabla 1: Pasos de Sutura con Asistencia Robótica.

Maniobra	Herr.*	Gesto	Descripción
Preparación de la Aguja	R	S_1^R	Insertar aguja y centrar
	C	S_2^C	Coger aguja sobre el tejido
Punción de la Aguja	R	S_3^R	Sostener tejido
	C	S_4^C	Punzar tejido, extraer aguja y tensar hilo
	R	S_5^R	Liberar tejido y abrir pinza
	C	S_6^C	Poner aguja en pinza del robot
Anudado	R	S_7^R	Posicionar aguja para nudo
	C	S_8^C	Rotar alrededor del hilo
	R	S_9^R	Tirar del hilo y cerrar nudo
Corte de Hilo	R	S_{10}^R	Acercar aguja a la sutura
	C	S_{11}^C	Tensar los 2 extremos del hilo
	R	S_{12}^R	Sostener bajo pinza cirujano
	C	S_{13}^C	Cambiar herramienta y cortar
	R	S_{14}^R	Extraer hilo y aguja

*Herramientas: (R) Robot y (C) Cirujano.

La secuencia de gestos descrita en la Tabla 1 puede ser implementada en el Asistente de Sutura como el diagrama de estados propuesto en la Figura 3. Cada estado del diagrama coincide con su gesto de la tabla, y cada marco representa una maniobra con su secuencia de estados. Todos los estados están conectados mediante transiciones T_k que permiten la evolución al siguiente estado. Los estados del robot S_k^R en color rojo denotan un gesto realizado por la herramienta de apoyo del robot. Del mismo modo, los estados del cirujano etiquetados como S_k^S en color verde identifican los gestos de la herramienta principal del cirujano. Los estados del robot planifican una trayectoria específica basada en un objetivo de fuerzas o localización, mientras que los estados del cirujano reconocen el gesto actual realizado por la herramienta del cirujano. En particular, el diagrama de estados siempre comienza desde el estado inicial S_0, el cual representa el comienzo del procedimiento de sutura.

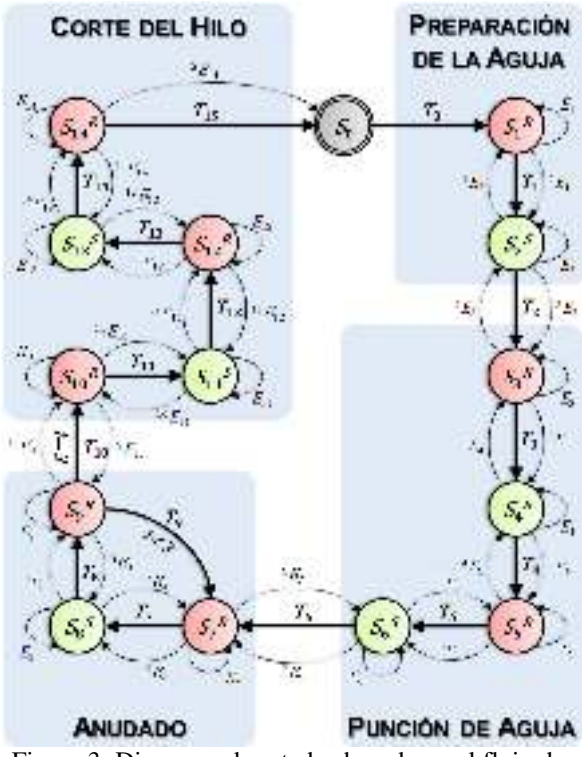


Figura 3: Diagrama de estados basado en el flujo de trabajo del procedimiento de sutura.

El flujo normal de trabajo del diagrama que sigue el procedimiento de sutura se representa mediante flechas sólidas etiquetadas por las transiciones T_k. Como caso particular, la maniobra de anudado debe ser completada tres veces. Por lo tanto, las transiciones T_9 y T_{10} del estado S_9^R están condicionadas por el iterador i, que indica el número de repeticiones que se han realizado.

El diagrama de estados de la Figura 3 también muestra transiciones adicionales entre los estados, representadas por flechas punteadas y denotadas como E_k o ${}^{k+1}E_k$. La notación depende de la fuente y el destino de la transición, en el sentido de que E_k vuelve al mismo estado S_k y ${}^{k+1}E_k$ comienza desde el estado S_k hasta el estado S_{k+1}. Estas transiciones especiales se incluyen para ofrecer al cirujano una forma de modificar el flujo normal del procedimiento de sutura. Tales situaciones consideran un error al reconocer un gesto, eventos inesperados durante el protocolo quirúrgico o incluso un comportamiento anómalo del robot asistente. Para este propósito, las transiciones relativas a un estado específico S_k se activan a través de su respectivo comando de voz para cada transición E_k, ${}^{k-1}E_k$ o ${}^{k+1}E_k$.

4 MANEJADORES DE ESTADOS

Los estados del Asistente de Sutura requieren de la actuación de dos sistemas que disparen la evolución del diagrama de estados a su siguiente estado. En primer lugar, la sección 4.1 describirá la interacción auto-guiada propuesta para la automatización de las tareas de la herramienta del robot. En segundo lugar, la sección 4.2 se centrará en el reconocedor de gestos desarrollado para analizar el movimiento de la herramienta del cirujano para detectar el gesto actual.

4.1 EL AUTO-GUIADO INTERACTIVO

El procedimiento de sutura expuesto en la Tabla 1 determina las principales tareas que el robot debe ser capaz de realizar:

- Mover el extremo de la herramienta del robot hacia una localización objetivo.
- Aplicar una fuerza sobre un tejido específico del paciente.
- Tensar el hilo de sutura.

De esta forma, la Interacción Auto-Guiada ha sido diseñada como se muestra en la Figura 4 a fin de gestionar estas tareas durante la evolución de cualquier estado del robot S_k^R del Asistente de Sutura. La planificación de trayectorias se puede calcular en función de un vector de localización objetivo $\mathbf{r_k}$ o la aplicación de una fuerza vectorial $\mathbf{F_k}$. Esta decisión la toma el elemento denominado *Selector de Objetivo* dependiendo del estado actual del robot, mientras que las salidas se obtienen con ayuda de los vectores de localización de las herramientas $\mathbf{r_t}$. La última entrada del robot es el vector de localización controlada $\mathbf{r_c}$, que consiste en la contribución entre la localización planificada $\mathbf{r_p}$ y el desplazamiento incremental $\mathbf{\Delta x_f}$. El vector $\mathbf{r_p}$ se calcula por medio del *Planificador Auto-guiado* y

representa la siguiente localización de la trayectoria, mientras que el vector $\Delta\mathbf{x_f}$ requiere de la actuación del bucle de control de fuerzas mostrado en la parte inferior de la Figura 4.

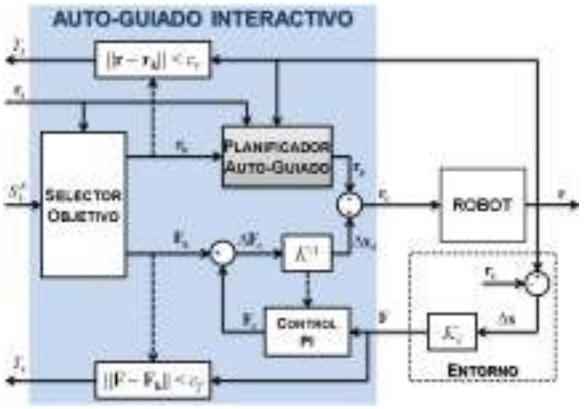


Figura 4: Sistema de Interacción Auto-Guiada para actuaciones automáticas.

La transición T_k de S_k^R se disparará una vez que se cumplan las siguientes condiciones:

$$\begin{aligned} \left\| \mathbf{r} - \mathbf{r_k} \right\| &< \varepsilon_r \\ \left\| \mathbf{F} - \mathbf{F_k} \right\| &< \varepsilon_f \end{aligned} \quad (1)$$

Siendo ε_r y ε_f los umbrales considerados para alcanzar tanto la localización objetivo como la aplicación de una fuerza determinada, respectivamente.

En primer lugar, el bucle de fuerzas se basa en un controlador de fuerzas acomodativo con realimentación PI que recibe el vector de fuerzas **F** medido por el robot y devuelve el vector de fuerza controlada $\mathbf{F_c}$. A continuación esta señal se resta a $\mathbf{F_k}$ y se transforma en $\Delta\mathbf{x_f}$ por medio de una ganancia K que modela la rigidez relativa robot-entorno con (2):

$$\Delta\mathbf{x_f} = K^{-1}\left(\mathbf{F_k} - \mathbf{F_c}\right) \quad (2)$$

Así, en el caso de una fuerza $\mathbf{F_k}$ no nula el robot se moverá en la dirección de $\Delta\mathbf{x_f}$ hasta que se aplique la fuerza deseada. La rigidez K puede variar dependiendo del estado S_k^R (por ejemplo, en función de si se sujeta tejido del paciente o se tensa el hilo de sutura). Por otro lado, el *Entorno* ha sido modelado con una K_λ que representa su rigidez real, y $\mathbf{r_e}$ es la posición donde el robot contacta con la superficie del entorno aplicando una fuerza nula.

En segundo lugar, el elemento Planificador Auto-guiado de la Figura 4 se centra en encontrar un camino mediante el algoritmo de Campos Potenciales Artificiales (CPA). Aunque este método por sí sólo calcularía una trayectoria para la herramienta del robot, se le ha añadido un sistema corrector de velocidad que modifica la velocidad del robot gracias a la predicción de una futura colisión en caso de que se prevea el cruce de un obstáculo a través de la trayectoria planificada por el CPA [2].

El CPA asocia un campo virtual de repulsión U_m^{rep} a cada uno de los m obstáculos y un campo virtual de atracción U^{att} a la localización del objetivo. Estos campos general fuerzas virtuales de atracción $\mathbf{F^{att}}$ y repulsión $\mathbf{F_m^{rep}}$ sobre la herramienta del robot, respectivamente. Así, el vector fuerza requerido para mover el robot a la siguiente localización planificada $\mathbf{r_p}$ es:

$$\mathbf{F_p} = -\nabla U^{att} - \sum_m \nabla U_m^{rep} = \mathbf{F^{att}} + \sum_m \mathbf{F_m^{rep}} \quad (3)$$

Los obstáculos considerados en este trabajo consisten en la herramienta del cirujano y el tejido interno de la cavidad abdominal, cuya superficie se representa mediante una malla de m-1 vértices generados a través de una técnica de SLAM monocular con relocalización para secuencias laparoscópicas [9]. Por otro lado, la expresión de $\mathbf{F^{att}}$ se ha elegido para que evolucione de cuadrática con la distancia hacia la localización objetivo mientras que cada $\mathbf{F_m^{rep}}$ se genera por el campo cilíndrico asociado a cada obstáculo, así pues:

$$\begin{aligned} \mathbf{F^{att}} &= 2K^{att}\Delta\mathbf{r_k} \\ \mathbf{F_m^{rep}} &= \begin{cases} K^{rep}\left(\frac{1}{\rho_m} - \frac{1}{\rho_m^0}\right)\left(\frac{\Delta r_k \boldsymbol{\rho}_\mathbf{m}}{\rho_m^2} + \frac{\mathbf{n_m}}{2}\right) & \rho_m \le \rho_m^0 \\ 0 & \rho_m > \rho_m^0 \end{cases} \end{aligned} \quad (4)$$

Las ganancias K^{att}, K^{rep} de (4) representan la relevancia de esa fuerza virtual respecto a las otras. El parámetro ρ_m define la distancia mínima entre la herramienta del robot y ese m-ésimo obstáculo, mientras que $\boldsymbol{\rho}_\mathbf{m}$ es el vector unitario con la misma dirección y ρ_m^0 es la distancia máxima a la que U_m^{rep} puede afectar a la trayectoria del robot. La distancia entre el extremo de la herramienta del robot y la localización objetivo se denota como Δr_k, y modifica la contribución de la fuerza de repulsión en las inmediaciones de la localización objetivo. Además, F_m^{rep} incluye la contribución de un vector unitario $\mathbf{n_m}$ que previene que la trayectoria encuentre posiciones de mínimos locales [6].

4.2 EL RECONOCEDOR DE GESTOS

Este elemento del manejador de estados interpreta los gestos realizados por el cirujano. Cada uno de estos gestos corresponde a un estado del cirujano S_k^S, tal y como se describió en la sección 3, y consiste en una secuencia específica de interacciones que ocurren

entre las herramientas quirúrgicas. Tales interacciones se consideran acciones básicas y se definen a través de características observables, las cuales incluyen información sobre el posicionamiento del instrumental y su cinemática [7]. Cada gesto se modela mediante un Modelo Oculto de Markov (HMM en inglés) que se describe por un diagrama estocástico con un número finito de estados, donde cada estado representa una acción básica. La topología de la red usada en este trabajo para los HMMs se basa en un Modelo de Bakis modificado, que en cada estado permite cierto grado de recorrido de la red hacia atrás [19].

El reconocimiento del gesto del cirujano actual requiere su comparación con todos los HMMs $\Gamma=[\lambda_1\ \lambda_2\ \ldots\ \lambda_k]$ de cada gesto S_k^S, los cuales se incluyen dentro de la Librería de Gestos mostrada en la Figura 5. En este esquema se considera tanto el Modo de Entrenamiento de los HMMs como el Modo de Reconocimiento del gesto actual. Estos procesos comparten el módulo de adquisición de la secuencia de características observables $E=[e_1\ e_2\ \ldots]$. En este sentido, el elemento de *Pre-procesado de Datos* recibe como entrada el vector localización de las herramientas $\mathbf{r_t}$ y transforma las velocidades de las puntas de ambas herramientas, el ángulo que forman y su distancia relativa en un vector característico C. A continuación, el módulo de *Codificado de Datos* le asigna una etiqueta o característica observable e dependiendo de la combinación de los componentes de C. El módulo *Selector de Secuencias* se centra en agrupar la secuencia de características observables E de cada tarea quirúrgica. Esta secuencia representa el gesto completo una vez que las velocidades de ambas herramientas V_t son nulas o el tiempo medio requerido para completar este gesto t_m haya sido sobrepasado.



Figura 5: Sistema de Reconocimiento de Gestos de los movimientos de las herramientas.

Por un lado, la Figura 5 muestra en línea discontinua el Modo de Entrenamiento, el cual procesa el entrenamiento de los HMMs Γ guardados en la Librería de Gestos. Para este propósito, el *Algoritmo de Baum-Welch* ajusta los parámetros de cada HMM λ_k con la ayuda de diversos ensayos de su gesto correspondiente [1].

Por otro lado, el Modo de Reconocimiento usa el *Algoritmo de Forward-Backward* para comparar la secuencia E con el conjunto de HMMs Γ y encontrar así el gesto más probable [19]. Si el gesto reconocido coincide con el estado actual S_k^S establecido por el módulo de Asistente a la Sutura, la transición T_k se dispara y el diagrama de estados descrito en la Figura 3 evoluciona a su siguiente estado.

5 IMPLANTACIÓN Y EXPERIMENTOS

El escenario quirúrgico de la Figura 6 muestra los elementos necesarios para la asistencia robótica en una intervención HALS con un paciente simulado. El cirujano sigue la intervención a través de una Pantalla que muestra la imagen de la Cámara laparoscópica. Este conjunto de óptica-cámara se encuentra sostenido por el Endoscopic Robotic Machine (ERM), un robot asistente desarrollado en la Universidad de Málaga cuya función consiste en mover el endoscopio a través de comandos de voz [15]. Las otras herramientas quirúrgicas se encuentran sostenidas por el cirujano y un manipulador PA10 de Mitsubishi, el cual realiza la navegación laparoscópica mediante un controlador de fuerzas de bajo nivel [3]. La localización de todas las herramientas quirúrgicas se registra a través de un dispositivo de Tracking 3D. A su vez, la otra Mano del Cirujano sostiene un órgano/tejido en el interior de la Cavidad Abdominal.



Figura 6: Configuración experimental para la validación del Asistente de Estados.

El experimento propuesto consiste en analizar la evolución de las principales maniobras de un procedimiento de sutura: la punción de la aguja y el anudado (ver sección 3). Así, la Figura 7 presenta los resultados obtenidos durante la secuencia de gestos para estas maniobras desde S_3 a S_9 (ver Tabla 1). Las

dos gráficas superiores grafican la evolución de la posición r y la fuerza ejercida F en el extremo de la herramienta del robot. Estos valores cambian siempre que el asistente de sutura evoluciona a un nuevo estado del robot, mandando su nueva posición r_k y/o fuerza F_k objetivos. En concreto, al final del gesto S_3^R el robot aplica una fuerza constante sobre el tejido para ayudar al cirujano a insertar la aguja en S_4^S, mientras que los estados S_7^R y S_9^R se centran en tirar del hilo para cerrar los tres anudados. Debajo, una tercera gráfica muestra el reconocimiento de los dos gestos más probables hechos por la herramienta del cirujano (valores mayores implican una probabilidad mayor). En este sentido, se puede deducir que los gestos correspondientes λ_k se reconocen con una alta fiabilidad debido a la gran diferencia con la segunda opción.

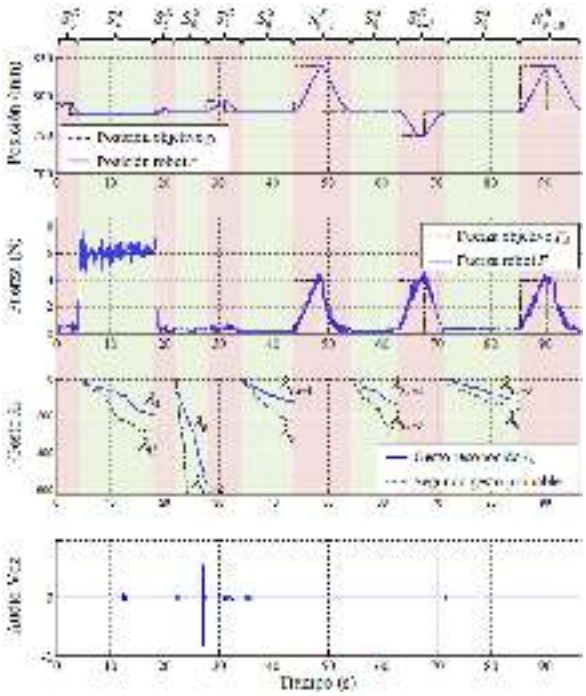


Figura 7: Secuencia de las maniobras de Punción de la Aguja y Anudado.

6 CONCLUSIONES

Este artículo ha presentado la arquitectura para una asistencia robótica en un escenario HALS. En concreto, se ha identificado que se requieren dos elementos que resultan necesarios para usar este tipo de robots con esta técnica quirúrgicas: un planificador auto-guiado flexible para calcular las trayectorias y la aplicación de fuerzas; y una comunicación natural cirujano-robot que implemente órdenes directas del cirujano así como reconocimiento de gestos en tiempo real. Los asistentes robóticos comerciales se centran normalmente en un interfaz teleoperado y, por consiguiente, resultan incapaces de manejar técnicas HALS debido al requisito de interacción conjunta cirujano-paciente.

Desde el punto de vista del guiado automático, los autores creen que una interacción más profunda entre los movimientos de la cámara y la herramienta del robot mejorarían tales tareas automatizadas. Además, otros procedimientos quirúrgicos podrían ser considerados para su automatización además de la sutura, como por ejemplo el corte, transporte o sujeción de tejidos. Del mismo modo, todas estas técnicas tienen protocolos de emergencia que pueden ser considerados para su automatización, por ejemplo en el caso de una rotura inesperada de una vena o de algún tejido

Referencias

[1] L.E. Baum, et al, "A maximization technique ocurring in the statistical analysis of probabilistic functions of markov chains", *The Annals of Mathematical Statistics,* 1970, vol. 41, no. 1, pp. 164-171.

[2] E. Bauzano, V.F. Muñoz , e I. Garcia-Morales, "A Multi-Behaviour Algorithm for Auto-Guided Movements in Surgeon Assistance", Int. J. Mechanics and Control, 2011, vol. 12, no. 1, pp. 35-41.

[3] E. Bauzano, V.F. Muñoz, I. Garcia-Morales, y B. Estebanez, "Three-Layer Control for Active Wrists in Robotized Laparoscopic Surgery", *IEEE Int. Conf. Intelligent Robots and Systems (IROS)*, 2009, pp. 2653-2658.

[4] A. Casals, M. Frigola, J. Amat, y E. Laporte, "Quasi Hands Free Interaction with a Robot for Online Task Correction". *Experimental Robotics IX, STAR 21*, 2006, pp. 175-184.

[5] H. Elhawary, y A. Popovic, "Robust feature tracking in the beating heart for a robotic-guided endoscope". *Int. J. Medical Robotics and Computer Assisted Surgery*, 7, pp. 459-468, 2011.

[6] S. Enxiu, C. Tao, H. Changlin, S. Enxiu, y G. Junjie, "Study of the New Method for Improving Artificial Potential Field in Mobile Robot Obstacle Avoidance". *IEEE Int. Conf. Automation and Logistics*, 2007, pp. 282-286.

[7] B. Estebanez, P. Saz-Orozco, I. García-Morales y V. F. Muñoz,"Interfaz Multimodal para un Asistente Robótico Quirúrgico: Uso de Reconocimiento de Maniobras Quirúrgicas", *Revista Iberoamericana de Automática e*

Informática Industrial (RIAI), 2011, vol. 8, no. 2, pp.24-34.

[8] J.E. Fischer, y K.I. Bland, *Mastery on Surgery*, vol. 1. Lippincott Williams & Wilkins, 2007.

[9] O.G. Grasa, J. Civera, y J.M.M. Montiel, "EKF monocular SLAM with relocalization for laparoscopic sequences", *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011, pp. 4816-4821.

[10] B. Herman, A. Hassan Zahraee, J. Szewczyk, y G. Morel, "Ergonomic and Gesture Performance of Robotized Instruments for Laparoscopic Surgery". *IEEE Int. Conf. Intelligent Robots and Systems (IROS)*, 2011, pp. 1333-1338.

[11] J. Hsu y S. Payandeh, "Toward Tool Gesture and Motion Recognition on a Novel Minimally Invasive Surgery Robotic System", *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Orlando, Florida, 2006, pp. 631-636.

[12] A. Knoll, H. Mayer, C. Staub, y R. Bauernschmitt, "Selective automation and skill transfer in medical robotics: a demonstration on surgical knot-tying". Int. J. Medical Robotics and Computer Assisted Surgery, D.O.I. 10.1002/res.149, 2012.

[13] S.Y. Ko, W.J. Lee, y D.S. Kwon, "Intelligent Interaction based on a Surgery Task Model for a Surgical Assistant Robot: Awareness of Current Surgical Stages based on a Surgical Procedure Model". *Int. J. Control, Automation and Systems*, 2010, vol. 8, no. 4, pp. 782-792.

[14] M. Kranzfelder, A. Schneider, S. Gillen, y H. Feussner, "New technologies for information retrieval to achieve situational awareness and higher patient safety in the surgical operating room: the MRI institutional approach and review of the literature". *Surgical Endoscopy*, 25, pp. 696-705, 2011.

[15] V.F. Muñoz, I. Garcia-Morales, C. Perez-DelPulgar, J.M. Gomez-DeGabriel, J. Fernandez-Lozano, A. Garcia-Cerezo, C. Vara-Thorbeck, y R. Toscano, "Control movement scheme based on manipulability concept for a surgical robotic assistant", *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 245-250.

[16] A. Nishikawa, et al, "FAce MOUSe: A novel human-machine interface for controlling the position of a laparoscope", 2003, IEEE Trans. Robotics and Automation, vol. 19, no. 5, pp. 825-841.

[17] S.T. Orcutt, C.L. Marshall, C.J. Balentine, C. N. Robinson, D.A. Anaya, A. Artinyan, D.H. Berger, y D. Albo, "Hand-assisted laparoscopy leads to efficient colorectal cancer surgery". *J. Surgical Research*, vol. 177, no. 2, pp. e53-e58, Octubre 2012.

[18] R. Pendlimari, S.D. Holubar, J. Pattan-Arun, D.W. Larson, E.J. Dozois, J. H. Pemberton, y R.R. Cima, "Hand-assisted laparoscopic colon and rectal cancer surgery: Feasibility, short-term, and oncological outcomes". *Surgery*, vol. 148, no. 2, pp. 378-385, Agosto 2010.

[19] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications on Speech Recognition", *Proceedings of the IEEE,* 1989, vol. 77, no.2, pp. 257-286.

[20] C. Staub, S. Can, B. Jensen, A. Knoll, y S. Kohlbecher, "Human-Computer Interfaces for Interaction with Surgical Tools in Robotic Surgery", *4th IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics*, Roma, Italia, 2012, pp. 81-86.

[21] C. Staub, S. Can, A. Knoll, V. Nitsch, I. Karl, y B. Farber, "Implementation and evaluation of a gesture-based input method in robotic surgery". *IEEE Int. Workshop on Haptic Audio Visual Environments and Games (HAVE)*, 2011.

[22] K. Wasen, "Person-friendly Robot Interaction: Social, Phychological and Technological Impacts in Health Care Work". *IEEE Int. Workshop on Robots and Human Interactive Communication*, 2005, pp. 643-648.

INCORPORACIÓN DE UN SISTEMA DE MINI-ROBOTS A LA CIRUGÍA LAPAROSCÓPICA DE INCISIÓN ÚNICA

M. Cuevas-Rodríguez, I. Rivas-Blanco, E. Bauzano, J. Gómez-deGabriel y V.F. Muñoz
Severo Ochoa, 4, 29590, vfmm@uma.es

Resumen

Este artículo presenta un sistema robótico diseñado con el objetivo de solventar los problemas que aparecen en las técnicas de cirugía laparoscópica de incisión única. El sistema está compuesto por un mini-robot cámara, un mini-robot iluminación, y un robot pinza. Todos estos dispositivos son introducidos en el interior de la cavidad abdominal a través del puerto único y adheridos a la pared abdominal mediante interacción magnética mediante una serie de sujeciones colocadas en el exterior. Dichas sujeciones son manejadas por dos brazos robóticos gracias a unas piezas de agarre colocadas en el efector final de los mismos. El movimiento de estos brazos permite el guiado de los mini-robots a lo largo de la cavidad abdominal. El manipulador encargado de guiar los mini-robots cámara e iluminación es manejado a través de comandos de voz, mientras que el otro, el cual sostiene el robot pinza, es teleoperado mediante un dispositivo háptico. Se han llevado a cabo una serie de experimentos in-vitro con el objetivo de comparar las ventajas del uso de este nuevo sistema con los procedimientos tradicionales de cirugía laparoscópica.

Palabras Clave: Cirugía laparoscópica, mini-robots, SILS, incisión única.

1 INTRODUCCIÓN

Las técnicas de laparoscopia de incisión única o SILS (de su acrónimo en inglés Single Incision Laparoscopic Surgery) han sido desarrolladas recientemente con el objetivo de reducir el número de incisiones en los procedimientos de Cirujía Minimamente Invasiva (CMI). Esta técnica consiste en realizar una única incisión transumbilical, en lugar de las cuatro o cinco que conlleva el procedimiento tradicional. Con este objetivo, se han desarrollado nuevos tipos de puertos, como son el Tripot™, el SILS™ Port, el Uni-X™ Single Port System, y el Air-seal™ [10]. Estos dispositivos están formados por un disco flexible de 2-3 cm de diámetro el cual permite la introducción de múltiples herramientas a través de un solo trocar con varios canales (trocar multipuerto) [1]. Los beneficios de esta nueva generación de métodos incluyen mejoras en la recuperación del paciente, menor dolor y tiempo de post-operación y mejoras estéticas [8]. A pesar de todas estas ventajas este método presenta una serie de inconvenientes difíciles de solventar. La proximidad entre las herramientas y la cámara provoca una importante pérdida de triangulación. Además, la proximidad de las herramientas entre sí, debido a que todas comparten el mismo puerto, limita el rango de movimiento de dichas herramientas por parte del cirujano [2].

Numerosas investigaciones han buscado solución a estos problemas desarrollando sistemas de mini-robots, los cuales son introducidos en la cavidad abdominal a través del puerto de entrada único. El posicionamiento y fijación de estos robots a la pared abdominal se puede realizar mediante sutura [5], sostenidos por agujas, o por interacción magnética [6][4][12]. La principal ventaja del uso de imanes respecto a la sutura o las agujas es el fácil desplazamiento del robot a lo largo de la pared abdominal mediante el desplazamiento la sujeción magnética exterior. Investigadores de la Universidad de Texas han realizado con éxito dos intervenciones laparoscópicas de nefrectomía y apendicectomía utilizando un sistema de cámaras de agarre magnético [3]. Lehman y otros [7] proponen un robot en miniatura magnético formado por dos brazos flexibles cuyos efectores finales están formados por un cauterizador y fórceps. La eficacia de este robot ha sido demostrada en experimentos in-vivo, con un modelo porcino, en tres procedimientos diferentes (exploración abdominal, manipulación intraabdominal del intestino y colecistectomía). En Simi y otros [11], una plataforma de visión basada en robots activados magnéticamente ha sido también probada en animales. La fijación de los robots en la pared abdominal se lleva a cabo con un conjunto imanes permanentes los cuales pueden ser fácilmente desplazados con la mano. Un concepto similar es utilizado en [14], donde el dispositivo interno consiste en una estructura triangular actuada magnéticamente sobre la que se colocan un conjunto de unidades robóticas al mismo tiempo. En Terry y otros [13], se presenta un dispositivo interno que puede ser sostenido por un anillo rígido que se coloca en el puerto de entrada o por sujeción magnética, y

ha sido testeado con animales en un procedimiento de colecistectomía.

Este artículo presenta un sistema robótico para procedimientos de laparoscopia de una única incisión compuesto por un mini-robot cámara, un mini-robot iluminación, y un robot pinza, todos ellos sostenidos a la pared abdominal con imanes permanentes. A diferencia que los trabajos anteriores, los imanes externos están sujetos al efector final de dos brazos robóticos, de los cuales uno de ellos lleva a cabo el guiado de los mini-robots cámara e iluminación y otro el robot pinza. La estructura seguida en este artículo es la siguiente. En primer lugar se realizará una descripción del sistema y de los elementos que lo forman. A continuación, en la sección 3 se describirá el modo en que se ha llevado a cabo la integración y los experimentos in-vitro realizados con el objetivo de comparar las ventajas del uso de sistemas robóticos frente a los procedimientos convencionales. Finalmente, en la sección 4 se presentarán las conclusiones de este trabajo.

2 SISTEMA ROBÓTICO

2.1 DESCRIPCIÓN DEL SISTEMA

En la Figura 1 se muestra la arquitectura global de control del sistema robótico, el cual está compuesto por un mini-robot cámara, encargado de ofrecer la imagen del interior de la cavidad abdominal, un mini-robot iluminación, que aporta luz a la escena y un robot pinza, que será utilizado como pinza auxiliar por parte del asistente.

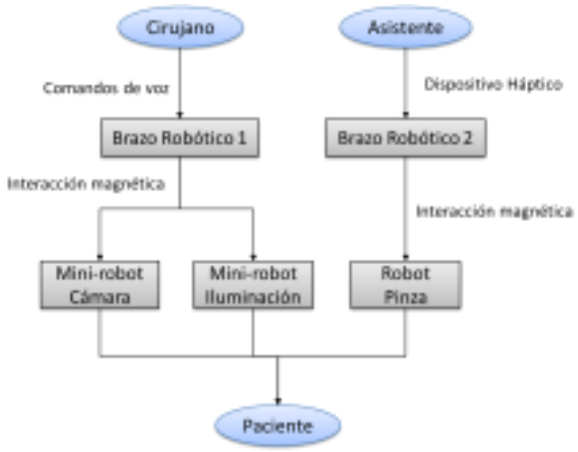


Figura 1: Arquitectura global del sistema

Estos dispositivos están formados por imanes permanentes y se encuentran dentro de la cavidad abdominal sujetos a la pared gracias a una serie de sujeciones magnéticas, que se hallan en el exterior. Estas sujeciones se manejan con dos brazos robóticos, uno de los cuales maneja tanto el mini-robot cámara como el iluminación y el otro el robot pinza.

El mini-robot cámara es controlado mediante comandos de voz, lo cual permite al cirujano guiar la cámara a través de la pared abdominal al mismo tiempo que controla su giro y orientación. El mini-robot iluminación proporciona la luz necesaria en el área de intervención, el cirujano será el encargado de indicarle al manipulador el que mini-robot que desea desplazar y este realizará el cambio automáticamente. Finalmente, el robot pinza es teleoperado mediante un dispositivo háptico el cual controla los movimientos del brazo robótico que maneja dicho robot.

En la Figura 2 se muestra la distribución de los diferentes elementos del sistema en el interior de la cavidad abdominal. Todos los dispositivos internos son inalámbricos y se introducen en el interior de la cavidad a través del puerto de entrada único. Una de las ventajas de este sistema es que el mini-robot cámara, una vez dentro puede alejarse del trocar por donde se introducirán las herramientas, solventando así el problema de pérdida de triangulación debido a la proximidad de la cámara con las herramientas que maneja el cirujano. Además, debido a que la cámara puede moverse a lo largo de toda la cavidad abdominal, se podrán obtener mayores ángulos de la imagen que con las cámaras de laparoscopia tradicionales, ofreciendo así al cirujano una visión más amplia de la cavidad abdominal. Por otro lado, al tratarse de robots inalámbricos, se reduce el número de instrumentos que comparten el multipuerto de entrada, resolviendo así el problema de la limitación de movimientos de las herramientas por parte del cirujano. Del mismo modo, en los procedimientos que requieran el uso de tres herramientas diferentes al mismo tiempo, como por ejemplo la funduplicatura, el robot pinza solventa el problema de tener que realizar una incisión adicional en el paciente. A continuación se describirán los diferentes elementos del sistema ya mencionados.

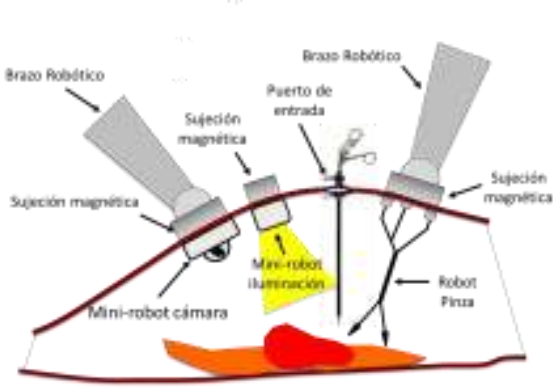


Figura 2. Sistema Robótico

2.2 MINI-ROBOT CÁMARA E ILUMINACIÓN

El sistema exterior encargado del desplazamiento del mini-robot cámara está compuesto por dos elementos: la sujeción magnética, y el agarre situado en la muñeca del brazo robótico el cual se encaja sobre la sujeción para poder desplazarla (Figura 3). El acoplamiento entre los dos elementos permite, como se ha mencionado anteriormente, desplazar la sujeción y por ende el mini-robot al lugar deseado. Una vez allí, el brazo puede desengancharse de dicha sujeción. De este modo, tanto el mini-robot cámara como el iluminación pueden ser manejados por el mismo manipulador. Al mismo tiempo este sistema permite tanto un guiado automático de los mini-robots a través del manipulador como un guiado a mano por parte del cirujano.

El mini-robot cámara posee dos grados de libertad: rotación e inclinación. El movimiento de rotación se consigue rotando la articulación enésima del manipulador. De este modo, el imán interno del mini-robot cámara sigue al imán externo de la sujeción. La inclinación se lleva a cabo gracias a la actuación del motor lineal que se encuentra en la muñeca, el cual desplaza el imán A de la Figura 3. Este imán interacciona con el B el cual se traslada del mismo modo causando el movimiento de un imán cilíndrico (C) localizado dentro del mini-robot. Este último imán se encuentra conectado directamente a la cámara, y así se consigue el cambio de orientación en esta. Con este mecanismo el motor lineal podrá mover la cámara con una orientación de ± 42º.



Figura 3. Sistema mini-robot cámara

La Figura 4 muestra el primer prototipo del mini-robot cámara, de dimensiones: 12 cm de largo por 4 cm de ancho. Este dispositivo está compuesto por una cámara inalámbrica encargada de captar la imagen del interior de la cavidad abdominal y trasmitirla a un receptor ubicado en el exterior. Para iluminar la escena se dispone de seis LEDs alrededor de la cámara, cada uno de los cuales emite 6.8 candelas con una intensidad de 20 mA. En cuanto a las baterías el mini-robot contiene 9 pilas de 9 voltios cada una (250 mAh), lo cual ofrece al dispositivo una autonomía de 4 horas.



Figura 4. Mini-robot Cámara

A pesar de que el mini-robot cámara está provisto una serie de LEDs, estos no proporcionan la suficiente iluminación en la zona de operación como los dispositivos de laparoscopia tradicionales. Incluir un gran número de LEDs en este mini-robot implicaría el aumento del consumo de potencia y por tanto del número de baterías, lo cual provocaría el aumento considerable del tamaño del robot. Por estos motivos ha sido añadido al sistema un robot encargado exclusivamente de la iluminación. Así mismo, el hecho de tener la cámara e iluminación independientes permite cambiar la dirección de la luz según las preferencias del cirujano, evitando así el problema de las sombras que provocan la pérdida de visibilidad y percepción de la profundidad del escenario.

El primer prototipo de mini-robot iluminación se muestra en la Figura 5, cuyas dimensiones son 35 mm de diámetro y 20 mm de alto. Como se mencionó anteriormente, este mini-robot es guiado a través de la cavidad abdominal gracias a una sujeción magnética situada en el exterior de esta, dicha sujeción puede ser desplazada manualmente o a través del brazo robótico. Trabajos futuros incluyen el movimiento automático de este mini-robot con el objetivo de liberar al cirujano de esta tarea. Los dispositivos LEDs son del mismo tipo que los situados en el mini-robot cámara. Dos pilas de botón de 3 voltios son utilizadas para alimentar el dispositivo, ofreciendo al robot al menos 3 horas de autonomía, lo cual es suficiente para un típico procedimiento de mínima invasión como puede ser una colecistectomía.



Figura 5. Mini-robot Iluminación

2.3 ROBOT PINZA

El tercer componente del sistema es el robot pinza, cuyo prototipo tiene 8 cm de largo y se muestra en la Figura 6.

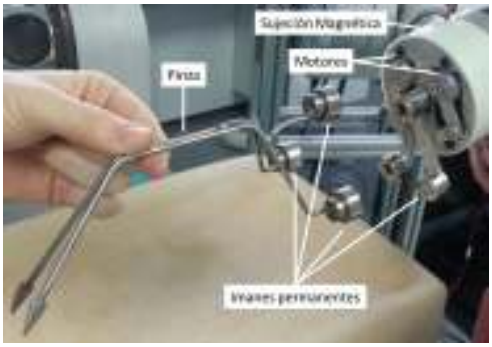


Figura 6. Robot Pinza

En este caso el dispositivo está formado únicamente por dos partes: la sujeción magnética, que se encuentra acoplada a la muñeca del manipulador y la pinza en sí. La sujeción magnética está compuesta por tres imanes permanentes: en el centro se sitúa un imán fijo y otros dos a ambos lados de este, estos últimos se desplazarán gracias a dos motores de rotación, cada uno de ellos conectados a un imán. La pinza, también formada por tres imanes permanentes, se colocará en el interior de la cavidad abdominal interaccionando magnéticamente con la sujeción. Sin más que accionar los motores se llevará a cabo la apertura y cierra de la pinza. Debido a que los motores se encuentran en el exterior, no será necesario introducir ningún cable dentro del abdomen. Como se ha mencionado, la sujeción está colocada en el efector final del brazo robótico, lo que permitirá teleoperar dicho brazo, haciendo uso de un dispositivo háptico de modo que sea posible controlar los movimientos de este. Debido a la flexibilidad de la cavidad abdominal la reorientación de la pinza puede llevarse a cabo presionando con diferentes ángulos la sujeción magnética que se encuentra en el exterior y así, el cambio de orientación de la sujeción provocará un cambio de orientación en la pinza.

3 INTEGRACION Y EXPERIMENTOS

Para la integración y comunicación de todos los elementos del sistema se ha utilizado el framework ROS (Robot Operating System). ROS se define como un "*meta-sistema operativo*" para robots, de código abierto, donde se implementan una serie de nodos que intercambian información entre ellos siguiendo una filosofía publicador/subscriptor [9]. Dichos nodos también pueden funcionar ofreciendo diversos servicios, de modo que estos puedan ser accesibles a través de la definición de una serie de clientes.

Cada uno de los elementos del sistema presentado en este trabajo tiene asociado un nodo de ROS lo cual facilita la comunicación entre todos ellos. Así mismo esta arquitectura distribuida permite colocar los nodos con más coste computacional en diferentes máquinas y conseguir así una gran eficiencia en el funcionamiento del sistema.

Se han llevado a cabo una serie de experimentos in-vitro para testear el sistema implementado, donde se utiliza como modelo para la intervención una vesícula simulada con un tumor debajo de esta, tal y como se muestra en el cuadro superior derecho de la Figura 7. El experimento consiste en realizar un corte en la vesícula biliar, retirarla y extraer el tumor del paciente. El objetivo de este experimento es comparar las ventajas del uso del sistema robótico con los procedimientos de laparoscopia tradicionales. En un procedimiento tradicional el cirujano realiza la intervención haciendo uso de dos herramientas diferentes mientras que el asistente es el encargado de sujetar la cámara laparoscópica y apartar la vesícula biliar. En este tipo de intervenciones es necesario realizar una incisión extra para poder introducir la herramienta que utilizará el asistente para sostener la vesícula, mientras que el resto de herramientas y la cámara se introducirán a través del multipuerto especial de las técnicas SILS.



Figura 7. Procedimiento quirúrgico con el sistema robótico

En el sistema robótico presentado, la cámara laparoscópica y la pinza auxiliar son sustituidas por los nuevos elementos del sistema, los mini-robots cámara e iluminación y el robot pinza, que, como ha sido explicado anteriormente, no necesitan de ninguna incisión extra para ser introducidos y guiados por la cavidad abdominal (ver Figura 7). En este segundo entorno la pinza auxiliar (robot pinza magnético) es sostenido por un manipulador el cual se encuentra teleoperado por el asistente mediante el uso de un dispositivo háptico (colocado fuera del escenario mostrado en la Figura 7). Al mismo tiempo, durante la intervención el cirujano manejará el segundo manipulador, indicándole, a través de comandos de voz dónde desea que se coloquen los

mini-robots cámara e iluminación. El interior de la cavidad abdominal se muestra en la Figura 8, donde se puede observar cómo se distribuyen los diferentes mini-robots por todo el espacio.

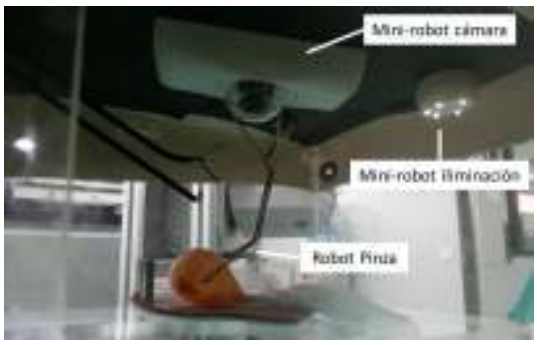


Figura 8. Interior cavidad abdominal experimento in-vitro

El procedimiento quirúrgico ha sido realizado por cinco cirujanos con poca experiencia. La Tabla 1 muestra el tiempo que ha tardado en ejecutarse cada tarea en ambos casos, por un lado realizando el procedimiento tradicional y por otro haciendo uso del nuevo sistema robótico presentado en este artículo. En el caso del uso del sistema robótico el cirujano tuvo que realizar varias pruebas para familiarizarse con las nuevas herramientas, una vez entrenado, este llevó a cabo la intervención y se contabilizaron los tiempos. El experimento comienza con la retirada de la vesícula biliar y finaliza con la extracción del tumor.

Tabla 1: Comparación de un entorno tradicional con un entorno robótico

Cirujano	**Entorno tradicional**	**Entorno robótico**
Cirujano 1	2 min 36 seg	2 min 55 seg
Cirujano 2	3 min 18 seg	2 min 59 seg
Cirujano 3	2 min 47 seg	2 min 41 seg
Cirujano 4	3 min 07 seg	2 min 43 seg
Cirujano 5	2 min 55 seg	3 min 08 seg

Como se puede observar en la Tabla 1 el tiempo de ejecución en ambas intervenciones es muy similar. En el caso del sistema robótico el cirujano emplea menos tiempo en cortar y retirar el tumor debido a que el rango de movimiento dentro de la cavidad no está limitado por problemas de triangulación con la cámara de laparoscopia. Sin embargo, en este caso, el asistente emplea más tiempo en retirar la vesícula biliar debido a que no está familiarizado con el uso un dispositivo háptico y el manejo directo de la pinza es más intuitivo que el teleoperado. Por otro lado, todos los cirujanos han demostrado más destreza a la hora de realizar la intervención cuando la cámara de laparoscopia no se encuentra compartiendo el puerto SILS. Además, el uso del robot pinza, gracias a su agarre magnético, ha evitado la necesidad de la creación de una incisión extra en el paciente y ha permitido que el asistente no tengo que sostener dos instrumentales a la vez (cámara y pinza) facilitando de este modo el manejo de dicha pinza.

En este experimento se ha estudiado un caso clínico concreto, la extracción de un tumor que se encuentra situado bajo la vesícula, debido a la necesidad del uso de todos los elementos del sistema en este tipo de intervenciones: cámara, iluminación extra y pinza auxiliar. Futuros experimentos se basarán en la realización de otro tipo de operaciones con el objeto de decidir cuál es la más adecuada para el uso de este nuevo sistema robótico.

4 CONCLUSIONES

El sistema robótico propuesto en este trabajo representa una solución al principal inconveniente del uso de técnicas SILS. El problema de la pérdida de triangulación en el procedimiento tradicional debido a la proximidad de la cámara con las herramientas quirúrgicas es solventado gracias a que el mini-robot cámara puede alejarse del puerto único de entrada. Así mismo, al no existir la necesidad de introducir la cámara laparoscópica a través del mismo trocar el rango de movimiento de las herramientas por parte del cirujano aumenta y estos pueden realizar las intervenciones de un modo más cómodo. Además el uso de una pinza robótica, adherida a la pared abdominal por interacción magnética, evita el hecho de tener que realizar una incisión extra en el paciente, en el caso de que una pinza auxiliar sea requerida para llevar a cabo la operación. Este robot será manejado por el asistente de forma teleoperada, así la zona de intervención donde se encuentran paciente y cirujano se encontrará más despejada. El sistema ha sido probado con una serie de experimentos in-vitro con el objetivo de comparar las ventajas del uso de este sistema con los procedimientos de laparoscopia tradicionales.

Agradecimientos

Este trabajo se ha realizado gracias a la financiación del proyecto nacional DPI2010-21126-C03-01.

Referencias

[1] S. Agrawal, A. Shaw, and Y. Soon, “Single-port laparoscopic totally extraperitoneal inguinal hernia repair with the TriPort system: initial experience,” Surgical Endoscopy, Vol. 24, pp. 952-956, 2010.

[2] P. Bucher, F. Pugin, and P. Morel, "Single port access lapaoscopic right hemiclectomy," International Journal Colorectal Dis., vol. 23, pp. 1013-1016, 2010.

[3] J. A. Cadeddu, R. Fernandez, M. Desai, R. Bergs, C. Tracy, S. J. Tang,P. Rao, M. Desai, and D. Scott, "Novel magnetically guided intraabdominal camera to facilitate laparoendoscopic single-site surgery: Initial human experience," Surg. Endosc., vol. 23, pp. 1894–1899, 2009.

[4] C.A. Castro, S. Smith, A. Alqassis, T. Ketterl, Y. Sun, S. Ross, A. Rosemurgy, P.P. Savage, and R.D. Gitlin, "MARVEL: a wireless miniature anchored robotic videoscope for expedited laparoscopy", 2010 IEEE Internatinal Conference on Robotics and Automation RiverCentre, pp. 2926-2931, May 2012.

[5] T. Hu, P.K. Allen, N.J. Hogle, and D.L. Fowler, "Insertable surgical imaging device with pan, tilt, zoom and lighting", 2008 IEEE International Conference on Robotics and Automation, pp. 2948-2953, May 2008.

[6] T. Kawahara, T. Takaki, I. Ishii, and M. Okajima, "Development of a broad-view camera system for minimally invasive surgery", 2010 IEEE/RSJ Internatinal Conference on Intelligent Robots and Systems, pp. 2810-2815, October 2010.

[7] A.C. Lehman, J. Dumpert, N.A. Wood, L. Redden, A.Q. Visty, S. Farritor, B. Varnell, and D. Oleynikov, "Natural orifice cholecystectomy using a miniature robot," Surgical Endoscopy Journal, Vol. 23, pp. 260-266, 2009.

[8] S.R. Philipp, B.W. Miedema, and K. Thaler, "Single-incision laparoscopic cholecystectomy using conventional instruments: early experience in comparison with the gold standard," Journal of the American College of Surgeons, Vol. 209, No. 5, pp. 632-637, 2009

[9] M. Quigley, B. Grerkey, K. Conley, J. Faust, Tully Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng, "ROS: an open-source Robot Operating System", ICRA Workshop on Open Source Software, 2009.

[10] J.R. Romanelli, T.B. Roshek III, D.C. Lynn, and D.B. Earle, "Singleport laparoscopic cholecystectomy: initial experience," Surgical Endoscopy, Vol. 24, pp. 1374-1379, 2010.

[11] M. Simi, M. Silvestri, C. Cavallotti, M. Vatteroni, P. Valdastri, A. Menciassi, and P. Dario, "Magnetically Activated Stereoscopic Vision System for Laparoscopic Single-Site Surgery," in IEEE/ASME Transactions on Mechatronics, Vol. 18, No.3, pp. 1140-1151, June 2013

[12] Y. Sun, A. Anderson, C. Castro, B. Lin, R. Gitlin, S. Ross, and A. Rosemurgy, "Virtually transparent epidermal imagery for laparo-endoscopic single-site surgery", 33rd Annual International Conference of the IEEE EMBS, pp. 2107-2110, 2011.

[13] B.S. Terry, Z.C. Mills, J.A. Schoen, and M.E. Rentschler, "Single-Port-Access Surgery with a Novel Magnet Camera System", IEEE Transactions on Biomedical Engineering, Vol. 59, No. 4, April

[14] S. Tognarelli, M. Salerno, G. Tortora, C. Quaglia, P. Daria, and A. Menciassi, "An endoluminal robotic platform for minimally invasive surgery", The fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, pp. 7-12, June 2012.

CONTROL FUERZA-POSICIÓN DE UNA CÁMARA ROBÓTICA PARA TÉCNICAS DE CIRUGÍA DE PUERTO ÚNICO

I. Rivas-Blanco, E. Bauzano, M. Cuevas-Rodriguez,P. del Saz-Orozco, V.F. Muñoz
Departamento de Ingeniería de Sistemas y Automática
Universidad de Málaga, vfmm@uma.es

Resumen

En este artículo sepresenta el diseño y la implementación de un sistema robótico de visión para técnicas de cirugía de puerto único. El sistema está formado por una mini-cámara robótica inalámbrica y un brazo robótico externo para su guiado sobre la pared abdominal. El robot cámara está provisto de una serie de imanes, y la sujeción magnética externa se encuentra en el efector final del manipulador. De esta manera, la cámara se puede desplazar para obtener puntos de vista adicionales de la cavidad abdominal. En el artículo se describe el primer prototipo del robot cámara, que incluye un sistema de iluminación. Además, se presenta el desarrollo de un control híbrido fuerza-posición para el correcto desplazamiento del brazo robótico sobre la pared abdominal, incluyendo un control de orientación para mantener el manipulador perpendicular a la misma. Se ha considerado que la superficie de contacto sigue un modelo elástico, cuya constante de rigidez se estima mediante un algoritmo de mínimos cuadrados. Finalmente, se presenta un experimento in-vitro para validar el esquema de control propuesto.

Palabras Clave: asistente robótico, SILS, mini-robot, control híbrido fuerza-posición.

1 INTRODUCCIÓN

Actualmente la Cirugía Mínimamente Invasiva (CMI) constituye una técnica ampliamente aceptada en todo el mundo, tanto por los cirujanos como por los pacientes, que se benefician de sus ventajas estéticas derivadas de las pequeñas incisiones a través de las cuales se realiza la operación. En comparación con las técnicas tradicionales de cirugía abierta, las técnicas laparoscópicas tienen la ventaja de reducir las pérdidas de sangre intraoperatorias y postoperatorias, así como el riesgo de infección y el tiempo de recuperación [1]. El siguiente paso natural para la cirugía laparoscópica consiste en reducir el número de incisiones a una. En este sentido, las técnicas de puerto único constituyen una alternativa en la que todas las herramientas se introducen a través de la misma incisión. Los beneficios de esta nueva técnica incluyen menor dolor, menor tiempo de recuperación, mejoras estéticas y mayor grado de satisfacción de los pacientes [2]. A pesar de estas ventajas, este tipo de técnicas presentan dos principales inconvenientes: la cercanía de los instrumentos quirúrgicos y la cámara, dado que comparten el mismo puerto de entrada, provoca una pérdida de triangulación entre la cámara y las herramientas, y limita su rango de movimiento en el exterior del paciente [3]. Este hecho hace que sea necesaria la utilización de instrumentos laparoscópicos flexibles o curvos especiales, de manera que se pueda abarcar un mayor espacio de trabajo sin chocar las herramientas. El uso de estos nuevos instrumentos requiere que los cirujanos se sometan a largos periodos de entrenamiento para habituarse a su uso, incluso para aquellos cirujanos con un alto grado de experiencia en cirugía laparoscópica tradicional [4]. Los endoscopios flexibles están compuestos de un mango flexible y una punta con capacidad de doblarse, cuyo ángulo de desviación se controla mediante un sistema de ruedas. El manejo de estos endoscopios requiere la combinación de varias acciones para realizar un determinado movimiento, lo que resulta muy complejo para los cirujanos [5]. En [6] se presenta una nueva generación de endoscopios flexibles teleoperados.

Una alternativa para resolver los inconvenientes de las técnicas de puerto único consiste en introducir mini-robots en el interior de la cavidad abdominal. De esta manera, se reduce el número de instrumentos que comparten el puerto único, y se evita la pérdida de grados de libertad (GDL) debido al efecto de fulcro. Además, al introducir la cámara en la cavidad abdominal, ésta se puede separar de las herramientas, evitando el problema de la pérdida de triangulación, y se puede proporcionar un mayor número de puntos de vista del abdomen. El posicionamiento y fijación de estos robots a la cavidad abdominal se puede realizar mediante sutura [7], mediante fijación con agujas [8]-[10], o mediante imanes externos, como en [11] o [12]. Actualmente existen un gran número de investigaciones centradas en el uso de imanes para el guiado y la fijación de dispositivos internos [13]-[18]. Lehman y otros. [19] proponen un mini-robot con dos brazos con capacidad de plegado, que se adhiere a la pared abdominal mediante interacción magnética. En Simi y otros. [20], se propone una plataforma de visión inalámbrica basada en un robot

actuado magnéticamente. En [21] se propone un concepto similar, donde el dispositivo interno consiste en una estructura triangular actuada magnéticamente sobre la que se colocan un conjunto de unidades robóticas.

En este artículo se presenta un mini-robot cámara inalámbrico que se adhiere a la pared abdominal mediante interacción magnética. A diferencia de los trabajos anteriores, la fuente magnética externa se coloca en el efector final de un brazo robótico, de manera que el desplazamiento de la cámara se puede realizar automáticamente mediante el desplazamiento del manipulador externo. Además, se presenta un control fuerza-posición con compensación de pares para el guiado del manipulador sobre la pared abdominal. Este artículo presenta la siguiente estructura. En la sección se describe el sistema robótico propuesto y se expone el diseño del robot cámara. La sección 3 muestra el control fuerza-posición con control de pares para el guiado de los imanes externos sobre la pared abdominal. Los resultados experimentales, junto con el algoritmo recurrente de mínimos cuadrados para la estimación de la constante de rigidez de la pared abdominal se presentan en la sección 4. Finalmente, las conclusiones se presentan en la sección 5.

2 MINI-ROBOT CAMARA

El sistema robótico propuesto en este trabajo está compuesto por un mini-robot cámarainalámbrico, que se introduce en la cavidad abdominal a través del puerto único, un brazo robótico sobre el que se adhiere una sujeción magnética (figura 1). El dispositivo interno está provisto de una serie de imanes, de manera que una vez en el interior del abdomen, se puede pegar a la pared abdominal mediante interacción magnética con la sujeción magnética. Como la sujeción magnética esta adherida al efector final de un brazo robótico, el guiado de la cámara se realiza mediante el desplazamiento del manipulador externo. De esta manera, el robot cámara se puede alejar del puerto de entrada, evitando el problema de la perdida de triangulación entre la cámara y las herramientas quirúrgicas. Además, gracias a que la cámara se puede desplazar a lo largo de la pared abdominal, se puede obtener un mayor número de puntos de vista de la cavidad abdominal de los que se obtendrían como una cámaralaparoscópica tradicional. Para el correcto desplazamiento del manipulador sobre la pared abdominal, se debe controlar tanto el movimiento del mismo en las direcciones tangentes a la superficie en el punto de contacto ($\vec{t}$), como la fuerza en la dirección normal ($\vec{n}$). Para ello, en la seccion 3 se propone un control hibrido fuerza-posicion con compensacion de pares. A continuación se describe el diseño de la cámararobótica.

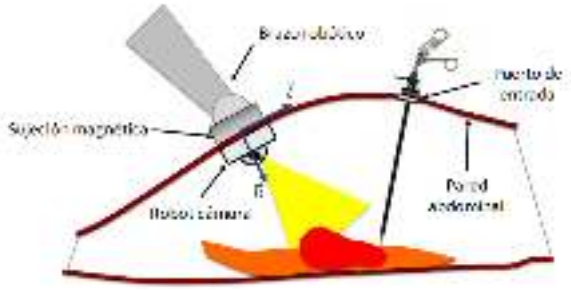


Figura 1. Sistema robótico completo

En la figura 2 se muestra tanto la cámara robótica como el sistema de agarre externo, formado por la sujeción magnética y la muñeca que va acoplada al manipulador. El sistema de acoplamiento entre la muñeca y la sujeción magnética permite que se pueda liberar la sujeción magnética una vez que la cámara se ha colocado en la posición deseada. De esta manera, se pueden manejar varios mini-robots con un único manipulador externo. Además, este sistema permite tanto el guiado automático de la cámara con el manipulador como la posibilidad de mover la sujeción magnética a mano. El robot cámara está provisto de dos GDL: rotación e inclinación. La rotación se consigue rotando la última articulación del manipulador externo, ya que los imanes internos de la cámara siguen el movimiento de la sujeción magnética. Por otro lado, el movimiento de inclinación está actuado por un motor lineal situado en la muñeca, cuyo movimiento provoca un desplazamiento del componente A de la figura 3. Este componente hueco está a su vez acoplado con el componente B, que tiene un pequeño imán en su parte inferior. Esta pieza está pegada al imán de la sujeción magnética C. Por otro lado, el desplazamiento del imán C provoca el desplazamiento de un imán cilíndrico (D), que se encuentra en el interior del robot cámara y está conectado con la cámara. De esta manera, el movimiento del motor permite variar la inclinación de la cámara en ± 42º.

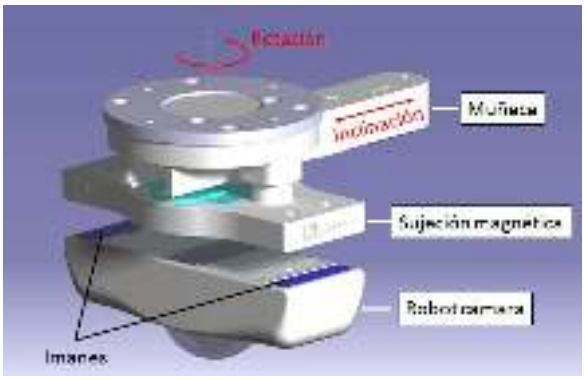


Figura 2. Diseño del robot cámara

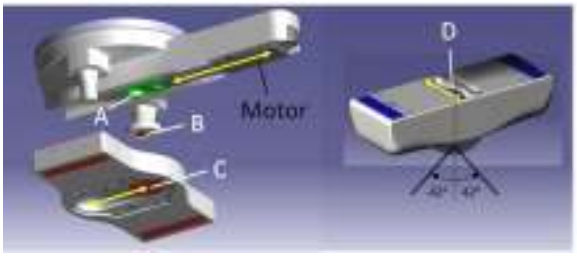


Figura 3. Inclinación de la cámara

La figura 4 muestra el primer prototipo de la cámara robótica, cuyas dimensiones son 12 cm de longitud y 4 cm de ancho. Las imágenes del interior de la cavidad abdominal se adquieren mediante una cámara inalámbrica, y son transmitidas a un receptor exterior. Además de los movimientos de rotación e inclinación, la cámara permite el uso de un zoom digital para enfocar pequeñas áreas. Para iluminar la escena, la cámara está provista de un conjunto de seis LEDs blancos, que emiten 6.8 candelas a 20 mA cada uno. Para alimentar el sistema se utilizan siete baterías de 9 voltios (250 mAh), que permiten que el sistema sea completamente inalámbrico. La autonomía del sistema es de alrededor de 4 horas, lo que resulta suficiente para un procedimiento de laparoscopia habitual, como una colecistectomía.



Figura 4. Primer prototipo del robot cámara

3 CONTROL FUERZA-POSICIÓN CON CONTROL DE PARES

Con el objetivo de asegurar que la interacción entre la sujeción magnética y la pared abdominal es apropiada, se ha desarrollado el control de fuerza-posición mostrado en la figura 5. Un planificador genera las referencias de fuerza y posición del sistema. La referencia de posición (P_r) especifica la posición deseada en las direcciones tangentes a la superficie ($\vec{t}$) en el punto de contacto, mientras que la referencia de fuerza (F_r) establece el valor de la fuerza en la dirección normal ($\vec{n}$). De esta manera, las acciones de control de fuerza y posición se encuentran desacopladas a través de las matrices *D* e *I-D*, evitando posibles interferencias entre ambos controladores. Como se muestra en la figura 5, el seguimiento de las referencias de fuerza y posición se realiza mediante un control híbrido con un *control de posición* y *fuerza* paralelos. Para mantener la orientación del efector final del manipulador perpendicular a la superficie, se incluye un bloque de *compensación de pares*, que permite que el robot se reoriente durante el movimiento. Este módulo es necesario debido a que las direcciones $\vec{t}$ y $\vec{n}$ son variantes en el tiempo, ya que la pared abdominal no es una superficie plana. El *control de orientación* permite el correcto seguimiento de la referencia de orientación. Dado que el efector final se mantiene perpendicular a la superficie durante el movimiento del manipulador, el sistema de referencia de la tarea coincide con el del manipulador ${}^0\boldsymbol{R_n}$, donde {0} representa el sistema de referencia de la base del robot y *n* la última articulación. La matriz de rotación anterior se utiliza para transformar las posiciones y orientaciones dadas en el sistema de referencia de la tarea al del manipulador, y viceversa. A continuación se describirá el control híbrido de fuerza-posición con compensación de pares propuesto.

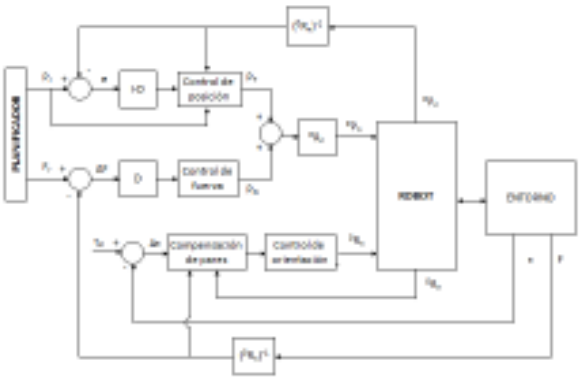


Figura 5. Esquema de control completo

3.1 CONTROL HÍBRIDO FUERZA-POSICIÓN

El control de fuerza transforma la referencia de fuerza en un movimiento a lo largo de la dirección normal (P_N), que sumado a la referencia de posición en las direcciones tangentes (P_T) proporciona una única referencia de posición ${}^0P_n$ al robot. Para el diseño del control de fuerza, se ha considerado que la interacción entre el robot y la superficie sigue un modelo elástico:

$$F = K_x \Delta P_N \qquad (1)$$

Donde K_x representa la constante de rigidez del entorno, y ΔP_N es la cantidad de desplazamiento. Si el error de fuerza ΔF representa la diferencia entre la referencia de fuerza y la fuerza medida, la ecuación (1) lleva a:

$$\Delta F = K_x\big(P_N(\Delta t) - P_N(t)\big) \qquad (2)$$

Por lo tanto, la posición requerida en la dirección normal para ejercer una fuerza de valor ΔF sobre la superficie es:

$$P_N(\Delta t) = P_N(t) + \Delta F \cdot \frac{1}{K_x} \qquad (3)$$

Por otro lado, el control de posición asegura el seguimiento de la referencia de posición. Esto se realiza mediante un controlador PI de ganancias K_Py K_I de acuerdo con la ecuación (4), donde P_r es la referencia de posición y *e(t)* el error de posición.

$$P_T(\Delta t) = P_r + K_P P_T(t) + K_I \int e(t)dt \qquad (4)$$

La dinámica del robot se fuerza a seguir una dinámica de primer orden [22], por lo que las ganancias PI se diseñan siguiendo la metodología de Ackerman de asignación de polos, con una estrategia dead-beat:

$$K_P = \frac{1+e^{-(T/\tau)}}{1-e^{-(T/\tau)}} \quad ; \quad K_I = \frac{1}{e^{-(T/\tau)}-1} \qquad (5)$$

Donde T representa el periodo de muestreo del sistema y τ la constante de tiempo.

3.2 COMPENSACIÓN DE PARES

El módulo de compensación de pares, representado en la figura 6, se encarga de mantener la orientación del manipulador perpendicular a la superficie en el punto de contacto. Cuando la fuerza de contacto no se distribuye de forma uniforme sobre la superficie, se generan una serie de pares en el efector final del manipulador. En este caso, la fuerza forma un ángulo αcon el eje*z* del efector final. Por lo tanto, el objetivo de la compensación de pares es mantener el eje $^{\mathbf{0}}\mathbf{z_n}$ paralelo a la fuerza de contacto, que es siempre perpendicular a la superficie. Para compensar los pares generados cuando el efector final no se encuentra perpendicular a la superficie, se realiza un giro de α en la dirección opuesta al vector de pares. El ángulo y dirección de rotación se calculan mediante el bloque *modelo geométrico* de la figura 6. Una vez que se obtiene la matriz de rotación, ésta se transforma al sistema de referencia del robot mediante la matriz $^{\mathbf{0}}\boldsymbol{R_n}$. La ganancia $\boldsymbol{K_\tau}$ de la figura 6 asegura el correcto seguimiento de la referencia nula de pares.

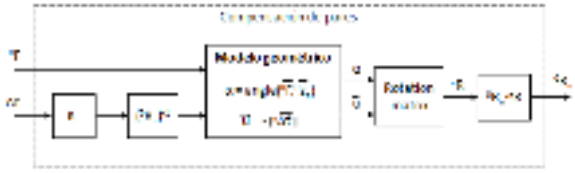


Figura 6. Módulo de compensación de pares

Una vez que se ha establecido la orientación deseada del manipulador, el *control de orientación* asegura su correcto seguimiento. Para ello se ha diseñado el controlador basado en cuaternios de la figura 7. La orientación deseada que proviene del módulo de compensación de pares en forma de matriz de rotación, se transforma en el cuaternio correspondiente q_d. La relación entre la velocidad articular $\dot{\boldsymbol{\theta}}$ y la derivada en el tiempo del cuaternio que representa la orientación del robot viene dada por medio de la matriz Jacobiana analítica $J_A(q)$ siguiente [23]:

$$\dot{q} = J_A(q)\dot{\theta} \qquad (6)$$

Considerando $q_e$como la diferencia entre el cuaternio deseado q_dy el actual:

$$q_e = q_d - q \qquad (7)$$

Dado que$q_d$es un valor constante, la combinación de las ecuaciones (6) y (7) proporcionan:

$$\dot{q}_e = -J_A(q)\dot{\theta} \qquad (8)$$

Definiendo la ley de control como$\dot{\theta} = J_A^T(q)Kq_e$, se elije la siguiente función cuadrática positiva como candidata a función de Lyapunov:

$$V(q_e) = \frac{1}{2}q_e^T K q_e \qquad (9)$$

Donde K es una matriz simétrica positiva. Derivando (10) respecto del tiempo se obtiene:

$$\dot{V}(q_e) = -q_e^T K J_A J_A^T K q_e \qquad (10)$$

Dado que tanto K como $J_A(q)$ son matrices definidas positivas, la derivada de la función de Lyapunov es negativa, y por lo tanto el sistema es asintóticamente estable.

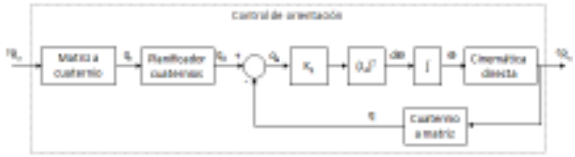


Figura 7. Control de orientación basado en cuaternios

Además se incluye un planificador de cuaternios para proporcionar una transición de orientación suave. La interpolación de los cuaternios se hace siguiendo una interpolación esférica lineal (SLERP) [24].

4 EXPERIMENTOS

4.1 IMPLEMENTACIÓN

Para validar el esquema de control propuesto en este artículo, se ha realizado un experimento in-vitro consistente en realizar un desplazamiento del robot cámara a lo largo de la dirección *y*, mientras se ejerce una fuerza en la dirección *z*. El montaje de dicho experimento se muestra en la figura 8. Se ha utilizado una caja de metacrilato de dimensiones 460 mm x 360 mm x 260 mm para simular la cavidad abdominal, con una capa de neopreno en la parte superior para simular un modelo elástico de la superficie. El guiado del robot cámara sobre la superficie se realiza con el brazo robótico de 7 GDL Barrett WAM. Este robot actuado mediante un sistema de cables presenta baja fricción y baja inercia [25], lo que hace posible mover el efector final a mano hasta la posición deseada. Además, sus características hacen de él un manipulador seguro para su interacción con humanos. Por otro lado, permite operar directamente en el dominio cartesiano, evitando la necesidad de largos cómputos para realizar la cinemática inversa. La fuerza de interacción entre el manipulador y la superficie se obtiene mediante un sensor de fuerza de 6 GDL integrado en el efector final [26]. Para garantizar la estabilidad del esquema de control completo, los controles de fuerza, posición y orientación se realizan con un periodo de muestreo inferior a la entrada de las referencias, lo que asegura que el robot se encuentra en estado estacionario antes de enviar una nueva referencia [27].



Figura 8. Implementación del sistema

El esquema de control descrito en la sección 3 se ha implementado mediante ROS, un framework especialmente diseñado para aplicaciones robóticas [28]. Esta plataforma permite integrar diferentes sistemas de actuación y sensado en el mismo ordenador. La principal ventaja que tiene utilizar ROS en nuestro sistema es la posibilidad de enviar y recibir información desde y hacia el robot y al motor que controla la inclinación de la cámara mediante un único interfaz. De esta forma, se consigue una interacción sencilla con el sistema al completo. Además, es posible integrar más dispositivos en el sistema de forma sencilla.

4.2 ESTIMACIÓN DE K_X

En el diseño de control de fuerza se ha considerado que el entorno (pared abdominal) sigue un modelo elástico. La estimación de su constante de rigidez se realiza mediante un algoritmo de mínimos cuadrados recurrentes. Este método emplea medidas de la posición y la fuerza obtenidas durante el movimiento del robot para actualizar el valor de la constante de rigidez K_x en tiempo real. El valor inicial de K_x se obtiene antes de la operación quirúrgica, colocando el manipulador a mano sobre la pared abdominal del paciente y ejerciendo una determinada presión sobre ella. Con los datos obtenidos se obtiene el valor inicial de la constante de rigidez, que se utilizará posteriormente en el control de fuerza durante el movimiento de la cámara. De esta manera, se tienen en cuenta las características particulares de cada paciente en el cálculo de la rigidez de la superficie de contacto.

Sean $F_z(t)$ y $z(t)$ las componentes *z* de la fuerza y la posición medidas respectivamente con respecto al sistema de referencia de la tarea en un instante determinado. La ecuación (2) se puede reescribir como:

$$F_z(t) - F_z(t-1) = K_x(z(t) - z(t-1)) \quad (12)$$

Definiendo los incrementos de fuerza y posición como $\Delta \boldsymbol{F} = \boldsymbol{F_z(t)} - \boldsymbol{F_z(t-1)}$y $\Delta \boldsymbol{z} = \boldsymbol{z(t)} - \boldsymbol{z(t-1)}$, la ecuación (12) resulta:

$$\Delta F = K_x \Delta z \quad (13)$$

El algoritmo de mínimos cuadrados recurrentes se inicializa en *t=0*. El valor estimado de la constante de rigidez K_x y la variable Q en el instante inicial se definen como:

$$\hat{K}_x(0) = \frac{\Delta F(0)}{\Delta z(0)} \quad (14)$$

$$Q(0) = \frac{1}{(\Delta z(0))^2} \quad (15)$$

Tomando nuevas medidas de los incrementos de fuerza y posición, la variable *R* se obtiene como:

$$R(t) = \frac{Q(t)\Delta z(t+1)}{1+\left(\Delta z(t+1)\right)^2 Q(t)} \quad (16)$$

Los valores consecutivos de K_x se obtienen mediante la siguiente ecuación:

$$\hat{K}_x(t+1) = \hat{K}_x(t) + R(t)\left(\Delta F(t+1) - \Delta z(t+1)\hat{K}_x(t)\right) \quad (17)$$

Y la variable Q se actualiza mediante:

$$Q(t+1) = \frac{Q(t)}{1+\left(\Delta z(t+1)\right)^2 Q(t)} \quad (18)$$

Para validar la fiabilidad del algoritmo propuesto, se ha realizado una serie de tres experimentos consistentes en medir la posición del efector final mientras se ejerce una fuerza sobre la superficie. Los resultados de dichos experimentos se muestran en la figura 9. El ruido que se observa se debe a que, debido a la alta rigidez de la superficie, los valores de los desplazamientos medidos son pequeños, especialmente al inicio de los experimentos. Como se puede observar, el valor de la constante de rigidez se estabiliza en un valor constante en los tres experimentos, de 1727.7 N/m para los dos primeros y de 1691.5 N/m en el tercero.

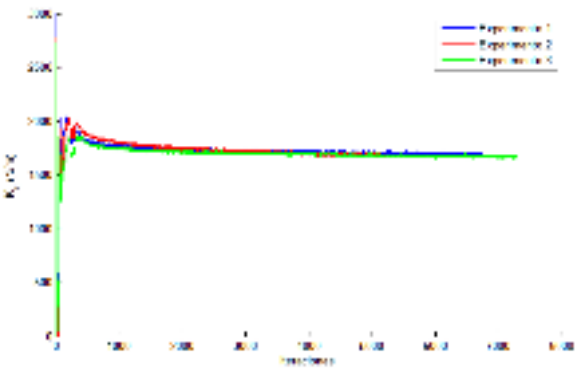


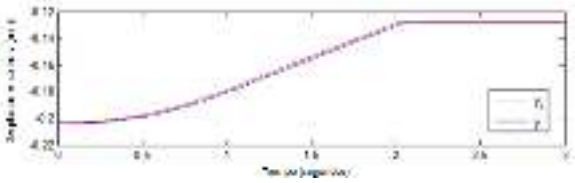
Figura 9. Estimación de la constante de rigidez

4.3 RESULTADOS

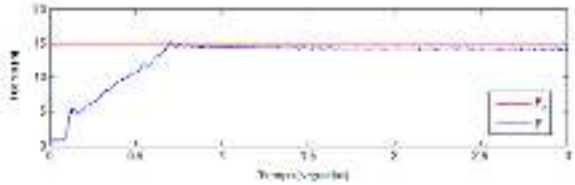
El experimento llevado a cabo consiste en desplazar el robot cámara 75 mm en la dirección *y* mientras se ejerce una fuerza de 15 N sobre la superficie que simula la pared abdominal. El periodo de muestreo es de 0.002 segundos para los controladores, y de 0.01 segundos para el resto del esquema de control. La constante de tiempo del sistema de primer orden impuesto al comportamiento dinámico del robot es $\tau = 0.01$ segundos, y las ganancias PI del controlador de posición son $K_P = 10.0333$ y $K_I = -5.5167$. Finalmente, las ganancias del control de orientación y de la compensación de pares son $K_\tau = 100$ y $K_q = 150$.

La figura 10 muestra el seguimiento de las referencias de posición, fuerza y par. El movimiento en posición se realiza mediante un perfil de velocidad trapezoidal, con una velocidad máxima de 0.05 m/s, con el fin de obtener un movimiento suave. Como se puede observar en la figura 10 (a), el seguimiento de la referencia de posición se realiza de forma precisa, con un pequeño retardo durante el estado transitorio. Por otro lado, la penetración en la superficie se realiza mediante un perfil lineal, tal y como se puede observar en la figura 10 (b). Los errores de fuerza y pares se deben principalmente a la resolución del sensor con el que se obtienen estas medidas, que es de 12 bits, con un error de 2 bits. En la figura 10 (c) se observa que el módulo del par medido comienza con un valor nulo, cuando no hay contacto con la superficie, y posteriormente se estabiliza en aproximadamente 0.3 N/m cuando la fuerza se estabiliza en el valor de referencia.

(a) Seguimiento de posición



(b) Seguimiento de fuerza



(c) Seguimiento de par

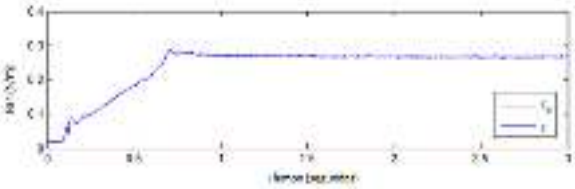


Figura 10. Resultados experimentales

5 CONCLUSIONES

El sistema robótico de visión propuesto en este trabajo representa una solución al problema de la pérdida de triangulación debido a la proximidad de la cámara y las herramientas quirúrgicas durante procedimientos de puerto único. Se ha diseñado un primer prototipo del robot cámara de 12 cm longitud y 4 cm de ancho. Los diseños futuros irán encaminados a reducir el tamaño total del robot. El robot incorpora un sistema de LEDs para iluminar la zona de operación. El guiado del robot a lo largo de la pared abdominal se realiza mediante interacción magnética con una sujeción que se coloca en el efector final de un manipulador externo. Este artículo

también propone un control híbrido fuerza-posición para desplazar el desplazamiento de la cámara. Se ha incluido un módulo de compensación de pares para mantener el manipulador perpendicular a la superficie de contacto durante el movimiento. La interacción con el entorno se ha modelado mediante un sistema elástico, cuya constante de rigidez se estima mediante un algoritmo de mínimos cuadrados recurrentes. Finalmente, se ha realizado un experimento in-vitro para validar el esquema de control propuesto. Futuros trabajos relacionados con este artículo incluirán validar el esquema de control con diferentes superficies de contacto.

Referencias

[1] J.R. Romanelli, L. Mark, P.A. Omotosho (2008). "Single port laparoscopic cholecystectomy with the TriPort system: a case report". Surgical Innovation, vol. 15, no. 3, pp: 223-228

[2] S.R. Philipp, B.W. Miedema, and K. Thaler, "Single-incision laparoscopic cholecystectomy using conventional instruments: early experience in comparison with the gold standard," Journal of the American College of Surgeons, Vol. 209, No. 5, pp. 632-637, 2009.

[3] P. Bucher, F. Pugin, and P. Morel, "Single port access lapaoscopic right hemiclectomy," International Journal Colorectal Dis., vol. 23, pp. 1013-1016, 2010.

[4] D.R. Cox, W. Zeng, M.M. Frisella and L.M. Brunt, "Analysis of standard multiport verus single-site access for laparoscopic skills training". Surg Endosc, vol. 25, pp. 1238-1244, 2011.

[5] L. Ott, Ph. Zanne, Fl. Nageotte, M. de Mathelin, and J. Gangloff,"Physiological motion rejection in flexible endoscopy using visual servoing," IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, May 19-23, 2008, pp. 2928-2933, 2008.

[6] V. Karimyan, M. Sodergren, J. Clark, G.Z. Yang, and A. Darzi, "Navigation systems and platforms in natural orifice translumenal endoscopic surgery (NOTES)," International Journal of Surgery, Vol. 7, No. 4, pp. 297-304, 2009.

[7] T. Hu, P.K. Allen, N.J. Hogle, and D.L. Fowler, "Insertable surgical imaging device with pan, tilt, zoom and lighting", 2008 IEEE International Conference on Robotics and Automation, pp. 2948-2953, May 2008.

[8] T. Kawahara, T. Takaki, I. Ishii, and M. Okajima, "Development of a broad-view camera system for minimally invasive surgery", 2010 IEEE/RSJ Internatinal Conference on Intelligent Robots and Systems, pp. 2810-2815, October 2010.

[9] C.A. Castro, S. Smith, A. Alqassis, T. Ketterl, Y. Sun, S. Ross, A. Rosemurgy, P.P. Savage, and R.D. Gitlin, "MARVEL: a wireless miniature anchored robotic videoscope for expedited laparoscopy", 2010 IEEE Internatinal Conference on Robotics and Automation RiverCentre, pp. 2926-2931, May 2012.

[10] Y. Sun, A. Anderson, C. Castro, B. Lin, R. Gitlin, S. Ross, and A. Rosemurgy, "Virtually transparent epidermal imagery for laparo-endoscopic single-site surgery", 33rd Annual International Conference of the IEEE EMBS, pp. 2107-2110, 2011.

[11] I. S. Zeltser, R. Bergs, R. Fernandez, L. Baker, R. Eberhart, and J. A. Cadeddu, "Single trocar laparoscopic nephrectomy using magnetic anchoring and guidance system in the porcine model," J. Urol., vol. 178, pp. 288–291, 2007.

[12] Endoscopy. 2010 Jun;42(6):481-6. A magnetic internal mechanism for precise orientation of the camera in wireless endoluminal applications.Valdastri P, Quaglia C, Buselli E, Arezzo A, Di Lorenzo N, Morino M, Menciassi A, Dario P.

[13] J. A. Cadeddu, R. Fernandez, M. Desai, R. Bergs, C. Tracy, S. J. Tang,P. Rao, M. Desai, and D. Scott, "Novel magnetically guided intraabdominal camera to facilitate laparoendoscopic single-site surgery: Initial human experience," Surg. Endosc., vol. 23, pp. 1894–1899, 2009.

[14] M. Fakhry, B. Gallagher, F. Bello, and G. B. Hanna, "Visual exposure using single-handed magnet-driven intra-abdominal wireless camera in minimal access surgery is better than 30◦ endoscope," Surg. Endosc., vol. 23, pp. 539–543, 2009.

[15] P. Swain, R. Austin, K. Bally, and R. Trusty, "Development and testing of a tethered, independent camera for notes and single-site laparoscopic procedures," Surg. Endosc., vol. 24, pp. 2013–2021, 2010.

[16] S. R. Platt, J. A. Hawks, andM. E. Rentschler, "Vision and task assistance using modular wireless in vivo surgical robots," IEEE Trans. Biomed. Eng., vol. 56, no. 6, pp. 1700–1710, Jun. 2009.

[17] M. Nokata, S. Kitamura, and T. Nakagi, "Capsule type medical robot with magnetic drive in abdominal cavity," Proceedings of the 2nd Biennial IEEE/RAS-WMBS International Conference on Biomedical Robotics and Biomechatronics, pp. 348-353, 2008.

[18] F. Carpi, N. Kastelein, M. Talcott, and C. Pappone, "Magnetically controllable gastrointestinal steering of video capsules," IEEE Transactions on Biomedical Engineering, Vol. 58, No. 2, pp. 231-234, 2011.

[19] A.C. Lehman, J. Dumpert, N.A. Wood, L. Redden, A.Q. Visty, S. Farritor, B. Varnell, and D. Oleynikov, "Natural orifice cholecystectomy using a miniature robot," Surgical Endoscopy Journal, Vol. 23, pp. 260-266, 2009.

[20] M. Simi, M. Silvestri, C. Cavallotti, M. Vatteroni, P. Valdastri, A. Menciassi, and P. Dario, "Magnetically Activated Stereoscopic Vision System for Laparoscopic Single-Site Surgery," in IEEE/ASME Transactions on Mechatronics, Vol. 18, No.3, pp. 1140-1151, June 2013

[21] S. Tognarelli, M. Salerno, G. Tortora, C. Quaglia, P. Daria, and A. Menciassi, "An endoluminal robotic platform for minimally invasive surgery", The fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, pp. 7-12, June 2012.

[22] V.F. Muñoz, I. García-Morales, C. Perez del Pulgar, J.M. Gomez-DeGabriel, J. Fernández-Lozano, A. García-Cerezo, C. Vara-Thorbeck, and R. Toscano, "Control movement scheme based on manipulability concept for a surgical robotic assistant", IEEE International Conference on Robotics and Automation, pp. 245-250, May 2006.

[23] H.L. Pham, V. Perdereau, B.V. Adorno, and P. Fraisse, "Position and orientation control of robot manipulators using dual quaternion feedback", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 658-663, October 2010.

[24] T. Barrera, A. Hast, and E. Bengtsson, "Incremental Spherical Linear Interpolation", SIGRAD 2004 Conference, pp. 7-10, November 2004.

[25] B. Rooks, "The harmonious robot" Industrial Robot: An International Journal, vol. 33, no. 2, pp. 125–130, 2006.

[26] Barrett Technology, Inc., "F/T Sensor", 2012. [Online]. Available: http://www.barrett.com/robot/DS_FTSENSOR.pdf

[27] E. Bauzano, V.F. Muñoz, I. García-Morales, and B. Estebanez, "Three-layer control for active wrists in robotized laparoscopic surgery", IEEE/RSJ Internatinal Conference on Intelligent Robots and Systems, pp. 2653-2658, October 2009.

[28] M. Quigley, K. Conley, B. Gerkey, J. Faust, T.B. Foote, J. Leibs, R. Wheeler, and A.Y. Ng, "ROS: an open-source Robot Operating System", International Conference on Robotics and Automation, ser. Open-Source Software workshop, 2009.

Desarrollo de un Sistema Multimodal de Rehabilitación asistida por Robots

F. Badesa[1], R. Morales[1], N. Garcia-Aracil[1], J.M. Sabater[1], Miguel Almonacid[2]
[1]Neuroingenieria Biomedica
Universidad Miguel Hernández
[2]Universidad Politécnica de Cartagena
email: rmorales@umh.es

Resumen

En este artículo se presenta un nuevo sistema robótico de ayuda a la rehabilitación de pacientes que hayan sufrido un ACV o un traumatismo craneoencefálico que tiene por objetivo principal maximizar la motivación y participación del paciente en la terapia y evaluar de forma continua el progreso de la terapia de rehabilitación desde el punto de vista funcional. La principal novedad del sistema es considerar al paciente dentro del lazo de control del sistema y emplear información multisensorial proveniente del paciente para modificar de forma adaptativa y dinámica la asistencia proporcionada por el dispositivo robótico y la realimentación multimodal del entorno de realidad virtual durante la ejecución de terapias de rehabilitación.

Palabras clave: Robótica, rehabilitación, sistemas multimodales, bioseñales

1 INTRODUCCIÓN

La expresión daño cerebral sobrevenido(DCS) designa una patología asociada a una afectación del sistema nervioso central, y configura una realidad sanitaria y social, de magnitud creciente y gravedad extraordinaria, que exige una respuesta cada vez más compleja y especializada, teniendo en cuenta los avances tecnológicos y el estado de la investigación en cada momento [1]. El daño se origina por un conjunto variado de etiologías, las más frecuentes de las cuales son el accidente cerebrovascular (ACV) y el traumatismo craneoencefálico (TCE), cuyo denominador común consiste en provocar una lesión cerebral que irrumpe de forma decisiva en el desarrollo vital de la persona. Un accidente cerebrovascular es una pérdida súbita, no convulsiva de la función neurológica debido a un evento vascular intracraneal isquémico o hemorrágico [2]. En general, los accidentes cerebrovasculares se clasifican por la ubicación anatómica en el cerebro, distribución vascular, etiología, edad del individuo afectado y naturaleza hemorrágica versus no hemorrágica [3]. La prevalencia del accidente cerebrovascular depende de la edad y la estructura por sexo de la población y se estima que es un 1% de la población [4] . El accidente cerebrovascular, junto con la cardiopatía isquémica, es el motivo mayor de la carga de la enfermedad: en los países de ingresos bajos y medios de Europa y Asia Central representan más de un cuarto de la carga total de la enfermedad [5]. Según el informe de la fundación española de enfermedades neurológicas sobre el impacto sociosanitario de las enfermedades neurológicas en España publicado en 2006 [6], el ictus (una de las accidentes cerebrovasculares más importantes) es la segunda causa de muerte en los españoles y la primera causa de muerte entre las mujeres (cuando se consideran las enfermedades específicas). La tasa de incidencia en España se estima en 200 ictus por cada 100.000 habitantes, por lo que en 2004 deberían haber ocurrido 86.000 nuevos ictus en nuestro país. Sin embargo desde hace más de 5 años, el número de pacientes con ictus que son tratados en los hospitales españoles superan los 100.000 pacientes por año y van a un ritmo creciente, por lo que las estimaciones se han quedado claramente por debajo de la realidad. El accidente cerebrovascular es una causa principal del deterioro crónico en la función del brazo y puede afectar a muchas actividades cotidianas. Al ingresar al hospital después de un accidente cerebrovascular, más de dos tercios de todos los pacientes presentaron una paresia del brazo que resultó en una reducción de la función de la extremidad superior [7, 8] y seis meses después del accidente cerebrovascular, el brazo afectado de aproximadamente la mitad de los pacientes permanecerá sin función [9] . La literatura indica que el grado de deterioro inicial junto con el gradiente de la recuperación de la paresia del brazo cumple una función importante en la recuperación de la función del brazo y en la reducción de las limitaciones de las actividades en individuos con un accidente cerebrovascular crónico [10, 11, 12]. Por lo tanto, para reducir esta carga, muchos pacientes reciben un enfoque de rehabilitación multidisciplinaria poco después del accidente cerebrovascular. Sin embargo, a pesar de los esfuerzos intensivos de rehabilitación, sólo aproximadamente el 5% al 20% alcanzan una recuperación funcional

completa [8]; en otras palabras, cuatro de cinco pacientes dejan la rehabilitación con una función del brazo restringida. Por lo tanto, todavía existe una necesidad urgente de alcanzar nuevas estrategias de rehabilitación y entrenamiento de hospitalizados y ambulatorios que concuerden con las necesidades específicas de los pacientes y sus parientes [13].

Según el informe de la fundación española de enfermedades neurológicas sobre el impacto sociosanitario de las enfermedades neurológicas en España publicado en 2006 [6], si las previsiones de población se cumplen en 2050 el coste que supondría para el sistema sanitario, la atención de todos los pacientes que presentasen un ictus, alcanzaría cifras, verdaderamente impresionantes: para una incidencia de 266,66/100.000 habitantes en la población general, se producirían unos 145.000 nuevos ictus y su coste rondaría los 1.900 millones de euros a lo que habría que sumar los costes de todos los pacientes dependientes supervivientes y el incremento de la incidencia correspondiente al aumento derivado de la mayor proporción de población anciana en las curvas de población.

Las estrategias futuras en ictus publicadas por el Ministerio de sanidad y consumo [14], una de las líneas prioritarias de investigación en ictus, extrapolable al TCE, es la investigación en la eficiencia de intervenciones terapéuticas rehabilitadoras. El sistema multimodal de rehabilitación robótica presentado en este artículo estaría en línea con las piroridades marcadas por el ministerio ya que su principal objetivo es maximizar la motivación y participación del paciente en la terapia y como resultado mejorar la eficiencia de la misma.

La evidencia indica que las terapias de realización repetitiva de tareas específicas asistidas por robots pueden ser más eficaces para la reducción a largo plazo de las alteraciones motrices en un paciente afectado por un ACV, tanto en términos de fuerza y estado motor como en lo relativo a la reducción de la espasticidad y el tono musculares [15, 22, 17, 19, 20] . Por otra parte, los tratamientos asistidos por dispositivos mecánicos pueden ofrecer mediciones objetivas del rendimiento de los pacientes que son útiles y fácilmente analizables por los clínicos y los fisioterapeutas.

Los dispositivos de rehabilitación robótica se pueden agrupar en dos categorías principales [18]:

- Dispositivos de rehabilitación del tipo exoesqueleto, es decir, dispositivos robóticos portátiles que están diseñados para aproximar el movimiento de la extremidad humana, tanto en el espacio articular como en el espacio operativo.

- Dispositivos para la rehabilitación funcional de tipo efector final, es decir, los dispositivos robóticos diseñados de tal manera que sólo el movimiento del efector final del robot en el espacio de funcionamiento es equivalente a la del efector natural de la extremidad humana.

En la literatura científica se pueden encontrar varios ejemplos de estudios clínicos basados en la aplicación de dispositivos robóticos a las terapia de rehabilitación motora de pacientes que han sufrido un ACV o TEC [21]. Sin embargo en la literatura científica hay pocos ejemplos de sistemas de rehabilitación asistida por robots que incorporen en el lazo de control del sistema información del estado del paciente y su desempeño durante la realización de una terapia virtual asistida por dispositivos robóticos [22, 23, 24, 25, 26, 28, 29, 30]. El primer trabajo, en el cual se propone un algoritmo que controla el nivel de asistencia de un robot de rehabilitación planar en función de la exactitud y la fuerza con la que se realiza el movimiento, fue presentado en 2003 por Krebs et al. [22]. Posteriormente, los trabajos de Novak y Koeing [24, 25] propusieron la utilización de medidas psicofisiológicas del estado del paciente y el análisis del desempeño de la terapia para ajustar la dificultad de una tarea de rehabilitación asistida por el robot Haptic Master. Cabe destacar en esta línea, el proyecto MIMICS es un proyecto de investigación financiado por la Comisión Europea, desarrollado por el Instituto Federal de Tecnología en Suiza (ETH Zurich), la Universidad de Ljubljana, la Universidad de Barcelona, la empresa HOCOMA AG, y la clínica neurológica de Bad Aibling [27]. La hipótesis principal de este proyecto es que el entrenamiento motor para la neurorehabilitación se puede mejorar sustancialmente mediante una realimentación sensorial inmersiva y multimodal. El enfoque es la adquisición en tiempo real de los datos de comportamiento y señales fisiológicas de los pacientes y el uso de estos parámetros para cambiar de forma adaptativa y dinámica las pantallas de un sistema de realidad virtual, con el objetivo de maximizar la motivación del paciente.

En este paper se presenta el desarrollo de un nuevo sistema robótico de ayuda a la rehabilitación de pacientes que hayan sufrido un ACV o un traumatismo craneoencefálico que tiene por objetivo principal maximizar la motivación y participación del paciente en la terapia y evaluar de forma continua el progreso de la terapia de rehabilitación desde el punto de vista funcional.

2 DISEÑO Y DESARROLLO DEL SISTEMA ROBÓTICO

La plataforma robótica multimodal propuesta en este capítulo debe cumplir con los requisitos cinemáticos impuestos por el espacio de trabajo necesario para realizar actividades de la vida diaria (AVD) tales como peinarse, lavarse la cara, beber un vaso de agua y similares. Para analizar dicho espacio de trabajo, es necesario disponer de un modelo cinemático del brazo humano. En este caso se emplea el modelo cinématico simplificado de tres eslabones rígidos y seis GDL mostrado en la Figura 1. Donde el eslabón del hombro representa la clavícula y la escápula, el eslabón del brazo representa el húmero y el eslabón del antebrazo representa el radio y el cúbito.

Debido a las propiedades anatómicas del brazo, de los huesos y de los músculos, algunos límites articulares dependen del valor articular de otras articulaciones. En el modelo empleado los límites articulares de la flexión-extensión q_4 se especifican como función lineal de la abducción-aducción según $q_4 = [-43° + q_3/3, 153° - q_3/6]$. Los límites de la rotación q_5 son función lineal de los ángulos de abducción-aducción y flexión-extensión según:

$$q_5 = [-90° + 7q_3/9 - q_4/9 + 2q_3q_4/810\,,\\ 60° + 4q_3/9 - 5q_4/9 + 5q_3q_4/810].$$

Los límites del resto de articulaciones son considerados constantes e independientes y toman los siguientes valores: $q_1 = [-17°,17°]$, $q_2 = [-6°,20°]$, $q_3 = [-9°,160°]$, $q_6 = [-90°,60°]$. Los valores de los límites articulares y las longitudes de los eslabones se han extraído del trabajo presentado en [31].

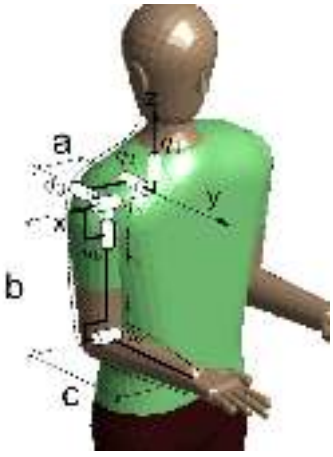


Figura 1: Modelo Cinemático Simplificado. Propuesto por [31].

Una vez establecida la cadena cinemática y establecidos los limites articulares de cada una de las articulaciones se procede a obtener el espacio alcanzable por el punto definido por la muñeca mediante la resolución de la cinemática directa. Los valores de las coordenadas articulares son discretizados dentro de sus límites. Los resultados del problema cinemática directo para cada uno de los valores discretizados son almacenados para generar el espacio alcanzable por la muñeca mostrado en la Figura 4. Dado que el espacio alcanzable es muy amplio y debido a que solo una parte de dicho espacio es empleado para actividades de la vida diaria, el siguiente paso se centra en analizar el subespacio necesario para realizar . Para este fin se emplean dos IMU colocadas sobre el brazo de un sujeto control. El sujeto realiza diversas AVD tales como comer, beber, coger un vaso, peinarse y similares mientras se recoge la información en las IMU. Posteriormente se reconstruye la cinemática del brazo mediante la información extraída de las IMU para extraer las trayectorias generadas en la realización de AVD.

El siguiente paso es diseñar una cadena cinemática robótica capaz de abarcar todo el espacio de trabajo necesario para la realización de actividades de la vida diaria obtenido mediante el análisis anterior. Como requerimientos adiciones del robot se establecen:

- Cadena cinemática serial.
- Accionamientos eléctricos ya que permite implementar distintos tipos de control con exactitud.
- Un único punto de agarre situado sobre la muñeca y que tenga una capacidad de carga de 10kg.
- Ligero y seguro.

Con el fin de cumplir los requisitos propuestos y alcanzar todo el espacio de trabajo necesario, se decide implementar un robot serial redundante de 7 GDL. Para el diseño de la configuración cinemática del robot se ha considerado los datos obtenidos en el estudio del espacio de trabajo requerido. Los parámetros de Denavit-Hatenberg de la configuración cinemática que cumple los requisitos de espacio de trabajo se muestran en la Tabla 2 Con el fin de cumplir los requisitos propuestos, se seleccionan los motores eléctricos PRL fabricados por schunk para los GDL activos del robot propuesto. Esto motores llevan integrada toda la electrónica necesaria para su funcionamiento así como la caja reductora haciendo de estos motores un sistema compacto. Cabe destacar que poseen eje hueco, por lo que facilita el cableado

de los diferentes motores en configuración serial. La electrónica de control de los motores permite controlarlos tanto en posición como velocidad, además posee un lazo de control en intensidad que podría aproximarse a un control de par, sin embargo, este tipo de control tiene el inconveniente de que la relación intensidad-par posee grandes zonas muertas que dificultan cualquier estrategia de control que necesite aplicar pares motores a las articulaciones. Concretamente se seleccionaron: el modelo PRL 120 que proporciona un par nominal de 210 Nm para las dos primeras articulaciones; el modelo PRL 100 que proporciona un par nominal de 81.5 Nm para las articulaciones 3 y 4; el modelo PRL 80 que proporciona un par nominal de 20.1 Nm para las articulaciones 5 y 6 y el modelo PRL 60 que proporciona un par nominal de 9.6 Nm para la séptima articulación. Con el objetivo de conseguir un acople rápido y seguro se diseña un efector final del robot que como medida de seguridad integra un electroimán con el fin de conseguir un desacople rápido entre el robot y la persona ante cualquier fallo. Debido a las necesidades de control, se integra también un sensor de fuerza-par de 6 GDL. La información proveniente de este sensor se emplea en el lazo de control para la interacción física hombre-robot. Finalmente, se realiza un estudio de requerimientos dinámicos del robot con el fin de evaluar si los motores seleccionados soportan el requisito de carga en el efector final del robot. La carga máxima a soportar por el robot debe alcanzar los 100 N que se considera fuerza suficiente para soportar el peso del brazo humano y las fuerzas de interacción. Así mismo, cabe destacar que este valor está por debajo de las cargas máximas que puede soportar el sensor de fuerza-par. En la Figura 2 se muestra la gráfica que relaciona los pares necesarios en función de la carga en el efector final y el par nominal de cada uno de los motores seleccionados.

Tabla 1: Parámetros Denavit-Hartenberg.

Link	$\boldsymbol{\theta_i}$	$\mathbf{d_i}$	$\mathbf{a_i}$	$\boldsymbol{\alpha_i}$
1	0	-0.3060	0	$\pi/2$
2	0	0	0	$-\pi/2$
3	0	-0.3910	0	$\pi/2$
4	0	0	0	$-\pi/2$
5	0	-0.2975	0	$\pi/2$
6	0	0	0	$-\pi/2$
7	0	-0.3771	0	π

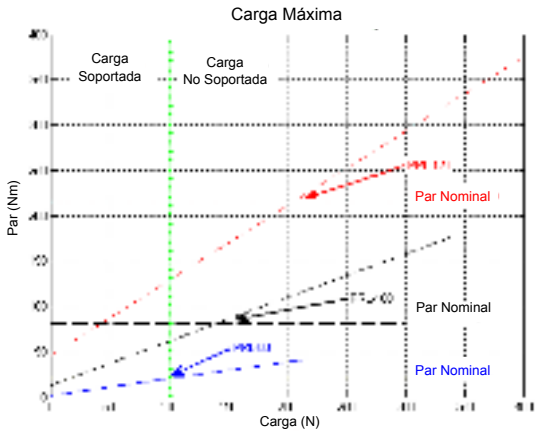


Figura 2: Gráfica Par-Carga en el Efector Final.

3 SISTEMA DE ESTIMACIÓN DEL ESTADO FISIOLÓGICO DEL PACIENTE

El sistema multimodal para la administración de terapias de rehabilitación se muestra en la Figura 5. El sistema está compuesto por dos computadores, una plataforma robótica en contacto con el paciente y un sistema de captación de señales fisiológicas. A continuación se describe con más detalle cada uno de los bloques. En la parte baja se muestra la terapia asistida por robots, de la cual se extrae la información del robot, del brazo humano y de las señales fisiológicas del paciente. La información proveniente del robot es tratada en la propia controladora del robot. Por otra parte, la reconstrucción cinemática del brazo humano se realiza mediante las IMU colocadas en el brazo del paciente y la posición del efector final del robot. Con esta reconstrucción es posible representar el brazo virtual y evaluar si los movimientos del paciente se realizan correctamente. En el sistema de registro y procesado de las señales fisiológicas se procesan *on-line* con el fin de interaccionar con los diferentes bloques del sistema. Toda la información recogida por el sistema llega al entorno de realidad virtual. El software es el encargado de gestionar y reenviar la información necesaria hacia: i) el bloque de control adaptativo, adapta el nivel de asistencia del robot y ii) hacia al paciente por medio de retro-alimentación visual, sonora y vibro-táctil. Las retro-alimentaciones visuales y sonoras son generadas en el propio ordenador mientras que la vibro-táctil se realiza mediante pequeños vibradores colocados en el brazo del paciente. Estos vibradores son accionados cuando el paciente alcanza una configuración incorrecta de su brazo y debe corregirla hacia la configuración correcta. Finalmente, el bloque control adaptativo analiza toda la información registrada para

determinar el estado del paciente y modificar el nivel de asistencia del robot en función de su estado. El control adaptivo modifica las ganancias a bajo nivel de la estrategia de control implementada en la controladora del robot. En la Figura 3 se muestran todos los elementos que componen el sistema en la fase de experimentación.



Figura 3: Sistema Multimodal de Rehabilitación en experimentación.

4 CONCLUSIONES

En este artículo se presenta el desarrollo de un sistema multimodal de rehabilitación asistida por robots teniendo en mente su uso en la rehabilitación funcional de la actividades de la vida diaria. El sistema se ha diseñado para poder asistir y motivar al paciente durante la ejecución de la actividades virtuales de la vida diaria. La principal motivación de este trabajo reside en diseñar nuevas estrategias de control para maximizar la motivación y participación del paciente en la terapia considerando al paciente dentro del lazo de control del sistema y empleando información multisensorial proveniente del mismo para modificar de forma adaptativa y dinámica la asistencia proporcionada por el dispositivo robótico y la realimentación multimodal del entorno de realidad virtual durante la ejecución de terapias de rehabilitación.

5 Agradecimientos

Este trabajo has sido financiado por el Plan Nacional de I+D+i a través del proyecto "Interpretación de la intención y actuación humana mediante señales biomédicas y el análisis cinemático y dinámico del movimiento" (DPI2011-29660-C04-04).

Referencias

[1] Daño cerebral sobrevenido en España: un acercamiento epidemiológico y sociosanitario. Madrid, 2005.

[2] World Health Organization. Cerebrovascular accident, stroke. *www.who.int*

[3] Adams HP Jr, Bendixen BH, Kappelle LJ, Biller J, Love BB, Gordon DL, et al. Classification of subtype of acute ischemic stroke. Definitions for use in a multicenter clinical trial. TOAST. Trial of Org 10172 in Acute Stroke Treatment. Stroke 1993;24(1):35-41.

[4] Bonita R, Solomon N, Broad JB. Prevalence of stroke and stroke-related disability. Estimates from the Auckland stroke studies. Stroke 1997;28(10):1898-902.

[5] Lopez A, Mathers C, Ezzati M, Jamison D, Murray C. Global and regional burden of disease and risk factors, 2001: systematic analysis of population health data. Lancet 2006;367:1747-57

[6] Impacto sociosanitario de las enfermedades neurológicas, Informe de la fundación española de enfermedades neurológicas, 2006

[7] Jørgensen HS, Nakayama H, Raaschou HO, Olsen TS. Stroke. Neurologic and functional recovery. The Copenhagen Stroke Study. Physical Medicine and Rehabilitation Clinics of North America 1999;10(4):887-906.

[8] Nakayama H, Jørgensen HS, Raaschou HO, Olsen TS. Recovery of upper extremity function in stroke patients: the Copenhagen Stroke Study. Archives of Physical Medicine and Rehabilitation 1994;75(4):394-8.

[9] Kwakkel G, Kollen BJ, van der Grond J, Prevo AJ. Probability of regaining dexterity in the flaccid upper limb: impact of severity of paresis and time since onset in acute stroke. Stroke 2003;34(9):2181-6.

[10] Ada L, O'Dwyer N, O'Neill E. Relation between spasticity, weakness and contracture of the elbow flexors and upper limb activity after stroke: an observational study. Disability and Rehabilitation 2006;28:891-7.

[11] Harris JE, Eng JJ. Paretic upper-limb strength best explains arm activity in people with stroke. Physical Therapy 2007;87(1):88-97.

[12] Hendricks HT, van Limbeek J, Geurts AC, Zwarts MJ. Motor recovery after stroke: a systematic review of the literature. Archives of Physical Medicine and Rehabilitation 2002;83(11):1629-37.

[13] Barker RN, Brauer SG. Upper limb recovery after stroke: the stroke survivors' perspective. Disability and Rehabilitation 2005;30(20):1213-23.

[14] Estrategias en Ictus del Sistema Nacional de Salud. Ministerio de sanidad y consumo. 2008

[15] Aisen, M.L., Krebs, H.I., Hogan, N., McDowell, F. and Volpe, B.T., The effect of robot-assisted therapy and rehabilitative training on motor recovery following stroke, Arch Neurol, 54 (1997) 443-6.

[16] H.I. Krebs, J.J. Palazzolo, L. Dipietro, M. Ferraro, J. Krol, K. Rannekleiv, B.T. Volpe, N. Hogan, "Rehabilitation Robotics: Performance-Based Progressive Robot-Assisted Therapy", Autonomous Robots, vol. 15, pp. 7–20, 2003.

[17] Krebs HI, Volpe BT, Aisen ML, Hogan N. Increasing productivity and quality of care: Robot-aided neurorehabilitation. J Rehabil-Res Dev. 2000;37: 639-52.

[18] . P. Dario, E. Guglielmelli, M.C. Carrozza, S. Micera, L. Didietro, F. Pisano, ?Sistemi robotici e meccatronici per la neuroriabilitazione?, Bioingegneria della Postura e del Movimento, A. Cappello, A. Cappozzo, P.E. di Prampero, Patron Ed., 2003, pp. 465-493.

[19] Volpe BT, Krebs HI, Hogan N, Edelsteinn L, Diels CM, Aisen ML: Robot training enhanced motor outcome in patients with stroke maintained over 3 years. Neurology 1999, 53:1874-1876.

[20] Volpe BT, Krebs HI, Hogan N, Edelstein OL, Diels C, Aisen M: A novel approach to stroke rehabilitation: robot-aided sensorimotor stimulation. Neurology 2000, 54:1938-1944.

[21] Prange GB, Jannink MJ, Groothuis-Oudshoorn CG, Hermens HJ, Ijzerman MJ. Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke. Journal of Rehabilitation Research and Development 2006;43(2):171-84.

[22] HI Krebs, JJ Palazzolo, L Dipietro, M Ferraro, J Krol, K Rannekleiv, BT Volpe, N Hogan. Rehabilitation robotics: performance-based progressive robot- assisted therapy. Autonomous Robots. 2003, 15: 7-20.

[23] S Balasubramanian,Hang Zhang,S Buchanan,H Austin,R Herman,He Jiping. Cooperative and active assistance based interactive therapy. Complex Medical Engineering (CME), 2010 IEEE/ICME International Conference on. 2010: 311-315.

[24] D Novak, M Mihelj, J Ziherl, A Olensek, M Munih. Psychophysiological mea- surements in a biocooperative feedback loop for upper extremity rehabilitation. Neural Systems and Rehabilitation Engineering, IEEE Transactions on. 2011, 19: 400-410.

[25] A Koenig, D Novak, X Omlin, M Pulfer, E Perreault, L Zimmerli, M Mihelj, R Riener. Real-time closed-loop control of cognitive load in neurological pa- tients during robot-assisted gait training. Neural Systems and Rehabilitation Engineering, IEEE Transactions on. 2011, 19: 453-464.

[26] A Duschau-Wicke, A Caprez, R Riener. Patient cooperative control increases active participation of individuals with sci during robot-aided gait training. Journal of Neuro-Engineering and Rehabilitation. 2010: 7.

[27] M. Munih, R. Riener, G. Colombo, L. Lunenburger, F. Muller, M. Slater, and M. Mihelj, ?MIMICS: Multimodal immersive motion rehabilitation of upper and lower extremities by exploiting biocooperation principles,? 2009 IEEE Interna- tional Conference on Rehabilitation Robotics, pp. 127?132, 2009.

[28] R Riener, M Munih. Guest editorial special section on rehabilitation via bio- cooperative control. Neural Systems and Rehabilitation Engineering, IEEE Transactions on. 2010, 18: 337-338.

[29] C Rodriguez Guerrero, J Fraile Marinero, J Perez Turiel, P Rivera Farina. Bio cooperative robotic platform for motor function recovery of the upper limb after stroke. EMBC Annual International Conference of the IEEE. 2010, 31: 4472-4475.

[30] Badesa, F.J.; Morales, R.; Garcia-Aracil, N.; Sabater, J.M.; Perez-Vidal, C.; Fernandez, E., "Multimodal Interfaces to Improve Therapeutic Outcomes in Robot-Assisted Rehabilitation", Systems, Man, and Cybernetics, Part C: Ap- plications and Reviews, IEEE Transactions on , vol.42, no.6, pp.1152,1158, Nov. 2012

[31] J. Lenarcic and A. Umek. Simple model of human arm reachable workspace. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(8):1239 –1246, aug 1994.

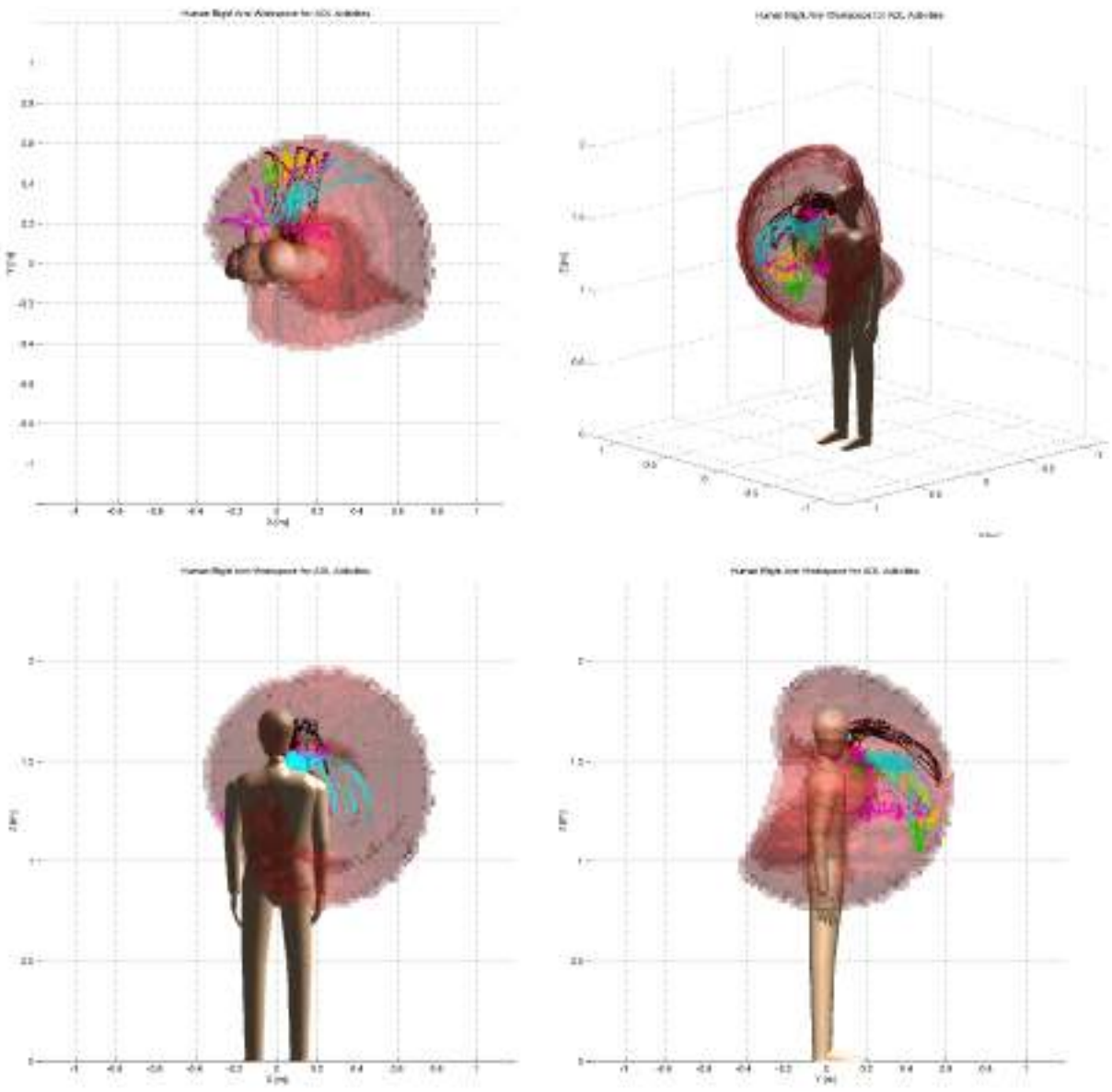


Figura 4: Espacio de Trabajo del Brazo Humano y Trajectorias 3D Típicas en AVD.

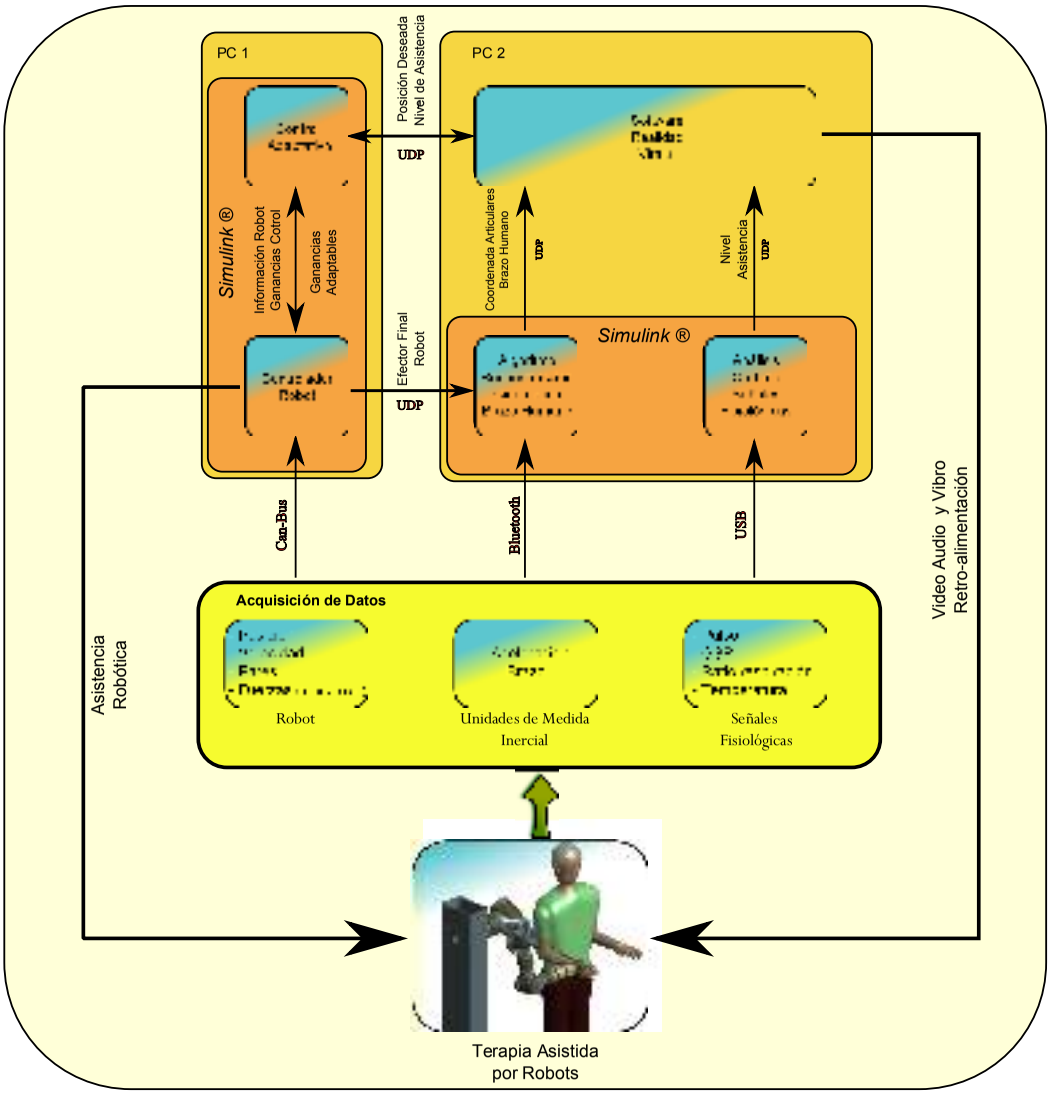


Figura 5: Diagrama del Sistema Completo.

ASPECTOS DE DISEÑO Y EVALUACIÓN PRELIMINAR DE LA PLATAFORMA ROBOTIZADA DE NEURO-REHABILITACIÓN PHYSIOBOT

Juan-Carlos Fraile Marinero, Javier Pérez Turiel
Escuela de Ingenierías Industriales, Universidad de Valladolid, C/ Paseo del Cauce 59, 47011 Valladolid, jcfraile@eii.uva.es

Carlos Rodríguez Guerrero, Rubén Alonso Alonso
División de Ingeniería Biomédica, Centro Tecnológico CARTIF, Parque Tecnológico de Boecillo, 205, 47151 Boecillo, Valladolid, rubalo@cartif.es

Equipo de fisioterapeutas y terapeutas ocupacionales de la unidad de daño cerebral del Hospital Beata María Ana de Jesús, Dra. Paulina Oliva Navarrete
Medicina Física y Rehabilitación, Hospital Beata María Ana de Jesús, C/ Dr. Esquerdo 83, 28007 Madrid

Resumen

Los pacientes de Accidente Cerebro-Vascular (ACV) tienden a sufrir limitaciones en su movilidad. Al margen de que afecte a los miembros superiores o a los inferiores, esta pérdida de movilidad o de coordinación se traduce directamente en una disminución en su independencia y calidad de vida. Si a estos pacientes se les inserta rápidamente en un programa de neuro-rehabilitación pueden recuperar algunas de las capacidades perdidas que tenían antes del incidente y así mejorar su calidad de vida. En este trabajo presentamos los aspectos de diseño de una plataforma robotizada para el soporte al tratamiento en tareas de neuro-rehabilitación de miembros superiores en sujetos víctimas de ACV. Asimismo incluimos los resultados de un primer estudio clínico realizado con el objetivo de evaluar la efectividad de las terapias asistidas por un sistema háptico robotizado cuando se aplica a estos pacientes. A pesar de la falta de un grupo de control, se ha detectado una mejora en la movilidad de los pacientes tanto de forma cuantitativa como cualitativa.

Palabras Clave: Neuro-rehabilitación, interacción hombre-robot, robótica asistencial, realimentación háptica.

1 INTRODUCCIÓN

El uso de dispositivos robotizados para rehabilitación ha demostrado ser una excelente herramienta que ayuda en la administración de terapia física a los pacientes que han sufrido enfermedades y daños neurológicos [7]. El objetivo de los algoritmos de control en terapias robotizadas es controlar los dispositivos robotizados diseñados para los ejercicios de rehabilitación de tal modo que el ejercicio seleccionado para su ejecución por el participante provoque plasticidad motora y por tanto mejore la recuperación motora. Sin embargo en la actualidad aún no se dispone de un conocimiento científico sólido de cómo puede alcanzarse este objetivo. Los algoritmos de control de terapias robotizadas se han diseñado por tanto sobre una base ad hoc, utilizando habitualmente conceptos de otros campos como son la rehabilitación, neurociencias y aprendizaje motor.

Existen múltiples ejemplos de plataformas modernas de rehabilitación [3] [5] [9] [11], muchas de ellas incluyen interfaces multimodales mejorados con entornos interactivos de realidad virtual, robots hápticos de altas prestaciones, exoesqueletos y otros componentes sofisticados. En la División de Ingeniería Biomédica de la Fundación CARTIF llevamos una década trabajando en este campo y nos hemos centrado en el desarrollo de plataformas robotizadas para ayuda en terapias de rehabilitación de miembros superiores.

El uso de tecnologías del ámbito de la robótica en neuro-rehabilitación ha sido sujeto de un gran desarrollo en los últimos 20 años [4]. El uso de sistemas robóticos en este campo permite el desarrollo de terapias interactivas más intensivas, con un mayor grado de control y orientadas a tareas.

Los resultados de estas terapias siguen sin ser concluyentes. A lo largo de los últimos años se han publicado numerosos estudios para demostrar la efectividad de la terapia asistida por robots en pacientes con daño cerebral traumático [1] [2]. La

mayor parte de ellos [8] muestran que los pacientes que reciben entrenamiento electromecánico mediante asistencia robotizada, aunque en general no muestran mejores capacidades para mejorar los movimientos de sus miembros superiores en la mayor parte de las actividades de la vida diaria, sí muestran una mejora significativa en la función motora.

En una revisión reciente [6] se llega a la conclusión de que las terapias robotizadas pueden ayudar a la recuperación de la función motora en el extremo proximal del miembro superior lo cual, a su vez, supone una mejora en el control motor haciendo que los resultados sean mejores que solamente con la aplicación de terapias tradicionales. Otro resultado interesante es que esta mejora puede observarse incluso en pacientes crónicos.

En este trabajo presentamos los aspectos más significativos de la evolución de la plataforma PHYSIOBOT, desde una primera versión desarrollada en torno a un robot industrial comercial hasta la versión actual, cuyos aspectos mecánicos, electrónicos y de control han sido diseñados teniendo en cuenta la experiencia adquirida con la versión previa. Mostramos también algunos resultados preliminares obtenidos con un grupo de pacientes con daño cerebral traumático que avalan la utilidad de la plataforma y nos proporcionan la realimentación necesaria para incluir mejoras que lleven, en el futuro, a una nueva versión, más avanzada.

El método de evaluación incluye la evaluación motora y la escala Fugl-Meyer junto con una métrica a medida basada en la puntuación obtenida en un juego diseñado para su utilización como parte del entorno terapéutico.

2 METODOLOGÍA

La versión inicial de la plataforma PHYSIOBOT se desarrolló en torno a un robot industrial Staubli RX90, versión de sala blanca, dotado de un sensor de par/fuerza JR3 instalado en su muñeca. Aunque sus dimensiones y características operativas no hacen que sea el hardware más adecuado para utilizar como herramienta en tareas de rehabilitación, la experiencia previa que teníamos en su utilización en aplicaciones biomédicas (asistente en cirugía ortopédica) así como la relativa facilidad para modificar ciertos modos de funcionamiento de su controlador, nos llevaron a desarrollar con él la primera versión de PHYSIOBOT.

La figura 1 muestra un diagrama de bloques que representa esta plataforma. El robot, incorporando realimentación háptica, es utilizado como núcleo de la plataforma. Los diferentes tipos de datos manejados fluyen a través de una red IP que facilita su intercambio. Diferentes tipos de datos dinámicos y fisiológicos se registran bajo la forma de series temporales para el control en línea y el análisis fuera de línea. Tanto la información en bruto como la procesada se almacena en una base de datos relacional para su posterior análisis e investigación.

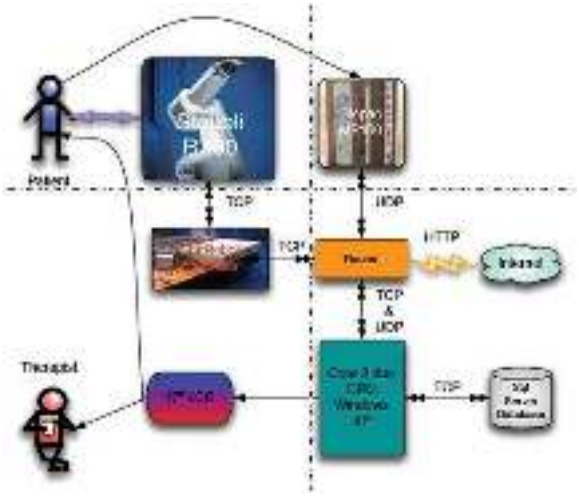


Figura 1. Diagrama de bloques de la plataforma PHYSIOBOT

Un sistema Biopac nos permite la captura, de modo ambulatorio y no invasivo, de señales fisiológicas (ritmo cardiaco, respuesta electrodérmica y temperatura superficial) con diferentes tasas de muestreo. Esta información permite evaluar el estado anímico-cognitivo del sujeto que está utilizando la plataforma y estimar la aparición de ciertos estados emocionales (interés, estrés, aburrimiento) de tal modo que se puede implementar una estrategia de control bio-cooperativa en la que el robot proporciona el grado de asistencia adecuado a la situación del sujeto (*assist as needed*)

Dadas las limitaciones de esta versión, y aprovechando la oportunidad de colaborar con la Unidad de Daño Cerebral del Hospital Beata María Ana de Jesús de Madrid, decidimos desarrollar una segunda versión manteniendo la arquitectura de la plataforma pero sustituyendo el robot Staubli por uno de diseño propio que cumpliese una serie de requisitos y especificaciones planteadas por los profesionales del centro sanitario, los cuales posteriormente se encargarían de la validación.

2.1 DISEÑO MECÁNICO Y CONTROL

El robot de esta segunda versión ha sido diseñado para ser un intrínsecamente seguro buscando esencialmente una dinámica transparente, reversibilidad (*back drivability*) y un coste reducido mediante la utilización de componentes comerciales

con un buen compromiso en el ratio prestaciones-precio.

La estructura mecánica diseñada se corresponde con la de un robot Gantry, similar al propuesto en [12], principalmente debido a las siguientes características intrínsecas:

- La similitud entre las mecánicas de ambos ejes permite obtener un tensor de inercia cuasi isotópico.
- La sencillez del diseño hace que tanto la cinemática directa como la inversa sean mucho más simples de calcular reduciendo por lo tanto los costes computacionales
- Las fuerzas debidas a la gravedad se compensan estructuralmente y por tanto no hay necesidad de incluir un modelo de compensación gravitatorio.

En la figura 2 se puede ver la plataforma siendo utilizada por un paciente durante la realización de una tarea de neuro-rehabilitación.



Figura 2. La plataforma PHYSIOBOT como soporte de tareas de neuro-rehabilitación.

Uno de los requisitos iniciales de diseño era mantener en niveles bajos la dinámica del robot. Este objetivo se pretendía alcanzar manteniendo el diseño mecánico lo más simple que fuera posible. Sin embargo, los componentes comerciales disponibles para la fabricación del robot (aquellos con un compromiso aceptable calidad-prestaciones-coste) estaban diseñados para su utilización en diseños mecánicos de alta rigidez, mediante engranajes y, en consecuencia, los fenómenos de fricción estática y dinámica en los raíles tenían niveles considerablemente altos.

Puesto que el principal objetivo del diseño era la seguridad intrínseca mediante la reversibilidad del conjunto accionamiento-engranajes, se eligieron motores de corriente continua sin escobillas maxon de altas prestaciones, complementados con una transmisión 4,3:1 para obtener una buena relación peso/par. Los efectos de las cajas de cambio de baja relación son altamente beneficiosos puesto que no se pierde la reversibilidad, se aumenta de forma significativa el par total proporcionado y también se pueden cancelar algunas no linealidades complejas de manejar como las discontinuidades de ficción superficial debidas al polvo o a la presencia de otras partículas sólidas en los raíles.

La plataforma contempla dos niveles de seguridad complementarios:

- Nivel de seguridad hardware: **PHYSIOBOT** incorpora seguridad redundante mediante hardware y software. Dispone de cuatro limitadores que cortan la corriente de alimentación a los motores cuando se alcanza la proximidad de los finales de carrera en cada raíl. Esto incrementa no sólo la seguridad sino que previene la degradación del hardware a lo largo del tiempo por posibles impactos.

- Nivel de seguridad software: junto con la seguridad hardware hemos implementado muros virtuales visco-elásticos próximos a los bordes de los límites del robot para prevenir impactos al mismo tiempo que se mantiene la compliancia.

2.2 CONTROL DE LA PLATAFORMA

Las estrategias de control háptico quedan determinadas la mayor parte de las veces por el tipo de hardware disponible. La disponibilidad o no de accionadores reversibles (*back drivables*) determina si puede aplicarse un control de impedancia o de admitancia. Por ejemplo, si se utilizan actuadores no reversibles, no es viable la implementación de un control de impedancia en lazo abierto.

Puesto que el hardware de nuestro robot es en principio reversible, es posible plantear una estrategia de control de impedancia. Sin embargo, debido a los altos coeficientes de fricción mecánica, viscosidad y fenómenos de Coulomb, hemos aplicado una solución basada en combinar un esquema de control de impedancia con realimentación de fuerza aumentado con un control basado en un modelo feed-forward. Esta compensación se calcula de forma experimental [10]. Ha sido necesario realizar una compensación feedforward porque el HW donde esta implementado el control tiene un ancho de banda bajo y no se podía recurrir a aumentar Kf (ganancia del error cometido entre la fuerza deseada y la real.)

Las variables y parámetros del sistema son los siguientes:

Variables a controlar:

- Fd: Fuerza deseada sobre el efector final.
- Xd: Posición deseada del efector final.

Variables de entrada:

- X: Posición del efector final.
- F: Fuerza ejercida por el paciente en el efector final.

Parámetros de control:

- M: Masa del sistema
- K: Constante de rigidez del sistema
- B: Constante de amortiguamiento del sistema
- Kf: Ganancia del error entre la fuerza deseada y la fuerza aplicada.

Los motores son controlados en corriente por un PID (control digital). El resultado puede verse en el diagrama de bloques de la figura 3, este esquema muestra un modelo de la planta que nos permite tratar con algunos de los fenómenos dinámicos indeseables que afectan a la transparencia del sistema.

La funcionalidad de los diferentes bloques del diagrama es la siguiente:

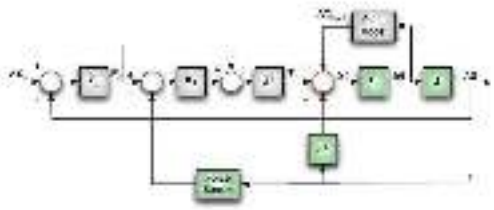


Figura 3. Diagrama de bloques del controlador

El bloque Zd es el operador de impedancia del sistema, el objetivo del mismo es dotar al sistema del las características de masa, amortiguamiento y rigidez deseadas.

$$M \cdot s + B + \frac{K}{s}$$

Siendo:

- M: componente Masa-inercial, describe como se resistirá el sistema a las aceleraciones y como almacenara la energía cinética.
- B: amortiguamiento, describe como el sistema se resiste a la velocidad y como disipa energía el sistema.
- K: rigidez, actúa de manera proporcional al error de posición y por lo tanto describe el nivel de asistencia/resistencia dado al paciente.

Bloque Kf: Este bloque solo contiene una ganancia Kf que multiplica al error entre la fuerza deseada y la fuerza producida. Valores altos de Kf hacen que el sistema converja rápidamente a la fuerza deseada. Esto hace que los errores debidos a la dinámica del dispositivo sean inversamente proporcionales a la ganancia Kf. Por lo tanto este valor ha de ser lo más alto posible, siempre con el compromiso de que no haga el sistema inestable.

$$K_f \cdot (F_d - F)$$

Bloque J: Es la matriz Jacobiana que incluye las relaciones entre las velocidades de movimiento de las articulaciones y las del extremo del robot.

Bloque Yh: Es la admitancia del dispositivo.

Bloque *Force Sensor*: Este bloque representa la función de transferencia del sensor de fuerza.

Bloque *Plant Model*: es un modelo de la fricción del sistema.

2.3 MODOS DE FUNCIONAMIENTO

PHYSIOBOT proporciona dos modos operativos: el primero es un modo pasivo en el cual el terapeuta mueve el extremo del robot (con el paciente agarrado) de tal forma que ejecuta la trayectoria que desea ser realizada tanto en el espacio como el tiempo, y esta puede ser almacenada para su posterior repetición. A continuación el terapeuta debe introducir el valor del nivel de asistencia, que es un porcentaje de la potencia total que el robot es capaz de suministrar, y el robot queda listo para repetir, durante las veces o el tiempo que se especifique, la trayectoria almacenada creando un túnel de fuerza visco-elástico en torno al recorrido.

El otro modo operativo es un modo de asistencia activa que proporciona una terapia orientada a tareas complementada con un algoritmo simple de asistencia a medida que modula la magnitud de la ayuda proporcionada por el robot, basado en las prestaciones de los sujetos y su capacidad de realizar las tareas.

Ambas versiones incluyen un entorno gráfico en el que pueden ejecutarse diferentes juegos diseñados para su utilización como herramientas terapeúticas, ya que los pacientes interactúan con el juego mediante el robot. En el más sencillo, un ratón virtual se desplaza aleatoriamente sobre la pantalla hasta que el paciente representado por una mano es capaz de alcanzarlo. Una vez que se ha capturado el ratón se incrementa la puntuación del juego y el ratón abandona su posición mediante un campo de repulsión virtual cuyo centro se sitúa en la mano del paciente.

3 RESULTADOS

Para la evaluación de esta segunda versión se diseñó en colaboración con los profesionales del hospital beata María Ana de Jesús de Madrid, un estudio longitudinal, pre-post, no controlado, de cara a evaluar la efectividad del sistema robotizado para el entrenamiento del tren superior en pacientes adultos con daño cerebral traumático de diferentes tipologías. Como grupo de estudio se seleccionaron ocho pacientes entre aquellos con daño cerebral traumático de la Unidad de Daño Cerebral del hospital, que cumpliesen los siguientes criterios de admisión:

- Sujetos con daño cerebral adquirido y limitaciones en los miembros superiores
- Tiempo transcurrido desde la lesión mayor de un mes
- Ashworth score <3
- Barthel score > 60
- Disminución de la movilidad articular en el hombro y el codo.
- Edad igual o superior a 18 años.

Evaluación: Cada paciente incluido en el estudio debió completar un tratamiento consistente en 10 sesiones de 30 minutos durante cinco días consecutivos a la semana. Cada sesión de tratamiento se organizó en una sesión asistida activa de 6 minutos, seguido por un período de reposo de 2 minutos hasta completar los 30 minutos totales de la práctica. Se consideró necesario llevar a cabo una sesión de evaluación previa para definir la configuración del sistema más adecuada para cada paciente (espacio de trabajo, postura ergonómica adecuada, etc).

Los resultados iniciales de la efectividad del estudio muestran como todos los pacientes tratados mediante terapia intensiva asistida por robot tuvieron un incremento significativo en las puntuaciones de los juegos a lo largo de las sesiones de entrenamiento como puede verse en la tabla 1 y figura 4.

Tabla 1. Escala de evaluación motora

Subject	Arm Pre	Arm Post	Hand Pre	Hand Post	Fine Pre	Fine Post
1	5	6	5	5	4	5
2	5	6	5	6	5	5
3	5	6	4	5	2	3
4	6	6	6	6	6	6
5	6	6	6	6	5	5
6	5	6	4	6	4	5
7	6	6	5	6	4	5
8	5	6	5	5	2	4

La escala de evaluación motora de la tabla 1 muestra una mejora en las puntuaciones para la función del miembro superior en los movimientos de la mano y movimientos finos.

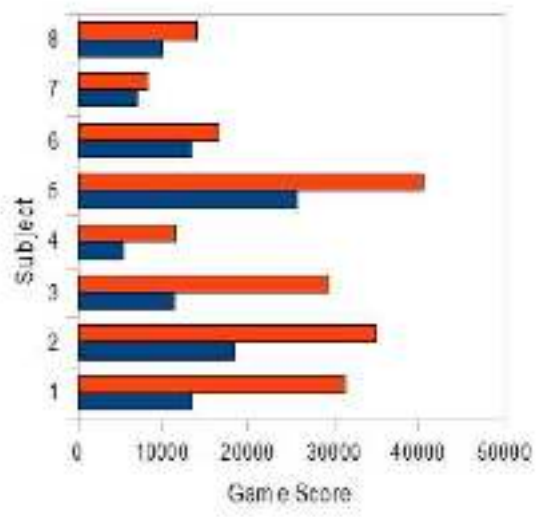


Figura 4. Puntuaciones globales para cada uno de los participantes (primera sesión en azul y última sesión en naranja)

Por último también observamos como las puntuaciones obtenidas en la escala Fugl- Meyer para los elementos de mano y muñeca aumentaban tras la realización de las sesiones de terapia asistida. Este hecho es muy significativo cuando se contempla al mismo tiempo que los resultados de la escala de valoración motora para la mano y muñeca, puesto que los movimientos que se programaron inicialmente en el desarrollo de los componentes software del sistema robotizado consideraban únicamente las articulaciones de hombro y codo y solo en menor medida las de mano y muñeca. Aún así la mejora funcional observada en la mano es comparable a los beneficios terapéuticos obtenidos en el nivel proximal del hombro y codo

Durante el desarrollo de las sesiones se observó que la terapia asistida por robots y el software utilizado constituyen una potente realimentación para el paciente que se le motivado alcanzar una puntuación superior en tres sesiones y aprecia positivamente los cambios que tiene lugar durante el tiempo de entrenamiento.

4 CONCLUSIONES

La versión actual de la plataforma PHYSIOBOT elimina muchas de las limitaciones de la versión inicial, aportando ventajas significativas desde los puntos de vista de seguridad, usabilidad y coste.

En la evaluación realizada hemos observado una mejora significativa en todos los pacientes que han realizado el programa de rehabilitación robótica intensiva descrito. Las puntuaciones en las escalas mejoraron sensiblemente y la apreciación subjetiva de los pacientes también fue positiva. Los sujetos percibieron de modo subjetivo un incremento en la habilidad funcional del miembro superior en el desarrollo de Actividades de la Vida Diaria (AVD) que muestra una correlación positiva con las mejoras objetivas en las escalas de valor funcional para los apartados de muñeca y mano.

La ausencia de un grupo de control y el pequeño tamaño de la población inciden en la necesidad de ensayos clínicos más elaborados que permitirán minimizar las desviaciones metodológicas que presenta este estudio preliminar

En la actualidad estamos trabajando en incorporar a la nueva versión la funcionalidad de captura ambulatoria de datos fisiológicos de los sujetos y adaptando los algoritmos de control bio-cooperativo para su ejecución en esta plataforma.

Referencias

[1] Hidler, J., Nichols, D., Pelliccio, M., & Brady, K. (2005). Advances in the understanding and treatment of stroke impairment using robotic devices. Top Stroke Rehabil, 12(2), 22-35.

[2] Hogan, N., Krebs, H. I., (2004). Interactive robots for neurorehabilitation., Restorative neurology and neuroscience, v. 22, pp. 349–58

[3] Krebs, H. I., Hogan, N., Aisen, M. L., & Volpe, B. T. (1998). Robot-aided neurorehabilitation. Rehabilitation Engineering, IEEE Transactions on, 6(1), 75-87.

[4] Kwakkel, G., van Peppen, R., Wagenaar, R. C., Dauphinee, S. W., Richards, C., Ashburn, A., ... & Langhorne, P. (2004). Effects of augmented exercise therapy time after stroke a meta-analysis. Stroke, 35(11), 2529-2539.

[5] Lum, P., Reinkensmeyer, D., Mahoney, R., Rymer, W. Z., & Burgar, C. (2002). Robotic devices for movement therapy after stroke: current status and challenges to clinical acceptance. Topics in stroke rehabilitation, 8(4), 40-53.

[6] Mehrholz, J., Platz, T., Kugler, J., & Pohl, M. (2009). Electromechanical and robot-assisted arm training for improving arm function and activities of daily living after stroke. Stroke, 40(5), e392-e393.

[7] Platz, T. (2003). Evidence-based arm rehabilitation--a systematic review of the literature]. Der Nervenarzt, 74(10), 841.

[8] Prange, G. B., Jannink, M. J., Groothuis-Oudshoorn, C. G., Hermens, H. J., & IJzerman, M. J. (2006). Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke. Journal of rehabilitation research and development, 43(2), 171.

[9] Reinkensmeyer, D. J., Kahn, L. E., Averbuch, M., McKenna-Cole, A., Schmit, B. D., & Rymer, W. Z. (2000). Understanding and treating arm movement impairment after chronic brain injury: progress with the ARM guide. Journal of rehabilitation research and development, 37(6), 653-662.

[10] Rodriguez-Guerrero, C., Ana, B. M., Navarrete, P. O., Fraile, J. C., & Farina, P. R. (2013). Preliminary Results from the Use of the SOFTROBOT Platform in Stroke Patients. In Converging Clinical and Engineering Research on Neurorehabilitation (pp. 215-226). Springer Berlin Heidelberg.

[11] Rodriguez Guerrero, C., Fraile Marinero, J., Perez Turiel, J., & Rivera Farina, P. (2010). Bio cooperative robotic platform for motor function recovery of the upper limb after stroke. In , 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) pp. 4472-4475

[12] Zollo, L., Accoto, D., Torchiani, F., Formica, D., & Guglielmelli, E. (2008). Design of a planar robotic machine for neuro-rehabilitation. In ICRA 2008. IEEE International Conference on Robotics and Automation, pp. 2031-2036

Sistema Robótico Planar para Neuro-rehabilitación: Primeros Resultados con pacientes

F. Badesa[1], R. Morales[1], A.Llinares[1], N. Garcia-Aracil[1], Maria Garcia-Manzanares[2], Daniel Tornero[2], M. Oliva[3]
[1]Neuroingenieria Biomedica
[2]Hospital Universitario de San Juan de Alicante
[3]Dept. de Ingeniería Mecáncia y Energía
Universidad Miguel Hernandez
email: fbadesa@umh.es

Resumen

En este artículo se presenta un nuevo sistema robótico para neuro-rehabilitación, enfocado a la rehabilitación de miembro superior para pacientes con hemiparesia debida a algún tipo de accidente cerebro-vascular. En la primera parte del artículo se presentan los requisitos que se han tenido en cuenta a la hora del diseño del robot, como el espacio de trabajo necesario a cubrir así como el estudio dinámico de la configuración adoptada. En la segunda parte del artículo se presenta el software de rehabilitación y realidad virtual que se ha desarrollado para tareas de neuro-rehabilitación asistida por el dispositivo robótico explicando las principales características del software, como el lenguaje de programación utilizado, librerías, etc. Al final de esta artículo, se presentan algunos resultados de un estudio clínico que se está llevando a cabo con el sistema robótico descrito.

Palabras clave: Robótica, rehabilitación, cinemática, dinámica

1 INTRODUCCIÓN

Según la Organización Mundial de la Salud, el número de personas mayores de 65 años aumentará en un 73% en los países industrializados, y en un 207% por todo el mundo. Para 2050, el porcentaje de la población europea mayor de 65 años casi se duplicará, pasando de 12,3% al 20,6% (de 40 a 80 millones). Este grupo de edad es particularmente propensa a sufrir algún tipo de accidente cerebrovascular (ACV). La incidencia relativa de ACV se duplica cada década para mayores de 55 años. De hecho, el ACV es la principal causa de incapacidad permanente en los países industrializados. Cada año, más de 920.000 europeos y 700.000 norteamericanos tienen un ACV; más de la mitad sobreviven, pero a menudo con discapacidades severas. Los principales síntomas son la pérdida de la fuerza muscular, espasticidad, y la falta de coordinación de la activación muscular [1]. El uso de dispositivos robóticos, como posible estrategia de rehabilitación para lograr la recuperación de la capacidad motora, se puede justificar debido a su potencial impacto sobre la mejora del tratamiento terapéutico y el aprendizaje motor. La evidencia indica que las terapias de realización repetitiva de tareas específicas asistidas por robots pueden ser más eficaces para la reducción a largo plazo de las alteraciones motrices en un paciente afectado por un ACV, tanto en términos de fuerza y estado motor como en lo relativo a la reducción de la espasticidad y el tono musculares [2, 3, 4, 5, 6] . Por otra parte, los tratamientos asistidos por dispositivos mecánicos pueden ofrecer mediciones objetivas del rendimiento de los pacientes que son útiles y fácilmente analizables por los clínicos y los fisioterapeutas. Recientemente, la American Heart Association ha publicado una serie de recomendaciones para el manejo interdisciplinario de la rehabilitación posterior al ACV basadas en las evidencias científicas disponibles hasta la fecha, incluyendo dentro de ellas la terapia asistida con robot para el miembro superior[7]. Sus recomendaciones y niveles de evidencia para cuestiones de tratamiento motor utilizando terapia asistida por robots son Clase I, Nivel de Evidencia A para atención de ACV en pacientes no hospitalizados y pacientes crónicos, y Clase IIa, Nivel de Evidencia A para la atención de ACV en pacientes hospitalizados [7].

En el présente artículo se présenta el diseño y desarrollo de un robot planar accionado por aire comprimido para la neuro-rehabilitación del miembro superior de pacientes afectados de un ACV. Al final del artículo se proporcionan los primeros resultados de un estudio clínico que se está realizando con el sistema robótico descrito en este artículo.

2 DISEÑO Y ESPACIO DE TRABAJO

El primero de los requisitos que hay que cumplir para el diseño del robot, es poder cubrir la mayor parte de los movimientos del miembro superior. Para ello, hay que definir el espacio de trabajo, o amplitud de movimiento, que se puede alcanzar al realizar distintos movimientos del brazo.

Para el análisis de la amplitud de movimiento del

brazo humano se ha hecho uso de un modelo simplificado de 6 g.d.l., 5 g.d.l. en el hombro y 1 g.d.l. en el codo. Los tres grados de libertad de la muñeca se obvian para simplificar el modelo, ya que no es necesaria la información de orientación de la mano. El modelo del brazo humano está compuesto por tres eslabones rígidos, el eslabón del hombro que representa la clavícula y escápula, el eslabón del brazo que representa el húmero y por último el eslabón del antebrazo que representa el cúbito y radio.

Debido a las propiedades anatómicas de los huesos y músculos del brazo, los límites de algunas articulaciones definidas en el modelo del brazo dependen de los valores de otras articulaciones. En concreto, los límites de la flexión-extensión (articulación q_4) se puede escribir como una función lineal de la abducción-aducción (q_3); y los límites de la rotación (q_5) se puede escribir como función lineal de los ángulos de abducción-aducción (q_3) y de la flexión-extensión (q_4). Los límites del resto de articulaciones se consideran constantes e independientes. En la *Tabla 1* se muestran los valores numéricos, tanto de las longitudes de los eslabones como de los límites de las articulaciones, utilizados para el cálculo del espacio de trabajo del modelo del brazo humano.

Eslabón	Longitud	G.D.L.	Límite superior	Límite inferior
Clavícula-escápula	0.15 m	q_1	-17°	17°
		q_2	-6°	20°
Húmero	0.31 m	q_3	-9°	160°
		q_4	$-43^\circ + q_3/3$	$153^\circ - q_3/6$
		q_5	$-90^\circ + 7q_3/9 - q_4/9 + 2q_3q_4/810$	$60^\circ + 4q_3/9 - 5q_4/9 + 5q_3q_4/810$
Cúbito-radio	0.27 m	q_6	-90°	60°

Tabla 1: Valores numéricos del modelo del brazo humano.

Para definir la amplitud de movimiento del miembro superior, se ha hecho uso de un sistema óptico de captura de movimiento proporcionado por NaturalPoint. El sistema utilizado se compone de 12 cámaras V100:R2 de OptiTrack, un traje con 34 puntos reflectantes de infrarrojo y el sistema software ARENA de NaturalPoint. El resultado de estos dos análisis es una elipse de eje mayor 1200mm y de eje menor 600mm que nos define el espacio de trabajo del brazo humano en tareas que se realicen en un plano horizontal. Como ya se ha definido, el sistema robótico va a restringir su movimiento a un plano horizontal, con lo que un sistema robótico de 2GDL es suficiente. La configuración idónea de robot seleccionada es el paralelogramo similar a la del MIT Manus [8]. Con esta configuración se obtiene una estructura más robusta y ligera que en el caso del robot serial y se tiene la posibilidad de colocar los actuadores sobre el eje de la base, con lo que se tendrían anclados a una parte inmóvil del robot (Figura 1).

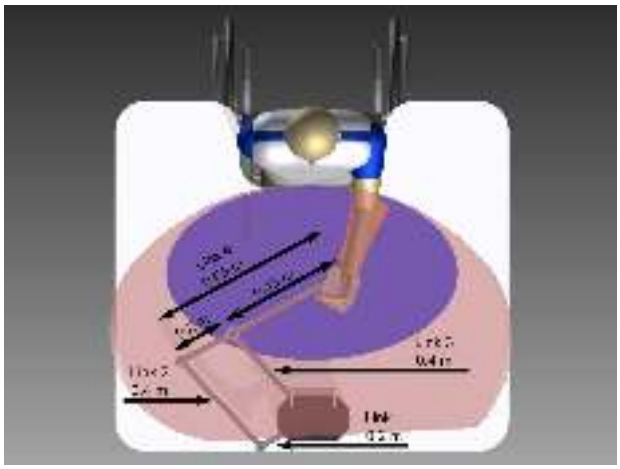


Figura 1: Espacio de Trabajo: Brazo humano (azul) y Robot (rojo)

Otro requisito adicional para el diseño del robot planar se extrae del análisis dinámico del mismo, ya que de una elección adecuada de la longitud de los eslabones (*links*) se puede obtener una configuración independiente y una matriz de inercia desacoplada. Estas características facilitan el control del robot y la planificación de trayectorias.

Es bien sabido que el modelo dinámico de un robot viene definido por la expresión:

$$\tau = B(\ddot{q}) + C(q, \dot{q})\dot{q} + g(q) \qquad (1)$$

Siendo B la matriz de inercias, donde las componentes b_{ii} corresponde al momento de inercia de la articulación i en la configuración actual del robot, cuando las demás articulaciones están bloqueadas; y las componentes b_{ij} corresponde al efecto de la aceleración de la articulación j sobre la articulación i. La matriz C contiene valores sobre el efecto centrífugo y el efecto de Coriolis inducido por una articulación sobre otra. Y finalmente, el vector $g(q)$ es el término debido a la fuerza de la gravedad, en la posición actual del robot, para cada articulación.

Para el cálculo del modelo dinámico del sistema robótico diseñado, se comienza por calcular la matriz de inercias B, definida en la *Ecuación 2.*

$$B(q) = \sum_{i=1}^{n} (m_{l_i} J_P^{(l_i)T} J_P^{(l_i)} + J_O^{(l_i)T} R_i I_{l_i}^i R_i^T J_O^{(l_i)} + m_{m_i} J_P^{(m_i)T} J_P^{(m_i)} + J_O^{(m_i)T} R_i I_{m_i}^{m_i} R_i^T J_O^{(m_i)})$$

En la *Figura 2* se muestra de forma esquemática

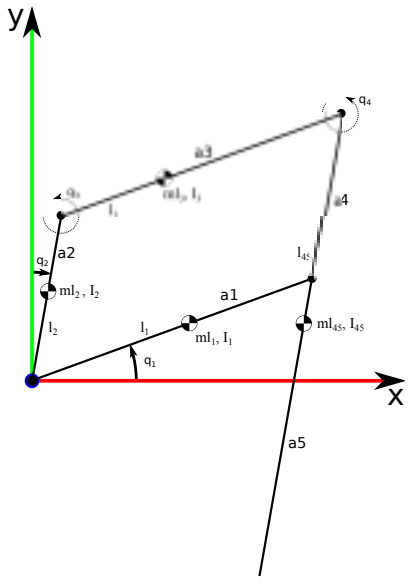


Figura 2: Brazo paralelogramo

el robot diseñado. Debido a la presencia de la cadena cerrada, se tiene en cuenta la estructura en árbol de cadena abierta equivalente. Se coge l_1, l_2, l_3 y l_{45} como las distancias entre los centros de masa de los cuatro eslabones a lo largo de las ramas del árbol desde los ejes de sus respectivas articulaciones. Del mismo modo, se coge m_{l_1}, m_{l_2}, m_{l_3} y $m_{l_{45}}$ como las masas de los respectivos eslabones, y I_{l_1}, I_{l_2}, I_{l_3} y $I_{l_{45}}$ como los momentos de inercia relativos a los centros de masa de sus respectivos eslabones. Para simplificar los cálculos, no se tienen en cuenta las contribuciones de los motores.

Teniendo en cuenta el sistema de coordenadas de la figura, se obtienen los siguientes resultados de las Jacobianas de las velocidades lineales de los centros de masa:

$$J_P^{l_2} = \begin{bmatrix} l_2\ C_2 & 0 & 0 \\ -l_2\ S_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$J_P^{l_3} = \begin{bmatrix} a_2\ C_2 - l_3\ C_{2\ 3} & l_3\ C_{2\ 3} & 0 \\ -a_2\ S_2 - l_3\ S_{2\ 3} & -l_3\ S_{2\ 3} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$J_P^{l_{45}} = \begin{bmatrix} a_2\ C_2 + a_3\ C_{2\ 3} + l_{45}\ S_{2\ 3\ 45} & a_3\ C_{2\ 3} + l_{45}\ C_{2\ 3\ 45} & l_{45}\ C_{2\ 3\ 45} \\ -a_2\ S_2 - a_3\ S_{2\ 3} - l_{45}\ S_{2\ 3\ 45} & -a_3\ S_{2\ 3} - l_{45}\ S_{2\ 3\ 45} & -l_{45}\ S_{2\ 3\ 45} \\ 0 & 0 & 0 \end{bmatrix}$$

$$J_P^{l_1} = \begin{bmatrix} -l_1\ S_1 \\ l_1\ C_1 \\ 0 \end{bmatrix}$$

Mientras, los resultados de las Jacobianas de las velocidades angulares vienen dadas por:

$$J_O^{l_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad J_O^{l_3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & -1 & 0 \end{bmatrix}$$

$$J_O^{l_{45}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad J_O^{l_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

La matriz de inercia del brazo virtual formado por las articulaciones $q2$, $q3$, $q4$ es:

$$B'_{q'} = \begin{bmatrix} b'_{11}(q3,q4) & b'_{12}(q3,q4) & b'_{13}(q3,q4) \\ b'_{21}(q3,q4) & b'_{22}(q4) & b'_{23}(q4) \\ b'_{31}(q3,q4) & b'_{31}(q4) & b'_{33} \end{bmatrix}$$

$$b'_{11} = I_{l_2} + I_{l_3} + I_{l_{45}} + m_{l_2} l_2^2 + m_{l_3}(a_2^2 + l_3^2 + 2a_2 l_3 C_3) + m_{l_{45}}(a_2^2 + a_3^2 + l_{45}^2 + 2a_2 a_3 C_3 + 2a_3 l_{45} C_4 + 2a_2 l_{45} C_{34})$$

$$b'_{12} = b'_{21} = I_{l_3} + I_{l_{45}} + m_{l_3}(l_3^2 + a_2 l_3 C_3) + m_{l_{45}}(a_3^2 l_{45}^2 a_2 a_3 C_3 + 2a_3 l_{45} C_4 + a_2 l_{45} C_{34})$$

$$b'_{13} = b'_{31} = I_{l_{45}} + m_{l_{45}}(l_{45}^2 + a_3 l_{45} C_4 + a_2 l_{45} C_{34})$$

$$b'_{22} = I_{l_3} + I_{l_{45}} + m_{l_3} l_3^2 + m_{l_{45}}(a_3^2 + l_{45}^2 + 2a_3 l_{45} C_4)$$

$$b'_{23} = b'_{32} = I_{l_{45}} + m_{l_{45}}(l_{45}^2 + a_3 l_{45} C_4)$$

$$b'_{33} = I_{l_{45}} + m_{l_{45}} l_{45}^2$$

Mientras que el momento de inercia del brazo virtual formado por $q1$ es:

$$b''_{11} = I_{l_1} + m_{l_1}\ l_1^2$$

Por lo tanto, las contribuciones de los pares de inercia de los dos brazos virtuales son respectivamente:

$$\tau'_i = \sum_{j=1}^{3} b'_{ij} \ddot{q}_j \quad \tau''_1 = b''_{11} \ddot{q}_1 \qquad (2)$$

En este punto, las contribuciones del par de inercia en las articulaciones actuadas del brazo de la cadena cerrada se puede calcular de la siguiente manera:

$$\begin{aligned} \dot{q}_o &= Y \cdot \dot{q}_a \\ \tau_a &= Y^T \cdot \tau_o \\ \tau_o &= B_o \cdot \ddot{q}_o \\ \ddot{q}_o &= \dot{Y} \cdot \dot{q}_a + Y \cdot \ddot{q}_a = Y \cdot \ddot{q}_a \\ \tau_a &= B_a \cdot \ddot{q}_a = Y^T \cdot B_o \cdot Y \cdot \ddot{q}_a \end{aligned} \tag{3}$$

Con los valores del sistema robótico diseñado y con los valores de B' y B'' calculados, se tiene la expresión de la matriz de inercias del sistema, mostrada en las siguientes expresiones:

$$B_a = Y^T \cdot B_o \cdot Y = \begin{bmatrix} b_{a11} & b_{a12} \\ b_{a21} & b_{a22} \end{bmatrix} \tag{4}$$

$$b_{a11} = I_{l_1} + I_{l_3} + m_{l_1} l_1^2 + m_{l_3} l_3^2 + m_{l_{45}} a_1^2 \tag{5}$$

$$\begin{aligned} b_{a12} = b_{a21} = (a_1 m_{l_{45}} (l_{45} - a_2) - \\ - a_2 l_3 m_{l_3}) sin(q_1 + q_2) \end{aligned} \tag{6}$$

$$\begin{aligned} b_{a22} = I_{l_2} + I_{l_{45}} + m_{l_2} l_2^2 + \\ + m_{l_3} a_2^2 + m_{l_{45}} (a_2^2 + l_{45}^2 - 2 a_2 l_{45}) \end{aligned} \tag{7}$$

La expresión *6* revela la posibilidad de obtener una configuración independiente y una matriz de inercia desacoplada. Para este fin, es suficiente diseñar los cuatro eslabones del paralelogramo de tal manera que cumplan la siguiente restricción:

$$\frac{m_{l_{45}} \bar{l}_{45}}{m_{l_3} l_3} = \frac{a_2}{a_1} \tag{8}$$

donde $\bar{l}_{45} = l_{45} - a_2$ es la distancia del centro de masa del eslabón $a_4 + a_5$ al eje de la articulación 4. Si se cumple esta condición, entonces la matriz de inercia es diagonal ($b_{a12} = b_{a21} = 0$) con:

$$b_{a11} = I_{l_1} + I_{l_3} + m_{l_1} l_1^2 + m_{l_3} l_3^2 \left(1 + \frac{a_1 a_2}{l_3 \bar{l}_{45}}\right) \tag{9}$$

$$b_{a22} = I_{l_2} + I_{l_{45}} + m_{l_2} l_2^2 + m_{l_3} a_2^2 \left(1 + \frac{l_3 \bar{l}_{45}}{a_1 a_2}\right) \tag{10}$$

Como consecuencia de este desacoplo dinámico, no hay contribuciones de Coriolis ni pares centrífugos y su puede implementar un control desacoplado de las articulaciones del robot.

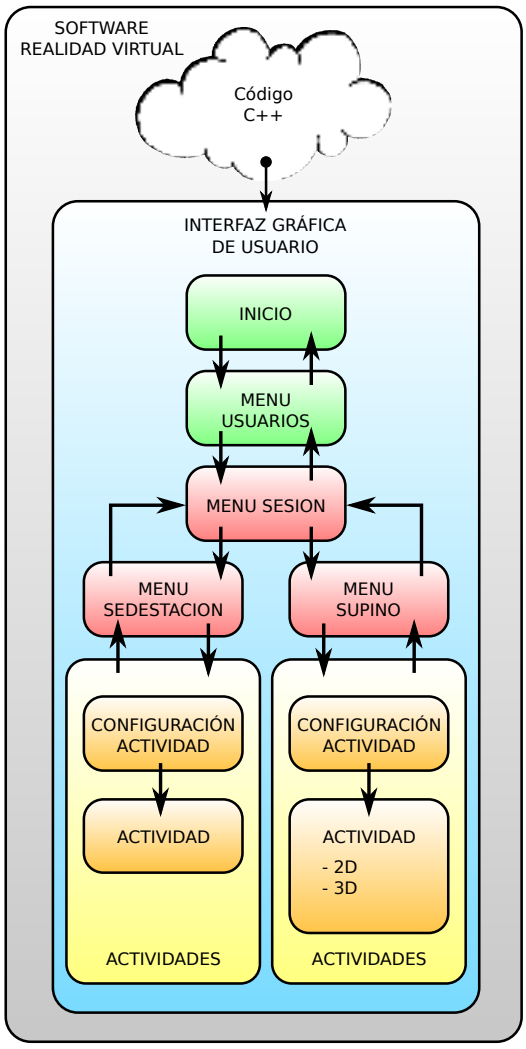


Figura 3: Esquema de la interfaz gráfica de usuario.

3 INTERFACE DE USUARIO

Una de las principales características que debe tener el software de rehabilitación y realidad virtual es su sencillez para que de forma intuitiva el terapeuta sea capaz de preparar la sesión de rehabilitación. Para ello, se piensa en realizar una interfaz gráfica de usuario sencilla y que el terapeuta pueda manejar mediante una pantalla táctil. Para la implementación de la interfaz gráfica, se ha hecho uso de las librerías Qt. Qt es un entorno de trabajo basado en C++ para el desarrollo de software en plataforma cruzada. Junto a una amplia librería en C++, Qt incluye las herramientas necesarias para escribir las aplicaciones de una forma muy rápida y sencilla. Junto con las versiones comerciales se distribuyen otras versiones Open Source integradas en KDE. Estas librerías incluyen un conjunto de widgets (controles, en la terminología de windows) encargados de proporcionar las funciones estándar de un interfaz o

GUI. Qt presenta una alternativa para la comunicación inter-objeto denominada "signals y slots" que remplaza al antiguo sistema de callbacks utilizado por otros sistemas además de proporcionar un modelo de eventos convencional para la captura de las pulsaciones del ratón, teclas u otras entradas del usuario haciendo posible la creación de aplicaciones multi-plataforma mediante funciones estándar, incluyendo drivers para Linux, Windows, etc., e incluso drivers nativos para bases de datos como Oracle ®, Microsoft® SQL Server, IBM DB2®, Borland.

En la *Figura 3* se muestra un esquema de la interfaz gráfica de usuario desarrollada. En este esquema se observan tres partes diferenciadas dentro del bloque de la interfaz gráfica de usuario.

La primera parte, los bloques de color verde en la *Figura 3*, son las pantallas iniciales del software. La primera es una pantalla de seguridad en la que hay que introducir una contraseña para poder iniciar el software de rehabilitación. La segunda pantalla, es el menú de usuarios, donde se pueden añadir, modificar o eliminar usuarios y sus datos. En esta pantalla se selecciona el usuario que va a realizar la sesión de rehabilitación. En la *Figura 4* se muestra una captura de la pantalla del menú de usuarios.

La segunda parte del software, los bloques de color rojo en la *Figura 3*, se compone de las pantallas de configuración de la sesión. En la pantalla de menú de sesión, se muestra la información de las sesiones anteriores del usuario seleccionado y se da a elegir si se desea realizar actividades en configuración de sedestación o de decúbito supino. Las otras dos pantallas que componen esta parte del software son las pantallas de selección de las actividades.

Por último, los bloques de color amarillo, están compuestos por las actividades y los menús de configuración de los parámetros modificables de la actividad seleccionada.



Figura 4: Pantalla de Menú de Usuarios.

Las actividades parametrizables para la posición del usuario en sedestación se pueden dividir fundamentalmente en dos tipos distintos, unas basadas en movimientos sencillos con realimentación visual en 2D; y otras basadas en movimientos y actividades de la vida diaria (AVD), con entornos 3D más realistas.

Para el robot planar presentado en esté artículo se han diseñado distintas actividades basadas en entornos sencillos en dos dimensiones no realistas en los que aparece algún tipo de imagen como puntero que maneja el usuario del dispositivo robótico, como puede ser un coche que se tiene que mover en línea recta por una carretera, recolectar manzanas de un árbol, etc (Figura 5). Para ello se hace uso de las mismas librerías que para el entorno gráfico, las librerías Qt.



Figura 5: Ejemplo de actividades 2D.

4 ESTUDIO CLÍNICO

Se está llevando a cabo un estudio experimental prospectivo y aleatorizado que mide los efectos de implementar esta tecnología robótica en el tratamiento de personas con hemiplejía secundaria a un ACV.

Durante los tres meses del estudio el grupo a estudio recibirá el tratamiento de fisioterapia aceptado y validado hasta el momento, con la frecuencia según el procedimiento habitual en el Servicio Nacional de Salud: 3 sesiones semanales de aproximadamente 30 minutos, lo que contabiliza un total de 36 sesiones al cabo de 3 meses. Además, recibirán el mismo número de sesiones de terapia asistida por el robot: 3 sesiones semanales de 45 minutos, un total de 36 sesiones. Las sesiones con el dispositivo robótico se realizarán antes o después de la fisioterapia convencional, según disponibilidad de la persona. En la primera sesión se llevará a cabo una valoración para fijar los parámetros que posteriormente se utilizarán en las tareas: altura de la pantalla y de la silla, rango de movimiento. Las tareas tendrán como objetivo la movilización del miembro superior(MS) parético siempre intentando alcanzar el máximo

rango articular funcional de la persona.

El tiempo de la sesión se organiza de la siguiente manera:

- 20 repeticiones de una actividad 2D (movimientos globales tanto de hombro como de codo), que se calcula que serán completadas en aproximadamente 2-3 minutos
- 3 minutos de descanso
- 20 repeticiones de de una actividad 2D
- 5 minutos de descanso
- 20 repeticiones de una actividad concreta cuya realización trabaja flexo extensión de codo y hombro específicamente
- 3 minutos de descanso
- 20 repeticiones de una actividad concreta cuya realización trabaja abducción aducción de hombro específicamente
- 5 minutos de descanso
- 10 repeticiones de una actividad que realiza movimiento global del MS (sinusoide)
- 10 minutos de descanso

Una de las actividades 2D consistía en mover el brazo robótico desde el punto central (que contaba con 1 cm de radio) hasta alcanzar el punto iluminado de la periferia (también con 1 cm de radio). En total eran 8 puntos periféricos distribuidos uniformemente en la circunferencia de un círculo y distantes hasta un máximo de 250 mm del punto central. El sujeto recibía retroalimentación visual de la posición de su mano mediante un pequeño punto blanco de 4 mm de radio. Cada uno de los puntos periféricos se iluminaba de manera aleatoria completando el número total de repeticiones indicado por el fisioterapeuta.

Para ilustrar los primeros resultados del estudio, se muestra en la figura 6 el progreso de las trayectorias realizadas por un sujeto durante las 36 sesiones. Se puede apreciar que la distancia entre el punto central y los puntos de la periferia se incrementan desde 50mm hasta 200mm entre la sesión 1 y la sesión 11 y el nivel de asistencia proporcionado por el robot se redujo de un nivel máximo de asistencia en la sesión 1 hasta un nivel nulo de asistencia en la sesión 30. Los resultados evidencian una mejora del paciente en la realización de la actividad que se corrobora con una mejora en los resultados de la evaluación clínica.

5 CONCLUSIONES

En este artículo se presenta el desarrollo de un sistema robótico planar para neuro-rehabilitación del miembro superior. Cabe destacar que se está llevando a cabo un estudio clínico que mide los efectos de implementar esta tecnología robótica en el tratamiento de personas con hemiplejía secundaria a un ACV. Los primeros resultados obtenidos son prometedores y se presentan en este artículo la evolución de uno de los sujetos del estudio.

6 Agradecimientos

Este trabajo has sido financiado por el Plan Nacional de I+D+i a través del proyecto "Interpretación de la intención y actuación humana mediante señales biomédicas y el análisis cinemático y dinámico del movimiento" (DPI2011-29660-C04-04).

Referencias

[1] R. van Peppen, G. Kwakkel, B. H. van der Wel, B. Kollen, J. Hobbelen, J. Buur- ke, J. Halfens, L. Wagenborg, M. Vogel, M. Berns, R. van Klaveren, H. Hendriks, and J. Dekker, ?KNGF clinical practice guideline for physical therapy in patients with stroke. Review of the evidence,? Nederlands Tijdschrift voor Fysiotherapie, vol. 114, no.5, 2004.

[2] Aisen, M.L., Krebs, H.I., Hogan, N., McDowell, F. and Volpe, B.T., The effect of robot-assisted therapy and rehabilitative training on motor recovery following stroke, Arch Neurol, 54 (1997) 443-6.

[3] H.I. Krebs, J.J. Palazzolo, L. Dipietro, M. Ferraro, J. Krol, K. Rannekleiv, B.T. Volpe, N. Hogan, "Rehabilitation Robotics: Performance-Based Progressive Robot-Assisted Therapy", Autonomous Robots, vol. 15, pp. 7–20, 2003.

[4] Krebs HI, Volpe BT, Aisen ML, Hogan N. Increasing productivity and quality of care: Robot-aided neurorehabilitation. J Rehabil-Res Dev. 2000;37: 639-52.

[5] Volpe BT, Krebs HI, Hogan N, Edelsteinn L, Diels CM, Aisen ML: Robot training enhanced motor outcome in patients with stroke maintained over 3 years. Neurology 1999, 53:1874-1876.

[6] Volpe BT, Krebs HI, Hogan N, Edelstein OL, Diels C, Aisen M: A novel approach

to stroke rehabilitation: robot-aided sensorimotor stimulation. Neurology 2000, 54:1938-1944.

[7] E. L. Miller, L. Murray, L. Richards, R. D. Zorowitz, T. Bakas, P. Clark, S. A. Billinger, and American Heart Association Council on Cardiovascular Nursing and the Stroke Council, ?Comprehensive overview of nursing and interdiscipli- nary rehabilitation care of the stroke patient: a scientific statement from the American Heart Association.,? Stroke; a journal of cerebral circulation, vol. 41, pp. 2402?2448, Oct. 2010.

[8] Krebs, H.I.; Hogan, N.; Aisen, M.L.; Volpe, B.T.; , ?Robot-aided neurorehabilitation,? Rehabilitation En- gineering, IEEE Transactions on , vol.6, no.1, pp.75-87, Mar 1998

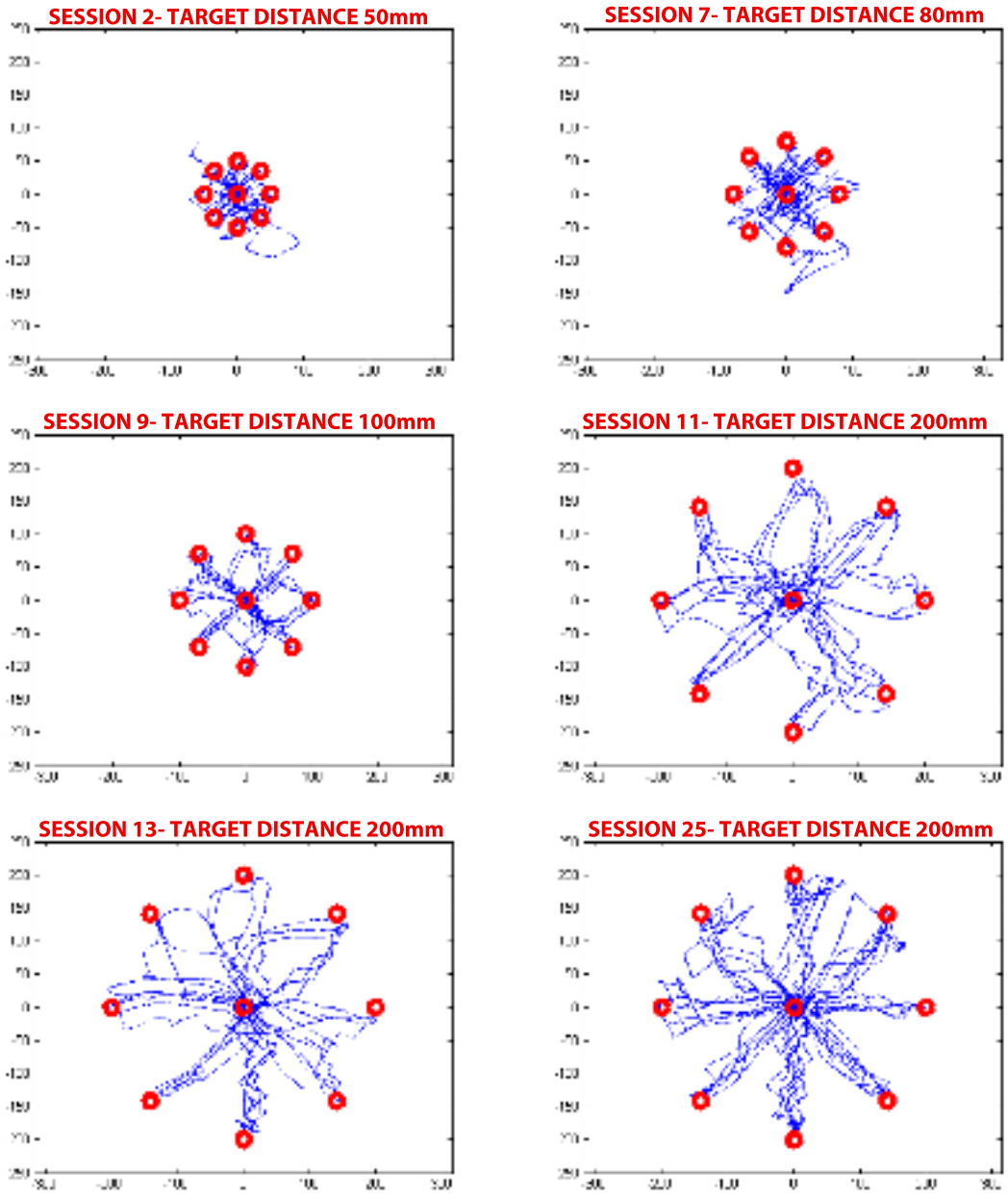


Figura 6: Resultados del estudio clínico para un sujeto. Se representa aleatoriamente una muestra de las trayectorias realizadas en las distintas sesiones

CLASIFICACIÓN DE SEÑALES DE POTENCIAL DE ERROR A TRAVÉS DE UNA INTERFAZ GRÁFICA CON REALIMENTACIÓN DE FUERZAS

Joaquín López, Andrés Úbeda, Eduardo Iáñez, D. Planelles, José M. Azorín, Javier Gimeno, José A. Flores
Universidad Miguel Hernández de Elche
Avda. de la Universidad s/n, 03202 Elche (Alicante)
eianez@umh.es

José M. Climent
Hospital General Universitario de Alicante
Pintor Baeza S/N, 03010 Alicante

Resumen

Este artículo muestra los resultados obtenidos en una interfaz gráfica con realimentación de fuerzas desarrollada para la detección de señales de error, ErrP. El objetivo principal de este trabajo es demostrar la efectividad de esta metodología para que pueda ser empleada en el futuro en la rehabilitación de usuarios que hayan sufrido un ictus. Para ello, se ha diseñado una interfaz gráfica que muestra diferentes objetivos que los usuarios deben alcanzar con un robot de escritorio (Phantom). Las señales electroencefalográficas y los movimientos de los usuarios se registran simultáneamente durante el desarrollo de las pruebas. Eventualmente, se produce un error y el robot de escritorio modifica el movimiento del brazo del usuario. Cuando el usuario percibe el error,se produce un potencial cerebral sobre el electrodo FCz que puede ser detectado. Para clasificar la aparición de este potencial se ha empleado un clasificador basado en la correlación cruzada normalizada. Los resultados obtenidos sugieren que es posible detectar de forma efectiva este tipo de potencial con la metodología propuesta.

Palabras Clave: Potencial de Error, Realimentación de Fuerzas, Interfaz Cerebro-Computador, No Invasivo, Rehabilitación

1 INTRODUCCIÓN

En la actualidad existe un gran número de personas con algún tipo de discapacidad motora. Estas personas experimentan disminuciones funcionales en alguno de sus miembros que dificultan su vida cotidiana, lo que ha originado que se hayan desarrollado numerosos sistemas de ayuda destinados a mejorar su nivel de vida, permitiendo una mayor independencia junto con mayores oportunidades de integración en la sociedad.

Un ejemplo de esos sistemas son las interfaces cerebrales (BCI) que permiten generar comandos de control con la única ayuda del pensamiento [11]. Para ello, se registra la actividad del cerebro mediante electrodos. Este tipo de interfaces ya han sido utilizadas con frecuencia en la asistencia de personas con discapacidad [5, 6, 7, 12].

Una de las causas más habituales de dificultad motora es el accidente cerebro-vascular (ACV) o ictus. Los pacientes aquejados de este problema pueden perder parcialmente la movilidad en miembros superiores, lo que les impide realizar tareas cotidianas de diversa índole. Para solventar esta problemática, una buena alternativa es la asistencia del movimiento del brazo a partir de interfaces cerebrales no invasivas, donde la señal se mide a través de electrodos superficiales colocados sobre el cuero cabelludo.

Existen ciertos potenciales que es posible detectar a partir de una interfaz de este tipo. El cerebro reacciona ante ciertos estímulos externos que se producen de forma síncrona. Por ejemplo, los potenciales visuales han sido ampliamente empleados en la literatura reciente [2, 7, 12].

Este trabajo se centra en el uso de un potencial particular conocido como potencial de error (ErrP). Un ErrP es un tipo de potencial que se genera en la región frontal del córtex cuando se percibe que algo no ha funcionado como el usuario interpreta que debe funcionar. La utilización del potencial de error se ha empleado en algunos trabajos relacionados con BCI [3, 4, 9].

El objetivo principal del trabajo es diseñar una interfaz multimodal para la clasificación de este

tipo de potenciales mediante la utilización de un clasificador basado en una correlación cruzada normalizada de señales. En el futuro, se pretende utilizar este tipo de potencial para detectar la intención de movimiento del miembro superior, o como mecanismo de apoyo de esa detección, en usuarios con movilidad reducida debido a un accidente cerebro-vascular.

Para detectar el potencial de error, se ha diseñado una interfaz visual que muestra tres objetivos a los que los usuarios deben llegar moviendo un robot de escritorio (Phantom) con la ayuda de la mano. Eventualmente, los usuarios son apartados del objetivo por una fuerza generada por el dispositivo, lo que genera una percepción de error que puede ser detectada por la interfaz cerebral.

El resto del artículo está dividido de la siguiente manera. En la sección 2 se describe el potencial de error. La metodología empleada para clasificar el potencial de error se presenta en la sección 3. La sección 4 muestra los resultados obtenidos durante las pruebas experimentales. Por último, la sección 5 contiene las conclusiones.

2 POTENCIAL DE ERROR

Un potencial de error (ErrP) es un potencial exógeno que se genera cada vez que un usuario percibe un error, o en otras palabras, cada vez que el sistema se comporta de manera distinta a la esperada.

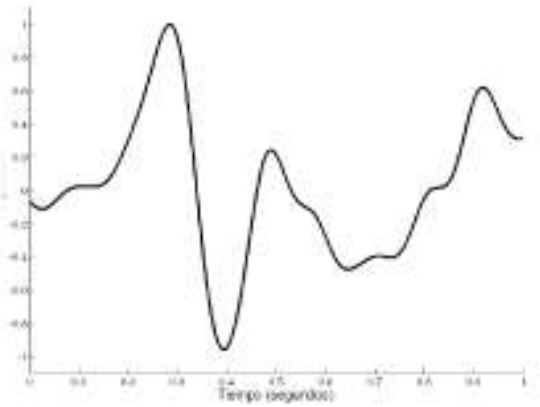


Figura 1: Ejemplo de señal de potencial de error sobre el electrodo FCz.

El potencial de error se genera en la zona frontal-central del córtex (corteza cingulada anterior), sobre el electrodo FCz. Consiste en una deflexión negativa que ocurre entre 50 y 100 ms después de la respuesta errónea. No obstante, es un fenómeno de gran variabilidad entre usuarios y, por tanto, su estudio debe estar sujeto a un modelado particular para cada uno de ellos. En la Figura 1 puede verse un ejemplo de una señal de potencial de error registrada sobre el electrodo FCz.

3 METODOLOGÍA

El registro de señales cerebrales se ha realizado en un entorno BCI como el mostrado en la Figura 2. El usuario se sitúa frente a la interfaz gráfica

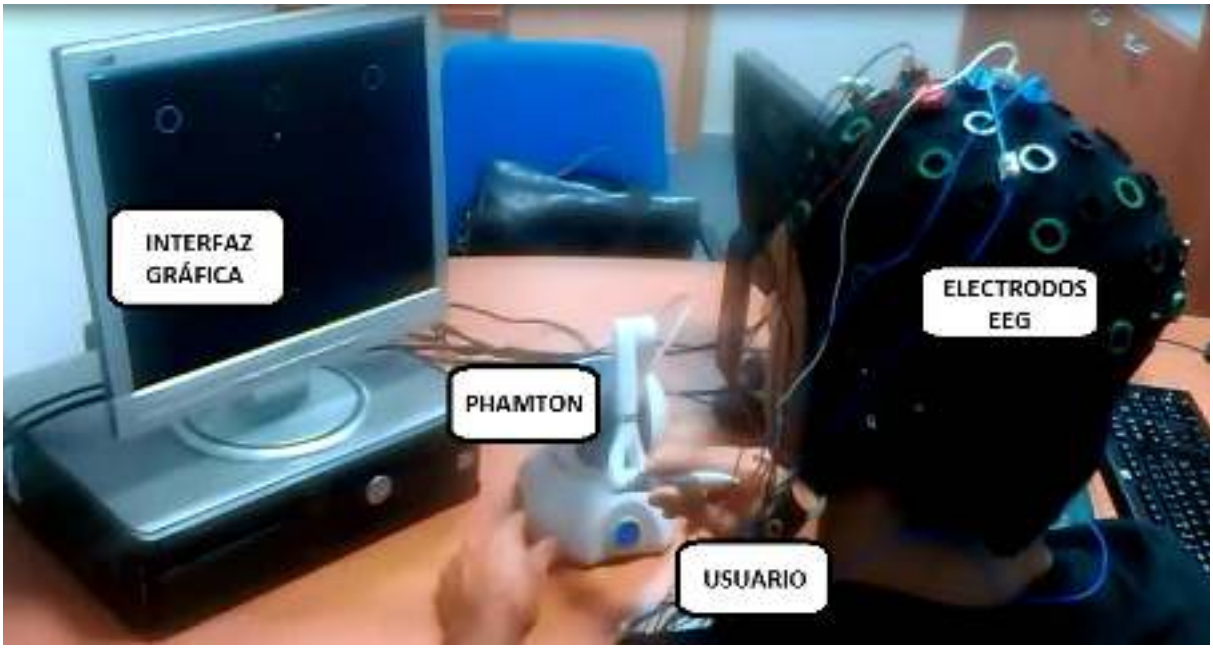


Figura 2: Entorno de trabajo EEG



Figura 3: Esquema de la arquitectura del sistema empleada.

(monitor).El amplificador recoge las señales cerebrales medidas con el gorro situado en la cabeza del usuario y se comunica con el PC vía USB. El usuario interactúa con la interfaz a través de un dispositivo de escritorio (Phantom). A través de este dispositivo, es capaz de mover un cursor en la pantalla para alcanzar los distintos objetivos que se presentan.

La Figura 3 muestra el esquema de la arquitectura empleada. El usuario percibe el error a través de la interfaz gráfica diseñada para tal efecto y que se explicará en los siguientes apartados. Simultáneamente, se registran las señales EEG del usuario y se procesan. La interfaz gráfica registra también en qué momentos se ha mostrado un error en la pantalla para poder sincronizar este error con los datos registrados del usuario. Finalmente, se obtienen los resultados del clasificador para verificar el funcionamiento del sistema.

3.1 REGISTRO Y PROCESAMIENTO

Para registrar las señales generadas por el ErrP se ha utilizado el equipo de registro g.USBamp de g.tec. Puesto que la señal ErrP se genera sobre la posición FCz (Sistema Internacional 10/20 [1]) se ha situado un electrodo en dicha posición. Además, para mejorar la calidad de la señal de ErrP durante el preprocesamiento, se ha recogido la señal de los 4e lectrodos adyacentes en las posiciones Fz, FC1, FC2 y Cz. En la Figura 4 puede verse dicha selección de electrodos.

El preprocesamiento consta de tres fases:

- En primer lugar se ha aplicado un filtro de media de referencia común (CAR: common average reference), para eliminar del electrodo principal (FCz) la contribución de señal de los electrodos adyacentes y potenciar la señal con la que se trabaja [10].
- A continuación, se ha aplicado un filtro paso banda entre 1 y 10 Hz.
- Finalmente, se ha realizado una normalización de la señal. Esto permitirá una mejor clasificación del potencial de error.

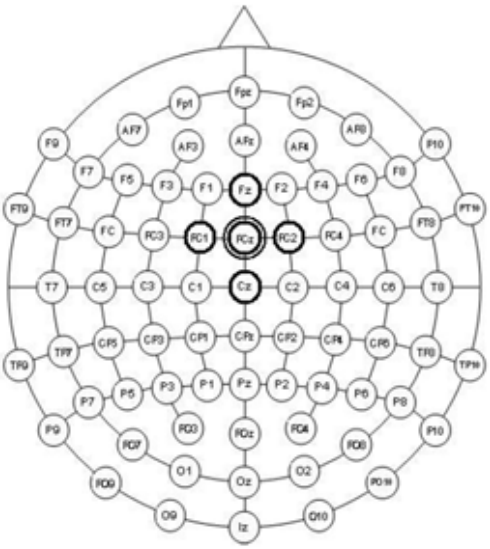


Figura 4: Selección de electrodos según el Sistema Internacional 10/20

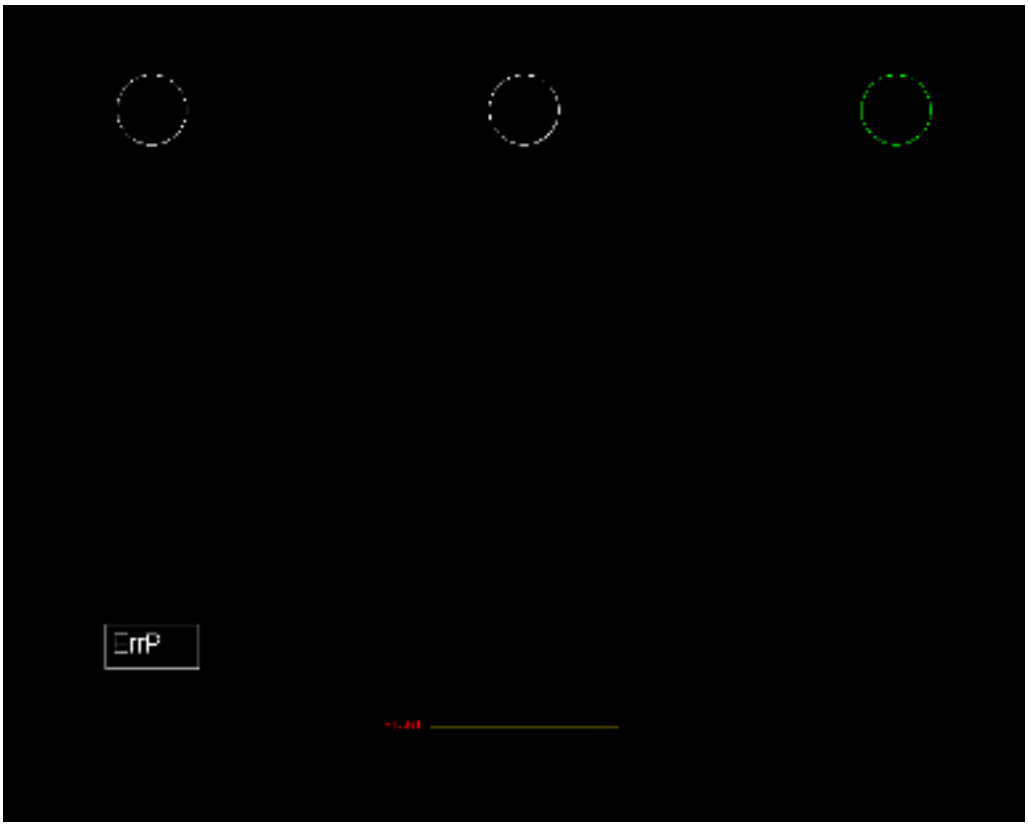


Figura 5: Interfaz Gráfica

3.2 CLASIFICACIÓN

Como se ha comentado en el apartado anterior, los registros realizados a los usuarios se emplean para el cálculo del modelo que identifica el ErrP y la posterior verificación del clasificador.

Para obtener los modelos, se dispone de la información temporal en la que se ha producido el error. Se ha establecido una ventana de 2 segundos desde el momento en el que se produce el error [4].

Todas las ventanas de error obtenidas para cada usuario y para cada porcentaje de aparición de error se promedian para obtener una señal modelo de ErrP que se utilizará en la clasificación final.

El clasificador utilizado se basa en una correlación cruzada normalizada de señales similar a la descrita en [3]. Los modelos se comparan cada 2 segundos, estableciéndose una ventana de entre 0.0166 segundos y 0.866 segundos donde se aplica la correlación cruzada normalizada [4]. Esta ventana se emplea para limitar la clasificación al entorno en el que se produce el ErrP (ver Figura 1) y así mejorar el acierto final.

El resultado de la correlación cruzada normalizada varía entre -1 y 1, por tanto, es necesario establecer un umbral que indique si las dos señales comparadas son suficientemente parecidas como para poder concluirse que se ha producido un ErrP. Los umbrales elegidos varían en función del usuario.

Tras la clasificación, se obtienen tanto los resultados de tasa de acierto en la detección del ErrP, como los resultados de tasa de falsos positivos.

3.3 INTERFAZ GRÁFICA CON REALIMENTACIÓN DE FUERZAS

El objetivo de este apartado es describir el funcionamiento de la interfaz gráfica desarrollada para que el clasificador pueda detectar la señal ErrP. La interfaz gráfica genera la realimentación visual de la respuesta del clasificador y se ha utilizado en combinación con dos tipos de realimentación de fuerzas. Ésta puede observarse en la Figura 5.

El usuario utilizará el dispositivo háptico Phantom para mover un cursor que deberá llevar desde la línea de “START” hacia uno de los tres objetivos situados en la parte superior de la interfaz que se pondrá de color verde. Una vez el usuario llegue al círculo objetivo, éste se volverá rojo y el usuario

deberá volver a la línea de "START" para comenzar un nuevo movimiento.

Durante la realización de la prueba, el movimiento tendrá "errores", es decir, en ciertas ocasiones, el usuario no podrá alcanzar el objetivo y el círculo objetivo se mostrará de color rojo antes de llegar a él.

Esto será realizado de dos formas distintas utilizando la realimentación de fuerzas mediante el dispositivo háptico y serán explicadas a continuación.

Se han establecido dos porcentajes de aparición del error distintos para evaluar la clasificación del ErrP: 20% y 40%. Con ello se pretende comprobar que ante una mayor cantidad de errores se obtienen menor número de detecciones, ya que el usuario tiende a acostumbrarse a la existencia de errores y por tanto deja de producirse la señal ErrP. La utilización del color rojo sobre el círculo objetivo antes de llegar a él al producirse un "error" pretende intensificar la sensación de error del usuario.

Junto con las señales cerebrales registradas, se han almacenado las posiciones del Phantom. Los datos registrados son procesados y clasificados durante la realización de la prueba mostrándose el resultado de dicha clasificación en la interfaz gráfica en tiempo real. Es decir, cada vez que se detecte la señal ErrP se iluminará el cartel situado en la parte inferior izquierda de la interfaz gráfica (Figura 5). Se han realizado un total de 20 registros por usuario realizando en cada uno 10 movimientos. 10 de estos registros se han realizado con el porcentaje de aparición del error del 20% y los otros 10 registros con el porcentaje de aparición del error del 40%.

Los modelos empleados para la utilización de las interfaces en tiempo real han sido obtenidos de sesiones previas en las que se ajustó el clasificador ErrP tal y como se describe en [4].

3.3.1 Interfaz con Realimentación de Fuerzas Aleatorias

Uno de los modos que se han utilizado para provocar un "error" impidiendo al usuario llegar al objetivo ha sido el de emplear la realimentación de fuerzas de forma aleatoria.

En el momento que sucede el error en uno de los movimientos (en función de los porcentajes de error indicados 20% y 40%) una fuerza impedirá llegar al usuario al objetivo. Esta fuerza obligará al usuario a cambiar su trayectoria dirigiéndolo hacia uno de los dos objetivos restantes. Esto provocará en el usuario la señal ErrP ya que el sistema no se comportará de la forma que éste esperaba.

La fuerza aplicada será aplicada moviendo el cursor hacia los objetivos restantes de forma aleatoria para que el usuario no se acostumbre a la fuerza aplicada y a la trayectoria que seguirá el cursor. También se busca el mayor realismo posible y puesto que el usuario utiliza el dispositivo háptico para mover el cursor también sentirá directamente dicha fuerza, además de ver que no es capaz de llegar al objetivo deseado.

Tabla 1: Resultados experimentales usando la interfaz gráfica con realimentación de Fuerzas Aleatorias y diferentes probabilidades de Error (Perr)

Perr = 0,20					
Usuarios	#1	#2	#3	#4	**Media ± DE**
Valor Umbral de Correlación	0,1	0,1	0,1	0,1	
Detección Error (%)	64,17	75,00	67,71	54,17	**65,26 ± 9,74**
Falsos Positivos (%)	23,13	23,72	23,83	16,89	**21,89 ± 5,01**

Perr = 0,40					
Usuarios	#1	#2	#3	#4	**Media ± DE**
Valor Umbral de Correlación	0,1	0,1	0,1	0,1	
Detección Error (%)	62,50	50,14	42,50	53,13	**52,07 ± 10,43**
Falsos Positivos (%)	33,33	17,09	19,57	21,84	**22,96 ± 10,37**

Tabla 2: Resultados experimentales usando la interfaz gráfica con realimentación de Fuerzas Modo Pared y diferentes probabilidades de Error (Perr)

Perr = 0,20					
Usuarios	#1	#2	#3	#4	**Media ± DE**
Valor Umbral de Correlación	0,1	0,1	0,1	0,1	
Detección Error (%)	63,54	77,34	58,33	67,50	**66,68 ± 10,66**
Falsos Positivos (%)	15,83	15,61	15,53	25,42	**18,09 ± 7,32**

Perr = 0,40					
Usuarios	#1	#2	#3	#4	**Media ± DE**
Valor Umbral de Correlación	0,1	0,1	0,1	0,1	
Detección Error (%)	43,39	69,40	54,76	50,00	**54,39 ± 15,01**
Falsos Positivos (%)	15,51	21,55	24,70	22,35	**21,03 ± 5,51**

3.3.2 Interfaz con Realimentación de Fuerzas modo Pared

En este modo, al igual que con la realimentación de fuerzas aleatorias, la fuerza se realizará en el momento en el que el sistema decida que debe producirse un error (en función de los porcentajes de error seleccionados, 20% y 40%).

En este caso, la fuerza impedirá el avance del cursor simulando una pared. De esta forma el usuario no será capaz de llegar al objetivo y deberá volver a la posición de "START" para comenzar un nuevo movimiento. Dicha pared aparecerá a partir de la mitad de camino entre "START" y el objetivo de forma que el usuario no pueda acostumbrarse conocer cuando ésta va a aparecer. Al igual que las interfaces anteriores se busca el mayor realismo posible en la interfaz.

4 RESULTADOS EXPERIMENTALES

En las pruebas experimentales han participado 4 usuarios, todos hombres sanos, con edades comprendidas entre los 25 y los 30 años.Las pruebas experimentales se han realizado utilizando ambas realimentaciones de fuerzas para poder comparar los resultados obtenidos con cada aproximación. Éstas han consistido en realizar 10 registros con cada probabilidad de error (Perr) realizando 10 movimientos en cada uno de ellos.

El usuario deberá alcanzar en cada movimiento el objetivo marcado, y como se ha comentado, en ciertos de estos movimientos se producirá un "error" que impedirá al usuario alcanzar su objetivo. En este punto una señal ErrP será generada. Durante las pruebas se procesarán y clasificarán las señales EEG en busca de dicho ErrP, de forma que cuando éste se detecte sea mostrado por pantalla.

A los usuarios se les muestran algunas pruebas de ejemplo para que se adapte al comportamiento de las interfaces y aprenda a utilizarlas.

Los resultados obtenidos pueden verse en la Tablas 1 y 2. En la parte superior de la Tabla 1 pueden verse los resultados para la probabilidad de error del 20% y abajo para el 40% utilizando una realimentación de fuerzas aleatoria. La primera fila muestra el umbral elegido para cada usuario, que como puede comprobarse ha sido seleccionado en 0.1 para todos ellos. Y finalmente se muestran el porcentaje de acierto en la detección del ErrP y el porcentaje de falsos positivos, es decir, cuando se detecta un ErrP y éste no se ha producido realmente.

Como puede observarse, la tasa de acierto en la detección del ErrP está en torno al 65%, siendo algo inferior cuando el error aparece con mayor frecuencia. Como previamente se ha nombrado, esto es debido a que si los errores provocados aparecen demasiado seguidos los usuarios se acostumbran a su aparición y por tanto la señal ErrP aparecerá en menor medida. Esto dificulta también su detección por parte del clasificador obteniéndose así un porcentaje de acierto inferior.

Por otro lado, la tasa de falsos positivos es de aproximadamente un 22%. En el caso de un Perr mayor (del 40%) esta sube ligeramente, pero no de forma significativa.

En el caso de la realimentación de fuerzas modo pared, los resultados obtenidos son similares. Para una aparición del error mayor (del 40%) los resultados son peores por la misma razón que en el caso anterior. Es decir, el usuario se acostumbra al aparecer tantos errores de forma consecutiva.

Para el caso de una Perr del 20% los resultados obtenidos son ligeramente mejores que con la aproximación con realimentación de fuerzas aleatoria. El porcentaje medio de acierto es del 66% y la tasa de falsos positivos se reduce hasta un 18%.

5 CONCLUSIÓN

En este trabajo se han mostrado los resultados en la detección del potencial de error (ErrP) mediante un clasificador basado en correlación cruzada-normalizada. Para ello, se ha diseñado una interfaz que detecta en tiempo real el ErrP cuando se provoca un error que impide al usuario alcanzar el objetivo indicado. Para llegar a los objetivos se ha utilizado una interfaz háptica (Phantom) y se han diseñado dos aproximaciones para provocar el error en el usuario.Los resultados obtenidos muestran unos porcentajes de detección del ErrP y de falsos positivos aceptables, lo que indica que el sistema propuesto puede ser de utilidad en posteriores trabajos utilizándolo como apoyo de otro tipo de interfaces.

Este trabajo puede servir en tareas de rehabilitación con personas que han sufrido un ictus cerebral como las basadas en la detección de la intención de movimiento. Estas tareas consisten en ayudar al paciente a realizar un movimiento del miembro débil con la ayuda de un mecanismo externo que apoya el movimiento y que se activa en función de si se ha producido una detección de intención de movimiento o no. Este trabajo permitiría reducir el número de ocasiones en que la asistencia se produce sin que el paciente desease realizar el movimiento y por tanto, mejoraría la calidad de la rehabilitación.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad de España a través del proyecto DPI2011-27022-C02-01 y por la Conselleriad'Educació, Cultura i Esport de la Generalitat Valenciana de España a través de las ayudas VALi+d ACIF/2012/135 y ACOMP/2013/018.

Referencias

[1] American Electroencephalographic Society, (1991) "American Electroencephalography Society guidelines for standard electrode position nomenclature", *Journal of Clinical Neurophysiology*, vol. 8(2), pp. 200-202.

[2] Bayliss, J. D., (2003) "Use of the evoked potential P3 component for control in a virtual environment", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, pp. 113-116.

[3] Chavarriaga, R., Millán, J. del R., (2010) "Learning from EEG error-related potentials in noninvasive brain-computer interfaces", *IEEE Transactions of Neural Systems and Rehabilitation Engineering*, vol. 18(4), pp. 381-388.

[4] López, J., Úbeda, A., Iáñez, E., Climent, J.M., Azorín, J.M., (2012) "Error Potential Detection to Assist Movement Intention Decoding in Stroke Patients", *Converging Clinical and Engineering Research on Neurorehabilitation ICNR2012,* pp415-419.

[5] Férrez, P. W., Millán J. del R., (2008) "Errorrelated EEG potentials generated during simulated brain-computer interaction", *IEEE Transactions Biomedical Engineering,* vol. 55(3), pp. 923-929.

[6] Gao, X., Dignfeng, X., Cheng, M., Gao, S., (2003) "A BCI-based environmental controller for the motion-disabled", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, pp. 137-140.

[7] Iáñez, E., Azorín, J. M., Úbeda, A., Ferrández, J. M., and Fernández, E. (2010) "Mental tasksbased brain–robot interface", *Robotics and Autonomous Systems,* vol. 58, pp. 1238-1245.

[8] Iturrate, I., Antelis, J. M., Kubler, A., Minguez, J., (2009) "A non-invasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation", *IEEE Transactions on Robotics*, vol. 25(3), pp. 614- 627.

[9] Lewis, J. P., (1995) "Fast normalized crosscorrelation", Industrial Light and Magic.

[10] Lopez-Larraz, E., Iturrate, I., Montesano, L., Minguez, J., (2010) "Real-time recognition of feedback error-related potentials during a timeestimation task", *2010 Annual International Conference Proceedings of the*

IEEE Engineering in Medicine and Biology Society (EMBC), Argentina, pp. 2670-2673.

[11] McFarland, D. J., McCane, L. M., David, S. V., Wolpaw, J. R., (1997) "Spatial filter selection for EEG-based communication", *Electroencephalography and Clinical Neurophysiology*, vol. 103, pp. 386-394.

[12] Nicolelis, M. A. L., (2001) "Actions from thoughts", *Nature*, vol. 409, pp. 403-407.

[13] Sirvent, J. L., Iáñez, E., Úbeda, A., Azorín, J.M., (2012) "Visual evoked potential-based brain– machine interface applications to assist disabled people", *Expert Systems with Applications*, vol. 39(9), pp.7908-7918.

MOVIMIENTO BIDIMENSIONAL DE UN CURSOR MEDIANTE EL USO DE ARTEFACTOS EN SEÑALES ELECTROENCEFALOGRÁFICAS

Álvaro Costa, Eduardo Iáñez, Enrique Hortal, José M. Azorín, Alberto Rodríguez, Daniel Tornero
Universidad Miguel Hernández de Elche
Avda. de la Universidad s/n, 03202 Elche (Alicante)
eianez@umh.es

José A. Berná
Universidad de Alicante
Carretera San Vicente S/N, 03690 Alicante

José M. Cano
Universidad Politécnica de Cartagena
Muralla del Mar S/N, 30202 Cartagena, Murcia

Resumen

Este artículo investiga la utilización de artefactos generados por un usuario y controlables mediante sutiles movimientos de la mandíbula para el control de un brazo robótico en dos dimensiones. El control se realizará ejerciendo una leve presión del lateral izquierdo o derecho de la mandíbula, presión cuyo nivel se podrá controlar hasta cierto punto por la interfaz creada permitiendo de esa forma al usuario controlar hasta 5 estados distintos que le permitan mover el cursor y conmutarlo entre las direcciones vertical y horizontal. Se ha probado el sistema diseñado con 3 usuarios, los cuales han adquirido de forma rápida el control bidimensional de un cursor que se desplaza por pantalla.

Palabras Clave: Artefacto, Control Bidimensional, BCI, No invasivo, Máxima Entropía

1 INTRODUCCIÓN

En general, los artefactos han sido objeto de problemas para los estudios realizados sobre interfaces cerebro-computador (BCI) sobre todo aquellos artefactos que aparecen en rangos de frecuencias entre 8 y 26 Hz. Esto se debe a que en dicho rango frecuencial se producen las variaciones más visibles del espectro ante pensamientos relacionados con tareas motoras [4].

Los artefactos son señales cerebrales provocadas por movimientos musculares que afectan generalmente a las frecuencias bajas [2][6].Según el movimiento que los produzca pueden llegar a afectar a frecuencias superiores.

Estos artefactos son fáciles de detectar ya que los niveles de potencia que generan están muchos órdenes por encima de los niveles de las señales con las que trabajan los sistemas BCI más comunes. Así, mediante un sistema de umbrales, estos artefactos pueden ser detectados y eliminados.

Los motivos por los que los artefactos son señales no deseadas son principalmente dos. El primero es que al ir enfocados estos estudios a la ayuda de personas con discapacidades motoras para facilitar su comunicación con el entorno, lo movimientos musculares necesarios para generar estos artefactos no pueden ser llevados a cabo con normalidad. El segundo es que el nivel de potencia de los artefactos es muy variable entre unos usuarios y otros así como en los mismos usuarios para pruebas realizadas en diferentes momentos, lo que ha provocado que la mayoría de estudios sobre los artefactos se hayan centrado en su detección y eliminación.

El estudio que nos ocupa pretende demostrar que existen artefactos que pueden ser controlados por el usuario. Estos artefactos podrían ser utilizados conjuntamente con señales electroencefalográficas (EEG) en lugar de ser únicamente desechados.

Los artefactos que se estudian son provocados por la presión, en distintas partes de la boca, de la mandíbula superior contra la inferior lo que genera un incremento notable en la amplitud del artefacto localizado en la zona correspondiente del cerebro en cuya vertical se haya producido la presión.

Las frecuencias a las que se han detectado estos artefactos son entre 57-77 Hz. Esto hace su análisis compatible con el de las frecuencias inferiores donde se encuentra la información referente a la imaginación de tareas motoras (8-26 Hz). De esta forma se pretende que esta investigación pueda ser utilizada de forma complementaria al estudio de sistemas BCI de detección de pensamientos motores.

La mayoría de personas con discapacidades motoras mantienen movilidad en la zona facial de la boca y los ojos. De esta forma todos los sistemas que puedan incorporarse a una interfaz multimodal, ya sean usando señales oculares, artefactos o imaginación de tareas motoras, aumentarán las interacciones que estas personas puedan tener con su entorno [7].

El objetivo final de esta investigación es la utilización de artefactos para el movimiento bidimensional de un cursor que posteriormente puede ser extrapolado al control de un brazo robótico en dos dimensiones.

El resto del artículo estará estructurado de la siguiente forma: En la sección 2 se describirá la arquitectura del sistema que permite el registro y procesado de señales así como su clasificación. En la sección 3 se especificará la metodología usada para el aprendizaje del usuario y las interfaces gráficas desarrolladas. En la sección 4 se comentarán los resultados obtenidos de aplicar la metodología anterior. Las conclusiones del estudio se muestran en la sección 5.

2 ARQUITECTURA DEL SISTEMA

2.1 REGISTRO

La arquitectura está formada por una interfaz no invasiva [5] que recoge las señales en la zona motora del cerebro. La información es procesada y analizada por un bloque decisor que determina qué estado, de aquellos establecidos en un modelo, está realizando el usuario. Se ha utilizado una interfaz gráfica para proveer al usuario de realimentación visual. Con ella, el usuario puede realizar un entrenamiento que le permite controlar el movimiento bidimensional de un cursor que se desplaza por pantalla.

Las señales se recogen a través del gorro GammaCap de g.Tec utilizando 10 electrodos activos. En la figura 1 podemos observar cuales son

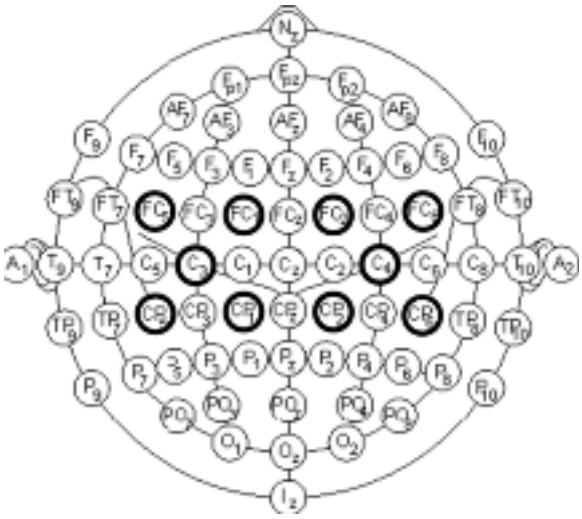


Figura 1: Configuración de electrodos

los electrodos usados. Los principales electrodos para nuestro análisis son C3 y C4, que corresponden a las zonas del cuero cabelludo donde se producen las variaciones más detectables de ondas EEG ante pensamientos motores de derecha e izquierda. De esta forma podrá combinarse esta investigación con estudios de detección y clasificación de pensamientos motores. El resto de electrodos (FC5, FC1, FC2, FC6, CP5, CP1, CP2, CP6) se usarán para eliminar las contribuciones de ruido que se hayan podido producir en C3 y C4. La tierra se sitúa en la posición AFz y la referencia en el lóbulo de la oreja derecha.

Para registrar las señales se utiliza el amplificador comercial gUSBamp de g.Tec. La frecuencia de muestreo de la señal es de 1200 Hz. Se aplica un filtro interno de 0.1 a 100 hercios y un filtro Notch de 50 hercios para eliminar las perturbaciones de la realimentación eléctrica. Tras ello las señales son procesadas y clasificadas. En la figura 2 se muestra una imagen del equipo utilizado.

2.2 PROCESAMIENTO

Las señales correspondientes a los artefactos producidos por la mandíbula se registran cada 50 ms en ventanas de 400 ms (con solape de 350 ms). Tras esto, a la información temporal de los electrodos C3 y C4 se le aplica un filtro Laplaciano de 4 vecinos [1]. Este algoritmo utiliza la información recogida por los 4 electrodos más cercanos a C3 y C4 y la distancia física con respecto a ellos para eliminar sus contribuciones de ruido de forma que la señal resultante quede más suavizada. La fórmula utilizada por este algoritmo es la siguiente:



Figura 2: Equipo utilizado

$$Vi^{LAP} = Vi^{CR} - \sum_{j \in Si} g_{ij} Vi^{CR} \quad (1)$$

Donde Vi^{LAP} es la señal resultante tras aplicar el algoritmo para el electrodo i, Vi^{CR} es la señal del electrodo i antes de ser procesada y

$$g_{ij} = \frac{\frac{1}{d_{ij}}}{\sum_{j \in Si} \frac{1}{d_{ij}}} \quad (2)$$

Siendo S_i el conjunto de electrodos que rodean a i-ésimo electrodo y d_{ij} la distancia entre los electrodos i y j.

A continuación se somete a las señales resultantes a un análisis espectral a través del método de máxima entropía (MEM) [3]. Por último, cada 50 ms se evalúa el nivel de potencia de la señal (de los últimos 400 ms) de los electrodos C3 y C4 en el rango de frecuencias comprendido entre 57 – 77Hz. Tras analizar todo el rango de frecuencias comprendido entre 0.1 y 100 hercios, se comprobó que con estas frecuencias se obtenían los mejores resultados. Finalmente, se aplica un clasificador a la diferencia de potencia de los electrodos C3 y C4.

2.3 CLASIFICACIÓN

Se han establecido 5 estados o tareas a clasificar en función de la señal de potencia diferencial producida por el usuario mediante movimientos de la mandíbula. Estos estados son:

- Estado 1: Derecha
- Estado 2: Reposo.
- Estado 3: Izquierda
- Estado 4: Derecha fuerte
- Estado 5: Izquierda fuerte

En la figura 3 podemos observar el valor de la señal diferencial para la mandíbula en reposo, es decir, sin ejercer presión (gráfica central de la figura) y ejerciendo presión a derecha e izquierda, estados 1 y 3 (graficas izquierda y derecha de la figura).

Es posible identificar que la señal procedente de C4 predomina sobre C3 cuando se presiona la zona izquierda y viceversa. En ambos casos, los niveles medios de señal están fuera del rango de los niveles medios de señal en el estado de reposo. Por ello, para su clasificación se ha establecido un sistema de 2 umbrales (umbrales Thr1 y Thr-1 de la figura 3). De forma que si se supera el umbral superior se detecta el estado 1(movimiento derecha), si se encuentra por debajo del umbral inferior se detecta el estado 3 (movimiento izquierda). Finalmente si se encuentra entre ambos el estado detectado será el 2 (reposo).

Tras comprobar el correcto funcionamiento de estos tres estados se han establecido dos umbrales más (Thr2 y Thr-2 de la figura 3), uno por encima del superior y otro por debajo del inferior. De esta forma se conseguirán dos estados más controlados por el nivel de presión que se realice con la mandíbula en cada lado.

Los estados 4 y 5 son alcanzados ejerciendo una

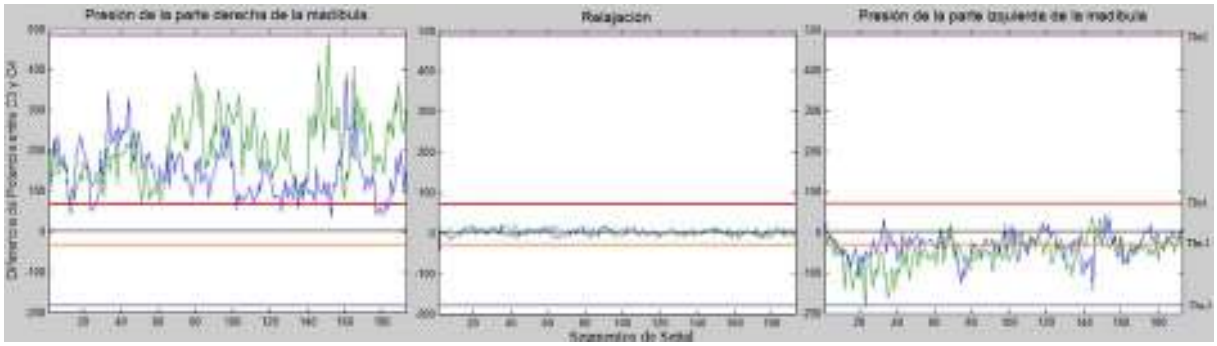


Figura 3: Resultados para 2 realizaciones de cada una de las 3 tareas y umbrales decisión

presión mayor que la necesaria para los estados 1 y 3. En ellos no se permanecerá de forma continua como en los estados del 1 al 3, evitando así las molestias que pueden generarle a un usuario tener que realizar dicha presión durante un tiempo prolongado.

3 METODOLOGÍA

El proceso de entrenamiento del usuario está dividido en tres fases. Las 2 primeras utilizan la misma interfaz gráfica y la tercera una modificación de ésta. En todos ellos se procesarán los datos en tiempo real y el usuario recibirá una realimentación visual de los resultados de la clasificación. De esta forma el usuario sabrá si lo está haciendo bien o mal.

3.1 INTERFACES GRÁFICAS

Las interfaces gráficas utilizadas son 2. En ambas el usuario dispondrá de un cursor con forma de círculo rojo que se desplazará por pantalla en función de los artefactos que genere.

El objetivo de la primera interfaz será indicar al usuario que tareas ha de realizar para alcanzar los estados deseados. Mediante esta información se evalúa el comportamiento de los artefactos para cada tarea de forma que se pueda generar un modelo que se adapte a un usuario concreto. Esta interfaz está compuesta de un fondo negro sobre el que se proyectará una cruz durante 3 segundos, tras ellos se mostrará durante 2 segundos una imagen representativa de la tarea que se pide al usuario y pasado ese tiempo aparecerá el cursor rojo en el centro de la pantalla. Este cursor se moverá en función de los artefactos producidos por el usuario. Cada tarea tiene una duración total de 15 segundos y este proceso se repetirá 16 veces, dando lugar a una sesión de 4 minutos.

La segunda interfaz se emplea para el entrenamiento del usuario una vez se ha definido un modelo para él. Está compuesta de un fondo negro sobre el que se proyecta un objetivo (circunferencia verde punteada) en una posición aleatoria de la pantalla. A los 3 segundos de la aparición del objetivo, volverá a aparecer el cursor rojo que será de nuevo controlado por el usuario, el cual deberá conducirlo hasta el objetivo. Tras alcanzarlo se mostrará un nuevo objetivo y este proceso se repetirá 10 veces. El tiempo máximo permitido para alcanzar el objetivo es de 20 segundos tras la aparición del cursor, si tras ese tiempo no ha sido alcanzado, se contará como un fallo y se mostrará

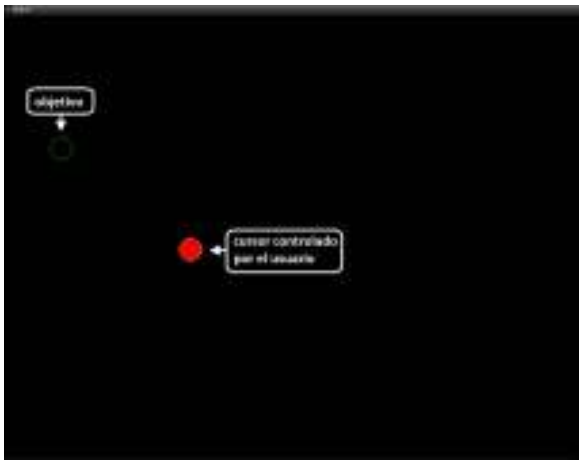


Figura 4: Interfaz gráfica utilizada

el siguiente objetivo. En la figura 4 se muestra esta interfaz gráfica.

3.2 FASES

Las fases del entrenamiento están diseñadas para, de forma gradual, instruir al usuario en el control de un cursor en 2 dimensiones. En cada una de ellas se partirá de una de las 2 interfaces gráficas vistas anteriormente y de unos umbrales de decisión concretos.

La primera dispone de unos umbrales fijos predefinidos de [100 -100] (Thr1 y Thr-1 de la figura 3 respectivamente). Se utiliza la primera interfaz gráfica descrita. Mediante ella se pide de forma pseudoaleatoria la realización de las tareas de movimientos de derecha e izquierda (estados 1 y 3). Alcanzar satisfactoriamente estos estados, supondrá un desplazamiento del cursor en las direcciones derecha e izquierda respectivamente. El objetivo de este paso es que el usuario se acostumbre al movimiento que ha de realizar con la mandíbula para obtener unos resultados deseados. Se ha comprobado experimentalmente que partiendo de estos umbrales, con una sola sesión, un usuario que nunca haya probado el sistema se adapta rápidamente a ambos estados.

La segunda fase comienza con los umbrales [10000 100 -100 -10000]. Se utiliza la misma interfaz que en el paso anterior con la única diferencia de que esta vez se incluirá también la tarea de reposo (estado 2). Alcanzar este estado supondrá que el cursor permanezca inmóvil. En este caso, tras cada tarea de 10 segundos, se modificará una matriz como la observada en la figura 5.
Se trata de una matriz de 3x5 cuyas filas representan la tarea concreta que se pide cada 10 segundos y las columnas representan las tareas que se han clasificado cada 50 ms en función de los

umbrales existentes (se toman 5 tareas de salida ya que existe la posibilidad de que el usuario sobrepase los umbrales thr2 y thr-2 de la figura 3). Por lo tanto cada 10 segundos es posible saber qué tarea se pidió y qué tareas se clasificó. En función del porcentaje de tareas mal clasificadas, se incrementarán o disminuirán los umbrales de partida para adaptarlos mejor a cada usuario. En el caso de que los aciertos sean del 100%, los umbrales que delimitan la tarea en concreto serán disminuidos un 10%, forzando de esa forma al usuario a aumentar su precisión.

Esto provoca que los umbrales vayan convergiendo hacia los óptimos y cuando estén muy cercanos a los mismos oscilan en torno a ellos.

Tras 5 pruebas de 4 minutos, los umbrales han convergido a aquellos que cada usuario mejor controla. Se seleccionarán con un margen de error de un 10% para proporcionarle al usuario cierto nivel de libertad.

En la última fase se usarán los umbrales obtenidos en la fase anterior y serán fijos. Esta vez se empleará la segunda interfaz gráfica nombrada. En esta fase el usuario tratará de alcanzar todos los objetivos moviendo el cursor en dos dimensiones.

El movimiento del cursor se controlará con los estados 1 y 3 (derecha e izquierda o arriba y abajo según en qué dimensión nos encontramos), los estados 4 y 5 serán equivalentes y su función será conmutar entre la dirección vertical y horizontal. El estado 2 será el de reposo.

Esta prueba sirve para que el usuario adquiera práctica con el control bidimensional del cursor. En



Figura 5: Matriz de clasificación de tareas

la figura 6 se muestra la señal producida por un usuario tratando de alcanzar un objetivo. En ella podemos apreciar claramente la diferencia entre los estados 1, 3, 4 y 5.

Se realizan 8 sesiones en cada una de las cuales el usuario trata de alcanzar 10 objetivos en un tiempo máximo de 25 segundos por objetivo.

Teniendo en cuenta que cada prueba es de una duración aproximada de 4 min y tomando descansos de 1 min entre prueba y prueba, se puede estimar el tiempo que requiere para un usuario controlar con soltura el cursor:

- 1 sesión inicial para acostumbrarse al movimiento a realizar.
- 5 sesiones de convergencia de los umbrales del modelo.
- 8 pruebas para coger soltura con el movimiento bidimensional.

En total se requieren 70 minutos aproximadamente.

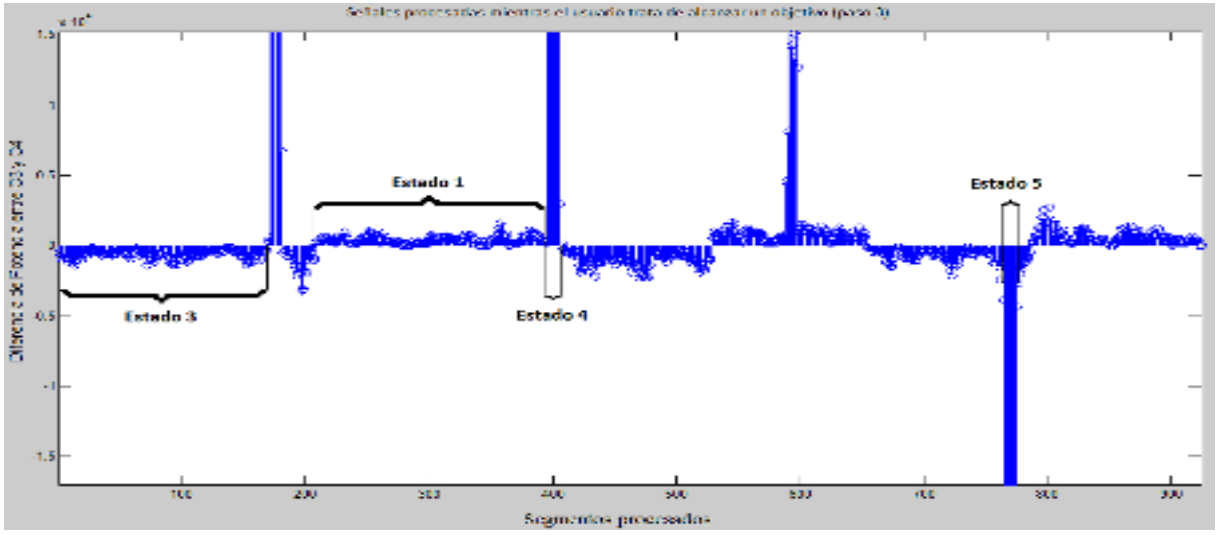


Figura 6: Resultados online de un usuario tratando de alcanzar un objetivo (fase 3)

Tabla 1: Porcentaje de aciertos y fallos en el alcance del objetivo de 3 usuarios a lo largo de 8 sesiones

SESIONES		S1	S2	S3	S4	S5	S6	S7	S8
Usuario 1	**aciertos**	30 %	70 %	90 %	90 %	100 %	90 %	100 %	90 %
	fallos	70 %	30 %	10 %	10 %	0 %	10 %	0 %	0 %
Usuario 2	**aciertos**	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
	fallos	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
Usuario 3	**aciertos**	40 %	80 %	90 %	90 %	100 %	100 %	80 %	100 %
	fallos	60 %	20 %	10 %	10 %	0 %	0 %	20 %	0 %

4 RESULTADOS EXPERIMENTALES

Las pruebas experimentales han sido realizadas por 3 usuarios, todos ellos hombres sanos con edades comprendidas entre los 23 y los 28.

Tanto en el primero como en el segundo paso del aprendizaje, los usuarios se han acostumbrado a ejercer presión en diferentes puntos de la mandíbula para controlar la dirección del cursor en una dimensión. Llegados a la tercera fase, los usuarios 1 y 2 les resultó más cómodo controlar el movimiento del cursor mediante leves desplazamientos de la mandíbula inferior a derecha e izquierda mientras que el usuario 3 siguió ejerciendo presión en la parte derecha e izquierda. Por otro lado, los estados 4 y 5 eran alcanzados por todos los usuarios con un golpe rápido de la parte derecha o izquierda respectivamente de la mandíbula superior contra la inferior.

En la tabla 1 se muestra la evolución de los 3 usuarios en función del porcentaje de veces que hayan alcanzado el objetivo en cada sesión.

El 100% de aciertos que se observa en el usuario 2 se debe a que fue con el que se probó todo el funcionamiento del sistema antes de diseñar las interfaces definitivas. Esto le hizo adquirir gran soltura con el desplazamiento del cursor. Los usuarios 1 y 3 realizaron una única prueba compuesta de los 3 pasos comentados con anterioridad.

Se observa la facilidad y rapidez con la que los usuarios se acostumbran al uso de esta interfaz. Además, la adaptación es muy rápida y estable. Aunque se ha establecido que fuesen 8 sesiones de entrenamiento, a partir de la tercera sesión, los resultados que se muestran son casi del 100%. Con lo que realmente se puede afirmar que con 3 sesiones de la última fase, los usuarios controlan perfectamente el movimiento bidimensional. Lo cual reduce a 35 minutos el tiempo necesario para adaptarse al control.

5 CONCLUSIONES

Se ha diseñado una interfaz basada en artefactos generados por movimientos mandibulares que permite el control bidimensional de un cursor de forma fluida siendo el tiempo de entrenamiento mínimo.

Se ha cumplido el principal objetivo, facilitando las interacciones de personas con discapacidades motoras con su entorno. Aunque no se trate de señales procedentes de pensamientos, la capacidad de mover la mandíbula es algo presente en la gran mayoría de personas con este tipo de discapacidades. Además, al producirse dichos artefactos a frecuencias entre 57 y 77 Hz, es posible su combinación con el procesado de señales procedentes del pensamiento ya que no interfieren en su rango frecuencial (8-26 Hz).

Como investigación futura se pretende probar este sistema para el control de un brazo robótico en dos dimensiones. Además, aprovechando que ambas técnicas, señales EEG y artefactos mandibulares, utilizan frecuencias distintas, podrían utilizarse de forma conjunta pudiendo generar un mayor número de comandos y por tanto realizar tareas más complejas, como el alcance y agarre de objetos con un brazo robótico.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad de España a través del proyecto DPI2011-27022-C02-01 y por la Conselleria d'Educació, Cultura i Esport de la Generalitat Valenciana de España a través de las ayudas VALi+d ACIF/2012/135 y ACOMP/2013/018.

Referencias

[1] Hjorth B. (1975). "An on-line transformation of EEG scalp potentials into orthogonal source derivations." *Electroenceph Clin Neurophysiol*; Vol. 39, pp. 526-530.

[2] Lacy, S. (2012). "Signal Artifact: Intrinsic, Extrinsic or Pathological?" .*European Society of Sleep Technologist, Paris*.

[3] Marple, S.L. (1987). Digital spectral analysis with applications. *Prentice-Hall*, Englewood Cliffs, New Jersey.

[4] McFarland, D.J., McCane, L.A., Miner, L.A., Vaughan, T.M. and Wolpaw, J.R. (2000). "EEG mu and beta rhythm topographies with movement imagery and actual movement", *Brain Topography*, Vol 12(3), pp. 177-186.

[5] Millán, JdR., Ferrez, P.W., Buttfield, A. (2005). Non Invasive Brain-Machine Interfaces. Final Report. IDIAP Research Institute - ESA.

[6] Reyes, C.E., Rugayan, J.L.C., Jason, C., Rullan, G., Oppus, C.M., Tangonan, G.L. (2012). "A study on ocular and facial muscle artifacts in EEG signals for BCI applications," *TENCON 2012 - 2012 IEEE Region 10 Conference*, pp. 1,6.

[7] Úbeda, A., Iáñez, E., Azorín, JM., Sabater, JM, Pérez, C., García, NM. (2012). "Interfaz multimodal basada en la combinación de información ocular, cerebral y háptica para el control de un brazo robot". *Actas de las XXXIII Jornadas de Automática -(JA 2012),* Vigo, España,pp. 47-52.

MRSeg - Herramienta interactiva para generar segmentaciones de referencia de imágenes médicas

F. Fumero, S. García, O. Núñez, J. Sigut, S. Alayón
Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores, Facultad de Física
Universidad de La Laguna (ULL). Avda. Astrofísico Fco. Sánchez s/n, 38206, La Laguna (Tenerife), España.
franfumero@isaatc.ull.es

Resumen

La segmentación manual de imágenes médicas es necesaria para construir segmentaciones de referencia que contengan las regiones de interés seleccionadas por un experto en la materia. En este trabajo se presenta la herramienta MRSeg, con la que se pretende simplificar este proceso, permitiendo generar de manera rápida y sencilla un conjunto de imágenes etiquetadas, que sirvan como Gold Standard para evaluar o entrenar nuevos métodos de segmentación automáticos. El proceso de segmentación interactiva está guiado por algoritmos de segmentación de propósito general, que pueden ser adaptados para trabajar con cualquier tipo de imagen médica bidimensional. Además, la aplicación se puede utilizar por más de un usuario, permitiendo segmentar varias veces la misma imagen, reduciendo la subjetividad del procedimiento.

Palabras clave: segmentación interactiva, imágenes médicas, segmentaciones de referencia, gold standard, ground truth.

1. INTRODUCCIÓN

La segmentación de imágenes es una etapa fundamental en muchas aplicaciones médicas. Existen numerosos métodos automáticos para resolver problemas concretos, aunque la segmentación manual de imágenes continúa siendo necesaria donde todavía no se ha conseguido automatizar el proceso o para el diseño y la evaluación de nuevos algoritmos. En este último supuesto, es imprescindible disponer de segmentaciones de referencia realizadas por expertos en el área, también llamadas *Ground Truth* o *Gold Standard*.

La segmentación manual de imágenes médicas es una tarea que consume mucho tiempo y resulta tediosa, sobre todo cuando el objetivo es crear un conjunto lo suficientemente amplio de segmentaciones de referencia para probar o entrenar un nuevo algoritmo de segmentación y obtener resultados fiables y que se puedan generalizar. Existen ya diversas aplicaciones con este fin. Sus principales desventajas son la excesiva intervención requerida por parte del usuario, el uso de un gran número de herramientas para realizar el etiquetado de objetos y la necesidad de entrenar los algoritmos en los que se apoyan para realizar la segmentación. Por todo ello, en la práctica resultan complicadas de utilizar o no están preparadas para imágenes 2D, como discutiremos en detalle en la sección 2

Con el objetivo de facilitar esta tarea y salvar algunas de estas dificultades hemos diseñado MRSeg, una herramienta de segmentación interactiva creada específicamente para generar segmentaciones de referencia de conjuntos de imágenes médicas. La aplicación se puede adaptar a cualquier tipo de imagen médica 2D, a color o en escala de grises, y permite definir tantas clases de objetos como sea necesario. El procedimiento de segmentación manual está guiado por un método de segmentación automático, que será el encargado de realizar una segmentación inicial de la imagen, dando como resultado un número reducido de regiones, por lo que la tarea del usuario se reduce a elegir qué regiones pertenecen a cada objeto de interés. Esta selección se realiza mediante el ratón de una manera sencilla e intuitiva, haciendo clic y arrastrando el puntero sobre el objeto que se quiera etiquetar, esto es, unir las regiones subyacentes generadas por el algoritmo automático. De esta manera conseguimos una imagen etiquetada con las distintas clases que la componen, que podemos usar como segmentación de referencia.

Para crear las segmentaciones iniciales, previas al etiquetado, se puede usar cualquier algoritmo de segmentación de propósito general, que sea capaz de producir buenas segmentaciones del tipo de imagen con la que deseamos trabajar. En la práctica esto se traduce en que las regiones generadas no mezclen partes de objetos de interés distintos, por lo que sería suficiente con una imagen sobresegmentada. La aplicación está pensada para que se puedan añadir nuevos métodos automáticos de manera fácil. Incluso se podrían utilizar algoritmos específicamente diseñados para un tipo de imagen y utilizar MRSeg para refinar o corregir los resultados del mismo.

Una vez terminada la segmentación interactiva se almacenan los resultados en formato Matlab,

para que puedan ser reutilizados cómodamente. Además, es posible crear varios usuarios de la aplicación, de modo que podamos tener más de una segmentación de la misma imagen, realizadas por diferentes expertos, reduciendo así la subjetividad al segmentar las distintas estructuras.

En la dirección `https://ddv.ull.es/users/medimrg/public` se puede descargar la última versión de MRSeg.

En las siguientes secciones haremos una revisión del trabajo relacionado con esta herramienta y describiremos en detalle sus características y funcionamiento.

2. TRABAJO RELACIONADO

En los últimos años se pueden encontrar numerosos trabajos sobre segmentación interactiva de imágenes médicas. Algunos de ellos tratan sobre nuevos métodos de segmentación que se apoyan en la interacción del usuario para conseguir mejores resultados. Otros presentan aplicaciones que permiten al usuario segmentar de manera interactiva distintos tipos de imagen, haciendo uso tanto de sus propios métodos de segmentación interactiva como de otros más estándar. En esta última categoría se pueden encontrar varias aportaciones, tanto comerciales como libres. En esta revisión nos centraremos en las herramientas que están disponibles de manera gratuita o son de código abierto y que permiten crear segmentaciones de referencia de imágenes médicas, con el fin de destacar las aportaciones de MRSeg, la aplicación presentada en este artículo.

La mayor parte de este tipo de herramientas sólo trabajan con imágenes médicas tridimensionales en escala de grises. Ejemplos de ellas son *TurtleSeg* [4][11][10], *ITK-SNAP* [13] y el entorno de segmentación interactiva [6] del *MITK* [12]. *TurtleSeg* se apoya en el algoritmo *Turtlemap 3D Livewire* [5] y permite crear segmentaciones de múltiples clases en imágenes médicas 3D, con una mínima interacción por parte del usuario. Para ello, han implementado además una característica llamada *Spotlight* [10] que guía al usuario en el proceso de segmentación. Por su parte, *ITK-SNAP* se basa en el uso de métodos de contorno activo para realizar segmentaciones semiautomáticas, permitiendo además corregir manualmente las segmentaciones proporcionadas por el algoritmo. También tiene soporte para ciertos tipos de imágenes en color, como los *diffusion tensor maps*. Por último, el entorno de segmentación interactiva del *MITK* permite realizar segmentaciones manuales de objetos de interés en imágenes médicas. Puede segmentar imágenes 3D y 3D + t. Como método principal para la segmentación 3D utiliza region growing para imágenes 2D basado en el nivel de gris del píxel bajo el ratón, es decir, construye una región conectada que tenga un valor de gris similar, usando un umbral modificable por el usuario, y posteriormente realiza una interpolación para construir la segmentación final en 3D. Asimismo proporciona distintas herramientas para corregir, rellenar o añadir la región que produce el método de segmentación. También es capaz de corregir "filtraciones" de la región hacia otras regiones que no son de interés: construye un esqueleto de la región y calcula la parte más estrecha por la que atraviesa dicho esqueleto, lo que permite eliminar una porción de la región segmentada basándose en esta información. Se puede utilizar esta herramienta para generar *Gold Standards* con las que probar la eficacia o entrenar métodos de segmentación. Aunque todas estas herramientas son bastante potentes, el hecho de estar centradas en imágenes 3D en escala de grises limita su aplicación. En cualquier caso, *ITK-SNAP* y *MITK* son de código abierto, por lo que su funcionalidad podría ser extendida para adaptarlos a otras necesidades.

Por otro lado, respecto a las herramientas que permiten trabajar con imágenes 2D, una de las pocas aplicaciones disponibles es *ilastik* [9], una herramienta interactiva de aprendizaje y segmentación. Permite entrenar un clasificador basado en *Random Forest* mediante el etiquetado de un número arbitrario de clases en imágenes de cualquier tipo, incluyendo imágenes médicas (ver Fig. 1). Una vez entrenado de manera interactiva, este clasificador se puede exportar para segmentar automáticamente un conjunto de imágenes. Permite trabajar con imágenes en color desde 2 hasta 4 dimensiones (3D + t). Las segmentaciones generadas se pueden usar posteriormente tanto para analizar las imágenes como para crear conjuntos de segmentaciones de referencia. La interfaz es cómoda de utilizar, aunque al estar apoyada en un clasificador puede ser difícil que funcione correctamente con imágenes que presenten mucha variabilidad entre ellas, y, si bien un usuario avanzado puede incorporar *plugins* para extraer características distintas a las disponibles por defecto, un usuario menos experto en el campo de la visión por ordenador podría encontrar ciertas dificultades para conseguir buenos resultados.

Por último, también existen herramientas que están diseñadas para trabajar específicamente con cierto tipo de estructuras, como por ejemplo *LiveVessel* [8][3] que sirve para localizar vasos en imágenes 2D de manera interactiva.

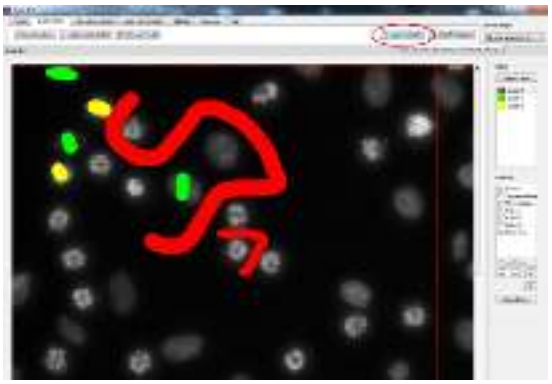


Figura 1: Ejemplo de entrenamiento del clasificador de *ilastik* mediante etiquetado manual de las distintas clases de objetos.

3. DESCRIPCIÓN DE LA APLICACIÓN

El objetivo principal de MRSeg es la creación de conjuntos de segmentaciones de referencia para cualquier tipo de imagen médica 2D. Puede trabajar con imágenes a color y es capaz de utilizar cualquier algoritmo de segmentación automático para guiar al usuario en el proceso de etiquetado de los distintos objetos que componen una imagen médica.

Se ha diseñado teniendo en cuenta la facilidad de uso y los requisitos deseables a la hora de generar segmentaciones de referencia de imágenes médicas como, por ejemplo, disponer de más de una segmentación de referencia para cada imagen, realizada por distintos usuarios, con el fin de minimizar los errores a la hora de delimitar las estructuras de interés.

Por tanto, la aplicación brinda la posibilidad de trabajar con múltiples métodos de segmentación y múltiples usuarios y se puede configurar para etiquetar cualquier tipo de imagen. Incluso podría servir para crear conjuntos de referencia de imágenes que no sean médicas pero que presenten clases de objetos.

En los siguientes apartados describiremos la estructura del sistema y sus funcionalidades y ventajas, entrando en detalle de las opciones de configuración y las herramientas disponibles, del proceso de segmentación interactiva y de la estructura de directorios donde se almacenarán los resultados de la aplicación.

3.1. ESTRUCTURA DEL SISTEMA

La aplicación MRSeg consta de 3 partes principales: la interfaz de usuario, la base de datos y los métodos de segmentación.

La interfaz de usuario está desarrollada en Java, utilizando Swing como librería gráfica, por su portabilidad y posibilidad de personalización de los componentes para adaptarlos al estilo del sistema operativo. Además, esta parte del sistema es la que controla el acceso a la base de datos y la invocación de los distintos métodos de segmentación.

Para la base de datos se ha optado por el sistema gestor H2, incorporado directamente en la aplicación para simplificar la distribución de la misma. Será la encargada de almacenar la configuración de la aplicación, de los usuarios de la misma, de los métodos automáticos y sus parámetros, y de los tipos de imagen con los que vamos a trabajar.

Con respecto a los métodos de segmentación, hemos decidido utilizar una capa intermedia entre la interfaz en Java y los métodos propiamente dichos. Esta capa está programada en Matlab y hace uso de MATLAB Builder JA para poder ser utilizada desde Java. Cabe destacar que no es necesario tener una licencia de Matlab, ya que sólo se necesita Matlab Compiler Runtime para que la aplicación funcione, que se puede distribuir de manera gratuita y está disponible para varios sistemas operativos. De esta manera se consigue una forma muy cómoda de utilizar cualquier método de segmentación, ya que los distintos algoritmos pueden estar escritos en Matlab o en cualquier otro lenguaje, como por ejemplo C++, en cuyo caso bastaría con disponer de un *wrapper* que permita invocarlos desde Matlab. Dado que existen numerosos algoritmos de segmentación con implementaciones en Matlab o con *wrappers* ya disponibles, esta capa intermedia facilita la inclusión de nuevos métodos para utilizarlos de manera interactiva en la aplicación.

Además, se ha aislado lo máximo posible la parte de la interfaz de la capa de invocación de los métodos desde Matlab, hasta el punto de que para incluir un nuevo método sólo es necesario añadir un *script* de Matlab con el nombre del método, que reciba como datos de entrada la imagen a segmentar y un vector con los parámetros del mismo, y devuelva la imagen segmentada. Posteriormente sólo sería necesario añadir el nuevo método para usarlo desde la interfaz, haciendo uso de las opciones de configuración disponibles, sin necesidad de modificar o recompilar la interfaz.

En la figura 2 se puede ver un esquema de la estructura global del sistema.

3.2. CARACTERÍSTICAS PRINCIPALES

Tal y como vimos en la sección 2 existen pocas herramientas que permitan trabajar con imágenes



Figura 2: Esquema de la estructura global del sistema. Se ilustra la forma de acceder a un método de segmentación programado en C++ y a otro programado en Matlab.

médicas 2D. Además, la única que hemos encontrado que permite trabajar con cualquier tipo de imagen se apoya en un clasificador que hay que entrenar y que puede no funcionar adecuadamente en algunos casos.

MRSeg está centrado en la segmentación de imágenes 2D y una de sus ventajas principales es que puede funcionar con cualquier método de segmentación automático que sea capaz de generar buenas segmentaciones de las imágenes con las que el usuario desea trabajar. Posteriormente, el usuario será el encargado de etiquetar la imagen para generar la segmentación final de la misma, mediante la unión de las distintas regiones que ha generado el algoritmo automático.

Por tanto, no es necesario usar algoritmos de segmentación interactivos propiamente dichos, es decir, que se basen en la información que proporciona el usuario para realizar la segmentación de la imagen. Esta independencia del algoritmo utilizado para guiar el proceso de segmentación manual proporciona una gran flexibilidad para utilizar métodos de segmentación de propósito general o incluso *ad-hoc* para el tipo de imagen que nos interesa. Además, como se explicó en la sección 3.1, añadir un nuevo método de segmentación resulta sumamente sencillo.

La aplicación actualmente incluye dos métodos de segmentación generales, que se pueden adaptar para funcionar bien en la mayoría de los casos, haciendo uso de las opciones de configuración de MRSeg, tal y como explicaremos en los siguientes apartados. Estos métodos son el *Mean Shift* [1][2] y el *Statistical Region Merging* [7]. La implementación que se ha usado del algoritmo *Mean Shift* es la que se conoce como EDISON, disponible en `http://coewww.rutgers.edu/riul/research/code.html`, además del *wrapper* disponible en `http://www.wisdom.weizmann.ac.il/~bagon/matlab.html` para utilizar EDISON desde Matlab. Para el *statistical region merging* se ha utilizado la implementación disponible en `http://www.lix.polytechnique.fr/~nielsen/`, que ya proporciona un código Matlab. Además, se han creado configuraciones de los distintos métodos que funcionan bien para imágenes a color de retinografías y de citologías.

3.2.1. Interfaz de usuario

La aplicación MRSeg proporciona distintas herramientas para facilitar la labor del usuario. Al acceder al programa, podemos ver el menú y las herramientas que se muestran en la figura 3.

El menú ***Método*** permite elegir qué método de segmentación queremos emplear para realizar la segmentación inicial de la imagen. Tanto los parámetros de los métodos como los tipos de imagen con las que vamos a trabajar se pueden configurar mediante las opciones correspondientes del menú ***Configuración***, tal y como veremos en la sección 3.2.2.

La barra de herramientas situada bajo el menú permite acceder de manera directa a las funciones de ***Abrir***, ***Guardar***, ***Deshacer***, ***Rehacer***, ***Alejar***, ***Acercar*** y ***Recortar***, disponibles también en los menús ***Archivo***, ***Editar*** e ***Imagen***. Estas funciones permiten, respectivamente, abrir una imagen, guardar el resultado de la segmentación, deshacer o rehacer los últimos cambios realizados en la imagen, alejar o acercar la imagen para trabajar más cómodamente y recortar una región de interés por si no queremos trabajar con la imagen completa.

Cuando abrimos una imagen, el programa nos pregunta de qué tipo de imagen se trata y comprueba si ya ha sido utilizada previamente, dando la posibilidad de cargar un recorte o una segmentación ya existente. Si se carga una segmentación previa tenemos la posibilidad de revisarla y corregirla si es necesario. Además, si no hemos terminado de trabajar con una imagen o tenemos dudas y no queremos cerrarla todavía, podemos abrir otra al mismo tiempo gracias a la estructura de pestañas que se ha implementado.

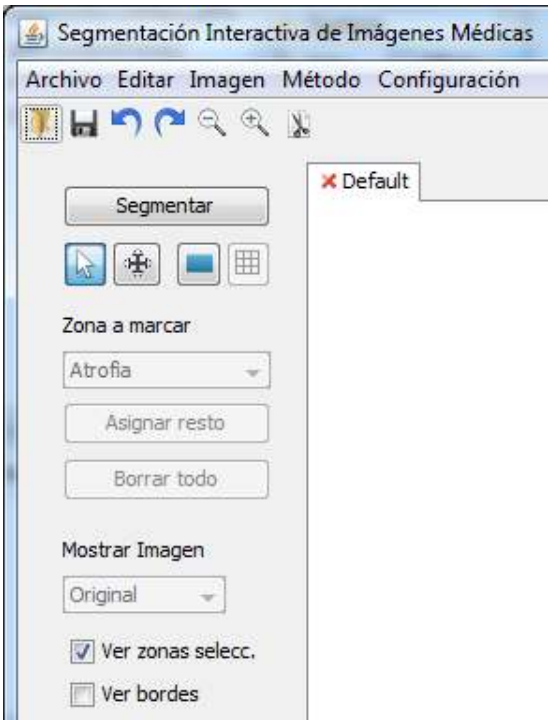


Figura 3: Menú y herramientas de MRSeg.

Debajo de la barra de herramientas están ubicadas las funciones ***Segmentar***, ***Marcar***, ***Mover***, ***Seleccionar*** y ***Refinar***. La función ***Segmentar*** realiza una segmentación automática de la imagen utilizando el método seleccionado en el menú correspondiente. Una vez obtenida esta segmentación inicial, podemos hacer uso de ***Marcar*** para seleccionar las regiones que pertenecen a cada objeto de la imagen, como se explica en la sección 3.2.4. La herramienta ***Mover*** sirve para desplazar la imagen por la pantalla si no cabe completa en la ventana. ***Seleccionar*** nos permite elegir una región de la imagen mediante selección rectangular que luego podremos recortar para trabajar solamente con ella. Por último, la función ***Refinar*** se puede utilizar cuando una región generada por el método de segmentación automático ha unido píxeles pertenecientes a dos objetos distintos, pudiendo así dividir dicha región en partes más pequeñas que podamos marcar por separado.

Bajo estas funciones se encuentra un menú desplegable que permite seleccionar la zona (objeto de interés) que queremos etiquetar de la imagen. El botón ***Asignar resto*** da la posibilidad de asignar a la zona elegida en ese momento el resto de regiones que no han sido etiquetadas todavía. Asimismo, el botón ***Borrar todo*** elimina todas las etiquetas que hayamos asignado a la imagen.

Finalmente, en la parte inferior de la interfaz aparecen distintas opciones de visualización para facilitar el etiquetado de los objetos. En primer lugar, podemos elegir entre mostrar la imagen original o la imagen segmentada por el algoritmo automático (ver sección 3.2.4). También podemos mostrar u ocultar las zonas seleccionadas, es decir, ya etiquetadas por el usuario, así como los bordes correspondientes a estas zonas.

3.2.2. Configuración

Como hemos comentado anteriormente, MRSeg puede ser utilizada por múltiples usuarios y se puede adaptar a distintos tipos de imagen. Para ello, hay que hacer uso de las distintas opciones de configuración de la misma que pasamos a describir a continuación.

En la configuración de usuarios podemos añadir nuevos usuarios al sistema o seleccionar uno distinto al actual. Al cambiar de usuario, la aplicación se encarga automáticamente de almacenar los resultados de manera independiente, con el objetivo de poder crear más de una segmentación de la misma imagen.

Para que MRSeg se pueda adaptar a cualquier tipo de imagen se han creado dos utilidades. Por un lado, la configuración de tipos de imagen permite especificar los objetos de interés que queremos marcar en la imagen, así como sus nombres y los colores con los que se resaltarán en pantalla (ver Fig. 4). Por otro lado, con la configuración de métodos podemos crear distintas combinaciones de parámetros para los métodos de segmentación automáticos que se usan para segmentar inicialmente las imágenes, de manera que produzcan buenos resultados según las características de cada imagen. Es posible crear tantos tipos de imagen y combinaciones de parámetros como se quiera, por lo que podemos configurar la aplicación para trabajar con más de un tipo de imagen al mismo tiempo.

Por último, también se ha traducido la interfaz de usuario a varios idiomas, pudiendo seleccionar el que se desee a través de la ventana de configuración correspondiente.

3.2.3. Estructura de directorios

MRSeg está pensada para crear segmentaciones de referencia de un conjunto de imágenes que, posteriormente, se utilizará para el diseño de nuevos algoritmos o para analizar las imágenes desde el punto de vista médico. En cualquier caso, es recomendable tener almacenado en un mismo directorio el conjunto de imágenes que se quiere segmentar, ya que al abrir una imagen se creará en esa misma ruta un subdirectorio donde se almacenarán los resultados, llamado “manual segmentations”.



Figura 4: Ventana de configuración de los tipos de imagen con los que queremos trabajar.

Dentro de "manual segmentations" se creará a su vez una carpeta por cada imagen segmentada, cuyo nombre será el mismo que el del fichero de la imagen. La segmentación final de cada imagen se almacenará en formato Matlab y guardará una copia de la imagen, que puede estar recortada de la original, la imagen etiquetada por el usuario y la imagen segmentada por el algoritmo automático. Asimismo, en el nombre del fichero de resultados se incluye la información del usuario que etiquetó esa imagen y del método y los parámetros utilizados en la segmentación inicial. De esta manera, basta con copiar el directorio de resultados para usar las segmentaciones de referencia desde otra aplicación.

3.2.4. Ejemplo de segmentación interactiva

Una vez configurado el programa para trabajar con el tipo de imagen que queremos segmentar, los pasos para crear una segmentación interactiva son:

1. Recortar la imagen a la región de interés (opcional).
2. Segmentar la imagen con un algoritmo automático.
3. Marcar las regiones que pertenecen a cada objeto.

En la figura 5 podemos ver un ejemplo de segmentación interactiva de una retinografía, donde se muestran los pasos anteriores, que describiremos en detalle a continuación.

El primer paso es opcional y consiste en seleccionar la región de interés dentro de la imagen, para crear la segmentación de referencia sólo de esta región y no de la imagen completa. Esta selección se realiza mediante un recorte rectangular de la imagen. En el ejemplo de la Fig. 5a hemos recortado la retinografía completa para quedarnos sólo con la zona del nervio óptico.

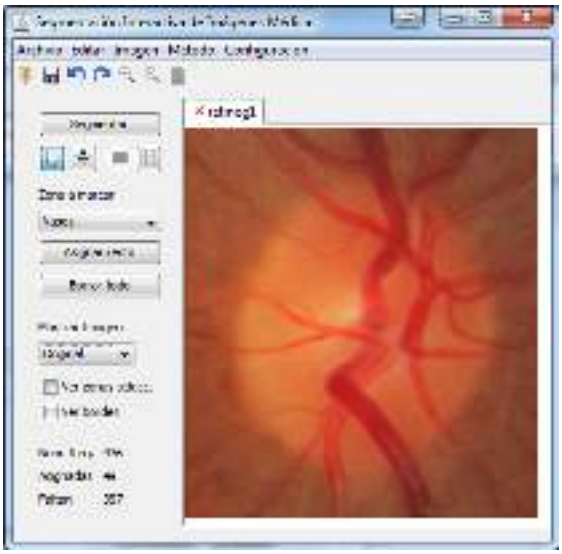
El segundo paso es segmentar la imagen mediante un algoritmo automático. Por lo general, estos algoritmos producirán una imagen sobresegmentada, como la que se muestra en la Fig. 5b obtenida con el *Mean Shift*, configurado previamente para generar buenas segmentaciones de este tipo de imágenes. En este caso hemos intentado que el algoritmo genere las mínimas regiones posibles sin que se produzca el fenómeno de *undersegmentation*.

Una vez obtenida la segmentación inicial de la imagen, el usuario tendrá que seleccionar las regiones que pertenecen a cada una de las clases previamente definidas para ese tipo de imagen, utilizando la herramienta ***Marcar***. En la figura 5c podemos ver el proceso de etiquetado de un objeto. El usuario selecciona la zona de la imagen que quiere etiquetar, en este caso los vasos sanguíneos, y a continuación, mediante el ratón, elige las regiones de la imagen sobresegmentada que pertenecen a los vasos. En amarillo se muestran las regiones ya marcadas y en verde la región que se va marcar a continuación. El proceso es muy sencillo, ya que el algoritmo anterior ha generado unas regiones relativamente grandes y se pueden seleccionar varias rápidamente haciendo clic y arrastrando con el ratón. Si nos equivocamos al marcar alguna, basta con hacer clic o arrastrar con el botón derecho para borrar la selección o con asignarla a otro objeto. Además, si alguna región no ha quedado bien segmentada por el algoritmo y se han unido píxeles pertenecientes a dos objetos distintos, podemos hacer uso de la herramienta ***Refinar*** para dividir esa región en regiones más pequeñas.

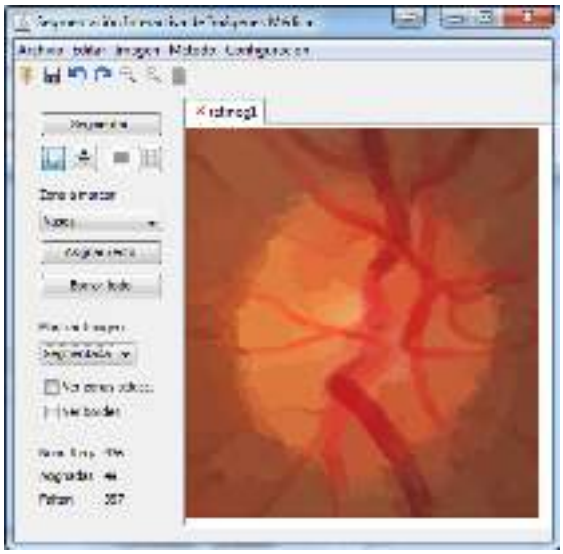
Finalmente, en la figura 5d se muestra la segmentación creada por el usuario donde se han marcado todas las zonas de la retinografía: vasos, excavación, cabeza de nervio óptico y fondo. En este caso, hemos marcado primero todos los objetos menos el fondo, de modo que las regiones pertenecientes al mismo se asignen de manera instantánea usando la función ***Asignar resto***. Una vez etiquetada toda la imagen se utiliza la función ***Guardar*** para añadir este resultado al conjunto de segmentaciones de referencia.

4. CONCLUSIONES Y LÍNEAS FUTURAS

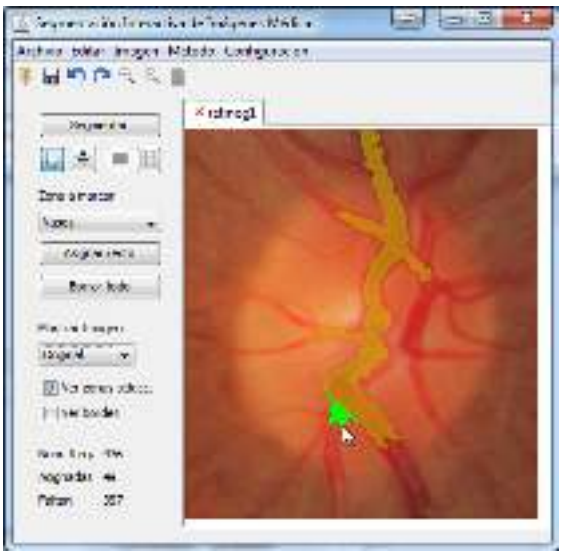
La aplicación MRSeg ha sido probada por distintos usuarios para crear segmentaciones de referencia de retinografías. Este tipo de imágenes pre-



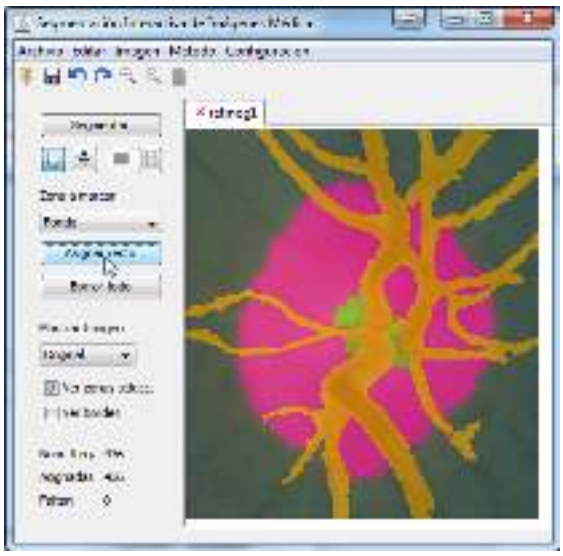
(a) Imagen original a segmentar, previamente recortada de una retinografía.



(b) Imagen sobresegmentada generada por el algoritmo Mean Shift de manera automática.



(c) Proceso de etiquetado de un objeto, en este caso los vasos sanguíneos.



(d) Segmentación final creada por el usuario donde se han marcado todas las zonas de la retinografía.

Figura 5: Ejemplo de segmentación interactiva de una retinografía. Se muestran los pasos principales para crear la segmentación final: recortar la imagen a la zona de interés, segmentar la imagen con un algoritmo automático, seleccionar las regiones que pertenecen a cada objeto y la segmentación resultante.

senta una gran variabilidad entre ellas y contiene un tipo de objeto, los vasos sanguíneos, que resulta difícil de segmentar a mano por otros procedimientos. Además, utilizar herramientas como *ilastik*, revisada en la sección 2, que se apoyan en algoritmos supervisados de propósito general, no es sencillo para usuarios con pocos conocimientos de visión por ordenador, sobre todo si queremos segmentar todas las estructuras de la imagen, a pesar de que los autores de la misma han conseguido buenos resultados segmentando los vasos en las imágenes de la base de datos STARE [9].

Sin embargo, a pesar de la buena acogida que ha tenido la herramienta, algunos usuarios de MRSeg han encontrado dificultades a la hora de etiquetar de manera precisa algunas imágenes. Esto se debe principalmente al algoritmo utilizado para crear las segmentaciones iniciales, que en determinadas circunstancias mezcla en la misma región algunos píxeles pertenecientes a dos objetos de interés distintos, teniendo que hacer uso de la herramienta de refinado que describimos en la sección 3.2.1. Para solucionar este inconveniente, como trabajo futuro se podrían incluir herramientas de corrección más versátiles o fáciles de usar, así como incorporar a la aplicación algoritmos basados en marcadores para crear las segmentaciones iniciales, de modo que sean un poco más precisas e incluso puedan reducir el número de regiones a etiquetar.

Agradecimientos

Este proyecto de investigación ha sido parcialmente financiado por el proyecto PI12/02301 del Fondo de Investigación Sanitaria Instituto Carlos III.

Referencias

[1] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:790–799, 1995.

[2] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[3] R. Dickie, G. Hamarneh, and R. Abugharbieh. Live-vessel: Interactive vascular image segmentation with simultaneous extraction of optimal medial and boundary paths. Technical Report TR 2009-23, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, November 2009.

[4] G. Hamarneh, R. Abugharbieh, and A. Top. Turtleseg: Interactive 3d image segmentation software (http://www.turtleseg.org/).

[5] G. Hamarneh, J. Yang, C. McIntosh, and M. Langille. 3d live-wire-based semi-automatic segmentation of medical images. In *SPIE Medical Imaging*, volume 5747, pages 1597–1603, 2005.

[6] D. Maleike, M. Nolden, H.-P. Meinzer, and I. Wolf. Interactive segmentation framework of the medical imaging interaction toolkit. *Computer Methods and Programs in Biomedicine*, 96(1):72 – 83, 2009.

[7] R. Nock and F. Nielsen. Statistical region merging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26:1452–1458, November 2004.

[8] M. Poon, G. Hamarneh, and R. Abugharbieh. Live-vessel: Extending livewire for simultaneous extraction of optimal medial and boundary paths in vascular images. In *Lecture Notes in Computer Science, Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 444–451, 2007.

[9] C. Sommer, C. Straehle, U. Koethe, and F. A. Hamprecht. ïlastik: Interactive learning and segmentation toolkit". In *8th IEEE International Symposium on Biomedical Imaging (ISBI 2011)*, 2011.

[10] A. Top, G. Hamarneh, and R. Abugharbieh. Spotlight: Automated confidence-based user guidance for increasing efficiency in interactive 3d image segmentation. In *Medical Image Computing and Computer-Assisted Intervention Workshop on Medical Computer Vision (MICCAI MCV)*, pages 204–213, 2010.

[11] A. Top, G. Hamarneh, and R. Abugharbieh. Active learning for interactive 3d image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 6893 of *LNCS*, pages 603–610. Springer Berlin / Heidelberg, 2011.

[12] I. Wolf, M. Vetter, I. Wegner, T. Böttger, M. Nolden, M. Schöbinger, M. Hastenteufel, T. Kunert, and H.-P. Meinzer. The medical imaging interaction toolkit. *Medical Image Analysis*, 9(6):594 – 604, 2005. ITK. Open science - combining open data and open source software: Medical image analysis with the Insight Toolkit.

[13] P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

ANÁLISIS ESPECTRAL NO LINEAL DEL EEG DE NIÑOS CON EPILEPSIA INTRATABLE

O. Portolés[1,2], R. Schroeder[2], M. Vallverdú[1], A. Voss[2], P. Caminal[1]

[1]Dept. ESAII, CREB, CIBER-BBN, Universitat Politècnica de Catalunya
[2]Department of Medical Engineering and Biotechnology, University of Applied Sciences Jena, Alemania

Resumen

En este estudio se analizó el espectro de potencia de señales EEG grabadas en niños con epilepsia intratable. Se estudiaron tres segmentos de la señal EEG: un segmento anterior al ataque, un segmento del inicio del ataque y un segmento posterior al ataque epiléptico. De cada segmento se extrajo una serie de índices lineales y no lineales del espectro de potencia. Estos índices se usaron para construir un clasificador para discriminar entre los tres segmentos. Se estudiaron independientemente los ataques epilépticos correspondientes al sujeto dormido y despierto. El espectro de potencias se calculó usando tres métodos diferentes: periodograma de Welch y el método autoregresivo de Burg, estimando el orden óptimo mediante el criterio de Akaike y el de Bayes. El espectro de potencias se obtuvo de la señal original, la señal filtrada entre 0.5 y 25 Hz y entre 0.5 y 40 Hz. Los mejores resultados de clasificación se obtuvieron de los índices que describen estadísticamente la morfología del espectro, la energía de los ritmos cerebrales y la dimensión fractal de Higuchi aplicada a una curva extraída del espectro.

Palabras Clave: Electroencefalograma, espectro de potencia, análisis de la dinámica no lineal, complejidad.

1 Introducción

La epilepsia es actualmente un tema objeto de muchos estudios. Esto es debido a varios factores entre los que se encuentra la alta tasa de afectados en el mundo, estimada en más de 50 millones de personas. Además, todavía es una incógnita su origen, es complejo diagnosticarla y los tratamientos actuales son de reducida eficiencia presentando efectos secundarios indeseados.

Uno de los campos de investigación es predecir mediante el análisis de la señal electroencefalográfica (EEG), u otros dispositivos, que un ataque epiléptico se va a producir. Se pretenden crear dispositivos que puedan predecir ataques epilépticos con una cierta antelación [8]. De tal modo que dicho dispositivo pueda alertar al enfermo de que va a sufrir un ataque epiléptico en breve, o dispensar una dosis de un fármaco neuroestimulador desde un dispositivo implantable que revoque el ataque antes de que se produzca [10]. Además, sería de gran utilidad en el diagnóstico y caracterización de la epilepsia porque se podría administrar al paciente un isótopo trazador justo antes de sufrir una crisis y obtener una imagen tomográfica en el momento del ataque [2]. Otro campo de investigación es el diagnóstico de la epilepsia y sus tipos. Cada tipo de epilepsia se manifiesta de manera diferente en la señal EEG, en función del estado cerebral del paciente, y para ello se apoyan en el análisis de señales EEG de larga duración [6].

Se ha demostrado a través de investigaciones que el proceso entre el estado normal y el ataque epiléptico no es aleatorio sino que hay una transición dinámica [7]. En este sentido, han sido definidos a lo largo de un ataque epiléptico varios parámetros que describen el comportamiento estático de la señal en un intervalo de tiempo [1], como la energía de la señal o los cambios frecuenciales. Sin embargo, dado que la señal EEG tiene un comportamiento caótico con una dinámica no lineal, el análisis de estas señales con técnicas que midan su complejidad puede mejorar la detección de ataques epilépticos.

En el presente estudio se analizará el espectro de potencia de la señal EEG de niños con ataques epilépticos, previo a un ataque epiléptico, durante el ataque y después de dicho ataque. También se estudiarán los registros en los que el paciente estaba durmiendo y en los que estaba despierto. Para ello se obtendrán índices que describan linealmente el espectro de potencia como son la energía de la señal y

los cambios frecuenciales. Además se analizarán las medidas no lineales de la irregularidad del espectro de potencia, como la entropía de Shannon y la dimensión fractal de Higuchi. Por último se evaluará la capacidad de los índices estudiados para discriminar entre el segmento anterior a un ataque, durante el ataque y al final del ataque; estando el paciente dormido o despierto. Estos análisis se aplicarán a las señales EEG, sin filtrar y a las señales filtradas según dos tipos de filtro.

2 MÉTODOS

2.1 DATOS ANALIZADOS

Las señales EEG analizadas pertenecen a la base de datos Physiobank, la cual contiene señales bien caracterizadas y anotadas [3]. Para este estudio se ha usado una colección de registros EEG superficiales registrados en niños entre 2 y 9 años que sufren algún tipo de epilepsia intratable. En total cinco pacientes cumplían los criterios para ser analizados. Estos criterios son la duración de los ataques epilépticos, el tiempo entre ataques y la distancia del ataque epiléptico respecto al inicio o al final del registro EEG. De modo que para cada ataque epiléptico se obtuvieran tres segmentos de la señal EEG relativos al ataque Además el sujeto debía cumplir estos requisitos como mínimo para un ataque sufrido mientras dormía y otro mientras estaba despierto. Los sujetos con los que se ha realizado el presente estudio corresponden a los códigos de la base de datos: chb 05, chb 08, chb 10, chb 12 y chb 22.

Todas las señales fueron registradas con una frecuencia de muestreo de 256 Hz. usando un conversor analógico/digital con 16 bits de resolución. En total son 23 canales distribuidos siguiendo el estándar internacional 10 – 20 y montados de manera bipolar. Además, siguiendo los criterios de análisis, de cada ataque epiléptico se obtienen tres segmentos (antes, durante y después del ataque epiléptico) de 8096 muestras cada uno. El primer segmento, anterior al ataque epiléptico, se tomó 5 segundos antes del comienzo de dicho ataque. El segundo segmento se tomó 5 segundos después del comienzo del ataque y el tercer segmento se tomó 5 segundos después de haberse finalizado el ataque epiléptico (ver figura 1).

En este trabajo se realizan tres estudios con diferente nivel de filtrado:

1) Datos originales sin filtrar.
2) Filtrado pasabanda con un filtro IIR Butterworth de orden 8 entre 0.5 y 25 Hz. Rango frecuencial en el que se producen la mayoría de los cambios en la actividad cerebral durante un ataque epiléptico [4].
3) Filtrado pasabanda con un filtro IIR Butterworth de orden 8 entre 0.5 y 40 Hz. Rango frecuencial en que se encuentra la actividad cerebral [4].

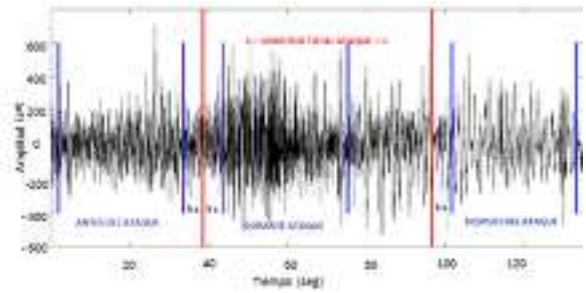


Figura 1: Electroencefalograma (EEG) donde se indica el ataque epiléptico entre barras rojas y entre barras azules están indicados los tres segmentos analizados: antes, durante y después del ataque epiléptico.

2.2 METODOLOGÍA

Para cada segmento de señal se obtuvieron las componentes frecuenciales mediante el periodograma de Welch y el método autorregresivo de Burg, usando los criterios de información de Akaike (IAC) y Bayes (BIC) para estimar el orden óptimo del modelo. El periodograma de Welch es una técnica no paramétrica que permite estimar la densidad espectral de potencia contenida en una señal $y(n)$. Esta señal se divide en k segmentos de M puntos (M = 1024 muestras) solapados el 25%, a los que se les aplica la transformada rápida de Fourier (*Fast Fourier Transform*), (1). En este caso se ha usado una ventana tipo *Hanning* para mejorar la resolución del espectro de potencia.

$$P_y^i(\omega) = \frac{1}{M} \left| \sum_{n=0}^{M-1} y(n) e^{-i\omega n} \right|^2 \qquad (1)$$

El algoritmo de Burg es un método paramétrico autorregresivo mediante el cual también se puede calcular la densidad espectral de potencia de una señal. Para ello se modela la señal como una regresión lineal de sí misma más un ruido aleatorio $u(n)$ con media nula y una cierta varianza (2), donde $a(k)$ son los parámetros que deben ser estimados por el modelo (3) y ρ es el orden del modelo autorregresivo que también debe ser estimado buscando un compromiso entre la resolución espectral y la aparición de picos espurios. En este caso el orden es estimado mediante los criterios de información de Akaike (IAC), (4), y Bayes (BIC), (5), donde $\sigma_{e_p}^2$ es la varianza del error de predicción. La estimación de los coeficientes de reflexión se ha realizado mediante la ecuación (3)

donde k_m son los coeficientes de reflexión y e^f y e^b son los errores de predicción lineal hacia delante y hacia atrás.

$$y(n) = \sum_{k=0}^{\rho} a(k)y(n-k) + u(n) \quad (2)$$

$$k_m = \frac{-E\left\{e^f_{m-1}(n)e^b_{m-1}\cdot(n-1)\right\}}{\sqrt{E\left\{\left|e^f_{m-1}(n)^2\right|\right\}}\sqrt{E\left\{\left|e^b_{m-1}(n)^2\right|\right\}}}, \quad 1 \le m \le \rho \quad (3)$$

$$\mathcal{M}_{AIC}(\rho) = N \ln \sigma^2_{\hat{e}_p} + 2\rho \quad (4)$$

$$\mathcal{M}_{BIC}(\rho) = \ln \sigma^2_{\hat{e}_p} + \rho \cdot \ln N \quad (5)$$

En la mayoría de los ataques epilépticos, calculados mediante el periodograma de Welch, se observó que el espectro de potencia del segmento tomado 5 segundos tras el comienzo de una crisis era consistentemente más abrupto e irregular, tal como se observa en la figura 2.

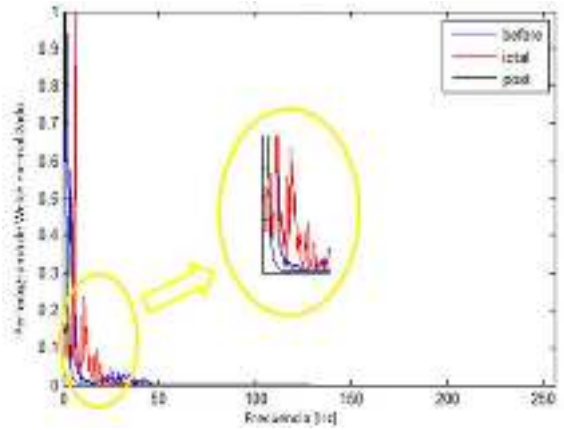


Figura 2: Comparación del espectro de potencia de los segmentos: anterior (azul), durante (rojo) y posterior (negro) a un ataque epiléptico, calculado mediante el periodograma de Welch.

Con la finalidad de caracterizar la irregularidad del espectro de potencia se siguieron dos metodologías. Una de ellas consistió en caracterizar la morfología de los picos del espectro de potencia contenidos entre ciertos rangos frecuenciales. El segundo método consistió en ajustar una curva suavizada al espectro de potencia que siguiera la tendencia de la curva del espectro pero que no se viera influenciada por los picos e irregularidades presentes en el espectro de potencia, como se muestra en la figura 3. Para calcular la curva ajustada suavizada se aplicó un filtro de tipo *moving-average* (6) en ventanas de N = 15 muestras de longitud sobre el espectro de potencia $z(n)$, comprobándose que era el mejor ajuste.

$$z(n) = \frac{1}{2N+1}\sum_{j=-N}^{N} P_y\ (\omega)(n+j) \quad (6)$$

A continuación se substrajo la curva del espectro de potencia a la curva ajustada a dicho espectro, figura 3. Sobre esta curva resultante se analizó la irregularidad, o complejidad, del espectro de potencia mediante técnicas de análisis no lineales.

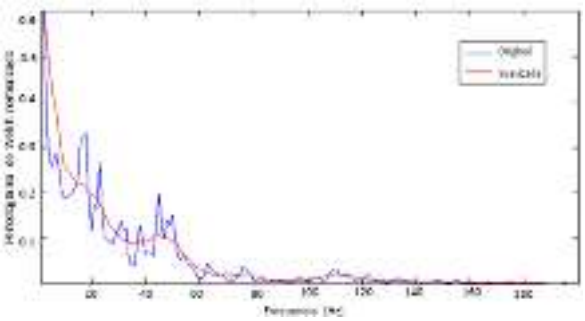


Figura 3: Curva suavizada (rojo) del espectro de potencia (azul).

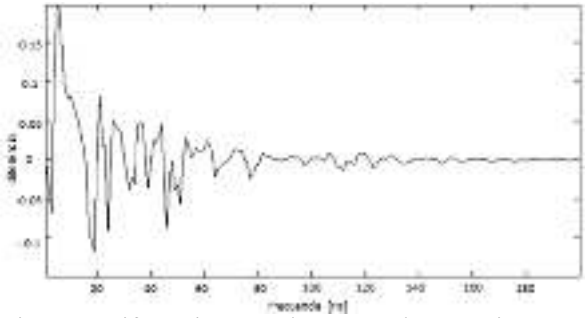


Figura 4: Diferencia entre el espectro de potencia y su curva suavizada.

2.2.1 Descriptores del espectro

Para caracterizar el espectro de potencia se calcularon una serie de índices estadísticos que permiten describir la morfología del espectro. Estos índices son:

Frecuencia central (*Fc*): Frecuencia que deja la mitad de la energía del espectro de potencia a un lado y la otra mitad de la energía al otro lado.

Frecuencia media (*Fm*): Media de todas las frecuencias presentes en el espectro.

Varianza (*var*): varianza de la frecuencia respecto a la media.

Frecuencia de pico (*Fp*): Frecuencia donde se encuentra la máxima potencia.

Frecuencia máxima (*Fmax*): Frecuencia por debajo de la cual está el 95% de la energía.

Cuartil 1 (*Q*1): Frecuencia por debajo de la cual está el 25% de la energía.

Cuartil 3 (*Q*3): Frecuencia por debajo de la cual está el 75% de la energía.

Rango intercuartilico (*RI*): *Q*1 menos *Q*3.

Coeficiente de asimetría de Pearson (*pearson*).

Coeficiente de apuntalamiento (*curtosis*).

Estos descriptores del espectro de potencia se calcularon para todos los segmentos seleccionados para el estudio, descritos anteriormente.

2.2.2 Bandas de energía

Para cada segmento estudiado se calculó la energía de cada ritmo cerebral característico del EEG. Estos son el ritmo delta (δ) de 0.5 a 4 Hz, teta (θ) de 4 a 8 Hz, alfa (α) de 8 a 13.5 Hz, beta (β) de 13.5 a 30 Hz y gamma (γ) superior a 30 Hz. Estos cálculos de energía se realizaron sobre el espectro de potencia, así como sobre la curva suavizada del espectro de potencia previamente explicada.

Además, estas 5 bandas frecuenciales (δ, θ, α, β, γ) se combinaron para analizar a otras 12 bandas, tal como se indica a continuación:

$V_1=\delta+\theta$
$V_2=\delta+\theta+\alpha$
$V_3=\delta+\theta+\alpha+\beta$
$V_4=\delta+\theta+\alpha+\beta+\gamma$
$V_5=\theta+\alpha$
$V_6=\theta+\alpha+\beta$
$V_7=\theta+\alpha+\beta+\gamma$
$V_8=\alpha+\beta$
$V_9=\alpha+\beta+\gamma$
$V_{10}=\delta+\alpha$
$V_{11}=\delta+\alpha+\beta$
$V_{12}=\delta+\beta$

Por tanto, en total se estudiaron 34 índices relacionados con la energía de los ritmos cerebrales, 17 extraídos del espectro de potencia y otros 17 del espectro de potencia suavizado.

2.2.3 Irregularidad del espectro de potencia

Tal como ha sido explicado previamente, el espectro de potencia de los segmentos durante la crisis epiléptica es más irregular y abrupto. Para caracterizar dicha irregularidad se la ha maximizado calculando la curva resultante de substraer el espectro de potencia y su curva suavizada, figure 4. Sobre esta curva se ha aplicado la entropía de Shannon que permite cuantificar la regularidad de una señal y el grado de dispersión de sus puntos [9]. También se ha calculado sobre la misma curva la dimensión fractal de Higuchi que proporciona un índice de la complejidad de una señal [5].

El algoritmo de Higuchi estima la dimensión fractal (D_f) de una curva calculando la longitud media de la curva, $L(k)$, usando un segmento de k muestras como unidad de medida. De modo que $L(k) \approx k^{-D_f}$.

El algoritmo realiza los siguientes pasos para estimar la dimensión fractal de Higuchi:

Paso 1: A partir de una serie temporal $y(i),\ldots, y(N)$, donde $i = 1,2,\ldots,N$ muestras, se construyen nuevas series temporales y_k^m desde $k = 1$ hasta k_{max} definidas como:

$$y_k^m: y(m), y(m+k),\ldots, y\left(m+\left[\frac{N-m}{k}\right]\cdot k\right) \quad (7)$$

con $m = 1,2,\ldots,k$

Paso 2: Se calcula la longitud $L_m(k)$ de cada serie temporal y_k^m de la siguiente manera:

$$L_m(k) = \left[\left(\sum_{i=1}^{M} |y(m+i\cdot k) - y(m+(i-1)\cdot k)|\left(\frac{N-1}{M*k}\right)\right)\right]\cdot k^{-1} \quad (8)$$

con $M = int\left(\frac{N-m}{k}\right)$.

El término $(N-1)\left(int\left(\frac{N-m}{k}\right)\cdot k\right)^{-1}$ es un factor de normalización de la longitud de la curva y_k^m.

Paso 3: La longitud de la serie temporal para cada intervalo k, $\langle L_m(k)\rangle$, se calcula como la media de los k valores de $L_m(k)$, con $m = 1, 2,\ldots, k$.

$$\langle L_m(k)\rangle = \frac{1}{k}\sum_{m=1}^{k} L_m(k) \quad (9)$$

Paso 4: Si $\langle L_m(k)\rangle \approx k^{-D_f}$, entonces la serie temporal tendrá una dimensión fractal D_f. En tal caso, el grafico de $ln(L_m(k))$ vs. $ln(k)$, será una línea recta decreciente cuya pendiente es igual a $-D_f$. Por tanto, la dimensión fractal puede calcularse usando el mejor ajuste realizado mediante el método de mínimos cuadráticos.

2.3 ANÁLISIS ESTADÍSTICO

A partir de los índices propuestos en este trabajo, se realiza un estudio estadístico con el fin de obtener aquellos índices que permitan discriminar entre el segmento anterior a un ataque, durante la propia crisis y después de la crisis: antes vs. durante; antes vs. después; durante vs. después.

En primer lugar se aplica la prueba de los signos de Wilcoxon, test que permite determinar si existen diferencias entre dos grupos de muestras relacionadas. Para este estudio, el nivel de significación estadística fue *p-valor* < 0.05. Para solventar el problema asociado a las comparaciones múltiples se ha aplicado la corrección de Bonferroni (10).

$$p_valor \text{ corregido } = \frac{p_valor}{N^{o} hipotesis} \qquad (10)$$

Los índices que permiten discriminar entre los segmentos con un *p-valor* menor al establecido por la corrección de Bonferroni se introducen en un algoritmo de clasificación empleando una función discriminante cuadrática. Para validar la clasificación realizada se usó la técnica de validación cruzada dejando uno fuera (LOOCV, *leave-one-out cross-validation*).

3 RESULTADOS Y DISCUSIÓN

Al comparar los espectros de potencia obtenidos mediante el periodograma de Welch y el método autoregresivo de Burg, estimando el orden optimo del modelo mediante los métodos de Akaike y Bayes, se ha observado que en muchos de los ataques epilépticos el espectro de potencia calculado mediante el periodograma de Welch es más abrupto e irregular. Estos resultados se pueden apreciar en la figura 5. Además, como ya ha sido explicado anteriormente, cuando se calcula el espectro de potencia de cada uno de los tres segmentos analizados mediante el periodograma de Welch, se observa que el segmento durante el ataque epiléptico es más abrupto e irregular que los segmentos anterior y posterior al ataque epiléptico (figura 2). Esta mayor irregularidad en el periodograma de Welch ha permitido que los índices extraídos de éste hayan caracterizado mejor al espectro de potencia. Por ello, todos los índices presentados en estos resultados se han obtenido a partir del espectro calculado mediante el periodograma de Welch.

Debido a la gran cantidad de resultados, se ha establecido un umbral mínimo a partir del cual un resultado va a ser considerado suficientemente satisfactorio como para ser valorado en el presente estudio. Este umbral mínimo está establecido por dos parámetros. Uno de ellos es la proporción de datos usados en el proceso de entrenamiento del clasificador que son incorrectamente clasificados (falsos positivos + falsos negativos). El segundo parámetro es la media de los datos mal clasificados (falsos positivos + falsos negativos) tras aplicar el método LOOCV a todos valores usados en la construcción del clasificador. Por tanto, en este estudio solamente se valorarán los índices que han obtenido al menos en 10 de los 23 canales un error de clasificación menor al 40 %, para ambos parámetros considerados.

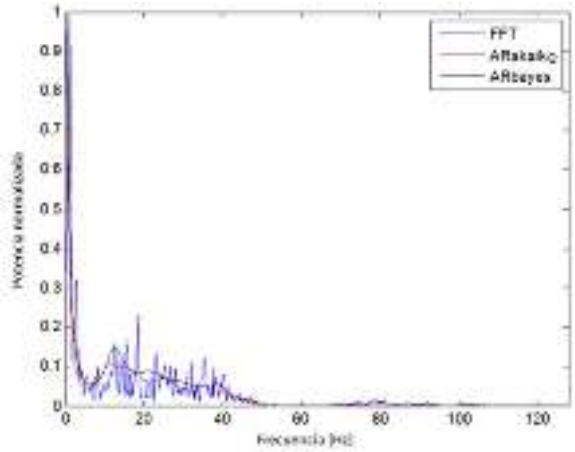


Figura 5: Espectro de potencia de un ataque epiléptico calculado mediante periodograma de Welch y método autoregresivo de Burg, estimando el orden óptimo mediante el criterio de Akaike o Bayes.

La característica común de todos los sujetos analizados es que sufren de epilepsia intratable, siendo además desconocida la localización del foco epiléptico. Por lo que se asume que cada sujeto puede tener un foco epiléptico diferente. Lo que conlleva que las variaciones del espectro de potencia estén distribuidas de manera irregular entre pacientes. Éste es el motivo por el que los resultados se presenten según el número total de canales en vez de comparar un determinado canal entre sujetos.

Cabe destacar que varios índices han podido clasificar entre: segmentos anteriores al ataque y segmentos posteriores al ataque o durante ataque (tablas 1 a 4). Así mismo, varios índices han podido clasificar los tres segmentos al ser comparados a la vez: segmentos durante el ataque, segmentos posteriores al ataque y los segmentos durante el ataque (tabla 5). Además, se observan diferencias a la hora de clasificar los segmentos de una misma crisis según el sujeto está

dormido o despierto. Por lo general, los resultados de clasificación de los segmentos cuando el sujeto está dormido son mejores que cuando la crisis epiléptica ocurre mientras el sujeto está despierto (ver tablas 1 a 5).

En cuanto a los tres estudios realizados con las señales filtradas y no filtradas (tal como se ha descrito en el apartado 2.1), sólo se aprecian ligeras diferencias en algunos de los índices Pero no se observa que sigan un patrón determinado.

A continuación se describen primero los índices que han conseguido los mejores resultados de clasificación comparando dos segmentos (tablas 1 a 4). Por último se comentan los resultados obtenidos cuando una misma variable caracteriza y clasifica los tres segmentos a la vez (tabla 5).

3.1 DESCRIPTORES DEL ESPECTRO DE POTENCIA

Entre los diez índices propuestos para caracterizar el espectro de frecuencias estadísticamente, los índices que han conseguido los mejores resultados como clasificadores de los tres segmentos relativos al ataque epiléptico son: el primer cuartil ($Q1$), el tercer cuartil ($Q3$) y la frecuencia máxima ($Fmax$). Estos índices permitieron discriminar correctamente entre dos segmentos (tabla 1), y también los tres segmentos pudiéndose usar estos índices para construir un clasificador (tabla 5).

Tabla 1: Número de canales estadísticamente bien clasificados al comparar dos tipos de segmentos usando los índices descriptores del espectro de potencia con un error menor al 40% al aplicar LOOCV.

Índice	**antes-durante**			**antes-después**			**durante-después**		
filtro [Hz]	**sin filt.**	**0.5 /25**	**0.5 /40**	**sin filt.**	**0.5 /25**	**0.5 /40**	**sin filt.**	**0.5 /25**	**0.5 /40**
Crisis estando dormidos									
Fc	-	-	-	-	-	-	-	20	-
Fmax	11	9	10	21	21	18	16	15	-
*Q*1	18	11	17	20	23	23	21	23	-
*Q*3	19	22	13	6	6	-	12	20	-
Crisis estando despiertos									
Fmax	7	6	-	21	16	18	-	-	-
*Q*1	20	8	16	23	21	13	-	-	-
*Q*3	18	18	20	14	12	13	-	-	-

Las tablas 1 y 5 muestran el número de canales que fueron correctamente clasificados con un error menor al 40% al aplicar LOOCV al clasificador. Estos tres índices presentan resultados similares indistintamente de que las crisis tuvieran lugar mientras los sujetos estuvieran despiertos o dormidos. Sin embargo, ninguno de los índices fue capaz de discriminar con un error menor al 40% entre los segmentos durante y después de una crisis en los sujetos despiertos (ver tabla 1).

3.2 ENERGÍA BAJO LA CURVA SUAVIZADA DEL ESPECTRO DE POTENCIA

En el caso de los índices extraídos de la curva suavizada del espectro de potencia, se puede observar en la tabla 2 que el comportamiento de los índices es diferente según las crisis tengan lugar mientras los sujetos están despiertos o dormidos. Un menor número de índices han permitido discriminar entre dos segmentos en el caso de que la crisis tenga lugar mientras el paciente está despierto. Esto puede ser debido a la variabilidad de la actividad cerebral mientras el sujeto está despierto debido a la diversidad de actividades o procesos en los que el cerebro está envuelto.

Los índices con mejor comportamiento hacen referencia a los ritmos cerebrales θ, α, β y sus combinaciones. Se observa, en las crisis epilépticas mientras el sujeto está despierto, que el índice del ritmo β no discrimina entre el segmento anterior y el segmento durante la crisis epiléptica, mientras que sí lo hace en los otros casos (ver tabla 2).

Tabla 2: Número de canales estadísticamente bien clasificados al comparar dos tipos de segmentos usando los índices energía bajo la curva suavizada del espectro de potencia con un error menor al 40% al aplicar LOOCV.

Índice	**antes-durante**			**antes-después**			**durante-después**		
filtro [Hz]	**sin filt.**	**0.5 /25**	**0.5 /40**	**sin filt.**	**0.5 /25**	**0.5 /40**	**sin filt.**	**0.5 /25**	**0.5 /40**
Crisis estando dormidos									
α	18	-	-	-	15	15	23	-	-
β	-	-	-	4	9	-	17	14	14
V_5	15	18	18	11	14	14	23	23	23
V_6	13	14	-	10	11	-	19	19	19
V_7	-	-	-	-	-	-	11	-	-
V_8	10	-	-	-	-	-	-	-	-
Crisis estando despiertos									
α	-	-	-	-	-	-	23	20	22
β	-	15	15	3	-	-	-	-	-
V_5	-	-	-	-	-	-	-	-	19
V_6	-	-	14	-	-	-	-	-	-
V_7	-	-	-	-	-	-	11	-	-
V_8	-	-	-	-	-	-	-	-	-

3.3 ENERGÍA BAJO LA CURVA DEL ESPECTRO DE POTENCIA

Los resultados obtenidos por estos índices son muy similares a los obtenidos con la curva suavizada del espectro de potencia, tal y como cabía esperar. La tabla 3 muestra estos resultados.

Tabla 3: Número de canales estadísticamente bien clasificados al comparar dos tipos de segmentos usando los índices de energía bajo la curva del espectro de potencia con un error menor al 40% al aplicar LOOCV.

Índice	**antes-durante**			**antes-después**			**durante-después**		
filtro [Hz]	**sin filt.**	**0.5 /25**	**0.5 /40**	**sin filt.**	**0.5 /25**	**0.5 /40**	**sin filt.**	**0.5 /25**	**0.5 /40**
Crisis estando dormidos									
α	19	-	-	-	14	14	23	-	-
β	-	-	-	-	-	-	17	14	14
V_5	17	-	-	14	12	12	23	23	23
V_6	14	14	14	-	10	12	19	19	19
V_7	-	-	-	-	-	-	11	-	-
V_8	9	-	-	-	6	-	-	-	-
Crisis estando despiertos									
α	-	18	18	-	-	-	-	-	-
β	-	12	12	-	-	-	-	-	-
V_2	-	-	-	-	-	-	-	13	13
V_5	-	-	-	-	-	-	-	-	-
V_6	-	-	-	-	-	-	-	-	-
V_7	-	-	-	-	-	-	-	-	-
V_8	-	-	-	-	-	-	-	-	-

3.4 IRREGULARIDAD DEL ESPECTRO DE POTENCIA

El índice que ha mostrado un mejor comportamiento para cuantificar la irregularidad del espectro de potencia ha sido la dimensión fractal de Higuchi. Este índice ha conseguido discriminar entre los segmentos anteriores a un ataque epiléptico y los segmentos tomados durante el ataque (ver tabla 4), aunque solo para aquellos segmentos correspondientes a sujetos estando despiertos en el momento de la crisis. Al contrario de lo que sucedía con los índices relativos a la energía de los ritmos cerebrales. Este resultado es coherente con el hecho de que el espectro de potencia de los segmentos anteriores a un ataque se observó más irregular y abrupto, ver figura 2.

Tabla 4: Número de canales estadísticamente bien clasificados al comparar dos tipos de segmentos, antes y durante un ataque epiléptico usando como índice la dimensión fractal de Higuchi con un error menor al 40% al aplicar LOOCV.

Índice	**antes – durante**		
filtro [Hz]	**sin filtrar**	**0.5 / 25**	**0.5 / 40**
Crisis estando despiertos			
Higuchi (k_{max} = 4)	12	12	-
Higuchi (k_{max} = 5)	-	14	-

3.5 DISCRIMINACIÓN ENTRE LOS TRES SEGMENTOS

En la tabla 5 se puede observar como algunos de los mismos índices que permitían discriminar entre dos segmentos, también pueden discriminar entre los tres segmentos y construir así un clasificador (antes, durante y después de un ataque epiléptico). Del mismo modo, se observa que los índices relativos a la energía de los ritmos cerebrales tienen un mejor comportamiento cuando los sujetos están dormidos en el momento de la crisis epiléptica.

Tabla 5: Número de canales estadísticamente bien clasificados por un mismo índice en los tres segmentos con un error menor al 40% al aplicar LOOCV.

Índice	**Crisis estando despiertos**			**Crisis estando dormidos**		
filtro [Hz]	**sin filt.**	**0.5/25**	**0.5/40**	**sin filt.**	**0.5/25**	**0.5/40**
Fmax	23	23	23	22	21	23
*Q*1	18	21	18	11	19	23
*Q*3	13	14	13	23	23	23
Energía bajo el espectro de potencia						
α	-	-	-	23	9	9
β	-	-	-	17	-	3
V_5	-	-	-	23	5	5
V_6	-	-	-	19	-	6
V_8	-	-	-	8	-	-
Energía bajo la curva suavizada del espectro de potencia						
α	-	-	-	23	9	9
β	-	-	-	13	9	9
V_5	-	-	-	22	-	9
V_6	-	-	-	21	-	-
V_8				7	-	-

4 CONCLUSIONES

Este trabajo consiste en un estudio preliminar realizado en niños que sufren epilepsia intratable. En él se ha analizado la variabilidad de los espectros de potencia de tres segmentos de señal característicos de los cambios que se producen durante un ataque epiléptico. En este estudio se ha diferenciado entre crisis sufridas mientras el sujeto estaba despierto o dormido. El estudio se ha realizado en señales EEG filtradas y no filtradas, pero no ha aportado diferencias entre los resultados. Se han definido diversos índices que han permitido caracterizar el espectro de potencia en términos de frecuencia, energía e irregularidad del espectro, basados en técnicas de medida lineales y no lineales. Los resultados obtenidos de los parámetros frecuenciales y energéticos extraídos del espectro de potencia muestran que se produce un desplazamiento del rango de frecuencias y energías predominantes en el espectro de potencia para cada una de las tres etapas

de un ataque epiléptico. Esto ha permitido que ciertos índices discriminen y clasifiquen los segmentos correctamente. Además, mediante la dimensión fractal de Higuchi, este estudio ha permitido demostrar que el espectro de potencia tiene distinto nivel de irregularidad cuando se comparan los segmentos anteriores y del comienzo de un ataque epiléptico.

Agradecimientos

Este trabajo ha sido realizado en el marco de una CICYT TEC2010-20886, un CIBER de Bioingenieria, Biomateriales y Nanomedicina (CIBER-BBN) que es una iniciativa de ISCIII, y una EU Marie Curie Initial Training Networks (ITN) Call FP7-PEOPLE-2011-ITN.

Referencias

[1] Aarabi, A., Fazel-Rezai, R., Aghakhani, Y., (2009) EEG Seizure Prediction: Measures and Challenges, 31st Annual International Conference of the IEEE EMBS, pp. 1864-1867.

[2] Cysyk, B. and Sepkuty, J., Lesser R., Civelek, A., Truly ictal SPECT is of major importance for reliable localization of seizure focus. *Epilepsia*, nº 38 (Supplement 8):146.

[3] Goldberger, A.L., Amaral, L., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G.,Mietus, J.E., Moody. G.B., Peng, C-K., Stanley, H.E., (2000) PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101(23):e215-e220.

[4] Gotman, J., Ives, J.R., and Gloor, P., (1981) Frequency Content of EEG and EMG at Seizure Onset: Possibility of Removal of EMG Artefact by Digital Filtering. *Electroencephalography and Clinical Neurophysiology*. vol. 52, no. 6. pp. 626-639.

[5] Higuchi, T., (1988) Approach to an Irregular Time Series on the Basis of the Fractal Theory. *Physica*.. Vol. 31. pp. 277-283.

[6] Hoppe, C., Poepel, A. and Elger, C.E., (2007) Epilepsy: Accuracy of Patient Seizure Counts. *Archives of Neurology*, vol. 64, no. 11. pp. 1595-1599.

[7] Mormann, F., Kreuz, T., Rieke, C., Andrzejak, R. G., Kraskov, A., David, P., Elger, C. E., and Lehnertz, K., (2005) "On the predictability of epileptic seizures", *Clin Neurophysiol*, vol. 116, pp. 569-87.

[8] Schelter, B., Timmer, J., Schulze-Bonhage, A., (2008), Seizure Prediction in Epilepsy: From Basic Mechanisms to Clinical Applications.

[9] Shannon, C.E., (1948) A mathematical theory of communication. *Bell System Technical Journal*. Vol. 27, pp. 379-423.

[10] Theodore, W.H. and Fisher, R.S., (2004) Brain Stimulation for Epilepsy. Lancet Neurology, vol. 3, no. 2. pp. 111-118.

EVALUACIÓN DE MÉTODOS DE SEGMENTACIÓN DE PROPÓSITO GENERAL SOBRE IMÁGENES DE FONDO DE OJO

O. Núñez, F. Fumero, J. Sigut, S. Alayón
Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores, Facultad de Física. Universidad de La Laguna (ULL). Avda. Astrofísico Fco. Sánchez s/n, 38206, La Laguna (Tenerife), España.
omar@isaatc.ull.es

Resumen

La segmentación de las imágenes de fondo de ojo, también llamadas retinografías, resulta de gran importancia para el diagnóstico automático de enfermedades que afectan a la retina tales como el glaucoma. Existen gran cantidad de métodos de segmentación ad-hoc específicos para este tipo de imágenes y para las distintas regiones de interés a detectar. Al tratarse de técnicas ad-hoc, hay que diseñarlas específicamente para cada problema y son difícilmente reutilizables. Una alternativa a esta estrategia podría ser la utilización de métodos de segmentación de propósito general que han sido ya extensamente probados con bastante éxito sobre imágenes de escenas naturales. La aplicación de estos métodos en el ámbito médico es todavía escasa. El objetivo principal de este trabajo es demostrar, a través de una evaluación exhaustiva, que es posible obtener buenos resultados aplicando también estas técnicas a retinografías, habitualmente más ruidosas y con menos contraste que las imágenes naturales. Para ello se han considerado algunos de los algoritmos de segmentación más citados y que están considerados como del estado del arte en numerosas aplicaciones.

Palabras Clave: segmentación, retinografía.

1 INTRODUCCIÓN

Los métodos de segmentación de imágenes agrupan píxeles en regiones visualmente significativas que podemos denominar 'superpixels', las cuales pueden ser utilizadas como elementos básicos para resolver tareas de visión por computador de más alto nivel como pueden ser reconocimiento o rastreo de objetos, reconstrucciones de imágenes, etc. Aunque los métodos de segmentación de propósito general han sido ampliamente estudiados durante muchos años, sigue siendo un problema abierto debido a su dificultad. A pesar de ello, existen algunos algoritmos de segmentación cuyos buenos resultados y la disponibilidad de implementaciones más o menos eficientes, los han convertido en una opción adecuada en la práctica para abordar diferentes problemas de visión. Entre estos métodos podemos encontrar el *Normalized Cuts* de Cour et al [13], el conocido algoritmo *Mean Shift* de Comaniciu y Meer [10], el *Efficient Graph* de Felzenszwalb y Huttenlocher [15], el Simple Linear Iterative Clustering (SLIC) de Achanta et al [2], y el gPb-owt-ucm de Arbelaez et al [6]. Los cuatro primeros son métodos orientados al agrupamiento de píxeles en regiones y el último está más bien orientado a la detección de contornos. Se explicarán con más detalle en una sección posterior. Estos trabajos se encuentran entre los más citados por sus numerosas aplicaciones sobre imágenes de escenas naturales [6] [2] [26]. Sin embargo la aplicación de estas técnicas en el ámbito de las imágenes médicas es muy escasa. La mayoría de los métodos en este campo siguen siendo soluciones muy ad-hoc, adaptadas a cada problema en particular y difícilmente aprovechables para otros análisis. Una de las razones que podría justificar la poca utilización de estos métodos es la falta de evidencias que apoyen su aplicación en imágenes médicas, habitualmente ruidosas y con poco contraste. Esto nos lleva a plantearnos la siguiente cuestión: ¿cabe esperar con este tipo de imágenes resultados tan buenos como los encontrados con imágenes naturales?

El objetivo de este trabajo es tratar de proporcionar una respuesta, al menos parcial, a esta pregunta, realizando una evaluación exhaustiva de los métodos de segmentación del estado del arte mencionados con anterioridad sobre imágenes de fondo de ojo, también llamadas retinografías, y en particular, sobre la zona de la cabeza del nervio óptico (papila) y las estructuras asociadas. Este tipo de imágenes son muy útiles para el diagnóstico del glaucoma [19].

La Figura 1 es un ejemplo de retinografía utilizada en este estudio. Se trata de una imagen de color en el visible que se captura con una cámara especializada. Se puede observar que los contornos de las estructuras presentes en la misma aparecen bastante difuminados y con un cierto ruido. En la Figura 2 se muestra la misma imagen con las regiones de interés para el diagnóstico médico indicadas y que por lo tanto serán aquellas que deben ser segmentadas correctamente.



Figura 1- Ejemplo de retinografía.

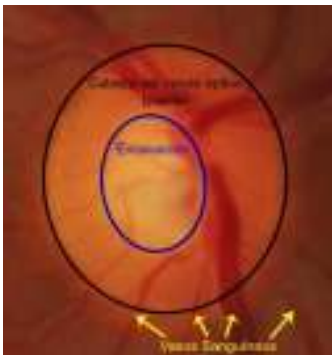


Figura 2 - Retinografía con indicaciones de las regiones de interés médico.

El resto de este artículo se organiza en los siguientes apartados: en la sección 2 se comienza con una revisión de los estudios ya realizados sobre la segmentación de las estructuras presentes en retinografías. Posteriormente, en la sección 3, se describen los métodos de segmentación seleccionados en este estudio, así como las métricas para evaluación de los resultados. En la sección 4 se presentan y discuten los resultados obtenidos, y finalmente se detallan las conclusiones en la sección 5.

2 TRABAJO RELACIONADO

La segmentación de la cabeza del nervio óptico es un problema que ha sido ampliamente estudiado y, sin embargo, continúa siendo una tarea compleja debido a la oclusión de los vasos sanguíneos, presencia de atrofias peripapilares, condiciones variables de imágenes, etc. La mayoría de los métodos existentes pueden ser clasificados en dos categorías: basados en detección de contornos o en clasificación de píxeles. Los métodos basados en contornos se caracterizan por su habilidad de extraer características que permiten diferenciar la papila del fondo y de otras estructuras. En [32] [5], la transformada circular y elíptica de Hough es utilizada para modelar el contorno del disco óptico por su eficiencia computacional. Los métodos basados en la clasificación de píxeles usan la intensidad, color o textura para caracterizar cada pixel y sus alrededores con el fin de detectar correctamente el disco [1]. Muramatsu et al [24] compararon métodos basados en contornos y en clasificación, y concluyeron que los resultados obtenidos eran similares.

A diferencia de lo que ocurre con la segmentación del disco óptico, la segmentación de la excavación en retinografías sin información de profundidad es un problema muy complicado debido a la falta de información visual fiable al respecto. A menudo, se usa como referencia el *pallor*, definido como el área de máximo contraste de color dentro del disco [30]. Otra referencia válida la ofrece la curvatura de los vasos [4] [30]. Debido a esta enorme dificultad, se han propuesto muy pocos métodos de segmentación de esta región.

De las tres regiones consideradas, la que está mejor estudiada y para la que se han conseguido mayores avances en su detección y segmentación son los vasos, apoyado además por el hecho de disponer de numerosas bases de datos con su correspondiente *ground truth* [28] [21]. Las técnicas más utilizadas son, en este caso, las de clasificación de píxeles y la identificación de estructuras con elongaciones y ramificaciones [25] [29].

La gran mayoría de métodos a los que nos hemos referido para la segmentación de las regiones de interés, operan a nivel de pixel y hacen uso de heurísticas ad-hoc para mejorar sus resultados. Como ya se comentó en la introducción, son muy pocos los trabajos en los que se pasa a un nivel de superpixels con técnicas totalmente generales de segmentación. Entre estos trabajos podemos citar el artículo de Xu et al [31] en el que se usa la técnica de Ncuts para generar superpixels con el fin de detectar daños por glaucoma, y la muy reciente aportación de Cheng et al [9] en la que se usa el método SLIC para obtener superpixels que luego son clasificados y combinados con modelos deformables.

3 MÉTODOS DE SEGMENTACIÓN Y MÉTRICAS UTILIZADAS

3.1 MÉTODOS DE SEGMENTACIÓN

3.1.1 Mean Shift

Mean Shift es una técnica no paramétrica para el análisis de una muestra en un espacio de características, presentada originalmente en 1975 por Fukunaga y Hosteler [18]. Es un método de segmentación en el que se trata de estimar de forma no paramétrica las modas de la distribución de puntos dada, y posteriormente se asocian todos los datos a los clusters definidos por dichos puntos. En una imagen, cada pixel x_i, $i=1,\ldots,n$ es tratado como un dato de entrada, y la densidad en el punto x es calculada por la función:

$$\hat{f}(x) = \frac{c}{nh^d}\sum_{i=1}^{n} K\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \qquad (1)$$

donde *h* es el parámetro de ancho de banda (*bandwith)*, *d* representa la dimensionalidad de los datos, *c* es una constante de normalización, y *K(.)* el kernel de estimación de densidad. En la implementación escogida del método Mean Shift, EDISON [11], se utiliza un kernel uniforme. Para localizar un máximo local de la densidad, se selecciona un punto inicial y_1 y es actualizado hasta su convergencia por la función:

$$y_{j+1} = \frac{\sum_{i=1}^{n} x_i K\left(\left\|\frac{y_j - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} K\left(\left\|\frac{y_j - x_i}{h}\right\|^2\right)} \qquad (2)$$

Una vez calculados los puntos locales de máxima densidad, la imagen puede ser segmentada en regiones simplemente asociando cada pixel a su punto de máxima densidad. En la implementación utilizada hay 3 parámetros ajustables: *spatial bandwith Hs, color bandwith Hr*, y el tamaño mínimo de región, *S*.

3.1.2 Normalized-Cut (N-Cut)

En este método una imagen es modelada como un grafo $G = (V, E)$ donde *V* es un conjunto de vértices correspondientes a los píxeles de la imagen, y *E* es un conjunto de aristas que conectan los píxeles vecinos. El peso de una arista $w(u, v)$ se describe como la afinidad entre los vértices *u* y *v*, basado en su similitud de intensidad y proximidad espacial. Usando este modelado de grafos, segmentar una imagen en dos regiones corresponde a un corte $cut\ (A, B)$ donde *A* y *B* son los vértices en los dos grafos resultantes. En N-Cut, el coste de segmentación se define como:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \qquad (3)$$

donde $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$ es el coste del corte de (A, B) y $assoc(A, V) = \sum_{u \in A, v \in V} w(u, v)$ es la asociación entre *A* y *V*. N-Cut segmenta la imagen buscando el $cut(A, B)$ con el mínimo coste de la ecuación (3). El parámetro principal de la implementación escogida [12] es el número de regiones a ser segmentadas, *K*.

3.1.3 Efficient Graph

Efficient Graph es un método de segmentación basado en la utilización de modelos de grafos, que trata de buscar evidencias de bordes entre dos segmentos utilizando para ello las diferencias de intensidad a través del borde, y las diferencias de intensidad en cada segmento. La diferencia en cada segmento es definida como el peso de la arista más larga del *spanning tree* mínimo construido desde este segmento. Por su parte, la diferencia de intensidad a través del borde, se define como el peso mínimo de arista que conecta estos dos segmentos. Efficient Graph tiene orden de complejidad *O(n log n)* para segmentar una imagen de n-píxeles. La implementación utilizada [16] dispone de 3 parámetros ajustables: un factor de filtrado σ, *smoothing*, relacionado con la escala del filtrado Gaussiano que se aplica de forma previa a la segmentación; *K* que controla el refinamiento de la segmentación y, por tanto, el tamaño de los segmentos; y finalmente un parámetro *S* de tamaño mínimo de región.

3.1.3 gPb-owt-ucm

La técnica gPb-owt-ucm presentada en [6] puede dividirse conceptualmente en dos etapas: la detección de contornos (gPb) y la segmentación (owt-ucm).
La detección de contornos gPb, variante de la función Pb(x, y, Θ) presentada en [22], utiliza características locales como brillo, color y textura, que son calculadas haciendo uso de operadores de gradiente orientado en cada punto de la imagen. Con estas características, se construye una matriz de afinidad que representa la similitud entre píxeles. A partir de esta matriz, se obtiene un problema de autovalores, que se resuelve para un número fijo de autovectores, los cuales contienen la información de los contornos.
Posteriormente, y puesto que la detección de contornos puede generar regiones no cerradas, se pasa a la etapa de segmentación haciendo uso del detector de contornos anterior, y de un algoritmo de clustering en dos pasos. El primero de ellos es una variante de la Transformada Watershed, denominada Oriented Watershed Transform (owt), que construye un conjunto inicial de regiones a partir de los datos de la etapa de detección de contornos. En este paso obtenemos una representación en la cual el valor asignado a un arco resultante de owt, puede ser visto como la probabilidad de que dicho arco sea un borde real. Finalmente, se utiliza un procedimiento de clustering para transformar las regiones en un modelo jerárquico que puede ser representado por un Ultrametric Contour Map (ucm). En este modelo, los niveles más altos de dicha jerarquía sólo mantienen los contornos con la probabilidad más alta, mientras que el nivel base mantiene incluso los bordes con menor probabilidad, lo cual proporciona una imagen sobresegmentada. Por tanto, este método dispone de la flexibilidad de poder obtener segmentaciones más refinadas seleccionando el nivel adecuado en la jerarquía.

La implementación utilizada [7] dispone de un único parámetro *k*, el cual representa la escala de la jerarquía a la que se extrae la segmentación. Un mayor valor dará un resultado limitado solamente a las regiones bien contrastadas.

3.1.3 Simple Linear Iterative Clustering (SLIC)

SLIC es un método de segmentación basado en la creación de clusters de píxeles según su similitud en color y en su proximidad en la imagen. Las regiones resultantes suelen tener un tamaño bastante similar, por lo que el principal parámetro de este método es el número de segmentos deseados en la imagen final, representado por *K*. Dado dicho parámetro, el algoritmo comienza eligiendo *K* centros de clústeres regularmente espaciados por $S = \sqrt{(N/k)}$ píxeles, donde *N* es el número total de píxeles de la imagen. Primero, los centros son desplazados hacia la posición de gradiente mínimo en un vecindario definido de 3 x 3. Cada píxel de la imagen es asociado al centro del clúster más cercano, cuya área de búsqueda alcanza a este píxel. Posteriormente, para cada centro C_k, SLIC busca, en un vecindario de 2*S* x 2*S*, el píxel de mayor coincidencia de acuerdo a las características de color y de proximidad espacial, y se calcula el nuevo centro del clúster basado en el píxel encontrado. La iteración continua hasta que la distancia entre los nuevos centros y los anteriores es lo suficientemente pequeña.
En la implementación elegida [3] existen 2 parámetros para ajustar: *K*, que representa el número de regiones deseadas en la imagen resultado; y un parámetro *m* que controla la uniformidad de las mismas.

3.2 SELECCIÓN DE PARÁMETROS

Para cada método de segmentación, hemos elegido de forma experimental el rango de valores de cada parámetro, con el objetivo de abarcar un espacio de combinaciones que nos asegure encontrar la mejor segmentación posible. A continuación se detalla el rango de valores que se ha fijado para cada método.

3.2.1 Mean Shift

La aplicación EDISON [11], dispone de 3 parámetros ajustables: *spatial bandwith Hs, color bandwith Hr*, y el tamaño mínimo de región, *S*.
Los valores elegidos se muestran en la Tabla 1.

Tabla 1 - Valores elegidos para parámetros de EDISON

Parámetro	Mínimo	Máximo	Intervalo
Hs	6	40	2
Hr	1	12	1
S	50	1000	50

3.2.2 Normalized-Cut

El método N-Cut dispone de un único parámetro *K*, el cual indica el número de segmentos deseados. El valor máximo aceptado es el número total de píxeles de la imagen.
El rango de valores seleccionado se muestra en la Tabla 2.

Tabla 2 - Valores elegidos para parámetros de Normalized-Cut

Parámetro	Mínimo	Máximo	Intervalo
K	2	150	1

3.2.3 Efficient Graph

La implementación utilizada para el método Efficient Graph [16], dispone de los siguientes 3 parámetros: σ utilizado por el filtrado previo a la segmentación; *K* que controla el refinamiento de la segmentación; y *S* el cual establece un tamaño mínimo de región.
Los valores seleccionados se muestran en la Tabla 3.

Tabla 3 - Valores elegidos para parámetros de Efficient Graph

Parámetro	Mínimo	Máximo	Intervalo
σ	0	1	0.1
K	1	51	5
S	1	2801	200

3.2.4 gPb-owt-ucm

Debido a que el algoritmo gPb-owt-ucm proporciona una jerarquía de segmentaciones, sólo dispone de un único parámetro *k*, el cual establece el umbral de probabilidad de la existencia de un contorno.
Se pudo comprobar experimentalmente, que un valor superior a 0.4 proporcionaba una segmentación con muy pocos segmentos en todas las imágenes del conjunto, lo cual refleja el poco contraste existente entre las regiones de las imágenes de fondo de ojo, si lo comparamos con las imágenes naturales. En la Tabla 4 se muestra el rango de valores seleccionado.

Tabla 4 - Valores elegidos para parámetros de gPb-owt-ucm

Parámetro	Mínimo	Máximo	Intervalo
k	0	0.4	0.001

3.2.5 Simple Linear Iterative Clustering

La implementación seleccionada [3] para el método SLIC, contiene dos parámetros ajustables: *K*, el cual representa el número de segmentos deseados en la imagen resultado; y *m*, que controla la uniformidad de dichos segmentos.

El rango de valores seleccionados se muestra en la Tabla 5.

Tabla 5 - Valores elegidos para parámetros de SLIC

Parámetro	Mínimo	Máximo	Intervalo
K	8	200	3
m	1	31	1

3.3 MÉTRICAS

Es ampliamente conocido que la adecuada evaluación de un método de segmentación sigue siendo objeto de debate [23]. Se ha realizado un gran esfuerzo en desarrollar métricas que midan directamente la calidad de una segmentación. Para este trabajo, se han elegido algunas de las métricas más utilizadas en segmentación supervisada. De esta manera, la evaluación cuantitativa se llevará a cabo a través de los siguientes índices que serán explicados en las siguientes subsecciones: Boundary Displacement Error (BDE), Variation of Information (VI), Rand Index (RI), y Segmentation Covering (SC).

3.3.1 Boundary Displacement Error (BDE)

BDE [17] es una métrica basada en los bordes para evaluar la calidad de la segmentación. Define el error de un pixel del borde como la distancia entre dicho pixel y su pixel más cercano en el borde de la imagen de referencia. Un valor cercano a 0 indicará una buena calidad de segmentación.

3.3.2 Variation of Information (VI)

La métrica Variation of Information fue presentada en [23] con el objetivo de poder realizar comparaciones de clusters. Mide la distancia entre dos segmentaciones en términos de su entropía media condicional, dada por:

$$\mathrm{VI}(\mathrm{S},\mathrm{S}_0) = \mathrm{H}(\mathrm{S}) + \mathrm{H}(\mathrm{S}_0) - 2\mathrm{I}(\mathrm{S},\mathrm{S}_0) \qquad (4)$$

donde H e I representan entropías e información mutua entre dos clusters de datos *S* y S_0. En este caso, dichos clusters son la segmentación resultado y la de referencia. La segmentación será mejor cuanto más cercano a 0 sea el resultado de VI.

3.3.3 Rand Index (RI)

Originalmente, Rand Index [27] fue introducido para evaluación de clusters. Su funcionamiento se basa en comparar la compatibilidad de asignaciones entre pares de elementos en dichos clusters. El resultado de aplicar esta métrica entre dos segmentaciones *S*, S_0 es dado por la suma del número de pares de píxeles que tienen la misma etiqueta en *S* y S_0, y aquellos que tienen diferente etiqueta en ambas segmentaciones, dividido por el número total de pares de píxeles:

$$R(S,S') = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i \neq j}} \left[II\left(l_i = l_j \wedge l_i' = l_j'\right) + II\left(l_i \neq l_j \wedge l_i' \neq l_j'\right) \right] \qquad (5)$$

Un resultado cercano a 1 representa buena calidad de segmentación.

3.3.4 Segmentation Covering (SC)

El overlap entre dos regiones *R* y *R'*, definido como:

$$\mathrm{O}(\mathrm{R},\mathrm{R}') = \frac{|\mathrm{R} \cap \mathrm{R}'|}{|\mathrm{R} \cup \mathrm{R}'|} \qquad (6)$$

ha sido usado para la evaluación de tareas de clasificación a nivel de píxel [14] [20]. En [8] se define el covering de una segmentación *S* por una segmentación *S'*, como:

$$C(S' \rightarrow S) = \frac{1}{N} \sum_{R \in S} |R| \cdot \max_{R' \in S'} O(R,R') \qquad (7)$$

donde *N* representa el número total de píxeles en la imagen. Al igual que en RI, un valor cercano a 1 indicará buena calidad de segmentación.

3.3.5 Elección de métricas

De lo anterior, podemos observar que no es posible definir un único criterio óptimo para evaluar la calidad de una segmentación en todos los casos. Por ejemplo, RI se basa en examinar la relación entre pares de píxeles, por lo que es más apropiado para algoritmos de segmentación basados en pares, como aquellos que utilizan particiones de grafos. Por otro lado, para métodos de clustering como Mean Shift, basados en la relación de un punto y el centro de su clúster, VI será una mejor opción. Por tanto, en este estudio se mostrarán los datos de todos los índices anteriormente detallados, para todos los métodos de segmentación utilizados.

Debido a que cada método dispone de una serie de parámetros variables, el resultado dependerá de los valores elegidos para los mismos. Se han considerado tres categorías: *optimal dataset scale (ODS), optimal image scale OIS* y *Best*, siguiendo el esquema planteado en [6]. ODS se basa en evaluar las segmentaciones utilizando valores fijos de parámetros para todas las imágenes del conjunto, por lo que obtendremos los valores de las métricas para la mejor combinación de valores de parámetros en todo el grupo de imágenes. OIS por su parte, se

estima seleccionando la mejor combinación para cada imagen, obteniendo lógicamente mejores resultados que para ODS. Best indica el mejor valor obtenido en todo el conjunto y para todos los valores de parámetros posibles.

4 RESULTADOS EXPERIMENTALES Y DISCUSIÓN

El presente estudio fue realizado sobre 104 retinografías de diferentes pacientes sanos, proporcionadas por el Hospital Universitario de Canarias. Las imágenes tienen un tamaño aproximado de 350 x 400 píxeles. Para cada imagen se dispone de una segmentación de referencia o *ground truth* en la que aparecen etiquetadas todas las regiones de interés presentes: vasos sanguíneos, papila, excavación, y resto (fondo). En la Figura 4 se muestra un ejemplo de estas imágenes de referencia.
En la Tabla 6 se resumen los resultados encontrados para las métricas calculadas sobre 104 imágenes.
Como se puede observar, los mejores resultados se obtienen con los métodos Mean Shift y gPb-owt-ucm, lo cual coincide con lo publicado en [6] para imágenes naturales. Destaca el método Mean Shift sobre el resto, obteniendo mejores valores de acuerdo a las métricas Rand Index, Variation Of Information y Segmentation Covering. Sin embargo, para BDE se obtiene un comportamiento muy similar entre Mean Shift y gPb-owt-ucm, llegando este último a mejorar los valores de ODS y Best. Hay que tener en cuenta, no obstante que, a diferencia de lo que ocurre en el estudio realizado en [6], el método no ha sido entrenado con imágenes de retinografías sino que se ha utilizado el entrenamiento por defecto con imágenes naturales. Además, la diferencia entre los valores OIS de ambos métodos es considerablemente baja frente a los demás, por lo que ambos algoritmos pueden considerarse similares de acuerdo a BDE.
Por otro lado, en lo que se refiere a los valores obtenidos para RI, el método Mean Shift consigue superar al resto con unas diferencias notables. Cabe destacar también los valores prácticamente similares de ODS y OIS de los métodos gPb-owt-ucm, N-Cuts y Efficient Graph para esta métrica.
Por último, si miramos los valores obtenidos para la métrica Segmentation Covering, el método Mean Shift consigue superar al resto con una amplia diferencia. Sin embargo, es importante destacar que en algunos casos se puede producir la pérdida de regiones de interés pequeñas por asociación a regiones de mayor tamaño (*undersegmentation*), sin que se vea penalizado de manera notable en los valores obtenidos. Esto es debido a que en la definición de SC, según la ecuación (7), el máximo overlap obtenido para cada segmento es multiplicado por el tamaño del mismo, lo que ocasiona que las regiones de pequeño tamaño no tengan una gran influencia en el resultado. En el caso de las imágenes de fondo de ojo, y especialmente con pacientes sanos, la excavación puede presentar un tamaño muy inferior al resto de elementos presentes en la imagen, y con un contraste que varía muy ligeramente respecto a la papila. Por tanto, puede ocurrir que se asocie dicha zona a la región papilar, lo cual es altamente indeseable desde el punto de vista médico.
Para poder realizar una comparativa visual de los resultados de los métodos, se ha elegido una retinografía del conjunto de imágenes, mostrada en la Figura 3. La delimitación manual de la misma se muestra en la Figura 4, y la mejor segmentación para cada método y métrica, se muestran de forma conjunta en la Figura 5. Como puede observarse, los resultados del Mean Shift y gPb-owt-ucm son muy similares en cuanto al tamaño de las regiones obtenidas y los contornos de las mismas. En el método Efficient Graph se puede apreciar cierta irregularidad en las regiones, siendo especialmente notable en los bordes de la papila. Un comportamiento similar ocurre en los resultados de SLIC donde, incluso para las métricas RI y VI, las mejores segmentaciones presentan un contorno papilar incorrecto.

5 CONCLUSIONES

Se ha llevado a cabo una evaluación exhaustiva de la calidad de las segmentaciones de imágenes de fondo de ojo utilizando para ello algunos de los métodos de segmentación generales más conocidos y que mejores resultados han dado con imágenes de escenas naturales. A pesar del ruido presente y del bajo contraste entre algunas zonas de la imagen, tanto desde un punto de vista cuantitativo, reflejado en las métricas usadas, como desde un punto de vista más cualitativo, podemos concluir que los resultados encontrados para las retinografías son prometedores y se aproximan bastante a lo publicado en aplicaciones de estos métodos con otro tipo de imágenes [6]. Si bien, es complicado obtener las regiones de interés perfectamente segmentadas, sí que ha sido posible generar superpixels de gran tamaño que se aproximan al objetivo buscado y que, en todo caso, pueden facilitar enormemente cualquier tipo de postprocesamiento adicional. En definitiva, todo parece indicar que el aprovechamiento de estas técnicas generales de segmentación es una alternativa a considerar, al menos para este ámbito de la medicina.

Agradecimientos

Este proyecto de investigación ha sido parcialmente financiado por el proyecto PI12/02301 del Fondo de Investigación Sanitaria Instituto Carlos III.

Tabla 6 - Resumen de datos obtenidos para todas las métricas y métodos.

	BDE			RI			VI			SC		
	ODS	**OIS**	**Best**	**ODS**	**OIS**	**Best**	**ODS**	**OIS**	**Best**	**ODS**	**OIS**	**Best**
MS	4,233	**3,0805**	1,7128	**0,7069**	**0,7897**	**0,9226**	**1,9946**	**1,7142**	**1,083**	**0,4686**	**0,5915**	**0,7831**
NCuts	5,1029	4,4275	2,608	0,6647	0,6888	0,8577	2,4076	2,3236	1,496	0,371	0,4385	0,7046
EG	4,8329	3,9387	2,6873	0,664	0,6815	0,7781	3,1263	2,918	1,7637	0,3315	0,3823	0,5623
gPb-owt-ucm	**3,7464**	3,3076	**1,6672**	0,6686	0,6888	0,8155	2,1022	2,0167	1,2499	0,3899	0,4505	0,6964
SLIC	7,9147	7,0097	4,252	0,6579	0,6627	0,7652	3,062	2,92	2,429	0,2927	0,3103	0,4214



Figura 3 - Retinografía seleccionada.



Figura 4 – Imagen de referencia para retinografía elegida.

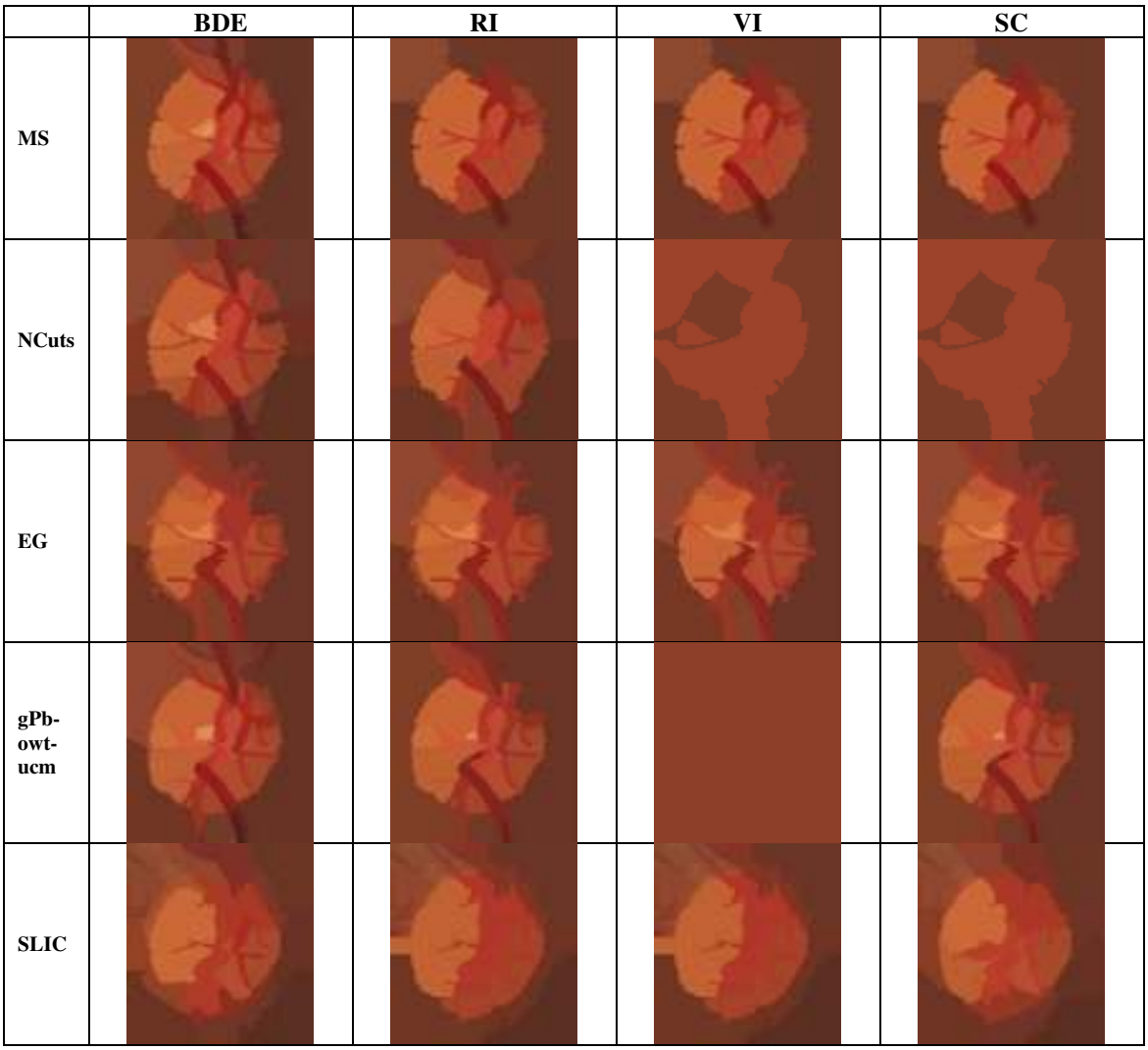


Figura 5 - Resultados de mejor segmentación para la retinografía elegida. **De arriba hacia abajo:** Métodos de segmentación utilizados. **De izquierda a derecha:** Mejor segmentación del método, de acuerdo a la métrica indicada (OIS).

Referencias

[1] M. D. Abràmoff et al., "Automated segmentation of the optic disc from stereo color photographs using physiologically plausible features.," *Investigative Ophthalmology Vis Sci.*, 2007.

[2] R. Achanta et al., "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. and Mach. Intell*, 2012.

[3] R. Achanta et al. SLIC Superpixels source code. [Online]. http://ivrg.epfl.ch/research/superpixels

[4] R. R. Allingham, K. F. Damji, S. Freedman, S. E. Moroi, and G. Shafranov, *Shields' Textbook of Glaucoma*., 2005.

[5] A. Aquino, M. Gegundez-Arias, and D. Marin, "Detecting the optic disc boundary in digital fundus images using morphological, edge detection and feature extraction techniques.," *IEEE Trans. Med. Imag.*, 2010.

[6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," 2011.

[7] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Image Segmentation Resources. [Online]. http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html

[8] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From Contours to Regions : An Empirical Evaluation.," *2009 IEEE Conference on Computer Vision and Pattern Recognition.*, 2009.

[9] J. Cheng et al., "Superpixel Classification Based Optic Disc and Optic Cup Segmentation for Glaucoma Screening.," *IEEE Transactions on Medical Imaging*, 2013.

[10] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *PAMI*, 2002.

[11] D. Comaniciu and P. Meer. Mean-shift image segmenation and edge detection source code. [Online]. http://www.caip.rutgers.edu/riul/research/code/EDISON/index.html

[12] T. Cour, F. Benezit, and J. Shi. Multiscale Normalized Cuts Segmentation Toolbox for MATLAB. [Online]. http://www.timotheecour.com/software/ncut_multiscale/ncut_multiscale.html

[13] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," *CVPR*, 2005.

[14] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman, "PASCAL 2008 Results," *http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html*, 2008.

[15] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image," *IJCV*, 2004.

[16] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based segmentation source code. [Online]. http://cs.brown.edu/~pff/segment/

[17] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration.," 2002.

[18] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition.," *IEEE Transactions on Information Theory*, 1975.

[19] M. Gonzalez de la Rosa et al., "Measuring hemoglobin levels in the optic nerve head: comparisons with other structural and functional parameters of glaucoma.," *Invest Ophthalmol Vis Sci.*, 2013.

[20] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," *ICCV*, 2005.

[21] T. Kauppi et al. DIARETDB1 - Standard Diabetic Retinopathy Database. [Online]. http://www2.it.lut.fi/project/imageret/diaretdb1/

[22] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color and texture cues.," *PAMI*, 2004.

[23] M. Meila, "Comparing clusterings: an axiomatic view.," *ICML*, 2005.

[24] C. Muramatsu et al., "Automated segmentation of optic disc region on retinal fundus photographs: Comparison of contour modeling and pixel classification methods.," *Comput. Methods Programs Biomed.*, 2011.

[25] M. Niemeijer, J. S. Staal, B. van Ginneken, M. Loog, and M. D. Abramoff, "Comparative study of retinal vessel segmentation on a new publicly available database.," *Medical Imaging: Image Analysis.*, 2004.

[26] B. Peng, L. Zhang, and D. Zhang, "A survey of graph theoretical approaches to image segmentation," *Pattern Recognition*, 2013.

[27] W. M. Rand, "Objective criteria for the evaluation of clustering methods.," *Journal of the American Statistical Association*, 1971.

[28] J.J. Staal, M.D. Abramoff, M. Niemeijer, B. van Ginneken, and M. A. Viergever. DRIVE: Digital Retinal Images for Vessel Extraction. [Online]. http://www.isi.uu.nl/Research/Databases/DRIVE/

[29] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and G. B. van, "Ridge-based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imaging*, 2004.

[30] J. Xu, O. Chutatape, E. Sung, C. Zheng, and P. C. T. Kuan, "Optic disk feature extraction via modified deformable model technique for glaucoma analysis.," *Pattern Recognit.*, 2007.

[31] J. Xu, H. Ishikawa, G. Wollstein, and J. S. Schuman, "3D Optical Coherence Tomography Super Pixel with Machine Classifier Analysis for Glaucoma Detection.," *Conf. Proc. IEEE Eng Med Biol Soc.*, 2011.

[32] X. Zhu and R. M. Rangayyan, "Detection of the optic disc in images of the retina using the hough transform.," *Int. Conf. of IEEE Eng in Med. and Bio. Soc.*, 2008.

REDES NEURONALES ARTIFICIALES EN UN REGULADOR PID PARA CONTROL AUTOMÁTICO DEL RUMBO DE UN BUQUE

José Luis Casteleiro-Roca, José Luis Calvo-Rolle
Dpto. de Ingeniería Industrial, Universidad de La Coruña
jose.luis.casteleiro@udc.es, jlcalvo@udc.es

Matilde Santos
Dpto. Arquitectura de Computadores y Automática, Universidad Complutense de Madrid
msantos@ucm.es

Resumen

El direccionamiento de un buque es uno de los problemas presentes en la navegación marítima para el que se han propuesto diferentes métodos de control. No es una tarea sencilla pues un barco es un sistema complejo, no lineal, de orden elevado, difícil de controlar con reguladores clásicos. En este trabajo se presenta como alternativa una estrategia de control de Ganancia Programada (Gain Scheduling) de reguladores PID combinada con una de las técnicas de la Inteligencia Artificial, las redes neuronales artificiales, lo que permite solventar algunos de los problemas propios de su implementación y obtener buenos resultados en simulación.

Palabras Clave: *Control automático de rumbo, Ganancia programada, Redes neuronales, Regulador PID, Barco.*

1 INTRODUCCIÓN

En el ámbito de la ingeniería de control se buscan nuevas alternativas a los métodos de regulación ya establecidos, para paliar algunas de las deficiencias en los ya existentes o para encontrar soluciones mejores, como por ejemplo las proporcionadas por el control inteligente [18].

Pese a estas nuevas estrategias avanzadas, hasta el momento el control basado en el regulador PID sigue siendo la opción más establecida a nivel de aplicaciones industriales, quizás por diversas razones como robustez, fiabilidad, relativa simplicidad, tolerancia a fallos, etc. También en el ámbito del control automático del rumbo de buques tienen una larga trayectoria, desde que fue aplicado por Nicolas Minorsky [8] en el año 1922 hasta la actualidad.

Además los controladores PID se han visto mejorados con funcionalidades avanzadas, en algunos casos gracias a la sinergia con técnicas de la Inteligencia Artificial. Por ejemplo, existen múltiples aportaciones para la optimización de la sintonía de los parámetros del PID [6, 12]. No obstante, en la mayoría de los casos esos valores se obtienen para determinadas condiciones de funcionamiento de la planta y no son generales.

Por otro lado, la inmensa mayoría de los sistemas reales no son lineales, hecho notorio en el direccionamiento de buques, motivo por el cual surgen trabajos como el de Nomoto [16], el de Norbbin [17], o el de Bech [7] para tratar a estos sistemas mediante aproximaciones lineales de orden reducido en determinados puntos de funcionamiento.

En función del punto de trabajo en el que se encuentre el sistema a controlar, para unas especificaciones determinadas que pueden ser iguales para todas las zonas de funcionamiento, serán necesarios valores diferentes de los parámetros del regulador. En este aspecto los reguladores PID autoajustables y adaptativos pueden resultar una buena solución [3, 4]. Pero su implementación suele ser difícil, cara, y está muy vinculada al tipo de proceso que se pretende regular.

El conocido método de Ganancia Programada (Gain Scheduling) es sin embargo sencillo de implementar, puede salvar alguna de las dificultades comentadas anteriormente y permite obtener resultados bastante satisfactorios. El concepto de *Gain Scheduling* surge en los comienzos de la década de los 60 [13], y es considerado como parte de la familia de los controladores adaptativos [3]. El principio de esta metodología consiste en dividir un sistema no lineal en varias regiones en las cuales el comportamiento se pueda aproximar por uno que sí lo sea, pudiendo de este modo tener en cada zona unos parámetros del controlador que permitan alcanzar unas especificaciones similares en todo el rango de funcionamiento de la planta.

Para aplicar el *Gain Scheduling* es necesario en primer lugar elegir las variables significativas del

sistema en función de las cuales se va a definir el punto de trabajo. Se suele optar por aquellas que se puedan medir fácilmente. Seguidamente se han de escoger puntos de funcionamiento a lo largo de todo el rango de operación de la planta. Esta operación hay que realizarla cuidadosamente ya que el sistema puede tener un comportamiento no estable entre los puntos elegidos [11]. No existe un procedimiento sistemático para estas tareas.

En este trabajo se aplican redes neuronales artificiales, una de las técnicas del Soft Computing, para solucionar el problema de la interpolación entre los puntos de operación. Existen casos similares al que se plantea en este trabajo resueltos de esta manera para otras aplicaciones [9, 15], así como con otras técnicas de la inteligencia artificial [1, 10, 19].

El presente documento se estructura comenzando con una breve introducción al control mediante la estrategia de ganancia programa para controladores PID, para a continuación describir su aplicación al modelo de un buque. Se han aplicado redes neuronales para la interpolación de los parámetros de sintonía entre las zonas de funcionamiento. Se realiza el control de direccionamiento con la metodología propuesta y se valida realizando simulaciones en diferentes condiciones.

2 SINTONÍA DEL PID CON GANANCIA PROGRAMADA UTILIZANDO REDES NEURONALES

Existen múltiples formas de representación del regulador PID [6], pero quizás la más difundida es la que viene dada por la ecuación (1).

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right] \quad (1)$$

Donde u es la variable de control y e es el error dado por la diferencia entre la referencia especificada y la salida medida del proceso, $e = y_{SP} - y$.

La variable de control es la suma de tres términos: el término P, proporcional al error; el término I, proporcional a la integral del error; y el término D, proporcional a la derivada del error. Los parámetros de sintonía son la ganancia proporcional K, el tiempo integral T_i y el tiempo derivativo T_d.

El ajuste de este controlador se lleva a cabo variando esos parámetros para que el regulador consiga que el sistema controlado alcance unas determinadas especificaciones de su respuesta.

El método de la Ganancia Programada puede considerarse como una realimentación no lineal de un tipo especial; posee un regulador lineal cuyos parámetros son modificados en función de las condiciones de operación mediante una serie de reglas (figura 1).

En múltiples ocasiones se sabe cómo varía la dinámica de un proceso según el punto de trabajo del mismo. Entonces es posible modificar los parámetros del controlador, monitorizando sus condiciones de operación y estableciendo reglas que los modifiquen de forma que se obtengan unas determinadas especificaciones a lo largo de todo el rango de funcionamiento del proceso.

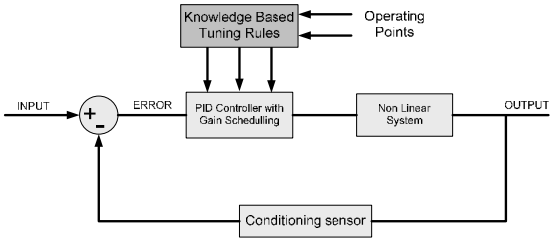


Figura 1: Estrategia de control de ganancia programada

En este trabajo las reglas que definen las ganancias del controlador PID en base al punto de trabajo del sistema se han sustituido por una red neuronal. Entre los puntos de operación para los que se ha ajustado el controlador mediante técnicas de control tradicionales, será la red neuronal la que proporcione los parámetros de ajuste. El objetivo fundamental que se persigue de este modo es hacer una interpolación lo más gradual posible, que mantenga las especificaciones del sistema en unos valores óptimos para todo el rango de operación. Sus entradas serían las condiciones de operación de la planta, y como salidas tendría los parámetros del controlador PID (K, T_i y T_d). La red neuronal empleada es una red tipo MLP (Multi Layer Perceptron).

3 MODELO DEL BARCO UTILIZADO PARA EL DIRECCIONAMIENTO

Para la descripción del movimiento del barco se ha utilizado el modelo de Nomoto [16], con el sistema de coordinadas indicado en la figura 2.

Se ha representado con V la velocidad del barco, que se proyecta en las componentes u y v correspondientes a los ejes x e y, respectivamente. El barco posee una velocidad angular r y una orientación Ψ. El ángulo del timón necesario para mantener el rumbo es δ, medido en grados con respecto al eje longitudinal del barco. La dinámica del sistema viene descrita por el sistema de

ecuaciones (2), donde *l* es la longitud del barco y los términos ***a*** y ***b*** son parámetros propios de cada barco.

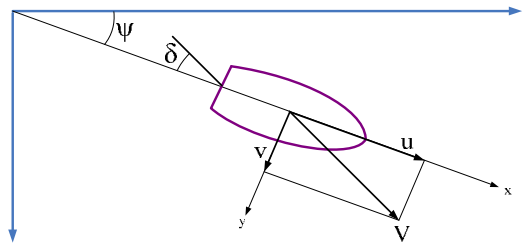


Figura 2: Direccionamiento de un buque [16]

$$\begin{cases} \dfrac{dv}{dt} = \dfrac{u}{l} a_{11} v + u \cdot a_{12} r + \dfrac{u^2}{l} b_1 \delta \\ \dfrac{dr}{dt} = \dfrac{u}{l^2} a_{21} v + \dfrac{u}{l} a_{22} r + \dfrac{u^2}{l^2} b_2 \delta \\ \dfrac{d\Psi}{dt} = r \end{cases} \quad (2)$$

La función de transferencia para el ángulo del timón viene dada en general por una expresión del tipo (3).

$$G(s) = \frac{K(1+sT_3)}{s(1+sT_1)(1+sT_2)} \quad (3)$$

En muchos casos el modelo puede quedar simplificado a una función de transferencia de segundo orden, con un integrador (4), denominada modelo de Nomoto de primer orden, cuya ganancia *b* puede ser expresada aproximadamente como (5).

$$G(s) = \frac{b}{s(s+a)} \quad (4)$$

$$b = c\left(\frac{u}{l}\right)^2 \left(\frac{A \cdot l}{D}\right) \quad (5)$$

donde *D* es el desplazamiento en m^3, *A* es el área del timón (m^2), y *c* es un parámetro cuyo valor empírico es aproximadamente 0.5 [16]. El parámetro *a* depende de la velocidad y la carga, y su signo puede cambiar con las condiciones de operación.

4 APLICACIÓN DE LA ESTRATEGIA DE CONTROL A UN CARGUERO

Para validar esta propuesta de control se ha aplicado al direccionamiento automático de un carguero de 161 m de eslora, cuyo desplazamiento va a oscilar desde los 8000 m^3 en vacío hasta los 20000 m^3 a plena carga. La velocidad a la que podrá navegar será mayor de 2 m/s, para la cual es perfectamente válido el modelo empleado, hasta los 8 m/s de velocidad máxima. El servo-timón opera a una velocidad de 4 m/s, limitado a ± 30º. El modelo del timón está representado en la figura 3.

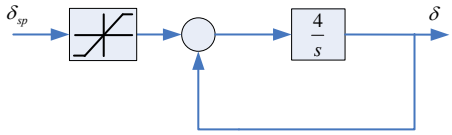


Figura 3: Modelo del servo-timón

La función de transferencia del carguero queda como se indica en la ecuación (6), donde los valores de *A* y *l* se sustituyen por los correspondientes al barco, y *D* y *u* dependen de sus condiciones de funcionamiento.

$$G(s) = \frac{0.5 \dfrac{u^2}{161} \dfrac{27.48}{D}}{s\left(s + a_0 \dfrac{u}{161}\right)} \quad (6)$$

Este modelo, junto con el control, se ha simulado en MatLab/Simulink (figura 4).

4.1 CONDICIONES DE OPERACIÓN DEL SISTEMA

Las condiciones de operación del sistema pueden ser muy variadas dentro del rango posible según el tipo de barco. Hay que seleccionar unos determinados puntos de operación ya que no tiene sentido obtener parámetros para un número excesivo de casos, lo que complicaría innecesariamente el control. Un criterio posible es escoger una cantidad razonable de datos equidistantes y observar la variación de los parámetros. Si esta variación es muy notable, se contemplará la posibilidad de coger nuevos puntos intermedios entre ellos.

En este sistema, las variables que van a definir las condiciones de operación y por lo tanto las reglas de ajuste de ganancias del controlador son el desplazamiento y la velocidad del buque. Como una aproximación realista, se han definido los puntos de trabajo que se muestran en la primera y segunda columnas de la tabla 1.

4.2 SINTONÍA DEL CONTROLADOR PARA CADA CONDICIÓN DE OPERACIÓN

Para la fase de obtención de los parámetros de ajuste de los diferentes puntos de trabajo se ha aplicado el método del Relé [2, 5]. Se ha trabajo con el sistema en lazo cerrado, obteniendo de forma empírica la ganancia crítica (*Kc*) y el periodo de oscilación

mantenida (*Tc*), aplicando posteriormente las fórmulas de Ziegler-Nichols con esos valores.

El método del Relay-Feedback tiene la ventaja de que el ajuste se puede realizar un determinado punto de consigna, pero tiene el inconveniente de que durante el proceso se podría llegar a hacer inestable el sistema.

Con las ganancias del PID obtenidas para cada zona de funcionamiento se dispone de unos valores iniciales que cumplen las especificaciones con una relación de caída de sobreoscilación de un cuarto ante las perturbaciones o cambios en la entrada.

Pero se requiere un ajuste fino del controlador para mejorar la respuesta del sistema, teniendo en cuenta la saturación de la señal de control que actúa sobre el timón. Se han mejorado los valores iniciales de las ganancias del controlador para evitar en lo medida de lo posible la saturación de los actuadores y, consecuentemente, evitar su deterioro.

Tabla 1: Parámetros de regulador obtenidos para cada punto de operación.

D (m^3)	u (m/s)	K	T_i	T_d
8000	2	3	350	90
	4	1	300	80
	6	1	350	80
	8	1	350	80
12000	2	4	350	80
	4	3	300	90
	6	1	350	95
	8	1	350	85
16000	2	6	400	90
	4	3	300	90
	6	1	350	90
	8	1	300	85
20000	2	6	450	90
	4	4	350	90
	6	1	350	90
	8	1	300	90

4.3 RED NEURONAL

La red neuronal MLP es una red de arquitectura sin realimentación ("hacia delante"). Es una de las topologías más típicas debido a su robustez y relativamente simple estructura. Sin embargo la arquitectura interna se debe seleccionar cuidadosamente para obtener unos buenos resultados. Se han realizado pruebas usando una capa oculta con entre 5 y 9 neuronas. Según [14] sólo es necesaria una capa intermedia para realizar correctamente la regresión, que es el uso que se le dará en este trabajo. La función de activación de las neuronas en la capa oculta ha sido seleccionada después de realizar diversas pruebas entre función de activación lineal, sigmoidal y tangente sigmoidal.

En este caso se ha decidido implementar una red neuronal por cada parámetro del controlador PID, en lugar de una sola red que tuviese las tres ganancias como salida. La entrada a las redes neuronales son los parámetros que identifican el punto de trabajo del sistema: el desplazamiento y la velocidad del barco.

En la tabla 2 se muestran los resultados del entrenamiento de la red neuronal para la ganancia proporcional del controlador. Para este entrenamiento se hace uso de los valores obtenidos en el ajuste del regulador para los distintos puntos de trabajo contemplados (tabla 1).

Tabla 2: Error de aproximación de la red neuronal para el parámetro K según la función de activación

Número de neuronas	Función de activación		
	Lineal	Sigmoidal	Tangente sigmoidal
5	$0.41 \cdot 10^{-1}$	$0.43 \cdot 10^{-2}$	$0.87 \cdot 10^{-3}$
6	$0.48 \cdot 10^{-1}$	$0.28 \cdot 10^{-2}$	$0.92 \cdot 10^{-3}$
7	$0.52 \cdot 10^{-1}$	$0.36 \cdot 10^{-2}$	$0.89 \cdot 10^{-3}$
8	$0.24 \cdot 10^{-1}$	$0.41 \cdot 10^{-2}$	$0.91 \cdot 10^{-3}$
9	$0.15 \cdot 10^{-1}$	$0.21 \cdot 10^{-2}$	$0.88 \cdot 10^{-3}$

Los mejores resultados se obtuvieron con 5 neuronas para la red neuronal que calcula K, y 6 para las que calculan T_i y T_d; la función de activación seleccionada para las tres ha sido del tipo tangente sigmoidal. La convergencia a un error del orden de 10^{-3} ha requerido 531, 705 y 686 épocas para K, T_i y T_d respectivamente. Es muy importante no caer en el sobre-entrenamiento de la red dado que lo que se pretende es una buena interpolación entre los puntos de operación definidos.

4.4 SIMULACIÓN DEL SISTEMA Y RESULTADOS

Se implementa en Simulink (figura 4) el sistema con el bloque de control denominado Neuro-PID.

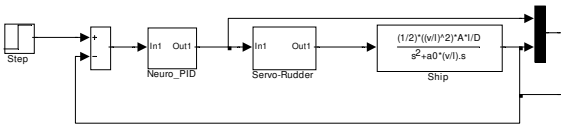


Figura 4: Sistema final implementado en Simulink

La salida del controlador es la entrada al servo timón.

De este modo el controlador implementado escogerá los parámetros más apropiados a la zona en la cual se encuentre trabajando. Es necesario destacar que se podrían obtener muchos más puntos para entrenar las

redes neuronales, pero sería más costoso en términos computacionales y, además, la propia red neuronal ya interpola adecuadamente entre ellos (dejando de manifiesto una de las ventajas de su empleo).

Para la validación del procedimiento se ha simulado el sistema para diferentes valores de velocidad y desplazamiento. En la figura 5 se muestran cuatro de las pruebas realizadas, como ejemplos representativos. El experimento consiste en hacer un cambio en el rumbo de un barco; partiendo de un rumbo de referencia, se varía éste primero a -5º y, una vez estabilizado, a +10º. La línea azul oscura representa la respuesta del sistema en grados ante el cambio en la consigna del timón, y la línea azul clara es la señal de control que se aplica al servo. La representación de esta última variable es simplemente para comprobar la agresividad con la que se opera sobre el actuador, y sus unidades no se corresponden con las de la consigna del timón; se trata de una unidad proporcional a la fuerza que se aplicaría.

La respuesta es satisfactoria en todos los casos teniendo en cuenta los puntos de operación de cada ejemplo. Cabe destacar que a velocidades pequeñas es necesario saturar la salida del controlador de una forma más brusca, opción no deseable, para mantener las especificaciones dentro de un rango de valores aceptables.

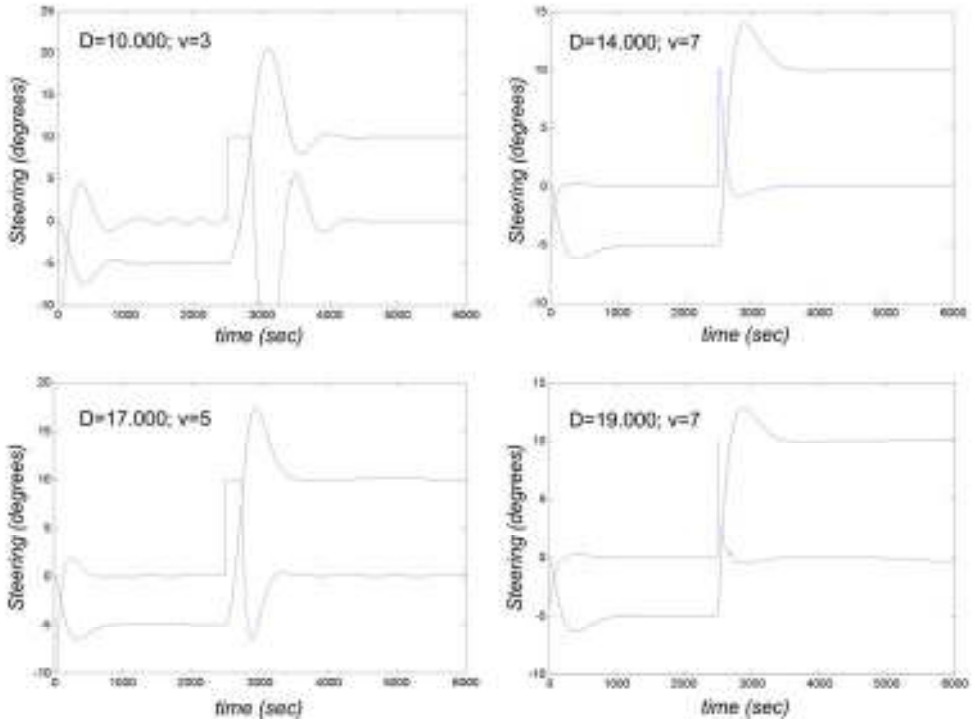


Figura 5: Respuestas de validación del sistema. Rumbo y potencia aplicada al timón (señal de control)

5 CONCLUSIONES Y TRABAJOS FUTUROS

En sistemas no lineales como en el direccionamiento de buques, donde además se trabaje en un rango amplio de condiciones de funcionamiento, una alternativa de control muy eficiente es la de Ganancia Programada, siempre que la dinámica del sistema se pueda aproximar linealmente por zonas.

En este trabajo se ha mostrado la viabilidad de adoptar esta estrategia, usando redes neuronales para la interpolación de los parámetros de sintonía de los controladores PID entre los puntos de operación. Con esto se consigue mantener unas especificaciones a lo largo de todo el rango de operación.

Como trabajos futuros se pretende hacer la transición de los parámetros de ajuste de los controladores con técnicas borrosas, de forma que la transición de las ganancias para los distintos puntos de operación no sea brusca, sino gradual, y evitar así posibles inestabilidades.

REFERENCIAS

[1] Applebaum, E., (2003) “Fuzzy Gain Scheduling for Flutter Suppression in Unmanned Aerial

Vehicles", *22nd Int. Conf. North American* 323-328, NAFIPS.

[2] Aström, K.J.; Hägglud, T., (1984) "Automatic tuning of simple regulators with specifications on phase and amplitude margins", *Automatica* 20, 645-651.

[3] Aström, K.J.; Wittenmark, B., (1989) Adaptive Control, Addison Wesley Publishing Company.

[4] Aström, K.J.; Hägglud, T., (2001) "The future of PID control", *Control Engineering Practice* 9 1163-1175.

[5] Aström, K.J.; Hägglund, T., (2002) Revisiting the Ziegler–Nichols tuning rules for PID control, Asian Journal of Control 4 (4) 364–380.

[6] Aström, K.J.; Hägglud, T., (2006) PID controllers: Theory, Desing and Tuning, ISA 2ª ed. Research Triangle Park.

[7] Bech, M.I.; Smith., L.W., (1969) Analogue Simulation of Ship Manoeuvers, Technical Report Hy-14, Hydro and Aerodynamics Laboratory, Lyngby, Denmark.

[8] Bennett, S., (1984) "Nicolas Minorsky and the automatic Steering of Ships", *Control System Magazine*, 10-15.

[9] Chai, J.S.; Shaohua, T.; Hang, C.C., (1996) "Gain Scheduling control of nonlinear plant using RBF neural network", *Proc. IEEE Int. Symposium on Intelligent Control*. 502-507, Michigan.

[10] Chiu, C.S; Lian, K.Y.; Liu, P., (2005) "Fuzzy Gain Scheduling for Parallel Parking a Car-Like Robot", *IEEE Transactions on control systems technology*, 13, 6, 1084-1092.

[11] Clement, B.; Duc, G., (2001) "An interpolation Method for Gain Schedulling", *Proc. 40th IEEE Conference on Decision and Control*, 1310-1315, Florida.

[12] Feng, Y. L.; Tan, K.C., (1998) "PIDeasyTM and automated generation of optimal PID controllers", *3rd Asia-Pacific Conf. Control & Measurement*, China, 29-33.

[13] Hoppensteadt, F.C., (1966) "Singular Perturbations on the Infinite Interval". Transactions of the American Mathematical Society, 123, 521-535.

[14] Hornik, K., (1991) "Approximation capabilities of multilayer feedforward networks". *Neural Networks*, 4, 2, 251-257.

[15] Jeng, J.T.; Lee, T.T., (1999) "A Neural Gain Scheduling Network Controller for Nonholonomic Systems", *IEEE Trans. Systems, Man, and Cybernetics*, 29, 6, 654-661.

[16] Nomoto, K.; Taguchi, T.; Sling, K.; Hirano, S., (1957) On the steering Qualities of Ships, Technical report, International Shipbuilding Progress, Vol.4.

[17] Norbbin. N.H., (1963) On the Design and Analyses of the Zig-Zag Test on Bases of Quasi to Line Frequency Response, Technical Report B 104-3, The Swendish State Experimental Shipbuilding Tank (SSPA), Gothenburg, Sweden.

[18] Santos, M., (2011) "Un enfoque aplicado del Control Inteligente", *Revista Iberoamericana de Automática e Informática Industrial*, 8, 283-296.

[19] Santos, M.; Dexter, A.L., (2002) "Control of a cryogenic process using a Fuzzy PID Scheduler", *Control Engineering Practice*, 10, 1147-1152.

CONTROL HÍBRIDO DIFUSO-DESLIZANTE PARA CONVERTIDOR DC/DC TIPO BOOST

Mónica L. Vásquez F.
Ingeniería Eléctrica, Universidad Distrital FJC, Colombia, monica_vasquez_27@hotmail.com

Edwar Jacinto G.
Ingeniería en Control e Instrumentación, Universidad Distrital FJC, Colombia, ejacintog@udistrital.edu.co

Fredy H. Martínez S.
Ingeniería Eléctrica, Universidad Distrital FJC, Colombia, fhmartinezs@udistrital.edu.co

Abstract

El presente artículo explora la operación en laboratorio de un esquema de control híbrido difuso-deslizante planteado por el grupo de investigación como estrategia de control para convertidores DC/DC tipo boost, estrategia que permita garantizar la estabilidad de operación, la regulación del voltaje de salida, y un reducido comportamiento transitorio en zonas de operación extrema. Para lograr esto, el esquema propuesto plantea la utilización de diferentes superficies de control, las cuales son unificadas en una única estructura gracias a la utilización de reglas de control difusas. La exploración culmina con la evaluación exitosa de un prototipo de laboratorio de 10 W que opera con la estructura de control propuesta.

Index Terms: Control deslizante, control difuso, convertidor boost.

1 INTRODUCCIÓN

Los reguladores electrónicos de potencia DC, o convertidores DC/DC, son parte fundamental del funcionamiento de la mayoría del equipo electrónico actual, así como de otras aplicaciones de potencia eléctrica que consumen corriente DC [1, 2]. Corresponden a un tipo de fuente de alimentación conmutada con dos interruptores semiconductores (al menos uno de los dos controlable) y algún elemento para almacenar energía (normalmente algún elemento inductivo, aunque es posible utilizar un condensador) [3].

Estos esquemas necesitan una corriente directa, pero dado que la magnitud varía dependiendo del tipo de aplicación, se hace necesario utilizar elevadores, reductores, elevadores/reductores o inversores. Su utilización se ha hecho extensiva dada la gran eficiencia que presenta al manejar grandes potencias, frente a otras soluciones como los reguladores lineales, y la gran capacidad para reducir tamaño y costo de los equipos de transformación dada la posibilidad de trabajar con frecuencias muy por encima de los 60 Hz.

El uso de estos convertidores tiene asociado a ellos ciertos problemas desde el punto de vista del control [4, 5]. Debido a la conmutación, su dinámica es altamente no lineal. Las estrategias de diseño tradicional tratan en general de linealizar su comportamiento alrededor del punto de operación del convertidor, lo cual permite el diseño con técnicas lineales. Sin embargo, el convertidor real no es lineal, y la robustez del esquema tiende a ser muy baja fallando la operación cuando se presentan cambios en la carga o en el voltaje de entrada. Además, se deben controlar condiciones especiales de operación como el arranque del convertidor, lo que dificulta el diseño de un control robusto, rápido y de buena regulación.

El grupo de investigación ha desarrollado previamente de forma exitosa esquemas de control difuso directo para solucionar el control de estos convertidores, agregando incluso condiciones de operación especiales como la corrección activa del factor de potencia (reconstrucción de la corriente de entrada) [6]. Continuando los trabajos en control inteligente para el convertidor boost, se propuso implementar un esquema de control híbrido difuso-deslizante, que mediante la definición de por lo menos dos superficies de control diferentes; una para condición estacionaría (operación cerca del *Set Point*) y otra para condición transitoria (durante el arranque, por ejemplo), se lograra garantizar la correcta operación del convertidor con una muy buena respuesta dinámica.

Para la conmutación entre las superficies de control, se propuso mezclar el esquema con los principios del control difuso, utilizando la máquina de inferencia de Takagi-Sugeno para lograr evaluar las diferentes superficies, a fin de lograr una conmutación suave de superficie a superficie. En cuanto a la estabilidad global del esquema, se garantiza asegurando la estabilidad individual en

cada una de las superficies de control.

El modelo que se asume para el convertidor de potencia es híbrido dados los dos comportamientos caracterizados en su dinámica. Dependiendo de las condiciones de operación del convertidor (arranque, cambios de carga, etc.) el sistema debe conmutar entre éstos dos comportamientos. esto significa que el espacio de estados del sistema se encuentra dividido en un conjunto de dos regiones L_1 y L_2. Una primera región L_1 cuando cuando el convertidor se encuentra en condición estacionaría, es decir, operación cerca del *Set Point*, y una segunda región L_2 cuando opera en alguna condición transitoria, por ejemplo durante el arranque. Cada una de estas dos regiones se caracteriza por un conjunto de ecuaciones diferenciales con variables de estado definidas como:

$$x_i \quad / \quad i = 1,\, 2,\, 3,\, \cdots,\, b \tag{1}$$

un conjunto de dimension b que permite describir la dinámica de los dos tipos de comportamiento. El sistema puede ser descrito entonces como:

$$\begin{gathered} H_p = \{\boldsymbol{X}_p,\, \boldsymbol{L}_p\} \\ \boldsymbol{X}_p \subset \mathbb{R}^b \end{gathered} \tag{2}$$

donde p es el número de comportamientos identificados en el sistema, en éste caso dos, y $\boldsymbol{X}$ es la representación en el espacio de estados para cada una de las p regiones del espacio de estados del sistema.

Bajo estas condiciones, el diseño y análisis de la estrategia de control del convertidor no solo toma ideas de investigaciones relacionadas con control de sistemas basado en el comportamiento [7, 8, 9, 10], sino que además ésta estructura, conocida en la literatura como autómata híbrido (*hybrid automaton*), permite el uso en el diseño de la capacidad de abstracción de los sistemas híbridos [11, 12, 13, 14].

A continuación, en la Sección 2 se presentan los antecedentes del diseño y la formulación del problema que permiten caracterizar el problema y proyectar el perfil de la solución. En la Sección 3 se resume el diseño del control híbrido difuso-deslizante propuesto, iniciando con la definición de la planta de prueba y su modelo lineal, y finalizando con la formulación del esquema de control. En la Sección 4 se muestran los detalles de la construcción física del prototipo y su evaluación en laboratorio. Aquí se ilustran dos esquemas de control para evaluación de desempeño: uno en lazo abierto con ciclo útil constante y el otro el control híbrido propiamente dicho. Finalmente, en la Sección 5 se presentan las conclusiones de la investigación.

2 FORMULACIÓN DEL PROBLEMA

La dinámica de los convertidores DC/DC de potencia incluye comportamientos altamente no lineales, esto en conjunto con la necesidad de altas velocidades de respuesta para variaciones en el voltaje de entrada y/o la carga de salida, conduce al desarrollo de estructuras de control que superen en desempeño a los tradicionales controles PID.

Como estrategias de solución desde el punto de vista del control, en los últimos años ha crecido el interés por la utilización de técnicas computacionales inteligentes, destacándose el natural paralelismo de las redes neuronales, y su habilidad para aproximar la dinámica desconocida de un sistema, y por tanto, asumir el comportamiento ideal de un esquema de control; y la simplicidad del control difuso, que ofrece además una respuesta de alta calidad al mismo tiempo que brinda una baja sensibilidad a entradas ruidosas e imprecisas.

Particularmente, la integración de un sistema basado en lógica difusa y un sistema de control en modo de deslizamiento se reporta en muchas aplicaciones con la intención de utilizar la lógica difusa para resolver problemas propios de la operación de los esquemas de control en modo de deslizamiento, o simplemente para mejorar su desempeño [15, 16, 17].

En el articulo [18] se presenta una serie de estructuras a través de las cuales se puede agregar 'inteligencia' a los controles en modo deslizante, al mismo tiempo que referencia una amplia lista de aplicaciones con lógica difusa, misma que ha servido de base para el análisis aquí presentado.

2.1 CONVERTIDOR BOOST EN LAZO ABIERTO

Partiendo del modelo en variables de estado promedio del convertidor Boost, es posible obtener la función de transferencia como se muestra en la ecuación 3. En esta ecuación, se tomaron los valores de diseño del prototipo Boost desarrollado en el proyecto (en las siguientes secciones se presentan los detalles de diseño): convertidor Boost operando a 25 kHz, alimentado desde una fuente DC de valor nominal 12 V DC, voltaje de salida deseado de 24 V DC, con un choque medido en laboratorio de 1 mH, un filtro de salida de 10 uF, una carga resistiva nominal de 60 Ω, ciclo útil esperado en estado estacionario de 0,5, resistencia parásita en el choque de 0,7 Ω y en el condensador

de salida de 0,5 Ω.

$$\frac{V_0(S)}{d(S)} = \frac{-74000S + 1,2 \times 10^9}{S^2 + 1366,7S + 2,5 \times 10^7} \qquad (3)$$

Esta función de transferencia corresponde a un filtro pasa bajos de segundo orden con dos polos conjugados complejos del lado izquierdo del plano complejo. El filtro tiene una frecuencia de corte:

$$\omega_0 = \frac{1-d}{\sqrt{LC_0}} = 5000\,\text{rad/s} \qquad (4)$$

En la ecuación 4 se observa que la frecuencia de corte es función del valor del ciclo útil. En un control en lazo cerrado en modo de voltaje, ésta dinámica cambia cuando se ajusta el valor del ciclo útil, es decir, la función de transferencia del sistema cambia continuamente. Esto hace que la función de transferencia del regulador Boost sea una función no lineal respecto al ciclo útil.

2.2 ESQUEMA HÍBRIDO DIFUSO-DESLIZANTE

Para el diseño de éste tipo de control, se utiliza el movimiento promedio de la corriente en el choque de entrada y del voltaje de salida como variables de estado, lo que se conoce formalmente como variables de estado promedio, esto permite simplificar considerablemente el proceso de diseño. La estructura híbrida difuso-deslizante se presenta como una consecuencia lógica de las limitaciones del control en modo deslizante y las bondades que permite agregar la lógica difusa a este esquema.

Atendiendo al concepto del control en modo deslizante, se define para el convertidor Boost a $\overline{v_0(t)}$ como el voltaje promedio en la salida, y V_{ref} como el voltaje de salida deseado. Por lo tanto, dado que el objetivo de control es la regulación del voltaje de salida, la superficie de deslizamiento en variables de estado para el control en estado estacionario (cerca del *set point*) esta descrita por $\overline{v_0(t)} = V_{ref}$, de tal forma que la superficie de deslizamiento de voltaje constante queda definida por:

$$S = \left\{ \boldsymbol{X}(t) \in \mathbb{R}^2 \quad : \quad \overline{v_0(t)} = V_{ref} \right\} \qquad (5)$$

En el trabajo [19] se presenta un estudio de esta superficie de deslizamiento de voltaje constante para varios convertidores, incluido el convertidor boost, llegándose a la conclusión de que aunque existe régimen de deslizamiento, el punto de equilibrio al que se llega para la corriente en el choque es inestable. Debido a esto, no es posible realizar un control de voltaje de salida con una superficie que sólo este formada por el error de voltaje de salida ($V_{ref} - \overline{v_0(t)} = 0$), o lo que es lo mismo, la regulación del voltaje de salida se obtiene a expensas de la inestabilidad de la corriente en el choque.

En este mismo trabajo también se muestra que para la corriente en el choque, $\overline{i_L(t)}$, con corriente constante ($\overline{i_L(t)} = K$), no solo existe régimen de deslizamiento sino que el punto de equilibrio es estable, razón por la cual se asume en éste trabajo. Así, la superficie de deslizamiento de corriente constante queda definida por:

$$S = \left\{ \boldsymbol{X}(t) \in \mathbb{R}^2 \quad : \quad \overline{i_L(t)} = K \right\} \qquad (6)$$

Sin embargo, el objetivo primario del convertidor sigue siendo la regulación del voltaje de salida, razón por la cual es necesario eliminar el error estacionario del mismo. De la teoría clásica de control, se sabe que esto es posible agregando una red de compensación PI, que se asumirá inicialmente como solución. Tanto la componente proporcional como la integral permiten reducir el tiempo de subida y el error de estado estacionario; sin embargo, también incrementan el sobrepaso. En primera instancia, no se considerará el problema del sobrepaso en los transitorios para el diseño de esta red PI, dado que como se mencionó antes, esta superficie operaría solamente cerca del punto de equilibrio.

Este nuevo lazo de realimentación PI modifica la expresión de la superficie de control como se muestra en la ecuación 7. Como es claro, el término integral garantiza la desaparición del error en estado estacionario.

$$S = \left\{ \boldsymbol{X}(\mathbf{t}) \in \mathbb{R}^{\mathbf{2}} \quad : \qquad (7) \right.$$

$$\left. \overline{i_L(t)} = k_1 \left(V_{ref} - \overline{v_0(t)} \right) + k_2 \int \left(V_{ref} - \overline{v_0(t)} \right) dt \right\}$$

La presencia de este termino integral conlleva a la aparición de problemas durante el transitorio de arranque, ya que si se observa la estructura de la ecuación 7, es obvio que éste impedirá el cambio de signo en la superficie de control. Además, se complica la selección de los parámetros del lazo de re-alimentación, ya que si se busca un sistema robusto se terminará también con uno lento en cuanto a respuesta frente a variaciones en la carga o en la entrada.

Así las cosas, se podría pensar en la necesidad de dos superficies de control diferentes para resolver el problema, una que opere cerca del punto

de equilibrio estacionario, y otra que opere durante el arranque del convertidor. Si se tiene en mente la estructura básica del esquema de control difuso, se puede concluir con facilidad el por qué mezclándolos en una estructura deslizante se podría resolver el problema de control global.

Un control basado en lógica difusa utiliza reglas de control del tipo SI ... ENTONCES, cuyos consecuentes resultan ser combinaciones lineales de las variables de entrada. Es decir, es posible construir una superficie de control diferente para cada regla del sistema, lo que permite utilizar la superficie de control más apropiada en cada momento.

La figura 1 muestra el esquema de control propuesto. En ella, se pueden apreciar dos lazos de re-alimentación, estructura muy común en los correctores activos de factor de potencia, uno de corriente y uno de voltaje. El lazo de corriente implementa la superficie de control:

$$S = \left\{ \mathbf{X}(t) \in \mathbb{R}^2 \quad : \quad \overline{i_L(t)} = I_{ref} \right\} \qquad (8)$$

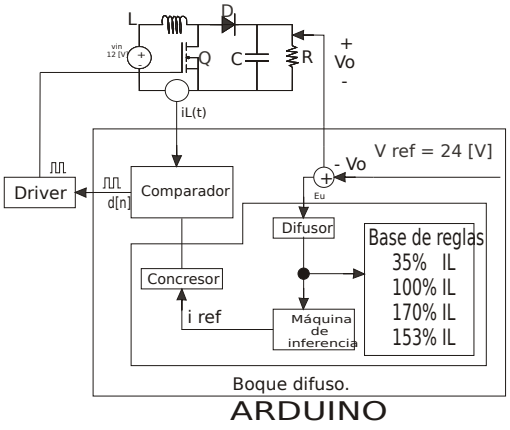


Figura 1: Esquema del control híbrido propuesto.

La dinámica de este lazo es mucho más rápida que la del lazo de voltaje, debido a que se requiere una rápida respuesta dinámica en función a la energía que maneja el convertidor, de forma similar a como lo realiza el control en modo de corriente. El lazo de voltaje, de dinámica más lenta a fin de evitar oscilaciones en el voltaje de salida a causa de ruido en la señal re-alimentada, es el encargado de establecer la referencia de corriente para el lazo interno (lazo de corriente).

Teniendo en cuenta la función que debe cumplir el control difuso en esta aplicación, que la acción de control se realiza cada cierto intervalo de tiempo dependiente de la unidad que implementa el control (hardware) y la frecuencia natural del sistema, y suponiendo que el valor DC de la corriente en el choque es automáticamente ajustado por el convertidor de acuerdo a la condición de balance de potencia, se seleccionan como variables de los antecedentes a: (1) el error del voltaje de salida, Eu, y (2) corriente en el choque del convertidor, i_L. Por lo tanto, los consecuentes que permitirán incluir en el conjunto de reglas las funciones de transferencia de los controladores PI diseñados alrededor del punto de equilibrio, tendrán la siguiente forma:

$$I_{ref}(t) = K_P Eu(t) + K_I \int Eu(t)\, dt \qquad (9)$$

En donde las constantes K_P y K_I son las constantes del lazo PI diseñadas para que el sistema opere en estado estacionario (cerca del *set point*, cerca del punto de equilibrio). Cuando el convertidor opera lejos del punto de equilibrio (por ejemplo, durante el arranque), la referencia de corriente se asume como un valor constante de corriente. El sistema de lógica difusa permite que las transiciones entre estas dos estrategias de control, es decir, la re-definición de la corriente de referencia de la ecuación 9 en estado estacionario y el valor constante del arranque, no se realice de forma brusca.

3 DISEÑO

3.1 DISEÑO DE LA PLANTA

La planta esta conformada por un pequeño convertidor Boost de 10 W. La figura 2 muestra el detalle del convertidor, y la tabla 1 sus valores nominales de diseño.

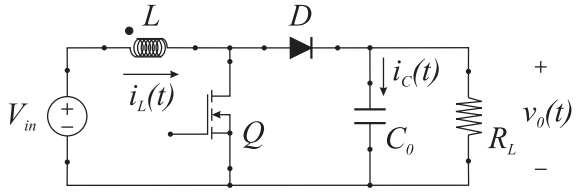


Figura 2: Circuito esquemático del convertidor elevador Boost. Mediante la conmutación de Q, y el almacenamiento de energía en L, se logra la regulación del voltaje de salida $v_0(t)$.

De acuerdo a los valores nominales definidos para la planta (tabla 1), se realizó el diseño del convertidor en operación en modo continuo de conducción (CCM).

- Inductancia crítica:

$$L_C = \frac{R_L D (1-D)^2}{2 f_S} = 144\ \mu\text{H} \qquad (10)$$

Tabla 1: Valores de diseño nominales del convertidor.

Parámetro	Símbolo	Valor
Voltaje DC salida (nominal)	V_0	24 V
Voltaje DC entrada	V_i	12 V
Potencia salida (nominal)	P_0	10 W
Corriente de salida	I_0	0.417 A
Frecuencia de conmutación	f_S	25 kHz
Rizado voltaje en salida	V_r	2.4 V (10% V_0)
Rizado corriente en choque	$\triangle I_L$	0.21 A (20% I_{in})

- Inductancia del choque:

$$L \approx (12.6)\frac{\triangle t}{\triangle I} = 1.2 \text{ mH} \tag{11}$$

- Selección del MOSFET:

$$V_Q = V_0 + 5\%V_0 + 10 = 35.2 \text{ V} \tag{12}$$

$$I_{Qrms} = 2\sqrt{2}\,(1)\sqrt{\frac{1}{6} - \frac{4\sqrt{2}\,(12)}{9\pi\,(24)}} = 0.73 \text{ A} \tag{13}$$

- Condensador de salida:

$$C_0 \geq \frac{(0.417)\times\left(1-\frac{V_{imin}}{V_0}\right)}{(25000)\times(2.4)} = 4.52\ \mu\text{F} \tag{14}$$

$$ESR \leq \frac{\triangle V_0}{\frac{I_{0max}}{1-D_{max}} + \frac{\triangle I_L}{2}} = 1.85\ \Omega \tag{15}$$

3.2 DISEÑO DEL CONTROL

Con el fin de combinar como salida deseada del sistema controlado, un transitorio de arranque suave con respuesta rápida y operación robusta con mínimo error estacionario cerca al punto de equilibrio (cerca del régimen estacionario), se optó por un lado seleccionar cuatro superficies de deslizamiento de corriente constante para la condición transitoria de arranque, y una superficie que introduce el efecto PI en operación estacionaria.

De acuerdo con los parámetros de diseño del circuito de potencia, y la eficiencia esperada del mismo, en el choque de entrada del regulador Boost se espera una corriente nominal estacionaria promedio de 1 A. Por lo tanto, recordando que el control debe limitar la corriente de arranque del convertidor, se seleccionan los siguientes valores de corriente de referencia para las cuatro superficies deslizantes durante el arranque:

- Cuando el voltaje de salida esta muy por encima de su valor de referencia: $I_{ref} = 35\%$ $I_{Nominal} = 0{,}35$ A.
- Cuando el voltaje de salida esta por encima de su valor de referencia y aún se encuentra lejos del mismo: $I_{ref} = 100\%$ $I_{Nominal} = 1{,}0$ A.
- Cuando el voltaje de salida esta muy por debajo de su valor de referencia: $I_{ref} = 170\%$ $I_{Nominal} = 1{,}7$ A.
- Cuando el voltaje de salida esta por debajo de su valor de referencia y aún se encuentra lejos del mismo: $I_{ref} = 135\%$ $I_{Nominal} = 1{,}35$ A.

Para el diseño de la red PI, el principio de diseño dicta que el control debe responder lo más rápidamente posible sin importar los sobre-pasos que se puedan generar (claro, estos serán asumidos por las otras superficies de control). Así, siguiendo los criterios de diseño clásicos para esquemas de control PI, se escoge el compensador mostrado en la ecuación 16.

$$G_{PI}\,(S) = \frac{80S + 100}{S} \tag{16}$$

Como se detalló en la Sección 2.2, la definición de la corriente de referencia, I_{ref}, depende del error de voltaje Eu tanto para el bloque PI cerca del punto estacionario, como para el control difuso en las zonas extremas, razón por la cual se toma al error de voltaje de salida como variable de entrada al bloque de control difuso-deslizante, definiéndose como:

$$Eu\,(t) = \frac{24 - v_0\,(t)}{24} \tag{17}$$

Variable que se encuentra normalizada en el rango [-1, 1]. $v_0\,(t)$ corresponde al voltaje de salida medido sobre el convertidor en cualquier instante t, y 24 corresponde al voltaje de referencia de salida, en el prototipo 24 V. Para la sección difusa, se define un total de siete conjuntos difusos para el error de voltaje de salida como se muestran en la Fig. 3.

Finalmente, las reglas de control para el bloque difuso quedan definidas de la siguiente forma:

- Si Eu es NG ENTONCES I_{ref} es 0,35 A.

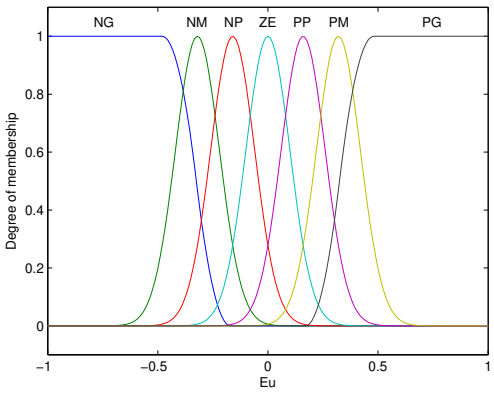


Figura 3: Conjuntos difusos para el error de voltaje de salida, sección difusa del control difuso-deslizante.

- Si Eu es NM ENTONCES I_{ref} es 1,0 A.
- Si Eu es NP ENTONCES I_{ref} es $80Eu + 100\frac{Eu}{S}$ A.
- Si Eu es ZE ENTONCES I_{ref} es $80Eu + 100\frac{Eu}{S}$ A.
- Si Eu es PP ENTONCES I_{ref} es $80Eu + 100\frac{Eu}{S}$ A.
- Si Eu es PM ENTONCES I_{ref} es 1,35 A.
- Si Eu es PG ENTONCES I_{ref} es 1,7 A.

La corriente de referencia suministrada por el bloque difuso es comparada directamente con la corriente en el choque, a fin de determinar el incremento a aplicar al ciclo útil actual, según la siguiente ecuación:

$$d[n] = \eta \times \delta d[n] + d[n-1] \tag{18}$$

η se utiliza como ganancia del incremento en el ciclo útil en cada actualización de la unidad de control, y para la aplicación final se ajustó en 0,01 a fin de producir variaciones suaves en el ciclo útil.

4 PROTOTIPO Y DESEMPEÑO

El choque se construyó sobre un núcleo T-157-26 de Amidon, con un total de 111 vueltas con conductor redondo de cobre 21 AWG, para un total de 1.2 mH de inductancia, y como transistor se utilizó el MOSFET IRF540 de ST Microelectronics (100 V - 22 A).

Con el propósito de realizar análisis de desempeño del esquema de control, se desarrolló un prototipo de esquema de control en lazo abierto, a fin de medir condiciones de referencia de operación del convertidor, condiciones que al ser comparadas con la operación del esquema difuso-deslizante, permitirían evaluar el desempeño mínimo del esquema propuesto.

Dado que se pretende realizar comparación de desempeño, se formularon dos principios básicos:

- La planta debería ser las misma para los dos esquemas de control, y debería operar durante la evaluación en idénticas condiciones de voltaje de salida y carga.
- Los dos esquemas de control, el de referencia en lazo abierto y el propuesto difuso-deslizante, deberían implementarse sobre estructuras hardware completamente independientes.

Siguiendo estos criterios, se optó por desarrollar un control en lazo abierto alrededor del PWM análogo TL494 con una frecuencia de conmutación de 25 kHz y con el circuito integrado TPS2811 de Texas Instruments como driver de la gate, y el control difuso-deslizante propuesto en lazo cerrado sobre el sistema digital Arduino Mega2560, que se soporta en el microcontrolador ATmega2560 de Atmel.

Para la conexión de Arduino con el convertidor de potencia, se desarrolló un circuito driver para la gate del MOSFET, en este caso utilizando dos transistores bipolares en configuración *push-pull*, conformando un circuito capaz de suministrar hasta 2,5 A a la gate.

La evaluación de desempeño se realizó midiendo el comportamiento de las variables de estado del convertidor en arranque y operación estacionaria. Luego se observó el desempeño del control frente a un incremento de la carga de salida del 20%.

En las figuras 4 y 5, y en el plano de fase mostrado en la figura 6, se puede observar el comportamiento de las variables de estado del sistema luego de 100 ms de operación. Como se aprecia, el sistema busca su punto de equilibrio a través de una variación suave de sus variables de estado. La corriente de entrada no supera los 2,3 A, valor muy inferior al pico de 5,5 A registrado por el sistema de control en lazo abierto. El voltaje de salida tampoco presenta grandes sobre-pasos (inferior a 35 V), los cuales son normales de esperar en un sistema con filtro capacitivo en su salida. El valor inferido por la unidad de control difuso-deslizante durante este mismo intervalo de tiempo, se puede observar en la figura 7.

El desempeño dinámico del esquema de control se evaluó aplicando un cambio de carga al conver-

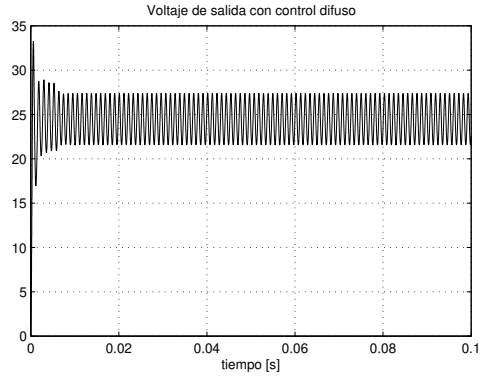


Figura 4: Voltaje de salida en arranque con carga nominal ($R_L = 60\Omega$).

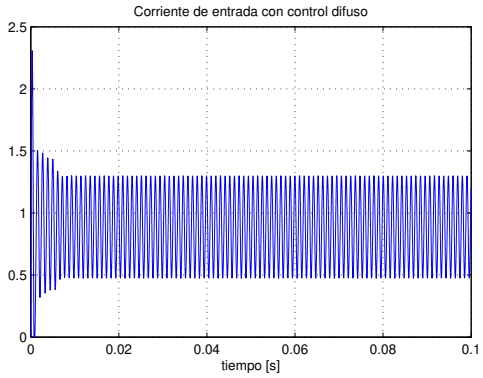


Figura 5: Corriente de entrada en arranque con carga nominal ($R_L = 60\Omega$).

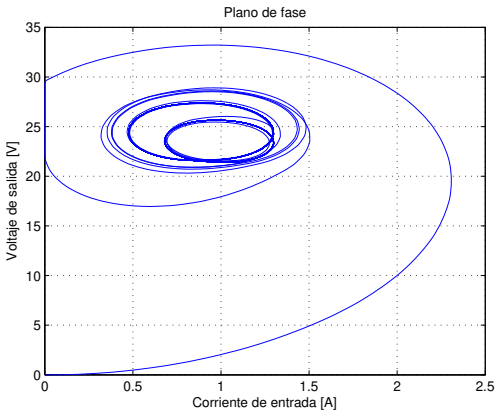


Figura 6: Plano de fase control difuso-deslizante.

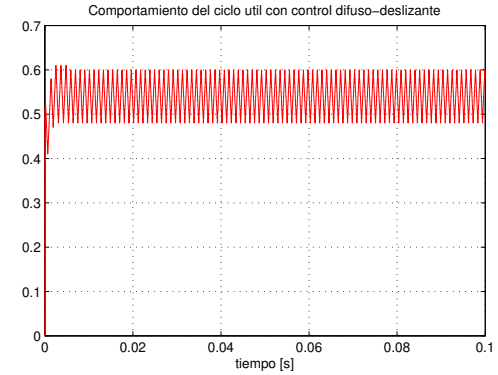


Figura 7: Comportamiento del ciclo útil durante los 100 ms iniciales del control difuso-deslizante.

tidor y midiendo el comportamiento en las variables de estado. La carga se incrementó en un 20%, reduciendo su valor resistivo de 60Ω a 50Ω. Las figuras 8 y 9 muestran el comportamiento detectado tanto en el voltaje de salida como en la corriente de entrada.

5 CONCLUSIONES

En éste trabajo de investigación se ha realizado la evaluación de una implementación de un control híbrido en donde la lógica difusa es utilizada como herramienta auxiliar en el diseño de controles deslizantes en modo de corriente para reguladores conmutados DC/DC tipo boost. Este convertidor, de particular interés para el grupo de investigación ARMOS, posee una dinámica de fase no mínima alrededor de su punto de equilibrio, raíz del interés en el desarrollo de su esquema de control.

Se desarrolla una estructura de control híbrida difuso-deslizante para el convertidor elevador tipo boost, utilizando como plataforma base el control en modo de corriente, e implementando estrategias de control deslizante y sistemas de lógica difusa a fin de introducir algún grado de comportamiento adaptativo en el sistema.

Los resultados sobre un prototipo de laboratorio con potencia de salida de 10 W, muestran no sólo la capacidad de regulación del voltaje de salida, sino estabilidad, una alta velocidad de respuesta, valores reducidos de transitorios, principalmente el de corriente de arranque, y una gran inmunidad frente a cambios en la carga de salida. Esta excelente respuesta dinámica es incluso mejor que la reportada por el control difuso directo en investigaciones previas del grupo de investigación, en donde frente a cambios en la carga se registraron sobrepasos en la salida del 12% al 18% con tiempos de estabilización de 50 a 70 ms.

El hecho de que el sistema de control híbrido se estructure sobre un control en modo de corriente, deja abierta la posibilidad de replantear las superficies de control, es decir, las corrientes de referencia, a fin de incluir en esta estructura capacidades adicionales de corrección activa del factor de potencia.

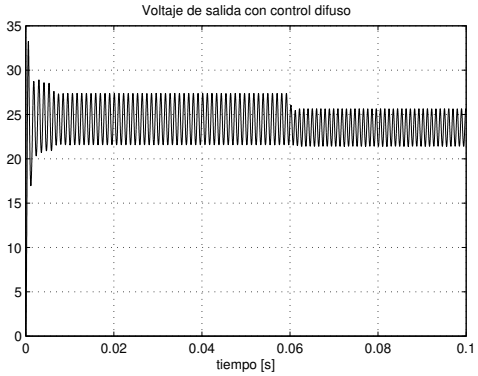


Figura 8: Voltaje de salida en arranque con variación de carga (de 60Ω a 50Ω).

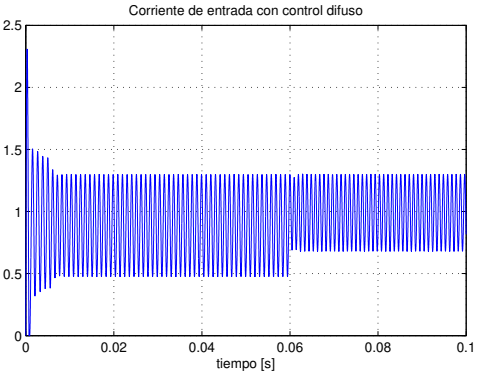


Figura 9: Corriente de entrada en arranque con variación de carga (de 60Ω a 50Ω).

Agradecimientos

Este trabajo fue financiado por la Universidad Distrital Francisco José de Caldas. Los puntos de vista expresados no son necesariamente respaldados por la Universidad Distrital.

Referencias

[1] Popović-Gerber, J., Oliver, J., Cordero, N., Harder, T., Cobos, J., Hayes, M., O'Mathuna, S., and Prem, E. (2012) *IEEE Transactions on Power Electronics* **27(5)**, 2338–2353 ISSN 0885-8993.

[2] Steimer, P. K. (2010) In 2010 International Power Electronics Conference (IPEC) : ISBN 978-1-4244-5394-8 pp. 11–15.

[3] Martínez, F. H. (2003) *Revista Tecnura* **1(12)**, 31–41 ISSN 012-3921X.

[4] Ling, B.-K., Bingham, C., Lu, H.-C., and Teo, K.-L. (2012) *Control Theory & Applications IET* **6(13)**, 1973–1983.

[5] Martinez S, F., Gomez M, D., and Castiblanco O, M. (2010) In 2010 International Power Electronics Conference (IPEC) : ISBN 978-1-4244-5394-8 pp. 53–57.

[6] Martinez S., F.H.and Gomez M, D. (2007) In 7th Internatonal Conference on Power Electronics ICPE '07 : ISBN 978-1-4244-1871-8 pp. 936–940.

[7] Egerstedt, M. and Hu, X. (2002) *Automatica* **38(1)**, 125–130.

[8] Gamage, G. W., Mann, G., and Gosine, R. G. (2009) In Proc. Canadian Conf. Electrical and Computer Engineering CCECE '09 : pp. 660–663.

[9] Lan, Y., Yan, G., and Lin, Z. (2009) In Proc. 48th IEEE Conf. held jointly with the 2009 28th Chinese Control Conf Decision and Control CDC/CCC 2009 : pp. 3032–3037.

[10] Hernandez-Martinez, E. G. and Albino, J. M. F. (2011) In Proc. IEEE Int Control Applications (CCA) Conf : pp. 941–946.

[11] Chaohong, C., Rafal, G., Sanfelice, R. G., and Teel, A. R. (2007) In Proc. Chinese Control Conf. CCC 2007 : pp. 29–36.

[12] Prieur, C., Goebel, R., and Teel, A. R. (2007) **52(11)**, 2103–2117.

[13] Zhang, H. and James, M. R. (2006) In Proc. 45th IEEE Conf. Decision and Control : pp. 4801–4806.

[14] Yoon, K.-H., Lee, J.-K., Kim, K.-H., Park, B.-S., and Yoon, J.-S. (2008) In Proc. Int. Conf. Control, Automation and Systems ICCAS 2008 : pp. 1367–1372.

[15] Mattavelli, P., Rossetto, L., Spiazzi, G., and Tenti, P. (1997) *IEEE Transactions on Power Electronics* **12(1)**, 79–86 ISSN 0885-8993.

[16] Geng, J., Zhang, C., and Luo, C. (2007) In 2nd IEEE Conference on Industrial Electronics and Applications ICIEA 2007 : ISBN 978-1-4244-0737-8 pp. 1907–1911.

[17] Tan, S.-C., Lai, Y., Tse, C., Martínez-Salamero, L., and Wu, C.-K. (2009) *IEEE Transactions on Industrial Electronics* **54(6)**, 3276–3286 ISSN 0278-0046.

[18] Kaynak, O., Erbatur, K., and Ertugnrl, M. (2001) *IEEE Transactions on Industrial Electronics* **48(1)**, 4–17 ISSN 0278-0046.

[19] Sira-Ramirez, H. (1987) *IEEE Transactions on Circuits and Systems* **34(8)**, 919–933 ISSN 0098-4094.

Social and Smart SANDS

Raúl Feliz Alonso
Fundación Cartif, Boecillo, Valladolid
Email: raufel@cartif.es

Eduardo Zalama Casanova
ITAP-DISA, Universidad de Valladolid
ezalama@eii.uva.es

Jaime Gómez García-Bermejo
ITAP-DISA, Universidad de Valladolid
jaigom@eii.uva.es

Bruno Apolloni
Department of Information Science, University of Milan
apolloni@di.unimi.it

Resumen

Este artículo presenta la iniciativa SANDS, un proyecto a desarrollar en los próximos años que tiene como objetivo la aplicación de los sistemas sociales inteligentes a la gestión de los electrodomésticos y servicios domésticos del hogar. El artículo presenta la arquitectura básica de SANDS describiendo su criterio funcional, comunicacional y estructural. El proyecto se enmarca dentro del programa ICT 2011.1.6 Future Internet Research and Experimentation.

Palabras Clave: redes sociales, internet de las cosas, domótica.

1 INTRODUCCIÓN

Las redes sociales pueden ser vistas como un repositorio de información que puede ser consultado cuando sea necesario para resolver problemas, conocer los procedimientos de resolución de los mismos o simplemente obtener información relevante para una determinada tarea. Esta metodología está muy asentada en las ciencias sociales, donde las redes sociales han servido para difundir las innovaciones educativas en el campo de los cuidados médicos [1], gestionar los programas de desarrollo de productos [2], y para la difusión de las innovaciones agrícolas por parte de los agricultores [3]. Vannoy y Palvia [4] proponen como definición de la computación social "las acciones dentro de un grupo social y empresarial practicadas por consenso y cooperación del grupo, donde tales acciones son posibles gracias al uso de las tecnologías de la información, y donde la interacción del grupo provoca a los miembros se adapten a ella e influencie a otros miembros a unirse al grupo".

Dada una determinada red social gestionada por un software implementado como un servicio web, se puede distinguir entre dos tipos de computación: computación consciente y subconsciente. El primero viene determinado por las decisiones y acciones llevadas a cabo por los agentes sociales (también conocido como los usuarios) sobre la base de la información proporcionada por el servicio social. Este tipo de computación está muy cerca de la definición propuesta por Vannoy y de Palvia que se referenció en el párrafo anterior. El segundo tipo viene determinado por el procesamiento de datos realizado automáticamente y de forma autónoma por los servicios web con el fin de buscar o producir información para ofrecerla a los usuarios, o para otros fines, tales como la minería de datos. Este tipo de computación puede considerarse inteligente puesto que es capaz de proponer nuevas soluciones a determinados problemas.

El proyecto Social and Smart (SANDS) financiado por la Unión Europea, VII Programa Marco tiene como objetivo sentar las bases para la obtención de sistemas sociales inteligentes en el ámbito de los electrodomésticos y los servicios domésticos [5].

2 UNA APROXIMACIÓN A LA ESTRUCTURA DE SANDS

La *figura 1* muestra una representación intuitiva de la estructura de los miembros del sistema y su interacción. La red social SANDS actúa de intermediaria entre los usuarios y su propio conjunto de electrodomésticos. Dicha red social consta de un

repositorio de tareas propuestas por los usuarios, así como de un repositorio de recetas que describen diversas formas de uso de los electrodomésticos.

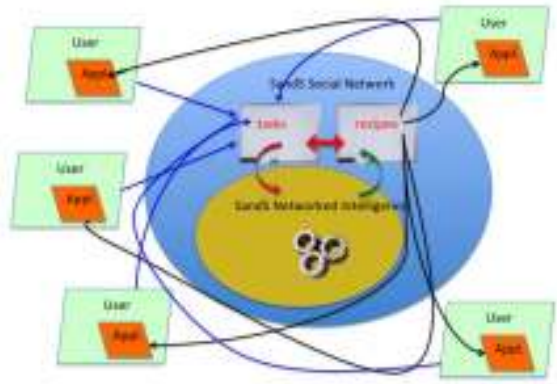


Figura 1: Esquema de la arquitectura prototipo de SANDS

Estos dos repositorios están relacionados mediante una serie de conexiones entre tareas y recetas. Dichas conexiones no tienen por qué ser uno-a-uno. Las flechas azules, por ejemplo, corresponden a las peticiones realizadas por los usuarios, mediante las cuales se solicita a la base de datos la resolución de una tarea específica. Si la tarea ya existe como tal, se envía al electrodoméstico correspondiente la receta adecuada para esa tarea (flechas negras). Sin embargo, cuando la tarea solicitada no exista como tal en el repositorio, la red social solicitará una solución para la misma a la "Inteligencia en red". Dicha solución consistirá en una nueva receta obtenida tomando como base la información o conocimiento ya almacenado en el repositorio de recetas. La receta, por tanto, será generada por medio de un sistema de razonamiento inteligente.

Para que el sistema inteligente pueda generar estas nuevas recetas deben existir otras recetas previas almacenadas en su repositorio correspondiente. Cuanto mayor sea el número de recetas almacenadas, lo que llamamos "conocimiento del sistema", mayor capacidad tendrá el sistema inteligente para generar nuevas recetas. Por tanto, los usuarios tendrán la capacidad de subir recetas para la resolución de determinadas tareas, basándose en su experiencia previa. Del mismo modo, los propios fabricantes podrán subir recetas para un correcto uso de los electrodomésticos que desarrollan. Esto es a lo que se llamará "computación consciente". A medida que vaya creciendo este repositorio, el sistema irá ofreciendo soluciones mejores y más precisas, por lo que mejorará sus prestaciones gracias a un sistema de aprendizaje reforzado. Las recetas producidas por el sistema sin la intervención de ningún usuario es lo que se llamará "computación subconsciente".

3 ARQUITECTURA BÁSICA DE SANDS

La arquitectura en la que se apoya el proyecto se detalla a continuación atendiendo a tres criterios:

3.1 Criterio Funcional:

Desde un punto de vista funcional, la arquitectura del sistema se puede ver en la *figura 2*.

Nodos Unificados (UN): constituyen la capa inferior y es una abstracción de lo que serían los electrodomésticos físicos, tales como frigorífico, lavadora, etc. Sus características principales serán identificar de manera inequívoca a cada dispositivo, representar las propiedades del dispositivo y ser un puente de comunicación bidireccional entre el dispositivo físico y la Infraestructura doméstica.

Infraestructura Doméstica (DI): está formada por una serie de módulos conectados que atienden e interpretan los comandos emitidos por el usuario a través de una interfaz apropiada. Para ello, debe ser capaz de interactuar con todos los dispositivos conectados en el hogar (UN) y de procesar reglas lógicas que determinen el comportamiento de dichos dispositivos. Los módulos de los que se compone son:
Estructura de Datos Extendidos (EDS): la información detallada que publican los UN sobre los dispositivos que representan se almacena aquí.
Manejador de Eventos (EH): es el núcleo de la DI. Este manejador recibe peticiones para eventos específicos y manda los comandos correspondientes a los dispositivos físicos. También se comunica con un bloque se control externo mediante un manejador de servicios "Hook".

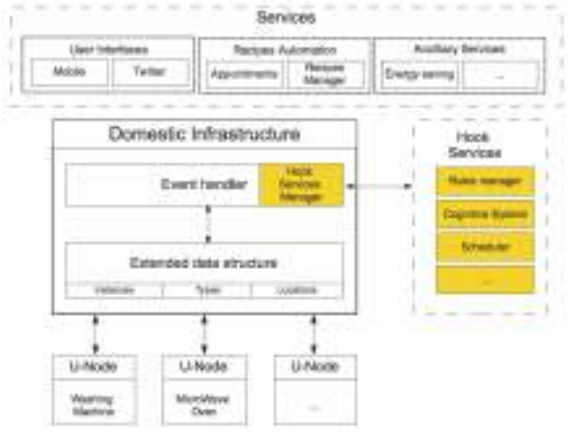


Figura 2: Una primera visión de la arquitectura SANDS

El bloque de Servicios Hook (HS): constituye la lógica del sistema. Está formado por:
Generador de Reglas (RM): comprueba y ajusta las instrucciones para hacerlas compatibles tanto con la lógica de las operaciones como con las propiedades de los dispositivos.
Sistema cognitivo (CS): aprende de los eventos y señales que llegan al sistema y con ello permite obtener un refinamiento de las instrucciones que genera.
Planificador (SC): controla, en tiempo de ejecución, el lanzamiento de las tareas que llevan a cabo la ejecución de los comandos.

La capa de servicios (SERV): es la parte con la que interactúan los usuarios. Son una serie de servicios que simplemente interactúan con la DI mediante el envío de comandos, sin tener conocimiento de la complejidad del sistema. En esta aproximación preliminar a la estructura final de SANDS, los servicios considerados son:
Interfaz de Usuarios (UI): constituye la parte del sistema con la que puede interactuar el usuario por medio de un móvil, tablet u otros dispositivos.
Automatización de Recetas (RA): es el encargado de producir recetas en la red social.
Servicios auxiliares (AS): conjunto de medios "*open-source*" facilitados por los fabricantes. Estos medios pueden encargarse de servicios como por ejemplo el ahorro de energía.

3.2 Criterio estructural:

Haciendo un símil con el cuerpo humano, se podría decir que el sistema SANDS tiene la cabeza en la nube y los pies en los electrodomésticos. El esquema general es el que se puede ver en la *figura 3*.

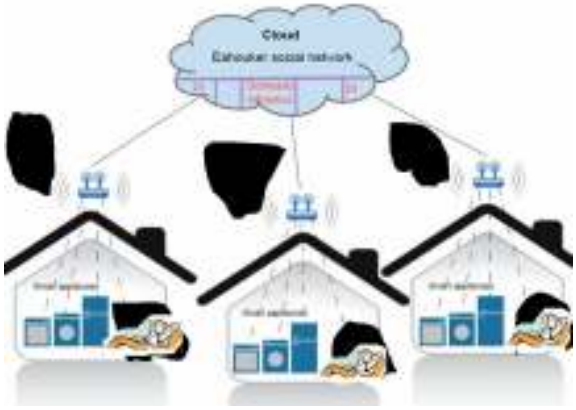


Figura 3: El sistema domótico

En la casa se encuentran los electrodomésticos y los usuarios. Ambos se conectan a internet a través de un router doméstico. Los primeros se conectan mediante algún dispositivo de trasmisión embebido mientras que los segundos están cómodamente en su sofá, enviando órdenes de vez en cuando a la DI. La forma en que el router se conecta al proveedor de internet no se verá alterada. A nivel local, para la conexión entre el router y los distintos electrodomésticos se empleará el mínimo hardware adicional posible para hacer factible el protocolo de comunicación desarrollado.

Por tanto, atendiendo a este escenario, las características del sistema serán:
1) Sistema lo menos invasivo posible. El hardware adicional será reducido a la mínima expresión. Los electrodomésticos serán los que se pueden encontrar habitualmente en las tiendas especializadas. Solamente se les añadirá una pequeña place electrónica que les permita conectarse a la red SANDS. Para este proyecto se ha optado por una placa Arduino. No hay necesidad de ordenadores ni de configuraciones iniciales.
2) El sistema debe ser lo más autónomo posible. Se buscan sistemas Plug and Play para la conexión y desconexión de los dispositivos a la red.
3) El sistema será lo más transparente posible al usuario.

3.3 Criterio comunicacional:

La transmisión de instrucciones desde la DI a los electrodomésticos (UN) ha de ser segura tanto para asegurar un perfecto direccionamiento (llega al electrodoméstico que queremos que llegue) como para evitar intrusiones. Las características principales requeridas son:
1) El volumen de datos ha de ser extremadamente bajo, tanto en lo relativo al número de mensajes por día como en la longitud de los mismos.
2) El tiempo de transmisión está controlado por eventos tanto en la parte de la DI (ej.: una vez al día se manda lavar la ropa, tarea de una hora de duración compuesta por 20 instrucciones) como en la parte de los electrodomésticos (comunicación de distintos estados del dispositivo, posibles alertas, comprobación de cómo se han realizado las instrucciones recibidas, etc.).
3) La estructura de las instrucciones (individuales o en secuencia) será decidida caso por caso.
4) El protocolo de comunicación ha de ser rápido en términos de ancho de banda, ya que la nube manejará simultáneamente miles de DI.

4 ESPECIFICACIÓN FORMAL DE SANDS

Siguiendo el criterio descriptivo expuesto en [5], definimos una colección de usuarios como $S = \{U_i\}_{(i=1)}^{N}$, donde cada usuario está definido por un

perfil personal P_i, un conjunto de electrodomésticos $A_i = \left\{A_i^1, A_i^2, \dots, A_i^{(nA_i)}\right\}$ y un conjunto de recetas $R_i = \left\{R_i^1, R_i^2, \dots, R_i^{(nR_i)}\right\}$, quedando de la forma $U_i = \{P_i, R_i, A_i\}$. En general, el número de usuarios así como la descripción de los mismos variarán con el tiempo, pero en esta sección se mantendrá invariante para una mayor facilidad de comprensión.

La inteligencia del sistema necesitará hacer búsquedas en el espacio de recetas para obtener la solución que mejor se ajuste a la petición del usuario, o para generar nuevas soluciones en caso necesario. En un intento de formalizar estas ideas, se puede cuantificar la similitud entre dos dispositivos de la siguiente forma: $d_A: AxA^* \rightarrow \mathbb{R}$ (donde A es lo que se considera como el espacio de los electrodomésticos) tal que $d_A(A, A^*) = 0$ si $A = A^*$, $d_A(A, A^*) > 0$ si $A \neq A^*$. De igual forma se pueden aplicar otros axiomas empleados para la definición de una distancia.

Siguiendo un razonamiento similar, la semejanza entre recetas también puede ser cuantificada: $d_R: RxR^* \rightarrow \mathbb{R}$ (donde R es el espacio de recetas). Por tanto la similitud entre dos electrodomésticos es altamente parametrizable, en tanto que serán más similares cuanta mayor sea su capacidad para llevar a cabo procesos semejantes. La similitud entre perfiles $d_P: PxP^* \rightarrow \mathbb{R}$ no afecta a la inteligencia del sistema.

Continuando con la formalización, se puede decir que las recetas tienen dos componentes:

1) La tarea que debe ser realizada.
2) Los parámetros de configuración del dispositivo que vaya a realizar la tarea.

Adicionalmente se puede incluir otro parámetro que refleje el grado de satisfacción del usuario. Por tanto, la receta quedaría formalizada de la siguiente forma $R = (\tau, \Gamma, \alpha)$, donde los componentes son la tarea τ, los ajustes del dispositivo Γ y el grado de satisfacción α. La relación entre recetas y tareas no tiene por qué ser necesariamente biyectiva. Varias tareas distintas pueden ser realizadas con el mismo conjunto de instrucciones realizadas por un dispositivo, del mismo modo que un usuario puede tener varias recetas para una misma tarea, que varíen por citar algún ejemplo en grado de satisfacción, modo de realización o tiempo de ejecución.

Conexión entre usuarios:
La relación entre usuarios puede quedar reflejada de manera explicita incluyendo los usuarios estas conexiones en sus propios perfiles, del mismo modo que se hace en una red social tradicional. Se puede decir que $d_P(P_i, P_j) = 0$ si los usuarios están explícitamente conectados. Entendemos por tanto la distancia entre usuarios como una medida de lo relacionados que están dichos usuarios.

Dado que los perfiles de los usuarios pueden tener estructuras complejas, la distancia entre dos usuarios puede ser muy pequeña incluso aunque no haya una conexión explícita. Se puede hablar de una conexión implícita entendida como una distancia calculada en función de lo similares que sean los perfiles de los usuarios, sus electrodomésticos y sus recetas: $d_U(U_i, U_j) = F\left[d_P(P_i, P_j), d_A(A_i, A_j), d_R(R_i, R_j)\right]$, y se considerará que la relación implícita entre dos usuarios se volverá explícita si el valor cae por debajo de una determinada tolerancia, cuya método de obtención se establecerá en futuras fases del proyecto.

5 MÁS ACERCA DE LA ESTRUCTURA DE SANDS

Se pueden encontrar similitudes entre la estructura de SANDS y la propuesta en [7] por la presencia de un componente de indexado que maneja la identificación de usuarios y etiquetas y un interfaz de usuario, un analizador de peticiones que permiten analizar el lenguaje del usuario y extraer las solicitudes realizadas y una función de selección del mejor resultado para una búsqueda, de modo que la solución sea la que mejor encaje con los parámetros enviados en la petición.

Cuando un usuario se une al sistema, se rellena su perfil como en una red social convencional, incluidos los enlaces explícitos a amigos ya están en el sistema. El sistema también reunirá información de los electrodomésticos y del estado de la red local de la familia. Esta información puede incluir algunas recetas estándar para diversas tareas. Con la evolución del usuario en el sistema, será cada vez mayor el número de recetas y/o electrodomésticos. Al igual que en [7], el sistema puede recopilar las opiniones de los amigos del usuario para obtener información adicional de la experiencia de dicho usuario en resolver una determinada tarea, así como de su grado de satisfacción. Las opiniones pueden tener la misma estructura que la receta (tareas, configuración, satisfacción) más alguna etiqueta de identificación. El sistema se beneficia de su naturaleza focalizada de manera que el análisis semántico se simplifica y la búsqueda de la excelencia está limitada por el conocimiento almacenado en el sistema.

5.1 Proceso de búsqueda:

El principal servicio de la red social SANDS, además de proporcionar una interacción convencional entre los usuarios, será la búsqueda automática de recetas para resolver una determinada tarea. El proceso se inicia cuando el usuario introduce una consulta para una tarea específica, la red trata de identificar la tarea y resolver una serie de parámetros específicos. La

especificación de tarea puede requerir alguna realimentación por parte del usuario, mostrando tareas similares ya almacenadas por el sistema en caso que el sistema de identificación de tareas no haya encontrado ninguna coincidencia. Se puede solicitar al usuario que se seleccione el electrodoméstico en caso de que no haya sido especificado en la petición. Una vez el sistema ha identificado la tarea, se puede pedir alguna confirmación previa antes de iniciar la búsqueda. La especificación de tareas se compara con recetas ya almacenadas, buscando una solución balanceada entre aquellas que tengan la máxima similitud entre los dispositivos a los que están destinadas y aquellas que tengan un mayor factor de satisfacción del usuario. La función de clasificación combinará la satisfacción con la receta tanto del usuario como de sus amigos que la hayan probado anteriormente, en caso de haberlos. Finalmente, la receta seleccionada es enviada al usuario que la ha solicitado.

6 DESCRIPCIÓN DE LA INFRAESTRUCTURA DOMÉSTICA

Físicamente se encuentra en el mismo servidor que la red social. Comparten la misma base de datos, pero con diferentes permisos de acceso. La comunicación de la DI con los diferentes actores que componen el escenario de SANDS es de la siguiente forma:

6.1 Comunicación entre el usuario y la inteligencia del sistema (NI):

La comunicación de usuario con el sistema de inteligencia ha de hacerse mediante interfaces lo más intuitivas y amigables posibles (Figura 4a). Dicha comunicación tiene diversos fines:

1) El usuario comunica al sistema la solicitud de realización de una tarea que será realizada por uno de sus electrodomésticos.
2) El usuario realiza la configuración de su perfil, tanto en lo referente tanto a sus datos personales como a determinadas características de sus dispositivos (figura 4b).
3) El usuario aporta una realimentación al sistema en el que expresa su experiencia y grado de satisfacción con la resolución de la tarea (figura 4c)

6.2 Comunicación entre el usuario y los dispositivos:

Una de las fortalezas de SANDS es que el usuario siempre está en el centro del sistema, por tanto es necesario que pueda estar al corriente del estado de todos sus electrodomésticos. Mediante la aplicación, los dispositivos comunicarán al usuario toda la información relevante acerca de su funcionamiento (figura 4c).



a) b)

c) d)

Figura 4: Interfaz de usuario de SANDS

6.3 Comunicación entre los electrodomésticos y la DI:

La DI se comunicará con los electrodomésticos para transmitir las recetas como un conjunto de instrucciones que deben ser ejecutadas por los mismos.

Las recetas deben ser especificadas en un lenguaje formal, con una sintaxis específica. Mientras que la especificación de la tarea consistirá en algunas palabras clave, la receta es una construcción formal. El lenguaje que se utilizará para transmitir los resultados en este proyecto es XML. Este enfoque garantizará la compatibilidad a nivel interno, a través de interfaces de programación de aplicaciones (API) y a nivel protocolos de redes de intercambio de información (SOAP) y de Transferencia de Estado Representacional (REST).

7 CONCLUSIONES

El proyecto SANDS es una ambiciosa iniciativa cuyos objetivos son, por un lado establecer las bases de un sistema social con una capa de inteligencia

computacional intrínseca, y por otro desarrollar un prototipo completamente funcional. Aunque el proyecto se centra en los aparatos domésticos, su filosofía puede ser exportada a muchos otros campos. La principal diferencia con otros enfoques es que el sistema SANDS actúa de forma autónoma usando el conocimiento adquirido de los usuarios de la red social para innovar y obtener soluciones a los nuevos problemas, aumentando así la satisfacción de los mencionados usuarios.
En su estado actual, los socios del proyecto están consolidando las herramientas computacionales que permitan lograr los objetivos fijados. Definir las interfaces entre las capas del sistema (servicio de red social, la inteligencia en red, la interfaz doméstica y el control de los dispositivos), permitirá un buen desarrollo del proyecto en sus principales líneas de investigación.

Agradecimientos

Este trabajo ha sido parcialmente financiado por proyecto SANDS, Unión europea, VII Programa Marco (Gran agreement 317947), el Ministerio de Ciencia e Innovación, Proyecto de Investigación Fundamental no Orientada (Ref. DPI2011-25489) y la Junta de Castilla y León, Prog. De Apoyo a Proys. de Invest. (VA013A12-2).

Referencias

[1] Jippes, E., Achterkamp, M.C., Brand, P.L., Kiewiet, D.J., Pols, J., van Engelen, J.M., (2010), Disseminating educational innovations in health care practice: Training versus social networks. Social Science & Medicine 70(10) 1509 – 1517

[2] Kratzer, J., Leenders, R.T., van Engelen, J.M., (2009), A social network perspective on the management of product development programs. The Journal of High Technology Management Research 20(2) 169 – 181

[3] Oreszczyn, S., Lane, A., Carr, S., (2010), The role of networks of practice and webs of influencers on farmers' engagement with and learning about agricultural innovations. Journal of Rural Studies 26(4) 404 – 417

[4] Vannoy, S.A., Palvia, P., (June 2010), The social influence model of technology adoption. Commun. ACM 53(6) 149–153

[5] M. Graña, B. Apolloni, M. Fiasché, G. Galliani, C. Zizzo, G. Caridakis, G. Siolas, S. Kollias, F. Barriento, S. San Jose, (2013), Social and Smart: towards an instance of subconscious social intelligence. IPMSC.

[6] Apolloni, B., Fiasche, M., Galliani, G., Zizzo, C., Caridakis, G., Siolas, G., Kollias, S., Grana Romay, M., Barriento, F., San Jose, S., (2013), Social things - the sands instantiation. In: IoT-SoS 2013, IEEE

[7] Horowitz, D., Kamvar, S.D., (April 2012), Searching the village: models and methods for social search. Commun. ACM 55(4) 111–118

DESARROLLO DE UN CONTROLADOR BORROSO PARA EL CONTROL DE LA PROFUNDIDAD ANESTÉSICA MEDIANTE ARDUINO

Israel Riverón Miranda [a]
iriverong@gmail.com

Juan A. Méndez Pérez [a]
jamendez@ull.es

Ayoze Marrero [a]

Ana León [b]

Isabel Martín [c]

[a] Dpto. de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores, Universidad de La Laguna, Avda. Astrofísico Francisco Sánchez s/n, 38206, La Laguna (Tenerife)

[b] Hospital Universitario de Canarias

[c] Dpto. de Física Fundamental y Experimental, Electrónica y Sistemas, Universidad de La Laguna

Resumen

Este artículo detalla el desarrollo de un motor de lógica borrosa para su implementación sobre las tarjetas de la plataforma de hardware libre ARDUINO orientado al control de la anestesia. El trabajo parte del interés de contar con una herramienta de fácil uso que permita una rápida reprogramación del controlador para su aplicación sobre procesos reales. Su utilización principal se enmarca en el ámbito del desarrollo de prototipos, investigación y docencia. El uso extendido de los microcontroladores ATMEL® (en los que se basa la tarjeta ARDUINO) y la existencia de compiladores de distribución libre de C para AVR®, amplía la perspectiva para el uso de esta herramienta sobre la aplicación final. Por último, se realiza una comparativa entre la simulación del controlador borroso para el control de la profundidad anestésica ejecutado sobre MATLAB® y la respuesta equivalente obtenida mediante el motor de lógica borrosa en ARDUINO.

Palabras Clave: Control Borroso, Microcontrolador, ARDUINO, Anestesia, Propofol, Índice Biespectral.

1 INTRODUCCIÓN

En la última década, se ha incrementado notablemente el uso de técnicas de Control Inteligente, motivado entre otras razones por las mejoras en la robustez que se obtienen con respecto a los métodos clásicos de control. Además, en el caso del control borroso, no es necesario contar con un modelo matemático del proceso y posibilita el diseño de un controlador para sistemas altamente complejos y/o con fuerte presencia de no-linealidades.

Sin embargo, se hace imprescindible contar con una estrategia de control en forma de reglas de inferencia, obtenidas a partir del conocimiento de un experto. Precisamente porque se ha de obtener un conocimiento empírico del sistema (que en muchos casos no se posee completamente o es impreciso), es habitual realizar un proceso de rediseño continuo del controlador borroso, a fin de mejorar su eficiencia. En este sentido, es de interés poder realizar la sintonización del controlador a partir de la respuesta obtenida en el proceso real.

El presente trabajo aborda el diseño de un motor de lógica borrosa para las tarjetas de desarrollo ARDUINO que permita una fácil programación de la base de reglas y definición de los conjuntos borrosos, como alternativa al uso de ordenadores personales en situaciones donde su uso no es posible o apropiado. Una solución similar se presenta en [2] que implementa un controlador borroso sobre una placa computadora Friendly ARM haciendo uso de MATLAB, y empleando una tarjeta ARDUINO para la adquisición de datos.

La motivación de este proyecto surge como solución frente a la necesidad de simplificar el conjunto de *har-*

dware utilizado hasta el momento en el quirófano para realizar el control de la profundidad anestésica en pacientes (software en MATLAB ejecutado desde un ordenador personal). Se dispone en su lugar de una tarjeta electrónica con software dedicado, que a su vez permite establecer comunicación con los equipos médicos involucrados.

2 APROXIMACIÓN AL CONTROL BORROSO

Un controlador borroso se encuentra constituido por cuatro elementos fundamentales: interfaz de borrosificación, base de conocimiento, mecanismo de inferencia e interfaz de desborrosificación (ver figura 1). En una primera etapa, la interfaz de borrosificación recibe el valor de un conjunto de *n* entradas comprendidas en un rango determinado (conocido como universo de discurso, U_i), que por lo general es distinto para cada una de ellas. Cada entrada *i* se trata de una variable lingüística x_i, que puede tomar diversos valores distintos denotados por el índice *j*, conocidos como valores lingüísticos o etiquetas, A_i^j. Estos valores lingüísticos se definen en U_i mediante la declaración de una función de pertenencia $\mu_{A_i^j}(x_i)$, donde A_i^j es un conjunto borroso. De esta forma, dado un valor de entrada x_i, la función de pertenencia determina su nivel de correspondencia con el valor lingüístico (grado de pertenencia) según una escala que varía desde 0 hasta 1 (mínima y máxima correlación con la etiqueta). Los grados de pertenencia obtenidos son los valores borrosos utilizados por el mecanismo de inferencia. El motor de lógica borrosa desarrollado en este trabajo utiliza un método de borrosificación de tipo *singleton*. Este conjunto difuso se caracteriza por tener grado de pertenencia unidad en el valor que coincide con el de la entrada, y cero en el resto del universo de discurso (ver figura 2).

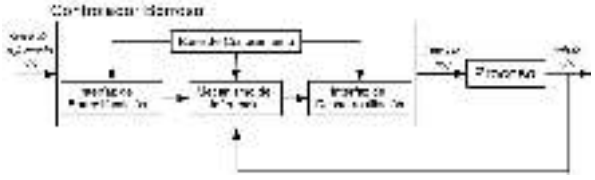


Figura 1: Esquema de control borroso

La base de conocimiento está formada por la base de reglas y la base de datos. La base de reglas constituye una descripción del conocimiento experto que se posee para realizar el control del sistema, expresado en forma de sentencias del tipo *if-else (modus ponens)*. De esta manera, dado un conjunto de *n* entradas, podemos enunciar reglas de la siguiente forma:

$$IF\ A_1^j\ and\ A_2^j\ and\ \ldots\ and\ A_n^m,\ THEN\ B_p^q \quad (1)$$

donde A_i^j corresponde al conjunto borroso de la etiqueta *j* perteneciente a la variable lingüística de la entrada *i*, y B_p^q al conjunto borroso de la salida *q* en el consecuente de la regla *s*. Por otro lado, la base de datos comprende la definición de los conjuntos borrosos de las variables lingüísticas a la entrada y salida del sistema.

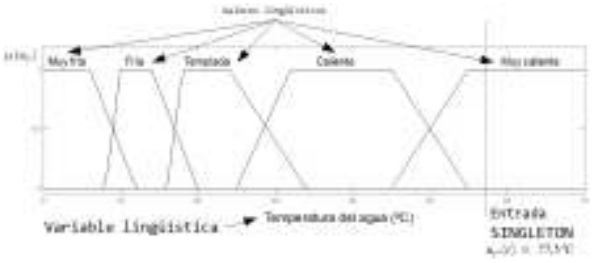


Figura 2: Ejemplo de funciones de pertenencia para una variable lingüística de entrada

Por su parte, el mecanismo de inferencia borrosa realiza un mapeo entre los conjuntos borrosos de entrada A_i^j y los de salida B_p^q, interpretando cada regla como una implicación difusa, de tal forma que se obtiene un nuevo conjunto borroso ΔB_p^q como resultado de tal implicación. El procedimiento se inicia determinando el grado de pertenencia del antecedente de la regla que posee mayor relevancia sobre la salida (fuerza de disparo). Si la regla se compone de múltiples antecedentes, se puede tomar el valor mínimo *(AND)* o máximo *(OR)* de entre todos los grados de pertenencia que forman parte del antecedente (ver ecuaciones 2 y 3).

$$\mu_{A_1^j \text{ and } A_2^j \text{ and } \ldots \text{ and } A_n^m}(x_1, x_2, \ldots, x_n) = \min[\mu_{A_1^j}(x_1), \mu_{A_2^j}(x_2), \ldots, \mu_{A_n^m}(x_n)] \quad (2)$$

$$\mu_{A_1^j \text{ or } A_2^j \text{ or } \ldots \text{ or } A_n^m}(x_1, x_2, \ldots, x_n) = \max[\mu_{A_1^j}(x_1), \mu_{A_2^j}(x_2), \ldots, \mu_{A_n^m}(x_n)] \quad (3)$$

A continuación, se debe establecer un operador que realice la implicación entre el antecedente y el consecuente de la regla. Los operadores de implicación más habituales son el operador *mínimo* y el *producto*. En el motor de lógica borrosa diseñado se ha optado por utilizar el operador *mínimo* (ecuación 4) del que resultan los conjuntos recortados ΔB_p^q de la figura 3.

$$\mu_{\Delta B_p^q}(y_p) = \min[\mu_{A_1^j}(x_1), \mu_{A_2^j}(x_2), \ldots, \mu_{A_n^m}(x_n), \mu_{B_p^q}(y_p)] \quad (4)$$

El conjunto borroso final se obtiene mediante la agregación de la totalidad de los conjuntos recortados ΔB_p^q denotada por el operador $\oplus$ en la siguiente expresión:

$$\mu_{B_p}(y_p) = \mu_{\Delta B_p^1}(y_p) \oplus \mu_{\Delta B_p^2}(y_p) \oplus \ldots \oplus \mu_{\Delta B_p^q}(y_p) \quad (5)$$

Por último, la interfaz de desborrosificación se encarga de calcular un valor numérico nítido y_0^* a la salida del controlador, haciendo uso de un método de desborrosificación. Existen diversos métodos para la desborrosificación basados en la geometría del conjunto borroso final, cada uno de los cuales suele producir distintos resultados. En este trabajo se utiliza el cálculo del centroide del área como técnica de desborrosificación, aunque el código desarrollado es flexible en cuanto a la futura incorporación de otros métodos. La ecuación 6 determina la coordenada horizontal del centroide del área,

triangulares (figura 3), que contiene dos constructores diferentes que permiten declarar un objeto conjunto borroso de cualquiera de los dos tipos. Los constructores de la clase requieren la definición de una función de pertenencia (dada por las coordenadas horizontales de los vértices del conjunto), además de los límites del universo de discurso. Un tercer constructor se utiliza para la obtención de los conjuntos borrosos recortados $\hat{B}_n^i$ a partir del valor de la fuerza de disparo $\mu_{B_i'}$ y las coordenadas horizontales del conjunto borroso original. El método *Grade* devuelve el grado de pertenencia de un valor de entrada sobre el conjunto.

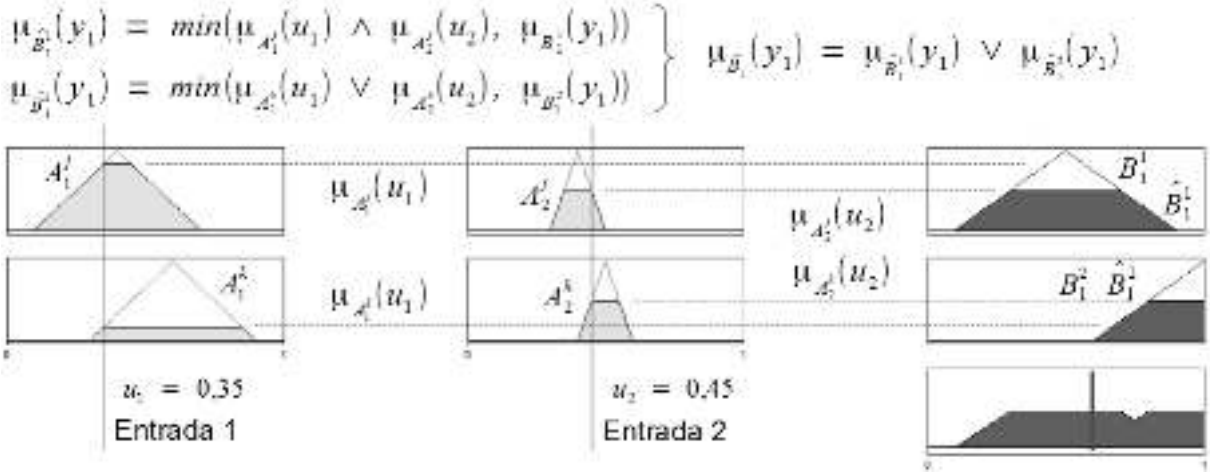


Figura 3: Ejemplo de mecanismo de inferencia

$$y_0^* = \frac{\sum_{i=1}^{R} b_i^1 \int_{y_n} \mu_{B_i'}(y_n)\, dy_n}{\sum_{i=1}^{R} \int_{y_n} \mu_{B_i'}(y_n)\, dy_n} \quad (6)$$

donde R es el número total de reglas, b_i^1 es la componente horizontal del centro del área del conjunto borroso $\hat{B}_n^i$, y la integral $\int_{y_n} \mu_{B_i'}(y_n)\, dy_n$ corresponde al área bajo la función de pertenencia $\mu_{B_i'}$ delimitada por el universo de discurso de la salida y_n.

3 IMPLEMENTACIÓN DEL MOTOR DE LÓGICA BORROSA

El motor de lógica borrosa que se presenta en este trabajo tiene como objetivo dotar a la plataforma ARDUINO de una herramienta que permita de forma práctica, la implementación de un controlador borroso. Además, la construcción modular de este motor hace posible una fácil extensión de las funcionalidades de la aplicación.

En primer lugar, se describe una clase para la definición de conjuntos borrosos trapezoidales (figura 2) y

```
TrapezoidSet(float xp0, float xp1, float
xp2, float plbound, float prbound);

TrapezoidSet(float xp0, float xp1, float
xp2, float xp3, float plbound, float pr-
bound);

TrapezoidSet(float mu, float Xp[4],
float plbound, float prbound);

float Grade(float input);
```

Es importante considerar que las coordenadas horizontales se deben declarar de menor a mayor valor. Por otro lado, no está permitida la definición de conjuntos borrosos con rectas de pendiente infinita. Por todo ello, se debe cumplir la ecuación 7, o de lo contrario se genera un conjunto de área nula en el que todas sus coordenadas están situadas sobre el origen.

$$x_{p0} < x_{p1} < x_{p2} < x_{p3} \quad (7)$$

De forma similar, se realiza una clase para la declaración de conjuntos borrosos con una función de pertenencia de tipo gausiano (figura 4). Observar que la función de pertenencia se describe en tres tramos según (8). Los tramos a los extremos de la función corresponden a una distribución de densidad de

probabilidad $N(c,\sigma)$, cuyos parámetros son distintos a cada lado. El constructor de esta clase requiere de un valor medio y desviación típica para cada extremo. En la figura 5 se representa un conjunto borroso gausiano modificado a consecuencia de un corte (9). Al igual que en el caso anterior, se dispone de un constructor para la definición de conjuntos recortados, además de un método para la obtención del grado de pertenencia.



Figura 4: Conjunto borroso gausiano

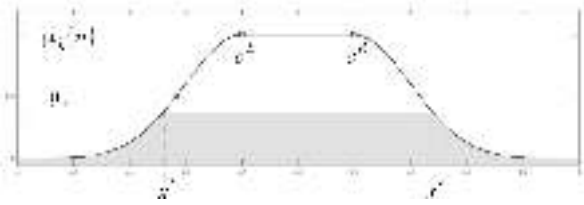


Figura 5: Conjunto borroso gausiano recortado

$$\mu_{A_{ij}}(x) = \begin{cases} \exp\left[-\frac{1}{2}\left(\frac{x-c^{i}}{\sigma^{i}}\right)^{2}\right] & \text{si } x \le c^{i} \\ 1 & \text{si } c^{i} < x < c^{d} \\ \exp\left[-\frac{1}{2}\left(\frac{x-c^{d}}{\sigma^{d}}\right)^{2}\right] & \text{si } x > c^{d} \end{cases} \quad (8)$$

$$\mu_{A_{ij}}(x) = \begin{cases} \exp\left[-\frac{1}{2}\left(\frac{x-c^{i}}{\sigma^{i}}\right)^{2}\right] & \text{si } x \le c^{i} \\ \mu_{x} & \text{si } c^{i} < x < c^{d} \\ \exp\left[-\frac{1}{2}\left(\frac{x-c^{d}}{\sigma^{d}}\right)^{2}\right] & \text{si } x > c^{d} \end{cases} \quad (9)$$

```
GaussianSet(float sigma_l, float med_l, float
plbound, float prbound);

GaussianSet(float sigma_l, float med_l, float
sigma_r, float med_r, float plbound, float
prbound);

GaussianSet(float mu, float sigmap[2], float
medp[2], float plbound, float prbound);

float Grade(float input);
```

De esta forma, el cálculo de los grados de pertenencia $\mu_{A_{ij}}(x_i)$ del antecedente de la regla se realiza a partir de la ejecución del método *Grade* sobre cada entrada. A continuación, se toma el valor máximo o mínimo (operadores *OR* o *AND*) de entre todos los $\mu_{A_{ij}}(x_i)$ calculados. El valor escogido como fuerza de disparo (μ_{R_j}) se aplica sobre el conjunto borroso B_j' del consecuente, dando como resultado el conjunto borroso recortado, $\hat{B}_j'$. La definición de $\hat{B}_j'$ se realiza utilizando el tercer constructor de las clases, que emplea el operador *mínimo* en la implicación.

```
// Valores lingüísticos para la entrada
0
TrapezoidSet a1j = TrapezoidSet(0.2,
0.45, 0.7, li0, ri0);
TrapezoidSet a1k = TrapezoidSet(0.3,
0.6, 0.8, li0, ri0);

// Valores lingüísticos para la entrada
1
TrapezoidSet a2j = TrapezoidSet(0.4,
0.5, 0.6, li1, ri1);
TrapezoidSet a2k = TrapezoidSet(0.5,
0.6, 0.7, li1, ri1);

// Valores lingüísticos para la salida
TrapezoidSet b1 = TrapezoidSet(0.2, 0.5,
0.7, lo, ro);
TrapezoidSet b2 = TrapezoidSet(0.3, 0.6,
0.9, lo, ro);

// Antecedentes
mu[0] = min(a1j.Grade(in[0]),
a2j.Grade(in[1]));
mu[1] = min(a1k.Grade(in[0]),
a2k.Grade(in[1]));

// Consecuentes
b_hat[0] = TrapezoidSet(mu[0], b1.X, lo,
ro);
b_hat[1] = Trapezoidset(mu[2], b2.X, lo,
ro);
```

Finalmente, los conjuntos $\hat{B}_j'$ se pasan a la función *centroid* como un vector. Esta función devuelve el valor de la coordenada horizontal del centroide del área resultante de la unión de los conjuntos. Es necesario especificar el parámetro λ, que representa el porcentaje en tanto por uno de la base del rectángulo de integración, con respecto al universo de discurso a la salida. La integración del área bajo la curva se realiza mediante el cálculo del área de los rectángulos, según se muestra en la figura 6. La utilización de este método de integración del área resulta en la siguiente ecuación para el cálculo de la componente horizontal del centroide (salida del controlador borroso):

$$y_0 = \frac{\sum_{i=1}^{n} x_i \cdot \mu_{B'}(x_i)}{\sum_{i=1}^{n} \mu_{B'}(x_i)} \quad (10)$$

donde n es el número total de intervalos de integración, x_i es la coordenada horizontal del centro de la base del rectángulo en el intervalo i, y $\mu_{B'}(x_i)$ es la altura máxima de entre todos los rectángulos de cada una de los conjuntos que se intersectan sobre la vertical de x_i.

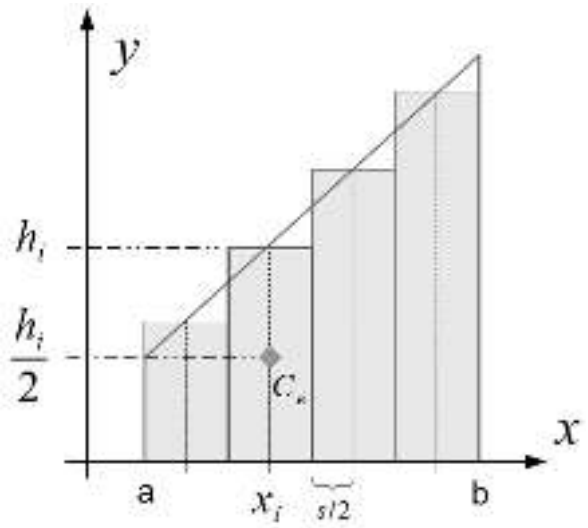


Figura 6: Cálculo del área bajo la curva

4 APLICACIÓN EN EL CONTROL DE LA PROFUNDIDAD ANESTÉSICA

Esta sección tiene como objetivo realizar una comparativa entre el funcionamiento del controlador de lógica borrosa ejecutado sobre ARDUINO y los resultados obtenidos para el mismo controlador con la herramienta *'Fuzzy Logic Toolbox'* de MATLAB. La realización de estas pruebas es necesaria a fin de asegurar la adecuación del controlador implementado, de tal forma que sea posible comprobar el grado de similitud que existe (en condiciones ideales) entre la respuesta del controlador diseñado y el que va a ser aplicado sobre el proceso real [4] [6]. Estas comprobaciones son de especial importancia para el caso del control de la anestesia, ya que es necesario constatar antes de su utilización en el quirófano, que el comportamiento de la tarjeta corresponde al que se puede observar en MATLAB durante la fase de sintonización del controlador.

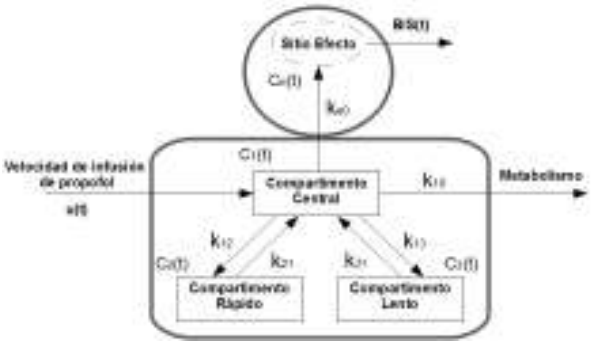


Figura 7: Modelo compartimental de Schnider

El problema del control de la profundidad anestésica puede formularse como el problema de llevar al paciente a un estado de hipnosis determinado, rechazando las perturbaciones que actúan sobre el mismo durante la intervención (estímulos quirúrgicos y pérdidas de sangre) [3]. La hipnosis es la inducción de un paciente a un estado de pérdida de la consciencia mediante el uso de fármacos. El nivel de consciencia que presenta un paciente sometido a la acción de un hipnótico, se trata de la variable de salida en el problema de control. Dado que la variable que se pretende regular no es medible directamente, se hace necesario el análisis de otros parámetros del paciente que sea posible observar y que permitan conocer su nivel de consciencia [1].

Este trabajo utiliza el índice biespectral *(BIS)* obtenido a partir del electroencefalograma del paciente como medida de su nivel de consciencia (variable controlada), y la infusión intravenosa de propofol como agente hipnótico (variable manipulada) [7]. La base de conocimiento utilizada corresponde a un controlador borroso que está siendo objeto de estudio por parte del Grupo de Control de Anestesia de la Universidad de La Laguna. Por otro lado, se realizó la simulación del sistema tomando la respuesta ante el suministro de propofol de un paciente ficticio según el modelo compartimental de Schnider [8]. Las ecuaciones siguientes representan el comportamiento farmacocinético (PK) del modelo de Schnider descrito según sus variables de estado:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (11)$$

$$x(t) = \begin{pmatrix} C_1(t) \\ C_2(t) \\ C_3(t) \\ C_e(t) \end{pmatrix} \quad (12)$$

$$x(0) = \begin{pmatrix} C_1(0) \\ C_2(0) \\ C_3(0) \\ C_e(0) \end{pmatrix} \quad (13)$$

$$A = \begin{pmatrix} -(k_{10}+k_{12}-k_{21}k_{12}) & \frac{V_1}{V_2}\cdot k_{12} & \frac{V_1}{V_3}\cdot k_{13} & 0 \\ \frac{V_2}{V_1}\cdot k_{21} & -k_{21} & 0 & 0 \\ \frac{V_3}{V_1}\cdot k_{31} & 0 & -k_{31} & 0 \\ k_{e0} & 0 & 0 & -k_{e0} \end{pmatrix} \quad (14)$$

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \quad (15)$$

$$C = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (16)$$

$$D = 0 \quad (17)$$

donde $C_e(t)$ es la concentración de propofol en el compartimento *i* medida en mg/l, V_i es el volumen en litros del compartimento *i*, y el parámetro k_{ij} corresponde a la constante de difusión del fármaco desde el compartimento *i* hasta el *j*.

pecto al valor de consigna ($BIS = 50$) y la velocidad de variación del error. La salida del controlador corresponde una variación de la velocidad de infusión $\delta u(t)$, que se añade al valor obtenido por el controlador en el instante anterior. Este resultado se suma al valor de la velocidad de infusión base (u_{base}), que posteriormente se aplica sobre un limitador que evita la aparición de velocidades de infusión negativas, o por encima de un determinado valor de seguridad. Finalmente, se realiza un cambio de unidades de ml/h a $mg/kg/min$, necesario para poder operar sobre el modelo.

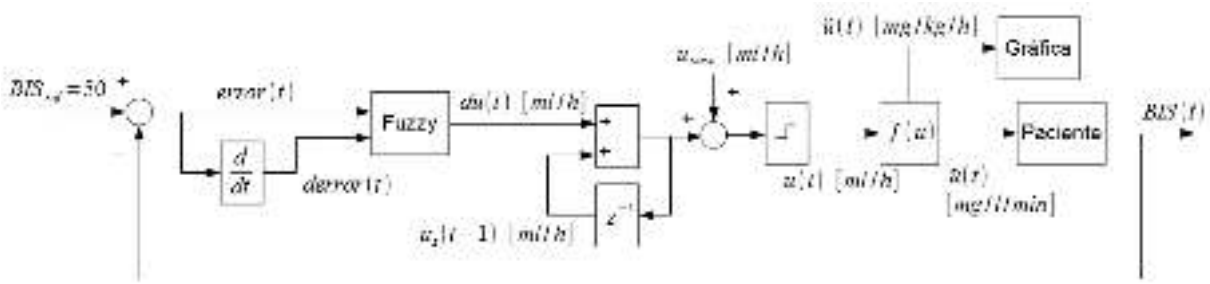


Figura 8: Esquema del control borroso para la profundidad hipnótica

El comportamiento farmacodinámico (PD) del propofol (es decir, el efecto que produce sobre el valor *BIS* del paciente) es función de su concentración sobre el sitio efecto $C_e(t)$. La función que relaciona la concentración del fármaco en $C_e(t)$ con el índice *BIS* se define mediante un modelo PK-PD [5]. Este tipo de modelos, ampliamente utilizados en farmacología, tratan de determinar la relación *concentración-efecto* de las drogas. En este caso, el efecto farmacodinámico se traduce en variaciones del índice *BIS* según la siguiente ecuación:

$$\Delta BIS(t) = \frac{\Delta BIS_{max} \cdot C_e(t)^{\gamma}}{C_e(t)^{\gamma} + EC_{50}^{\gamma}} \quad (18)$$

donde γ es un exponente que determina el grado de no linealidad del modelo que representa la sensibilidad del paciente ante pequeñas variaciones de la concentración, EC_{50} es la concentración efectica al 50% (que produce un efecto igual al 50% de su efecto máximo), $\Delta BIS(t)$ es la variación del índice *BIS* en el instante *t* con respecto a su valor nominal BIS_0, y ΔBIS_{max} es la diferencia entre el nivel de máxima depresión hipnótica (BIS_{max}) y el valor nominal en estado de vigilia (BIS_0).

$$\Delta BIS(t) = BIS(t) - BIS_0 \quad (19)$$

$$\Delta BIS_{max} = BIS_{max} - BIS_0 \quad (20)$$

La figura 8 representa el esquema utilizado para el control de la profundidad anestésica. Las entradas al controlador borroso son el error de la salida con res-

La figura 9 muestra la evolución de la velocidad de infusión y el índice biespectral BIS de un paciente ficticio obtenidos mediante simulación en MATLAB. El mismo modelo del paciente se utilizó para el cálculo de las respuestas de simulación del controlador de la tarjeta ARDUINO de las figuras 10 y 11. Estas pruebas fueron realizadas con una tarjeta ARDUINO ATMEGA2560 que incorpora un microcontrolador ATMEL de 8 bits del mismo nombre, de hasta 16 MIPS.

El controlador borroso que se utiliza en la aplicación se encarga de la corrección de las desviaciones del BIS que se producen con respecto al punto de consigna. Inicialmente, se ha de inducir al paciente hasta un estado hipnótico profundo. Esto se realiza mediante la aplicación de una alta dosis de propofol conocida como bolo inicial. Trascurrido un tiempo de espera en torno a los dos minutos tras la aplicación de este bolo, la acción del controlador borroso comienza a actuar siempre que el valor BIS sea superior a 30. De esta forma, en las figuras 9, 10 y 11 podemos observar que el controlador borroso no se inicia hasta que han pasado alrededor de cuatro minutos desde el inicio. Al margen de la acción del controlador borroso, existe un conjunto de mecanismos de seguridad que se sitúan sobre un nivel de prioridad mayor, orientados a evitar tanto los despertares intraoperatorios (aplicación de microbolos) como la sobredosificación (detención de la infusión).

La correspondencia entre la respuesta de la simulación de la tarjeta y la obtenida mediante MATLAB viene

determinada en gran medida por la precisión de las operaciones realizadas en ARDUINO, y en general será distinta según las características de cada controlador definido. No obstante, la variación del tamaño de la base del rectángulo de integración puede determinar una mayor o menor similitud entre ambas simulaciones. La figura 12 muestra la diferencia en tanto por ciento de la respuesta *BIS* obtenida mediante la tarjeta, con respecto a la simulación en MATLAB, según distintos valores del parámetro λ (tanto por uno de la base del rectángulo de integración, con respecto al universo de discurso a la salida). Debido a la precisión de las operaciones con números *float* en ARDUINO, no es aconsejable utilizar tamaños de la base del rectángulo inferior al $1 \cdot 10^{-3}$ *(tanto por uno)*, al tratarse de valores cercanos a cero que pueden resultar en áreas nulas para los rectángulos. De cualquier modo, se insiste en la necesidad de realizar pruebas comparativas como las que se presenta en este apartado, a fin de detectar errores de funcionamiento en el controlador borroso.

Tabla 1: Correspondencia clínica del índice BIS.

BIS	ESTADO
100-80	**Despierto** Responde a comandos de voz normal
80-60	Responde a comandos de voz fuerte o suave con agitación y/o pinchazos
60-40	**Anestesia General** Baja probabilidad de recuerdos explícitos Ausencia de respuesta a estímulos verbales.
40-20	**Estado hipnótico profundo**
20-0	**Brote supresión**

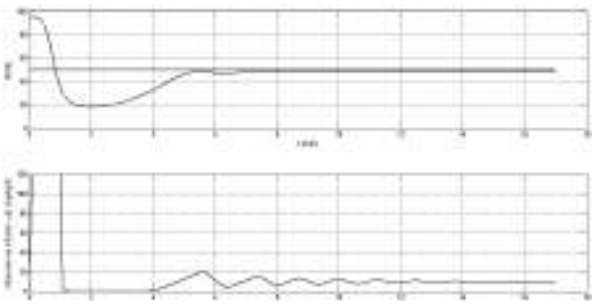


Figura 9: Simulación de la respuesta del paciente en MATLAB

Por otra parte, el controlador debe realizar una actualización del valor de infusión cada 5 segundos. En este sentido, es necesario considerar que la velocidad de cómputo para la obtención de la salida del controlador disminuye conforme la base del rectángulo de integración se hace menor. Aquellos sistemas que presentan una dinámica rápida en la que el tiempo de actuación del controlador es crítico, deben poseer mecanismos para el tratamiento de demoras implementados como eventos e interrupciones sobre el microcontrolador, que eviten la aparición de problemas en el control durante su funcionamiento.

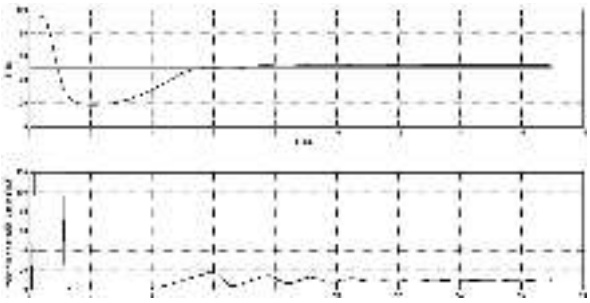


Figura 10: Simulación de la respuesta del paciente con el controlador ARDUINO, $\lambda = 1 \cdot 10^{-1}$ *(tanto por uno)*

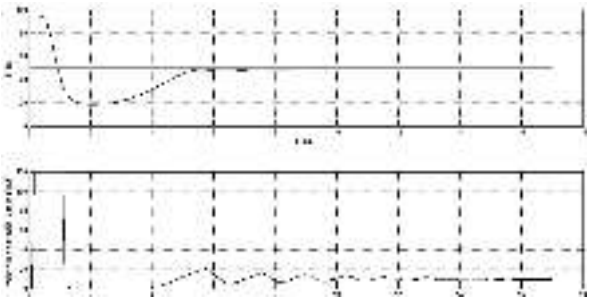


Figura 11: Simulación de la respuesta del paciente con el controlador ARDUINO, $\lambda = 1 \cdot 10^{-2}$ *(tanto por uno)*

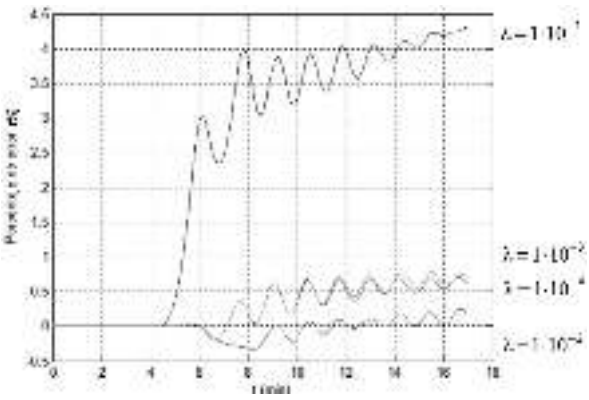


Figura 12: Diferencia en % de la respuesta BIS obtenida mediante el controlador de la tarjeta ARDUINO con respecto a la simulación en MATLAB, para diferentes valores de λ *(tanto por uno)*

5 CONCLUSIÓN

En este trabajo se presenta la implementación de un motor de inferencia borrosa sobre un microncontrolador de bajo coste para la regulación de la profundidad anestésica en pacientes. La plataforma en la que se ha realizado la implementación es la plataforma de *hardware* libre ARDUINO. La herramienta desarrollada permite la implementación de forma sencilla de sistemas de inferencia borrosa y su aplicación para el control de procesos. El trabajo presenta las diferentes estapas de desarrollo de esta plataforma que usa una inferencia de tipo Mamdani y su validación a través de la herramienta MATLAB. Como aplicación, se ha desarrollado un controlador para mantener el nivel de inconsciencia de un paciente en el valor adecuado para la intervención. Las simulaciones llevadas a cabo atestiguan la eficiencia del método de control planteado y proyectan este trabajo como una herramienta de gran interés en el ámbito médico, debido a sus características de sistema compacto y de bajo costo.

ENGLISH SUMMARY

Development of a Fuzzy Logic Controller for Automatic Control of Depth of Anesthesia.

ABSTRACT

This article details the development of a fuzzy logic inference engine for deployment on electronic prototyping over ARDUINO's open-source platform boards for automatic control of depth of anesthesia. This device is realized due to the interest to be provided with a user-friendly tool which allows quick programming of the controller for real processes applications. Its principal use is framed in the area of development of prototypes, research and teaching. The widespread use of ATMEL's® microcontrollers (which are used on the ARDUINO's boards) and the existence of free distribution of C compilers for AVR®, extends the perspective for the use of this tool on the final application. Finally, a comparative is realized between the simulation of the fuzzy logic inference engine for the control of depth of anesthesia executed on MATLAB® and the equivalent response obtained by mean of the inference engine in ARDUINO.

Keywords: Fuzzy control, Microcontroller, ARDUINO, Anesthesia, Propofol, Bispectral Index.

Agradecimientos

Este trabajo ha sido financiado a través del proyecto de investigación del Ministerio de Economía y Competitividad DPI2010-18278.

Referencias

[1] Eshghi, N., Aliyar, M., Teshnehlab, M., 2009. Anesthesia control based on intelligent controllers. 3rd International Conference on Bioinformatics and Biomedical Engineering iCBBE 2009.

[2] Krivic, S., Hujdur, M., Mrzic, A., Konjicija, S., 2012. Design and implementation of fuzzy controller on embedded computer for water level control. 35th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, 1747–1751.

[3] Méndez, J. A., Torres, S., Reboso, J. A., Reboso, H., 2011. Estrategias de control en la práctica de anestesia. Revista Iberoamericana de Automática e Informática Industrial 8, 241–249.

[4] Moore, B., L.D., P., Doufas, A., 2009. Fuzzy control for closed-loop, patientspecific hypnosis in intraoperative patients: a simulation study. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society 2009, 083–3086.

[5] Nunes, C., Mahfouf, M., Linkens, D., 2006. Fuzzy modelling for controlled anaesthesia in hospital operating theatres. Control Eng.Pract 14, 563–572.

[6] Roubos, J., Mollov, S., Babuska, S., Verbruggen, H., 1999. Fuzzy model-based predictive control using takagi-sugeno models. International Journal of Approximate Reasoninga 22, 3–30.

[7] Sahinovic, M., Absalom, A., Struys, M., 2010. Administration and monitoring of intravenous anesthetics. Curr.Opin.Anaesthesiola 23, 734–740.

[8] Schnider, T., Minto, C., Gambus, P., Anderson, C., Goodale, D., Shafer, S., Youngs, E., 1998. The influence of method of administration and covariates on the pharmacokinetics of propofol in adult volunteer. Anesthesiology 88, 1170–1182.

Modelado borroso de una Pila de Combustible Nexa™ 1.2 kW Ballard Power System

Antonio Javier Barragán Piña, Francisca Segura Manzano
José Manuel Andújar Márquez, Miguel Ángel Martínez Bohórquez
{antonio.barragan, francisca.segura, bohorquez, andujar}@diesia.uhu.es

Escuela Técnica Superior de Ingeniería, Universidad de Huelva

Resumen

El principal objetivo de las tecnologías de conversión de energía es la producción de energía no contaminante. En este sentido, los sistemas basados en pilas de combustible son una alternativa prometedora por su gran eficiencia, alta fiabilidad y bajo impacto ambiental, especialmente las pilas de combustible de membrana de intercambio de protones (PEM) [5]. El objetivo de este trabajo es realizar el modelado de una pila de combustible real mientras ésta se encuentra en funcionamiento, concretamente de la pila Nexa™ 1.2 kW de Ballard.

Palabras clave: Filtro de Kalman, Modelado borroso, Pila de combustible

1. Introducción

El objetivo principal de las nuevas tecnologías de conversión de energía es la producción de energía no contaminante. En este sentido, los sistemas basados en células de combustible son una fuente de energía alternativa prometedora gracias a su alta eficiencia, bajo impacto ambiental, y alta fiabilidad válida para aplicaciones estacionarias, de automoción y espaciales [5]. En comparación con otras tecnologías, como la eólica o la generación fotovoltaica, las pilas de combustible puede situarse en cualquier lugar, sin restricciones geográficas o ambientales. En concreto, las células de combustible PEM son consideradas una de las mejores alternativas como fuentes de generación distribuida, ya que debido a que utilizan un polímero sólido como electrolito, reduciendo problemas de transporte y de seguridad y son capaces de funcionar a bajas temperatura (50 °C–100 °C). Además, son muy respetuosas con del medio ambiente, ya que el único subproducto resultante de la generación eléctrica es agua, lo que permite su uso en una amplia gama de aplicaciones. Por ejemplo, en aplicaciones estacionarias, las células de combustible PEM pueden estar conectados a la red eléctrica [22], instalada como generadores independientes [20], u operar en los vertederos y plantas de tratamiento de aguas residuales [23]. En las aplicaciones de transporte, hay un gran campo de investigación debido a los vehículos eléctricos (ciclomotores, coches, autobuses o camiones), debida a la creciente escasez de combustibles fósiles [5, 13]. Otra aplicación interesante de las células de combustible son las aplicaciones móviles, donde pilas de combustible pueden cambiar el mundo ubicuo proporcionando baterías de larga duración a teléfonos móviles, ordenadores portátiles, etc. [25].

Es importante ser capaz de predecir el comportamiento de las pilas de combustible para un uso eficiente de éstas, por lo tanto, la obtención de un modelo preciso es una tarea fundamental antes de diseñar una estrategia de control. Sin embargo, no siempre tenemos suficiente información para obtener un modelo matemático aceptable, o es necesario obtener un modelo más realista y preciso, por lo que se debe recurrir a técnicas de modelado basado en datos de entrada/salida.

El modelado de una planta mediante datos entrada-salida permite obtener modelos más realistas de la planta en estudio, por lo tanto, dentro de su ámbito de aplicación, estos modelos son más precisos que los modelos teórico de la misma. El modelado borroso, especialmente modelado basado en sistemas Takagi-Sugeno (TS), permite obtener modelos de alta precisión con un número pequeño de reglas [30], por lo que resultan una herramienta ideal para modelar una planta no lineal como es una pila de combustible. Hasta la actualidad han sido propuestos muchos algoritmos de modelado borroso basado en datos de entrada-salida [7, 16, 17], pero en este artículo, para realizar el modelo de la pila Nexa™ de la figura, se va a emplear un sistema borroso de tipo TS, y la metodología basada en el filtro de Kalman extendido (*EKF*) previamente desarrollada [9], la cual permite obtener modelos en línea con gran precisión, incluso en presencia de ruido.

Los elementos auxiliares que acompañan a una pila de combustible son elementos de conmutación, ruidosos por naturaleza, por lo que resulta especialmente interesante las capacidades del filtro de Kalman para trabajar en ambientes ruidosos.

El filtro de Kalman se ha utilizado anteriormente junto con la lógica borrosa en diversas aplicacio-



Figura 1: Pila Nexa™ de 1.2 kW de Ballard.

nes [18, 21, 24]. En 2002, Simon introdujo el uso del filtro de Kalman para el ajuste de los parámetros de un modelo difuso [28], suponiendo que los antecedentes eran funciones de pertenencia de tipo triangular, haciendo uso de su centro de gravedad para llevar a cabo el proceso de adaptación paramétrica. Posteriormente, se han realizado otras propuestas [14], generalizándose finalmente en [8, 9].

Este artículo se organiza en las siguientes secciones: en la sección 2 se muestra el funcionamiento de una pila de combustible de tipo PEM. A continuación, en la sección 3, se presenta el problema de modelado borroso y la notación que se empleará en este aspecto. En la sección 4 se presenta la aplicación del filtro de Kalman extendido al modelado borroso, el cual se aplica en la sección 5 para modelar la pila de combustible Nexa™ de 1.2 kW de Ballard. Finalmente se presentan algunas conclusiones.

2. Pila de combustible tipo PEM

Pilas de combustible PEM, o células de combustible de membrana de intercambio de protones, ofrecen una alta densidad de energía y una serie de ventajas, como su bajo peso y el volumen, en comparación con otras células de combustible. Las pilas de combustible PEM operan a baja temperatura (50ºC–100 ºC), lo que les permite iniciarse con más rapidez (requieren menos tiempo de calentamiento), y resulta en un menor desgaste en los componentes del sistema y una mejor durabilidad.

La estructura básica de una pila de una sola célula, o unidad de PEM, sobre la base de la cual se construye una pila de combustible, consiste en una capa de electrolito en contacto con un ánodo y un cátodo en ambos lados, ver figura 2. Una célula de combustible PEM produce energía electroquímica

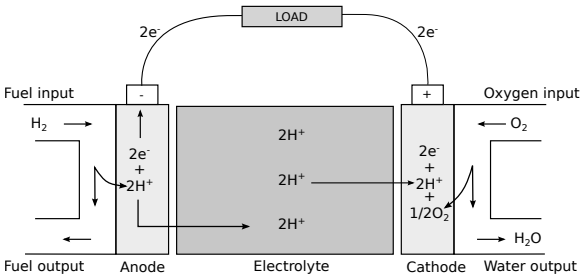


Figura 2: Diagrama de una pila PEM.

cuando un gas rico en hidrógeno pasa a través del ánodo y un gas rico en oxígeno (o aire) pasa a través del cátodo, con un electrolito entre el ánodo y el cátodo, que permite el intercambio de carga eléctrica (iones). La disociación de las moléculas de hidrógeno produce el flujo de iones a través del electrolito y una corriente eléctrica a través de un circuito externo. El único residuo generado es agua pura. En condiciones de funcionamiento normales, una sola célula produce aproximadamente 1,2 V. Para el uso en sistemas de generación de energía, donde se requiere una potencia relativamente alta, se conectan varias células en serie, formando una pila que puede suministrar energía eléctrica. Las células de combustible PEM están disponibles comercialmente en una amplia gama de potencias (desde pocos W hasta algunos MW), lo que permite su uso en un gran número de aplicaciones [5].

Una pila de combustible se comporta dinámicamente como un sistema no lineal que genera energía eléctrica no regulada por una reacción electroquímica. Por lo tanto, el uso eficiente de la energía requiere una regulación. De esta forma, para la aplicación formal de estrategias de control sobre una célula de combustible, es muy importante contar con un modelo de esta [2, 3, 15, 19, 29]. En la práctica, el control de la potencia de salida de una pila de combustible se realiza a través de un convertidor CC/CC conectado en sus terminales de [1, 6, 26, 27], lo que puede suponer una fuente de ruido eléctrico que perturbe el resto de componentes del sistema.

3. Formulación del problema

Sea un modelo borroso general de n entradas y m salidas:

$$\begin{aligned} &R^{(l,i)}\text{: Si } x_1(k) \text{ es } A_{1i}^l \text{ y} \ldots \text{y } x_n(k) \text{ es } A_{ni}^l \\ &\text{Entonces } y_i^l(k) = a_{0i}^l + \sum_{j=1}^{n} a_{ji}^l x_j(k), \end{aligned} \tag{1}$$

donde $l = 1..M_i$ es el índice de la regla y M_i el número de reglas de la i-ésima salida del sistema,

y_i. Si $\tilde{\mathbf{x}} = (1, x_1, \ldots, x_n)^{\mathrm{T}}$ [4]:

$$y_i(k) = h_i(\mathbf{x}(k)) = \sum_{j=0}^{n} a_{ji}(\mathbf{x})\tilde{x}_j(k),$$

donde:

$$a_{ji}(\mathbf{x}) = \frac{\sum_{l=1}^{M_i} w_i^l(\mathbf{x}) a_{ji}^l}{\sum_{l=1}^{M_i} w_i^l(\mathbf{x})},$$

y

$$w_i^l(\mathbf{x}) = \prod_{j=1}^{n} \mu_{ji}^l(x_j(k), \boldsymbol{\sigma}_{ji}^l), \tag{2}$$

siendo $\boldsymbol{\sigma}_{ji}^l$ los parámetros adaptables de los antecedentes del modelo borroso.

4. Modelado Borroso mediante *EKF*

Sea $\mathbf{p}(k)$ el conjunto de parámetros a estimar de (1), el sistema (3) permite su obtención mediante la aplicación de las ecuaciones (4) a (6) del *EKF* [9].

$$\begin{aligned} \mathbf{p}(k+1) &= \mathbf{p}(k) \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k), \mathbf{p}(k)) + \mathbf{e}(k) \end{aligned} \tag{3}$$

$$\mathbf{K}_k = \mathbf{P}_{k-1}\mathbf{C}_k^{\mathrm{T}} \left(\mathbf{C}_k \mathbf{P}_{k-1} \mathbf{C}_k^{\mathrm{T}} + \mathbf{R}_e\right)^{-1} \tag{4}$$

$$\hat{\mathbf{p}}_k = \hat{\mathbf{p}}_{k-1} + \mathbf{K}_k \left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right) \tag{5}$$

$$\mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{K}_k \left(\mathbf{C}_k \mathbf{P}_{k-1}\right). \tag{6}$$

La expresión

$$\mathbf{C}_k = \frac{\partial \mathbf{h}(\cdot)}{\partial \mathbf{p}(\mathbf{k})}, \tag{7}$$

evaluada en $\mathbf{p}(k) = \hat{\mathbf{p}}_k$, es la matriz jacobiana del modelo borroso respecto de sus parámetros. Esta expresión fue obtenida en [9], siendo:

$$\frac{\partial h_i}{\partial a_{JI}^L} = \begin{cases} \frac{w_I^L x_J}{\sum_{l=1}^{M_i} w_i^l} & \text{si } i = I \\ 0 & \text{si } i \neq I, \end{cases} \tag{8}$$

para los consecuentes; y

$$\frac{\partial h_I}{\partial \boldsymbol{\sigma}_{JI}^L} = \frac{\partial w_I^L}{\partial \boldsymbol{\sigma}_{JI}^L} \sum_{j=0}^{n} \left(\frac{\sum_{l=1}^{M_I} w_I^l (a_{jI}^L - a_{jI}^l)}{\left(\sum_{l=1}^{M_I} w_I^l\right)^2} \right) \tilde{x}_j \tag{9}$$

para los antecedentes, siendo

$$\frac{\partial w_I^L}{\partial \boldsymbol{\sigma}_{JI}^L} = \frac{\partial \mu_{JI}^L(x_J, \boldsymbol{\sigma}_{JI}^L)}{\partial \boldsymbol{\sigma}_{JI}^L} \prod_{q=1, q\neq J}^{n} \mu_{qI}^L(x_q, \boldsymbol{\sigma}_{qI}^L). \tag{10}$$

1: $\mathbf{p}_c = \mathbf{0}$
2: $\mathbf{p}_a$ = inicializa los parámetros de los antecedentes
3: $\mathbf{P}_c(0|-1) = \mathbf{I}\alpha$
4: $\mathbf{P}_a(0|-1) = \mathbf{I}\beta$
5: **Para** $k = 0..k_{end}$ **hacer**
6: Estima $\tilde{\mathbf{y}}(k)$ empleando el modelo borroso
7: Calcula $\mathbf{C}_c(k)$ mediante (8)
8: Obtiene $\mathbf{K}_c(k)$ mediante (4)
9: Actualiza $\tilde{\mathbf{p}}_c(k|k)$ mediante (5)
10: Actualiza $\mathbf{P}_c(k)$ mediante (6)
11: .....
12: Estima $\tilde{\mathbf{y}}(k)$ empleando el modelo borroso
13: Calcula $\partial\mathbf{w}/\partial\boldsymbol{\sigma}$ mediante (10)
14: Calcula $\mathbf{C}_a(k)$ mediante (9)
15: Obtiene $\mathbf{K}_a(k)$ mediante (4)
16: Actualiza $\tilde{\mathbf{p}}_a(k|k)$ mediante (5)
17: Actualiza $\mathbf{P}_a(k|k)$ mediante (6)
18: **Fin Para** k

Figura 3: Algoritmo EKF para modelos TS.

4.1. Algoritmo de modelado

El algoritmo EKF que se muestra en la figura 3 es realmente la ejecución secuencial de dos filtros. En primer lugar, se realiza el ajuste de consecuentes, y posteriormente el ajuste de los antecedentes. Hemos comprobado que mediante otro secuencia de ajustes se obtienen normalmente peores resultados [10, 11], por lo que no se han considerado. Antes de ejecutar el algoritmo de modelado debe ser inicializado. Si no se posee información previa sobre el sistema que permita realizar una mejor inicialización, todos consecuentes pueden tomar el valor 0, o se pueden emplear otros procedimientos de inicialización [12]. Igualmente los antecedentes se inicializan con una distribución uniforme en cada uno de los universos de discurso de las entradas del modelo borroso.

α y β son números enteros positivos que representa la poca certeza que el algoritmo debe suponer para los parámetros iniciales, y $\mathbf{I}$ es la matriz identidad. α se utiliza para consecuentes y β para los antecedentes. La inicialización de los antecedentes suele ser mejor que la de los consecuentes, ya que estos se inicializan uniformemente distribuidos, por lo que la distinción de estos dos parámetros permite mejorar el rendimiento del algoritmo.

5. Modelado de la pila Nexa™ 1.2 kW

Para representar la dinámica de la pila se ha empleado un modelo discreto con 9 reglas, donde $i(k)$ es la corriente en el stack (entrada) y $v(k+1)$ la estimación de tensión en las bornas de la pila en el siguiente instante de muestreo (salida), siendo $T = 1$ms y asumiendo que la covarianza del ruido

es $R_e = 0{,}1$.

Los antecedentes se han inicializado como una distribución uniforme de una función de pertenencia de tipo Z, una *Gaussiana* y una función de pertenencia de tipo S para cada una de las variables de entrada, dadas por (11), (12) y (13), respectivamente.

$$\mu_{ZMF}[a,b](x)=\begin{cases}1 & \text{si} \quad x \le a \\ 1-2\left(\frac{x-a}{a-b}\right)^2 & \text{si } a < x < \frac{a+b}{2} \\ 2\left(\frac{b-x}{a-b}\right)^2 & \text{si } \frac{a+b}{2} \le x < b \\ 0 & \text{si} \quad x \ge b \end{cases} \tag{11}$$

$$\mu_{Gauss}[c,\beta](x) = e^{-\left(\frac{x-c}{\beta}\right)^2} \tag{12}$$

$$\mu_{SMF}[a,b](x)=\begin{cases}0 & \text{si} \quad x \le a \\ 2\left(\frac{x-a}{b-a}\right)^2 & \text{si } a < x \le \frac{a+b}{2} \\ 1-2\left(\frac{b-x}{b-a}\right)^2 & \text{si } \frac{a+b}{2} < x < b \\ 1 & \text{si} \quad x \ge b \end{cases} \tag{13}$$

Se ha aplicado al sistema la entrada figura 4, empleando el algoritmo descrito en la sección 4.1 con $\alpha = 100$ y $\beta = 10$ mientras se realizaba en modelado en línea hasta los 7s. En este instante se detuvo el modelado y los datos restantes se emplearon para validar el modelo obtenido. Los resultados pueden observarse en las figuras 5 y 6.

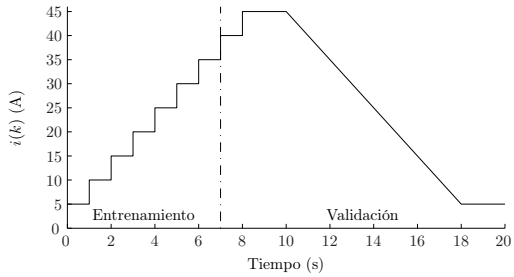


Figura 4: Corriente requerida a la pila.

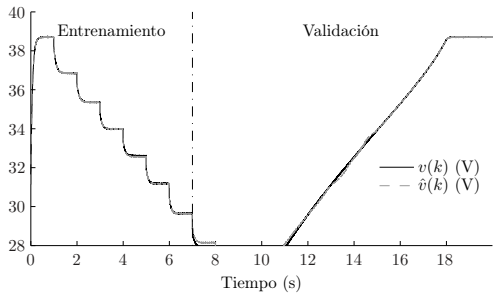


Figura 5: Salida real, $v(k)$, y del modelo, $\hat{v}(k)$.

Atendiendo a los resultados anteriores, se ha demostrado que mediante un borroso TS ajustado

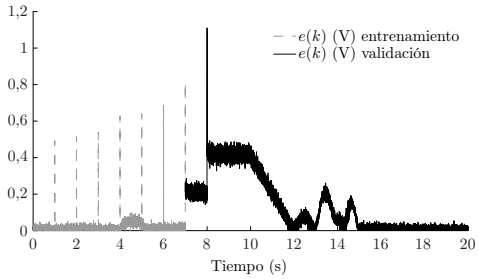


Figura 6: Errores del modelo final.

a partir del algoritmo EFK se puede obtener un modelo no lineal muy preciso de la pila de combustible mientras ésta está en funcionamiento, manteniendo la calidad del modelo borroso a lo largo el espacio de trabajo recorrido. Este modelo describe las relaciones dinámicas entre las diferentes variables eléctricas con precisión, y es válido para cualquier punto de funcionamiento de la pila de combustible Nexa™.

El modelo borroso final obtenido por el algoritmo de ajuste, está representado por el siguiente conjunto de reglas:

SI $v(k)$ es ZMF(29.6668;32.9965) y $i(k)$ es ZMF(14.7412;29.1693)
ENTONCES $v(k+1)$ = 0.377315+0.933322*$v(k)$+0.351124*$i(k)$

SI $v(k)$ es ZMF(31.3909;33.0613) y $i(k)$ es GAUSSMF(26.4112;2.41434)
ENTONCES $v(k+1)$ = -0.259369+1.14315*$v(k)$-0.206034*$i(k)$

SI $v(k)$ es ZMF(29.7019;33.3294) y $i(k)$ es SMF(19.3452;35.2423)
ENTONCES $v(k+1)$ = -1.7993+1.00791*$v(k)$+0.0446638*$i(k)$

SI $v(k)$ es GAUSSMF(31.7362;1.52197) y $i(k)$ es ZMF(14.9893;29.3646)
ENTONCES $v(k+1)$ = 1.01203+0.972514*$v(k)$-0.00493071*$i(k)$

SI $v(k)$ es GAUSSMF(31.6104;1.66393) y $i(k)$ es GAUSSMF(25.9034;3.6526)
ENTONCES $v(k+1)$ = -2.40872+0.937403*$v(k)$+0.176077*$i(k)$

SI $v(k)$ es GAUSSMF(32.7601;1.2303) y $i(k)$ es SMF(19.5093;34.3062)
ENTONCES $v(k+1)$ = 2.46914+1.11499*$v(k)$-0.181523*$i(k)$

SI $v(k)$ es SMF(30.9826;35.2274) y $i(k)$ es ZMF(14.6052;29.9377)
ENTONCES $v(k+1)$ = 0.539332+0.986622*$v(k)$-0.00451959*$i(k)$

SI $v(k)$ es SMF(30.1432;35.1474) y $i(k)$ es GAUSSMF(24.6284;2.59194)
ENTONCES $v(k+1)$ = 2.51013+1.12075*$v(k)$-0.317147*$i(k)$

SI $v(k)$ es SMF(30.9834;35.1675) y $i(k)$ es SMF(19.8283;35.7224)
ENTONCES $v(k+1)$ = 1.19081+2.27903*$v(k)$-1.56005*$i(k)$

donde ZMF, GAUSSMF y SMF representan respectivamente las funciones de pertenencia Z, S y *Gauss* dadas por las expresiones (11), (12) y (13), cuya distribución final puede verse en la figura 7.

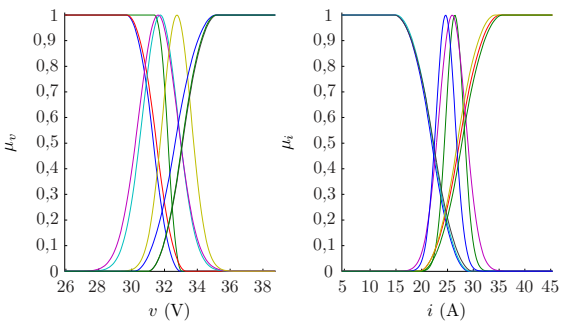


Figura 7: Antecedentes finales.

6. Conclusiones

En este trabajo se ha abordado el problema de modelar un sistema complejo como es una pila de combustible de tipo PEM. La metodología de modelado en línea basada en el filtro de Kalman extendido ha mostrado un excelente comportamiento, permitiendo obtener un modelo preciso de la pila Nexa™ 1.2 kW de Ballard en presencia de ruido, con una rápida convergencia y manteniendo un coste computacional reducido.

Referencias

[1] Md Shah Alam and M. Tanrioven. Modeling and fuzzy logic control of DC-DC converter for proton exchange membrane fuel cell. In Hamza M.H., editor, *International Conference on Modelling and Simulation, IASTED'05*, pages 404–407, Cancunm, Mexico, May 2005.

[2] John C. Amphlett, Rob M. Baumert, Ronald F. Mann, Brant A. Peppley, Pierre R. Roberge, and Thomas J. Harris. Performance modeling of the Ballard Mark IV solid polymer electrolyte fuel cell i. Mechanistic model development. *Journal of the Electrochemical Society*, 142(1):1–8, January 1995.

[3] John C. Amphlett, Ronald F. Mann, Brant A. Peppley, Pierre R. Roberge, and Aida Rodrigues. A model predicting transient responses of proton exchange membrane fuel cells. *Journal of Power Sources*, 61(1-2):183–188, February 1996. cited By (since 1996) 216.

[4] José Manuel Andújar, Antonio Javier Barragán, and Manuel Emilio Gegúndez. A general and formal methodology for designing stable nonlinear fuzzy control systems. *IEEE Transactions on Fuzzy Systems*, 17(5):1081–1091, October 2009.

[5] José Manuel Andújar and Francisca Segura. Fuel cells: History and updating. A walk along two centuries. *Renewable and Sustainable Energy Reviews*, 13(9):2309–2322, December 2009.

[6] José Manuel Andújar, Francisca Segura, Eladio Durán, and Luciano Alonso Rentería. Optimal interface based on power electronics in distributed generation systems for fuel cells. *Renewable Energy*, 36(11):2759–2770, November 2011.

[7] Robert Babuška. Fuzzy modeling – a control engineering perspective. In *Proceedings of FUZZ-IEEE/IFES'95*, volume 4, pages 1897–1902, Yokohama, Japan, March 1995.

[8] Antonio Javier Barragán, José Manuel Andújar, Mariano Aznar, and Agustín Jiménez. Application of the extended Kalman filter to fuzzy modeling: algorithms and practical implementation. In Sylvie Galichet, Javier Montero, and Gilles Mauris, editors, *7th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2011) and LFA-2011*, volume 1 of *Advances in Intelligent Systems Research*, pages 691–698, Aix-les-Bains, France, July 2011.

[9] Antonio Javier Barragán, José Manuel Andújar, Mariano Aznar, and Agustín Jiménez. Methodology for adapting the parameters of a fuzzy system using the extended Kalman filter. In Sylvie Galichet, Javier Montero, and Gilles Mauris, editors, *7th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2011) and LFA-2011*, volume 1 of *Advances in Intelligent Systems Research*, pages 686–690, Aix-les-Bains, France, July 2011.

[10] Antonio Javier Barragán, José Manuel Andújar, Agustín Jiménez, and Basil M. Al-Hadithi. Adaptación paramétrica de un sistema borroso mediante el filtro de Kalman extendido. In CEA-IFAC, editor, *XXXI Jornadas de Automática*, pages 1–6, Jaén, España, September 2010.

[11] Antonio Javier Barragán, Mariano Aznar, and José Manuel Andújar. Estimador recursivo subóptimo para el ajuste de modelos borrosos ts. In *XVI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF'12)*, pages 354–359. European Society for Fuzzy Logic and Techonology (EUSFLAT), February 2012.

[12] Farida Benmakrouha. Parameter identification in a fuzzy system with insufficient data. volume 1, pages 537–542. IEEE, Piscataway, NJ, United States, 1997.

[13] M. Bertoluzzo and G. Buja. Development of electric propulsion systems for light electric vehicles. *Industrial Informatics, IEEE Transactions on*, 7(3):428–435, August 2011.

[14] K. Chafaa, M. Ghanaï, and Khier Benmahammed. Fuzzy modelling using Kalman filter. *IET Control Theory and Applications*, 1(1):58–64, 2007.

[15] P. Famouri and R.S. Gemmen. Electrochemical circuit model of a PEM fuel cell. In *Power Engineering Society General Meeting, 2003, IEEE*, volume 3, pages 1436–1440, July 2003.

[16] Shin-ichi Horikawa, Takeshi Furuhashi, and Yoshiki Uchikawa. On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm. *IEEE Transactions on Neural Networks*, 3(5):801–806, September 1992.

[17] Jyh-Shing Roger Jang. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, May 1993.

[18] Tao Jiang and Yao Tang Li. Generalized defuzzification strategies and their parameter learning procedures. *IEEE Transactions on Fuzzy Systems*, 4(1):64–71, August 1996.

[19] Junbom Kim, Seong-Min Lee, Supramaniam Srinivasan, and Charles E. Chamberlin. Modeling of proton exchange membrane fuel cell performance with an empirical equation. *Journal of the Electrochemical Society*, 142(8):2670–2674, August 1995.

[20] A. Kirubakaran, S. Jain, and R.K. Nema. A review on fuel cell technologies and power electronic interface. *Renewable and Sustainable Energy Reviews*, 13(9):2430–2440, December 2009.

[21] Fernando Matía, Agustín Jiménez, Basil M. Al-Hadithi, Diego Rodríguez-Losada, and Ramón Galán. The fuzzy Kalman filter: State estimation using possibilistic techniques. *Fuzzy Sets and Systems*, 157(16):2145–2170, August 2006.

[22] Marcos V. Moreira and Gisele E. da Silva. A practical model for evaluating the performance of proton exchange membrane fuel cells. *Renewable Energy*, 34(7):1734–1741, July 2009.

[23] Józef Paska, Piotr Biczel, and Mariusz Kłos. Hybrid power systems - an effective way of utilising primary energy sources. *Renewable Energy*, 34(11):2414–2421, November 2009.

[24] Pramath Ramaswamy, Martin Riese, Robert M. Edwards, and Kwang Youl Lee. Two approaches for automating the tuning process of fuzzy logic controllers. In *32nd IEEE Conference on Decision and Control. Part 2 (of 4)*, San Antonio, TX, USA, December 1993.

[25] D. Ross. Power struggle [power supplies for portable equipment]. *IEE Review*, 49(7):34–38, July 2003.

[26] Francisca Segura, José Manuel Andújar, and Eladio Durán. Analog current control techniques for power control in PEM fuel-cell hybrid systems: A critical review and a practical application. *IEEE Transactions on Industrial Electronics*, 58(4):1171–1184, april 2011.

[27] A.Y. Sendjaja and V. Kariwala. Decentralized control of solid oxide fuel cells. *Industrial Informatics, IEEE Transactions on*, 7(2):163–170, May 2011.

[28] Dan Simon. Training fuzzy systems with the extended Kalman filter. *Fuzzy Sets and Systems*, 132(2):189–199, 2002.

[29] H.P.L.H. Van Bussel, F.G.H. Koene, and Ronald K.A.M. Mallant. Dynamic model of solid polymer fuel cell water management. *Journal of Power Sources*, 71(1-2):218–222, March 1998.

[30] Li-Xin Wang. *A course in fuzzy systems and control*. Prentice Hall, New Yersey, USA, 1997.

CONTROL DE ANESTESIA MEDIANTE LÓGICA DIFUSA CON DINÁMICA INVERSA

Ayoze Marrero Ramos[1]
[1]Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores (ISAATC), Facultad de Física, Universidad de La Laguna (ULL), Avda. Astrofísico Fco. Sánchez s/n, 38206, La Laguna (Tenerife), España, ayozemarrero@gmail.com

J.A. Méndez Pérez[1], J.A. Reboso Morales[2], A. M. León Fragoso[2], I. Martín Mateos[3], E. Morell González[1]
[2]Departamento de Farmacología del Hospital Universitario de Canarias, La Laguna, España, jreboso@comtf.es
[3]Departamento de Física Fundamental y Experimental, Electrónica y Sistemas, Facultad de Física, Universidad de La Laguna (ULL), Avda. Astrofísico Fco. Sánchez s/n, 38206, La Laguna (Tenerife), España

Resumen

En este trabajo se presenta una metodología para el modelado y control del proceso anestésico en pacientes humanos usando como fármaco propofol intravenoso en la región de anestesia general. En concreto el trabajo presenta técnicas difusas para describir el comportamiento del Índice Biespectral (BIS) en función de la velocidad de infusión de propofol. En el proceso de ajuste del modelo se emplean técnicas basadas en algoritmos genéticos para ajustar los modelos a cada paciente particular. El modelo resultante es invertido y usado para calcular la infusión basal que se aplica al paciente. El cálculo final de la velocidad de infusión se realiza sumando a esta acción nominal la salida de un controlador difuso que usa información del BIS y su derivada. Para el desarrollo del sistema de control difuso, que consta de dos entradas y una salida, se incorpora la experiencia de un experto médico con lo que se consigue reproducir el comportamiento del médico en cada estado anestésico. Para validar la propuesta se han realizado simulaciones sobre pacientes cuya dinámica es descrita mediante modelos compartimentales.

Palabras Clave: Anestesia, BIS, Propofol, Modelado Difuso, Control Difuso, Lógica Difusa, Modelo Inverso.

1 Introducción

El uso de técnicas de automatización aplicadas al campo de la anestesiología es una realidad que se encuentra en pleno desarrollo. El objetivo de la anestesiología es proteger al paciente de la agresión de una intervención quirúrgica, basado en los principios de la farmocología clínica. Un proceso anestésico ideal puede ser definido como un estado farmacológico de un paciente donde se garantiza la relajación muscular (limitación de movimientos), la analgesia (nivel de dolor) y la hipnosis (nivel de inconciencia). Para conseguir una anestesia adecuada los anestesistas ajustan regularmente la configuración de diferentes dispositivos de infusión de fármacos para así mantener las variables de interés en los niveles adecuados.

El presente trabajo trata el modelado y control del nivel de hipnosis del paciente durante la intervención quirúrgica sometido a anestesia general, usando propofol intravenoso como fármaco.

Actualmente la técnica más eficiente para obtener una medida del grado de hipnosis se basa en el procesamiento de la señal del electroencefalograma (EEG). Una de las medidas más fiables es el índice biespectral (BIS) [1][2]. Este índice se calcula como un promedio ponderado de tres parámetros que analizan la relación entre fase, frecuencia e intensidad de la respuesta del EEG.

El BIS es por lo tanto un parámetro empírico, adimensional, derivado del análisis del EEG que se relaciona lineal y directamente con el nivel de consciencia, su valor disminuye desde 100, que es el estado de máxima alerta posible, pasando progresivamente por la pérdida de funciones corticales hasta 0, que es la ausencia de actividad eléctrica cortical y por tanto el estado de inconsciencia más profundo. La banda de anestesia general estaría en el rango 40-60. En nuestro caso consideramos como BIS objetivo el valor 50.

Existen diferentes propuestas en la literatura sobre control de hipnosis con algoritmos basados en PID [12], control predictivo [14], control bayesiano [5], control por modelo interno [3], etc. También existen algunas propuestas que avanzan en el desarrollo de sistemas de control basados en lógica difusa [10]. Si embargo, la propuesta que se hace en este paper se beneficia con respecto a la spropuestas existentes en que se combina información del modelo del paciente, se integra la experiencia del experto médico y se

realiza una adaptación del controlador a las necesidades reales del paciente particular.
Además, el paciente se beneficiará porque se evitan sobredosis (que podría ser un riesgo para su salud) y también da como resultado tiempos de recuperación más bajos.
El trabajo se divide en dos partes fundamentales: la primera explicará las técnicas de modelado directo e inverso de los pacientes, y en la segunda parte se explicará el algoritmo de control del perfusor usando para ello el modelo difuso inverso y un sistema de inferencia difuso de control en lazo cerrado.

2 Fundamento de la Lógica Difusa

La opción de la Lógica Difusa está motivada por la versatilidad de esta herramienta en el modelado de sistemas complejos en los que el conocimiento sobre los principios físico químicos que intervienen es limitado. La idea es sintetizar modelos online que se adapten a las características de cada paciente. Esto resultaría de gran ayuda a la hora de saber cómo actuar frente a pacientes sensibles a diversas drogas o con grados de reacción poco comunes.
Esta simplicidad en la síntesis de modelos difusos surge como consecuencia de la característica de la lógica difusa que intenta modelar la forma en que los seres humanos representamos y procesamos la información sensorial a "alto nivel", y la forma "difusa" (relativa/subjetiva) con que manejamos lingüísticamente los conceptos.
Existen numerosas publicaciones que explican el funcionamiento general de los sistemas de inferencia difusa [6][8][9][13].
Se distinguen dos elementos fundamentales: los conjuntos y las reglas. Los conjuntos expresan las zonas en las que "lingüísticamente" se mueven las variables de entrada y salida. En este trabajo se ha considerado un sistema difuso con tres entradas (BIS, pendiente del BIS, velocidad de infusión) y una salida (variación de la velocidad de infusión). Para ilustrar la definición de los conjuntos difusos considérese el caso de la variable BIS. En la Figura 1 observamos que sus conjuntos se definen como BIS "bajo" y "alto". Cada uno cubriendo con un grado de pertenencia distinto el universo de discurso del BIS, desde 0 hasta 100.

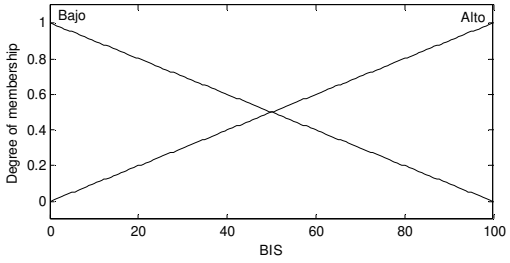


Figura 1. *Conjuntos de la variable de entrada BIS del modelo difuso.*

Así tenemos que un BIS medio estará comprendido entre 40 y 60, teniendo su mayor relevancia en las cercanías de 50 debido a que ninguna función de pertenencia tiene un grado de pertenencia excesivamente mayor a otra. El proceso por el que se transforma el valor numérico del BIS en un conjunto lingüístico de pertenencia es llamado fuzificación.
Las reglas son las que actúan sobre los conjuntos de entrada, obteniendo los conjuntos de salida adecuados. En nuestro trabajo dichas reglas, al igual que los conjuntos, se crean a través de datos obtenidos en sesiones de quirófano reales, de la experiencia del profesional, y de algoritmos inteligentes para mejorar el resultado final.
La fórmula que rige las reglas es como sigue:
SI la entrada *"A"* es el conjunto *"A_x"* Y la entrada *"B"* es el conjunto *"B_x"* ENTONCES la salida *"S"* es el conjunto *"S_x"*
Así un ejemplo de una regla aplicada al caso del modelo quedaría de la siguiente forma,
SI "el BIS" es "alto" Y "la pendiente del BIS" es "ascendente" Y "la velocidad de infusión" es "alta" ENTONCES "el incremento del BIS" es "muy negativo"
Por último, tras obtener el conjunto de salida se pasa al proceso de defuzificación, por el cual convertimos el conjunto en una variable numérica de salida. Utilizamos el método del centro del área (o centroide).
Como se observa estas reglas son de fácil interpretación al utilizar un lenguaje de alto nivel similar al humano, y es por ello que también son de gran utilidad para incorporar la experiencia médica en la creación de modelos o en el diseño de controladores.

3 Modelado del paciente

En este trabajo se usan diferentes modelos para describir la dinámica del paciente. Por un lado se usará un modelo estándar compartimental que describe la farmacocinética y la farmacodinámica del paciente. Este modelo se empleará para simular el comportamiento del paciente. [16]
Por otra parte se modelará usando lógica difusa el comportamiento de este paciente. En realidad el controlador no usa este modelo directamente sino que invierte este modelo para obtener un valor basal de infusión que será la que se aplique al paciente. En las siguientes subsecciones se explica la obtención tanto del modelo directo como de su inversión para el cálculo de la dosis de base.

3.1 Modelado directo

Para desarrollar el modelo difuso directo se definen inicialmente las variables de entrada y de salida de las que constará. En un paciente real se regula la

velocidad de infusión del propofol en relación a varias variables (nivel de inconsciencia, presión arterial, saturación, etc.). En este trabajo el fármaco se infundirá tomando en cuenta solo el nivel de inconsciencia, que es la variable de mayor influencia en el proceso. Como ya se ha comentado la medida de la inconsciencia del paciente de hará mediante el BIS. Por lo tanto y como se observa en la figura 2 el parámetro de entrada del modelo del paciente es la velocidad de infusión, y como entradas adicionales existe una realimentación del BIS y su derivada, estas entradas adicionales son necesarias para conocer el último punto de BIS conocido y la tendencia que éste lleva. La salida será la tendencia en el instante siguiente del BIS, también llamada variación o pendiente del BIS.

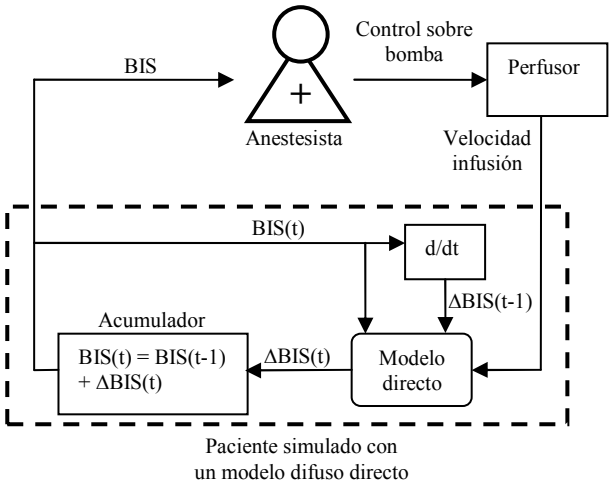
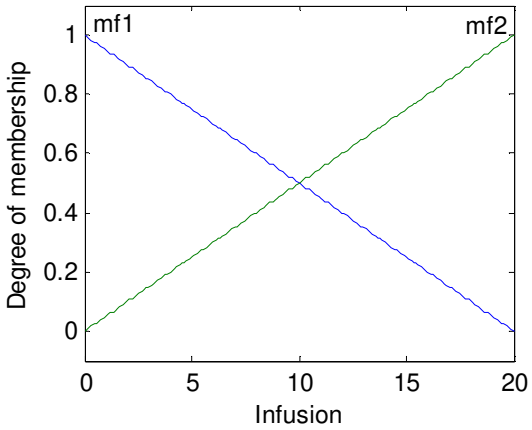


Figura 2.*Ejemplo de modelo directo con Lógica Difusa.*

El diseño de las funciones de pertenencia se puede observar en las siguientes figuras, la entrada para la variable de la velocidad de infusión y del BIS constan de dos funciones de pertenencia, la tercera entrada se forma con tres funciones de pertenencia, siempre repartidas equidistantemente.



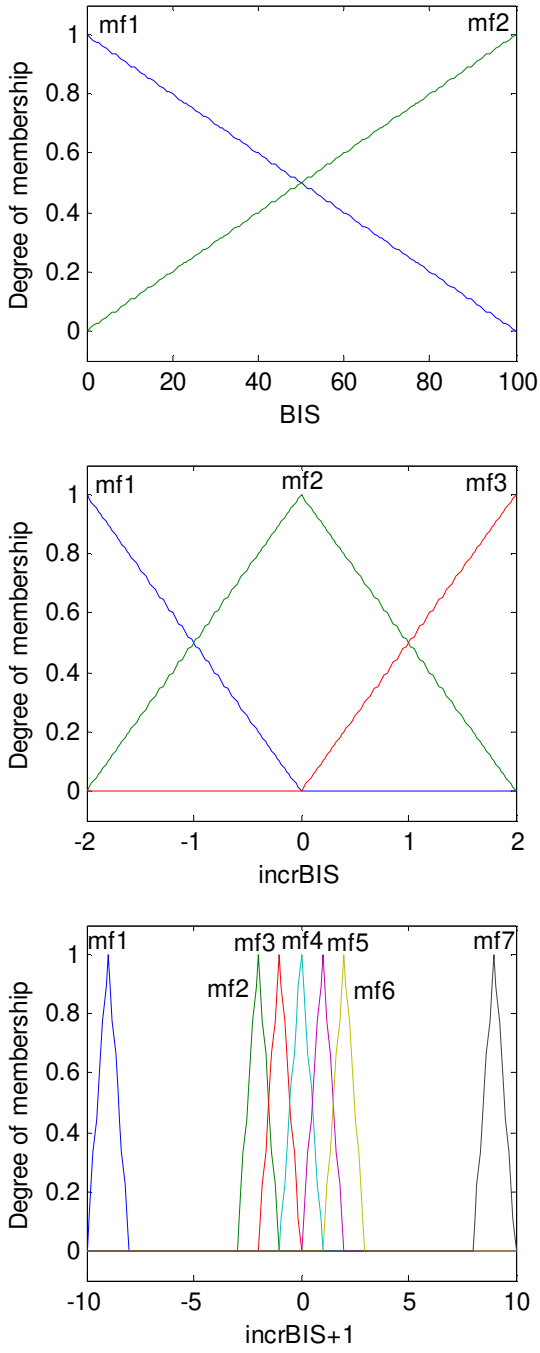


Figura 3.*Entradas y salida del modelo directo difuso.*

Tras aplicar un entrenamiento con un algoritmo automático sobre un conjunto de 15 casos reales de pacientes e incorporando la experiencia del experto médico, se generan un conjunto de 50 reglas.
Este algoritmo se basa en la siguiente estrategia:
Con los datos reales de un paciente se genera una serie de ternas en un intervalo dado. Así llamando W a esta serie de datos se tiene

$$W = [X, Y, Z] \qquad (1)$$

donde

$$X = [u_{a-r}, u_{a-r+1}, ..., u_b] \quad (2)$$

$$Y = [BIS_a, BIS_{a+1}, ..., BIS_b] \quad (3)$$

$$Z = [\Delta BIS_{a+1}, \Delta BIS_{a+2}, ..., \Delta BIS_{b+1}] \quad (4)$$

siendo

$$\Delta BIS_t = BIS_t - BIS_{t-1} \quad (5)$$

Donde *a* y *b* representan el intervalo de puntos recogidos [a, b], *u* es la velocidad de infusión, y *r* representa el retardo generado entre la velocidad de infusión y la relación del BIS.

Con las muestras ya tratadas en el conjunto W, se buscan todas las combinaciones de reglas que se pueden hacer con los conjuntos del modelo borroso ya prefijado.

$$L_1 = If(MF_{W_a(1)}) \& (MF_{W_a(2)}) \& (MF_{W_a(3)}) Then(MF_{W_{a+1}(3)})$$

…

$$L_{b-a} = If(MF_{W_b(1)}) \& (MF_{W_b(2)}) \& (MF_{W_b(3)}) Then(MF_{W_{b+1}(3)})$$

Donde $MF_{Wi(j)}$ representa la función de pertenencia de mayor grado para el valor *i* en la entrada *j*. A continuación se eliminan todas las reglas repetidas que puedan haber surgido.

Tras obtener las reglas para el modelo borroso, es necesario un entrenamiento de dicho sistema para ajustarlo a los pacientes. Este entrenamiento se hace aplicando un algoritmo genético para el ajuste exclusivo de los pesos de las reglas.

Los algoritmos genéticos [4][7][11][15] están ampliamente estudiados y se usan con éxito en multitud de aplicaciones. En la siguiente figura se observa el diagrama de flujo básico de un algoritmo genético.

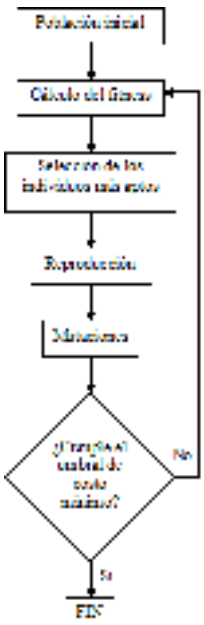


Figura 4.*Diagrama de flujo de un algoritmo genético básico.*

El algoritmo genético utilizado en este caso se basa en la creación de individuos, cuyos elementos o genes serán los valores a aplicar en los pesos de las reglas del sistema borroso.

La población inicial está formada por un conjunto de *n* individuos con los genes creados aleatoriamente.

$$P = [ind_1, ind_2, ..., ind_n] \quad (6)$$

$$ind = [w_1, w_2, ..., w_k] \quad (7)$$

$$0 \le w \le 1 \quad (8)$$

Donde *P* es la población elegida, *ind* es un individuo, que está compuesto por *k* genes, donde *k* es el número de reglas del sistema borroso. *w* es el peso que se añadirá a cada regla.

Cada individuo modifica los pesos del sistema borroso, se ejecuta la simulación sobre los datos originales del paciente y se obtiene una función de costo $J(ind_x)$. Este paso es llamado el cálculo del *fitness*.

Seguidamente se seleccionan los mejores individuos, aquellos con la menor función de costo, eliminando los restantes. En nuestro caso usamos un número *n* par, seleccionando justo la mitad con mejores resultados de función de costo.

Con los individuos seleccionados se crean nuevos individuos. En este paso se mezclan los genes de dos individuos elegidos aleatoriamente, creando uno nuevo. Esta mezcla se lleva a cabo usando el denominado *crossover uniforme*, en el que se decide de manera aleatoria los genes que serán intercambiados entre ambos individuos para que sean heredados por el nuevo individuo.

Tras la creación de la nueva generación, volviendo a tener un número *n* de individuos, se genera una mutación en uno de los nuevos individuos creados. Esta mutación se representa como el cambio de uno o más genes de manera aleatoria, creando de esta manera diversidad dentro de la población.

Si no se ha cumplido el objetivo deseado mínimo de la función de costo (o no se ha alcanzado un número de iteraciones determinado) se repite el proceso desde el cálculo del *fitness* con la nueva población. En caso de que se haya alcanzado el umbral mínimo deseado de costo (o se haya repetido el bucle un número de veces determinado) el algoritmo genético finaliza, dando como resultado el individuo más óptimo encontrado para los pesos de las reglas del sistema borroso.

Este modelo al que llamamos modelo estándar es una generalización con la cual se reproducen comportamientos aceptables en la mayoría de pacientes. Pero cada paciente individual tiene su propia respuesta a los fármacos que lo hace único. Para ajustar el modelo a un paciente particular se construye un algoritmo genético que minimiza el error del modelo sobre el paciente actual en un intervalo dado. Este proceso se realiza online a medida que se reciben datos del comportamiento real del paciente.

La sensibilidad del paciente se obtiene modificando el ancho de las funciones de pertenencia de la velocidad de infusión.

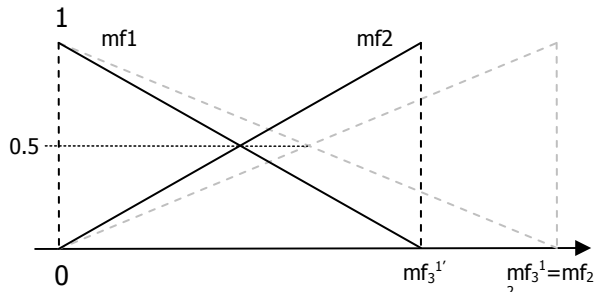


Figura 5.*Ajuste de las funciones de pertenencia de la velocidad de infusión.*

Como se muestra en la figura 5 se modifica la posición el punto mf_3^1 que coincide con el borde derecho de la función de pertenencia mf1 y con el punto máximo de la función mf2. Con este cambio ensanchamos o estrechamos las funciones, lo cual implica que el modelo reaccionará con mayor o menor intensidad a las mismas velocidades de infusión.

3.2 Modelado inverso

El interés del modelo inverso es la incorporación de estrategias que usan dinámica inversa para calcular la ley de control. De dicho modelo se obtendrá la velocidad aproximada que deberá suministrarse al paciente para alcanzar un BIS determinado.
Al contrario que en el modelo directo, en el modelo inverso deseamos conocer la velocidad de infusión de propofol que debe suministrarse a un paciente para obtener el BIS deseado en dicho paciente. Es por ello que el sistema difuso en este caso consta de una única variable de entrada, el BIS objetivo, y una variable de salida, la velocidad de infusión, como se puede observar en la figura 6.

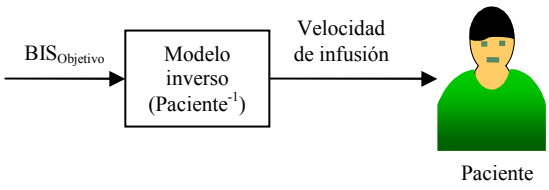


Figura 6.*Esquema del sistema de control para el modelo difuso inverso.*

Para la creación del modelo inverso es necesario conocer el valor estacionario del BIS del paciente correspondiente a distintas velocidades de infusión, y con estos datos modelar un sistema de inferencia difuso con capacidad de generalización. De este modo se podría conocer la velocidad aproximada de infusión que debe suministrarse al paciente para alcanzar cualquier punto del intervalo del BIS.

La generación de la dupla de datos [BIS estacionario; velocidad de infusión] se obtiene sometiendo el modelo directo difuso a distintas velocidades constantes equidistantes entre ellas, simulando durante un tiempo suficientemente largo para obtener una tendencia al estacionario aproximada. En la figura 7 se observa un ejemplo

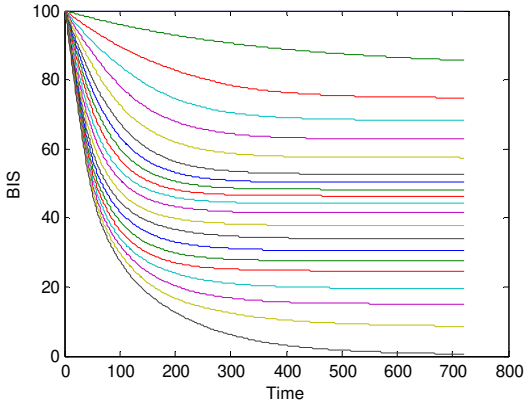


Figura 7.*Tendencia hacia el estacionario del modelo directo difuso según distintos valores constantes de velocidad equidistantes entre sí.*

Recolectando el conjunto de datos formados por duplas que relacionan una velocidad de infusión con un BIS aproximado al estacionario, se genera un modelo difuso inverso. Esta generación se obtiene mediante un sistema neurodifuso, usando los datos de entrada/salida. El sistema de inferencia difuso (FIS) construido se basa en el uso de funciones de pertenencia ajustadas usando solamente un algoritmo neuronal de retropropagación (backpropagation), o en combinación con un método de mínimos cuadrados. El resultado de la generación del modelo inverso puede observarse en la figura 8. Donde aplicando en la entrada el BIS que se desea alcanzar el sistema devuelve como respuesta en la salida la velocidad nominal consecuente.

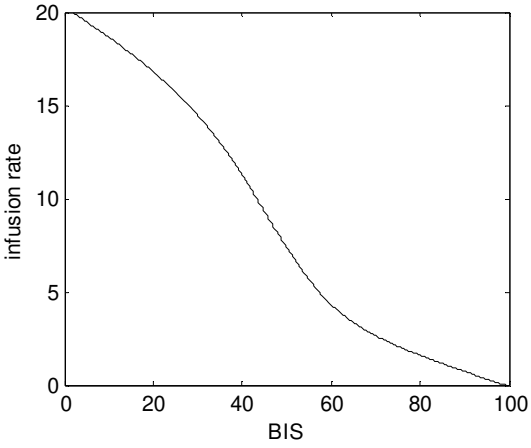


Figura 8.*Respuesta del modelo inverso difuso creado con el modelo directo difuso.*

4 Algoritmo de control

El algoritmo utilizado en este trabajo se basa en un sistema de control y de dinámica inversa. En la figura 9 se observa el diagrama de bloques del sistema de control.

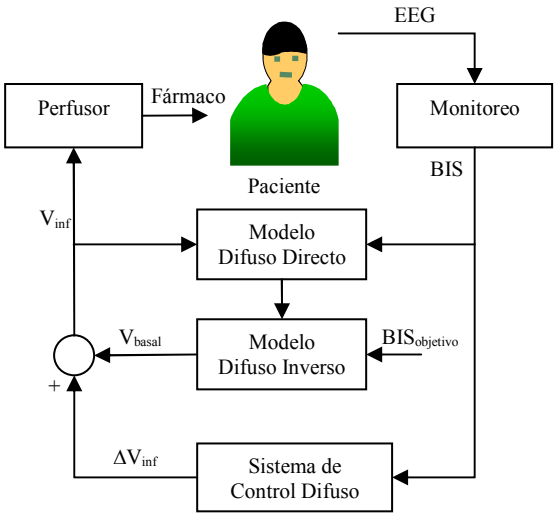


Figura 9. *Diagrama del sistema de control difuso con dinámica inversa.*

El modelo difuso directo se obtiene y ajusta comparando en intervalos de tiempo determinados la velocidad de infusión V_{inf} con la respuesta de BIS del paciente. Cuando el modelo directo ha sido ajustado el modelo difuso inverso se genera utilizando el algoritmo anteriormente explicado en el capítulo 3. Dicho modelo inverso dará como resultado una velocidad de infusión basal V_{basal} que dependerá del BIS objetivo que se desea alcanzar.

Para obtener un ajuste fino y evitar offset generados por los modelos sobre el paciente, se incluye un sistema de control difuso que mide el error de BIS actual y como respuesta genera una variación, ΔV_{inf}, que se agrega a la velocidad basal. Por lo tanto la velocidad final dada al perfusor del fármaco es

$$V_{\inf} = V_{basal} + \Delta V_{\inf} \qquad (9)$$

5 Resultados

Se ha validado esta propuesta mediante simulación usando un modelo de paciente habitualmente empleado para describir el proceso anestésico (modelo de Schnider). El tiempo de muestreo fue de 5 segundos y la consigna para el BIS fue de 50.

La primera fase de la simulación hace uso de un modelo difuso directo estándar del cual se determina un primer modelo inverso. Tras los 10 primeros minutos, y en intervalos de 5 minutos, el modelo directo es ajustado según los datos de BIS y velocidad de infusión obtenidos en los 10 minutos anteriores al tiempo actual. En la figura 10 se observa claramente que en los primeros 20 minutos el sistema de control difuso genera un gran desvío sobre le velocidad basal, ya que se ha usado un modelo estándar aún sin ajustar y el primer ajuste aún no es preciso. También se observa que con el paso del tiempo la basal va tendiendo aproximadamente a 5 mg/kg/h y ΔV_{inf} creado por el sistema de control difuso deja de tener un papel relevante.

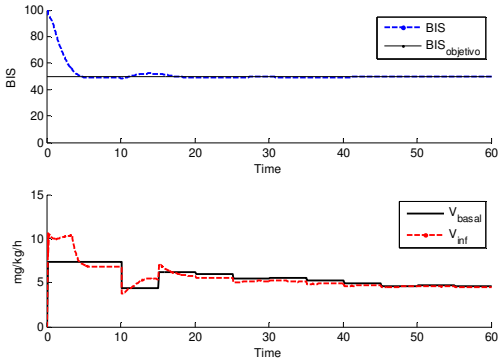


Figura 10. *Resultado en simulación.*

Obsérvese que el comportamiento de la señal del BIS es muy satisfactorio con una convergencia hacia la consigna suave y relativamente rápida.

6 Conclusiones

En este trabajo se presenta una metodología para el control del proceso anestésico de pacientes sometidos a intervención quirúrgica.

Las técnicas empleadas se basan en el uso de lógica difusa para modelar dinámicas directas e inversas de la hipnosis de los pacientes, y para crear un sistema de control difuso como alternativa a otras técnicas estándar basadas en controladores PID o predictivos.

El algoritmo propuesto incorpora el modelado en tiempo real del paciente de modo que se garantiza la robustez de la estrategia frente a la variabilidad interpaciente. Se han llevado a cabo simulaciones de esta estrategia con resultados satisfactorios que muestran una evolución de la señal de BIS con offset nulo y con transitorios suaves.

A partir de la valoración de estos primeros resultados simulados se concluye que este algoritmo aparece como una alternativa viable para las intervenciones de pacientes sometidos a anestesia intravenosa con propofol.

Agradecimientos

Esta investigación ha sido financiada con fondos del proyecto DPI2010-18278 del Ministerio de Ciencia e Innovación.

Referencias

[1] Agarwal, J., Puri, G.D., Mathew, P.J., (2009) *"Comparison of closed loop ve. Manual administration of propofol using the Bispectral index in cardiac surgery"*, Acta Anaesthesia Scand, 53, 3, 390-397, England

[2] Altermatt, F.R., y otros, (2012) *"Evaluation of the effect of intravenous lidocaine on propofol requirements during total intravenous anaesthesia as measured by bispectral index"*, Br.J.Anaesth.

[3] Anna, R., Wen, P., (2010) *"Depth of anesthesia control using internal model control techniques"*, IEEE/ICME Int. Conf. Complex Med. Eng., CME, 294-300, Gold Coast, QLD

[4] Booker, L.B., Goldberg, D.E., Holland, J.H. (1989) *"Classifier systems and genetic algorithms"*. Artif. Intell. 40, 1-3, 235-282

[5] De Smet, T., Struys, M., y otros, (2008) *"The Accuracy and Clinical Feasibility of a New Bayesian-Based Closed-Loop Control System for Propofol Administration Using the Bispectral Index as a Controlled Variable"*, Anesthesia & Analgesia, 107, 4, 1200-1210.

[6] Fuzzy Logic Toolbox™ 2, User's Guide. The MathWorks, Inc. 2008.

[7] Goldberg, D.E. (1986) *"GENETIC ALGORITHM APPROACH: WHY, HOW, AND WHAT NEXT?", 347-253.*

[8] J. Espinosa, J. Vandewalle, V. Wertz. (2005) *"Fuzzy Logic, Identification and Predictive Control"*. Ed. Springer. (ISBN: 1-85233-828-8)

[9] J. Espinosa, J. Vandewalle. (1999) *"Predictive control using fuzzy models"*, submited to the 3rd. On-Line World Conference on Soft Computing in Engieneering Design and Manufacturing, Springed-Verlag London Limited, pag. 187-2201.

[10] Moore, B.L., Doufas, A.G., Pyeatt, L.D., (2011) *"Reinforcement learning: a novel method for optimal control of propofol-induced hypnosis"*, Anesthesia & Analgesia, 112, 2, 360-367, United Stated.

[11] Rand, W., Riolo, R., Holland, J.H., (2006) *"The effect of crossover on the behavior of the GA in dynamic enviroments: A case study using the shaky ladder hyperplane-defined functions"*. Genetic and Evolutionary Computation Conference. 2, 1289-1296.

[12] Reboso, J.A., Méndez, J.A., Reboso, H.J., León, A.M., (2012) *"Design and implementation of a closed-loop control system for infusion of propofol guided by bispectral index (BIS)"*. Acta Anaesthesiol. Scand, 56, 8, 1032-1041.

[13] S. Torres, L. Acosta, J.A. Méndez. (1999) *"Análisis comparativo de dos estrategias para el control dinámico de robots: control PID frente a control difuso"*. Memoria de Licenciatura. Departamento de Física Fundamental y Experimental, Centro Superior de Informática, Universidad de La Laguna.

[14] Sawaguchi, Y., Furutani, E., Shirakami, G., Araki, M., Fukuda, K., (2008) *"A model-predictive hipnosis control system under total intravenous anesthesia"*, IEEE Transactions on Biomedical Engineering, 55, 3, 874-887

[15] Shieh, J.S., Kao. M.H., Liu, C.C., (2006) *"Genetic fuzzy modellong and control of bispectral index (BIS) for general intravenous anaesthesia"*. Medical Engineering and Physics. 28, 2, 134-148.

[16] Schnider, T., Minto, C., (2008) *"Pharmacokinetic models of propofol for TCI"*, Anaesthesia, 36, 2, 206

CONTROL AUTÓNOMO DEL SEGUIMIENTO DE TRAYECTORIAS DE UN VEHÍCULO CUATRIRROTOR

José Manuel Tur Ortega
jmturortega@gmail.com

Matilde Santos Peñas
msantos@ucm.es

Resumen

El presente trabajo se ha realizado como respuesta al Benchmark 2011-2012 propuesto por el grupo de Ingeniería de Control, del Comité Español de Automática. El objeto del mismo era la obtención de un modelo y el control de un vehículo aéreo no tripulado (UAV), en concreto un cuatrirotor. Esta contribución se centra en la parte del control. Los datos de partida, proporcionados por la organización, fueron el resultado de la simulación de un modelo no lineal. A partir de ellos se han identificado varios modelos que representan su dinámica. Se ha seleccionado uno de ellos para diseñar e implementar un controlador para el seguimiento de trayectorias del UAV. La misma organización ha suministrado unos scripts que han permitido evaluar tanto el modelo como el controlador, obteniéndose resultados muy satisfactorios para una de las propuestas.

Palabras clave: Vehículo aéreo no tripulado (UAV), cuatrirrotor, identificación, control de trayectorias, control multivariable, control fuzzy.

1. INTRODUCCIÓN

El presente trabajo se ha realizado como respuesta al Benchmark 2011-2012 propuesto por el grupo de Ingeniería de Control, del Comité Español de Automática. El objeto del mismo era la obtención de un modelo y el control de un vehículo aéreo no tripulado (UAV), en concreto un cuatrirrotor. Esta contribución se centra en la parte del control.

El control de vehículos aéreos no tripulados (UAV - *Unmanned Aerial Vehicles*) resulta de gran interés en el campo de la ingeniería de control [2]. Se trata de sistemas de 6 grados de libertad, inmersos en un entorno que puede ser cambiante, sometido a perturbaciones indeseadas. Lo que se persigue con el control de estos sistemas suele ser optimizar el seguimiento de algún objetivo o simplemente de trayectorias de forma fiable, sin desviarse de la ruta especificada y a la máxima velocidad posible. Como se ve, es un tema muy adecuado para un Concurso de Ingeniería de Control.

Se han probado diferentes estrategias de control para estos vehículos [2], [7]: control lineal, linealización por realimentación, control inteligente, control predictivo, por modos deslizantes, control robusto, algoritmos de aprendizaje, algoritmos genéticos, algoritmos adaptativos, control fuzzy, control por redes neuronales, y por Lyapunov. En este trabajo se presentan dos aproximaciones, una basada en lógica fuzzy y otra original deducida del comportamiento no lineal del sistema objeto de estudio, un cuatrirrotor de Parrot [1].

El presente documento está organizado en 7 secciones. En la sección 2 se describe el sistema de estudio, un cuatrirotor comercial. La sección 3 muestra brevemente la fase de identificación y enumera las técnicas utilizadas para la obtención de modelos del UAV. En la sección 4 se discute la estrategia de control planteado, y su implementación con control fuzzy (sección 5) y con un control lineal a tramos (sección 6). Por último, en la sección 7 se presentan las conclusiones.

2. DESCRIPCIÓN DEL SISTEMA DE ESTUDIO

Se trabaja con un vehículo AR.Drone, de Parrot [1] [6]. Se trata de un cuadricóptero hecho de fibra de carbono y plástico PA66 de gran resistencia. Se controla por medio de una conexión Wifi b/g, con un aparato bajo el sistema operativo *iOS* o *Android*. El Ar.Drone está equipado con cuatro motores tipo brushless de alta eficiencia, alimentados por una batería de litio de 11.1 V y 1000 mAh que le confiere una autonomía de vuelo de aproximadamente 12 minutos. Dispone de una carcasa para proteger las hélices en vuelos interiores. Su masa es de 380/420 g (sin/con carcasa).

La aproximación que se hace en [3] considera el seguimiento de trayectorias 2D, en un plano que supone que la altura z permanece constante. La parte de despegue y aterrizaje del vehículo, que se pueden realizar de forma automática, quedan fuera del alcance de este estudio. Además el propio cuatrirrotor cuenta con sistemas internos de control que prácticamente desacoplan las velocidades según los ejes x, y, z, solidarios al vehículo. El ob-

jeto del concurso consiste en modelar y controlar este cuatrirrotor para el seguimiento de trayectorias en el plano $x-y$.

La organización ha suministrado los datos de un ensayo realizado sobre su propio modelo no lineal, de manera que con las variables manipuladas Ref_pitch, Ref_roll, se debe reproducir de la mejor manera posible las variables controladas, que son las posiciones y velocidades en el plano $x-y$ (x, y, v_x, v_y). Finalmente, se debe desarrolar un controlador para optimizar el seguimiento de trayectorias sobre el plano $x-y$. Dicho control es un bloque que a partir de las entradas del siguiente punto a alcanzar (Ref_x, Ref_y), y con los valores instantáneos de (x, y, v_x, v_y), sea capaz de generar las consignas adecuadas de Ref_pitch, Ref_roll, que conduzcan al vehículo allí en el menor tiempo posible.

Para valorar el modelo y el controlador, la organización ha suministrado sendos scripts en Matlab© que valoran cuantitativamente ambos [4]. Esta valoración se realiza por medio del error medio y absoluto en las variables x, y para el modelo, y de las distancias máximas y medias, así como el tiempo del recorrido, para el controlador.

3. OBTENCIÓN DEL MODELO DEL UAV

Aunque este trabajo está focalizado en el control, se presenta en este apartado un resumen de la metodología aplicada para la obtención de modelos del UAV. Los datos del proporcionados por la organización se habían obtenido a partir de un modelo no lineal [3]. Además el vehículo, por medio de sus controladores internos, tiene prácticamente desacopladas las velocidades según los ejes del vehículo. Así que era de esperar que los mejores modelos fueran no lineales con escasa interacción entre las variables, por lo tanto monovariables.

Todo trabajo de identificación tiene tareas previas de filtrado de las señales para eliminar los ruidos superpuestos, así como la separación de los datos en dos conjuntos: uno que será usado para la estimación (aprendizaje) y otro de validación. El disponer de la posibilidad de solicitar un segundo ensayo a la organización con los valores de las variables deseadas facilitó esta última tarea.

3.1. Análisis cualitativo de los datos del ensayo

El grupo de datos inicial contiene dos zonas bien diferenciadas. Una compuesta por escalones alternos, sucesivos, crecientes, y no simultáneos, en las variables manipuladas. Al final hay un periodo de tiempo sometido a una señal pseudoaleatoria. Como el objetivo final del trabajo es diseñar un controlador para el seguimiento de trayectorias rectas entre distintos puntos, a priori parece que la estrategia de control debería generar señales escalonadas, más que de tipo pseudoaleatorio, lo que nos ha llevado a priorizar la parte escalonada ya que parece más adecuada para las condiciones de trabajo.

Un estudio previo de los datos nos permite observar la existencia de límites de saturación para los valores de las variables manipuladas Ref_pitch, Ref_roll. Para Ref_pitch este límite parece estar alrededor de 0,4. Para Ref_roll está alrededor de 0,6. Esto significa que para consignas superiores a estos valores no se aprecian cambios significativos en las velocidades del vehículo. Además, en contra de la primera impresión, resulta que el sistema no es simétrico. En realidad el vehículo no lo es, así que tiene lógica que la máxima velocidad según los ejes $x-y$ sean diferentes. En concreto los límites aproximados para las velocidades máximas son $v_x = 3$ m/s, y $v_y = 2{,}15$ m/s.

Por último, los datos obtenidos de los ensayos de la organización presentaban ruido aleatorio que distorsiona las medidas reales. Se ha realizado un filtrado de los mismos. Dado que el periodo de muestreo es de 0,06 s, tras varias pruebas se ha visto que el límite óptimo de frecuencia de filtrado está entre 2 y 3 rad/s.

3.2. Modelos obtenidos

La idea inicial era trabajar con la técnica ANFIS [5] para obtener un modelo de las velocidades del vehículo. Esto permitirá tratar la no linealidad del sistema y también trabajar con una o varias señales de entrada para cada salida, según convenga. De hecho con esta técnica se han obtenido diferentes modelos monovariables y multivariables para v_x y v_y. También se ha trabajado con los valores de las consignas sin modificar y truncados a partir de los límites de saturación comentados anteriormente (0,4 y 0,6, respectivamente).

En cualquier caso, la posibilidad de evaluar los distintos modelos por medio del script suministrado por la organización ha permitido comprobar que el modelo obtenido para v_x era muy mejorable. Esto nos ha llevado a probar otras estrategias de identificación que han proporcionado mejores resultados que con ANFIS, para ambas velocidades [8]. Se obtuvieron modelos con redes neuronales, en concreto se realizaron algunas tentativas (con redes *Levenberg-Marquardt backpropatation*), pero los resultados obtenidos ($d_{max}x = 11{,}29, d_{med}x = 2{,}55, d_{max}y = 8{,}71, d_{med}y = 5{,}98$) no invitaron a

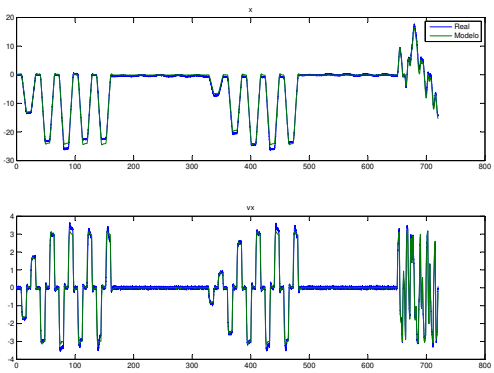


Figura 1: Comparativa del modelo no lineal de vx y los resultados experimentales

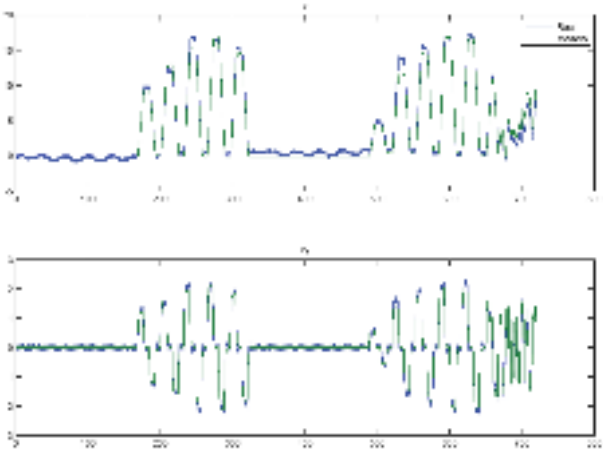


Figura 2: Comparativa del modelo no lineal de vy y los resultados experimentales

continuar por esta vía.

También se estimaron modelos lineales para ver qué resultados arrojan. En [6] se ha identificado un modelo de este tipo para el posterior diseño de un controlador y ha dado buenos resultados. De acuerdo a las consideraciones anteriores, se ha trabajado con los valores de la consigna del experimento, o bien con diferentes saturaciones. Incluso se ha llegado a probar un modelo con integrador partiendo de la pareja de datos (x,Ref_Pitch) y (y,Ref_Roll). La idea era que eliminando el integrador nos quedase un modelo de vx y vy. Pero los resultados eran mucho peores, así que se han descartado todos ellos.

Tras numerosas pruebas se han encontrados modelos no lineales tanto para vx como para vy que ofrecen los mejores resultados de todos los analizados. Al trabajar con modelos no lineales no es preciso establecer valores de saturación para las variables de entrada ya que esta restricción está bien resuelta por el modelo. Los resultados numéricos obtenidos en ambos casos son similares, tanto para la distancia máxima como para la distancia media. En las Figuras 1 y 2 se muestra su comportamiento y se puede observar como se ajustan bien a los datos reales.

4. ESTRATEGIA DE CONTROL

Como se ha indicado el objetivo del control es múltiple. Por un lado hay que llevar al vehículo en línea recta entre los puntos de la trayectoria, alejándose lo menos posible de la misma. Además el recorrido deberá realizarse en el menor tiempo posible, es decir, a la máxima velocidad. Es de esperar que circular a mayor velocidad desestabilice la trayectoria, ya que las perturbaciones sobre el cuatrirrotor lo alejarán más de la trayectoria ideal hasta que el controlador reconduzca la dirección. Es un problema de optimización multiobjetivo para el que habrá que encontrar una solución de compromiso.

Además se dan las circunstancias de que las velocidades máximas alcanzables según los distintos ejes son diferentes (según el eje x de 3 m/s y según el eje y de 2,15 m/s). Si se sigue una trayectoria formando un ángulo de unos 35,5º con el eje x la máxima velocidad posible será de unos 3,5 m/s. Eso significa que las trayectorias según el eje y serán siempre más lentas que sobre el eje x, y éstas más lentas que las que sigan una dirección con el ángulo indicado antes. Esto establece límites absolutos a los tiempos mínimos del recorrido, haciendo que la valoración obtenida con el script de la organización no sea directamente comparable para trayectorias diferentes ya que la velocidad tomada como límite en todos los casos es de 3 m/s.

Dado que existen dos obejtivos de control, ajuste a la trayectoria y máxima velocidad posible, se puede descomponer el problema de control en dos problemas menores:

1. Establecer la velocidad a la que debe circular el vehículo.

2. Darle la dirección apropiada.

4.1. Velocidad del vehículo

En principio la velocidad debería ser la máxima alcanzable para llegar al siguiente punto de la trayectoria lo antes posible. Esto sería lo ideal si sólo hubiera que recorrer un tramo recto. Pero como la trayectoira incluye varios tramos, al llegar al punto final de un tramo a máxima velocidad nos alejaremos de él antes de poder corregir la dirección, cometiendo así un error grande en la trayectoria.

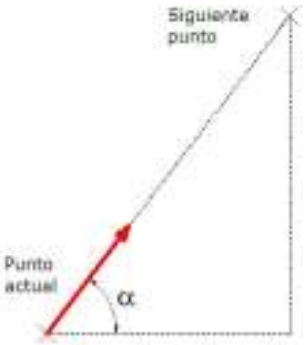


Figura 3: Control de la dirección del movimiento

Esto nos ha llevado a tener en cuenta la distancia al siguiente punto de la trayectoria. El nuevo criterio de control se podría enunciar como ir a *máxima* velocidad si estamos *lejos* del siguiente punto y avanzar de forma *lento* si estamos *cerca*. Este criterio necesita evaluar numéricamente el significado de *cerca* y *lejos*, así como de *lento*. Esto encaja perfectamente en la filosofía y aplicación del control *fuzzy*, por lo que ha sido la primera propuesta con la que se ha trabajado para desarrollar el controlador.

4.2. Dirección del vehículo

La determinación de la dirección del vehículo para llevarlo del punto actual al siguiente punto de la trayectoria es un sencillo problema de trigonometría. Se trata de calcular el ángulo α que marca la dirección que hay que seguir (ver Figura 3). A partir de ahí, la velocidad se debe descomponer en sus componentes horizontal y vertical:

$$v_x = v\cos\alpha, \tag{1}$$

$$v_y = v\sin\alpha. \tag{2}$$

La determinación del ángulo α también es muy fácil con la información disponible, simplemente tenemos en cuenta que es

$$\tan\alpha = \frac{error_y}{error_x}. \tag{3}$$

5. CONTROL FUZZY

La primera tentativa de control se ha basado en un sistema de inferencia borrosa (FIS) con las siguientes reglas:

1. Si la distancia al siguiente punto es *grande*, la velocidad debe ser *grande*.

2. Si la distancia al siguiente punto es *media*, la velocidad debe ser *media*.

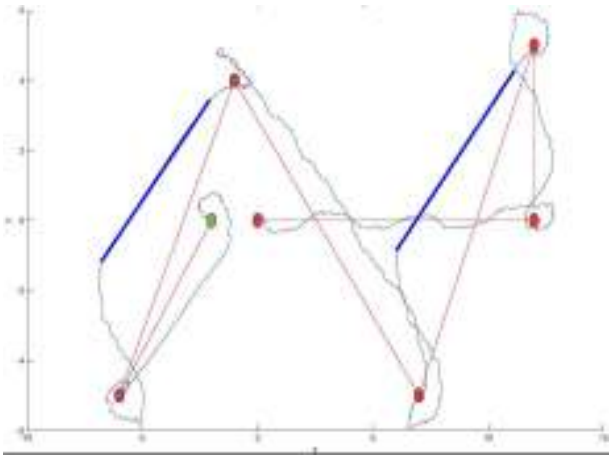


Figura 4: Desviación de la línea recta en trayectorias oblicuas

3. Si la distancia al siguiente punto es *pequeña*, la velocidad debe ser *pequeña*.

Partiendo de esta estructura básica se han diseñado diferentes controladores que, a partir de la distancia al siguiente punto, establecen una consigna para la velocidad entre 0 (nula) y 1 (máxima). Pero con este planteamiento aparece un problema debido a la asimetría de las velocidades según los ejes x-y. Como se ha dicho, la velocidad máxima según el eje x es mayor. Esto hace que cuando hay que ir a alta velocidad y se sigue un movimiento oblicuo, se recorra más rápidamente la distancia en x que la distancia en y. Este efecto se ve claramente en la Figura 4, donde el punto verde es el punto final de la trayectoria establecida (recordemos que es paramétrica y se puede ajustar). Los círculos rojos son los puntos de paso; tienen un radio de 0,2 m y cualquier punto dentro de su superficie se considera válido. Los distintos segmentos rectos en rojo representan la trayectoria ideal, que une dos puntos consecutivos en línea recta, minimizando la distancia entre ellos y, por tanto, el tiempo de vuelo. Finalmente las líneas azules indican la trayectoria del vehículo con el control fuzzy. En esta figura se ve que con ese controlador borroso se recorre oblicuamente un tramo hasta situarse aproximadamente en la línea vertical del siguiente punto, a partir del cual el movimiento es básicamente vertical.

Este efecto nocivo se puede reducir teniendo en cuenta la relación que hay entre las velocidades máximas:

$$\frac{\text{máx}(v_x)}{\text{máx}(v_y)} = \frac{3}{2{,}15} \approx 1{,}4 \tag{4}$$

Así, podemos decrementar la consigna de la velocidad x para que no supere a la de y, lo que mejora el funcionamiento del controlador como se puede

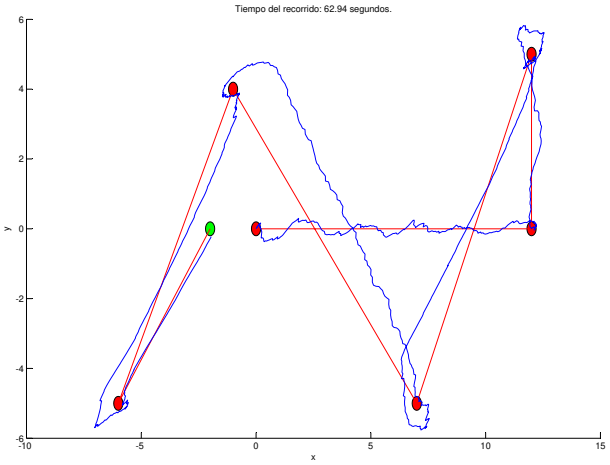


Figura 5: Trayectoria seguida con el control fuzzy

observar en la Figura 5. En este caso las trayectorias seguidas son más cercanas a las trayectorias ideales, aunque todavía se puede apreciar que existe posibilidad de ajustar algo más las líneas oblicuas. Como la mejora de este sistema podía ser laboriosa, se ha planteado una nueva propuesta.

6. CONTROL LINEAL A TRAMOS

Al observar que el control de las velocidades es muy importante se ha procedido a estudiar su perfil. En los datos proporcionado de los dos ensayos hay suficiente información para obtener las relaciones aproximadas entre las señales de control y las velocidades según cada eje, tal y como se puede observar en la Figura 6. Así se tiene que cada una de las velocidades tiene tres tramos.

Para la velocidad según el eje x se puede considerar que es máxima siempre que el valor de consigna supere 0,4. De 0 a 0,4 se han establecido dos tramos rectos de pendientes algo diferentes, ya que con un solo tramo no se obtenían resultados que se ajustaran bien a los datos del experimento.

La velocidad según el eje y se puede considerar máxima siempre que el valor de la consigna supere 0,6. Además, para valores de referencia entre 0,3 y 0,4 la velocidad obtenida es la misma. Finalmente, entre 0 y 0,3, y entre 0,4 y 0,6 se establecen dos tramos rectos que den continuidad al perfil.

Además las velocidades alcanzables funcionan de forma sencilla mientras $v_y < 2{,}15$ m/s. En el momento que sea precisa una velocidad superior a ésta según el eje y tendremos que hacer ajustes para reducir v_x, ya que v_y no puede aumentar más. Para solucionar esta restricción se decidió trabajar con tres velocidades prefijadas de antemano en vez de con una variación continua (de tipo *fuzzy*)

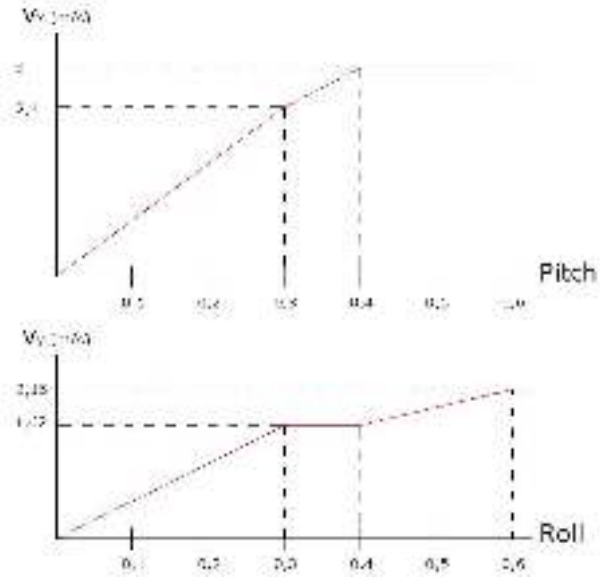


Figura 6: Curvas de relación de las velocidades con las consignas

sobre la velocidad el vehículo, aunque ajustándose al esquema de reglas anterior. Es decir, se aplican distintas velocidades por tramos:

1. Si la distancia al siguiente punto es $d > d_{max}$, la velocidad debe ser v_{max}.
2. Si la distancia al siguiente punto es $d_{min} < d < d_{max}$, la velocidad debe ser v_{med}.
3. Si la distancia al siguiente punto es $d < d_{min}$, la velocidad debe ser v_{min}.

Cuadro 1: Valores iniciales y finales del control de tipo lineal a tramos

Parámetro	Valor inicial	Valor ajustado
d_{min} (m)	1	1
d_{max} (m)	4	4
v_{min} (m/s)	0.5	0.4
v_{med} (m/s)	1.75	0.92

Los valores de partida para *lejos* (d_{max}) se han tomado teniendo en cuenta la máxima velocidad alcanzable (3,5 m/s) y que en algunas estimaciones se ha detectado la existencia de un retardo de unas 3 muestras (0,18 s); hay que tener en cuenta que en ese tiempo se recorren más de 60 cm a esa velocidad. Se ha considerado que un valor razonable para definir *lejos* en nuestro caso es una distancia mayor que 4 m. El segundo tramo de distancia está establecido entre 1 y 4 m. *Cerca* (d_{min}) son las distancias menores que 1 m. Después de muchas pruebas y comparando los resultados se

ha desechado el control tipo *fuzzy* por uno escalonado con velocidades de 0,92 m/s y 0,4 m/s para distancias intermedias y bajas, respectivamente (Tabla 1).

6.1. Señales de control

Finalmente quedan por establecer las señales de control que hay que generar en cada caso para que el vehículo se mueva a la velocidad deseada. Para ello, a partir de los datos de los ensayos, se han aproximado las relaciones $Pitch - v_x$ y $Pitch - v_y$ según las formas y rangos mostrados en la Figura 6.

Con todas estas consideraciones obtenemos las expresiones que nos dan las consignas para las variables manipuladas Ref_pitch y Ref_roll. Para distancias cercanas ($d < 1$ m) la velocidad, v, se ha fijado en 0,4 m/s y se obtiene con las consignas:

$$Ref_pitch = \frac{v}{A}\cos\alpha, \tag{5}$$

$$Ref_roll = \frac{v}{B}\sin\alpha. \tag{6}$$

Para distancias intermedias ($1 \leq d < 4$) la velocidad, c, se ha fijado en 0,92 m/s y se alcanza con consignas del tipo:

$$Ref_pitch = \frac{v}{C}\cos\alpha, \tag{7}$$

$$Ref_roll = (D \cdot v - E)sin\alpha. \tag{8}$$

Finalmente, cuando la distancia es grande ($d \geq 4$) tenemos que la velocidad tiene que ser máxima.

Eso divide el plano en 4 regiones. Limitándonos para la explicación, sin pérdida de generalidad, al primer cuadrante, resulta que si el ángulo del movimiento es superior al límite indicado anteriormente de 35,5º (o equivalentemente, $\tan\alpha > 0{,}71$) v_y puede ser máxima mientras que v_x tendrá que ajustarse para respetar la dirección del movimiento. Como el perfil de velocidad de v_x visto en la Figura 6 tiene dos tramos, esta región queda a su vez subdividad en dos tramos, cuyo límite viene establecido por $\tan\alpha = 0{,}89$. Así tenemos que para $0{,}71 < \tan\alpha < 0{,}89$ las consignas son de la forma:

$$Ref_pitch = \frac{F}{\tan\alpha} + G, \tag{9}$$

$$Ref_roll = 0{,}6, \tag{10}$$

y para $\tan\alpha \geq 0{,}89$ son:

$$Ref_pitch = \frac{H}{\tan\alpha}, \tag{11}$$

$$Ref_roll = 0{,}6, \tag{12}$$

Análogamente, cuando $\tan\alpha < 0{,}71$ la velocidad que hay que limitar es v_y, siendo v_x máxima. También tenemos dos tramos para v_y, tal como se ve en la Figura 6. Se llega de forma análoga a lo expresado anteriormente a que si $\tan\alpha \leq 0{,}54$ las ecuaciones son de la forma:

$$Ref_pitch = 0{,}4, \tag{13}$$

$$Ref_roll = I\tan\alpha, \tag{14}$$

y para $0{,}54 < \tan\alpha < 0{,}71$ son:

$$Ref_pitch = 0{,}4, \tag{15}$$

$$Ref_roll = J\tan\alpha - K. \tag{16}$$

Los valores de los parámetros A-K según las consideraciones anteriores se listan en la segunda columna del Cuadro 2. Se han ajustado empíricamente para intentar optimizar los resultados del control, por medio de la herramienta de valoración disponible, obteniendo los valores de la tercera columna de la tabla. La herramienta de evaluación permitía definir rutas mediante los puntos a seguir en la trayectoria deseada. Esto ha permitido establecer trayectorias que quedan dentro de cada uno de los rangos angulares deseados y así poder calcular los parámetros correspondientes. En general los ajustes han sido pequeños, así que a priori parece que el criterio utilizado para su obtención teórica no estaba desencaminado.

Cuadro 2: Valores de los parámetros del control de velocidad

Parámetros	**Valor teórico**	**Valor ajustado**
A	8	7.75
B	5.4	5.4
C	8	7.75
D	0.3846	0.3846
E	0.223	0.23
F	0.214	0.169
G	0.06	0.05
H	0.2675	0.22
I	0.555	0.6
J	1.154	1.35
K	0.223	0.2

Por último, se presentan las consignas generadas (ver Figura 8) por el controlador. En este gráfico

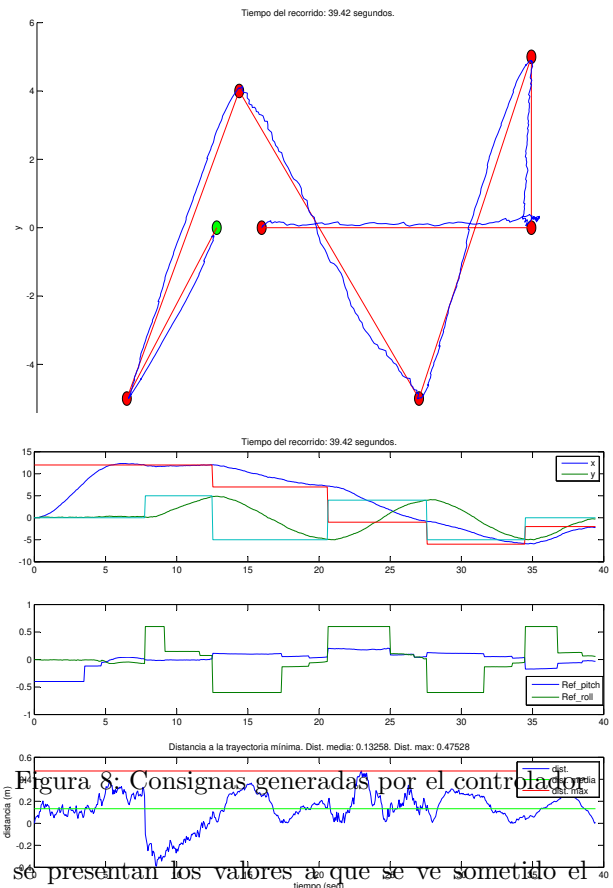


Figura 8: Consignas generadas por el controlador

se presentan los valores a que se ve sometido el vehículo (o su modelo) para llevarlo de un punto de la trayectoria al siguiente. En la fase de identificación se supuso que se trabajaría con consignas de tipo escalonado para ir siempre a la máxima velocidad posible, ya que al realizar trayectorias rectas se esperaba dicho comportamiento. Esta gráfica permite validar dicha hipótesis.

7. RESULTADOS Y CONCLUSIONES

La evaluación del controlador diseñado está ligada a la bondad del modelo obtenido. La valoración del modelo seleccionado con los datos iniciales disponibles daba resultados Deseables para las distancias medias y Tolerables para las máximas. En cualquier caso, aunque la precisión no sea la mejor, sí se puede considerar que el modelo describe cualitativamente bien la respuesta del sistema.

Esto nos lleva a la evaluación del controlador. Como se ha indicado anteriormente la valoración del tiempo de recorrido no es independiente de éste. Sin embargo las valoraciones que tienen que ver con la distancia máxima y media son más independientes, aunque siempre algo correlacionadas debido a la influencia de la velocidad en la precisión. Probando diferentes trayectorias extremas se obtienen valores que están en los rangos Tolerable-Deseable para las distancias media y máxima, y entre Tolerable-Altamente Deseable para el tiempo de recorrido, obteniendo este último cuando las trayectorias son zigzagueos cercanos al ángulo en que se alcanza la máxima velocidad.

Los resultados publicados de la primera fase correspondientes a los seis mejores equipos se muestran en el Cuadro 3. En él aparece en sexto lugar el equipo ISC-Master 1, representando a la Universidad Complutense de Madrid, que se corresponde con el trabajo que se expone aquí. De dichos resultados se pueden extraer las siguientes conclusiones:

Cuadro 3: Resultados de la primera fase del Benchmark 2011-2012 del CEA

EQUIPO	GPP Modelado	GPP Control
ETSII-UPM	5.63	17.33
CVG-UPM	41.44	6.51
Teleco. Sist. - UAB	42.73	21.42
IMPULSA	125.08	49.11
VIGO-1	216.34	61.95
ISC-Master 1	469.73	33.05

1. El modelo identificado no era cuantitativamente bueno. Ya se preveía con los resultados obtenidos antes del concurso, pero aquí se ha confirmado.

2. El modelo era cualitativamente bueno y, basado en él, ha sido posible realizar un control razonablemente bueno.

En conclusión, aunque el controlador se ha ajustado con el modelo obtenido (que era muy mejorable), la forma paramétrica debería permitir un ajuste sencillo de los parámetros en los parámetros para obtener un control más que satisfactorio.

Referencias

[1] Ardrone (2013) www.ardrone.parrot.com.

[2] Béjar, M. Ollero, A. (2008). Modelado y control de helicópteros autónomos. Revisión del estado de la técnica. Revista Iberoamericana de Automática e Informática Industrial, 5, 4, 5-16.

[3] Blasco, F.X. García-Nieto, S. Reynoso-Meza, G. Concurso en Ingeniería de Control 2012. Comité Español de Automática. http://www.ceautomatica.es/og/ingenieria-de-control/es/concurso-ingenieria-control-2012

[4] Blasco, F.X. García-Nieto, S. Reynoso-Meza, G. (2012). Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor. Simulación y evaluación de propuestas. Revista Iberoamericana de Automática e Informática Industrial, 9, 2, 194-199.

[5] Jang, J. S. (1993). ANFIS: Adaptive-network-based fuzzy inference system. Systems, Man and Cybernetics, IEEE Transactions on, 23(3), 665-685.

[6] Krajník, T., Vonásek, V., Fier, D., Faigl, J. (2011). AR-drone as a platform for robotic research and education. In Research and Education in Robotics-EUROBOT 2011 (pp. 172-186). Springer Berlin Heidelberg.

[7] Santos, M. (2011). Un enfoque aplicado del control inteligente. Revista Iberoamericana de Automática e Informática Industrial RIAI, 8(4), 283-296.

[8] Tur Ortega, J.M. (2012) Control del seguimiento de trayectorias de un vehículo autónomo cuatrirrotor. TFM, UNED-UCM.

MODELADO DE UN VEHÍCULO AEREO NO TRIPULADO MEDIANTE APLICACIÓN CONJUNTA DE TÉCNICAS PARAMÉTRICAS Y NEURONALES

J. Enrique Sierra
Universidad de Burgos, jesiegar@gmail.com

Matilde Santos
Universidad Complutense de Madrid, msantos@ucm.es

Resumen

A partir de datos de entrada/salida reales de un vehículo aéreo no tripulado (UAV) se han generado diferentes modelos mediante distintas técnicas de identificación de sistemas, incluyendo las que proporciona el soft computing. El dispositivo que se ha utilizado es un cuatrirrotor, por lo tanto un sistema complejo, no lineal, multivariable y con acoplamientos. Se han analizado y comparado los resultados de la aplicación individual y conjunta de técnicas paramétricas y de estrategias inteligentes para la obtención de modelos.

Palabras Clave: Soft computing, redes neuronales, técnicas paramétricas, modelización, identificación, vehículos aéreos no tripulados (UAV).

1 INTRODUCCIÓN

La identificación de sistemas es una de las aproximaciones más importantes para obtener modelos útiles de sistemas complejos. Fundamentalmente existen dos vías para alcanzar este objetivo: la primera de ellas consiste en aplicar las ecuaciones físicas conocidas que gobiernan el comportamiento dinámico, y emplear este conocimiento para construir las ecuaciones diferenciales que lo representan. Esta metodología resulta muy útil cuando las variables son perfectamente conocidas y las relaciones entre ellas no son demasiado complejas. La otra vía consiste en la medida y el tratamiento de las señales a la entrada y la salida del sistema, y la aplicación de diferentes mecanismos de mapeado para describir las relaciones internas subyacentes entre ellas. En esta categoría se encuentran, entre otras, algunas de las técnicas proporcionadas por la Inteligencia Artificial, como son las redes neuronales, la lógica borrosa y los algoritmos genéticos [4].

La modelización de un vehículo aéreo no tripulado, UAV (*Unmanned Aerial Vehicle*), no es una tarea sencilla. La complejidad proviene de diferentes fuentes: la aleatoriedad de los flujos de aire y de las fuerzas exógenas, la no homogeneidad de su estructura mecánica interna, y la fuerte no linealidad de la dinámica del mismo. Todos estos factores convierten a las técnicas basadas en la inteligencia artificial en un camino idóneo para identificar estos sistemas.

Otros trabajos que han aplicado estrategias inteligentes para modelar estos sistemas, en concreto las redes neuronales, son [7] y [8]. También se pueden encontrar ejemplos de la aplicación de estos métodos heurísticos para identificar otros sistemas complejos no lineales como, por ejemplo, vehículos marinos [9].

En este trabajo se realizan dos aportaciones novedosas al estudio de los UAVs: la aplicación combinada de técnicas paramétricas y neuronales como metodología de identificación, y la utilización del error cuadrático medio de la respuesta al escalón, estimada mediante análisis de correlación, y del error cuadrático medio de la respuesta estimada en frecuencia, como dos herramientas de evaluación y validación de la eficacia del modelo.

Empleando datos reales de entrada/salida de un UAV, proporcionados por el grupo de Ingeniería de Control de CEA (Comité Español de Automática) [2], se han obtenido diferentes modelos basados en redes neuronales. También se han utilizado métodos tradicionales de identificación paramétrica y los resultados se han comparado. Por último, se ha aprovechado la sinergia entre ambos métodos, combinándolos para mejorar los modelos obtenidos individualmente.

El vehículo a modelar es el ARDRONE 2.0 comercializado por Parrot (Figura 1). Este aparato se fabrica en fibra de carbono y plástico PA66. Está equipado con cuatro motores tipo *brushless* de alta eficiencia, alimentados por una batería de litio de

11.1 V y 1000 mAh que le confiere una autonomía de vuelo de aproximadamente 12 minutos. La masa es de 380 g sin carcasa y 420 g con carcasa. Dispone de un acelerómetro de 3 ejes, un giróscopo de 2 ejes (balanceo y cabeceo), y un giróscopo de precisión para el ángulo de guiñada [6].



Fig. 1: Vehículo cuatrirrotor utilizado [6].

A continuación se detalla la organización del artículo. En la sección 2 se desarrolla un análisis preliminar describiendo las señales examinadas, el análisis de correlación y el análisis espectral. La sección 3 se destina al estudio de los diferentes modelos contrastados: los basados en técnicas paramétricas, los que utilizan redes neuronales, y la aplicación conjunta de ambas metodologías. Los resultados son discutidos y comparados en la sección 4. El documento finaliza con las conclusiones.

2 ANÁLISIS PRELIMINAR

2.1 SEÑALES EMPLEADAS

Para realizar el modelado se parte de un conjunto de datos suministrados por el Grupo de Ingeniería de Control de CEA para el concurso de Ingeniería de Control 2013 [2]. El conjunto recoge los siguientes datos: periodo de muestreo, señales *pitch*, *roll*, posición (*x*,*y*), y velocidades en los dos ejes, v_x e v_y, obtenidas durante 9 minutos con un tiempo de muestreo de 60 ms.

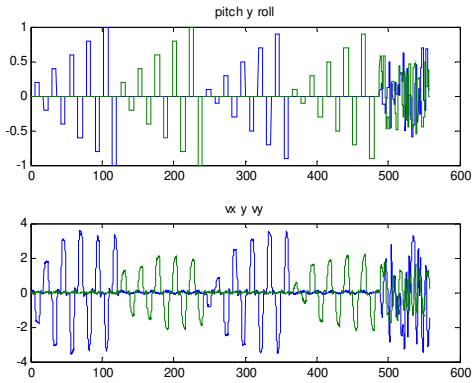


Fig. 2: Señales empleadas.

Las señales de entrada y salida se representan en la Figura 2. En azul v_x y *pitch*, y en verde v_y y *roll*. Puede adelantarse ya el importante desacoplo entre las variables que se corroborará mediante la correlación

Nos centraremos en el modelado de las velocidades v_x y v_y, puesto que la posición (*x*,*y*) viene determinada por la integración de esas velocidades.

2.2 ANÁLISIS DE CORRELACION

A partir de la estima de la covarianza entre la entrada y la salida de un sistema, y la estima de la varianza de la entrada, se puede estimar la respuesta al impulso con las expresiones siguientes:

$$\hat{R}_{yu}(\tau) = \frac{1}{N}\sum y(t)u(t-\tau) \qquad (1)$$

$$\hat{\lambda} = \frac{1}{N}\sum u(t)^2 \qquad (2)$$

$$g = \frac{\hat{R}_{yu}(\tau)}{\hat{\lambda}} \qquad (3)$$

Siendo *y* la salida del sistema, *u* la entrada, *N* el número de datos, *R* la covarianza, λ la varianza, y *g* la respuesta estimada al impulso. Integrando la respuesta al impulso puede obtenerse la respuesta al escalón, la cual puede ser muy útil para ayudarnos a estimar el grado de acoplamiento entre las variables del sistema. Esta metodología de estudio suele denominarse análisis de correlación [1].

En la respuesta al escalón de la matriz de funciones de transferencia de la Figura 3 se aprecia como la ganancia en estado estacionario de los elementos de la diagonal principal, (1,1) y (2,2), son bastante mayores que el resto. Esto indica que las dinámicas del sistema están parcialmente desacopladas, pudiendo convertir el sistema MIMO (Multi-Input Multi-Output) en dos sistemas SISO (Single-Input Single-Output).

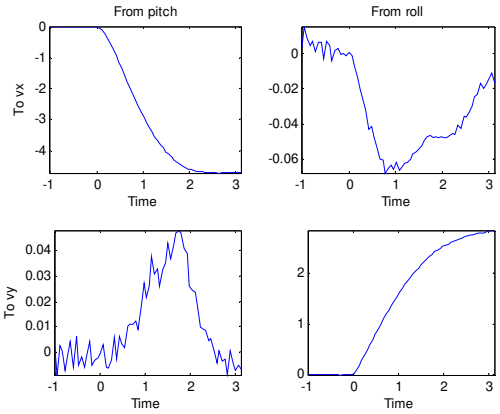


Fig. 3. Respuesta estimada a un escalón del sistema MIMO

Aprovechando esta simplificación, los esfuerzos en modelado en este trabajo se centrarán en la influencia del *pitch* sobre la velocidad en el eje *x*, y la del *roll* sobre la velocidad en el eje *y*.

2.3 ANÁLISIS ESPECTRAL

Una vez calculada la estima de la covarianza, ésta puede emplearse para calcular la respuesta estimada en frecuencia aplicando para ello la transformada de Fourier ponderada por una función de enventanado [1], como se describe en las expresiones siguientes:

$$\widehat{\Phi}_{yu}(\omega) = \sum w_M \, \hat{R}_{yu}(\tau) e^{-i\omega\tau} \qquad (4)$$

$$\widehat{\Phi}_{u}(\omega) = \sum w_M \, \hat{R}_{u}(\tau) e^{-i\omega\tau} \qquad (5)$$

$$G(e^{iwT}) = \frac{\widehat{\Phi}_{yu}(\omega)}{\widehat{\Phi}_{u}(\omega)} \qquad (6)$$

3 MODELOS DEL UAV OBTENIDOS

3.1 BASADOS EN TÉCNICAS PARAMÉTRICAS

Se han probado diferentes modelos paramétricos y se ha escogido, por su sencillez y efectividad, el modelo paramétrico ARX. Los modelos ARX (A*utoregressive Exogenous Model*), descritos en [3], representan el sistema con una ecuación de la siguiente forma:

$$Ay(t) = Bu(t) + e \qquad (7)$$

Donde *y* es la salida, *u* la entrada, *e* el error de la estima, y *A* y *B* son polinomios de ajuste. Si se restringe el grado máximo de los polinomios a 30, el mejor resultado conseguido es con la configuración ARX (10, 10, 1), donde 10 es el grado de *A* y de *B*, y 1 son las muestras de retardo.

3.2 BASADOS EN REDES NEURONALES

Se han probado distintos tipos de redes neuronales y se han escogido las redes de base radial por su aptitud para la aproximación de funciones [4, 5]. Las redes de base radial están formadas por tres capas: una de entrada donde se reciben las señales del exterior; una capa oculta, conectada con la capa de entrada donde cada elemento tiene asociada una función de base radial; y una capa de salida donde se combinan linealmente los resultados de las funciones de activación de la capa oculta [5].

La salida de cada elemento de la capa oculta, $z_i(n)$, se calcula como la distancia entre el patrón de entrada al centro del *cluster* ponderada por d_i, y aplicando este valor a una función de base radial, según la siguiente expresión:

$$z_i(n) = \Phi\left(\frac{\sqrt{\sum(x_j(n) - c_{ij})^2}}{d_i}\right) \qquad (8)$$

$$y_k(n) = \sum w_{ik} z_i(n) + \mu_k \qquad (9)$$

Siendo w_{ik} los pesos de la capa de salida y μ_k los umbrales de activación. La función de base radial Φ utilizada es:

$$\Phi(n) = e^{-n^2} \qquad (10)$$

Para realizar la identificación del modelo el conjunto de datos de entrada de la red se ha formado con la señal de entrada en el instante actual, *pitch(t)*, y valores en los 10 instantes de muestreo anteriores, *pitch(t-iT$_s$)* donde *i* = 1…10, y *Ts* es tiempo de muestreo. El conjunto de datos objetivo se ha generado con los valores de la señal de salida en el instante actual $v_x(t)$. Un proceso similar se ha llevado a cabo para la relación *roll* → v_y. Para simplificar el modelo y establecer las comparaciones en términos justos se ha limitado el número de neuronas de la capa oculta a 50.

Se han empleado 5000 muestras de las señales, el 60% para entrenar las redes y el resto para su simulación y validación.

3.3 APLICACIÓN COMBINADA DE TÉCNICAS PARAMÉTRICAS Y NEURONALES

La combinación de las técnicas anteriores empleando la topología y la metodología adecuadas proporciona mejores resultados en términos de complejidad computacional y del error cometido en el modelo. En este estudio se han empleado dos topologías de combinación diferentes: la conexión en cascada y la conexión en paralelo. En las figuras 4 y 5 se representan ambas configuraciones.

En el primer caso, configuración en cascada, el entrenamiento de la red se realiza de la forma siguiente:

- Se realiza la identificación del modelo paramétrico que mejor aproxime a las señales.
- Se simula la salida esperada del sistema a partir de un conjunto de señales de entrada conocido.
- Se entrena la red neuronal tomando como conjunto de entrada el resultado de la simulación anterior, y como conjunto de salida los datos de salida reales conocidos.

Para la conexión en cascada se substituye el tercer paso por:

- Se entrena la red neuronal tomando como conjunto de entrada el conjunto de señales de entrada conocido del paso 2, y como conjunto de salida la señal de error entre la salida esperada del paso 2 y los datos de salida reales conocidos del paso 3.

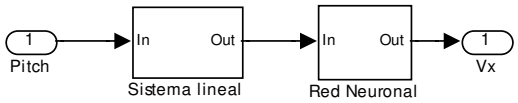


Fig. 4. Conexión en cascada de las dos técnicas de modelado.

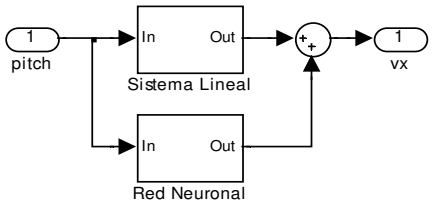


Fig. 5. Conexión en paralelo de las dos técnicas de modelado.

Para que las comparaciones entre los modelos de las secciones 3.1 y 3.2 y la técnica combinada sean equiparables, se escoge como modelo lineal el ARX-10,10,1 y como red neuronal la red de base radial con 30 neuronas en la capa oculta.

4 RESULTADOS

En esta sección se presenta una comparación de los modelos obtenidos por las diferentes técnicas expuestas. Todas las estrategias se han implementado en Matlab/Simulink. Los modelos se evalúan tanto en el dominio del tiempo como en el de la frecuencia. Se define el error como la diferencia entre la salida real del sistema (datos experimentales proporcionados por los ensayos de la organización) y la salida del modelo correspondiente. Esta diferencia también se calcula para la respuesta al escalón y para la respuesta estimada en frecuencia.

Las figuras 6 y 7 muestran la respuesta del sistema (velocidad en el eje *x*, figura 6, y velocidad en el eje *y*, figura 7) cuando se aplican diferentes escalones como señal de referencia. Si ordenamos los resultados en base a la similitud con la salida real, en ambos casos la conexión en cascada es la que mejor comportamiento proporciona, seguida por la conexión en paralelo, base radial, y por último la identificación mediante ARX.

Por lo tanto la utilización combinada de técnicas paramétricas y redes neuronales da mejores resultados que la utilización individual de las mismas.

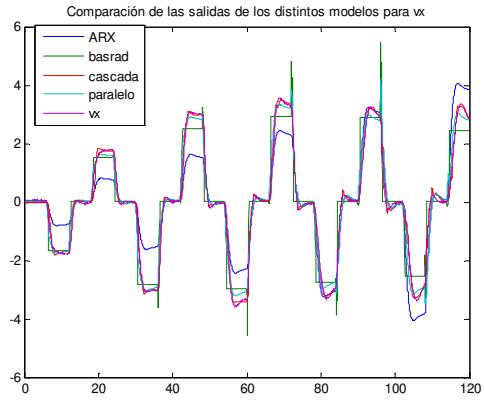


Fig. 6. Comparación de las salidas de los modelos para v_x.

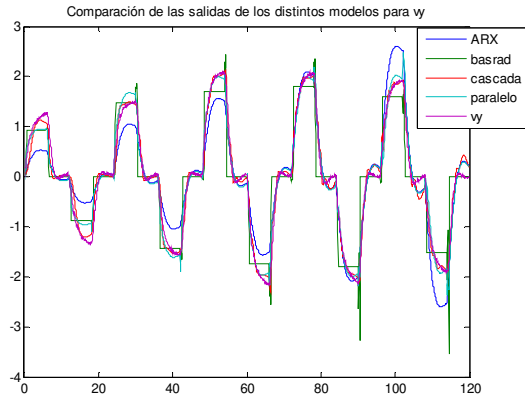


Fig. 7. Comparación de las salidas de los modelos para v_y.

Las respuestas al escalón se representan en las figuras 8 (velocidad en el eje x) y 9 (velocidad en el eje y). El mejor resultado para v_x es el proporcionado por la conexión en paralelo seguida por la conexión en cascada, ARX, y base radial. Para v_y es la conexión en cascada la que mejor se aproxima a la respuesta al escalón estimada, seguida por ARX, conexión en paralelo y base radial.

En este caso queda patente la asimetría del sistema, ya que se obtienen diferentes aproximaciones en una dirección que en otra, y unas técnicas identifican mejor según la variable, y por lo tanto el eje, del que se trate.

Las figuras 10 (velocidad en el eje x) y 11 (velocidad en el eje y) muestran la comparación de la respuesta en frecuencia estimada para los diferentes modelos estudiados. En ambos casos el modelo que más se acerca a la respuesta en frecuencia estimada mediante

los datos reales de entrada y salida es el ARX seguido por la conexión en cascada, la conexión en paralelo y base radial.

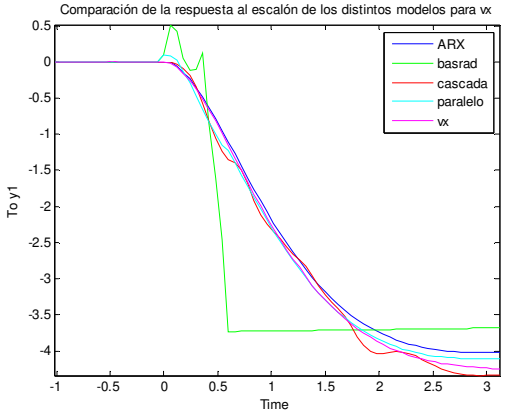


Fig. 8. Comparación de la respuesta al escalón de los modelos para v_x.

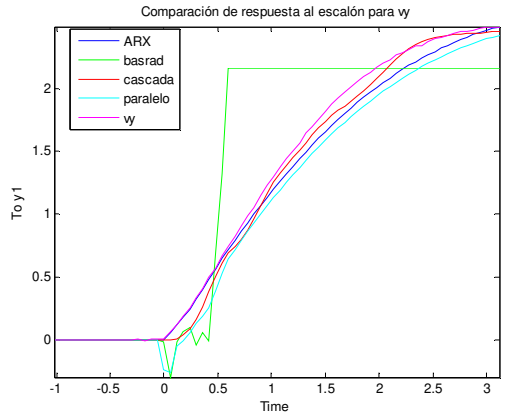


Fig. 9. Comparación de la respuesta al escalón de los modelos para v_y.

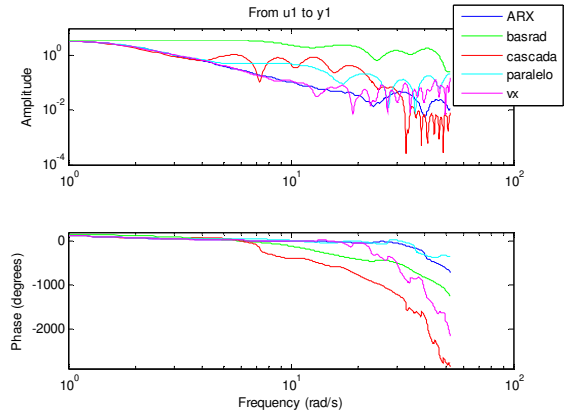


Fig. 10. Comparación de la respuesta en frecuencia de los modelos para v_x.

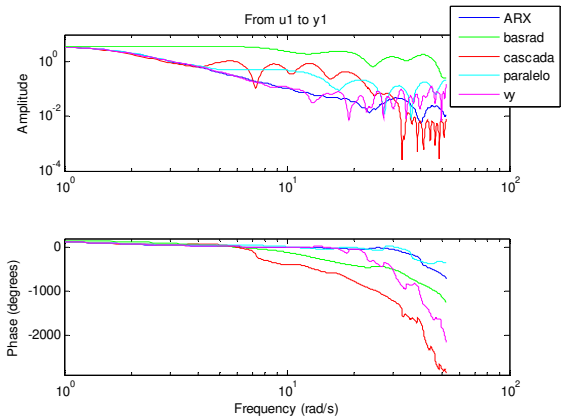


Fig. 11. Comparación de la respuesta en frecuencia de los modelos para v_y.

Por lo tanto en el dominio de la frecuencia se puede concluir que se obtienen mejores resultados con técnicas clásicas para esta aplicación con los datos disponibles.

La tabla 1 presenta los resultados cuantitativos en términos de error cuadrático medio, MSE (*Mean Squared Error*) para cada técnica estudiada.

Tabla 1: Comparación del MSE para valores en el dominio del tiempo, la respuesta al escalón estimada y la respuesta en frecuencia estimada.

Técnica	**Salida**	**MSE**	**MSE respuesta al escalón**	**MSE respuesta en frecuencia**
ARX10,10,1	Vx	0.5056	0.0122	0.0014
ARX10,10,1	Vy	0.1440	0.0085	0.0032
Base radial	Vx	0.3074	0.6680	3.2289
Base radial	Vy	0.1432	0.2271	1.4363
Cascada	Vx	0.0155	0.0080	0.0767
Cascada	Vy	0.0207	0.0048	0.0100
Paralelo	Vx	0.0815	0.0050	0.0252
Paralelo	Vy	0.0337	0.0269	0.0800

Los resultados muestran que las topologías en cascada y en paralelo proporcionan mejores resultados que la utilización independiente de la red de base radial, que es la que proporciona peores resultados tanto en el dominio del tiempo como para el dominio de la frecuencia.

La comparativa indica que en el caso del ARX, cuando se emplea la conexión en cascada, se consiguen mejores resultados en lo que respecta al MSE y a la respuesta al escalón. En el caso de la respuesta en frecuencia se ha observado que el modelo ARX presenta un mejor comportamiento que

el ofrecido por el resto de los modelos. Esta diferencia se habría visto reducida si se hubiera aumentado el número de neuronas de la capa oculta de la red de base radial.

5 CONCLUSIONES

Los UAVs son sistemas difíciles de modelar. Su complejidad deriva de la aleatoriedad de los flujos de aire y de las fuerzas exógenas, y de su dinámica fuertemente no lineal. Estos factores señalan a las técnicas inteligentes como estrategias eficientes para modelar estos sistemas. Para probarlo, en este trabajo se han implementado un conjunto de técnicas tanto analíticas como inteligentes que permiten identificar cualquier tipo de UAV en base a datos experimentales.

En concreto, para modelar el UAV objeto de estudio del que se cuenta con datos reales de ensayos, se han implementado y comparado técnicas paramétricas y modelos basados en redes neuronales. Se han aplicado tanto en el dominio del tiempo como en el dominio de la frecuencia.

Por otro lado, se han combinado las técnicas paramétricas con las inteligentes, en dos configuraciones sencillas, en serie o cascada y en paralelo.

Los resultados son altamente satisfactorios: el error MSE –que se ha usado como criterio para evaluar los modelos- obtenido mediante la aplicación conjunta de las técnicas paramétricas y neuronales indica que esto modelos proporcionan un comportamiento satisfactorio tanto para la secuencia temporal como para la respuesta en frecuencia.

En última instancia se ha contrastado la idoneidad de las técnicas basadas en redes neuronales para identificar y modelar estos sistemas, y se ha probado como la sinergia de técnicas clásicas e inteligentes puede ser una alternativa muy eficiente en problemas complejos.

AGRADECIMIENTOS

A los autores les gustaría agradecer al Grupo de Ingeniería de Control del Comité Español de Automática [2] la aportación del fichero con datos reales empleado para modelar el UAV.

REFERENCIAS

[1] Bendat, J. S., & Piersol, A. G. (1980). Engineering applications of correlation and spectral analysis. New York, Wiley-Interscience, 1980. 315 p.

[2] Grupo Ingeniería de Control (2013), Comité Español de Automática (CEA), www.ceautomatica.es/og/ingenieria-de-control.

[3] Isaksson, A. J. (1993). Identification of ARX-models subject to missing data. Automatic Control, IEEE Transactions on, 38(5), 813-819.

[4] Nørgård, P. M., Ravn, O., Poulsen, N. K., Hansen, L. K. (2000). Neural Networks for Modelling and Control of Dynamic Systems-A Practitioner's Handbook. Springer-London.

[5] Park, J., Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. Neural computation, 3(2), 246-257.

[6] Parrot. www.parrot.com (2013)

[7] San Martin, R., Barrientos, A., Gutierrez, P., Del Cerro, J. (2006). Neural networks training architecture for UAV modelling. In: IEEE World Automation Congress WAC'06, 1-6.

[8] San Martin, R., Barrientos, A., Gutierrez, P., Del Cerro, J. (2006). Unmanned aerial vehicle (UAV) modelling based on supervised neural networks. In: Proc. IEEE Int. Conference on Robotics and Automation ICRA 2006, 2497-2502.

[9] Santos, M., López, R., & De La Cruz, J. M. (2006). A neuro-fuzzy approach to fast ferry vertical motion modelling. Engineering Applications of Artificial Intelligence, 19(3), 313-321.

Comparación de técnicas de optimización multi-objetivo clásicas y estocásticas para el ajuste de controladores PI

Helem Sabina Sánchez*, Gilberto Reynozo-Meza†, Ramon Vilanova*, Xavier Blasco†
* Departament de Telecomunicació i Enginyeria de Sistemes, Universitat Autónoma de Barcelona, 08193 Bellaterra, Barcelona, España
{helemsabina.sanchez, ramon.vilanova}@uab.cat
†Departamento de Ingenierí de Sistemas y Automática, Universidad Politécnica de Valencia, Camino de Vera N°14, 46022 Valencia, España
{gilreyme@upv.es, xblasco@isa.upv.es

Resumen

Para abordar problemas de control desde un enfoque multiobjetivo compararemos una técnica clásica y una estocástica, el Normal Normalized Constraint (NNC) y el Multiobjective Differential Evolution Algorithm with Spherical Pruning (sp-MODE) ambos con una formulación basada en la optimización multiobjectivo en los cuales se analiza la solución de compromiso entre los diferentes índices de rendimiento.

1. Introducción

El controlador (PI) es el algoritmo de control más común. Su éxito se debe a su sencilla estructura y funcionamiento, esto hace que sea más fácil de utilizar e implementar en comparación con otras técnicas de control avanzada. Esta simplicidad es muy importante para la comunidad científica que muestra constante interés por nuevos métodos de sintonización que sean capaces de encontrar los parámetros óptimos [1].

En el diseño de controladores debe considerarse toda una serie de requerimientos e índices de desempeño para una implementación adecuada. Sin embargo cabe la posibilidad de que algunos de estos índices entren en conflicto entre sí, por lo que se debe buscar una solución de compromiso cuyas prestaciones se encuentren dentro de los requerimientos de operación. Desde un punto de vista de optimización, todos estos índices pueden ser utilizados para establecer un problema de optimización multiobjetivo, en donde se considera de forma individual a todos los objetivos involucrados, de manera que generan un conjunto de soluciones (frente de Pareto) todas ellas óptimas entre sí, diferentes únicamente por el grado de compromiso que ejercen ante los objetivos [2].

En este trabajo se presenta una comparación entre una técnica clásica y una estocástica, el NNC y el sp-MODE respectivamente. Utilizando como criterio de selección la solución de Nash presentada en [3]. El resto del artículo se organiza de la siguiente forma: Sección 2 presenta las funciones de transferencia del modelo del proceso de control y el controlador; Sección 3 se describe brevemente algunos conceptos relacionados con la optimización multi-objectivo, el método NNC y el sp-MODE; en la Sección 4 planteamiento del problema multiobjetivo; Sección 5 se presenta la comparación entre el NNC y el sp-MODE através de un ejemplo; Sección 6 finalmente se presentan las conclusiones.

2. Formulación del Problema

Considere el sistema de control en lazo cerrado en figura (1), donde $P(s)$ y $C(s)$ son el modelo del proceso controlado y la función de transferencia del controlador, respectivamente. En este sistema, $r(s)$ la referencia, $u(s)$ es la señal de salida del controlador, $d(s)$ la perturbación al sistema, $y(s)$ es la salida del proceso, $n(s)$ es la medición de ruido y $y'(s)$ es la señal de realimentación contaminada con el ruido. La señal de salida del sistema de control, $y(s)$ a un cambio en sus entradas, $r(s)$ y $d(s)$, está dada por:

$$y_s = M_{yr}(s)r(s) + M_{yd}(s)d(s) \tag{1}$$

donde

$$M_{yr}(s) = \frac{C_r(s)P(s)}{1 + C_y(s)P(s)}, \tag{2}$$

$$M_{yd}(s) = \frac{P(s)}{1 + C_y(s)P(s)}, \tag{3}$$

las cuales están relacionadas por

$$M_{yr}(s) = C_r(s)M_{yd}(s) \tag{4}$$

$M_{yr}(s)$ es la función de transferencia desde la referencia a la salida del proceso y, $M_{yd}(s)$ es la función de transferencia desde la perturbación del sistema a la salida del proceso.

2.1. Modelo del Proceso

El proceso a controlar $P(s)$ se supone generalmente bajo la forma de la función de transferencia de un modelo de primer orden con retardo:

$$P(s) = \frac{Ke^{-Ls}}{(Ts+1)} \tag{5}$$

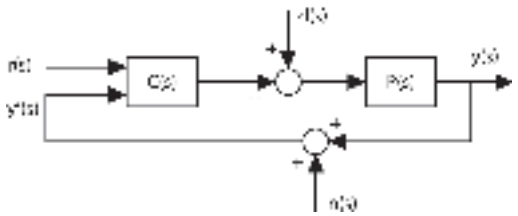


Figura 1: Sistema de Control en Lazo Cerrado.

donde K, T, L son el modelo de ganancia, constante de tiempo y de tiempo muerto, respectivamente. El tiempo muerto normalizado está dado por, $\tau_o = \frac{L}{T}$.

2.2. PI_2:Controlador Proporcional-integral 2-DoF

El proceso se controla con un controlador de dos grados de libertad (2-DoF) proporcional-integral (PI_2) [4], cuyas salidas son

$$u(t) = K_p \left\{ \beta r(t) - y(t) + \frac{1}{T_i} \int_0^\infty [r(\tau) - y(\tau)] \right\} \tag{6}$$

o

$$u(s) = K_p \left\{ \beta r(s) - y(s) + \frac{1}{T_i s} [r(s) - y(s)] \right\} \tag{7}$$

donde K_p es la ganancia proporcional del controlador, T_i la constante de tiempo integral, y β el peso de la señal de referencia. Las ecuaciones (6) y (7) por conveniencia matemática son reescritas de la siguiente forma:

$$u(s) = K_p \left\{ \beta + \frac{1}{T_i s} \right) r(s) - K_p \left(1 + \frac{1}{T_i s} \right) y(s) \tag{8}$$

en su forma compacta como

$$u(s) = C_r(s) r(s) - C_y(s) y(s) \tag{9}$$

donde

$$C_r(s) = K_p \left(\beta + \frac{1}{T_i s} \right), \tag{10}$$

$$C_y(s) = K_p \left(1 + \frac{1}{T_i s} \right) \tag{11}$$

$C_r(s)$ es la función de transferencia del controlador desde la referencia y $C_y(s)$ es la función de transferencia del controlador realimentado. Para nuestro caso, solo nos ocuparemos de la perturbación de carga por lo cual asumiremos que $\beta = 1$, por lo tanto $C_r(s) = C_y(s)$.

2.3. Requerimientos del Sistema de Control

Si nos concentramos en la atenuación de la perturbación como un interés primordial en el proceso de control, el rendimiento de control se caracteriza usualmente por la integral del Error Absoluto (IAE) o integral del Error (IE):

$$J_{IAE} = \int_0^\infty |e(t)| dt \qquad J_{IE} = \int_0^\infty e(t) dt \tag{12}$$

donde e es el error del sistema cuando se encuentra con una perturbación en forma de escalón unitario. Estos son reconocidos como buenas medidas de rendimiento para sistemas de control con acción integral. Aún más, si el sistema está bien amortiguado ambos criterios generan aproximadamente el mismo valor, siendo la integral del error igual al inverso de la integral del tiempo ($IE = 1/T_i$).

Aunque el objetivo principal sea optimizar el rendimiento, no debe dejarse de lado el análisis de las características dinámicas del esfuerzo de control, ya que es conveniente, para evitar el deterioro prematuro del elemento actuador, evitando que los cambios del mismo no sean bruscos ni extremos. La evaluación de las variaciones del esfuerzo de control, se puede realizar mediante un índice de su variación total, definido por:

$$TV_u = \sum_{k=1}^{\infty} |u(k+1) - u(k)| \tag{13}$$

el cual debe ser lo menor posible. Este es una indicación de la suavidad de la señal de control. A veces, también para evitar grandes variaciones en la señal de control originada por el ruido, esto puede ser abordado a través de un filtro de ruido ($G_f(s)$) que entra en el camino de realimentación como un filtro de medición. Una medida plausible es la ganancia de alta frecuencia de la combinación del controlador y filtro de medición. Una opción aconsejable [5] es el uso de un filtro de segundo orden

$$G_f(s) = \frac{1}{s^2 T_f^2/2 + s T_f + 1} \tag{14}$$

Para los sistemas de control basados en PI y P esta ganancia de alta frecuencia está determinada esencialmente por la ganancia del controlador y el tiempo constante de filtro T_f. Para un sistema de control basado en PID, la acción derivativa también entrará en juego. Una discusión detallada de los efectos del ruido de medición se puede encontrar en [1] o [6].
La robustez del sistema (estabilidad relativa) la máxima sensibilidad del sistema está definido por

$$M_s = \max_w \left| \frac{1}{1 + C_y(jw) P(jw)} \right| \tag{15}$$

donde C_y esta definido en la ecuación (11), P es el proceso controlado definido en la ecuación (5). La

robustez en la ecuación (15) suele abordarse como una restricción de desigualdad. En [3] se realizaron experimentos donde se calcularon frentes de pareto usando como restricciones tres niveles de robustez ($M_s = 1{,}8, M_s = 1{,}6, M_s = 1{,}4$) para compararlos con un frente de pareto libre de restricciones. Y se observo la relación entre el M_s y TV, a mayor rubustez ($M_s = 1{,}4$) menor es el TV. Como se señala en [7] sí existe una correlación entre la robustez y el esfuerzo de control. Por lo tanto, podemos asociar directamente la robustez al índice de rendimiento TV y pensar en este índice no sólo como uso de entrada. Sino también como una medida de robustez del sistema. Esto nos permite pensar en el problema de optimización multiobjectivo simplemente como un compromiso entre TV y J_{IAE}, lo cual nos facilita la generación del frente.

3. Optimización Multi-objectivo

El problema de optimización multi-objectivo consiste en minimizar m objetivos, todos ellos de igual importancia para el diseñador. Esto implica la existencia de un conjunto de soluciones, donde todas son óptimas y diferentes entre sí solo por el grado de compromiso en el espacio de objetivos [8]. Este conjunto de soluciones se conoce como el conjunto de Pareto y su imagen en el espacio de objetivos como **frente de Pareto**. Cada solución en el frente de Pareto es una solución no-dominada y pareto-óptima. Además, dicho frente de pareto es de ayuda para tener una idea preliminar del espacio de objectivos [9], esto es de gran ayuda cuando se desee explicar y justificar el proceso de selección. Este proceso de optimización es conocido como Optimización Multiobjetivo (MOO) y puede ser considerado una técnica genere primero-seleccione después (GFCL) [10].

Después del proceso de optimización, viene la fase final de toma de decisiones y selección multicriterio (MCDM) donde se analizan las soluciones obtenidas y se selecciona la más adecuadas según las preferencias del diseñador. El éxito de un MOO depende de la correcta definición del MOP y de la buena metodología o herramienta para el proceso de MCDM [11].

En los últimos años, los algoritmos evolutivos han sido empleados y popularizados para la optimización multiobjetivo [12]. En el problema que planteamos, se realizará una comparación entre un método a posteriori (GFCL), Normal Normalized Constraint (NNC)[1] y el Multiobjective Differential Evolution Algorithm with Spherical Pruning *(sp-MODE)* [2]. Los algoritmos utilizados para determinar el frente de pareto y conjunto de soluciones han sido desarrollados en *Matlab Central*©.

3.1. Normal Normalized Constraint (NNC)

Al utilizar este método, el problema de optimización se separa en varios problemas de optimización individuales con su respectiva restricción. Después de una serie de optimizaciones, se obtiene un conjunto de soluciones distribuidas uniformemente sobre el frente de Pareto. El método NNC incorpora una aplicación crítica de asignación lineal en los objetivos de diseño. Esta asignación tiene como propiedad deseable que el rendimiento resultante del método sea completamente independiente de las escalas de los objetivos de diseño y tiene la capacidad de generar un buena distribución de soluciones en el espacio de objetivos e incluso en situaciones exigentes numéricamente [13]. El método NNC se presenta aquí para resolver un problema bi-objetivos, pero puede ser generalizado a n-objetivos, para más detalles sobre el método [13].

3.2. Multiobjective Differential Evolution Algorithm with Spherical Pruning (SP-MODE)

En este caso, se ha optado por el algoritmo sp-MODE [14] que ha sido empleado con anterioridad en tareas de ajuste de controladores PI/PID [15, 16]. Se trata de un MOEA que emplea como motor de búsqueda el algoritmo Differential Evolution [17,18]. Para garantizar la diversidad de las soluciones en el espacio de objetivos, emplea un sistema de archivado junto con un sistema de filtrado basado en coordenadas esféricas.

4. Planteamiento del Problema de Optimización Multiobjetivo

Por lo mencionado anteriormente, contamos con dos funciones objetivos, $J_1(\theta) = J_{IAE}(\theta)$ y $J_2(\theta) = TV(\theta)$ que han de ser minimizados con respectos a los parámetros del controlador PI ($\theta = (K_p, T_i)$). Quedando nuestro problema de optimización de la siguiente forma:

$$\min_{\theta} \quad = [J_1(\theta)\ J_2(\theta)] \tag{16}$$

[1] http://www.mathworks.es/matlabcentral/fileexchange/38976

[2] http://www.mathworks.es/matlabcentral/fileexchange/39215

4.1. Visualización del frente de Pareto

Elegir un punto sobre el frente de pareto no es una tarea fácil, ya que no hay manera de mejorar un punto del Pareto sin aumentar o disminuir el valor de al menos, una de las funciones objetivo. Sobre esta base, todos los puntos dentro del frente de Pareto son soluciones igualmente aceptables. Sin embargo, existen muchas formas de hacer la selección y de hacer un análisis cuantitativo y cualitativo de las alternativas de diseño; lo anterior se hace dependiendo del tiempo que el diseñador quiera invertir a este proceso de selección. Se pueden usar reglas heurísticas, reglas cuantitativas, visualizadores, entre otras. Aquí se propone utilizar la solución de Nash para elegir la solución final al MOP como se propuso en [3]. Probablemente la opción más intuitiva para elegir una solución final sería elegir la que está más cerca del punto de utopía: la *Solución de Compromiso* (CS). La CS se calcula mediante la elección del punto en el frente de Pareto que reduce al mínimo la norma-2 desde el punto utopía. Otra posibilidad es aplicar el hecho de que las soluciones que se encuentran en el frente de Pareto son soluciones no dominadas. Ambas situaciones se reflejan en la figura (2). Para explicar la segunda opción, se introduce lo que se puede llamar el punto de desacuerdo. Si pensamos en los dos objetivos de forma independiente, ninguno de los dos estaría de acuerdo en este punto como una solución común, ya que representa la peor situación. El área del rectángulo definido por los puntos (NS, A, B) y el punto de desacuerdo proporciona una medida de la cantidad de soluciones que el punto NS mejora en ambos objetivos al mismo tiempo. Por lo tanto, vamos a elegir la solución de Nash que es la que maximiza dicha zona. La denominación proviene de la identificación de este punto como solución de *Nash* que es un juego de negociación entre ambos objetivos [19].
Cabe señalar que la solución de Nash depende de los extremos del frente de pareto, es por eso que la utilizaremos para la comparación entre las técnicas anteriormente mencionadas.

5. Ejemplo de Comparación

Teniendo en cuenta el proceso de control de cuarto orden propuesto como punto de referencia en [20] por la función de transferencia

$$P_\alpha(s) = \frac{1}{\prod_{n=0}^{3}(\alpha^n s + 1)} \qquad (17)$$

con $\alpha \in \{0{,}5\}$.

Utilizando el proceso de identificación de tres puntos *123c* [21] los modelos de primer orden con

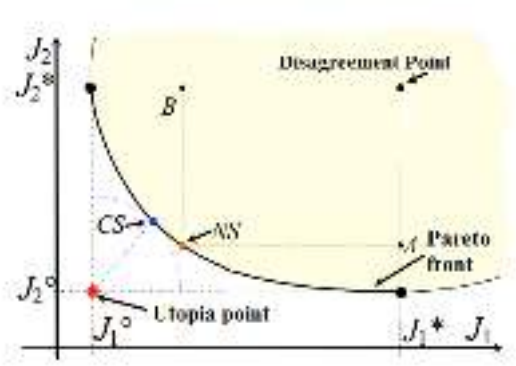


Figura 2: Visualizacion del frente de Pareto

tiempo muerto (FOPDT) han sido obtenidos y sus parámetros se muestran en el Cuadro 1. Este modelo se utilizará para la comparación entre el NNC y el sp-MODE.

Cuadro 1: Example - Modelo FOPDT

α	K	T	L	τ_o
0.50	1	1.247	0.691	0.554

La figura (3) representa el Caso 1, este frente de Pareto fue obtenido utilizando el NNC y un punto inicial de referencia. Este punto fue calculado a través de la sintonía de Ziegler-Nichols ($K_p = 1{,}62$ $T_i = 2{,}30$). La solución que nos ofrece el Ziegler-Nichols es dominada por otras soluciones del frente. Luego tenemos el caso 2, que utiliza el sp-MODE con criterio de pertinencia. La pertinencia se refiere a la capacidad de dar una solución práctica desde el punto de vista del diseñador. En este caso, se acota el frente de pareto, si a priori se sabe que soluciones se buscan y son de interés. Es una opción de utilidad cuando se sabe que en determinados casos, la mejora de uno de los objetivos no justifica la fuerte degradación en los restantes. Como criterio de pertinencia fueron utilizados los valore de las funciones objetivos ($IAE = 0{,}9217$ $TV = 0{,}1319$) calculados con los parámetros antes mencionados. En la figura (4) se observa como el algoritmo centra su búsqueda específicamente en los rangos que abarca los valores de pertinencia que le fueron dados. Para nuestro último caso utilizamos el SP-MODE pero esta vez sin pertinencia, ver figura (5).

Para cada uno de estos casos se cálculo la solución de Nash, por lo tanto en el Cuadro 2 se muestran los parámetros del controlador (PI) y en el Cuadro

Cuadro 2: Ejemplo - Parámetros del Controlador PI

	$\alpha = 0{,}5$	
Métodos	Kp	Ti
Caso 1: NNC	1.34	1.86
Caso 2: sp-MODE (pertinencia)	1.05	1.18
Caso 3: sp-MODE	0.84	1.37

3 estan los valores de robustez y desempeño del sistema. Ademaás, en la figura (6) se encuentra la respuesta del sistema a una perturbació y la señal de control.

Cuadro 3: Ejemplo - Rendimiento y Robustez.

	Caso 1 NNC	Caso 2 sp-MODE (pertinencia)	Caso 3: sp-MODE
$\tau_{0=0,5}$			
IAE	0.9022	0.7662	1.0601
TV	0.1023	0.1118	0.0856
M_s	1.64	1.67	1.44
Y_{max}	0.3016	0.3167	0.3446

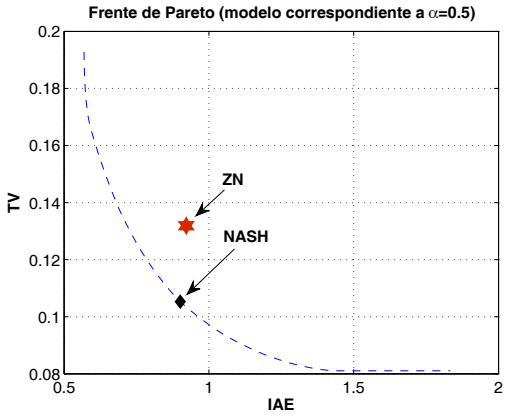


Figura 3: Frente de Pareto correspondiente al modelo $\alpha = 0{,}5$ (Caso 1).

6. Conclusiones

Despés de ver los resultados de nuestra comparación podemos señalar que los tres casos han aproximado soluciones pareto óptimas ya que no se dominan entre ellas en los objetivos de diseño IAE y TV. Teniendo en cuenta que la solución de nash depende del frente de Pareto, podemos decir que si tenemos un buena aproximación del frente, la solución de Nash sería un buen criterio de selección .

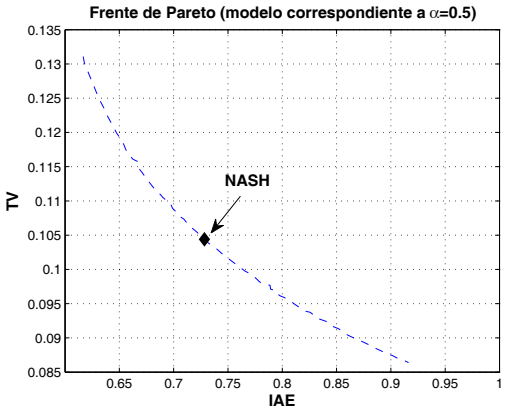


Figura 4: Frente de Pareto correspondiente al modelo $\alpha = 0{,}5$ (Caso 2).

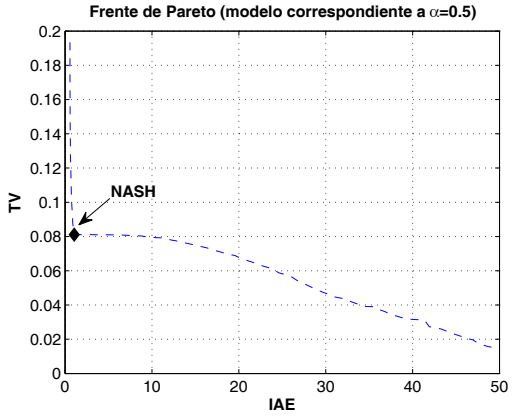


Figura 5: Frente de Pareto correspondiente al modelo $\alpha = 0{,}5$ (Caso 3).

7. Agradecimientos

Este trabajo ha recibido el apoyo financiero del programa CICYT español bajo la subvención DPI 2010-15230.

Referencias

[1] K. J. Åström and T. Hägglund, *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society, 2006.

[2] M. A. Martínez, J. Sanchis, and X. Blasco, "Multiobjective controller design handling human preferences," *Engineering applications of artificial intelligence*, vol. 19, no. 8, pp. 927–938, 2006.

[3] R. V. Helem Sabina Sánchez, "Multiobjective tuning of PI controller using the NNC Method: Simplified problem definition and guidelines for decision making," in *18th IEEE Conference on Emerging Technologies & Factory*

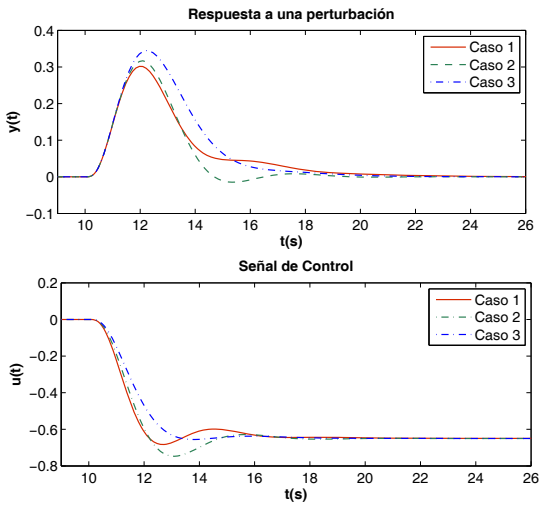


Figura 6: Señal de salida y de control para $\alpha = 0{,}5$

Automation (ETFA), 10-13 September, Cagliari, Italy, 2013.

[4] K. J. Åström, *PID controllers: theory, design and tuning*. Research Triangle Park, 1995.

[5] O. Garpinger, T. Hägglund, and K. J. Åström, "Criteria and trade-offs in PID design," *Proceedings of the IFAC Conference on Advances in PID Control PIDÍ2, 2012*, 2012.

[6] B. Kristiansson and B. Lennartson, "Evaluation and simple tuning of PID controllers with high frequency robustness." *Journal of Process Control*, vol. 16, no. 2, pp. 91–103, 2006.

[7] M. Foley, N. Ramharack, and B. Copeland, "Comparison of PI controller tuning methods," *Ind. Eng. Chem. Res.*, vol. 44, no. 17, pp. 6741–6750, 2005.

[8] K. Miettinen, *Nonlinear multiobjective optimization*. Springer, 1999, vol. 12.

[9] P. P. Bonissone, R. Subbu, and J. Lizzi, "Multicriteria decision making (MCDM): a framework for research and applications," *Computational Intelligence Magazine, IEEE*, vol. 4, no. 3, pp. 48–61, 2009.

[10] C. A. Mattson and A. Messac, "Pareto frontier based concept selection under uncertainty, with visualization," *Optimization and Engineering*, vol. 6, no. 1, pp. 85–115, 2005.

[11] G. Reynoso-Mezaa, X. Blasco, and J. Sanchis, "Optimización evolutiva multi-objetivo y selección multi-criterio para la ingeniería de control," *In: X Simposio CEA de Ingeniería de Control*, Marzo 2012a.

[12] K. Deb, "Advances in evolutionary multi-objective optimization," *Search Based Software Engineering*, pp. 1–26, 2012.

[13] A. Messac, A. Ismail-Yahaya, and C. A. Mattson, "The normalized normal constraint method for generating the pareto frontier," *Structural and multidisciplinary optimization*, vol. 25, no. 2, pp. 86–98, 2003.

[14] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez, "Multiobjective design of continuous controllers using differential evolution and spherical pruning," in *Applications of Evolutionary Computation, Part I*, C. D. Chio, S. Cagnoni, C. Cotta, M. Eber, A. Ekárt, A. I.Esparcia-Alcaráz, C.-K. Goh, J. J.Merelo, F. Neri, M. Preuss, J. Togelius, and G. N.Yannakakis, Eds., vol. LNCS 6024. Springer-Verlag, 2010, pp. 532–541.

[15] G. Reynoso-Meza, J. Sanchis, X. Blasco, and J. M. Herrero, "Multiobjective evolutionary algortihms for multivariable PI controller tuning," *Expert Systems with Applications*, vol. 39, pp. 7895 – 7907, 2012.

[16] G. Reynoso-Meza, S. García-Nieto, J. Sanchis, and X. Blasco, "Controller tuning using multiobjective optimization algorithms: a global tuning framework," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 445–458, 2013.

[17] R. Storn and K. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341 – 359.

[18] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *Evolutionary Computation, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –28, 2010.

[19] R. Aumann and S. H. (ed.), *Handbook of Game Theory with Economic Applications*. Elsevier, 1994.

[20] K. J. Åström and T. Hägglund, "Benchmark Systems for PID Control," in *IFAC Digital Control: Past,, Present and Future of PID Control (PID'00)*, 5-7 April,, Terrasa, Spain, 2000.

[21] V. M. Alfaro, "Low-Order Models Identification from the Process Reaction Curve," *Ciencia y Tecnología (Costa Rica)*, vol. 24(2), pp. 197–216, 2006, (in Spanish).

Utilización del software AnyLogic en la enseñanza del Control Automático

José María González de Durana, Oscar Barambones
Universidad del País Vasco, EUI, Nieves Cano 12, 01006 Vitoria, España
josemaria.gonzalezdedurana@ehu.es, oscar.barambones@ehu.es

Resumen

Los conocimientos de nuestros estudiantes sobre Matemáticas, Física Informática, fundamentales para iniciar el estudio en muchas áreas de la Ingeniería, vienen decreciendo últimamente. Los profesores nos vemos obligados paliar de alguna forma esta carencia, organizando actividades y sugiriendo trabajos que les permitan recuperar algo de esos conocimientos olvidados. No resulta tarea fácil porque asimilar algunos de esos conceptos requiere un tiempo considerable. Se trata de exponer una experiencia personal en este sentido, basada en la utilización de aplicaciones informáticas que permitan a los alumnos realizar diferentes ejemplos de simulación para conseguir una mejor comprensión de los conceptos teóricos. En particular en este trabajo se propone la utilización de la herramienta software AnyLogic para realizar estas tareas.

1. INTRODUCCIÓN

Dicen que nuestros alumnos no tienen el hábito de la lectura pero que son expertos en juegos de videoconsola y de ordenador. Y que si les pones muchas matemáticas en la pizarra, más de uno asusta y no vuelve. ¿Como pues les vamos a enseñar los fundamentos de los Sistemas de Control, plenos de derivadas, integrales y ecuaciones diferenciales?

La simulación de un depósito es un excelente ejemplo para introducir algunos de estos conceptos. Si fuéramos capaces de explicarlo sin fórmulas, a lo mejor la cosa cambiaría.

Comencemos por un simple experimento. Saquemos una pelota de ping-pong y dejémosla caer. Y lancemos al auditorio la siguiente cuestión: si el rozamiento es despreciable, ¿cual será la velocidad de la bola al llegar al suelo? Si, como es de esperar, hay alguno que levanta la mano, le pediremos que pase a la pizarra a dar una explicación: si no hay rozamiento, la energía potencial cuando la bola de masa m está en reposo a una altura h, $E_p = mgh$, será igual a la energía cinética cuando la bola llegue al suelo $E_c = \frac{1}{2}mv^2$. Igualando $E_p = Ec$ se obtiene

$$v = \sqrt{2gh}$$

No vamos bien, ya salió la primera fórmula. El siguiente experimento consistirá en la simulación de un un gran número de bolas botando y chocando entre sí, emulando el comportamiento de las moléculas de un fluido.

2. METODOLOGÍA

2.1. Contenidos

Las ecuaciones diferenciales constituyen una parte conceptual importante no solo de la Automática y los Sistemas de Control sino también de muchas otras áreas de la Ciencia y de la Ingeniería, cada una de las cuales ha desarrollado métodos propios para encontrar soluciones de forma rápida. El integrador de rueda y disco inventado por James Thomson [1], hermano de Lord Kelvin, fue el primer aparato que permitió realizar (mecánicamente) operaciones de *cálculo analógico*. Usando el *integrador* como elemento básico, los dos hermanos construyeron un dispositivo para calcular la integral del producto de dos funciones dadas. Después Kelvin diseño otras máquinas capaces de integrar ecuaciones diferenciales de cualquier orden, si bien no llego a construirlas. Para hallar la solución de una ecuación diferencial dada en forma explícita,

$$\frac{d^n y}{dt^n} = f\left(\frac{d^{n-1}y}{dt^{n-1}}, \ldots, \frac{dy}{dt}, y, u, t\right),$$

junto con los valores de las condiciones iniciales, la idea de Kelvin consistía en integrar con su dispositivo n veces $\frac{d^n y}{dt}$, obteniendo así los valores $\frac{d^{n-1}y}{dt^{n-1}}, \ldots, \frac{dy}{dt}, y$, realizar con ellos las operaciones aritméticas necesarias para obtener la función $f(\frac{d^{n-1}y}{dt^{n-1}}, \ldots, \frac{dy}{dt}, y, u, t)$, y cerrar el lazo [2].

En los años 50, Jay W. Forrester [3], un famoso ingeniero del MIT que pasó del mundo del diseño de sistemas de control para aviones al de la gestión empresarial y de la investigación operativa, se vio en la necesidad de hacer una simulación (de tipo

control de inventario) para la compañía *General Electric*. Esta primera simulación que Jay hizo, con lápiz y papel, se puede considerar como el inicio de la Dinámica de Sistemas (*System Dinamics*) [4]. Más tarde pidió ayuda a Richard Bennett para resolver las ecuaciones con ayuda del ordenador y entonces éste creó un compilador, llamado SIMPLE, para ello. Curiosamente, el método de Bennet era el mismo (adaptado al computador) que Kelvin usó para resolver las ecuaciones diferenciales por métodos mecánicos.

En el área del Control Automático se aplican muy a menudo los métodos denominados *clásicos* y los denominados *modernos*. Los primeros, basados en el empleo de la transformada de Laplace, sirven sobre todo como modelo de los *sistemas lineales*, dan lugar al *modelo externo*, representado por la *función de transferencia*, y al *diagrama de bloques* y permiten analizar la *respuesta de frecuencia*, gráficamente, de forma manual. Los métodos modernos, mediante un estudio sistemático obtienen el denominado *modelo interno* o *modelo de estado*, representado por un sistema de ecuaciones diferenciales de primer orden en forma explícita.

También durante los años 50 surgió el *calculador analógico*, dotado de *integradores electrónicos* realizados mediante válvulas electrónicas y basado en las mismas ideas de Kelvin. Este dispositivo permitía obtener soluciones de las ecuaciones diferenciales en forma de señales eléctricas. A pesar de su gran eficacia (sobre todo comparados con los métodos de integración mecánica), los calculadores analógicos perdieron importancia con la aparición de los ordenadores y hoy en día los métodos digitales dominan la situación. De todos modos el método de Kelvin (o de Forrester) se sigue aplicando en los algoritmos numéricos.

Como complemento y a veces en sustitución de las ecuaciones diferenciales, se han utilizado el el área de Control, de forma continuada desde los 50's, otras *herramientas*, unas para para el análisis y diseño de los sistemas en función de la frecuencia (*diagramas de Nyquist, de Bode, de Nychols*, etc.) y otras en función del tiempo (*lugar de las raíces* o gráfico de Evans, etc.).

2.2. Enseñanza-aprendizaje

La tarea docente ha experimentado profundos cambios durante la última década. A medida que han ido apareciendo más y más fuentes de información, sobre todo a través de Internet, los estudiantes se han hecho cada vez más autónomos para el aprendizaje y el profesor, antes supuestamente dotado de enciclopédicos conocimientos y transmisor prácticamente exclusivo de los contenidos de las asignaturas, ha pasado a ser consultor, estratega y motivador de ese aprendizaje. Las métodos de enseñar y de aprender han evolucionado drásticamente. Aparte de las *clases magistrales* han aparecido otras nuevas tareas para el profesor, que ha de ofrecer a los estudiantes:

- métodos de auto aprendizaje
- estrategias de auto-aprendizaje efectivo
- lugares para buscar la información
- ánimo para aumentar su auto confianza
- apoyo moral en situaciones de fracaso

Y ha de aplicar nuevos métodos para transferir los conocimientos, posiblemente utilizando nuevas herramientas informáticas y de comunicación a través de redes locales y de Internet.

2.3. Método tradicional

La cantidad de materias que podemos apreciar como objeto de aprendizaje al leer las palabras escritas en letra cursiva de la sección 2.1, es elevada. Puede ser que a los profesores o profesionales del área de Control nos pueden parecer conceptos sencillos, por ser conocidos, pero no es así para los nuevos alumnos, los cuales pueden sentirse abrumados ante el cúmulo de conocimientos que deben aprender. Si bien pensamos que un estudiante de ingeniería debe de conocer todos o casi todos esos contenidos, entre oras cosas porque son necesarios para entender cualquier libro o artículo que trate sobre temas de Control, hay que tener en cuenta que además de que son muchos, son muy variados y que a veces se expresan en base a conceptos matemáticos que ellos no dominan.

Tradicionalmente, el método de estudio de los sistemas de control seguía las fases siguientes

1. análisis de los sistemas
2. diseño de controladores
3. simulación y/o realización.

Con este esquema, muchas veces las dos primeras partes ya cubrían casi todo el curso académico y los pocos alumnos que llegaban a la tercera con conocimientos suficientes disponían de poco tiempo para poder aplicarlos al diseño. La simulación aparecía en la última fase, muy próxima a la realización, y en definitiva no era sino una visualización más o menos realista del modelo matemático.

El que el alumno no pueda hacer una simulación del sistema hasta no tener un profundo

conocimiento matemático del modelo, tras pasar larga fase previa de estudio matemático, supone un grave inconveniente desde el punto de vista didáctico, porque esta fase previa suele resultar desalentadora para un buen número de estudiantes los cuales, por no entender muy bien el objetivo al que pretenden llegar se desaniman y, en lugar de estudiar para tratar entender la materia, lo que hacen es aprender, más o menos de memoria, una colección de "problemas tipo" de los que creen que pueden caer en el examen.

Por todo esto planteamos la siguientes cuestiones: ¿podríamos conseguir de alguna manera un aprendizaje más efectivo? En caso afirmativo ¿como podríamos hacerlo?

2.4. Objetivos

Si lo que queremos conseguir es un aprendizaje más efectivo, surge de pronto otra pregunta: ¿es posible aplicar los métodos denominados *lean manufacturing* al proceso de enseñanza-aprendizaje en un contexto universitario?. Pues bien, parece que sí, y algunos que ya lo están haciendo describen en [5] cómo aplicar los conceptos y técnicas desarrolladas para el proceso de producción de Toyota al proceso de enseñanza-aprendizaje y denominan *lean teching* al método. Aplicando estos métodos tratan de conseguir un aprendizaje *Just-In-Time*, reducir el tiempo consumido en actividades didácticas, ofrecer una experiencia educativa mejor a los alumnos, reducir la complejidad de las materias, y eliminar los contenidos basura.

Para aplicar estos métodos se precisa una profunda revisión de los contenidos de las asignaturas. Los contenidos pueden ser más o menos los mismos pero es esencial establecer un orden para el aprendizaje. Además es preciso adelantar la simulación a la primera fase, es decir a la de análisis. De este modo las fases de aprendizaje serían:

- análisis – asistido por simulación
- diseño del control
- realización y/o posible simulación final.

Si desde el comienzo del curso el alumno comienza a realizar ejercicios de simulación, en los que pueda apreciar los conceptos importantes y la problemática de control existente, sin duda crecerá su motivación para el aprendizaje.

Así pues, comenzaremos dirigiendo al alumno hacia el aprendizaje de aquellos contenidos que son fundamentales para poder entender las primeras simulaciones. Posiblemente estas materias son conocidas para el alumno en ese momento por lo que el esfuerzo necesario será mínimo. Continuaremos con más ejercicios de simulación, para los que precisará aprender nuevos contenidos, y dejaremos para el final el aprendizaje de los contenidos más accesorios. Están en este grupo las herramientas clásicas de análisis en el tiempo y en la frecuencia, lugar de las raíces, etc., todo esto es de momento accesorio. Y posponemos el aprendizaje de los métodos basados en la *función de transferencia* y *diagrama de bloques* en favor de los de *dinámica de sistemas*.

Con este método pretendemos que el alumno, ya desde las primeras sesiones de clase, tenga la posibilidad de realizar simulaciones de algunos sistemas. Pero no las típicas simulaciones basadas en el modelo matemático sino de otro tipo, dotadas de animación, que le permitan ver con más claridad el problema real para poder analizarlo, obtener después el modelo, y más tarde aplicarle las técnicas de control que considere oportunas.

No nos es posible abordar aquí toda la problemática planteada, ni tampoco desarrollar por completo los métodos indicados, por lo que nos limitaremos a presentar unos ejercicios de simulación con el programa AnyLogic, que pensamos son de interés para introducir algunos de los conceptos básicos del Control Automático y que han sido realizados por nuestros alumnos el pasado curso.

2.5. Programas de simulación

Es probablemente Matlab el programa de simulación más extendido en el ámbito universitario de los Sistemas de Control, si bien hay otros muchos programas como Maple, Matemática, Octave, Scilab, etc. que también se utilizan.

Con Matlab es posible realizar simulaciones en el sentido matemático, que a la postre consisten en aplicar métodos numéricos para resolver unas ecuaciones diferenciales que describen un sistema. Sin embargo, al ser un programa diseñado con las tecnologías de los años 50-60, no tiene las ventajas de otros programas más modernos basados en la programación orientada a objeto. Estas ventajas se ponen de manifiesto si el sistema a modelar es del tipo de *eventos discretos* y más aún si se trata de *un sistema híbrido* o un sistema *basado en agentes*.

AnyLogic es un programa de reconocido prestigio en la comunidad de *System Dynamics* y poco conocido en las áreas de Control y Automatización, que está basado de los últimos avances en la teoría de modelado de sistemas complejos y que utiliza los métodos más avanzados de mod-

elado orientado a objetos que han surgido durante la última década y los estándares más actuales de diseño de sistemas [6]. AnyLogic dispone actualmente de tres métodos o *paradigmas de modelado*:

SD System Dynamics (Dinámica de Sistemas)

DE Discrete Event (Eventos Discretos)

AB Agent Based (Basado en Agentes)

Estos tres paradigmas son compatibles entre sí, de modo que un sistema puede modelarse con sólo uno o con varios varios de ellos, según convenga. Por ejemplo, para hacer el modelo de un sistema híbrido utilizaremos el método SD para modelar la parte continua de las ecuaciones diferenciales y el método DE para modelar los eventos. Para confeccionar el modelo, el usuario, mediante una interfaz gráfica va arrastrando (con el ratón) los objetos que están colocados en una ventana llamada *Palette*, que contiene las librerías de objetos (*Model*, *Análisis*, *Enterprise Library*, etc.), a la ventana creada para el modelo. Una vez que en el modelo están los objetos gráficos, el usuario ha de asignarles las propiedades convenientes a cada uno de ellos y conectarlos entre sí adecuadamente.

Esta forma de trabajar recuerda en cierto modo al funcionamiento de *Simulink*, si bien los objetos son muy diferentes. El elemento fundamental del modelo de AnyLogic es el objeto activo (*Active Object*) que es una construcción mucho más potente que la de los objetos de *Simulink*: tiene interfaz, comportamiento interno, puede encapsular otros objetos, tiene parámetros, puede heredar propiedades de otros objetos, etc. Esto lo convierte en una construcción muy reutilizable: una vez desarrollado se pueden obtener instancias del mismo en diferentes contextos y con diferentes parámetros, o incuso utilizar la encapsulación y la herencia para definir un comportamiento "básicamente similar, pero un poco diferente".

Otra ventaja de AnyLogic es que está basado en Java. El usuario no está limitado a usar un lenguaje tipo *script* sino que puede emplear expresiones y código en Java para definir parámetros o para implementar funcionalidades adicionales. Esto está permitido en cualquier lugar en la modelo. Java también se puede emplear para acceder a programas y datos externos y para construir modelos de simulación de gran accesibilidad. El usuario tiene acceso directo a una enorme cantidad de paquetes de Java que se están desarrollando en todo el mundo, ya sea de matemáticas, estadística, gráficos, o de otro tipo.

Los modelos de AnyLogic son aplicaciones Java portátiles que funcionan por sí mismas, no necesitan el entorno de AnyLogic para ejecutarse. Además son multiplataforma y se pueden ejecutar en cualquier lugar donde esté instalado Java *runtime*. Por ello lo mismo se pueden ejecutar en un navegador Web que como *Java applets*, con lo que podemos dar a conocer nuestros modelos sin más que ponerlos en una página web.

Los métodos de animación de AnyLogic están también orientados a objeto, igual que el propio modelo. Resulta muy sencillo desarrollar animaciones de los objetos activos porque el montaje de la imagen se realiza de forma automática. De este modo las animaciones son altamente reutilizables y pueden ser mostradas en las applets.

En AnyLogic es posible crear fácilmente modelos escalables, ya que se pueden definir arrays de objetos cuyo tamaño sea un parámetro de nuestro modelo. Incluso se puede variar dinámicamente la estructura del modelo: se pueden añadir o eliminar objetos o cambiar su interconexión durante el tiempo de ejecución, para reflejar los cambios dinámicos que se pueden dar en sistema real.

En lo que respecta a los algoritmos de simulación, pensamos que están en la cima del estado del arte actual en esta área. Los métodos numéricos implementados comprueban la exactitud del modelo en la fase de compilación y chequean la la asignación de salidas, la duplicación de variables en el lado izquierdo de la fórmula, los bucles algebraicos, etc., y algunos de ellos han sido modificados para trabajar en entornos híbridos.

Por último, para la optimización del modelo, AnyLogic incluye en su versión profesional el optimizador OptQuest de OptTek, líder mundial en optimización.

3. RESULTADOS

Debido a la corta extensión de estas notas no es posible exponer todos los ejemplos realizados durante el curso sino más bien ofrecer una muestra para dar una idea de la aplicación de los métodos.

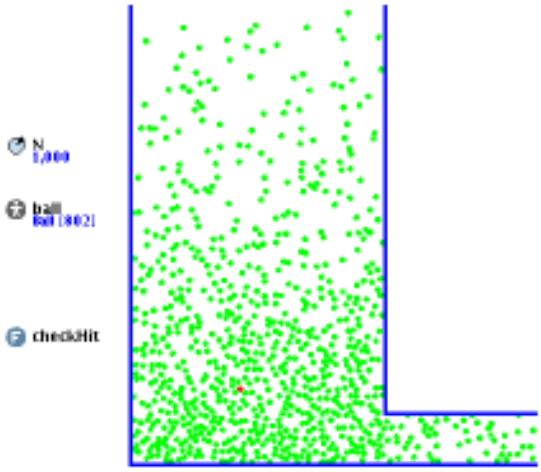
3.1. Depósito de agua

El ejercicio del depósito es muy apropiado para empezar, todo el mundo tiene la idea intuitiva de un recipiente que se va llenando de agua, y puede ser un buen ejemplo que estimule al recuerdo del alumno de algunos conceptos importantes. En él subyace de de modo natural la idea de integral así como las nociones de ecuación diferencial y de problema de condiciones iniciales. El ejercicio se puede plantear al comienzo de un curso básico sobre Regulación Automática, ya que el alumno no necesita estudiar nada nuevo para realizarlo

sino que sólo tiene que emplear algunos conceptos matemáticos que ya conoce. Pero también intentaremos facilitarle el recuerdo de los mismos mediante la simulación.

3.1.1. Modelo de agentes.

En primer lugar vamos a realizar un *modelo basado en agentes* del depósito. Con la pelota de ping-pong de la introducción in mente, vamos a suponer que el comportamiento de un depósito de fluido es como el de un gran número de bolitas que caen bajo la acción de la gravedad y rebotan entre sí y contra el suelo. Se trata de una suposición irreal pero optamos por ella por la facilidad que supone modelar cada bolita como la pelota de ping pong de nuestro primer experimento. Después los alumnos podrán adaptar el comportamiento a otras hipótesis más reales.



Durante la simulación el alumno puede ver algunas variables interesantes e intentar asociarlas a las de un depósito de agua. Por ejemplo, número de bolas → volumen, numero de bolas que salen por segundo → caudal, etc. Esto puede ayudar al profesor en la tarea de repasar conceptos tales como los de derivada e integral (volumen = integral del caudal), ecuación diferencial, condiciones iniciales, etc.

Además adquirirá quizás su primeras nociones de *evento* (las bolitas chocan entre sí y contra las paredes), *sistema híbrido*, etc. y habrá realizado un *modelo basado en agentes*, algo considerado como muy complicado hasta hace poco.

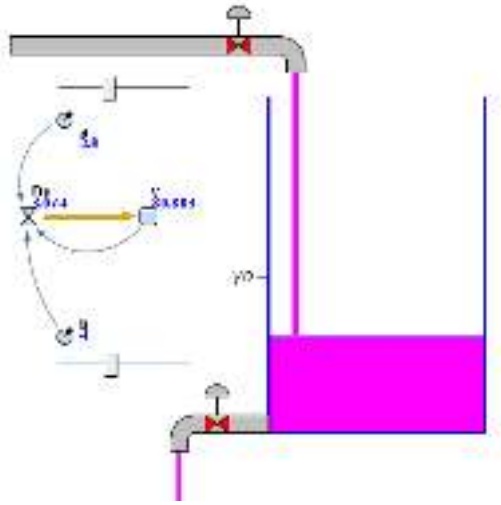
3.1.2. Modelo SD

El alumno sabe por sus nociones elementales de Física que si el rozamiento es despreciable, la velocidad de salida es $v_s = \sqrt{2gy}$ (¡igual que de la bola de nuestro primer experimento!), el caudal de salida es $q_s = bv_s = b\sqrt{2gy}$ y, por tanto, $q - q_s = A\frac{dy}{dt}$, es decir, $\frac{ap}{A} - b\sqrt{2gy} = A\frac{dy}{dt}$, siendo p la presión de entrada, a y b las áreas de las secciones de las válvulas, A el área constante de la sección del depósito y c otra constante. Despejando la derivada obtenemos la ecuación diferencial, y si añadimos el valor y_0 del nivel en el instante inicial $t = 0$, es decir $y(0) = y_0$, obtenemos

$$\begin{cases} \dfrac{dy}{dt} = -\dfrac{b}{A}\sqrt{2gy} + \dfrac{cp}{A}a \\ y(0) = y_0 \end{cases}$$

que es el problema de condiciones iniciales. De la comparación de este modelo (SD) con el anterior (AB) el profesor podrá sacar partido para reciclar conceptos.

La simulación de lo descrito hasta aquí se podría hacer con cualquier programa, por ejemplo con Matlab-Simulink, pero AnyLogic tiene la ventaja de que permite realizar fácilmente una animación, sin más que dibujar un rectángulo al que le asignamos la propiedad dinámica altura (height) igual a y. Con ello vemos cómo se llena el depósito en la simulación. Una imagen (¡animada!) vale más que mil palabras.



En la figura podemos ver el diagrama SD (*System Dynamics*) correspondiente al PCI y el dibujo de la animación del depósito, con dos válvulas, cada una de las cuales lleva un *slider* asociado con el que puede variarse su sección. De este modo el depósito puede llenarse y vaciarse a voluntad del usuario durante la simulación.

Nuestros alumnos, al jugar con la simulación (hay que tener presente que si algunos tienen mermado el recurso de la lectura entonces casi seguro que son expertos en juegos digitales), pronto se darán

cuenta de que el nivel de líquido puede rebasar la altura máxima del depósito, lo cual, en la realidad provocaría un derrame de líquido. Y surgirá de modo inevitable la pregunta:

«Profesor, ¿porqué sigue echando agua la válvula una vez que el depósito se ha llenado hasta arriba?»

Entonces, el profesor, con cara de perplejidad y la mente puesta en la teoría de las ecuaciones diferenciales, quizás le de una respuesta poco convincente. Creo que, en lugar de eso, ha de resultar más efectivo volver a utilizar los recursos del alumno, las ideas que tiene, sin saberlo, sobre sistemas de eventos discretos y sistemas híbridos. Él ha aprendido a llenar una vasija de agua y sabe que debe *observar* el nivel y que al llegar a un nivel próximo al máximo ha de *actuar* sobre la válvula de entrada para cerrarla. Con la ayuda del profesor y poniendo un poco de esfuerzo de su parte, seguro que el alumno será capaz de aprender a utilizar las cartas de estado (*Statecharts*) y añadir al modelo la parte DE (*discrete event*). Para ello habrá de poner sensores de nivel, sustituir las válvulas por electroválvulas, etc., hasta completar el modelo a su antojo.

Con ello el alumno no sólo habrá conseguido aumentar su motivación, sino que también habrá adquirido algunas nociones básicas del área de Automatización, que le serán más adelante de gran ayuda para el aprendizaje de otras asignaturas relacionadas con este tema.

3.2. Diseño de controladores

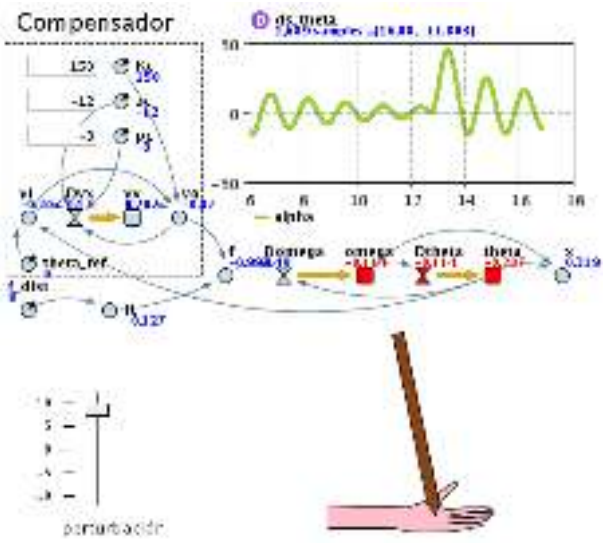
Con el ejercicio del depósito y algunos otros que el profesor haya propuesto junto con las clases magistrales a las que, supongamos, haya asistido con gran interés, el alumno habrá aprendido algunos conceptos básicos sobre la Teoría de Control y los habrá puesto en práctica.

Entonces no le resultará difícil entender las lecciones de clase sobre diseño.

Un ejercicio interesante que puede hacerse con AnyLogic es el de tratar de controlar una varilla de masa m y longitud l colocada sobre la mano en posición vertical. Muchos ya habrán hecho este experimento en su casa pero es fácil repetirlo en clase.

Como puede apreciarse en la figura, la ecuación diferencial

$$\begin{cases} \dfrac{d\theta}{dt} = \omega \\ \dfrac{d\omega}{dt} = \dfrac{3g}{2l}\sin\theta - \dfrac{3}{2ml}\cos\theta \cdot f, \end{cases}$$



junto con la ecuación de salida

$$x = -\frac{l}{2}\sin\theta$$

y el controlador de adelanto-retraso de fase, se han modelado empleando el método SD. El modelo realizado permite cambiar los parámetros z_c, p_c y K_c del controlador, así como la intensidad del impulso de perturbación, antes y/o durante la ejecución. De ese modo el alumno podrá aplicar los métodos de diseño de y de sintonía, viendo el resultado inmediatamente en la simulación.

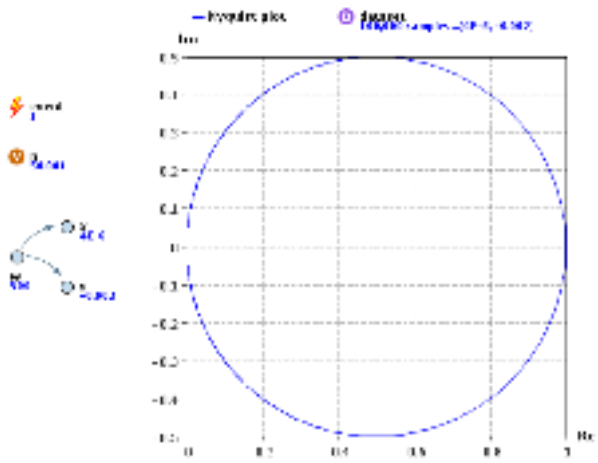
3.3. Herramientas clásicas

Con el método propuesto, el estudio de las herramientas clásicas de análisis y diseño queda pospuesto. De cualquier modo, evidentemente, su exposición en clase y propuesta de trabajos de aprendizaje deberá hacerse con anterioridad a la presentación de los métodos de diseño. Es razonable pensar que se utilice Matlab (o algún otro programa similar) en esta fase, si bien también sería posible efectuar todos los cálculos necesarios con AnyLogic. Se indican a continuación algunos ejemplos ilustrativos sobre ello.

3.3.1. Respuesta de frecuencia y lugar de las raíces

El diagrama de Nyquist es una gráfica en coordenadas paramétricas de un función racional, la función de transferencia $G(s)$, i.e., del conjunto $\{(G(i\omega).\text{real}, G(i\omega).\text{imag})\}$, $\omega \in \mathbb{R}$. Se puede realizar fácilmente como se ve en la figura, que corresponde a $G(s) = \frac{1}{s+1}$.

De modo similar también puede obtenerse el diagrama de Bode.



El Lugar de las Raíces se puede obtener utilizando alguna librería de Java de Algebra Lineal, como por ejemplo JAVAL. También podrían implementarse otras herramentas de análisis y diseño similares a las que porporcionan las *toolboxes* de Matlab.

3.4. Diseño de sistemas secuenciales

Una interesante aplicación de las cartas de estado (*statecharts*), disponibles en AnyLogic, es la del diseño de sistemas digitales. Utilizando cartas de estado se puede diseñar la implementación de practicamente cualquier sistema secuencial, y se puede simular inmediatamente su funcionamiento, lo que proporciona un eficaz y rápido método de realizar protitipos (*rapid prototyping*). Por este procedimiento se ha diseñado de un sistema con el microprocesador PIC16F628A para realizar un mando a distancia para TV utilizando un único pulsador, para su utilización por personas discapacitadas [7].

4. CONCLUSIONES

Los métodos desarrollados consisten básicamente en utilizar la simulación (con el programa AnyLogic) para el *Aprendizaje Basado en el Trabajo* de los contenidos elementales del área de Automática, los cuales han sido previamente cribados empleando algunos conceptos propios de *lean manufacturing*.

Quizás la base del éxito reside en conseguir aumentar la eficacia del proceso de enseñanza-aprendizaje ya desde el inicio de la asignatura, mediante el empleo de los métodos antedichos. Por ello cobra gran importancia la etapa inicial: al comenzar el curso es preciso que los alumnos, con los escasos conocimientos que disponen de Matemáticas, Física e Informática y con unas mínimas nociones para el manejo del programa de simulación, sean capaces de realizar unos motivadores ejercicios. Y será labor no trivial del profesor elaborar tales ejercicios para que el alumno, al realizarlos, adquiera sin darse cuenta algunas de las nociones fundamentales de Automática: sistema dinámico, proceso continuo, eventos. La incorporación de las herramientas clásicas de análisis y diseño tales como la Respuesta de frecuencia y lugar de las raíces quedan pospuestas para cuando el alumno haya adquirido la suficiente pericia en el manejo de los sistemas de control y esté preparado para asimilar esos conocimientos.

Los resultados obtenidos al aplicar estos métodos en los últimos cursos han sido francamente buenos, consiguiendo la adquisición por parte de los alumnos de conocimientos básicos del área de Automática. A través de encuestas y foros, hemos observado que el grado de aprendizaje de los contenidos es mucho mayor que usando los métodos tradicionales y también que el grado de satisfacción de los alumnos es elevado.

Creemos que se ha alcanzado el principal objetivo: obtener un aprendizaje más efectivo de dichos contenidos básicos. Queremos destacar la importancia didáctica de la simulación en aprendizaje de los sistemas continuos y de eventos discretos.

Referencias

[1] Thomson, J. *On an Integrating Machine Having a New Kinematic Principle*. Proc. of the Royal Society of London, 24 : 262-5, 1876

[2] Kailath, T, *Linear Systems*, Prentice Hall, Inc., 1980

[3] Forrester, J.W., *The Beginning of System Dynamics*, Banquet Talk at the international meeting of the System Dynamics Society, Stuttgart, Germany, 1989

[4] Gordillo, F., Aracil, J., *Dinámica de Sistemas*, Alianza Editorial, 1998.

[5] Turner, K. *Lean teaching*. 10 Annual Learning and Teaching Conference, Brighton, UK, 2005. http://staffcentral.brighton.ac.uk/clt/events/conf/2005/index.htm

[6] Borshchev, A., Filippov, A. *From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools*. The 22nd Int. Conf. of System Dynamics Society, Oxford, England, 2004.

[7] Kremers, E., González de Durana, J.M., Barambones, O., *Emergent synchronisation properties of a refrigerator demand side management system*, Elsevier, Applied Energy, 101, 0, 709-717, 2013

SISTEMAS DE DOCUMENTACIÓN Y ELABORACIÓN DE TEXTOS CIENTÍFICOS: UNA ASIGNATURA PARA APRENDER A PUBLICAR EN INGENIERÍA

Jorge L. Martínez, Anthony Mandow y Alfonso García-Cerezo
Dep. Ingeniería de Sistemas y Automática, E. T. S. Ingenieros Industriales, Universidad de Málaga
jlmartinez@uma.es, amandow@uma.es, gcerezo@ctima.uma.es

Resumen

Hoy en día, la elaboración y presentación de informes de investigación resulta indispensable para muchos profesionales de la ingeniería. En este artículo se describe una asignatura de máster universitario cuyo principal objetivo es proporcionar los conocimientos necesarios para poder escribir y exponer resultados de investigación en ingeniería mecatrónica. De esta manera, se complementa la formación de los ingenieros para que puedan desenvolverse con éxito en el complejo mundo de la investigación.

Palabras clave: aprendizaje basado en proyectos, publicaciones científicas, ingeniería mecatrónica, iniciación a la investigación.

1 INTRODUCCIÓN

Publicar forma parte esencial del proceso de investigación, ya que es el medio a través del cual se socializan los resultados alcanzados y se ponen a disposición de la comunidad científica [1]. Sin embargo, la enseñanza acerca de la publicación de resultados rara vez ha estado incluida en la docencia reglada con la que se forman los posgraduados hispanohablantes [2]. Es más, la irrupción de las tecnologías de la información y la comunicación (TIC) en distintas etapas de la documentación y el proceso editorial así como la creciente necesidad de interacción del investigador con el resto de la comunidad científica requiere dotar a estas asignaturas de un enfoque eminentemente práctico.

Por todo ello, hoy en día resulta imprescindible para todos aquellos ingenieros que quieran doctorarse o desarrollar nuevas tecnologías saber plantear, elaborar y presentar resultados de investigación. Hasta tal extremo, que todo el esfuerzo puede pasar desapercibido si no se consigue una comunicación adecuada en un medio de relevancia a nivel internacional.

Precisamente, en este artículo se propone una asignatura de máster universitario cuyo principal objetivo consiste en dotar a nuestros alumnos de competencias para lograr el éxito en la publicación científica dentro de un contexto de altísima competitividad internacional. Con este fin, se propone desarrollar una estrategia de aprendizaje basado en proyectos [3] que complemente la base teórica con actividades donde el estudiante se enfrente de manera realista a los retos y dificultades que encontrará en el desempeño de su actividad innovadora.

El resto de este artículo se organiza como se indica a continuación. En la sección 2 se presenta el contexto académico de la asignatura. Después, en la sección 3 presenta la metodología docente propuesta. La sección 4 desarrolla el contenido teórico y práctico de cada uno de los bloques temáticos. En la seccione 5 se reseñan textos apropiados para la asignatura. La sección 6 se dedica a las conclusiones.

2 CONTEXTO ACADÉMICO

"*Sistemas de Documentación y Elaboración de Textos Científicos*" (abreviado SIDETEC) es una asignatura obligatoria de 5 créditos del Máster Oficial en Ingeniería Mecatrónica que imparte la E.T.S. de Ingenieros Industriales de la Universidad de Málaga desde el curso académico 2011/2012.

El objetivo fundamental del Máster es formar a ingenieros con capacidad de investigación en mecatrónica. Ésta es una rama transversal de la ingeniería que trata las metodologías y tecnologías que permiten la creación de productos avanzados y sistemas de ingeniería mediante la combinación sinérgica de las tecnologías mecánica, electrónica, eléctrica, automática, informática y telecomunicación.

De acuerdo con la memoria de verificación del Máster [4], la competencia específica de SIDETEC es: "*Conocer los diferentes tipos de publicaciones científicas, los mecanismos de búsqueda y evaluación y las técnicas y estilos a utilizar en cada una de ellas*". Los aspectos relacionados con la protección de resultados de investigación se abordan en otra asignatura obligatoria del Máster denominada "*Iniciación a la transferencia del conocimiento*". Por ello, en SIDETEC

se considera que el contenido objeto de publicación es abierto y que se pretende proporcionar la máxima difusión al mismo.

SIDETEC posee un carácter universal para el campo de las ingenierías, y se cursa simultáneamente con la realización del trabajo fin de máster en su tercer y último semestre. De ahí que como objetivo secundario de la asignatura, se considere ayudar al alumno en la elaboración de su trabajo fin de máster.

SIDETEC es impartida en 16 sesiones semanales de dos horas y cuarto cada una. Los docentes cuentan con una larga trayectoria investigadora y experiencia en publicaciones avalada por sexenios de investigación. Actualmente, la mayoría de alumnos procede de las titulaciones de ingeniería industrial e informática, pero se espera que se vayan incorporando progresivamente los nuevos graduados en ingeniería. El número de alumnos matriculados en cada curso académico está limitado a 25.

3 METODOLOGÍA DOCENTE

La asignatura se propone como un taller práctico en un aula de ordenadores haciendo uso de la filosofía de aprendizaje basada en proyectos [3], lo que permite a los alumnos recrear en el aula el ciclo de vida de una publicación y la presentación de resultados de investigación. De esta manera, se persigue motivar a los estudiantes para que ellos mismos descubran los elementos clave en la escritura de textos científicos.

Cada sesión se inicia con una breve lección magistral sobre los correspondientes objetivos y conceptos a tener en cuenta. Continúa con la realización de ejercicios supervisada por el profesor. Al final de cada clase, los alumnos deben entregar los resultados obtenidos a través del Campus Virtual de la Universidad de Málaga.

El conjunto de las sesiones se vertebra en torno a la elaboración de un breve artículo científico a lo largo del curso. La temática de este artículo es de libre elección para los alumnos y suele estar relacionada con el trabajo fin de máster. Aparte de las clases presenciales, el desarrollo del artículo y otras actividades relacionadas requieren de un buen número de horas de trabajo autónomo por parte del alumno.

Los criterios de evaluación del alumnado se resumen en la Tabla 1. El mayor peso recáe sobre el artículo presentado: una versión final y revisada del texto elaborado. De esta forma, se comprueban las habilidades adquiridas por el alumno durante el curso mejor que con un examen final. Pero también se consideran importantes tanto la participación activa en clase como la realización de las prácticas propuestas.

Tabla 1: Evaluación del alumnado.

Actividad	**Porcentaje**
Asistencia a clases	20%
Realización de prácticas	30%
Artículo entregado	50%

4 DESARROLLO DEL CURSO

En la primera clase se realiza la presentación general de la asignatura, subrayando la importancia de la divulgación de resultados en el método científico. También se realiza un coloquio con los alumnos para tratar de conocer los conocimientos previos con los que parten y sus intereses sobre la materia.

Aparte de la clase introductoria, el temario se compone de tres grandes bloques temáticos (véase la Tabla 2), que se desarrollan en los siguientes apartados.

Tabla 2: Temario de SIDETEC.

Bloque	**Tema**
Documentación	1. Bases de datos
	2. Sistema de citas
Elaboración	3. Edición de textos
	4. Organización de textos
Presentación	5. Proceso de publicación
	6. Presentación oral

4.1 DOCUMENTACIÓN

La documentación científica resulta esencial para situar cualquier investigación dentro de su contexto, conocer las técnicas consolidadas, y enfatizar la originalidad del trabajo propio. En este bloque se abordan dos temas: las bases de datos de publicaciones en ingeniería y el sistema de citas científicas.

En el tema 1 se descubre el papel fundamental de las bases de datos científicas en el proceso de la documentación. Así, se abordan las siguientes cuestiones:

- Introducción a la documentación, sobre la necesidad de establecer el estado del arte sobre el que versa una investigación.
- Criterios de búsqueda para encontrar y seleccionar publicaciones fiables, relevantes y adecuadas mediante las bases de datos.

- Bases de datos de libre acceso en internet, tales como *Google académico, Scirus, CiteSeerX* o el directorio de revistas de libre acceso (DOAJ).

- Bases de datos con suscripción que proporciona la Universidad de Málaga, tales como *IEEE Xplore, Science Direct, Springer Link, Wiley Interscience*, o revistas de *Sage, Cambridge, Oxford, Taylor & Francis*, ...

En las prácticas el alumno tiene que localizar artículos tanto de un determinado autor como de una temática específica. Asimismo, debe buscar y seleccionar de manera razonada al menos diez publicaciones relevantes relacionadas con su temática de estudio.

Por su parte, el tema 2 se dedica al sistema de citas científicas. Concretamente, se desarrollan los siguientes contenidos:

- Introducción al sistema de citas y su importancia para establecer el impacto de una publicación en la comunidad científica.

- Búsqueda de citas mediante bases de datos de libre acceso (*Google académico* y *CiteSeerX*) y con suscripción (*Scopus* e *ISI Web of Knowledge*).

- Índicadores de rendimiento, como los índices h y g para los investigadores y las herramientas automáticas de cálculo como *Publish or Perish, Scopus* e *ISI Web of Knowledge*.

- Clasificación de las revistas. Se estudia el factor de impacto de las revistas elaborado por el *Journal Citation Reports* y cómo seleccionar una revista adecuada para enviar un artículo según lo investigado.

En este tema se pide al alumno buscar todas las citas de unas determinadas referencias. Además, se propone calcular los índices h y g de algún profesor con repositorio de publicaciones accesible. De cara a su proyecto, el alumno tiene que escoger un conjunto de revistas adecuadas para publicar hipotéticos resultados de investigación.

4.2 ELABORACIÓN

En el bloque de elaboración se incide sobre el proceso de escritura de textos científicos. Para ello cuenta con dos temas: la edición de textos con LaTeX y la organización de documentos.

Respecto de la edición de textos, los contenidos del tema 3 incluyen:

- Introducción al LaTeX. Aparte de familiarizar a los alumnos con el uso de esta herramienta, se señalan las ventajas e inconvenientes con respecto a los editores de texto convencionales.

- Edición de fórmulas, tablas y figuras: se describe cómo incluir figuras, dar formato a tablas y cómo obtener fórmulas matemáticas complicadas. También se estudia cómo etiquetarlas e insertar citas dentro del texto a las mismas.

- Edición de citas para proporcionar formato a las referencias bibliográficas con el entorno *thebibliography* y con el programa BibTeX.

Para las prácticas del tema 3, cada estudiante utiliza una plantilla LaTeX para empezar a editar un artículo de ensayo que incluya al menos una figura y una tabla. Después, se le añaden las fórmulas matemáticas propuestas por el profesor. Finalmente, debe introducir citas a las referencias seleccionadas en el tema anterior.

En el tema 4 se estudia la organización de textos científicos, con atención a las siguientes cuestiones:

- Tipos de publicaciones, según el contenido de las mismas y por el medio de difusión donde aparecen.

- Estructura de los textos científicos y los contenidos asociados a cada una de las secciones.

- El estilo de escritura científico y sus particularidades. Puesto que la forma de expresarse condiciona la interpretación que puede darse al texto, se recomienda ser precisos, claros y concisos para, además de informar, convencer al lector.

- Uso del Inglés escrito. Se presta especial atención al uso de este idioma, ya que es el que se emplea comúnmente en las publicaciones internacionales y no es el primer idioma de los alumnos del curso.

En las prácticas, el alumno elabora un título y un resumen para un artículo real al que se le han suprimido estos apartados. Este resumen también se escribe en Inglés. Por último, los estudiantes comienzan a desarrollar un artículo breve que incluya el título, resumen, estado del arte y objetivos de su temática particular.

4.3 PRESENTACIÓN

La presentación de resultados culmina y da valor al esfuerzo de investigación. A este respecto, se

tratan dos temas: el proceso de publicación y las presentaciones orales.

Para el proceso de publicación (tema 5) se contempla lo siguiente:

- El proceso de revisión por pares. El conocimiento de los criterios de revisión permite a los estudiantes desarrollar un sentido auto-crítico respecto a su trabajo. Aparte de ésto, se les prepara para aceptar y responder a las críticas recibidas.
- Derechos y obligaciones de los autores. En este sentido, resultan interesantes las posibilidades de los repositorios personales en internet.
- Ética en las publicaciones, donde se aborda el comportamiento ético y responsable de los científicos e investigadores [5], incluyendo la problemática de repetir contenidos en diferentes publicaciones, el uso de material ajeno y el falseamiento de datos.

En las prácticas, los alumnos someten su artículo al escrutinio de sus pares de manera anónima y a través de formularios tipo. Después, deberán corregir su propio artículo de acuerdo con las críticas recibidas por parte del profesor y de sus compañeros.

Por su parte, en el tema 6 se desarrollan los siguientes contenidos sobre las presentaciones:

- Elaboración del material, para diseñar diapositivas bien estructuradas y organizadas, con gráficos efectivos que atraigan la atención del público.
- Presentación oral. Se introducen los diferentes estilos de comunicación y se dan consejos para conectar con la audiencia y afrontar con éxito las preguntas que surjan.

En este tema, las prácticas se refieren a la elaboración de una presentación del artículo (utilizando *PowerPoint* u otras herramientas similares). Después, se simula una sesión de congreso donde alumno debe hacer la exposición al resto de alumnos con una duración de diez minutos incluyendo cinco minutos para el turno de preguntas y críticas.

5 SOBRE LA BIBLIOGRAFÍA

El primer bloque de la asignatura sobre documentación apenas cuenta con bibliografía, ya que la reciente irrupción de nuevas herramientas bibliométricas en internet ha supuesto una gran revolución en los temas que se tratan. Tampoco es fácil encontrar bibliografía en Español sobre la materia en general.

Un libro reciente en Inglés que cubre gran parte de lo tratado en la asignatura y muchos más temas relacionados es el de *Johnson-Sheehan* [6]. Más específicos son los libros de *Alley* [7] y *Day & Gastel* [8] que se centran en el proceso de escritura, el libro de *Alley* [9] sobre presentaciones científicas, y el de libre acceso de *Oetiker* y otros [10] sobre LaTeX. Este último también está disponible en Español.

6 CONCLUSIONES

Al igual que muchos investigadores de las universidades españolas, hemos desarrollado a lo largo de los años una actividad investigadora en la que nos hemos enfrentado de manera casi autodidacta con los retos que plantea la publicación exitosa de resultados. Por este motivo, estamos convencidos de que la inclusión de estas competencias en asignaturas de las titulaciones de ciencia y tecnología resultará crucial para mejorar la excelencia de los futuros innovadores.

En este artículo se ha descrito la asignatura "*Sistemas de documentación y elaboración de textos científicos*" impartida en el Máster en Ingeniería Mecatrónica de la Universidad de Málaga. Los objetivos de la asignatura son la formación del ingeniero en la búsqueda, seguimiento y evaluación de publicaciones y la capacitación para el desarrollo de textos científicos.

Se propone la filosofía de aprendizaje basada en proyectos para el desarrollo práctico de la asignatura. En este sentido, se plantea un taller práctico que persigue motivar una actitud crítica, y auto-crítica, en los estudiantes. Para ello, los estudiantes tienen como objetivo realizar un breve artículo sobre el estado del arte de un tema de su elección vinculado al trabajo fin de Máster. Con esta finalidad, se realizan ejercicios que les guiarán en las distintas fases desde la búsqueda y selección de referencias relevantes hasta la elaboración y corrección de un breve artículo.

La evaluación de los alumnos en los pasados cursos evidencia no sólo una mejora destacable en las competencias adquiridas sino un alto grado de satisfacción con la forma de impartir la asignatura. Asimismo, se aprecia una elevada motivación causada en parte tanto por la elección de un tema personalizado de interés como por el aprovechamiento a la hora de elaborar su trabajo fin de máster.

Por otra parte, cabe señalar la ganancia en transversalidad que aporta la asignatura en el contexto del Máster de Mecatrónica. Esto se debe a la participación de todos los alumnos en actividades donde deben evaluar de manera crítica los trabajos de sus compañeros, que pueden versar sobre temas muy variados.

Esta propuesta constituye una experiencia innovadora no sólo por aplicar conceptos de aprendizaje activo a la enseñanza de estas metodologías, sino por abordar de una manera integral, actualizada y práctica el conjunto de competencias que permiten al investigador sacar el máximo partido de su actividad innovadora y, de esta manera, contribuir a la excelencia de las empresas e instituciones donde desarrollen su labor. Asimismo, dado el carácter universal de la asignatura objeto de este trabajo en el campo de las ingenierías y las ciencias, esta propuesta ofrece un alto potencial de transferencia a otras titulaciones.

Agradecimientos

Este trabajo ha sido subvencionado en parte por el proyecto CICYT DPI 2011-22443 y el Proyecto de Innovación Educativa "Aprendizaje basado en proyectos para mejorar la excelencia de publicaciones científicas y transferencia del conocimiento en ingeniería" PIE 13-155 de la convocatoria 2013-2015 de la Universidad de Málaga.

Referencias

[1] Carlos M. Zapata y Juan D. Velásquez, *Algunas pautas para la escritura de artículos científicos*, Ingeniare-Revista chilena de ingeniería, Vol. 16, No. 1, pags. 128-137 (2008).

[2] Omar Sabaj, *Descubriendo algunos problemas en la redacción de Artículos de Investigación Científica (AIC) de alumnos de postgrado*, Revista Signos, Vol. 42, No. 69, pags. 107-127 (2009).

[3] C. L. Dym, A. M. Agogino, O. Eris, D. D.Frey y L. J. Leifer, *Engineering design thinking, teaching, and learning*, Journal of Engineering Education, 94 (1), pág. 103-119 (2005).

[4] *Página web del Máster de Mecatrónica de la Universidad de Málaga*, consultada el 8 de Julio de 2013, www.isa.uma.es/C6/mastermecatronica

[5] *Página web del Grupo de Ética en la Investigación, Comité Español de Automática*, consultada el 8 de Julio de 2013, www.ceautomatica.es/content/grupo-de-etica-en-la-investigacion

[6] Richard Johnson-Sheehan, *Technical Communication Today*, Cuarta edición, Pearson (2012).

[7] Michael Alley, *The craft of scientific writing*, Tercera edición, Springer (1996).

[8] Robert A. Day y Barbara Gastel, *How to Write and Publish a Scientific Paper*, Sexta edición, Greenwood Press (2006).

[9] Michael Alley, *The craft of scientific presentations. Critical steps to succeed and critical errors to avoid*, Springer (2003).

[10] Tobias Oetiker, Hubert Partl, Irene Hyna y Elisabeth Schlegl, *The Not So Short Introduction to LaTeX 2ε*, Versión 5.01 (2011).

simLab: Laboratorio de simulación de la planta Festo

Silvia Alayón[(1)], Norena Martín[(2)], Omar Nuñez[(1)], Francisco Fumero[(1)]

silvia@isaatc.ull.es, nmartin@ull.es, omar@isaatc.ull.es, franfumero@isaatc.ull.es
[(1)]Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores,
[(2)]Departamento de Expresión Gráfica, Universidad de La Laguna
Universidad de La Laguna

Resumen

El objetivo de simLab es permitir la simulación dinámica tridimensional de la Planta Industrial Festo, maqueta de un proceso industrial compuesta por varias estaciones y controlada con autómatas Siemens, que se encuentra físicamente en el Laboratorio de Automatización del Departamento de Ingeniería de Sistemas y Automática y Arquitectura de Ordenadores de la Universidad de la Laguna.

simLab ha sido implementado con Autodesk Inventor, para el desarrollo de los modelos 3D de las estaciones de la planta, y con Ciros Mechatronics, para la simulación del control realizado por los autómatas programables sobre los modelos 3D de la planta.

simLab está actualmente en desarrollo, y será una herramienta importante para la realización de las prácticas de varias asignaturas de automatización impartidas por este departamento.

Palabras Clave: Automatización, Planta Festo, Simulación, Autodesk Inventor, Ciros Mechatronics.

1 INTRODUCCIÓN

Durante las últimas décadas se ha producido a nivel mundial un importante crecimiento tecnológico que ha cambiado y revolucionado la Ingeniería Industrial, y en concreto, la Ingeniería de Procesos, donde la Simulación ha demostrado ser una de las herramientas tecnológicas más útiles [15].

La simulación de sistemas complejos puede ayudar a entender mejor la operación del sistema, a detectar las variables más importantes que interactúan en el mismo y a entender mejor las interrelaciones entre estas variables. Además, en el caso que nos ocupa, la simulación puede ser una herramienta educativa de gran valor [14].

El trabajo presentado en este artículo tiene como objetivo principal implementar simLab, una herramienta que permita la simulación dinámica tridimensional de la Planta Industrial Festo [10], maqueta de un proceso industrial compuesta por varias estaciones y controlada con autómatas Siemens, que se encuentra físicamente en el Laboratorio de Automatización del Departamento de Ingeniería de Sistemas y Automática y Arquitectura de Ordenadores de la Universidad de la Laguna.

Las plantas Festo tienen un alto coste, por lo que la posibilidad de disponer de esta herramienta de simulación abre un enorme campo de posibilidades para utilizarla en tareas docentes para los estudiantes de las distintas asignaturas de Automatización y Control de los grados de Ingeniería Industrial.

La simulación de la planta permitirá predecir el comportamiento del sistema en tiempo real bajo diversas situaciones reales o previsibles y analizar las posibles alternativas sin tener que alterar físicamente la planta. Esto supone un gran avance en cuanto a espacio requerido en el laboratorio, ya que el alumnado podría recurrir a la previa simulación para afianzar sus conocimientos e incluso completar la misma práctica en dicho entorno.

La motivación principal de este proyecto es la creación de nuevos puestos en el laboratorio para que más personas puedan asistir a las sesiones prácticas simultáneamente y se puedan hacer sesiones prácticas de mayor duración y calidad.

Estos nuevos puestos serían virtuales, ya que añadir más estaciones físicas a la planta Festo conllevaría un coste muy elevado además de acarrear dificultades por las limitaciones físicas del espacio disponible. El poder trabajar con estaciones virtuales simuladas permite usar un aula de ordenadores como laboratorio de Automatización en donde el estudiante podrá aprender a programar automatismos del mismo modo que sobre la planta física real del laboratorio.

Por otro lado, el diseño e implementación de simLab ha dado lugar al desarrollo de varios proyectos fin de carrera de alumnos de Ingeniería Técnica Industrial, esp. Electrónica Industrial [6, 9, 11], proyectos

orientados principalmente a los conceptos didácticos de esta titulación como son: el conocimiento de los procesos Industriales, los elementos que lo integran, el diseño y la optimización de conjuntos que conforman el proceso, la programación y el control de autómatas y por último el desarrollo y estudio de software específico para actividades industriales.

Concretamente, simLab ha sido implementado con Autodesk Inventor, para el desarrollo de los modelos 3D de las estaciones de la planta, y con Ciros Mechatronics, para la simulación del control realizado por los autómatas programables sobre los modelos 3D de la planta. En las siguientes secciones se describirá detalladamente el trabajo realizado.

2 DESCRIPCIÓN DE LA PLANTA FESTO Y EL PROBLEMA ABORDADO

La planta Festo está situada en el Laboratorio de Automatización del Departamento de Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores de la Universidad de La Laguna, en la planta 0 del Edificio de Física y Matemáticas. El laboratorio consta de 5 estaciones (figura 1):

1. La estación de distribución
2. La estación de medición de longitud de piezas
3. La estación de procesado de piezas
4. La estación con brazo manipulador
5. La estación de clasificación de piezas

Las estaciones están controladas con autómatas Siemens Simatic S7-200 y S7-300 [12].



Figura 1: Planta Festo del laboratorio

Este proyecto surge al querer resolver el problema de espacio que existe en el desarrollo de las prácticas en las asignaturas de Automatización de los estudios de grado en Ingeniería Industrial. Los alumnos deben esperar turnos semanales para poder acceder a la planta industrial Festo y sólo pueden dedicarle unas horas. En la planta sólo pueden estar diez personas simultáneamente (dos estudiantes por estación). El proceso de programar con compañeros una planta puede resultar complicado, lo que requiere dedicarle más horas de las que por horario de prácticas es posible.

Mediante la creación de simLab se podría contar con las mismas estaciones que existen físicamente en el laboratorio, pero en número ilimitado, ya que el objetivo de simLab es modelar cada estación en 3D y permitir la simulación del control de las estaciones realizado por autómatas, todo ello en un ordenador. Es decir, la idea principal es convertir un aula de ordenadores en la continuación del Laboratorio de Automatizacón.

3 SIMLAB: DISEÑO E IMPLEMENTACIÓN

El desarrollo de simLab comprende dos partes bien diferenciadas:

1. Modelado 3D de las estaciones.
2. Simulación dinámica del control de un autómata sobre el modelo 3D de las estaciones.

3.1 Elección de las herramientas para la implementación de simLab

En primer lugar se analizaron diversas herramientas para elegir las más idóneas para nuestro proyecto: Cosimir [5], Labview [8], ISG-Virtuos [7], Ciros Automation Suite [3] y Autodesk Inventor [1]. Algunas de ellas no permitían fácilmente el modelado 3D, otras sí lo permitían, pero resultaba complicado simular sobre el modelo el control de un autómata. Finalmente se eligieron estas dos herramientas, que nos permitieron abrir dos líneas paralelas en la implementación de simLab:

1. Autodesk Inventor [1], que es un software CAD 3D producido por la empresa Autodesk, el cual se utiliza principalmente para diseñar modelos tridimensionales y simular procesos (figura 2).

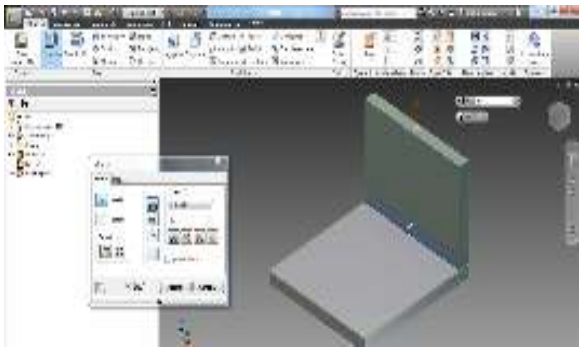


Figura 2: Autodesk Inventor. Entorno de piezas

Este programa permite realizar un modelado virtual tridimensional de toda la planta y una posterior simulación dinámica basada en parámetros de posición y tiempo, pudiéndose tener en cuenta estudios alternativos de resistencia de materiales ante las fuerzas ejercidas entre los distintos componentes al moverse. Así como una creación rápida y eficaz de toda la documentación técnica pertinente y exportación de los planos necesarios para la construcción de la planta. Además de una presentación dinámica e intuitiva de todo el montaje mediante el software Inventor Publisher [2] e Inventor Studio, que ayudará a la mejor comprensión del funcionamiento de la planta y su disponibilidad para utilizarlo como recurso de aprendizaje.

Inventor proporciona soporte para la simulación de movimiento, así como el análisis de elementos finitos, estáticos y modales de piezas, los ensamblajes y los marcos de soporte de carga.

2. CIROS Mechatronics [4], un paquete informático perteneciente a Ciros Automation Suite [3], desarrollado por la empresa alemana Festo, fabricante de la planta que tenemos en el laboratorio. Este programa ya dispone de los modelos 3D de las estaciones de la planta (figura 3).



Figura 3: Ciros Mechatronics

El software simula el comportamiento de un autómata s7-300, permite la introducción de código en KOP o AWL y simula dinámicamente sobre el modelo 3D el resultado de la programación introducida.

La gran ventaja de este software es que dispone de 15 plantas de procesos industriales educativas para simular y estudiar, lo cual aporta numeroso material didáctico de provecho.

En las siguientes secciones se explica detalladamente el desarrollo implementado con cada paquete.

3.2 Primera parte: Modelado 3D de las estaciones con Autodesk Inventor

Para realizar el modelado 3D de la planta Festo con esta herramienta se tomaron medidas exhaustivas de las estaciones y sus piezas y engranajes. Inventor permite el modelado 3D por medio de los siguientes entornos:

1. Entorno de piezas (figura 2): para crear archivos de pieza. Al abrir un nuevo archivo se accede al entorno de boceto automáticamente, en el que los comandos de boceto y operaciones permiten crear, editar, restringir y acotar la geometría para formar piezas. Una vez el boceto terminado, se sale del entorno y se accede al entorno de modelado de piezas. En éste, se le puede dar volumen y forma a la pieza que se quiere crear. Una de las ventajas que proporciona Inventor, es que permite dejar la geometría del boceto sin acotar, dejando la posibilidad de cambiar el tamaño posteriormente.

2. Entorno de ensamblajes (figura 4): permite unir piezas y subensamblajes para formar un único componente, y posteriormente definir las restricciones que controlan la posición y el movimiento de los componentes del entorno.

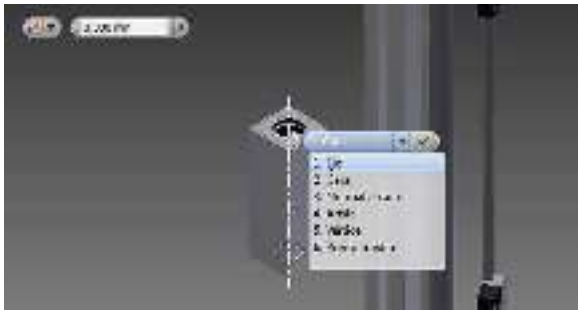


Figura 4: Comando ensamblar - eje

3. Entorno de simulación (figura 5): permite observar el movimiento de los distintos componentes de la planta modelada con una serie de comandos internos.

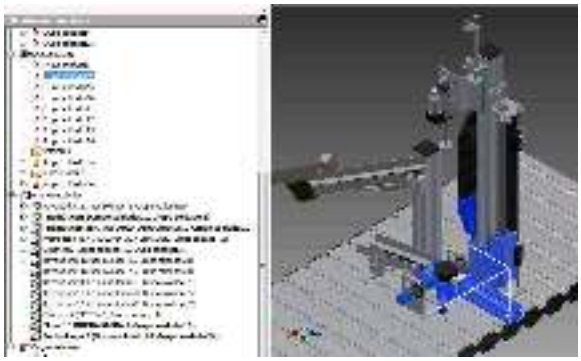


Figura 5: Entorno de simulación

3.3 Segunda parte: Simulación dinámica de modelos 3D con Ciros Mechatronics

El programa CIROS Mechatronics es una herramienta de simulación tridimensional que permite comprender el modo de funcionamiento y la estructura de los modelos de procesos preconfeccionados que incorpora. También da la posibilidad de realizar programaciones sobre un autómata y de practicar la comprobación de las mismas.

Los modelos 3D de procesos que trae incorporados son representaciones de las distintas instalaciones técnicas de automatización de diferente complejidad de la empresa Festo, y son denominados celdas de trabajo.

CIROS Mechatronics contiene un simulador de PLC, el PLC interno, y da la posibilidad de cargar los códigos de programación con extensión ".s7p" y de ejecutarlos para controlar el modelo 3D de la planta. En realidad este simulador puede ejecutar el código de programación en los lenguajes KOP, FUP, AWL y GRAPH creados en STEP 7, aunque nuestro interés está centrado en KOP y AWL.

EL programa CIROS Mechatronics permite visualizar las señales aplicadas en las entradas y salidas del PLC de modo gráfico. Las señales que están a 0 se representan de color rojo y las señales que están a 1 de color verde. Además, si a la hora de ejecutar la simulación no se aprecia bien el desarrollo de la misma desde la perspectiva en la que el usuario se encuentra, se puede cambiar dicha perspectiva.

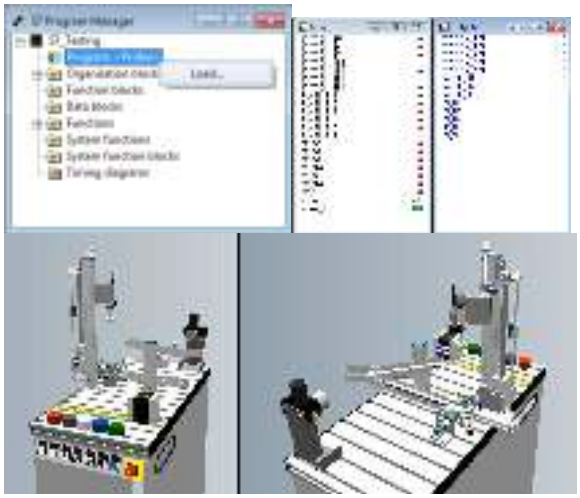


Figura 6: Carga de programa en el simulador del autómata, visualización del estado de entradas y salidas y cambio de perspectiva con Ciros Mechatronics

4 SIMLAB: ESTADO ACTUAL DE DESARROLLO

En estos momentos se han modelado en 3D con Autodesk Inventor y simulado dinámicamente con Ciros Mechatronics tres de las cinco estaciones de la planta Festo: la estación de test de piezas (figura 7), la estación de procesado de piezas (figura 8) y la estación de almacén de piezas. Observando las figuras 7 y 8 se puede observar que los modelos 3D generados con Inventor son mucho más precisos que los ofrecidos por el software Ciros Mechatronics.

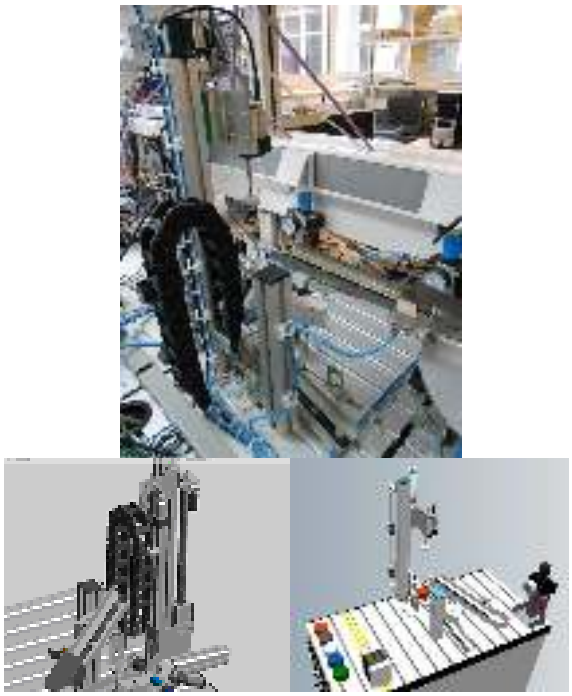


Figura 7: Estación de test de piezas del laboratorio y modelos 3D. Izquierda: con Inventor. Derecha: con Ciros

Adicionalmente, se ha empezado a desarrollar materiales educativos sobre la planta Festo y simLab para móviles y tabletas. Toda este material se muestra en la página web del proyecto simLab (figura 9) [13].

5 CONCLUSIONES

El objetivo de simLab es complementar el actual laboratorio de Automatización del Departamento de Ingeniería de Sistemas y Automática y Arquitectura de Ordenadores de la Universidad de La Laguna, dotando de más puestos al mismo a bajo coste y permitiendo aprovechar las ventajas educativas de la simulación dinámica de sistemas.

El principal objetivo ha sido alcanzado, pero no desarrollado en su totalidad. Falta el modelado y la simulación de dos estaciones de las cinco que componen la planta Festo, y la prueba de la simulación de otras estaciones ofrecidas por el software Ciros Mechatronics diferentes a las existentes físicamente en el laboratorio. Además, los materiales educativos que han comenzado a ser desarrollados deben ser completados con información de más estaciones.

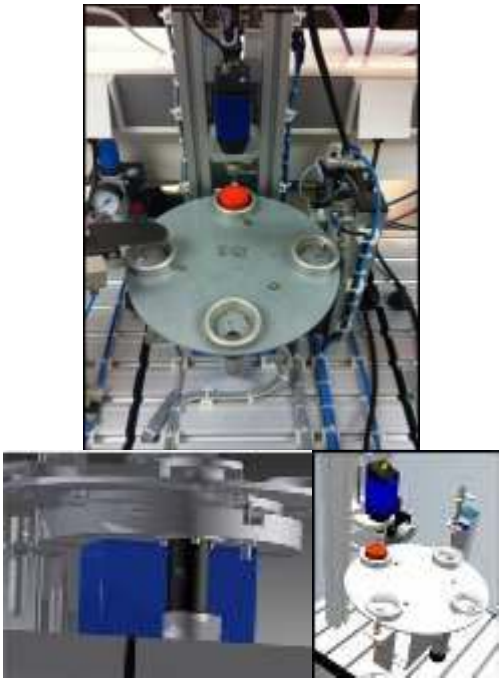


Figura 8: Estación de procesado de piezas del laboratorio y modelos 3D. Izquierda: con Inventor. Derecha: con Ciros



Figura 9: Página web del proyecto simLab [13]

Hay tres líneas abiertas que dan continuidad a este trabajo:

1. Terminar el modelado de las estaciones que faltan de la planta Festo con Inventor y completar así la totalidad de la planta. Aunque las tres estaciones ya desarrolladas serán incorporadas a las prácticas el curso que viene, con el software Ciros Mechatronics, nuestro objetivo a corto plazo es poder completar todas las estaciones con Inventor, dado que los modelos 3D generados son mucho mejores con esta herramienta.

2. Debido a que Ciros Mechatronics es un software de pago, hemos planteado una nueva línea de investigación a medio plazo consistente en el desarrollo de un compilador que permita traducir instrucciones de AWL en movimientos sobre los modelos 3D generados con Autodesk Inventor, comunicándose directamente con el entorno de simulación de Inventor. Si consiguiéramos este objetivo podríamos simular cualquier planta industrial modelada en 3D con Inventor, no sólo las existentes físicamente en el laboratorio, a coste cero.

3. Y a largo plazo nos planteamos la posibilidad de virtualizar el laboratorio y permitir que los estudiantes accedan a los modelos 3D desde sus casas y puedan simular su comportamiento con sus códigos AWL utilizando el compilador descrito en el punto anterior.

Referencias

[1] Autodesk Inventor: http://www.autodesk.es/adsk/servlet/pc/index?id=14569016&siteID=45575. Revisado el 19/12/12

[2] Autodesk Inventor Publisher: http://usa.autodesk.com/adsk/servlet/pc/index?id=14640987&siteID=123112

[3] Ciros Automation Suite: http://www.festo-didactic.com/mx-es/learning-systems/software-e-learning/cirosautomation-suite/cirosmechatronics.htm?fbid=bXguZXMuNTY0LjE0LjE4LjExMTAuNjY5Mw. Revisado el 19/12/12

[4] CIROS Mechatronics. Manual. Disponible en: http://www.festodidactic.com/ov3/media/customers/1100/ciros_mechatronics_manual_es.pdf. Revisado el 19/12/12

[5] COSIMIR. Manual. Disponible en: http://www.provendor.fi/servo/manuals/Cosimir-StartManual.pdf. Revisado el 19/12/12

[6] Cutillas V., García D. (2012) "Simulación dinámica de la Estación de Procesado de Piezas de la Planta Industrial Festo". Proyecto fin de carrera de la titulación Ingeniería Técnica Industrial, esp. Electrónica Industrial de la Universidad de La Laguna.

[7] ISG-Virtuos: http://www.isg-stuttgart.de/index.php?id=50&L=1

[8] Labview - Software de Desarrollo de Sistemas: http://www.ni.com/labview/esa/

[9] Lods S. (2012) "Simulación dinámica de la Estación de Test de Piezas de la Planta Industrial Festo". Proyecto fin de carrera de la titulación Ingeniería Técnica Industrial, esp. Electrónica Industrial de la Universidad de La Laguna.

[10] Planta de automatización didáctica Festo: http://www.festo-didactic.com/es-es/productos/mps-sistema-de-produccion-modular/?fbid=ZXMuZXMuNTQ3LjE0LjIwLjU4NQ

[11] Olave J. J. (2012) "Simulación dinámica de la Estación de Clasificación de Piezas de la Planta Industrial Festo". Proyecto fin de carrera de la titulación Ingeniería Técnica Industrial, esp. Electrónica Industrial de la Universidad de La Laguna.

[12] Autómatas Siemens Simatic S7: http://www.automation.siemens.com/mcms/programmable-logic-controller/en/simatic-s7-controller/Pages/Default.aspx

[13] simLab, página web: http://norenamartin.wix.com/simlab

[14] Suñer, J. L., Mata, V., (2002) "Utilización de Programas de Simulación en la Docencia de la Ingeniería Mecánica", Actas del *X Congreso de Innovación Educativa en las Enseñanzas Técnicas*.

[15] Villegas-Izquierdo F., (1999) "La simulación como herramienta industrial" *Dyna*, vol. 74-5 pp. 9-10

Simuladores 3D y evaluación automática para prácticas de Automatización Industrial

Adolfo J. Sánchez del Pozo, David Muñoz de la Peña, Fabio Gómez-Estern

Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla.
Camino de los Descubrimientos, 41092-Sevilla, Spain

Resumen: El uso de simuladores facilita en gran medida el trabajo de los especialistas ya que permiten prevenir errores en los sistemas reales. Mediante el uso de un simulador para plantas industriales se puede reducir el tiempo de programación de la automatización de las mismas al poder realizar pruebas previas a su implantación. Los simuladores también pueden usarse en el campo de la educación. En este trabajo se presenta una nueva plataforma de simulación de maquinaria industrial, mediante realidad virtual en 3D, para la realización de prácticas de programación de PLC's (programmable logic controllers). Para ello se ha desarrollado un motor físico que se integra en la aplicación y una interfaz gráfica que servirá al usuario para diseñar, programar y visualizar el resultado de la simulación de la planta. Adicionalmente, se conectará el simulador con la plataforma de evaluación automática Goodle GMS, de modo que las prácticas realizadas en el aula o en casa sean calificadas automáticamente mediante un protocolo telemático.. Se muestran los resultados de la utilización de esta plataforma en las prácticas de las asignaturas Automatización y Control Distribuido durante el curso 2012-2013.

Palabras clave: Software de simulación, programación PLC, laboratorios virtuales, sistemas industriales.

1. INTRODUCCIÓN

En los últimos años, la innovación docente en ingeniería se articula en torno a tres aspectos básicos (Dormido et al., 2005; Farias, 2010a): comunicaciones, visualización e interactividad. El último de ellos es la esencia de los videojuegos, una industria que sorprende año tras año con nuevos productos cada vez más realistas. Con el tiempo, los videojuegos mejoran en calidad gráfica, jugabilidad, interactividad, etc. En los videojuegos actuales los usuarios interactúan con los objetos de su entorno; pueden cogerlos, tirarlos y empujarlos para sus propios propósitos. Un ejemplo de esto se da en Half-Life 2, lanzado al mercado en 2004 y que permite una gran interacción con el entorno, ver Figura 1. Si a los videojuegos se les incluye un sistema físico muy realista para hacerlos más atractivos, ¿por qué no hacer lo mismo con las plataformas informáticas de aprendizaje?

El uso de simuladores 3D realistas de plantas industriales, abre una vía para múltiples aplicaciones de entrenamiento, ensayos o evaluación de estrategias de control. Existen actualmente propuestas comerciales que, en un entorno interactivo en 3D, incluyen un sistema físico realista, como ITS PLC de Realgames (www.realgames.pt), en este caso dedicado a la educación en Ingeniería. Este simulador permite ser controlado mediante un autómata o PLC externo, que se conecta al PC que ejecuta la aplicación mediante una tarjeta de entradas y salidas externa, como se muestra en la figura 2 (a y b).



Figura 1: Usuarios de Half-Life 2 cogiendo objetos del escenario.

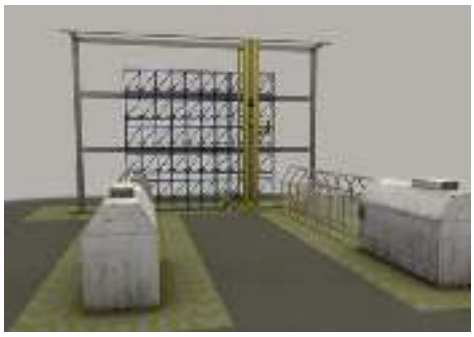


Figura 2a: Simulador industrial ITS-PLC



Figura 2b: Conexionado del autómata para el control del por interfaz física simulador ITS-PLC mismo (E.T.S. de Ingeniería, Sevilla)

Sin embargo, ITS PLC consta de un número limitado de plantas industriales preprogramadas, lo que lo hace bastante rígido para el aprendizaje.

En este contexto, proponemos dar un paso más mediante el desarrollo de un software que reúna las características de ITS PLC y al mismo tiempo permita al usuario definir libremente la estructura de las plantas a simular. Esta aplicación realizará simulaciones de maquinaria industrial en 3D integrando en el mismo un sistema físico para poder observar el resultado de la simulación sobre los objetos con los que interaccionarán los elementos industriales. La simulación se realizará sobre objetos sólidos modelados mediante poliedros. El sistema de colisiones será el responsable de comprobar la interacción entre los diferentes objetos de un escenario. Se presenta la problemática de las colisiones, el sistema físico y simulación de la maquinaria.

Sin embargo, un sistema completo de asistencia a la docencia no se limita a ofrecer plataformas tecnológicas avanzadas de entrenamiento y ensayo. También debe permitir la definición de objetivos de aprendizaje y una metodología de evaluación para el seguimiento del trabajo de los estudiantes. Si este sistema está automatizado, más eficiente y sencilla será su utilización, y más fácil será hacerlo extensivo a contexto no presenciales o casos de grupos numerosos.

La evaluación automatizada es un campo de expansión reciente y en continua evolución. Desde los clásicos test de elección múltiple, han aparecido multitud de herramientas y funcionalidades que amplían el espectro con el fin de posibilitar procesos evaluativos más creativos y personalizados. Ya no solo se pretende evaluar conocimientos, sino tareas más sofisticadas como la resolución de problemas de diseño personalizados que no tienen una solución única. En este sentido la plataforma Goodle GMS ha abierto grandes posibilidades, al poder incluir simulaciones y cálculos complejos sobre una base de Matlab para el cálculo de las notas (Muñoz de la Peña et al., 2012). Aunque esta herramienta se ha concebido inicialmente para una interacción con el estudiante basada texto (dentro de un marco de la arquitectura de la fusión de código fuente), se ha extendido recientemente para hacerla compatible con los laboratorios virtuales. El primer laboratorio virtual Goodle era una plataforma de análisis y control no lineal para pruebas y entrenamiento desarrollado dentro de un proyecto de investigación de control no lineal, (Farias, et al., 2012).

La plataforma desarrollada integra todos los aspectos mencionados, y se ha utilizado para desarrollar un proyecto de fin de curso en las asignaturas Automatización de 5º de Ingeniero de Telecomunicaciones y Control Distribuido, de 3º de Ingeniería Técnica Informática especialidad Sistemas. El sistema incluye la funcionalidad de la *evaluación automática competitiva,* y se ha probado en la asignatura Automatización Industrial del Grado en Ingeniería de las Tecnologías Industriales. Una de las mayores ventajas de la plataforma, es que los alumnos pueden desarrollar su trabajo fuera de la Universidad. La comunicación entre el PLC y la plataforma se realiza mediante OPC.

2. DESCRIPCIÓN DEL SIMULADOR

La propuesta se basa en una interfaz gráfica en la que el usuario pueda diseñar un simulador mediante el uso de los elementos industriales disponibles. Esta interfaz será capaz de ofrecer al usuario la posibilidad de escoger de entre un listado de elementos, posicionarlos, cambiar la orientación, añadir sensores etc. Básicamente será un constructor o como se suele llamar en terminología anglosajona de videojuegos un "WorldBuilder". Posteriormente al diseño del simulador, se podrá compilar para obtener la versión ejecutable del simulador. El simulador obtenido deberá ser capaz de conectarse con un PLC o bien con un simulador de PLC. Para esto se han barajado diferentes opciones. La primera sería realizar una conexión física con un PLC comercial mediante los puertos USB o serie RS232C adaptando las señales mediante una tarjeta de adquisición de datos. La segunda opción para la conexión sería utilizar un servidor OPC (ole process control) como plataforma intermedia para el paso de datos entre PLC-Simulador. Esta solución es abierta y flexible al clásico problema de los drivers propietarios con la ventaja añadida que la mayoría de los grandes fabricantes incluyen OPC en sus productos. Algunos fabricantes de PLCs suministran una aplicación que simula el comportamiento del PLC y que, cuando se ejecuta, constituye un proceso independiente, que utiliza los drivers propios para comunicar con el entorno de programación. Eso permite contar con la ventaja de no necesitarse un PLC comercial para realizar la programación del simulador. Con todo esto un usuario sería capaz de diseñar, compilar, programar y comprobar los errores de una planta industrial, sin necesidad de acceso a la maquinaria real.

Para el primer desarrollo y prueba del prototipo de simulador 3D se ha utilizado Unity Pro XL de Schneider Electric. La razón de utilizar este software ha sido la disponibilidad de algunos Modicon M340 de Schneider Electric en los laboratorios de la Universidad de Sevilla y por lo tanto ya se están estudiando en algunas de las asignaturas de las diferentes Ingenierías. Para el servidor OPC y dado que se está utilizando software de Schneider Electric, se decidió, a priori, utilizar el servidor OPC de Schneider, OFS (OPC Factory Server), mientras que el cliente OPC que se incluye en el simulador 3D es un cliente programado en C# debido a

que debe integrarse perfectamente en la solución de la aplicación. Los aspectos más importantes para la realización de este software, que serán tratados en los siguientes apartados, son el sistema físico, el sistema de colisiones y el desarrollo de maquinaria industrial.

El software desarrollado consta de dos elementos principales. El primero es una aplicación de edición de escenarios donde el usuario (profesor) puede concebir entornos de producción industrial a medida a partir de una librería de elementos típicos (cintas transportadores, grúas, etc). El usuario puede añadir elementos y definir su posición, orientación, y parámetros internos. Este elemento de la aplicación es lo que se conoce como "World Builder" (creador de mundos) en la industria de videojuegos. El creador de mundos virtuales genera el entorno de simulación 3D empleado por los estudiantes.

El simulador 3D es el segundo elemento, y ha sido diseñado con alto grado de interactividad y realismo. Permite al estudiante interaccionar con diferentes elementos, por ejemplo generando fallas o comprobando los controladores diseñados. Además, el simulador 3D permite el control en bucle abierto y cerrado mediante el protocolo OPC (OLE Process Control). Es ésta es una funcionalidad clave ya que a menudo las plataformas de programación de PLCs constan de simuladores con capacidad de comunicación OPC, lo que permite cerrar el bucle de control sin necesidad de un PLC real.

En este artículo se presenta a versión extendida del software de simulación de desarrollo propio del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Sevilla. En esta versión, el interfaz de comunicación gha sido diseñado específicamente para la plataforma de programación Unity Pro XL de Schneider Electric, que está dotada de un simulador para productos diferentes del fabricante, incluido el autómata Modicon M340, el PLC que emplean los alumnos en las Prácticas de automatización Industrial en el Departamento de Ingeniería de Sistemas y Automática. Con este propósito, se ha empleado el servidor OPC Schneider, el OPC Factory Server (OFS), capaz de comunicarse con el simulador de PLC Unity Pro XL PLC automáticamente. El simulador 3D implementa el cliente OPC mediante el lenguaje C#. La siguiente sección presenta los elementos más relevantes del software,.

3. ELEMENTOS DEL SIMULADOR

La simulación de la física se hace indispensable para visualización de un entorno donde existen objetos que se mueven e interaccionan con otros, ver , see Crowe et. at. (2009) and Kim et.al. (2002).. La simulación de la física es una disciplina muy amplia, este trabajo se centra principalmente en la física de objetos rígidos. Lenguajes de programación posibles son C++, C#, y Lua muy utilizado en los videojuegos con soporte para programación orientado a objetos. Debido a que se trabajará en un entorno en 3D se tendrán vectores con tres componentes que permitirán manejar posiciones, aceleraciones, fuerzas, etc. El motor físico con el que se ha desarrollado la aplicación es un motor genérico que dispone de las leyes básicas de fuerzas, velocidades y aceleraciones debido a que está diseñado para el diseño de videojuegos donde no se hace necesario leyes termodinámicas, electromagnéticas o de otro tipo.

La gran mayoría de los motores físicos están basados en las leyes de movimiento de Newton, que describen el comportamiento de una masa puntual. Los objetos, por tanto, poseerán las cuatro características básicas que son velocidad, aceleración, fuerza y masa. Además de las leyes de Newton, se debe simular la interacción de objetos y fuerzas. Por ejemplo, la fuerza de la gravedad estará presente en cualquier entorno de simulación. Otro aspecto importante será el momento de inercia de un objeto de masa uniforme que viene implementado en el motor físico. Para procesar todas las fuerzas que están actuando sobre un mismo objeto en el simulador el motor físico implementa el principio de D'Alambert.

Como los objetos podrán estar tanto en reposo como en movimiento, se deben tener en cuenta los dos tipos de coeficientes de rozamiento, el estático y el dinámico. Estos coeficientes son característicos para cada par de materiales que se encuentran en contacto y dependen de otros factores como la temperatura. Una opción sería usar valores aproximados para ciertos materiales, Aunque el motor físico que se ha integrado incluye la posibilidad de modificar los coeficientes estático, dinámico y elástico, no trae configurados valores para materiales genéricos por defecto. Para este primer prototipo se han utilizado valores para estos coeficientes que permitían un comportamiento realista de los objetos dado el material del que supuestamente estarían construidos.

La dinámica del sólido rígido es un tipo de simulación empleada en videojuegos que dota a los objetos de un comportamiento "físico" aparente, Mirtich (2000), Stewart (2000), y para ello debe contar con un sistema adecuado de gestión de colisiones. Este sistema deberá comprobar si existen colisiones entre cada par de objetos del simulador. Comprobar una colisión supone conocer la geometría de cada objeto y obtener todos los puntos de colisión existentes entre dos objetos. Si existen muchos objetos, esta comprobación puede consumir demasiado tiempo de procesado y ralentizar la simulación. Una solución a este problema es utilizar dos tipos de detectores de colisiones. El primero de ellos haría una localización de los objetos con posibilidad de contacto sin entrar en detalles. El segundo recibiría la lista de objetos y realizaría un chequeo más exacto hallando las colisiones y la información necesaria para que sean tratadas por el motor físico. De las posibles colisiones también se eliminarían aquellas colisiones que se producen entre un par de objetos que no poseen velocidad alguna y se encuentran en reposo. Es decir, sólo se tendrán en cuentan las posibles colisiones entre objetos reposo-activo y activo-activo. En general se debe introducir en el sistema un conjunto básico de figuras geométricas ya que los cuerpos físicos tendrán una forma que será una figura geométrica o en casos más complejos una combinación. Entre las formas básicas destacan el cubo, cilindro y esfera. Con estas tres se podría modelar cualquier objeto de forma aproximada.

El sistema de colisiones se ha programado en C# empleando las librerías XNA y JigLib. Los objetos definidos permiten el diseño de una librería de objetos industriales, mediante modelos sólidos.

4. GENERADOR DE ESCENARIOS

En esta sección presentamos las funcionalidades generación interactiva de escenarios industriales del simulador 3D.

4.1 Editor de escenarios

El propósito del editor es que el usuario pueda diseñar plantas industrial según sea necesario. Por lo tanto, el software desarrollado proporciona una característica muy interesante de escalabilidad. El editor también incluye todos los parámetros y opciones necesarios cuando se realiza la simulación de elementos industriales, tales como el tiempo de uso de dispositivos, la probabilidad de fallo o el tiempo medio entre fallos. Todos estos parámetros configurables por el usuario permiten un uso más eficiente y una larga vida del software. La Figura 7 muestra una captura de pantalla del editor.

El editor de escenarios se basa completamente en el simulador desarrollado. Con esta aplicación el usuario es capaz de seleccionar, posicionar y ajustar los parámetros deseados de los elementos de la planta. Proporciona al usuario un entorno de programación basado en formularios de edición de Windows con una ventana principal donde se puede ver el resultado de lo que se está diseñando. El concepto de editor de escenarios es un concepto interesante en el desarrollo tanto de los simuladores educativos y de desarrollo comercial como para aplicaciones para el control, la automatización y optimización de procesos..



Fig. 3. Cinta transportadora.



Fig. 4. Mesa giratoria.

4.2 Escenario empleado en Automatización Industrial

La planta prototipo que se ha desarrollado para la asignatura de Automatización Industrial incluye algunos de los dispositivos más utilizados en la industria como pueden ser cintas transportadoras, grúa y pistones. En la Figura 8 se muestra una captura del escenario. El objetivo de la planta es la clasificación y transporte de 3 tipos diferentes de cajas. La clasificación se realiza por tamaño y para ello se hará uso de un conjunto de sensores de barrera a diferentes alturas para su clasificado. El objetivo de la Grúa es tomar las cajas medianas y pasarlas de una cinta transportadora a otra, mientas que las cajas grandes y pequeñas pasaran por la cinta giratoria que las enviará a sus correspondientes rampas de destino. El escenario es conceptualmente simple, pero en él se puede realizar una programación del PLC suficientemente compleja para los alumnos como la necesaria para obtener un rendimiento óptimo del funcionamiento de la planta. En este rendimiento se puede considerar, por ejemplo, el número de cajas procesadas por hora, número de cajas procesadas correctamente, errores producidos, tiempos de funcionamiento de los dispositivos para calcular consumos, etc.



Fig. 5. Grúa.



Fig. 6. Sensores de barrera.

A partir de estas ideas, se ha desarrollado una extensión del simulador que envía a un servidor la información del trabajo realizado por los alumnos, a través de un sistema de login y llamada a procedimiento remoto XML, con el fin de evaluar el rendimiento y establecer calificaciones competitivas. Este aspecto se describirá en la siguiente sección..

. 5. ARQUITECTURA DE EVALUACIÓN AUTOMÁTICA

La siguiente componente del simulador es totalmente novedosa y requiere la interacción de cuatro procesos informáticos distribuidos:

- El simulador industrial, ya descrito.
- El simulador de PLC (inclido en Unity Pro de Schneider).
- El servidor OPC, sincronizando los dos procesos anteriores.
- Un servidor de evaluación automática Goodle GMS para recoger telemáticamente informes de desempeño generados por el simulador industrial.

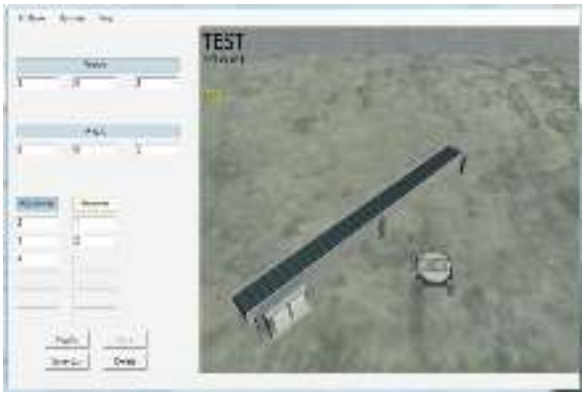


Fig. 7. Interfaz de usuario del generador de escenarios.



Fig. 8. Planta creada mediante el generador de escenarios.

La finalidad de esta implementación es la dotar de capacidad de calificación totalmente automatizada de las prácticas realizadas por los alumnos en el laboratorio de Automatización.. La última versión del simulador implementa un servicio integrado capaz de registrar los eventos importantes que suceden en el transcurso de la simulación, y enviarlos por red al servidor Goodle GMS. Este último consiste en una plataforma telemática de e-learning especializada en la evaluación de prácticas de ingeniería. Goodle GMS emplea una conexión interna de servidor con Matlab. (Muñoz et al., 2012) para ampliar su capacidad computacional.

El motor de la evaluación automática en Goodle GMS sigue un nuevo paradigma basado en una arquitectura de fusión de código fuente. Esta arquitectura permite puntuar los trabajos de los alumnos de disciplinas científicas, siempre que éstos produzcan resultados cuantitativos y estén codificados en un formato compatible con Matlab (o traducidos automáticamente a Matlab mediante un middleware específico). En el funcionamiento básico de Goodle, el alumno entrega por web unas líneas de código, y éstas se ejecutan posteriormente en el servidor en secuencia con otro fragmento de código proporcionado por el profesor. El código Matlab del profesor comprueba las variables generadas en el *workspace* tras la ejecución del código del alumno, y en función de ello se genera una nota que es almacenada en una base de datos. (Fig. 8).

Aunque pueda parecer que esta lógica está limitada a cursos de programación o en los que los alumnos son programadores, en realidad las aplicaciones son más amplias. El código fuente puede ser visto como una manera universal de formalizar la solución a un problema numérico, sin restricciones de formato, longitud o estructura propias de las herramientas clásicas de e-learning, como sucede en las pruebas tipo test.

De hecho, el código fuente se emplea aquí para añadir máxima flexibilidad al interfaz del estudiante. Esto es así porque la manera más rápida de implementar un interfaz web de entrega de soluciones de ejercicios es mediante un campo de texto que el estudiante completa con asignaciones

numéricas o alfanuméricas, por ejemplo, en lenguaje Matlab, con respuestas específicas como en el ejemplo siguiente:

```
R1=10.5;     % respuesta numérica
C1=3.8;      % respuesta numérica
f1=cos(x1); % respuesta funcional
respuesta3='si'; % respuesta textual
```

Además, al estar la lógica de la evaluación determinada exclusivamente por el código proporcionado por el profesor, ésta puede incluir operaciones de gran complejidad, como la resolución de ecuaciones, simulaciones numéricas, etc, haciendo posible, como caso notorio, la verificación funcional de los algoritmos de control continuo.

5.1 Conexión del simulador industrial con Goodle GMS

La principal diferencia con respecto al comportamiento estándar de Goodle GMS descrito en (Muñoz et al., 2012) reside en el hecho de que el navegador desaparece del interfaz entre el alumno y el servidor Goodle. En su lugar, el estudiante no será siquiera consciente de la existencia del servidor (salvo para registrarse al comienzo del curso), ya que los datos se intercambian entre el simulador y Goodle GMS; de forma transparente al usuario.

El conector Goodle se muestra en el simulador como una serie de opciones de menú. Su función es registrar los eventos de simulación durante un tiempo fijo, generar y enviar al servidor Goodle GMS un informe sobre el rendimiento del sistema de control diseñado por el alumno al interaccionar con el simulador. En esencia, este informe consiste en una lista de eventos que tienen lugar dentro de una ventana temporal fija, junto con los instantes de tiempo en que se producen. De hecho, el hecho más relevante para la evaluación automática es la llegada de piezas o paquetes a su correspondiente *puerta de salida* en la factoría.

Una *automatización correcta* es aquella en la que todos los paquetes de un tipo específico que lleguen al destino asignado a dicho tipo. Es decir, el *rutado* es correcto, cosa que se puede verificar sencillamente a partir del informe de simulación. Alternativamente, entendemos por *automatización eficiente*, aquella que es correcta y además registra tiempos de rutado relativamente cortos, cosa que se puede favorecer mediante estrategias de simulación adecuadas como son la paralelización de los procesos. De nuevo, disponiendo de un informe adecuado de instantes de llegada de piezas se puede evaluar la eficiencia de forma sencilla.

Para hacer esto posible, el estudiante debe seguir unos pasos del tipo "grabar y enviar" a partir del menú del conector Goodle en el simulador. Los pasos, concretamente, son *diseño local, simulación supervisada y envío al servidor.*

Mientras que el *diseño local* es la actividad típica de laboratorio con este tipo de simuladores, sin evaluación automática, la *simulación supervisada*, es la ejecución de el simulador en conexión con el controlador del alumno, en un modo restringido para garantizar la homogeneidad de condiciones para los alumnos que van a ser evaluados. La simulación supervisada se ejecuta sin posibilidad de que el usuario interfiera en los procesos (el interfaz gráfico está parcialmente bloqueado), durante una ventana de tiempo fija, igual para todos los alumnos. En este modo, los eventos del mundo simulado son grabados en memoria para generar el informe a su término.

Al final de la *simulación supervisada*, el informe se genera automáticamente en el simulador y se envía al servidor Goodle GMS. Con este fin, el estudiante selecciona la opción *envío de informe* en el menú, y se abre una ventana de diálogo solicitando los siguientes datos:

- URL del servidor Goodle GMS.
- ID del estudiante y clave (tal y como aparece registrado en Goodle)
- ID del ejercicio Goodle (tal y como aparece en el interfaz del profesor, que ha creado y configurado una tarea específica para interactuar con el simulador)..

5.2 Estructura y evaluación automática del informe de rendimiento-

Tras la confirmación de los valores del formulario, el informe de rendimiento se envía por el conector a Goodle GMS en forma de mensaje XML, organizado en dos partes: una cabecera (con los datos de login y del ejercicio), y una lista de descripciones de eventos en forma de pares nombre-valor. Al llegar éstos al servidor, los pares nombre-valor se transforman en *sentencias de asignación Matlab,* como ilustra el ejemplo siguiente:

Extracto del mensaje XML de envío:

```
<variables>
<packets>
[1,2,1,4,2,4,1]
</packets>
<stations>
[1,2,1,1,2,1,1]
</stations>
<times>
[10.12,16.18,19.03,21.57,25.79,27.19,32.44]
</times>
</variables>
```

Asignaciones Matlab generadas en el servidor:

```
packets =[1,2,1,4,2,4,1];
stations =[1,2,1,1,2,1,1];
times =[10.12,16.18,19.03,21.57,25.79];
```

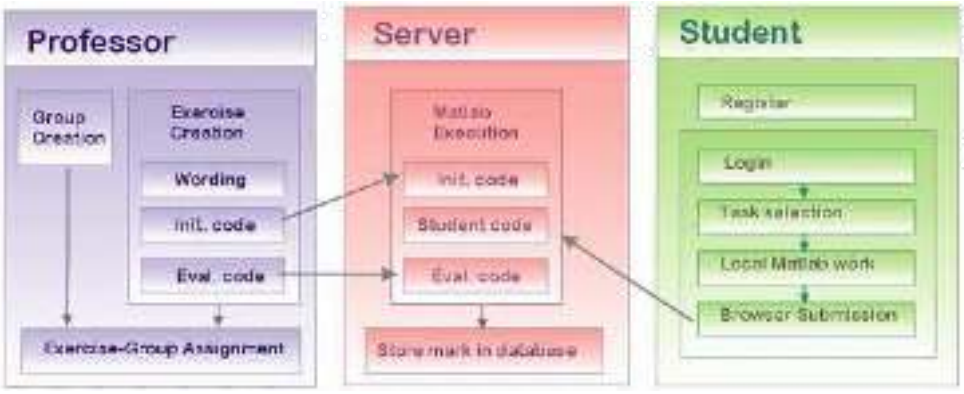


Fig. 9: Ilustración de la arquitectura de fusion de código de Goodle GMS

Como se aprecia en el ejemplo, el envío consta de tres vectores que definen los eventos de llegada, concretamente *packets* (la lista de tipos de caja o paquetes que han ido llegando a los puntos de salida del sistema), *stations,* indicando a qué estación de salida llegó exactamente cada paquete), y *times,* conteniendo los tiempos de los eventos de llegada. En este caso particular, las cajas de tipos 1 y 2 han sido rutadas adecuadamente a las puertas de salida 1 y 2, mientras que los paquetes de tipo 4 han sido rutados erróneamente hacia la puerta 1, dado que en el enunciado se especificó que éstos debían ser rutados a la puerta 4.

Los informes enviados con este formato se almacenan en el servidor de evaluación automática (en forma de código Matlab). Se organizan en una base de datos que contiene código específico de pre y post evaluación para cada ejercicio. Al finalizar el plazo de entregas, el profesor debe lanzar el proceso de evaluación por lotes para todos los alumnos de un curso en un ejercicio particular.

Para cada envío de alumno, el código enviado se funde con el de pre y post evaluación como se muestra en la figura 9, con el fin de obtener automáticamente una calificación y almacenarla en la base de datos. El servidor, además, implementa una opción de evaluación *competitiva*, donde la nota de cada estudiante se calcula de acuerdo con su rendimiento relativo respecto al resto de la clase. Esta comparación se basa en una figura de mérito de libre elección por el profesor. En el caso particular de la asignatura Automatización Industrial, la figura de mérito comparación es el tiempo medio de llegada de un número fijo de paquetes que atraviesa el sistema.
Los detalles adicionales sobre las tareas realizadas del lado del servidor pueden ser consultados en artículos previos sobre Goodle GMS tales como (Farias, et al, 2012), donde la arquitectura de conversión XML-Matlab se propueso inicialmente, en un contexto totalmente diferente (Laboratorios Virtuales).
.

6. EXPERIENCIAS EN EL AULA

Por parte del profesorado, la posibilidad de ofrecer a los alumnos una plataforma basada en programas de libre distribución o licencia gratuita fue muy bien acogida, ya que permite complementar la formación práctica del alumno. La evaluación de los alumnos en función del tiempo de procesado despertó mucho interés por ambas partes, los alumnos en particular la percibieron como un reto al competir contra sus compañeros.

Durante el curso 2012-13 se ha utilizado el Simulador 3D en la asignatura Automatización Industrial de tercero del Grado de Ingeniería en Tecnologías Industriales. Esta asignatura tiene 86 alumnos que hacen sus prácticas con PLC de la serie M340 de Schneider en los laboratorios del Departamento de Ingeniería de Sistemas y Automática. Para complementar las prácticas de la asignatura, se ofreció la posibilidad de realizar un proyecto voluntario desarrollando un programa para controlar el sistema de clasificación de cajas del simulador en 3D. El problema de automatización es el mismo utilizado durante el curso 2011-12 en la asignatura Control Distribuido, clasificar las cajas minimizando el tiempo.
Se proporcionó a los alumnos un paquete con los programas con licencia de prueba o gratuita necesaria para el desarrollo del proyecto con sus ordenadores, incluyendo Unity Pro V6 versión universidad y un servidor OPC de prueba.
Participaron en la experiencia 5 alumnos, los cuales resolvieron el problema utilizando las herramientas de programación estudiadas en el curso. Para la entrega del proyecto, se preparó una versión que automatizaba la ejecución de la prueba, con una secuencia determinista de cajas que bloqueaba la posibilidad de acción manual sobre el sistema. Al final de la ejecución del programa, el Simulador 3D envía a Goodle GMS los resultados de la simulación. La entrega se realizó en el aula de ordenadores del departamento. La evaluación se realizó utilizando la opción *competitiva* descrita en el apartado anterior. Mediante esta opción el sistema ordena los alumnos según el tiempo utilizado en clasificar 10 cajas. Para esta prueba, se añadieron además los resultados de los alumnos de Control Distribuido del curso 2011-12, siendo finalmente el ganador un alumno de esta asignatura.

La experiencia fue muy positiva. Los alumnos que participaron mostraron un gran interés, dedicando un gran esfuerzo en la instalación de la plataforma de prácticas, ya que los programas desarrollados presentaban problemas de compatibilidad con ciertas versiones del sistema operativo Windows.



Figura 10: Clase práctica de instalación y uso del simulador 3D

Al igual que en el curso anterior, los mayores problemas de compatibilidad los presentó el servidor OPC utilizado, necesitando una solución alternativa para ofrecer una plataforma de prácticas suficientemente robusta para ser utilizada con diferentes sistemas operativos en grupos de alumnos numerosos. Es importante hacer notar, que la plataforma es operativa una vez instalada, por lo que se puede utilizar sin problemas en aulas de ordenadores. En particular, en el aula de computadores del Departamento de Ingeniería de Sistemas y Automática se instalaron 21 puestos de prácticas con PLC basadas en el simulador 3D, siendo una alternativa real para realizar prácticas de laboratorio.

7. CONCLUSIONES

Se ha descrito un simulador 3D industrial de diseño propio destinado a la realización de prácticas de programación PLC. Esta aplicación incluye un motor físico, se puede comunicar con PLCs reales mediante protocolos OPC y permite verificar los diseños de controlado0res de los alumnos en tiempo real. La aplicación incluye un generador de escenarios con una librería de elementos industriales típicos.

La aplicación ha sido ampliada recientemente para dotarla de evaluación automática del trabajo del alumno, tanto para evaluación individual como competitiva. Esto se ha logrado mediante una conexión con Goodle GMS, una plataforma de e-learning especializada en calificación automática en disciplinas científicas y técnicas.

Como trabajo futuro se considera el desarrollo de nuevos elementos para la librería del generador de escenario, con un GUI más completo, y un servidor OPC que permita la comunicación con cualquier PLC industrial. Asimismo se está estudiando la posibilidad de controlar el simulador por interfaz física a través de una tarjeta de conversión analógico-digital, dotando de más realismo al montaje y al mismo tiempo de más versatilidad y simplicidad de implementación.

El conjunto ha sido puesto en práctica por primera vez en 2013 en una asignatura de Automatización Industrial, donde se han puesto de relieve las debilidades y fortalezas de la herramientas, logrando una evaluación de tipo competitivo totalmente automática y satisfactoria para un grupo reducido de estudiantes.

AGRADECIMIENTOS

Los autores desean expresar su agradecimiento al Ministerio de Economía y Competitividad por su financiación parcial de este trabajo bajo el proyecto DPI2009-09961.

REFERENCIAS

Farias, G. (2010a), *Adding Interactive Human Interfaces to Engineering Software*. Ph. D. Thesis, Computer Science and Automatic Control Dept., UNED.

Dormido, S., Esquembre, F., Farias, G., and Sánchez, J. (2005). Adding interactivity to existing Simulink models using easy java simulations. *In Proceedings 44th IEEE CDC-ECC'05*, pp. 4163–4168.

Crowe, C. T., Elger, D. F., Williams, B. C., Roberson, J. A.(2009): Engineering Fluid Mechanics, 9th Edition. John Wiley & Sons, Inc. Chapter 11. Sections 11.2, 11.3.

Kim, Y. J., Otaduy, M. A., Lin, M. C., Manocha, D. (2002): Fast Penetration Depth Computation for Physically-based Animation. Symposium on Computer Animation - SCA , 2002.

Millington, I. (2007): Game Physics engine development. Morgan Kaufmann Publishers. Elsevier Inc. pp.69-71, 111-119.

Mirtich, B. (2000): Timewarp rigid body simulation. In Proc. SIGGRAPH 2000, 193–200.

Stewart, D. E. (2000): Rigid-body dynamics with friction and impact. SIAM Review Vol. 42, No. 1, 3–39.

Schneider Electric. Unity Pro. Program Languages and Structure. Reference Manual. April 2009.

Muñoz de la Peña, G., Gómez-Estern, F., Dormido, S. A new Internet tool for automatic evaluation in Control Systems and Programming, Computers & Education, Volume 59, Issue 2, September 2012, Pages 535-550.

Farias, G., Gomez-Estern, F., De la Torre, L., Muñoz de la Peña, D. Sánchez, C., Dormido, S. Enhancing Virtual and Remote Labs to Perform Automatic Evaluation. IFAC Conference on Advances in Control Education, Nizhny Novgorod, Russia, 19-21 Junio 2012.

HERRAMIENTAS DE HARDWARE Y SOFTWARE LIBRE PARA LA IDENTIFICACIÓN EXPERIMENTAL, EL DISEÑO Y LA IMPLEMENTACIÓN DE CONTROLADORES PID

Sanchis Llopis, Roberto
Dep. Enginyeria de Sistemes Industrials i Disseny, Universitat Jaume I email: rsanchis@uji.es
Estupiña Ariño, Silvia
Dep. Enginyeria de Sistemes Industrials i Disseny, Universitat Jaume I

Resumen

En este trabajo se presenta un conjunto de herramientas basadas en hardware y software libre, concretamente Arduino, Processing e Easy Java Simulations, para realizar el ciclo completo de identificación experimental de procesos y diseño e implementación de controladores PID. La plataforma hardware es una tarjeta Arduino UNO conectada por USB a un ordenador personal. Los elementos de software constan del firmware de la tarjerta Arduino, una aplicación desarrollada en Processing que corre en el PC y se comunica con la tarjeta Arduino, y 3 aplicaciones para PC desarrolladas en Easy Java Simulations: una para la identificación experimental, otra para el ajuste experimental de PID y otra para el diseño de PID basado en modelo mediante la respuesta en frecuencia. Las herramientas están disponibles en https://sites.google.com/a/uji.es/freepidtools. Como ejemplo se desarrolla con detalle la identificación y el control de velocidad y posición en un motor de continua.

Palabras Clave: PID, identificación experimental, ajuste experimental, software libre, hardware libre.

1 INTRODUCCIÓN

Las plataformas hardware y software para la identificación, diseño e implementación de controladores basadas en software y hardware propietario tienden a tener un coste elevado y a ser difíciles de adaptar a las necesidades específicas de la docencia planteadas por cada profesor. Un ejemplo es la utilización de software como Matlab o Labview con tarjetas de adquisición de datos para bus PCI. La aparición de plataformas para el desarrollo de software libre basado en Java, como Easy Java Simulations ([4]), o Processing ([3]), y de plataformas de hardware libre de bajo coste como Arduino, permiten desarrollar con un esfuerzo mínimo herramientas prácticamente gratuitas y que permiten trabajar con procesos diversos utilizando interfaces bastante más versátiles que las que se tienen en una tarjeta de adquisición de datos de propósito general, y que se limita a tener entradas y salidas analógicas y digitales. La plataforma Arduino Uno ([2]) permite, por ejemplo, tratar directamente señales de un codificador incremental, o señales en frecuencia, o bien atacar amplificadores conmutados con señales PWM.

El presente trabajo describe un conjunto de herramientas que inicialmente se desarrollaron de forma independiente, pero que se han modificado para formar un paquete completo que permite llevar a cabo el ciclo completo de diseño de un controlador PID: realización de un experimento en la planta, identificación experimental del proceso, ajuste experimental de un PID, diseño de un PID basado en el modelo identificado y prueba del PID diseñado en la planta real.

La estructura del artículo es la siguiente: en las secciones 2 y 3 se describen con detalle las herramientas desarrolladas, en la sección 4 se describen dos ejemplos, y finalmente en la sección 5 se resumen las conclusiones.

2 APLICACIÓN DE INTERFAZ PARA ARDUINO

La parte del sistema desarrollado que permite interaccionar con el proceso real está formada por una tarjeta Arduino Uno programada con un firmware apropiado, que se comunica por USB con una aplicación que corre en un PC, desarrollada en el entorno Processing.



Figura 1: Sistema desarrollado

El firmware desarrollado para la tarjeta Arduino incluye una rutina de interrupción periódica asociada a un temporizador, cuyo periodo es configurable. Esa rutina es la que implementa el muestreo periódico y el cálculo del controlador. Se puede configurar en tres modos de funcionamiento: en bucle abierto (para

realizar experimentos con cambios de la entrada en forma de escalón), en bucle cerrado con realimentación por relé (para realizar experimentos de realimentación a relé para el ajuste experimental), y finalmente en bucle cerrado con controlador PID (para implementar un controlador PID diseñado). La acción de control es una salida PWM cuya resolución y frecuencia también son configurables. La salida del proceso puede configurarse entre tres opciones: entrada analógica, entrada de captura, y entrada de codificador incremental. En el caso de la entrada analógica, la resolución es de 10 bits, mientras que tanto la entrada de captura como la del codificador se implementan por medio de las dos entradas de interrupción externa del Arduino. En las interrupciones asociadas se mide el periodo de la señal por medio del temporizador, o se incrementa (o decrementa) el contador del codificador, según corresponda. Todos los parámetros se configuran enviando mensajes desde el PC a través del puerto serie: marcha/paro, periodo de muestreo, parámetros del PID, parámetros del relé, modo de funcionamiento (bucle abierto, relé ó PID), tipo de entrada, frecuencia y resolución del PWM, valor de la referencia (en bucle cerrado), valor de la entrada (en bucle abierto). Además, cada periodo de muestreo, la tarjeta Arduino envía al PC a través del puerto serie los valores de la señal de entrada y de la salida medida, de forma que la aplicación del PC puede visualizar las señales y/o guardarlas en fichero.

La aplicación de interfaz con el usuario se ha desarrollado mediante el entorno Processing para Windows. Además de permitir configurar la tarjeta para realizar experimentos en bucle abierto o en bucle cerrado (con relé), o implementar un controlador PID, permite visualizar las señales y guardar en fichero tanto los datos de entrada salida, como los parámetros del controlador PID. La grabación de datos en fichero se inicia y se para pulsando un botón. También permite leer un fichero con los parámetros de un controlador PID diseñado con otra de las herramientas que se describirán en la sección siguiente. Los formatos de los ficheros son compatibles con esas herramientas, lo que contribuye a formar un sistema de desarrollo completo.



Figura 2: Aplicación de interfaz para Arduino

3 APLICACIONES JAVA DE IDENTIFICACIÓN, AJUSTE EXPERIMENTAL Y DISEÑO DE PID EN FRECUENCIA

Para completar el sistema que cubre el ciclo completo de diseño de controladores PID se han desarrollado tres aplicaciones en el entorno Easy Java Simulations. La primera de ellas permite la identificación de un modelo del proceso a partir de la respuesta ante entrada escalón. Los datos experimentales se leen de un fichero guardado con la aplicación de interfaz para Arduino descrita en la sección anterior. Como salida de la aplicación de identificación se guarda en un fichero el modelo de la planta. La segunda aplicación permite leer ese fichero con el modelo del proceso, y diseñar un controlador PID adecuado. El diseño del PID se basa en la metodología descrita en [1], que utiliza la respuesta en frecuencia y maximiza la ganancia integral fijando un margen de fase determinado, y un margen de ganancia superior a un valor mínimo. Finalmente, como alternativa a las dos aplicaciones anteriores, la tercera aplicación sirve para realizar directamente el ajuste experimental del controlador PID a partir de los datos de los experimentos (tanto de respuesta ante escalón en bucle abierto, como de realimentación a relé) obtenidos con la tarjeta Arduino.

3.1 APLICACIÓN DE IDENTIFICACIÓN

Esta aplicación permite identificar el modelo del proceso a partir de la respuesta ante escalón. La herramienta permite ajustar un modelo que, en el caso más general, tiene la estructura siguiente:

- Si no se incluye un integrador:

$$G(s)=\frac{k(1\pm\beta s)e^{-Ts}}{(1+\tau_1 s)(1+\tau_2 s)(1+\tau_3 s)(1+2\xi\omega_n^{-1}s+\omega_n^{-2}s^2)} \quad (1)$$

- Si se incluye un integrador:

$$G(s)=\frac{k(1\pm\beta s)e^{-Ts}}{s(1+\tau_2 s)(1+\tau_3 s)(1+2\xi\omega_n^{-1}s+\omega_n^{-2}s^2)} \quad (2)$$

La figura 3 muestra la ventana principal de la aplicación. En la zona B se seleccionan los términos deseados del modelo (polos y/o ceros). En la zona A se ajusta manualmente el valor de los polos y ceros elegidos, así como del retardo. La zona D muestra una gráfica con los datos experimentales y la salida del modelo superpuesta. La zona C permite abrir una ventana para importar los datos experimentales y seleccionar el rango deseado (figura 4). En la zona F se muestra el error entre los datos experimentales y la salida del modelo, así como un selector para abrir una ventana que permite guardar en un fichero el modelo identificado, tanto en un formato binario, para ser leído por la aplicación de diseño de PID, como en un formato de texto compatible con Matlab. El ajuste manual de los distintos parámetros es muy interesante para que los estudiantes relacionen los

valores de polos y ceros con la respuesta de los sistemas. Un procedimiento sistemático para sistemas sobreamortiguados consiste en ajustar en primer lugar un modelo con un solo polo real y retardo, de forma que se ajuste bien la parte superior de la respuesta ante escalón, dejando la parte inferior de la respuesta mal ajustada. Después, según lo pronunciada que sea la "ese" de la respuesta, se aproxima el retardo por una de las siguientes opciones:

- Un polo real de constante de tiempo igual al retardo.
- Dos polos reales de constante de tiempo igual a la mitad del retardo.
- Tres polos reales de constante de tiempo igual a la tercera parte del retardo.
- Un cero positivo si la respuesta es de fase no mínima.

Si la respuesta es inicialmente muy plana, se puede usar cualquiera de las opciones anteriores, pero solo convirtiendo en polos una parte del retardo, dejando la otra parte del retardo como tal.

Una vez la respuesta es bastante aproximada se termina de perfeccionar el modelo realizando un ajuste fino de los parámetros.

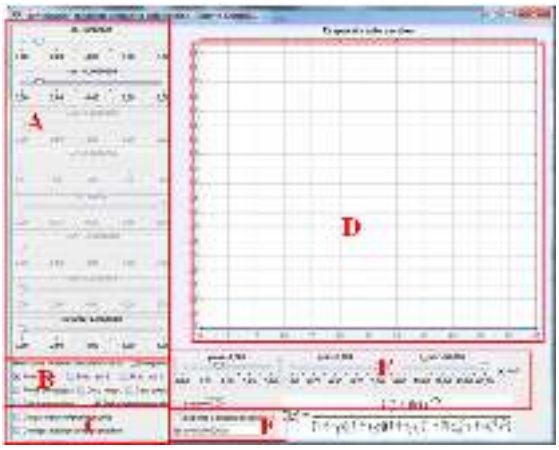


Figura 3: Ventana de la aplicación de identificación

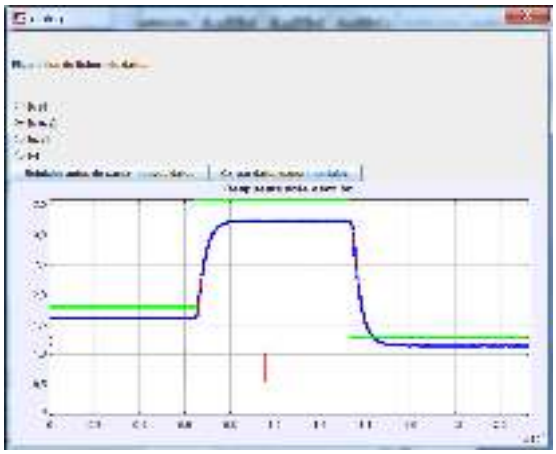


Figura 4: Ventana de carga de datos y selección del rango de la aplicación de identificación

3.2 APLICACIÓN DE DISEÑO DE PID EN FRECUENCIA

Esta aplicación permite diseñar un controlador PID siguiendo el procedimiento propuesto en [1]. El método se basa en maximizar la ganancia integral fijando un margen de fase igual al requerido, y un margen de ganancia mayor o igual que un límite inferior prefijado. La maximización de la ganancia integral es próxima a la minimización del IAE si la respuesta no es demasiado oscilatoria, lo cual se garantiza con los márgenes de fase y ganancia. Para resolver la optimización se utiliza un parámetro auxiliar adimensional, que es el cociente entre el cero del controlador y la frecuencia de cruce de ganancia. La aplicación permite cambiar el valor de ese parámetro con una deslizadera, calculándose de forma automática el controlador para cada valor del mismo. El procedimiento para utilizar la aplicación es como sigue:

1. Se carga el modelo del proceso desde un fichero (guardado desde la aplicación de identificación). También se puede introducir manualmente como una función de transferencia.
2. Se fija el valor deseado del margen de fase y el valor mínimo requerido del margen de ganancia, así como el parámetro *N* del filtro del derivador.
3. Se selecciona el tipo de controlador (PI, PD ó PID).
4. Se arrastra la deslizadera del parámetro de ajuste ("*a*") hasta que la ganancia (*Kc*) sea máxima, pero garantizando que el margen de ganancia sea mayor o igual que el mínimo requerido.
5. Se abre la ventana de respuesta temporal, y se ajustan los factores de ponderación de la referencia *b* y *c*, si es necesario, observando la respuesta ante escalón. Si la respuesta no es adecuada, se vuelve al punto 2.
6. Se guarda el controlador diseñado en un fichero, para poder cargarlo posteriormente en Arduino.

La aplicación también permite cambiar los parámetros del PID directamente, de forma manual.

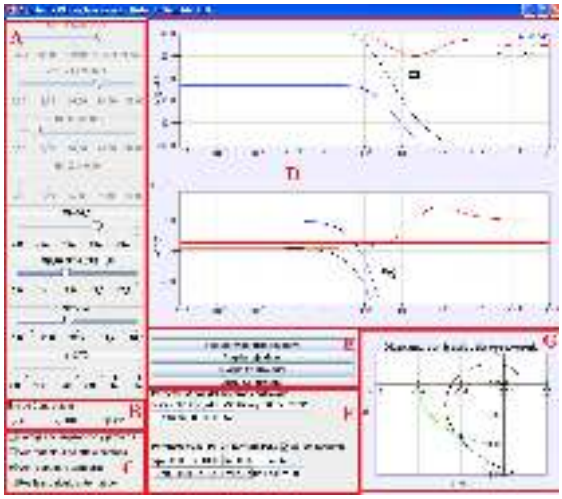


Figura 5: Ventana de la aplicación de diseño de PID



Figura 6: Ventana de respuesta temporal de la aplicación de diseño de PID

3.3 APLICACIÓN DE AJUSTE EXPERIMENTAL DE PID

Como alternativa a la identificación de un modelo de la planta, y el ajuste de un PID basado en ese modelo, la aplicación de ajuste experimental permite ajustar un PID de forma muy sencilla a partir de los datos de un experimento, bien en forma de entrada escalón en bucle abierto, bien en forma de realimentación a relé. Los dos experimentos se pueden realizar mediante la tarjeta Arduino, con el sistema desarrollado, importándose después los datos guardados desde la aplicación de ajuste. La ventana principal de la herramienta es la que sigue:

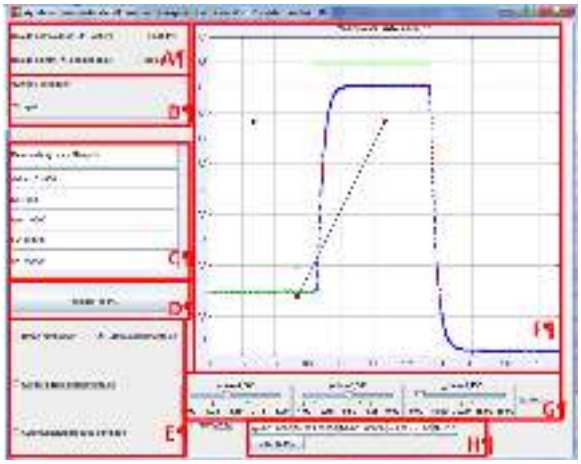


Figura 7: Ventana de la aplicación de ajuste experimental de PID

La aplicación permite seleccionar un controlador PI o PID, y el tipo de ajuste a partir de la respuesta ante escalón o de la respuesta en bucle cerrado con relé. Para cada una de estas opciones se pueden elegir diversos métodos de ajuste, seleccionados de [5], de forma que se pueda elegir un ajuste más lento y robusto, o uno más rápido y oscilatorio. En el caso del escalón, los métodos se basan en dibujar la recta tangente al punto de máxima pendiente. Se ofrecen dos opciones si no se utiliza la información de la ganancia, (ZNrapido y ZNlento), que son variaciones del método de Ziegler Nichols, y otras dos opciones si se utiliza la ganancia (FOTDrapido y FOTDlento). En el caso de la realimentación a relé, se utiliza el periodo de oscilación y la amplitud, y se ofrecen tres opciones (BCrapido, BCintermedio y BClento), más una cuarta opción que requiere introducir además la ganancia estática del sistema (obtenida de otro experimento).

La recta tangente al punto de máxima pendiente se sitúa manualmente, lo que permite al estudiante ser más consciente de cuál es el fundamento del método.

4 EJEMPLOS DE UTILIZACIÓN

En esta sección se detallan varios ejemplos de utilización del sistema completo desarrollado.

4.1 CONTROL DE UN CIRCUITO ELÉCTRICO

La figura siguiente muestra el esquema de conexiones del proceso a controlar

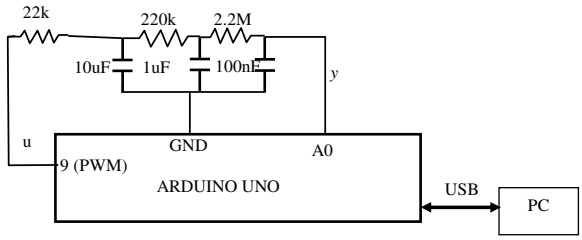


Figura 8: Esquema del circuito eléctrico

La planta es extremadamente simple, pues no requiere alimentación externa. El modelo teórico del circuito es de tercer orden, con tres polos reales diferentes.

En la aplicación de interfaz con Arduino se deben configurar los siguientes valores: Periodo= 10 ms, Frecuencia de PWM = 7812.5 Hz, U en paro=0 V, U_max=5 V, U_min=0 V. Se selecciona modo bucle abierto y tipo de entrada analógica. Tras pulsar marcha y cambiar varias veces la entrada al sistema se tiene la gráfica de la aplicación:

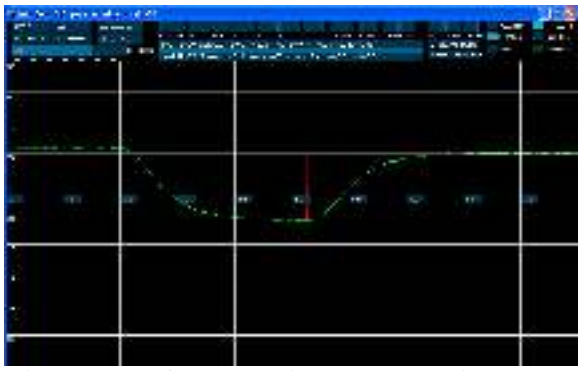


Figura 9: Interfaz de Arduino con el experimento del circuito eléctrico

Una vez se estabiliza la salida, se pulsa inicio de grabación, se cambia la entrada y después se pulsa final de grabación, con lo que se genera un fichero con los datos de entrada y salida. En este caso se realiza un ajuste experimental de un PID a partir de la respuesta ante escalón. Cargando los datos guardados en la aplicación de ajuste experimental, y seleccionando controlador PID, ajuste escalón, y método de ajuste ZNrapido se tiene:

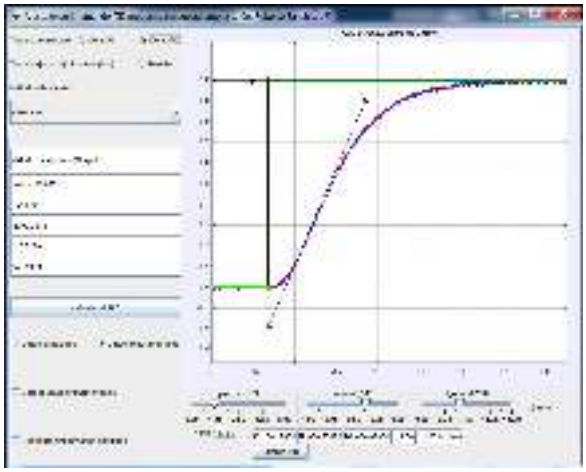


Figura 10: Ajuste experimental de PID para circuito eléctrico

En la figura se muestra el ajuste, una vez se ha colocado adecuadamente la recta tangente y el punto que indica el valor final y el inicio del escalón. El ajuste experimental no calcula ningún valor especial para el parámetro del filtro del derivador, *N*, que se deja en 10 por defecto (filtrado intermedio). El PID ajustado se guarda en un fichero para cargarlo después con la aplicación de interfaz de Arduino. En ésta se selecciona ahora Bucle cerrado, y se pulsa el botón para cargar controlador. Al darle marcha, y cambiar varias veces la señal de referencia se obtiene:



Figura 11: Prueba en planta de PID (ZNrapido) para circuito eléctrico: azul referencia, verde salida, rojo acción de control

4.2 CONTROL DE VELOCIDAD DE UN MOTOR DE CONTINUA

La figura siguiente muestra el esquema de conexiones del proceso a controlar

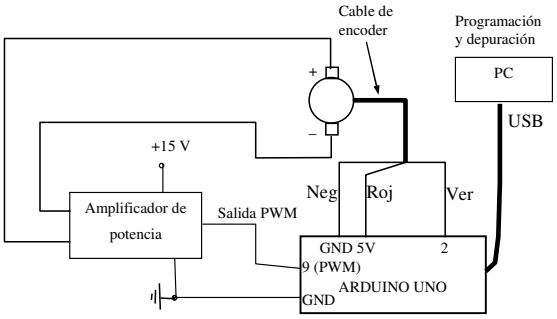


Figura 12: Esquema del control del motor de continua

El motor de continua dispone de un codificador incremental con dos salidas en cuadratura. De ellas, solo se utiliza una para medir y controlar la velocidad. Para hacer el experimento se selecciona Bucle abierto y entrada Captura en la aplicación de interfaz de Arduino, dejando la frecuencia del PWM en su valor máximo (7812.5). Tras pulsar marcha, y darle varios cambios a la entrada se obtienen varios escalones, guardando uno de ellos en fichero (se pulsa inicio grabar, se cambia la entrada y se pulsa stop grabar):



Figura 13: Interfaz de Arduino con el experimento del motor de continua

Para obtener el modelo del sistema, se cargan los datos guardados en la aplicación de identificación, y se selecciona el rango correspondiente a un escalón:

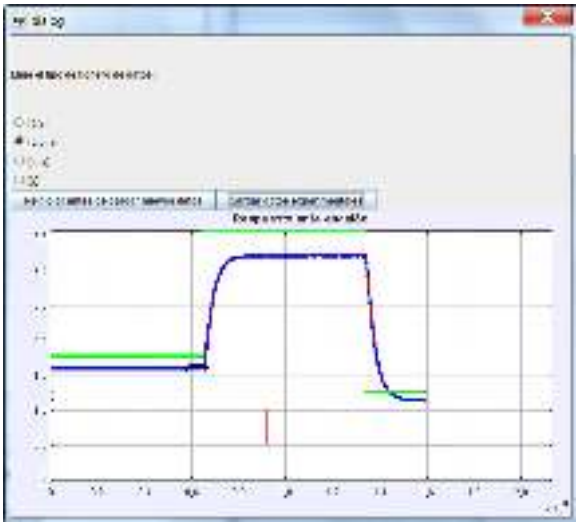


Figura 14: Datos cargados del experimento del motor de continua y selección del rango

Se selecciona un polo real y se ajusta la ganancia, el polo y el retardo para que la parte final del escalón ajuste lo mejor posible:

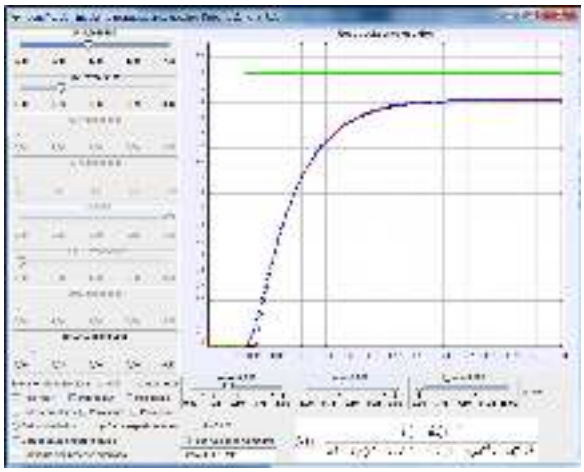


Figura 15: Aproximación del motor de continua a modelo de primer orden con retardo

Ahora se sustituye el retardo por un segundo polo real de constante de tiempo igual al retardo, y se hace el ajuste fino guiándose con el error:

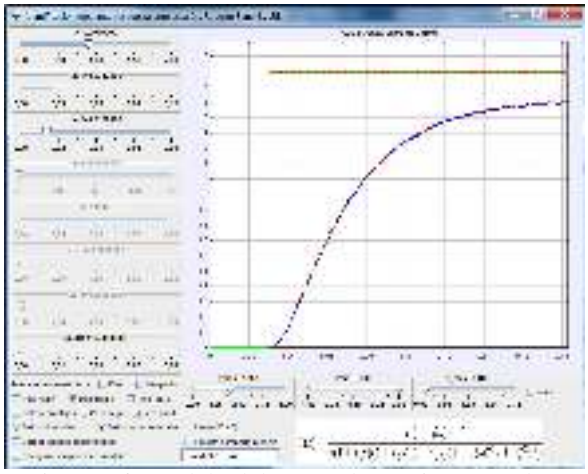


Figura 16: Modelo del motor de continua con dos polos reales

Una vez obtenido el modelo, se guarda en un fichero en el disco. Para el diseño del controlador PID, se carga este modelo en la aplicación de diseño de PID. Se selecciona un margen de fase de 60º y un margen de ganancia mínimo de 8 dB, así como un parámetro de filtro del derivador N=5 (filtrado importante debido al ruido de la señal de captura). Ajustando el valor del parámetro a con la deslizadera hasta que la ganancia Kc es máxima se obtiene el controlador PID:

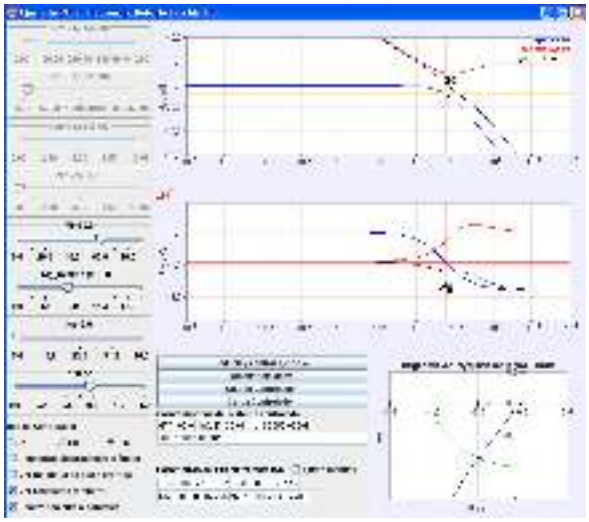


Figura 17: Diseño de PID para motor de continua

cuya respuesta ante referencia y perturbación escalón, tras ajustar el factor de ponderación de la parte proporcional y derivada de la referencia es:



Figura 18: Respuesta simulada de PID para motor de continua

Una vez diseñado el PID, se guarda en un fichero para cargarlo desde la interfaz de Arduino y probar el controlador sobre el proceso real. Para ello se elige bucle cerrado:



Figura 19: Respuesta real de PID para motor de continua: azul referencia, verde salida, rojo acción de control.

4.3 CONTROL DE POSICIÓN DE UN MOTOR DE CONTINUA

El esquema del sistema a controlar es similar al de la figura 12, salvo que se conecta la segunda salida del codificador al pin 3 de Arduino. Por otra parte, el amplificador de potencia ahora es distinto, pues da 0 voltios al motor cuando el ciclo de trabajo del PWM es del 50%, dándole tensión negativa si el ciclo de trabajo es inferior, y tensión positiva si es superior al 50%. En este caso se plantea realizar un experimento con realimentación a relé para realizar después el ajuste experimental. Se configura la aplicación de interfaz para Arduino en modo relé y entrada encoder. Se fijan los parámetros del relé: histéresis=0.05, u+=5, u-=0, y se fija el valor de u en paro=2.5 voltios (cuando la salida PWM da 2.5 V, el motor estará parado), y se pulsa marcha. Como resultado se establece una oscilación, de la que se guardan en disco los datos de varios ciclos:

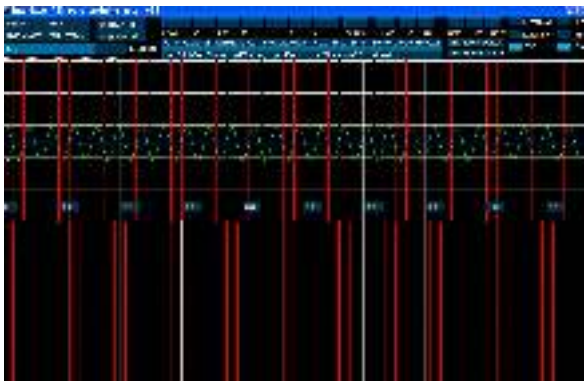


Figura 20: Experimento con relé en posición de motor

Para ajustar el controlador se cargan los datos en la aplicación de ajuste experimental. Tras hacer un escalado adecuado en la gráfica para ver bien la salida, se selecciona control PID, tipo de ajuste relé y método BCintermedio, y se colocan los puntos que marcan el semiperiodo de oscilación y el inicio del escalón, quedando:

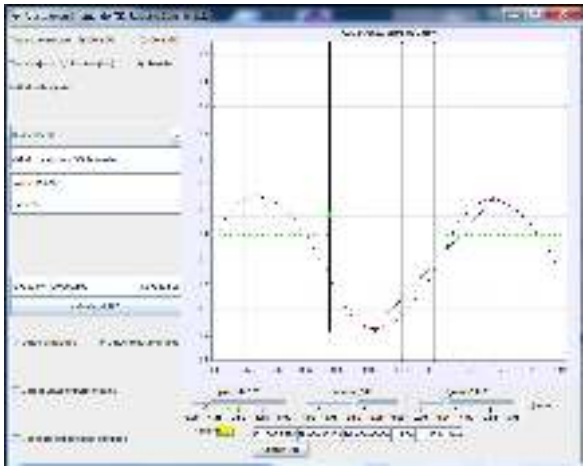


Figura 21: Ajuste de PID a partir de experimento con relé para posición de motor

Una vez ajustado el controlador PID, se guarda en un fichero para cargarlo con el interfaz de Arduino y probarlo en el proceso real. La figura 22 muestra el comportamiento sobre el proceso real del controlador PID ajustado. Se observa una sobreoscilación un poco alta, que se podría mejorar reduciendo el factor de ponderación de la referencia, *b*, a un valor menor que 1.



Figura 22: Respuesta de PID ajustado mediante experimento con relé para posición de motor

5 CONCLUSIONES

En este trabajo se presenta un conjunto de herramientas basadas en hardware y software libre, concretamente Arduino, Processing e Easy Java Simulations, para realizar el ciclo completo de identificación experimental de procesos y diseño e implementación de controladores PID. La plataforma hardware es una tarjeta Arduino UNO. Los elementos de software incluyen una aplicación de interfaz para Windows que se comunica con la tarjeta Arduino, y 3 aplicaciones JAVA: una para la identificación experimental, otra para el ajuste experimental de PID y otra para el diseño de PID basado en modelo mediante la respuesta en frecuencia.

El sistema desarrollado es de libre acceso (está disponible en la página web https://sites.google.com/a/uji.es/freepidtools), y permite realizar prácticas de identificación, ajuste experimental y diseño de PID a un coste muy bajo, y de forma muy sencilla e intuitiva. Además, los ficheros que se intercambian las distintas aplicaciones son de formato ASCII sencillo, por lo que pueden usarse parte de las aplicaciones desarrolladas junto con otros sistemas (por ejemplo se podría usar la parte de interfaz de Arduino, y los datos guardados ser leídos desde Matlab).

Agradecimientos

DPI2011-27845-C02-02.

Referencias

[1] Sanchis, R., Romero, J. and Balaguer, P. (2010) "Tuning of PID controllers based on simplified single parameter optimisation", *International Journal of Control,* Vol. 83 (Issue 9),pp. 1785-1798. DOI: 10.1080/00207179.2010.495162.

[2] Arduino Home Page. http://www.arduino.cc.

[3] Processing Home Page. http://www.processing.org.

[4] Easy Java Simulations Home Page. http://fem.um.es/Ejs.

[5] O'Dwyer, Aidan (2006). Handbook of Pi and PID Controller Tunning Rules, Imperial College Press, England.

[6] Página web de Freepidtools. https://sites.google.com/a/uji.es/freepidtools.

Laboratorio Virtual y Remoto de robots paralelos

Arturo Gil, Adrián Peidró, José María Marín, Óscar Reinoso, David Valiente, Luis Miguel Jiménez, Miguel Juliá
Departamento de Ingeniería de Sistemas y Automática.
Universidad Miguel Hernández de Elche, Avda. de la Universidad s/n, Elche (Alicante).
arturo.gil@umh.es

Resumen

*En este artículo presentamos una herramienta desarrollada para la docencia de robots paralelos. La aplicación se puede emplear como complemento a las clases teóricas mediante clases prácticas tanto en laboratorio como no presenciales. La aplicación se organiza como un conjunto de herramientas desarrolladas mediante EJS (*Easy Java Simulations*) que ayudan al estudiante en la comprensión de la cinemática de los robots paralelos. Un conjunto de estructuras paralelas clásicas han sido implementadas: la herramienta es capaz de resolver la cinemática directa e inversa de estructuras de tipo 5R, 3RRR y Delta. Una interfaz gráfica intuitiva permite al estudiante variar tanto las coordenadas articulares como las coordenadas cartesianas del extremo del robot mientras se observa el resultado con una representación gráfica del robot. Además, se plantean un conjunto de cinco sesiones prácticas basadas en la aplicación presentada. Durante estas sesiones prácticas el alumno analiza la cinemática inversa y directa de los mecanismos. Además, la herramienta permite analizar de forma sencilla las singularidades que aparecen en la solución de la cinemática directa e inversa de los robots. Estas singularidades se analizan a través de una herramienta que permite la planificación de trayectorias en el espacio cartesiano y articular de los robots, que permite analizar la posición y velocidad del extremo en función de las trayectorias y velocidades de los actuadores. La herramienta se ha utilizado durante el pasado curso 2012–2013 en la Universidad Miguel Hernández de Elche y recibió una buena acogida. Durante los últimos meses se ha añadido un módulo que permite también manejar un robot real de tipo 5R de forma remota.*

1. INTRODUCCIÓN

En este artículo se presenta una herramienta educativa que se centra en el campo de los robots paralelos. La aplicación proporciona una vista muy realista de tres estructuras paralelas clásicas y facilita la realización de una serie de experimentos con ellas. La *toolbox* fue desarrollada con la intención de facilitar la comprensión de los conceptos fundamentales dentro de la robótica de mecanismos paralelos, haciendo especial hincapie en los problemas cinemáticos directo e inverso.

Es sabido que la experimentación con equipos reales, así como con laboratorios remotos o virtuales son un buen complemento a las clases magistrales de teoría [11, 15]. Así, durante estas últimas el alumno adquirirá las herramientas y los conceptos fundamentales. Pero también, la experimentación y la posibilidad de manejar los mecanismos reales motivará al estudiante a aprender los aspectos teóricos con más interes. También está ampliamente aceptado que la utilización de diferentes estrategias educativas resulta positivo en el aprendizaje de los estudiantes. Además, en el caso particular de los estudios de ingeniería, las clases de laboratorio y prácticas deben tener una importancia mayúscula. En este sentido, algunos autores aseguran que las sesiones prácticas no sólo proporcionan un estímulo adicional para el aprendizaje, sino que inducen muchos otros aspectos positivos y una comprensión profunda de los conceptos tratados [3].

Los robots paralelos tiene la característica fundamental de ser capaces de soportar grandes cargas mientras, al mismo tiempo, son capaces de posicionar el extremo del robot con gran precisión. Sin embargo, los conceptos de robótica paralela son bastante más difíciles de enseñar y aprender [12]. Aun así, la formación en este campo de la robótica de resulta interesante, por existir ya un gran número de modelos comerciales basados en estructuras paralelas. Así pues, es necesario abordar este tipo de estructuras paralelas en la docencia de carreras como Ingeniería Industrial o Grado en Ingeniería Electrónica y Automática Industrial, por ejemplo. La docencia de los conceptos fundamentales de robots paralelos no puede ser tratada como una mera continuación de la teoría de robots de tipo serie. Esto es una consecuencia de las grandes diferencias fundamentales que existen entre ambos tipos de manipuladores. Por ejemplo, el conocimiento de la teoría de la Cinemática Directa en manipuladores de tipo serie no puede ser extrapolada directamente a la Cinemática Directa de manipuladores de tipo paralelo. Siguiendo

con el concepto anterior, los estudiantes encuentran generalmente más sencillo entender el concepto de Cinemática Directa en robots de tipo serie que en robot de tipo paralelo. Así, pues, para abordar estos aspectos educativos es totalmente necesario contar con una herramienta que proporcione a los estudiantes la posibilidad de visualizar los conceptos cinemáticos aplicados al caso de los robots paralelos. La ausencia casi total de este tipo de herramientas disponibles para la docencia nos motivó al desarrollo de nuestra propia herramienta que se presenta aquí[1].

La herramienta presenta claramente el estado del robot y sus características principales. La aplicación no requiere ser instalada, pues puede ser ejecutada *online*, embebida en un explorador web. Esta característica es muy importante, pues permite a los estudiantes realizar sus sesiones prácticas desde casa sin limitaciones de tiempo y sin necesidad de instalar ningún software adicional. En nuestra opinión, la herramienta aquí descrita es la primera aplicación que integra las estructuras de tipo 5R, 3RRR y Delta dentro de una misma arquitectura para facilitar el aprendizaje de estos robots paralelos. La herramienta se ha bautizado como LABEL, por *LABoratory for paralLEL robots*.

La herramienta ha sido utilizada en la Universidad Miguel Hernández de Elche durante el curso pasado en los estudios de Ingeniería Industrial y en el Máster Universitario en Investigación en Tecnologías Industriales y de Telecomunicación. En particular, la herramienta resuelve de forma amena los problemas cinemático directo e inverso de los robots de tipo 3RRR, 5R y Delta, todos integrados bajo una interfaz gráfica común e intuitiva. Los parámetros geométricos de cada robot pueden ser variados de forma sencilla mientras se comprueban la existencia de las soluciones de la cinemática directa e inversa. Además, la herramienta ofrece la posibilidad de visualizar las singularidades en la cinemática inversa y directa. Finalmente, se ha integrado un planificador de trayectorias que permite al estudiante evaluar los efectos de la existencia de la singularidades en el espacio de trabajo del robot. Un conjunto de sesiones prácticas se ha elaborado alrededor de la herramienta para cubrir los conceptos fundamentals de la robótica de mecanismos paralelos. Recientemente se ha añadido un módulo a la aplicación que permite conectar con un mecanismo de tipo 5R real en un laboratorio remoto. Las mismas actividades que se pueden realizar con la aplicación se pueden visualizar y realizar de forma de forma remota y real por el estudiante.

[1]la herramienta puede ser descargada y ejecutada desde arvc.umh.es/label

El resto del artículo se organiza de la siguiente manera: primero, el apartado 2 presenta otros trabajos en el área de interés relacionados con la aplicación presentada aquí. Las instrucciones de uso de la herramienta y los robots incluidos se presentan en el apartado 3. Un conjunto de sesiones prácticas y actividades se resumen en el apartado 4. Finalmente, las conclusiones principales se exponen en el apartado 5.

2. TRABAJO RELACIONADO

El uso de las nuevas tecnologías en Educación ha ampliado el abanico de opciones con que cuentan los estudiantes para realizar sus sesiones prácticas. Con la aparición de los laboratorios virtuales, que emulan el comportamiento de equipos reales [2], o el uso de laboratorios remotos que permiten un acceso remoto al equipamiento real [9, 10] los estudiantes tienen la posibilidad de incrementar el tiempo que dedican a realizar actividades prácticas y de planificar este tiempo de la mejor manera posible. Por ejemplo, en [9] un laboratorio remoto permite al estudiante acceder a un servomotor real y variar sus parámetros de control mediante un entorno remoto Matlab/Simulink.

En el ámbito de los manipuladores robóticos es importante mencionar el trabajo presentado en [1] donde un manipulador de tipo serie es simulado en un entorno virtual. Además, la herramienta permite teleoperar un brazo robótico real mientras el estudiante observa los resultados a través de una cámara web. Una aplicación basada en EJS (Easy Java Simulations) se presenta en [5]. La aplicación permite simular un laboratorio virtual programado en base a las librerías EJS, así como manejar un robot real dentro de la misma infraestructura. No obstante, la herramienta está dedicada únicamente al estudio de mecanismos de tipo serie.

En el área de las estructuras paralelas es necesario mencionar el trabajo presentado en [10], donde un laboratorio remoto permite a estudiantes e investigadores experimentar y controlar un robot paralelo de 3 GDL, así como sistemas térmicos y motores de corriente continua.

Los robots paralelos también se pueden simular mediante otro tipo de herramientas de propósito general, como las descritas en [8]. Por ejemplo, [8] presenta el uso de 20-Sim y Sim Mechanics para el modelado de manipuladores paralelos. Ambas aplicacaciones permiten diseñar y analizar sistemas dinámicos de una manera sencilla e intuitiva. Sin embargo, estas aplicaciones son propietarias, lo que limita el acceso de los estudiantes desde casa. Además, algunos conceptos, como, por ejemplo, el análisis del espacio de trabajo o el estudio

de las singularidades son difíciles de analizar con estos programas.

3. USO DE LA HERRAMIENTA Y ROBOTS INCLUIDOS

La aplicación lidia con los siguientes objetivos de tipo educativo:

- Cinemática inversa.
- Cinemática directa.
- Espacio de trabajo y singularidades.
- Laboratorio remoto.

Estos son los conceptos fundamentals que se deberían tratar en un curso de introducción a los robots paralelos. Otros conceptos importantes como la dinámica directa e inversa son de gran importancia pero no se introducen como objetivos educativos en este curso por falta de tiempo.

Todos los aspectos educacionales abordados en el apartado anterior pueden ser realizados mediante la herramienta. En este sentido, se comienza con el análisis cinemático del robot 5R, según se describe en el apartado 3.1.

3.1. El robot 5R

El mecanismo de tipo 5R es más sencillo de analizar que los robots de tipo Delta o 3RRR y sirve como un punto de comienzo muy adecuado en el estudio de los robots paralelos. Durante las sesiones de teoría se analiza la solución a la cinemática inversa de este robot, ya que presenta unas ecuaciones más sencillas que la solución al problema cinemático directo. Más tarde se aborda la solución al problema cinemático directo.

El robot de tipo 5R posee 2 GDL (grados de libertad) y presenta un movimiento contenido en un plano. El mecanismo consta de 4 eslabones móviles y 5 articulaciones, dos de ellas activas y movidas por motores eléctricos. La figura 1(a) presenta las características principales del mecanismos, donde las articulaciones A_1 y A_2 son activas, mientras que el resto B_1, B_2 y P son pasivas. El efector final del robot se considera como el punto P, según se muestra en la figura. La cinemática de este mecanismo ha sido estudiada ya por otros autores [6], presentando una solución sencilla para la cinemática inversa: dado que se conoce $P = (x, y)$, existen 4 soluciones diferentes para las posiciones articulares θ_1 y θ_2. El análisis cinemático directo considera el cálculo de $P = (x, y)$ como una función de las variables articulares θ_1 and θ_2. En este caso existen 2 soluciones diferentes según se demuestra en [6].

El simulador del robot 5R se presenta en la figura 2. Ayuda a los estudiantes a simular y entender los conceptos de cinemática directa e inversa aplicados a este mecanismo en particular. Para hacer esto, LABEL presenta las coordenadas cartesianas del extremo y las coordenadas articulares en la misma ventana. El alumno puede variar fácilmente las coordenadas cartesianas del punto $P = (x, y)$ arrastrándolo con el ratón mientras observa, al mismo tiempo, cómo varían las coordenadas articulares (θ_1, θ_2). Esta faceta permite al estudiante comprender y relacionar ambos problemas cinemáticos: directo e inverso. Las soluciones existentes pueden ser seleccionadas desde el panel de la derecha (cuatro variantes para el caso inverso y dos para el directo).

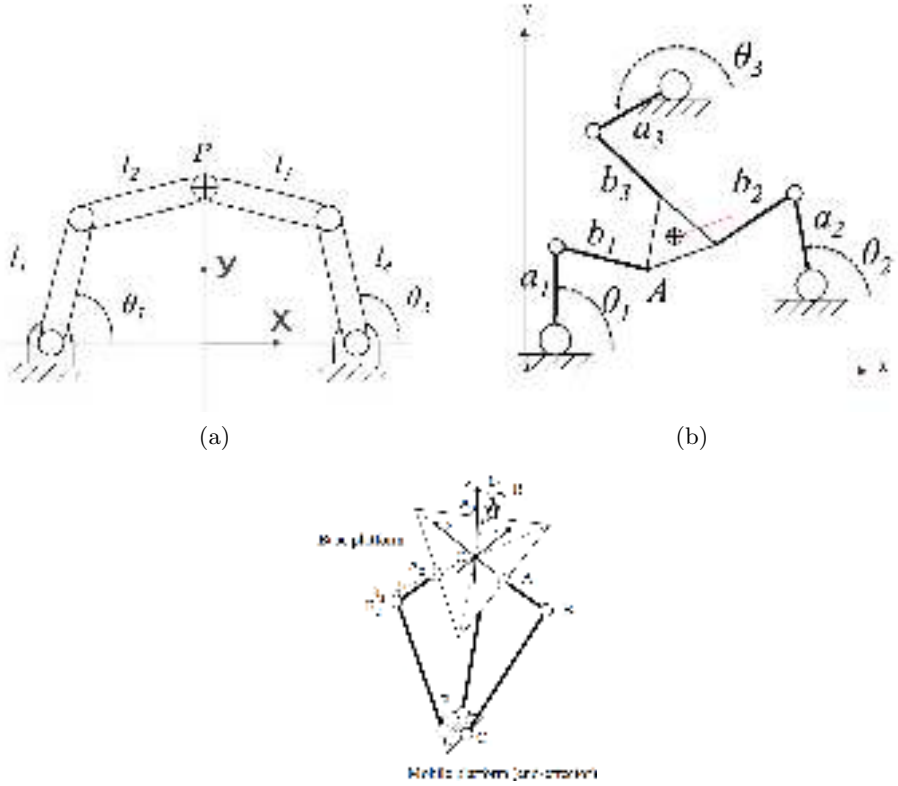
3.2. El robot 3RRR

El robot 3RRR consta de una plataforma móvil unida a una base fija mediante tres cadenas cinemáticas de tipo RRR, constando cada una de ellas, por tanto, de tres articulaciones rotacionales. El mecanismo se gobierna mediante tres articulaciones activas $(\theta_1, \theta_2, \theta_3)$ y posee un total de 6 juntas pasivas (figura 1(b)). Según se indica en la figura 1(b), esta configuración cuenta con tres grados de libertad, pudiendo fijar la posición del punto $A = (x_A, y_A)$ y la orientación ϕ de la plataforma de forma independiente.

El mecanismo de tipo 3RRR es más complicado de analizar que el robot 5R, así pues, su análisis es abordado una vez que el alumno ha adquirido los conocimientos necesarios. Durante las lecciones de teoría se analiza la solución a la cinemática inversa del 3RRR, siendo esta bastante sencilla. Además, una solución a la cinemática directa se analiza siguiendo los principios establecidos en [4, 7].

3.3. El robot Delta

El robot Delta es descrito en la figura 1(c), donde se presentan tres cadenas cinemáticas que conectan el extremo con la base del robot. Cada cadena cinemática está formada por dos eslabones que conectan la base y el efector final a través de tres articulaciones rotacionales. Las juntas restringen el movimiento del efector de manera que la orientación se mantiene constante respecto de la base. La figura 1(c) presenta tres articulaciones activas con variables articulares θ_1, θ_2 y θ_3 que pueden ser modificadas. Estas articulaciones son movidas por tres motores independientes instalados en las juntas denotadas A_1, A_2 y A_3. La cinemática in-



(a) (b)

(c)

Figura 1: Los robots 5R, 3RRR y Delta.

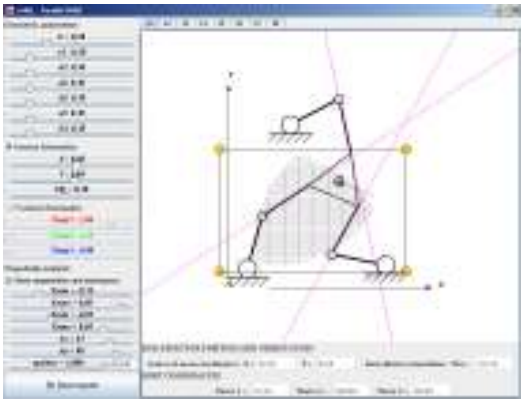


Figura 4: La figura presenta la aplicación desarrollada para el robot 3RRR robot.

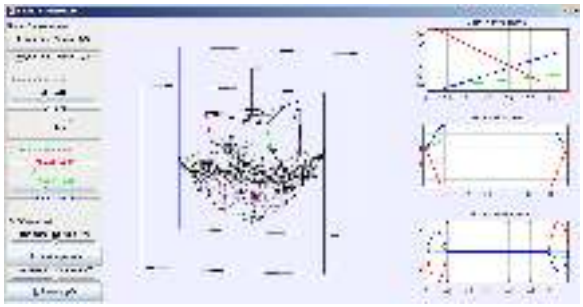


Figura 5: La figura presenta el robot Delta. Los puntos representan los límites del espacio de trabajo.

versa del robot Delta puede ser resuelta de formas diferentes, como en [14], donde se presenta una solución cerrada. La solución de la cinemática directa es más compleja, según se describe en [16].

4. SESIONES PRÁCTICAS Y MANEJO DE LA HERRAMIENTA

En este apartado se presentan una serie de actividades que pueden ser llevadas a cabo mediante la herramienta presentada. Las actividades se han diseñado de manera que la herramienta se presenta de forma gradual y se combina con los conceptos de teoría introducidos durante las clases magistrales. Las sesiones prácticas propuestas son:

- Sesión 1: Introducción a la aplicación LABEL

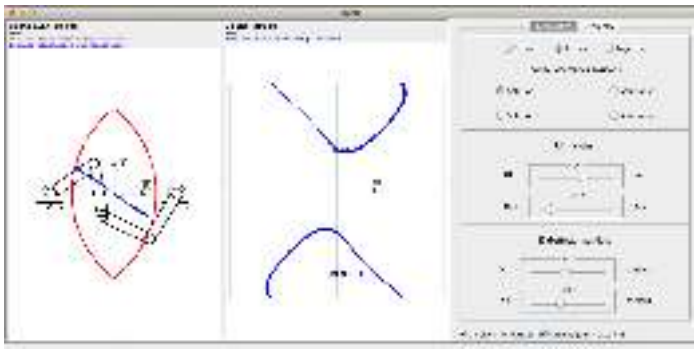


Figura 2: La figura presenta la aplicación del robot 5R. El espacio de trabajo alcanzable se presenta en línea continua. Las singularidades de la cinemática directa se presentan en línea discontinua.



Figura 3: La figura presenta la aplicación que permite conectar con el robot real 5R en nuestro laboratorio.

y análisis del movimiento de las estructuras paralelas.

- Sesión 2: Cinemática inversa.
- Sesión 3: Cinemática directa.
- Sesión 4: Singularidades y espacio de trabajo.
- Sesión 5: Planificación de trayectorias.

4.1. Introducción a la aplicación LABEL y análisis del movimiento de las estructuras paralelas

Se plantea esta sesión como una toma de contacto con los mecanismos de tipo paralelo. Esta sesión se realiza en el laboratorio. Se le indica al estudiante que analice el movimiento de una serie de mecanismos de tipo paralelo en función de las restricciones impuestas por cada eslabón y tipo de articulación. Los grados de libertad de cada mecanismo se calculan utilizando el criterio de Grübler [13]. Se le indica al alumno que calcule los grados de libertad del robot 5R, 3RRR y Delta y que después observe el movimiento mediante la representación gráfica de LABEL.

4.2. Cinemática inversa

Primero, el estudiante analiza la cinemática inversa del robot 5R. Se le indica al estudiante los pasos necesarios para que él mismo derive las ecuaciones. A continuación, el estudiante comprueba y valida todas las soluciones factibles utilizando la herramienta LABEL. Los robots 3RRR (figura 3.3) y Delta (figura 3.3) se analizan de la misma manera.

Los estudiantes son capaces de observar de forma rápida las posibles configuraciones que permiten al robot alcanzar la misma posición, o bien la misma posición y orientación del efector final. En el caso del robot 5R, existen 4 soluciones diferentes para el problema cinemático inverso, según se aprecia en la figura 2.

4.3. Cinemática directa

Esta sesión pretende ayudar a los alumnos en el estudio del problema cinemático directo en los robots paralelos. Esta sesión práctica se desarrolla en parte en el laboratorio y, en parte, como trabajo autónomo del estudiante. Se le indica a los estudiantes que resuelvan las restricciones geométricas de los robots 5R, 3RRR y Delta en función de las coordenadas articulares. Con esto se pretende que los estudiantes entiendan la complejidad de las ecuaciones involucradas en la solución. La solución de estas ecuaciones, o su modelado en cualquier lenguaje de programación no se abordan, debido a la cantidad de tiempo que esto requeriría.

Durante esta práctica los estudiantes son capaces de encontrar las relaciones entre los problemas cinemático e inverso. Para ello, comparan las posibles soluciones para el caso del robot 5R.

4.4. Análisis de singularidades

Durante esta sesión el alumno deberá entender el concepto de punto singular, tanto en la cinemática directa como en la inversa. Esta sesión se realiza a partes iguales en el laboratorio y de forma autónoma por parte del estudiante.

Por ejemplo, la figura 2 presenta en línea continua las singularidades de la cinemática inversa del mecanismo en el espacio cartesiano $x - y$. En el caso del robot 5R, estas singularidades corresponden con los límites del espacio de trabajo del robot. Las singularidades en la cinemática directa son visualizadas de forma clara y comprensible mediante la aplicación LABEL.

4.5. Planificación de trayectorias

En esta última sesión, los estudiantes hacen un recorrido global por el problema de la planificación de trayectorias en el robot 5R y el Delta. Primero se expone la importancia del problema de la planificación de trayectorias y se presentan algunas posibles soluciones. A continuación, se les indica a los alumnos que simulen las trayectorias que vienen integradas en la herramienta que simula el robot Delta. En este caso LABEL no integra una detección ni evasión de puntos singulares, con lo que los estudiantes encontrarán problemas en este caso y entenderán la necesidad de diseñar trayectorias libres de singularidades. A continuación, realizarán la misma actividad utilizando el simulador para la plataforma 5R, donde se ha incluído un planificador de trayectorias que evita los puntos singulares. Los estudiantes visualizan al mismo tiempo la trayectoria del extremo en el espacio cartesiano y el espacio articular $\theta_1 - \theta_2$, con lo que las singularidades se visualizan al mismo tiempo en ambos espacioes.

4.6. Laboratorio remoto

Recientemente se ha añadido a la herramienta un módulo que permite conectar con un mecanismo de tipo 5R real. Así pues, la herramienta LABEL se puede convertir también en un laboratorio remoto manteniendo la misma apariencia (figura 3). De esta manera, el estudiante puede manejar un sistema real de forma remota moviendo el mecanismo en su cinemática directa e inversa, mientras comprueba los mismos conceptos abordados por la herramienta virtual.

5. CONCLUSIÓN

Se ha presentado una herramienta educativa que se centra en el análisis cinemático de mecanismos paralelos. Tres mecanismos paralelos clásicos han sido implementados: los robots 5R, 3RRR y Delta. Una aplicación gráfica intuitiva permite al estudiante cambiar las variables articulares, las variables cartesianas del extremo, así como los parámetros geométricos del mecanismo. Además, se han diseñado un conjunto de sesiones prácticas que se pueden realizar con la herramienta. Las sesiones plantean el estudio de la cinemática directa e inversa de las estructuras paralelas comentadas. Además, las singularidades que aparecen en la soluciones de la cinemática directa en inversa también son analizadas. La herramienta también dispone de un módulo para el cálculo de trayectorias que permite al alumno analizar la posición y velocidad del extremo en función de la posición y velocidad de los actuadores, de manera que el alumno puede entender fácilmente la existencia de singularidades en el espacio de trabajo del robot. La herramienta se ha utilizado durante el pasado curso académico 2012-2013. Finalmente, durante los últimos meses se ha añadido a la herramienta un módulo que permite conectar con un mecanismo de tipo 5R real. De esta manera, el estudiante puede manejar un sistema real de forma remota y observar los mismos resultados que obtenía con la plataforma simulada en la aplicación. Como trabajo futuro se plantea incluir en la herramienta un análisis dinámico de los mecanismos integrados.

Agradecimientos

Este trabajo ha sido posible gracias al proyecto DPI2010–15308 del Ministerio de Ciencia e Innovación, titulado "Exploración integrada de entornos mediante robots cooperativos para la creación de mapas 3D visuales y topológicos que puedan ser usados en navegación con 6 grados de libertad".

Referencias

[1] Candelas, F. A., Puente, S. R., Torres, F., Ortiz, F., Gil, P., and Pomares, J. A virtual laboratory for teaching robotics. *International Journal of Engineering Education 19*, 3 (2003), 363–370.

[2] Dormido, S. Control learning: present and future. *Annual reviews in Control*, 28 (2004), 115–136.

[3] Fry, H., Ketteridge, S., and Marshall, S. *A Handbook for Teaching & Learning in Higher Education: Enhancing Academic Practice*. Kogan Page, London, U.K., 2003.

[4] Gosselin, C. M., and Sefrioui, J. Polynomial solution for the direct kinematic problem of planar three-degree-of-freedom parallel manipulators. In *IEEE Int. Conf. on Robotics & Automation* (1991).

[5] Jara, C. A., Candelas, F. A., Gil, P., Torres, F., Esquembre, F., and Dormido, S. EJS+EjsRL: An interactive tool for industrial robots simulation, computer vision and remote operation. *Robotics and Autonomous Systems 59*, 6 (2011), 389 – 401.

[6] Liu, X.-J., Wang, J., and Pristchow, G. Performance atlases and optimum design of planar 5R symmetrical parallel mechanisms. *Mechanism and Machine Theory 41* (2006), 119–144.

[7] Oetomo D., Hwee Choo Liaw, A. G., and B., S. Direct kinematics and analytical solution to 3RRR parallel planar mechanisms. In *9th International Conference on Control, Automation, Robotics and Vision* (2006), pp. 1–6.

[8] Parasuraman, S., and Liang, P. Development of RPS parallel manipulators. In *Computer and Network Technology (ICCNT), 2010 Second International Conference on* (april 2010), pp. 600–605.

[9] Puerto, R., Jiménez, L., and Reinoso, O. Remote control laboratory via internet using Matlab and Simulink. *Computer Applications in Engineering Education 18*, 4 (2010).

[10] Santana, I., Ferre, M., Izaguirre, E., Aracil, R., and Hernández, L. Remote laboratories for education and research purposes in automatic control systems. *IEEE Transactions on Industrial Informatics* (To appear).

[11] Shirsavar, S. A., Potter, B. A., and Ridge, I. M. L. Three-phase machines and drives. *IEEE Trans. Educ. 49*, 3 (2006), 383–388.

[12] Tan, D.-P., Ji, S.-M., and Jin, M.-S. Intelligent computer-aided instruction modeling and a method to optimize study strategies for parallel robot instruction. *Education, IEEE Transactions on PP*, 99 (2012), 1.

[13] Tsai, L.-W. *Robot Analysis. The Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons, Inc., 1999.

[14] Tsai, L. W., and Stamper, R. E. A parallel manipulator with only translational degrees of freedom. *System Research 72* (1997).

[15] Williams, J. M., Clae, J. L., Benavides, N. D., Wooldridge, J. D., Koeing, A. C., Tichenor, J. L., and Pekarek, S. D. Versatile hardware and software tools for educating students in power electronics. *IEEE Trans. Educ. 47*, 4 (2004), 436–445.

[16] Zhang, J., Shi, L., Gao, R., and Lian, C. The mathematical model and direct kinematics solution analysis of delta parallel robot. In *IEEE Int. Conf. on Computer Science and Information Technology. ICCSIT 2009* (2009).

BENCHMARK DE CONTROL Y SUPERVISIÓN DE REDES DE DISTRIBUCIÓN DE AGUA

Gerard Sanz, Ramon Pérez
gerard.sanz@upc.edu, ramon.perez@upc.edu
Grupo Sistemas de Control Avanzado de la Universidad Politécnica de Catalunya, Terrassa 08222, España

Resumen

La disponibilidad de agua potable es posible gracias a las infraestructuras construidas que conforman las redes de distribución. Las compañías de agua son las encargadas de mantener estas redes en buen estado, así como asegurar unas condiciones de calidad y presión en el consumidor final. Estas condiciones se verifican mediante medidas realizadas en los sensores de la red. Si se dispone del modelo de la red de distribución de agua, estas medidas pueden utilizarse también para su calibración, así como en metodologías de detección de fugas [3], control de la calidad del agua [2], etc. Estas metodologías se desarrollan en lenguaje Matlab, combinadas con otros entornos que permiten la simulación de redes hidráulicas, como EPANET [4]. Este trabajo presenta una interfaz gráfica que combina ambas herramientas, permitiendo su interconexión y facilitando la aplicación por parte del usuario de funciones ya existentes, a la vez que permite incorporar nuevos módulos personalizados.

Palabras clave: Redes de distribución de agua, modelos, simulación, control de presión.

1. INTRODUCCIÓN

Los modelos de simulación son la base de una gran parte de la investigación y transferencia de tecnología relacionada con las redes de distribución de agua. La herramienta EPANET permite simular el comportamiento de redes hidráulicas dadas unas condiciones de contorno, mientras que existen otros software con gran poder de cálculo matemático, así como un extenso abanico de funciones ya programadas. La interacción de ambos tipos de herramientas permite el análisis del comportamiento de las redes y el estudio del funcionamiento de diferentes metodologías de control y supervisión. Durante el desarrollo de la tesis del autor se programará una interfaz gráfica para facilitar la manipulación e interacción de las herramientas nombradas. En este trabajo se presenta el estado de la interfaz actual, que incluye un módulo de simulación y un módulo de control de presión con el que ya se ha podido trabajar y obtener resultados [5]. En la página web[1] del grupo de investigación de los autores se puede descargar la interfaz, juntamente con un manual de uso, algunas redes académicas y propuestas didácticas.

2. MODELOS DE REDES DE DISTRIBUCIÓN DE AGUA

Existen diferentes representaciones de redes de distribución de agua, como pueden ser los modelos matemáticos, redes neuronales o modelos borrosos. Todos estos modelos son válidos si cumplen la función para la cual han sido diseñados. En este trabajo se utiliza un modelo de simulación donde cada elemento tiene su comportamiento físico representado matemáticamente, teniendo en cuenta también la interacción entre los diversos elementos.

A continuación se presentan los componentes básicos de un modelo de distribución de agua potable [1][7].

2.1. Elementos físicos

Una red de distribución de agua está formada por tuberías, bombas, válvulas, tanques, depósitos y consumos. Los diversos elementos se interconectan para producir una red formada de nodos y ramas. Las bombas, tuberías y válvulas constituyen las ramas de la red, mientras que las uniones de estas ramas son los nodos: tanques, depósitos y consumos. Las propiedades físicas para describir el comportamiento de los elementos son los caudales en las tuberías y las alturas piezométricas o cargas de los nodos. En la figura 1 se puede observar una red académica con dos válvulas de reducción de presión en las entradas que conectan un sector de seis consumidores con dos depósitos fuente.

2.2. Elementos no físicos

Por otro lado, es necesario describir tres objetos informativos encargados de describir el compor-

[1] `http://sac.upc.edu/training-benchmarks/simulador-de-xarxes-de-distribucio-daigues`

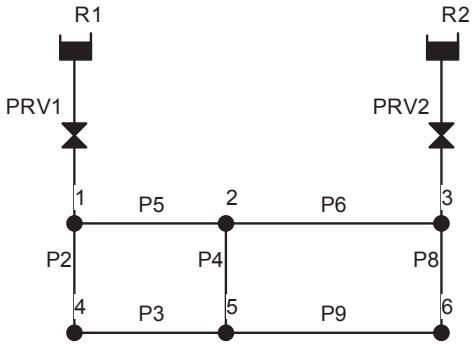


Figura 1: Representación de una red de agua en EPANET

tamiento y los aspectos operacionales del sistema de distribución de agua: las curvas, los patrones temporales y los controles.

Las curvas se utilizan para representar la relación existente entre dos magnitudes de los elementos de la red, como por ejemplo la curva de rendimiento de una bomba, la curva de volumen de un tanque o la curva de pérdidas en una válvula.

Los patrones temporales son colecciones de factores multiplicativos que se aplican a un valor constante para representar su variación en el tiempo. Se utilizan en el caso de las demandas de los nodos, alturas de los depósitos, programas de bombas, etc.

Finalmente, los controles son consignas que determinan cómo trabaja la red, especificando el estado de las tuberías seleccionadas, el nivel de agua en un tanque o las presiones de determinados puntos de la red.

3. DESCRIPCIÓN DEL SOFTWARE

En esta sección se presentan las diferentes herramientas informáticas utilizadas tanto para la simulación de los modelos de las redes de agua, como para la programación y desarrollo de metodologías a aplicar en las mismas redes.

3.1. EPANET

EPANET es un programa de simulación para el análisis del comportamiento hidráulico y de la calidad del agua en redes de distribución a presión. EPANET se encarga de determinar el caudal circulante, presión, niveles de agua y concentración de componentes químicos en la red durante un determinado período de tiempo. Existen otros tipos de software dedicados al análisis de redes de agua, como Piccolo o Waterworks.

Sin embargo, todos estos simuladores ofrecen muchas limitaciones para la programación de funciones. Uno de los programas más utilizados en ingeniería automática para poder desarrollar y validar diferentes metodologías es el Matlab.

3.2. Matlab

Matlab es un lenguaje de computación técnico de alto nivel y un entorno interactivo para el desarrollo de algoritmos, visualización y análisis de datos, cálculo numérico, etc. La utilización de Matlab está abierta a muchas aplicaciones, desde procesamiento de señales e imágenes hasta el análisis financiero y biología computacional, pasando por diseño de sistemas de control y modelado de sistemas.

Con la disponibilidad del simulador EPANET y del software de cálculo Matlab, se requiere una herramienta que permita trabajar en un solo entorno aprovechando las funcionalidades de cada uno de los software anteriormente nombrados. Esta herramienta es la toolkit EPANET-Matlab.

3.3. Toolkit EPANET-Matlab

La toolkit es una DLL (Dynamic Link Library) que permite a los programadores personalizar el motor de cálculo de EPANET para sus propias necesidades. Esta herramienta se puede incorporar a cualquier aplicación de Windows que comprenda el lenguaje C/C++, Delphi Pascal o Visual Basic; o que sea capaz de hacer llamadas a funciones de una DLL, como es el caso de Matlab. Gracias a esta DLL es posible integrar todas las funciones propias de EPANET en otras aplicaciones, permitiendo así una mayor libertad para tratar y modificar los datos de la red.

Una vez se han conectado EPANET y Matlab, el usuario habitual del simulador se puede encontrar con la dificultad de desconocer el entorno de programación Matlab, de la misma forma que el usuario que domina Matlab puede carecer de altos conocimientos en redes de distribución de agua.

3.4. Interfaz gráfica desarrollada

Este artículo presenta una interfaz cuyo objetivo principal es facilitar el trabajo a todos los usuarios, ofreciendo un entorno gráfico donde el conocimiento de programación Matlab y de los detalles de redes de distribución de agua pasan a un segundo plano. Asimismo, ofrece muchas posibilidades tanto de desarrollo como de prueba de diferentes metodologías sobre redes complejas.

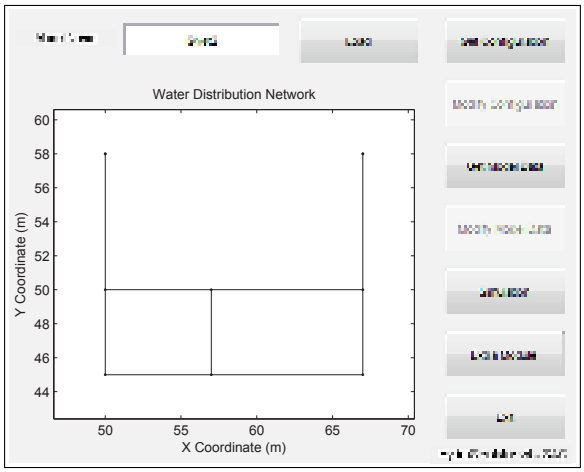


Figura 2: Interfaz del módulo de simulación

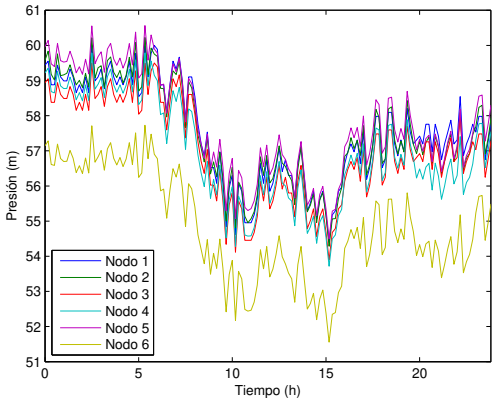


Figura 3: Presiones obtenidas en los 6 nodos de la red

4. SIMULACIÓN

La interfaz desarrollada está formada por un conjunto de módulos independientes. El módulo principal incluye funciones para el análisis y simulación de la red. Desde este módulo es posible llamar a otros módulos, que a su vez pueden hacer uso de las funcionalidades de la interfaz principal. En la figura 2 se visualiza la interfaz del módulo de simulación y análisis.

El módulo principal permite obtener las características propias de los elementos de la red. El usuario puede generar de forma rápida una matriz que contiene los identificadores de todos los nodos, sus cotas, demandas, patrones de comportamiento asociados y sus tipos. De la misma forma se puede generar una matriz con las características propias de las ramas de la red: identificador, longitud, diámetro, rugosidad, estado y tipo. También es posible obtener los diferentes patrones de comportamiento.

Sin embargo, la función principal del módulo básico es la de realizar simulaciones. Indicando las condiciones de contorno de la red (presiones en las entradas y demandas en los nodos), el usuario puede obtener las medidas de presión y caudal en los sensores que se definan durante el período de tiempo deseado.

En la figura 3 se observa el resultado de una simulación con muestreo diez-minutal en la que se han medido las presiones de los seis nodos de la red de la figura 1, también disponible en la web.

Uno de los usos potenciales de este módulo (incluyendo pequeñas modificaciones) sería la monitorización de la red. La principal ventaja de la herramienta es la libertad de programación desde la línea de comandos de Matlab una vez se han almacenado todos los resultados de las simulaciones en el espacio de trabajo, permitiendo su manipulación de forma fácil y rápida.

5. CONTROL DE PRESIÓN

Como ejemplo de las posibilidades que ofrece el benchmark se ha implementado un módulo de control de presión que permite mantener la presión de un nodo determinado en una consigna deseada. El actuador usado en este caso es una válvula de reducción de presión cuya entrada es la consigna aguas abajo. Estas válvulas (en el ejemplo hay dos) suelen aparecer en las entradas de los sectores de redes complejas. El control de presión es una práctica usada habitualmente para la reducción de fugas en redes [6]. Asignando una consigna de baja presión a la entrada se garantiza de forma simple la reducción de la presión en todo el sector. Sin embargo, para garantizar un servicio de calidad y debido a que la presión depende de la demanda en la red, en este benchmark se ejemplifica un control en lazo cerrado de la zona (nodo) más crítica del sector. La figura 4 representa el control aplicado.

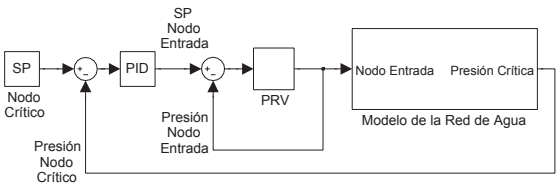


Figura 4: diagrama de bloques del control de presión

El sistema es estático y el comportamiento varía cuando varían sus entradas: condiciones de contorno (presiones) y demandas. El módulo de control de presión tiene, igual que el de simulación,

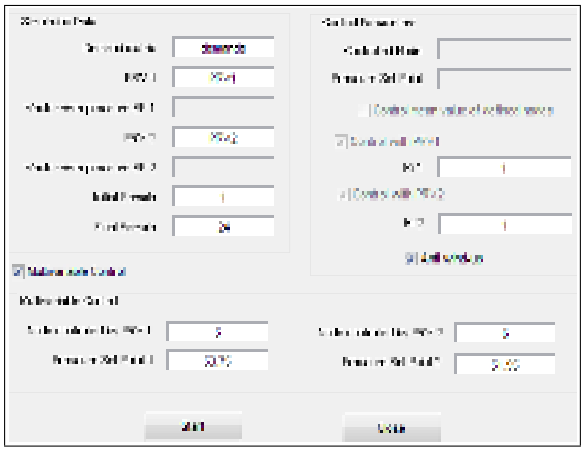


Figura 5: Interfaz gráfica del módulo de presión

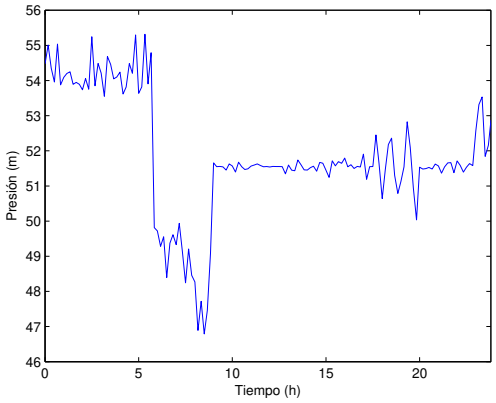


Figura 6: Presión en el nodo 6 aplicando control monovariable $P_{SP_6} = 51{,}55\ m$ con PRV2 sin antiwindup

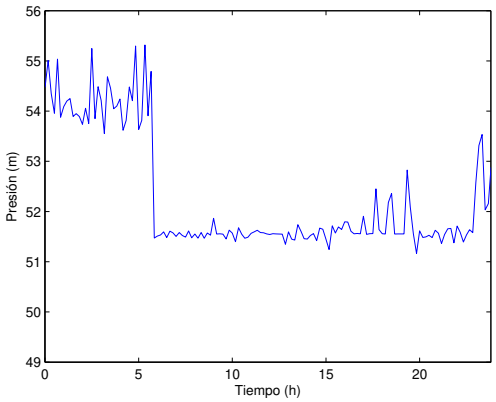


Figura 7: Presión en el nodo 6 aplicando control monovariable $P_{SP_6} = 51{,}55\ m$ con PRV2 y antiwindup

una interfaz gráfica que podemos observar en la figura 5. En la parte izquierda se introducen los datos correspondientes a la simulación puesto que el benchmark trabaja con la simulación ejecutada en EPANET del sistema controlado desde Matlab. En la parte derecha se introducen las características del control a efectuar. En caso de marcar el control multivariable aparece un cuadro en la parte inferior para configurarlo. El benchmark y el material didáctico disponible está pensado para trabajar las siguientes estructuras de control:

- Control monovariable usando un actuador (PRV1 o PRV2) con un integrador
- Inclusión del antiwindup
- Control rango partido (usando ambas válvulas)
- Control multivariable

Inicialmente se decide realizar el control con un único actuador. Se escoge controlar la presión en el nodo más crítico, que es aquel cuya presión se observó mínima durante la simulación en lazo abierto (figura 3). Como consigna se elige la presión mínima que alcanzó, en este caso, el nodo 6: $P_{SP_6} = 51{,}55\ m$. Se elige la válvula PRV2 como actuador puesto que es la más próxima, por lo cual el efecto será mayor. Como controlador se usa un integrador con el fin de eliminar el error. El sistema carece de dinámica y es el integrador quien la introduce. La constante unitaria del integrador significa que a cada muestra se añade a la acción de control el error anterior (ambas variables son metros). En la figura 6 se muestran los resultados de este control, donde se pueden observar dos fenómenos. Por un lado, durante la noche, debido a la influencia de la otra válvula, el actuador no puede reducir más la presión puesto que llega a su saturación (completamente cerrada). Además se observa que la presión cae al empezar el día (6 de la madrugada) pero no se corrige hasta pasar un tiempo demasiado largo, fenómeno relacionado con el windup.

En la figura 7 se observa que al introducir el antiwindup en el controlador la respuesta del control mejora. Obviamente la saturación sigue presente, pero no tiene un efecto pernicioso en el control una vez se ha entrado en la zona no saturada. Esta saturación es debida al efecto de la segunda válvula de la red (PRV1), que fija una presión más alta que la de la consigna, impidiendo que la presión pueda bajar en el nodo 6 debido al poco caudal nocturno.

Si se usan los dos actuadores disponibles con una estructura rango partido, compartiendo ambos actuadores la misma ley de control, el control mejora

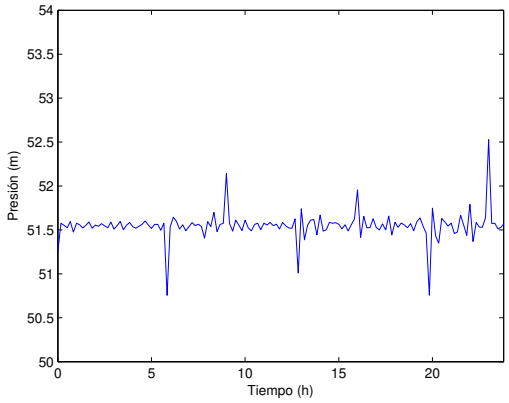


Figura 8: Presión en el nodo 6 aplicando control monovariable rango partido $P_{SP_6} = 51{,}55\ m$ con antiwindup

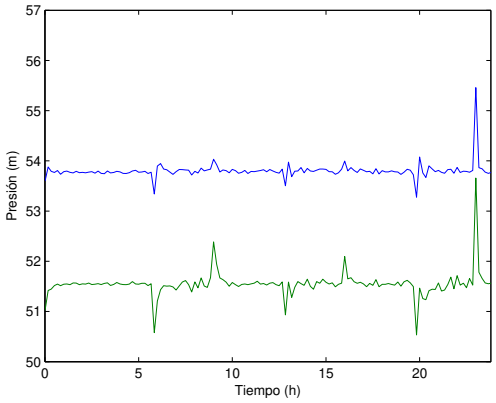


Figura 10: Presión en los nodos 5 y 6 aplicando control multivariable $P_{SP_5} = 53{,}75\ m$; $P_{SP_6} = 51{,}55\ m$ con antiwindup

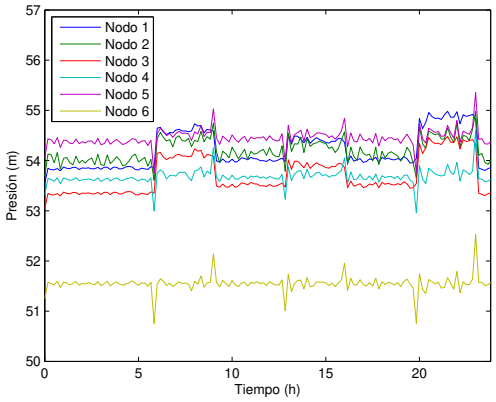


Figura 9: Presión en los seis nodos aplicando control monovariable rango partido en el nodo 6 $P_{SP_6} = 51{,}55\ m$ con antiwindup

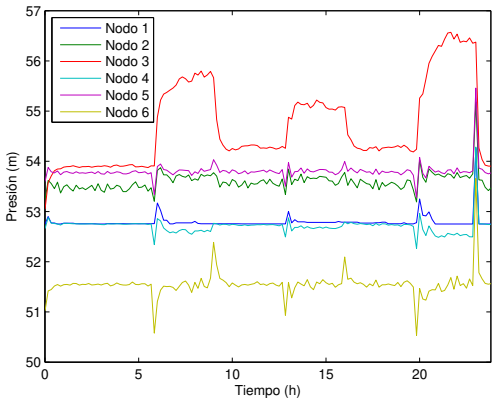


Figura 11: Presión en los seis nodos aplicando control multivariable $P_{SP_5} = 53{,}75\ m$; $P_{SP_6} = 51{,}55\ m$ con antiwindup

notablemente. En la figura 8 se observa un seguimiento perfecto de la consigna donde las perturbaciones debidas al cambio de las demandas (picos en el gráfico) se corrigen en la siguiente muestra (diez minutos).

Las presiones resultantes en los otros nodos de la red (figura 9) pueden obtenerse mediante el módulo de simulación, definiendo como consignas de las válvulas las acciones de control generadas en el módulo de presión.

Finalmente, el módulo de control permite controlar simultáneamente la presión en más de un punto de la red aprovechando que se dispone de dos actuadores. Se elige mantener el nodo más crítico ($P_{SP_6} = 51{,}55\ m$) mientras el nodo con presión máxima también se reduce a su presión mínima observada en el control anterior ($P_{SP_5} = 53{,}75\ m$). Se observa en la figura 10 como ambas presiones se mantienen en la consigna deseada. El control es multivariable con dos lazos sin desacopladores. En la figura 11 se observa que a pesar de conseguir el seguimiento de las consignas el nodo 3 supera la presión del nodo 5, que anteriormente era el de máxima presión. Esto se debe a que el nodo 3 es el nodo de entrada a la red y asume las variaciones de presión debidas a los cambios de demanda.

No se ha implementado un control con desacopladores puesto que el acoplamiento entre ambos lazos si bien es fuerte no se contradice con la acción directa. Simplemente hay combinaciones de consignas imposibles.

Se propone como alternativa un control monovariable rango partido para controlar la presión media de los nodos 5 y 6 a la media de sus mínimos $P_{SPmedia_{5,6}} = 52{,}65\ m$. El comportamiento de los nodos controlados es parecido al del control mul-

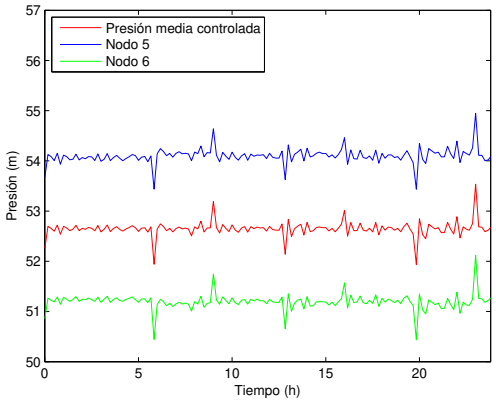


Figura 12: Presión en los nodos 5, 6, y media resultante aplicando control monovariable rango partido sobre el promedio de 5 y 6 $P_{SP_{media}} = 52{,}65\ m$

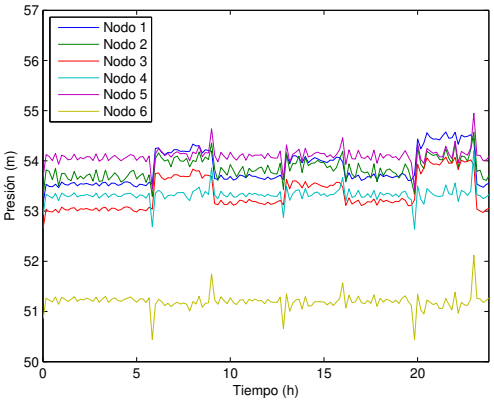


Figura 13: Presión en los seis nodos aplicando control monovariable rango partido sobre el promedio de 5 y 6 $P_{SP_{media}} = 52{,}65\ m$ con antiwindup

tivariable, tal como se observa en la figura 12. Sin embargo, en la figura 13 se observa como el conjunto de presiones de la red es más homogéneo, ya que las variaciones de presión debidas a los cambios de demanda se absorben entre todos los nodos, evitando así presiones altas en el nodo 3.

6. CONCLUSIONES

Este artículo presenta una interfaz gráfica que combina la capacidad de simulación de redes de distribución de agua de EPANET con el potencial de programación y de cálculo de Matlab.

Se presentan el módulo principal, con el que se puede obtener información sobre los elementos de la red y realizar simulaciones de la misma; y un módulo extra que permite realizar diferentes estructuras de control de presión sobre uno o varios nodos de la red.

El potencial de la interfaz mostrada radica en la capacidad de programar módulos personalizados que puedan utilizar funciones ya incorporadas en el módulo principal, así como la libertad de trabajo sobre los resultados mediante el uso de comandos Matlab.

El autor del trabajo desarrollará diversos módulos extra durante el transcurso de su tesis doctoral.

Agradecimientos

Este trabajo está financiado en parte por el proyecto de MINECO ref. DPI2011-2643 (SHERECS); y por la Universidad Politécnica de Catalunya.

Referencias

[1] M. A. Bryds and B. Ulanicki. *Operational Control of Water Systems: Structures, Algorithms and Applications*. Prentice Hall International, 1994.

[2] F. Nejjari, R. Pérez, V. Puig, J. Quevedo, R. Sarrate, M. Cugueró, G. Sanz, and J. Mirats. Abnormal quality detection and isolation in water distribution networks using simulation models. *Drinking Water Engineering and Science Discussions*, 5(1):435–453, Aug. 2012.

[3] R. Pérez, V. Puig, J. Pascual, J. Quevedo, E. Landeros, and A. Peralta. Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks. *Control Engineering Practice*, 19(10):1157–1167, Oct. 2011.

[4] L. Rossman. *EPANET 2 Users Manual*. Water Supply and Water Resources Division, National Risk Management Research Laboratory, 2000.

[5] G. Sanz, R. Pérez, and R. Sánchez. Pressure Control of a Large Scale Water Network using Integral Action. In *IFAC Conference on Advances in PID Control*, Brescia, 2012.

[6] M. Sterling and A. Bargiela. Leakage reduction by optimised control of valves in water networks. *Transactions of the Institute of Measurement and Control*, 6(6):293–298, Oct. 1984.

[7] T. Walski, D. Chase, D. Savic, W. Grayman, S. Beckwith, and E. Koelle. *Advanced Water Distribution Modeling and Management*. Haestad Press, 2003.

Aprendizaje colaborativo e interinstitucional en entornos virtuales

Antonio Javier Barragán Piña[1], José Manuel Andújar Márquez[1], Yolanda Ceada Garrido[1]
{antonio.barragan, andujar}@diesia.uhu.es, yolanda.ceada@sc.uhu.es
Eloy Irigoyen Gordo[2], Fernando Artaza Fano[2], Vicente Gómez Garay[2]
{eloy.irigoyen,fernando.artaza,vicente.gomez}@ehu.es

[1]Universidad de Huelva, [2]Universidad de el País Vasco

Resumen

El objetivo principal del proyecto que se presenta en este artículo, es crear un recurso didáctico que permita la participación activa de los alumnos en el proceso de enseñanza-aprendizaje, estimulando la motivación de los estudiantes y desarrollando sus competencias de trabajo en grupo y en equipos multidisciplinares e interuniversitarios.

Palabras clave: Aprendizaje, colaborativo, docencia, TICs, virtual

1. Introducción

El fomento del aprendizaje autónomo por parte del alumno es uno de los objetivos perseguidos por el Espacio Europeo de Educación Superior (EEES). Dicho aprendizaje debe permitir al discente adquirir las competencias adecuadas para organizar la información disponible, transformarla en conocimientos y desempeñar de manera válida su actividad profesional, para lo cual el alumno debe participar activamente en dicho proceso de enseñanza-aprendizaje. Las metodologías a emplear deben enfatizar, en la medida de lo posible, el autoaprendizaje, trabajo guiado, la conexión entre teoría y práctica, el acercamiento a la realidad laboral y el aprendizaje colaborativo y en grupo [22].

La Declaración de Bolonia [18] estableció la colaboración institucional como una de las claves para la consecución del objetivo de potenciar las dimensiones europeas en la enseñanza superior *la promoción de las dimensiones europeas necesarias en la enseñanza superior, sobre todo en lo que respecta al desarrollo curricular, colaboración interinstitucional, planes de movilidad y programas integrados de estudio, formación e investigación.* Los estudiantes de los distintos Grados de Ingeniería Industrial e Informática deben adquirir competencias sobre organización, planificación, trabajo en equipo, habilidades en las relaciones interpersonales y capacidades para trabajo en equipos multidisciplinares. La importancia de estas competencias en los distintos planes de estudio de ingeniería vienen a reflejar el mundo laboral actual, el cual se caracteriza por el trabajo en equipo y una fuerte dependencia de habilidades de comunicación y relaciones interpersonales propias de este tipo de escenario, junto con el manejo, cada vez más presentes, de entornos virtuales de comunicación y telecolaboración.

Dado los antecedentes presentados, la enseñanza y el aprendizaje colaborativos deben considerarse como imprescindibles en el sistema universitario actual, por lo que los profesores debemos adaptar nuestros temarios y metodologías de enseñanza-aprendizaje para conseguir que los discentes adquieran dichas competencias, además, por supuesto, de las propias competencias específicas de cada asignatura. Se ha de entender por enseñanza colaborativa *cualquier experiencia académica en la cual dos profesores trabajan juntos en el diseño y enseñanza de un curso en el que se utilizan técnicas de aprendizaje en grupo* [17]; y por aprendizaje colaborativo o cooperativo *aquel en el que las metas de los miembros del grupo están compartidas y cada individuo alcanza su objetivo sólo si también consiguen sus compañeros los suyos* [6]. Desde el punto de vista de la teoría pedagógica constructivista, el término aprendizaje colaborativo se centra en la interdependencia social, y se sostiene que los estudiantes consolidan su aprendizaje mediante la enseñanza de unos a otros [1].

En este trabajo se propone la utilización del simulador de física realista de vehículos TORCS (http://torcs.sourceforge.net) como una herramienta de trabajo para la enseñanza de la automática, robótica, programación e inteligencia artificial y control inteligente. Este simulador permite la práctica de múltiples conceptos relacionados con dichas materias [3, 4, 10], tanto a nivel básico como avanzado. Dada la complejidad que conlleva el control completo de un vehículo, se propone la realización de grupos heterogéneos de alumnos procedentes de distintas asignaturas, titulaciones y universidades. Cada uno de estos alumnos deberá aportar sus conocimientos para que su grupo consiga realizar la tarea de forma satisfactoria, fomentando el trabajo en equipo y la colaboración a través de un entorno virtual (telecolaboración).



Figura 1: Imagen de TORCS.

La experiencia docente que presentamos en este artículo se está desarrollando conjuntamente por profesores la Universidad de Huelva y la Universidad del País Vasco. Los participantes de la Universidad de Huelva lo conforman un grupo amplio de profesores de los Departamentos de Ingeniería Electrónica, Sistemas Informáticos y Automática, y de Tecnologías de la Información, mientras que desde la Universidad del País Vasco participan varios profesores del Departamento de Ingeniería de Sistemas y Automática. Esta unión de tres departamentos y dos universidades conforma un grupo heterogéneo de docentes cuyas especialidades cubren la automática, el control de sistemas complejos, la programación básica y avanzada y la utilización de técnicas de inteligencia artificial aplicadas a la robótica y el control.

Este artículo está organizado en las siguientes secciones: en la sección 1 se han presentado los antecedentes y motivaciones del proyecto a desarrollar, a continuación, en la sección 2 se presentará en simulador a emplear en el proyecto. La sección 3 describe el proyecto que se pretende realizar y las ventajas docentes que se espera obtener. Finalmente se presentan algunas conclusiones y trabajos futuros.

2. TORCS

TORCS es un simulador 3D multiplataforma, cuyo código fuente está disponible bajo licencia General Public License GPL v2. Gracias al carácter de software libre, su modularidad y extensibilidad, TORCS se ha empleado como base de muchos trabajos de investigación: cálculo automatizado de configuraciones de automóviles [9], conducción basada en al copia de la conducta de un conductor humano [20], conducción automática [19], etc. Desde 2008, existen varios concursos organizados en la *IEEE Conference on Computational Intelligence and Games* que se centran en la aplicación de técnicas de inteligencia artificial a la conducción del vehículo en una carrera, y que están basados en este simulador; lo que demuestra su versatilidad e interés por parte de la comunidad científica en su uso. Empleando el parche disponible para la *IEEE Conference on Computational Intelligence and Games* (CIG) [11], el simulador TORCS se convierte en un servidor cuyo esquema se muestra en la figura 2. Este servidor admite la conexión de múltiples clientes a través de Internet, lo que facilita la competición de distintos controladores (bots) de forma simultánea.

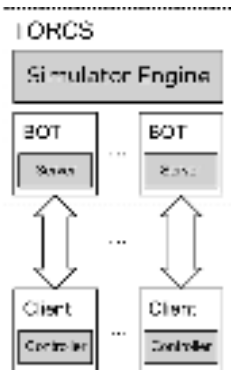


Figura 2: Configuración Cliente-Servidor.

3. Aprendizaje colaborativo e interinstitucional

El objetivo principal del proyecto que se presenta en este artículo, es crear un recurso didáctico que permita la participación activa de los alumnos en el proceso de enseñanza-aprendizaje, estimulando la motivación de los estudiantes y desarrollando sus competencias de trabajo en grupo y en equipos multidisciplinares e interuniversitarios. Conjuntamente con el objetivo principal del proyecto se pretenden cubrir una serie de objetivos secundarios más genéricos:

- Fomentar la creación de material docente dinámico en un contexto virtual.
- Establecer un vínculo de colaboración interuniversitario entre profesores de las Universidades de Huelva y el País Vasco, el cual permita el enriquecimiento mutuo y la participación en nuevas experiencias de innovación docente y, en definitiva, de la mejora de la calidad en la docencia.
- Motivar al alumnado y reducir las tasas de abandono y fracaso fomentando su incorporación en la creación de material docente e incentivando el seguimiento continuado de materia impartida en las distintas asignaturas involucradas en el presente proyecto.

La necesidad de realizar prácticas de laboratorio en ingeniería, y por supuesto en otras disciplinas educativas, para permitir que los estudiantes adquieran habilidades en la solución de problemas reales, es incuestionable. Sin embargo, el intento de cumplir completamente esta necesidad puede presentar problemas logísticos, económicos y educativos, incluyendo los siguientes [2]:

- Los recursos disponibles en los laboratorios, tanto de software como de hardware, son limitados.
- Los modelos reales de laboratorio son complejos y costosos, y es muy complicado poder proporcionar material individualizado para cada estudiante.
- Los horarios de laboratorio deben ajustarse a propios de la universidad, en lugar de adaptarse a la disponibilidad del discente.
- El tiempo disponible para cada grupo de trabajo siempre es insuficiente, ya que los laboratorios son compartidos entre los diferentes materias y titulaciones.

La mayoría de los laboratorios virtuales de entornos de aprendizaje son diseñados para ser utilizados individualmente, por lo que no se permite el trabajo en grupo o la colaboración entre estudiantes y profesores [16]. En este trabajo se propone un sistema de colaboración entre estudiantes de distintas materias, titulaciones e incluso universidades a través de un simulador basado en un esquema cliente servidor, el cual permite la interacción simultánea de múltiples clientes con el simulador. El simulador funcionará como un servicio online para la realización de prácticas remotas sobre un coche virtual de física realista. Este sistema permitirá la realización de prácticas desde los niveles más básicos, por ejemplo el control de velocidad del vehículo, hasta los más avanzados, como el control del vehículo completo o la conducción a la máxima velocidad posible (competición). El control automático de vehículos es una histórica aspiración de las empresas de automoción [12] que poco a poco se está convirtiendo en realidad [13], y que implica la utilización de tecnologías y conocimientos de múltiples áreas del saber (instrumentación, modelado, control, robótica, programación, etc.):

- La gran mayoría de los vehículos incorporan actualmente de serie un sistema de control automático de velocidad, el cual permite mantener el vehículo a una velocidad constante independientemente de la inclinación de la carretera, lo que aumenta el confort de la conducción de forma importante en grandes desplazamientos y en situaciones de tráfico lento.
- Los dispositivos de frenado de emergencia y prevención de colisiones se han convertido en uno de los nuevos sistemas activos de protección que incorporan de serie los nuevos vehículos [14, 21].
- El cambio de marchas automático, pese a ser un sistema existente desde hace décadas, ha sufrido una fuerte renovación en los últimos años para permitir una mayor adecuación del sistema al estilo de conducción del propietario del vehículo o a las condiciones del entorno.
- La automatización se está extendiendo al resto de sistemas de los vehículo, incluyendo la conducción de éstos, fundamentalmente con la intención de aumentar la seguridad en los desplazamientos y reducir el número de accidentes [15].
- En varios estados de Estados Unidos ya existen leyes que regulan la circulación de vehículos autónomos por sus carreteras http://www.dmvnv.com/news/12001-regulations-for-self-driving-cars.htm
- Google Inc, la empresa detrás del famoso buscador Google, posee un proyecto de vehículo autónomo que ya ha recorrido más de 500.000 Km por carreteras públicas de Estados Unidos.
- El fabricante sueco de automóviles Volvo Car Corporation, con su proyecto SARTRE financiado con fondos europeos, realizó en mayo los primeros test en carretera abierta, y con otros usuarios en la vía, de un tren de carretera formado por un vehículo que actúa como guía y por otros coches dirigidos de forma electrónica. http://www.sartre-project.eu/en/Sidor/default.aspx
- Muchos otros fabricantes, como Toyota, Nissan o Hitachi (en coches eléctricos), también tienen proyectos para el desarrollo de vehículos autotripulados.

Por lo tanto, se ofrecerá a los estudiantes un entorno virtual y colaborativo para la realización de prácticas de diversa índole, inspirado en las últimas tecnologías del mundo de la automoción. Desde nuestro punto de vista, proveer a los alumnos de esta herramienta supondrá un enriquecimiento en la calidad de las prácticas de las asignaturas involucradas, con unos desarrollos de última generación, pero realistas y acordes a la actualidad del mercado y la industria más avanzada. La colaboración entre distintas asignaturas, titulaciones y

universidades permitirá a los alumnos desarrollar, entre otras muchas competencias, habilidades de trabajo en equipo, habilidades en las relaciones interpersonales y capacidades para trabajo en equipos multidisciplinares; pero también permitirá a los discentes ver en un único sistema los conceptos aplicados de múltiples asignaturas de su plan de estudios (instrumentación, mecánica, control, robótica, inteligencia artificial, ...). Una de las grandes ventajas del sistema propuesto, radica en que la modularidad y complejidad del mismo permite la creación de actividades con distintos grado de complejidad, incluso actividades grupales entre alumnos de distintas asignaturas, titulaciones y universidades. Por ejemplo, para que un vehículo circule a la máxima velocidad posible por una vía, es necesario diseñar la trazada óptima que éste ha de seguir, un trabajo propio de asignaturas relacionadas con la robótica móvil; sin embargo, es necesario realizar un control del giro del volante para que el vehículo se mantenga sobre la trazada diseñada, el cual es un trabajo más propio de asignaturas de control; además, todo ello debe programarse para la creación del cliente, lo cual implica conocimientos de programación.

Para establecer los canales necesarios de comunicación entre los estudiantes, se propone la utilización de técnicas de telecolaboración basadas en comunicación tanto síncrona como asíncrona. Normalmente, la telecolaboración usa principalmente herramientas de comunicación asíncronas [8], pero éste tipo de comunicación puede causar sentimientos de aislamiento en los estudiantes y, por lo tanto, reduce su motivación [5]. Así pues, sin olvidar que las redes sociales y los correos electrónicos personales tienen un gran potencial para la telecolaboración [7], es interesante fomentar entre los estudiantes el uso de alternativas síncronas, como los chats o audio/vídeo-conferencias.

Un ejemplo de propuesta de práctica conjunta podría ser la siguiente: un alumno que haya finalizado sus estudios de grado (o máster) y que pretenda realizar su proyecto final de carrera (o máster) sobre la plataforma presentada haría las veces de coordinador del proyecto. Este alumno, además de realizar el trabajo más complejo dada su mayor formación, podría colaborar con alumnos de asignaturas de control, robótica y/o programación para la realización de algunos de los trabajos necesarios, como el diseño del control de velocidad, la planificación de las trayectorias a seguir, o la programación del bot, respectivamente. Estos grupos estarán formados por estudiantes de distintas asignaturas de las dos Universidades participantes. Este planteamiento permite adquirir a los alumnos las competencias sobre organización, planificación, trabajo en equipo, habilidades en las relaciones interpersonales y capacidades para trabajo en equipos multidisciplinares requeridas por los planes de estudio actuales, además de ofrecer una prácticas atractivas y novedosas a los alumnos de los cursos inferiores.

4. Conclusiones y trabajos futuros

En este artículo se ha propuesto un proyecto para crear un recurso didáctico que permita la participación activa de los alumnos en el proceso de enseñanza-aprendizaje, estimulando la motivación de los estudiantes y desarrollando sus competencias de trabajo en grupo y en equipos multidisciplinares e interuniversitarios. Este proyecto ofrecerá a los estudiantes un entorno virtual y colaborativo para la realización de prácticas de diversa índole, inspirado en las últimas tecnologías del mundo de la automoción, lo que esperamos pueda revertir en un mayor grado de comprensión e índice de aprobado de las asignaturas involucradas.

Actualmente, el equipo de profesores que participa en este trabajo está desarrollando las herramientas software necesarias para la realización del proyecto, así como la preparación de los manuales necesarios para los alumnos; por lo que esperamos poder presentar resultados al respecto en breve.

Agradecimientos

Este trabajo ha sido financiado por el Servicio de Innovación Docente de la Universidad de Huelva a través del proyecto de investigación en docencia universitaria PIE1315062.

Referencias

[1] Maryam Alavi. Computer-mediated collaborative learning: An empirical evaluation. *MIS Quarterly: Management Information Systems*, 18(2):159–174, June 1994.

[2] José Manuel Andújar, Andrés Mejías, and Marco Antonio Márquez. Augmented reality for the improvement of remote laboratories: an augmented remote laboratory. *IEEE Transactions on Education*, 54(3):492–500, August 2011.

[3] Christos Athanasiadis, Damianos Galanopoulos, and Anastasios Tefas. Progressive neural network training for the open racing car simulator. In *2012 IEEE Conference on Computational Intelligence and Games, CIG 2012*, pages 116–123, September 2012.

[4] Matteo Botta, Vincenzo Gautieri, Daniele Loiacono, and Pier Luca Lanzi. Evolving the

optimal racing line in a high-end racing game. In *2012 IEEE Conference on Computational Intelligence and Games, CIG 2012*, pages 108–115, September 2012.

[5] Maged N. Kamel Boulos, Andrea D. Taylor, and Alice Breton. A synchronous communication experiment within an online distance learning program: A case study. *Telemedicine Journal and e-Health*, 11(5):583–593, October 2005.

[6] Fernando Doménech Betoret. *El proceso de enseñanza-aprendizaje universitario: aspectos teóricos y prácticos*. Col.lecció universitas. Universidad Jaume I. Servicio de Comunicación y Publicaciones, 1999.

[7] Arturo Jaime, César Domínguez, Ana Isabel Díez Sánchez, and José Miguel Blanco. Interuniversity telecollaboration to improve academic results and identify preferred communication tools. *Computers and Education*, 64:63–69, May 2013.

[8] Carlos A. Jara, Francisco A. Candelas, Fernando Torres, Sebastián Dormido, Francisco Esquembre, and Óscar Reinoso. Real-time collaboration of virtual laboratories through the internet. *Computers & Education*, 52(1):126–140, January 2009.

[9] M. Kemmerling and M. Preuss. Automatic adaptation to generated content via car setup optimization in TORCS. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 131–138, August 2010.

[10] KyungJoong Kim, JunHo Seo, JungGuk Park, and JoongChae Na. Generalization of torcs car racing controllers with artificial neural networks and linear regression analysis. *Neurocomputing*, 88:87–99, July 2012.

[11] Daniele Loiacono, Luigi Cardamone, and Pier Luca Lanzi. *Simulated Car Racing Championship Competition Software Manual*. Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, 2013.

[12] Wesley R. Master. State of the art remote and automatic control of vehicles. *SAE Technical Papers*, September 1962.

[13] Vicente Milanés and Luis Miguel Bergasa. Introduction to the special issue on "New trends towards automatic vehicle control and perception systems". *Sensors (Switzerland)*, 13(5):5712–5719, May 2013.

[14] Vicente Milanés, Enrique Onieva, Joshué Pérez, Javier Simó-Reigadas, Carlos R. González, and Teresa De Pedro. Making transport safer: V2V-based automated emergency braking system. *Transport*, 26(3):290–302, September 2011.

[15] Vicente Milanés, Enrique Onieva, Joshué Pérez, Jorge Villagrá, Jo Godoy, Javier L. Alonso, Carlos R. González, Teresa De Pedro, and Ricardo R. García. AUTOPIA program advances: How to automate the traffic? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6928 LNCS(Part 2):374–381, February 2012.

[16] Lorenzo Moreno, Carina S. Gonzalez, Iván Castilla, Evelio José Gonzalez, and José Francisco Sigut. Applying a constructivist and collaborative methodological approach in engineering education. *Computers and Education*, 49(3):891–915, November 2007.

[17] Ann I. Nevin, Jacqueline S. Thousand, and Richard A. Villa. Collaborative teaching for teacher educators. What does the research say? *Teaching and Teacher Education*, 25(4):569–574, May 2009.

[18] European Ministers of Education. *Bologna Declaration*. The European Higher Education Area, June 1999.

[19] Enrique Onieva, David Alejandro Pelta, Jorge Godoy, Vicente Milanés, and Joshué Pérez. An evolutionary tuned driving system for virtual car racing games: The AUTOPIA driver. *International Journal of Intelligent Systems*, 27(3):217–241, March 2012.

[20] Joshué G. C. Pérez, Vicente Milanés, Jorge Godoy, Jorge Villagrá, and Enrique Onieva. Cooperative controllers for highways based on human experience. *Expert Systems with Applications*, 40(4):1024–1033, March 2013.

[21] Ardalan Vahidi and Azim Eskandarian. Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 4(3):132–153, September 2003.

[22] Carmen Vizcarro Guarch, José Ramón Carrillo, and Juan José Sobrino. *Experiencias de innovación docente en la UCLM*. Aprendizaje y docencia universitaria. Ediciones de la Universidad de Castilla-La Mancha, 2011.

HERRAMIENTA DE SIMULACIÓN DEL COMPORTAMIENTO DE UN BIORREACTOR DISCONTINUO

Adrián Casimiro Álvarez, Javier de Pedro López, Antonio Coca Riega, Diego García-Ordás

{acasia00,jdepll00,acocar00}@estudiantes.unileon.es, dgaro@unileon.es

Resumen

Una simulación de un experimento real permite reducir costes y tiempo, ya que los cálculos necesarios no se hacen sobre el ambiente real, sino sobre un modelo basado en la situación real. En este caso se muestra cómo se ha creado una herramienta que permite realizar simulaciones sobre la bacteria Zymomonas mobilis *y conocer cuál es su tasa de crecimiento ante ciertos factores condicionantes como la cantidad de sustrato proporcionado.*

Keywords: Biorreactor, Biorreactor Discontinuo, Simulación, Aprendizaje, Educación

1. INTRODUCCIÓN

La simulación es una técnica para remplazar y/o amplificar experiencias del mundo real, permitiéndonos interactuar con ella y descubrir qué resultados nos encontraríamos ante diferentes entradas sin tener que alterar el estado real. La utilización de simulaciones en la educación puede ser una buena forma de desarrollar conocimientos, habilidades y atributos sin la necesidad de desperdiciar tiempo y recursos [11]. En algunos casos la simulación puede evitar riesgos innecesarios, como en el caso de la educación médica cuya experimentación real podría llegar a no ser éticamente correcta [7].

Las relaciones microbianas son procesos muy complejos influenciados por múltiples condiciones físicas como la temperatura, la dinámica del fluido y la transferencia de masa. Hay otros factores como el ambiente químico (niveles de nutrientes o concentraciones del producto) [2] que se deben tener en cuenta.

Hay muchos casos de microorganismos cuya morfología al igual que su comportamiento metabólico cambian al ritmo que lo hacen las condiciones experimentales. En el caso del crecimiento de la *Escherichia coli*, la cantidad de fuentes de carbono y los niveles oxígeno disponibles juegan un papel importante en sus procesos metabólicos y por tanto en su crecimiento [3]. El crecimiento aeróbico en exceso de glucosa lleva a la formación de acetato. La acumulación de productos finales como el acetato a altos niveles puede inhibir el crecimiento y afectar la síntesis de producto. Por otro lado, si la concentración de acetato es baja, se utiliza como una fuente de carbono secundaria prolongando el crecimiento de la célula [6].

Adicionalmente la concentración de gases como el oxígeno y dióxido de carbono en el medio afectan a la fisiología. El oxígeno es un factor limitante de crecimiento y por debajo de un valor crítico afecta a la tasa de crecimiento [4] pero al mismo tiempo tiene efectos inhibidores cuando está presente en exceso [8]. De manera similar, bajos niveles de dióxido de carbono estimulan el crecimiento, mientras que niveles altos tienen efectos inhibidores progresivos [1].

La habilidad para predecir con exactitud los niveles de oxígeno y CO_2 es de considerable interés ya que establece los niveles de producción y crecimiento. Los niveles de oxígeno y dióxido de carbono disueltos se ven afectados afectados por el consumo o la producción respectivamente por parte de los microorganismos, y por la tasa de transferencia entre los gases. Un ejemplo simple se aprecia en el hecho de que el oxígeno se disuelve poco en agua y es relativamente escaso en el aire (20,8 %).

Todos estos experimentos en la realidad se realizan en un biorreactor, que nos permite mantener el medio controlado y medir los diferentes factores que condicionan la reacción. Pero si lo que se pretende es realizar experimentos para saber cuáles podrían ser los valores óptimos para un proceso metabólico o conocer qué sucedería ante unos valores iniciales, podría llegar a considerarse un derroche de tiempo y de dinero. Este derroche se produce tanto en los materiales utilizados como en el tiempo que se ha dedicado en esperar a que termine el proceso manteniendo los factores en el rango deseado.

Aquí es donde entra la herramienta que presentamos en este artículo. Esta herramienta permite realizar simulaciones relativas al proceso de crecimiento de la *Zymomonas mobilis* (aunque el microorganismo es configurable), considerando un biorreactor discontinuo.

En las siguientes secciones se explicará el concepto de biorreactor discontinuo, así como el uso y desarrollo. Posteriormente, en la sección 4, comentaremos los resultados obtenidos tras usar la aplicación en un entorno de enseñanza, para finalmente detallar las conclusiones extraídas de nuestro trabajo.

2. FUNCIONAMIENTO DE UN BIOREACTOR DISCONTINUO

Un biorreactor es un recipiente o sistema que mantiene un ambiente biológicamente activo. En algunos casos, un biorreactor es un recipiente en el que se lleva a cabo un proceso químico que involucra organismos o sustancias bioquímicamente activas derivadas de dichos organismos. Este proceso puede ser aeróbico o anaeróbico dependiendo de la presencia de oxígeno o su ausencia para la reacción.

En términos generales, un biorreactor busca mantener ciertas condiciones ambientales propicias (pH, temperatura, concentración de oxígeno, etc.) al organismo o sustancia química que se cultiva. En función de los flujos de entrada y salida, la operación de un biorreactor puede ser de tres modos distintos:

- Discontinuo o de lote (batch) que será el que tratemos en este trabajo.
- De lote alimentado (fed-batch).
- Continuo o quimiostato.

Si analizamos los biorreactores como sistemas podemos presentar un enfoque más relacionado con conceptos de ingeniería. En un sistema discontinuo tenemos tres etapas: carga, arranque y descarga. Un ejemplo de operación discontinua sería aquella en la cuál hay que hacer un CIP (*cleaning in place*) limpieza y posteriormente un SIP (*steaming in place*), es decir una esterilización [2] [14] [10].

Teniendo en cuenta que los biorreactores están preparados para poder esterilizar en ciertas condiciones adversas y parte de nuestra tarea es dejar que entre aire estéril, podemos conseguir inocular y fermentar, sin necesidad de parar el biorreactor, lo que nos puede evitar una gran cantidad de tiempos muertos.

El tipo de biorreactor que hemos tratado de modelar y simular es un biorreactor discontinuo. Esencialmente es un biorreactor al uso como se ha definido con anterioridad, pero con la peculiaridad de que todo el material que se produce y que se introduce en el biorreactor está desde el comienzo de la reacción. Eso significa que el material producido no se extrae hasta que no finaliza toda la reacción, y que de la misma manera tampoco entra más material mientras que dicha reacción se lleva a cabo.

Aún no se ha mencionado, pero el objetivo final de un biorreactor es la producción de un material cuya fabricación mediante técnicas químicas o industriales resulta muy costosa o lleva una gran cantidad de tiempo. La ventaja de los biorreactores es que no requieren una alta inversión para la fabricación del material puesto que es el resultado de un proceso biológico. En muchas ocasiones no solo es más barato, sino que no se puede obtener de otra manera, como es el caso de la fermentación láctica [15].

La ventaja de realizar la simulación en este tipo de biorreactores es la simplificación y la optimización de los procesos, pues si no entra más sustrato que el que se encontraba al principio en el interior del biorreactor eliminamos una variable que cambia con el tiempo (simplificando considerablemente su simulación). Por otra parte si tampoco se introducen nuevos productos ni se extrae durante la reacción también eliminaremos parte de la complejidad.

Los principales elementos a tener en cuenta a la hora de simular un biorreactor son:

- *Biomasa*: recibe este nombre la cantidad de microorganismos medidos en gramos presentes en el medio de cultivo (donde se desarrolla la actividad de crecimiento) por unidad de volumen. La razón por la que se utiliza esta medida es porque al ser tan pequeños no tiene sentido realizar un conteo de manera individual.
- *Cantidad de sustrato*: denominamos sustrato al material que utilizarán los microorganismos para llevar a cabo la reacción química que les permite la supervivencia en el medio.
- *Cantidad de producto*: la cantidad de producto hace referencia a la masa de material obtenido como resultado de la reacción biológica al haberse producido una serie de reacciones químicas. El producto se encontrará en el medio de cultivo mezclado con sustrato, biomasa y el propio medio, por lo que tras su extracción necesitará de un proceso de separación. Este proceso de separación no forma parte del estudio.
- *El tiempo*: en un biorreactor tanto continuo como discontinuo el tiempo juega un papel fundamental en el proceso. En función del

tiempo irán cambiando los valores de todas las variables del proceso y hay que tenerlo muy presente para evitar errores. En la sección de problemas a modelar lo trataremos de manera más concreta.

- *Cinética de crecimiento*: también es algo común a todos los biorreactores. La cinética hace referencia a la forma en la que crecen los microorganismos a lo largo del tiempo. Normalmente suele ser exponencial, es decir, a más microorganismos, más se reproducen. Sin embargo, pueden presentar diferentes fases de crecimiento que también se deben tener en cuenta.

Esta dependencia del tiempo hace que modelar y simular un biorreactor tenga muchísimo sentido, pues los investigadores tienen que esperar físicamente a que todo el proceso finalice, con la repercusión en el coste que eso tiene.

El diseño de un biorreactor es una tarea normalmente compuesta por dos fases: diseño del equipamiento que se va a utilizar y el diseño computacional. Dado que nosotros nos centramos en el problema computacional, el diseño de las características del biorreactor pasará por evaluar las condiciones tanto externas como internas del propio biorreactor. Entre estas condiciones nos encontramos los cambios relacionados con el pH del medio, la temperatura, cinéticas de crecimiento y la muerte de los microorganismos.

Dada la complejidad de las operaciones, cuando hablemos sobre el problema a modelar estableceremos una serie de hipótesis para simplificar el problema, asumiendo determinados parámetros y extrapolando algunas características relacionadas con la reacción en cuestión.

2.1. Crecimiento de microorganismos

El crecimiento de los microorganismos viene determinado por la velocidad a la que se reproducen (varía en función de cada especie y generalmente está clasificado) y de la velocidad a la que suelen morir. La gráfica de la figura 1 describe el proceso general que siguen todos, aunque luego habrá variantes relacionadas con la velocidad de los procesos [19].

Las distintas fases [19] en las que se suele clasificar el proceso de crecimiento de microorganismos son:

- *Fase de lag*: es un pequeño período de tiempo desde que se realiza el cultivo hasta que los microorganismos adaptan al medio y comienzan a realizar sus funciones.

- *Fase exponencial*: en el momento en que empiezan a reproducirse, como el proceso suele estar relacionado con la subdivisión de las células, ese crecimiento se realiza de manera exponencial. Como consecuencia en poco tiempo se tiene un cultivo con una gran cantidad de microorganismos, algo que nos interesa en los biorreactores a la hora de producir el material a comercializar. Se rige por la fórmula $\frac{dN}{dt} = \mu N$. La variable μ hace referencia a un concepto que tendremos siempre presente en la simulación que es la velocidad de crecimiento, y N es la biomasa medida en número de microorganismos por unidad de espacio.

- *Fase estacionaria*: es la fase en la que los microorganismos comienzan a dejar de reproducirse como consecuencia de la falta de espacio o de sustrato en el medio de cultivo. Cuando las condiciones no son idóneas, los microorganismos dejan de realizar la función de reproducción y permanecen en el régimen denominado estacionario. Como es lógico, la fórmula que rige esta fase es $\frac{dN}{dt} = 0$.

- *Fase de muerte*: en la fase de muerte la cantidad de microorganismos comienza a descender como consecuencia directa de la falta de espacio. Al haber poco espacio donde moverse, y cada vez consumirse más sustrato, el medio comienza a dejar de tener las propiedades idóneas para la vida del microorganismo, ya que el producto de la reacción inunda el medio. Lo que para nosotros es un producto para los microorganismos es un deshecho, por lo que vivir entre sus deshechos les provoca la muerte. Esta fase vendrá regida por la constante k que da una relación de muerte de los microorganismos $\frac{dN}{dt} = -kN$.

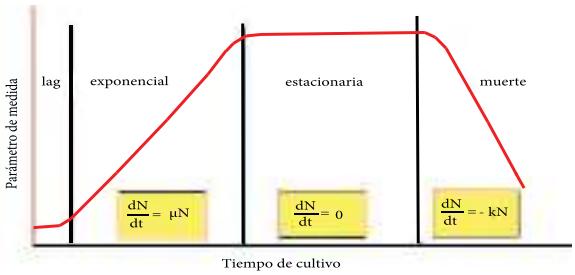


Figura 1: Gráfica de crecimiento de los microorganismos.

2.2. Ecuaciones cinéticas

Existen diferentes modelos matemáticos para el problema del crecimiento de microorganismos [12].

Nosotros nos hemos decantado por el modelo Mathusian [13], que a veces también recibe el nombre de exponencial simple. Es el más utilizado por la mayoría de los científicos por su simplicidad y porque la mayoría de los microorganismos sigue un modelo muy similar salvo raras excepciones.

Bajo condiciones ideales de crecimiento, se puede observar experimentalmente que la cantidad de biomasa aumenta exponencialmente con el tiempo. Este comportamiento se explica por el hecho de que todas las células tienen la misma probabilidad de multiplicarse. Por ello la cantidad de formación de biomasa es proporcional a la biomasa existente en un instante t. Podemos describir la reacción como:

$$r_X = \mu X \tag{1}$$

donde r_X es la probabilidad de crecimiento celular medido en kg/m^3h, X es la concentración de biomasa en mg/l y μ es la constante cinética de crecimiento medido en h^{-1}.

Para un sistema discontinuo como el que estamos tratando, este comportamiento exponencial se refleja así.

$$\frac{dX}{dt} = \mu X(t) \tag{2}$$

La solución analítica a esta ecuación diferencial de primer orden es:

$$X(t) = X_0 e^{\mu t} \tag{3}$$

donde X_0 es la concentración de biomasa inicial suponiendo $t = 0$. La concentración obtenida empleando este modelo se vuelve menos realista cuando el t es muy grande pues al ser exponencial los resultados de concentración dejan de ser válidos al ignorar las limitaciones de espacio.

La siguiente gráfica 2 representa el crecimiento basado en Mathusian.

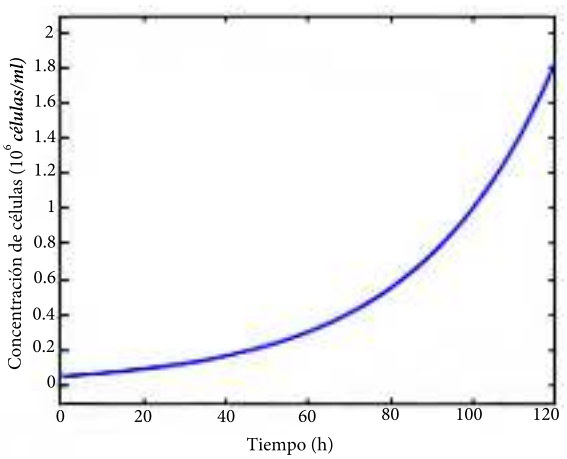


Figura 2: Solución del modelo Mathusian con $X_0 = 0{,}3 * 10^5$ y $\mu = 0{,}03$.

2.3. Órdenes de reacción

En cinética química, el orden de reacción con respecto a cierto reactivo es definido como la potencia (exponencial) a la cuál su término de concentración en la ecuación de tasa es elevado. Sin embargo si hablamos de reacciones relacionadas con seres vivos se refiere a la velocidad de crecimiento y cómo ésta se ve afectada por la cantidad de sustrato necesario [18].

En cinética de microorganismos nos vamos a encontrar principalmente dos órdenes de reacción: orden 0 y orden 1.

Una reacción es de *orden 0* si la concentración del sustrato no afecta a la velocidad de la reacción y crecimiento. Es decir $R_A = k$. Si se trata de consumo de sustrato hay que colocarle un signo negativo delante de la fórmula.

Una reacción es de *orden 1* si la concentración del sustrato afecta a la velocidad de la reacción y de crecimiento, dando como resultado la siguiente expresión: $R_A = kC_A$. Igualmente que en el caso anterior si la reacción es de consumo entonces hay que colocar un signo negativo.

Tenemos que tener en cuenta que las constantes de velocidad varían siguiendo la ecuación de Arrhenius, muy utilizada en muchos problemas químicos [16].

$$k = Ae^{-\frac{Ea}{RT}}$$

donde:

- A es la constante de Arrhenius.
- e es el número de Euler.
- Ea es la energía de activación de la reacción.

- R es la constante universal de los gases ideales.
- T es la temperatura absoluta en grados Kelvin.
- k constante cinética.

Esta ecuación establece que si la temperatura aumenta la velocidad de la reacción aumenta de la misma manera, pero estableciéndose un umbral, a partir del cuál la velocidad deja de aumentar.

3. DESARROLLO Y USO DE LA APLICACIÓN

Llegados a este punto debemos de hacernos a la idea de la cantidad de variables y modelos que pueden influir en el modelado de un biorreactor tanto de manera real como computacional [17](en tanto que se trata de acercar al propio modelo real). En esta sección explicaremos los fundamentos matemáticos [5] necesarios para la resolución del problema de una forma un poco más profunda.

Para realizar el simulador lo primero que nos planteamos fue a qué preguntas queríamos responder con la simulación, es decir, orientamos la creación del simulador a una clase de problemas en concreto y a sus variables de salida ya que un simulador nunca responde a todos los problemas de un determinado ámbito. Dado que los principales puntos de interés de un biorreactor son tres variables fundamentales, nos centramos en la obtención de su valor.

Las tres variables fundamentales son:

- La cantidad de *biomasa*.
- La cantidad de *sustrato*.
- La cantidad de *producto*.

3.1. Cantidad de biomasa

El experimento a simular está pensado para la bacteria *Zymomonas mobilis* para la producción de etanol, pero puede extrapolarse a todas aquellas bacterias que se ajusten al modelo de Mathusian simplemente cambiando los valores de las constantes dependientes del microorganismo.

De ahora en adelante, todas las concentraciones de sustrato, producto y biomasa estarán referidas a una cantidad por unidad de volumen del recipiente que las contiene. De esta manera también podemos tener en cuenta de alguna manera el espacio en el que se desarrolla el proceso.

Como hemos adelantado, la cinética de crecimiento es un modelo que puede ser experimental o no y nos permite saber la forma en la que crecen los microorganismos. En este caso el proceso que se sigue en es la cinética Mathusian ya que simplifica los cálculos y funciona bien para pequeños lapsos de tiempo. Como consecuencia el crecimiento bacteriano estará determinado por:

$$\frac{\delta X_A}{\delta t} = \mu_{max} X_A$$

donde:

- X_A es la concentración de biomasa en gramos (g).
- μ_{max} es la velocidad máxima de la reacción medido en h^{-1}.

La primera hipótesis del experimento se corresponde con suponer que μ no varía con el tiempo y con el consumo del sustrato, pues empleamos siempre la μ máxima del microorganismo. Esto no sería cierto ya que esta velocidad no se alcanza hasta que no lleva un tiempo de ejecución del experimento. Sin embargo, dado que las variaciones de velocidad solo se producen al principio o al final (si tenemos en cuenta el espacio), y dado que es un biorreactor discontinuo (el flujo $Q = 0$) se verifica que:

$$\mu = \frac{\mu_{max} S}{K_s + S} \tag{4}$$

Debido a que la cantidad de sustrato siempre es mucho mayor que la constante del sustrato ($S >>>>> Ks$) entonces se verifica que $\mu = \mu_{max}$.

La segunda hipótesis que se realiza es que el rendimiento de las operaciones es ideal, es decir del 100 %, rechazando cualquier tipo de pérdida en calor, evaporación o fallos del proceso. En la realidad estos factores se monitorizan para mantenerlos en un estado ideal o aproximadamente ideal.

La tercera hipótesis, relacionada con el propio modelo Mathusian es que el proceso siempre se mantiene en fase exponencial sin pasar por ninguno de los declives propios del crecimiento de la biomasa. Esta suposición se realiza debido a que la parte que interesa a la industria dedicada a biorreactores es el proceso de crecimiento, y no tanto los declives de la reacción.

Si resolvemos la ecuación diferencial:

$$\int_{X_{A0}}^{X_A} \frac{\delta X_A}{X_A} = \int_{t_0}^{t} \mu_{max} \delta t \tag{5}$$

obtenemos como resultado:

$$X_a = X_{A0} e^{\mu_{max} t} \tag{6}$$

Dado un t en un instante dado y teniendo un X_{A0} inicial podemos calcular la cantidad de biomasa que hay para ese instante t gracias a esta cinética de crecimiento. Igualmente podemos calcular el tiempo que necesitamos para que se obtenga una cantidad de biomasa X_A despejando:

$$t = \frac{1}{\mu_{max}} ln \frac{X_A}{X_{A0}} \tag{7}$$

3.2. Cantidad de sustrato

El sustrato considerado para el experimento es la glucosa, pero puede personalizarse en función de las constantes del propio sustrato o bien pueden realizarse pequeñas variaciones en el simulador para ajustarlo a otros problemas.

Según la mayor parte de estudios [9] y referencias sobre el consumo de sustrato por parte de microorganismos viene ajustado por una serie de parámetros que iremos explicando a continuación. La fórmula que rige dicho consumo es experimental pero da muy buenos resultados salvo para algunas especies[9].

$$\frac{\delta S}{\delta t} = -(\frac{\mu_{max}}{Y_{XS}} + \frac{q_p}{Y_{PS}} + ms)X \tag{8}$$

A continuación explicamos en detalle cada uno de los elementos que intervienen en la reacción de consumo del sustrato.

- S se corresponde con la cantidad de sustrato existente en un tiempo t.
- t hace referencia al tiempo transcurrido tomando como referencia $t_0 = 0$.
- El signo $-$ implica que la reacción que se produce es de consumo, es decir, la biomasa está consumiendo el sustrato a partir de un sustrato inicial.
- μ_{max} es la misma variable que en el caso de la producción de la biomasa y se corresponde con la velocidad de la reacción.
- Y_{XS} es una constante de rendimiento relacionada con el consumo. Una bacteria parte del consumo la emplea en el crecimiento, por lo tanto hace referencia al rendimiento de la biomasa respecto del sustrato.
- Y_{PS} al igual que la variable anterior es una constante que hace referencia al rendimiento del producto respecto del sustrato. Se calcula como $Y_{PS} = Y_{PX}(g.producto/g.biomasa) * Y_{XS}(g.biomasa/g.sustrato)$.
- ms constante de la reacción, es decir, gasto adicional que produce la reacción en sí misma.
- q_p es una constante de gasto en producción, es decir, de creación de los deshechos de la reacción, que para nosotros será el producto. Se calcula como:

$$q_p = Y_{PX} * \mu_{max} + mp \tag{9}$$

siendo mp una constante de mantenimiento en supervivencia del microorganismos.

Para simplificar el proceso de integración de la ecuación diferencial a la parte que se encuentra entre paréntesis en la ecuación 8 se la suele renombrar como B ya que todo ello es constante y depende del microorganismo, la sustancia que sirve como sustrato y la sustancia producto. Los valores se extraen mediante bases de datos que se pueden consultar de manera gratuita y que tienen los resultados de datos experimentales.

Integrando el consumo de sustrato y aplicando la cinética de crecimiento Mathusian al crecimiento de microorganismos (mientras consumen evidentemente también crecen), obtenemos:

$$\frac{\delta S}{\delta t} = r_s = -BX_0 e^{\mu_{max} t} \tag{10}$$

$$\int \delta S = -BX_0 \int_{t_0}^{t} e^{\mu_{max} t} \delta t \tag{11}$$

$$S_0 - S = \frac{BX_0}{\mu_{max}} * (e^{\mu_{max} t} - 1) \tag{12}$$

3.3. Cantidad de producto

Si recordamos en el apartado anterior utilizamos la constante q_p como constante de gasto en producción al representa la cantidad de producto obtenido por unidad de tiempo. De manera lógica, podemos pensar que la velocidad de producción de etanol se obtiene mediante la siguiente deducción siguiendo un proceso similar al sustrato:

$$rp = \frac{\delta P}{\delta t} = q_p * X \tag{13}$$

$$\int_{P_0}^{P} \delta P = q_p X_0 \int_{t_0}^{t} e^{\mu_{max} t} \delta t \tag{14}$$

La resolución proporciona como resultado:

$$P = \frac{q_p X_0}{\mu_{max}} * (e^{\mu_{max} t} - 1) \tag{15}$$

3.4. Interfaz del programa

Con todo esto ya tendríamos modelados todos los elementos que necesitamos. Dado un tiempo t y una serie de constantes podemos calcular la cantidad de biomasa (X_A), la cantidad de sustrato restante (S) y la cantidad de producto resultante

(P). Además también es preciso calcular una serie de variables internas que se han visto en los apartados anteriores.

Por ello la primera y principal ventana del programa contiene todas las variables. En la parte superior las variables iniciales (figura 3) que se pueden modificar y debajo las variables intermedias (figura 4) y finales (figura 5) que se mostrarán tras realizar la simulación, todo esto creemos que facilita la comprensión de los resultados finales de un solo vistazo.

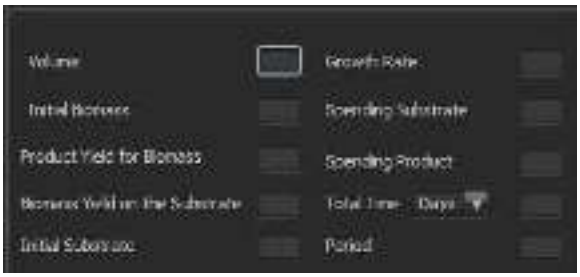


Figura 3: Panel de datos de entrada.



Figura 4: Panel de datos intermedios.



Figura 5: Panel de datos de salida.

En la parte inferior se encuentran los botones, uno para comenzar la simulación, otro para restaurar todos los valores y un último botón que nos llevará a otra ventana (figura 6) donde podremos ver las gráficas que muestran los valores que han tenido a lo largo del tiempo las variables de salida. Se puede cambiar entre cada gráfica mediante el panel superior y se puede interactuar con cada una para conocer el valor en un instante dado.

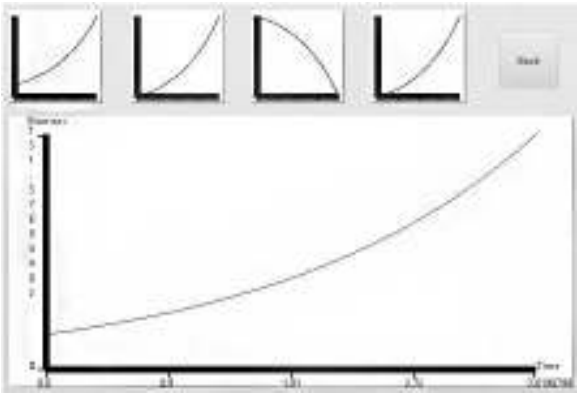


Figura 6: Panel de resultados.

4. RESULTADOS

Una vez puesta en uso hemos comprobado que las premisas se han cumplido. Cuando una simulación real puede llegar a tardar días, nuestro programa tarda menos de un minuto y muestra el valor de cada variable para cada uno de los días.

Por otro lado también se puede ver claramente que no ha sido necesaria la utilización de ningún microorganismo, ni de ningún sustrato para realizar las pruebas de cómo se comportaría la bacteria *Zymomonas mobilis* ante unos valores iniciales. Además no es necesario realizar cálculos matemáticos para comprobar cual sería el estado en un tiempo dado ya que lo realiza la herramienta.

Esto último beneficiaría a los estudiantes al no tener que estar realizando costosas pruebas para conocer cuál sería el estado de crecimiento en un momento dado de una bacteria. También permite introducir el uso de las bases de datos de microorganimos al impulsarles a buscar los valores necesarios para uno de los campos de entrada del programa, facilitando de una manera visual la compresión del proceso.

5. CONCLUSIONES

La simulación facilita mucho las tareas de aprendizaje, se reducen los costes y se evita perder tiempo. En este caso se ha visto cómo realizando la simulación de una bacteria se puede conocer cuál es su crecimiento, cuánto sustrato consume y cuanto producto se genera sin tener que realizar un experimento real. Es decir, se ha visto cómo la herramienta que hemos creado reduce drásticamente el tiempo y los costes necesarios en un experimento real, y elimina la necesidad de realizar los cálculos matemáticos para conocer cuál es el estado en un instante dado.

Referencias

[1] M. M. Kole D. Ward D. F. Gerson A. Lacoursiere, B. G. Thompson. Effects of carbon dioxide concentration on anaerobic fermentations of escherichia coli. *Applied Microbiology and Biotechnology.*, 23(5):404–406, feb. 1986.

[2] Maria Jesus Guardia Alba and Eloy García Calvo. Characterization of bioreaction processes: Aerobic escherichia coli cultures. *Journal of Biotechnology*, 84(2):107 – 118, 2000.

[3] Brian W. Boesch Amit Varma and Bernhard O. Palsson. Stoichiometric interpretation of escherichia coli glucose catabolism under various oxygenation rates. *Applied and Environmental Microbiology*, 59(8):2465–2473, aug. 1993.

[4] Tannahill A.L. Shuler M.L. Chen, J. Design of a system for the control of low dissolved oxygen concentrations: critical oxygen concentrations for azotobacter vinelandii and escherichia coli. *Biotechnology and Bioengineering.*, 27(2):151–155, feb. 1985.

[5] I.J. Dunn. *Biological Reaction Engineering: Dynamic Modelling Fundamentals With Simulation Examples*. Wiley-VCH GmbH & Company KGaA, 2003.

[6] Dijkhuizen L. Harder W. Strategies of mixed substrate utilization in microorganisms. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 297(1088):459–480, jun. 1982.

[7] Bates DW Jha AK, Duncan BW. Making health care safer: A critical analysis of patient safety practices. Technical Report 43, Agency for Healthcare Research and Quality U.S. Department of Health and Human Services, 2101 East Jefferson Street Rockville, MD 20852, jul. 2001.

[8] Sattler I. Zeeck A. Kaiser D., Onken U. Influence of increasing dissolved oxygen concentration on the formation of secondary metabolites by manumycin-producing streptomyces parvulus. *Applied Microbiology and Biotechnology.*, 27(3):309–312, may. 1994.

[9] K. Kovarova-Kovar and T. Egli. Growth kinetics of suspended microbial cells: from single-substrate-controlled growth to mixed-substrate kinetics. *Microbiol. Mol. Biol. Rev.*, 62(3):646–666, Sep 1998.

[10] Zelimir Kurtanjek. Principal component analysis of bioreactor fed-batch operation by computer simulation. *Mathematics and Computers in Simulation*, 44(3):287 – 294, 1997. Information Technology Interfaces.

[11] Fatimah Lateef. Simulation-based learning: Just like the real thin. *Journal of Emergencies, Trauma, and Shock*, 3(4):348–352, oct. 2010.

[12] Books Llc. *Empirical Laws: Zipf's Law, Empirical Validation, Malthusian Growth Model, Mooers'Law, Heaps'Law, First Law of Geography*. General Books LLC, 2010.

[13] T.R. Malthus and P. James. *An Essay on the Principle of Population*. Cambridge University Press, 1990.

[14] Mário Mihaľ, Richard Vereš, Jozef Markoš, and Vladimír Štefuca. Intensification of 2-phenylethanol production in fed-batch hybrid bioreactor: Biotransformations and simulations. *Chemical Engineering and Processing: Process Intensification*, 57–58(0):75 – 85, 2012.

[15] Noelia Rodríguez, José Manuel Salgado, Sandra Cortés, and José Manuel Domínguez. Antimicrobial activity of d-3-phenyllactic acid produced by fed-batch process against salmonella enterica. *Food Control*, 25(1):274 – 284, 2012.

[16] Juan M. Rodríguez-Díaz and M. Teresa Santos-Martín. Study of the best designs for modifications of the arrhenius equation. *Chemometrics and Intelligent Laboratory Systems*, 95(2):199 – 208, 2009.

[17] Hickman EF Morris BJ Blackburn TJ Ryll T. Smelko JP, Wiltberger KR. Performance of high intensity fed-batch mammalian cell cultures in disposable bioreactor systems. *Biotechnology progress*, 27(5):1358–64, sep. 2011.

[18] Richard J. Willson and A.E. Beezer. A mathematical approach for the calculation of reaction order for common solution phase reactions. *Thermochimica Acta*, 402(1–2):75 – 80, 2003.

[19] M. H. Zwietering, I. Jongenburger, F. M. Rombouts, and K. van't Riet. Modeling of the bacterial growth curve. *Applied Environmental Microbiology*, 56(6):1875–1881, Jun 1990.

PLATAFORMA ROBÓTICA DE BAJO COSTE Y RECURSOS LIMITADOS BASADA EN ARDUINO Y DISPOSITIVOS MÓVILES.

A. Soriano, L. Marín, R. Juan, J. Cazalilla, A. Valera, M. Vallés, P. Albertos
{ansovi, leomarpa, jcazalilla, giuprog, mvalles, pedro}@ai2.upv.es
Instituto de Automática e Informática Industrial
Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain

Resumen

Este trabajo presenta una plataforma didáctica basada en materiales de bajo coste que se encuentran en pleno auge de desarrollo y que frecuentemente se están implantando como herramientas de uso cotidiano en educación e investigación. Desde el punto de vista hardware, la plataforma consiste en un vehículo móvil capaz de manipular objetos mediante un brazo articulado. Para controlar el funcionamiento de ésta se ha dispuesto de una tarjeta Arduino a la que se ha añadido un dispositivo de comunicación vía Bluetooth y se ha desarrollado una aplicación para su control y manejo basada en el sistema operativo Android. La plataforma se ha programado completamente mediante el entorno Simulink, un software bien conocido en el campo de la automática y que recientemente ha incorporado librerías de programación para las tarjetas Arduino. Mediante estas librerías la programación de código queda encapsulada en el entorno gráfico Simulink, ofreciendo una abstracción absoluta del código C/C++ con el que originalmente se programa Arduino, lo cual facilita su uso en prácticas de laboratorio.
La plataforma permite la enseñanza de conceptos incluidos dentro del campo de la automática como el control de motores, el control de posición y velocidad, control de trayectorias, refuerzo de conocimientos sobre electrónica, manejo de comunicaciones inalámbricas, integración con sistemas Android… de una manera aplicada y sin requerir conocimientos avanzados de programación. Así, el artículo describe los elementos hardware utilizados para la construcción de la plataforma educacional y el procedimiento seguido para la construcción de ésta, el software utilizado para su programación y algunas actividades docentes que es posible realizar con la misma.

Palabras Clave: Automática, robótica, educación en automática, Arduino, Matlab-Simulink, plataforma móvil didáctica, tecnología pedagógica.

1 INTRODUCCIÓN

La automática y, en concreto, su enfoque hacia la robótica móvil, es un campo de éxito dentro de ciclos formativos, en la educación de bachillerato y de universidad debido a una combinación de factores [18]. El uso de una herramienta adecuada para el trabajo de laboratorio se convierte en una decisión crítica para promover la motivación de experimentar físicamente conceptos teóricos que se estudian en clase. Tiempo atrás, el material educativo que ofrecía el mercado resultaba altamente costoso. No obstante, hoy en día el campo de la educación en automática está en constante evolución; cada día surgen productos y plataformas nuevos sobre los que aplicar conceptos propios de esta área. Además existe una tendencia generalizada que está desencadenando la expansión de productos de hardware libre de bajo coste [7] con la intención de aportar simplicidad, estandarización y, en resumen, facilidades a las comunidades investigadoras o educativas para hacer uso de estos dispositivos.

Por otro lado, la educación en programación [15] es un área que comienza a abrirse paso entre las asignaturas de educación en institutos de bachillerato y ciclos formativos. Generalmente, la introducción a la programación no se lleva a cabo mediante el aprendizaje directo de un lenguaje de programación, si no que se realiza mediante organigramas o mediante programación en bloques por ser una metodología más intuitiva y mediante la cual la curva de aprendizaje es mucho más relajada. Existen diferentes herramientas multiplataforma con estas características como Scratch (desarrollada por Massachusetts Institute of Technology) [16], blockly (desarrollada por Google) [8], Alice (desarrollada por Carnegie Mellon University) [4], Greenfoot (University of Kent) [9] y herramientas con la misma filosofía pero que ofrecen un mayor potencial como Simulink [14] o Labview [10]. Este modo de programar ofrece rápidos resultados que ayudan a promover la motivación del alumnado.

Al mismo tiempo, la rápida evolución de la tecnología ha desembocado en un conocimiento generalizado del uso y la familiarización de sistemas inalámbricos como tabletas o teléfonos móviles. Hoy en día, con cierta edad, los jóvenes conocen al detalle

su manejo y prácticamente le dan un uso diario. El área de la educación poco a poco debe ser capaz de aprovechar la familiarización con esta tecnología y su inherente asociación al ocio para explotar su uso como herramienta educativa.

Este artículo presenta una plataforma orientada a la educación que integra dispositivos hardware de bajo coste junto con una programación sencilla e intuitiva mediante bloques, a la vez que incorpora el uso de tabletas o teléfonos móviles para interactuar con ella.

2 HARDWARE UTILIZADO

La plataforma debe ofrecer cierta versatilidad y al mismo tiempo, no encarecer su coste. El hardware escogido para la construcción del prototipo viene a ser una combinación de dispositivos que cada vez en mayor medida se utilizan como material didáctico para experimentos educacionales y/o demostrativos.
A continuación se detallan los dispositivos que forman la plataforma desarrollada.

2.1 TARJETA MICROCONTROLADORA

Para este trabajo se ha escogido una tarjeta Mega de Arduino (Figura 1) con un microcontrolador ATmega2560. Arduino [17] es un hardware libre de bajo coste fácilmente programable en lenguaje C/C++. Ofrece librerías para comunicación serie y conversor Analógico-Digital. Tiene 54 pines de entrada/salida digital, 16 entradas analógicas, 4 UARTs (puertos serie), una memoria flash de 256 KB, una frecuencia de reloj de 16 MHz y un puerto USB para conectarlo a cualquier computador. Las cuatro UARTs dotan al sistema de mayor flexibilidad permitiendo hacer funcionar cada una en una frecuencia de baudios distinta.



Figura 1: Arduino Mega

Para el control de motores, la misma compañía ofrece una tarjeta de expansión denominada Arduino Motor Shield la cual se monta sobre la misma placa MEGA y permite el control de dos motores de corriente continua con la posibilidad de invertir el sentido (integra un puente en hache).



Figura 2: Arduino motor shield

Para programar la placa Arduino el fabricante ofrece un software libre multiplataforma denominado IDE Arduino el cual está disponible en su propia página web. Es necesario también instalar los drivers para el puerto FTDI que incorpora la placa y conectarlo por USB al computador. El entorno del IDE Arduino consiste en un "sketch", un simple editor de texto que permite cortar/pegar y buscar/remplazar texto para escribir el código, un área de mensajes, una consola de texto útil para depurar y verificar el código, un compilador para código C/C++ y una barra de herramientas con botones para las funciones comunes como compilar el código, descargar el programa a la placa, creación, apertura y guardado de programas, monitorización serie, selección del puerto de comunicaciones asignado a la placa, etc.
No obstante, debido a que la transmisión de datos entre el PC y el Arduino es serie, la programación se puede llevar a cabo desde cualquier lenguaje de programación que ofrezca este soporte, desde java hasta Simulink, el cuál es el método empleado en este trabajo.
Debido a su bajo coste y su política de hardware libre, en pocos años Arduino ha conseguido extenderse ampliamente dentro de la comunidad investigadora y científica. Formando parte como cerebro de distintos proyectos como desarrollos de impresoras 3D [5], creación de robots [1], gestión de redes [6], controladoras de brazos manipuladores [11], investigación en vehículos submarinos [3], e incluso en sus versiones más ligeras, en vehículos aéreos no tripulados[13].

2.2 MOTORES LEGO MINDSTORMS NXT

Para construir el vehículo móvil correspondiente a la plataforma desarrollada se ha utilizado la plataforma LEGO Mindstorms NXT y se ha establecido la forma de controlar los motores de la plataforma desde la tarjeta Arduino. La plataforma LEGO Mindstorms NXT [12] es una herramienta muy utilizada hoy en día en asignaturas de introducción a la robótica y a la automática debido a su bajo coste y a su amplia oferta de posibilidades con las que explotar el producto. A pesar de que se trata de una plataforma cerrada, en cualquiera de sus versiones comerciales, los kits incluyen unos motores de corriente continua que incorporan encoders (Figura 3), permitiendo obtener una realimentación de la posición del motor al aplicar cierta acción de control.



Figura 3: Motor de LEGO NXT.

Estos motores funcionan a 9 voltios mediante un puerto I2C que contiene 6 cables. Dos de ellos son la alimentación de los encoders a 4.3 voltios y tierra. Otros dos se encargan de enviar la potencia mediante una señal PWM a una frecuencia que determina la velocidad y mediante su inversa, el sentido al que debe moverse el motor. Y los dos últimos se utilizan para la lectura de los encoders. Como se muestra en la Figura 4, los pulsos de los encoders están desfasados de modo que es posible determinar la dirección del motor según la secuencia que se detecte. De esta manera, a pesar de ofrecer una resolución real de 720 grados por vuelta, en la práctica se reducen a la mitad debido a la necesidad de usar dos ciclos para la detección del sentido del motor.



Figura 4. Relación entre las secuencias de encoders.

2.3 BRAZO ROBÓTICO

Con el fin de ampliar la versatilidad de la plataforma se ha incorporado a la misma un brazo robótico manipulador (Figura 5) de bajo coste con 4 grados de libertad.



Figura 5. Brazo robótico AL5D y controladora SSC-32.

El brazo se compone de 5 servomotores: uno en la base, uno en función de hombro, uno como codo, otro en la muñeca y el quinto para la abertura y cierre de la pinza.
El control de los motores se realiza mediante una controladora de servos SSC-32 que permite una conexión serie para el envío y recepción de comandos. Los comandos guardan el siguiente formato: "#X PY TZ <cr>". Donde X es el identificador del servo que se desea mover, Y corresponde a la posición donde se desea que se posicione el servo y Z es el tiempo que se desea que dure la operación. Para finalizar cualquier comando siempre es obligatorio incluir el carácter de retorno de carro.

2.4 TABLETA O DISPOSITIVO INALÁMBRICO BASADO EN ANDROID

La integración de dispositivos de uso cotidiano para la educación supone, además de un ahorro en coste de material, tener la ventaja de que el alumno ya conoce el manejo del aparato y también el hecho de que aprender a sacarle el máximo partido, aumenta su interés y motivación.
Existen multitud de dispositivos inalámbricos basados en el sistema operativo libre Android. Poseer un dispositivo Android tiene la ventaja de que al tratarse de software libre, la marca ofrece todo lo necesario para el desarrollo libre de aplicaciones. Además su publicación y distribución dentro del mercado Android puede ser totalmente gratuita, algo que no ocurre en otros sistemas.
Para este trabajo se ha utilizado una Tableta modelo Samsung Galaxy Note 10.1 (Figura 6) con una pantalla de 10.1 pulgadas a una resolución de 1280x800, con un procesador de cuatro núcleos a 1,4 GHz, A-GPS, una memoria de 32 GB, 2 GB de RAM, conexión WI-FI, Bluetooth y con opción a la conexión 3G.



Figura 6. Samsung Galaxy Note 10.1

No obstante, cualquier dispositivo Android con conectividad vía Bluetooth es suficiente para el desarrollo de la aplicación que se propone en este trabajo.

3 SOFTWARE UTILIZADO

En cuanto al software utilizado en este trabajo, se debe hacer una diferenciación de entre las distintas aplicaciones planteadas.
Por lo que se refiere al control de la plataforma con el brazo robot, el algoritmo necesario se ejecutará en el Arduino. Como se ha comentado anteriormente, se

tiene la opción de utilizar el entorno IDE de Arduino para llevar a cabo esa programación directamente en la placa. Pero, puesto que uno de los objetivos de este trabajo es la programación de manera sencilla e intuitiva mediante bloques, se ha utilizado Matlab-Simulink junto con la integración de bloques para Arduino para realizar esta tarea.
Por otro lado, para poder interactuar con la plataforma de manera remota, se ha implementado una aplicación Android para la tableta anteriormente comentada usando el entorno de desarrollo Eclipse, con la SDK de Android.

3.1 ARDUINO SUPPORT FOR SIMULINK

Como es bien conocido, Matlab es un software matemático que ofrece un entorno de desarrollo con un lenguaje propio de programación de alto nivel. Además, sobre el entorno de programación de Matlab se encuentra Simulink, siendo una herramienta de extremada utilidad para analizar, modelar y simular sistemas.
Pese a que Matlab ya es de por sí un lenguaje de alto nivel, el uso de Simulink permite programar directamente por bloques y de forma gráfica, pasando a un segundo plano la generación de código. También, debido a la gran cantidad de bloques disponibles en Simulink, la complejidad y funcionalidad de los modelos que se implementan vienen determinados por el nivel de conocimientos del diseñador. Esto hace que sea una herramienta de trabajo verdaderamente interesante en cualquier ámbito académico (desde la Educación Secundaria hasta la Universidad).
En cuanto al uso de bloques de Simulink para la placa Arduino, se ha distribuido recientemente la librería "Arduino support from Simulink", la cual ofrece una serie de bloques diseñados específicamente para la placa en cuestión. De este modo, se puede implementar el esquema deseado en Matlab-Simulink y transferirlo a la placa Arduino de una forma totalmente transparente para el usuario, despreocupándose de la conversión entre bloques de Simulink y código para Arduino.
De entre los bloques para Arduino que se proporcionan, destacan los siguientes (figura 7):
- Arduino Analog Input. Mediante este módulo, se puede leer el voltaje que se esté aplicando por un pin determinado. La precisión de la salida que proporciona este bloque será de 10 bits.
- **Arduino PWM.** A través de este bloque se envía una señal PWM por el pin seleccionado. La frecuencia de la señal cuadrada es fija a 490Hz, pudiéndose modificar el *duty cicle* con una precisión de 8 bits (valores entre 0 y 255).
-Arduino Digital Input/Output. Con estos dos bloques se pueden leer y escribir señales digitales en los puertos seleccionados.
-Arduino Serial Receive/Transmit. Mediante estos bloques se es capaz de enviar y recibir bytes de datos mediante el puerto serie.

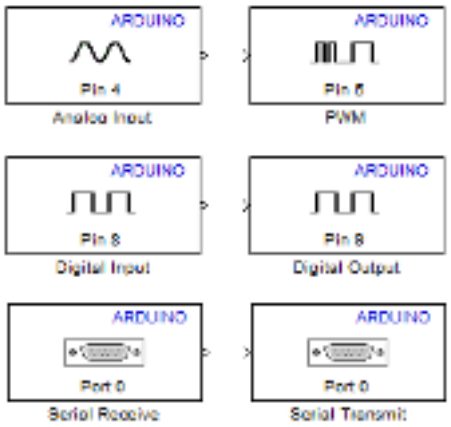


Figura 7: Bloques Arduino Simulink

Como se puede observar, la placa Arduino, junto con la integración para Simulink, permite a los estudiantes entender el proceso de desarrollo de software para sistemas embebidos sin realizar necesariamente una programación manual del algoritmo, sino de una forma tan sencilla e interactiva como es añadiendo, conectando y modificando bloques con una funcionalidad completa.

3.2 PROGRAMACIÓN EN ANDROID

Para poder desarrollar la aplicación de control nativa para dispositivos con el sistema operativo Android es necesario en primer lugar preparar el entorno de desarrollo.
Para facilitar el trabajo a los desarrolladores, Google facilita de manera gratuita y libre un paquete con la última versión del IDE Eclipse junto con el SDK de Android y las herramientas ADT (Android Developer Tools) necesarias. Su instalación es tan sencilla como descomprimir el contenido del paquete descardado de la página de desarrolladores de Android [2] en una carpeta del sistema. Dentro se encuentra el ejecutable del Eclipse el cual lanzará el IDE con todos los componentes necesarios instalados para empezar a desarrollar aplicaciones Android.
La aplicación desarrollada ha sido pensada para ejecutarse en una Tableta de la marca Samsung modelo Galaxy Note 10.1. Para la correcta comunicación del dispositivo con el entorno de desarrollo, es necesaria la previa instalación de la aplicación Kies, software de Samsung que proporciona los drivers y software necesario para la comunicación entre la Tableta y el PC. Una vez instalado el software, el dispositivo es reconocido por el sistema, y desde el IDE Eclipse es posible cargar los programas realizados directamente al dispositivo, simplemente seleccionando como target del proyecto el propio dispositivo.

4 INTEGRACIÓN Y MONTAJE DEL HARDWARE.

Para la construcción de la plataforma móvil robotizada se ha construido una base sólida apoyada sobre dos motores en configuración diferencial y cada uno con una tracción de tipo oruga no orientable como puede observarse en la Figura 8. Todo mediante piezas de LEGO fáciles de ensamblar.



Figura 8.Base de la plataforma montada mediante piezas de LEGO vista desde abajo.

Sobre la base, en el centro, se ha situado el brazo robótico AL5D junto a la controladora SSC-32, de modo que la plataforma queda balanceada con el peso del brazo y de modo que no se despeguen las ruedas del suelo a pesar de cualquier posición que pueda adoptar el manipulador. Al mismo tiempo, a cada lado del brazo robótico se han posicionado dos placas de Arduino MEGA para ofrecer la posibilidad de controlar los movimientos del brazo y el control de motores al mismo tiempo. Esto es necesario debido a que, por el tipo de programación utilizada no es posible realizar operaciones multihilo en el microcontrolador y por tanto, si se desea gestionar procesos concurrentes, es necesario añadir un controlador por proceso.

En este trabajo se ha decidido que un Arduino se encargue de la gestión del posicionamiento del manipulador y el otro, mediante la instalación de la tarjeta de expansión Arduino motor shield, maneje el control de velocidad y de dirección los motores. Se ha instalado también un dispositivo inalámbrico Bluetooth al que están conectados los dos micro-controladores y mediante el cual se creará la conexión entre la plataforma y la Tableta.

Otro elemento fundamental para dotar de autonomía a la plataforma es la alimentación portátil. Se ha instalado una batería LiPo de dos celdas con una capacidad de 5 A. y un voltaje de 7,4 V. para alimentar las dos placas de Arduino con las extensiones de Arduino motor shield y los cinco servomotores que forman parte del brazo robotizado. El esquema de los distintos elementos que forman la plataforma puede observarse en la Figura 9:

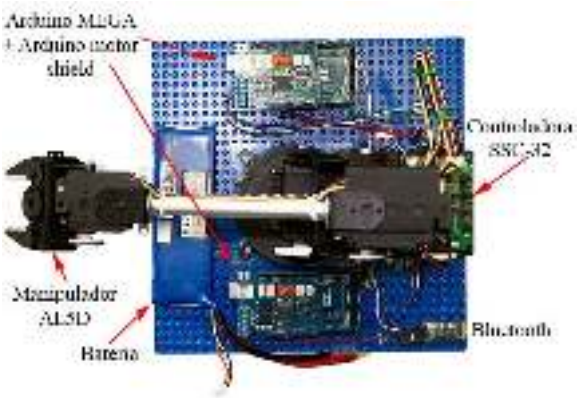


Figura 9: Vista superior de los elementos que forman parte de la plataforma educativa.

Como ya se ha comentado anteriormente, la programación de las tarjetas Arduino se ha realizado exclusivamente con la nueva librería para Simulink con soporte para Arduino. En la siguiente sección se exponen diferentes conceptos relacionados con el campo de la automática, que se han llevado a la práctica mediante la creación de la plataforma y que pueden ayudar a reforzar los métodos de aprendizaje de una manera vistosa y atractiva.

5 PROPUESTA EDUCACIONAL DE LA PLATAFORMA ROBÓTICA

Debido a las características que tiene la plataforma presentada en este trabajo, ésta es una herramienta muy útil y versátil que se puede utilizar para desarrollar una gran variedad de actividades prácticas relacionadas con el aprendizaje de la Automática. A continuación se proponen un conjunto de actividades, aunque por supuesto, dependiendo de los conceptos o contenidos que sean necesarios abordar, éstas se podrían ampliar mucho más.

5.1 ELECTRÓNICA

Para el montaje de la plataforma es necesario conocer bien el hardware y las distintas opciones de interconexión que ofrecen los distintos elementos. De esta forma, por ejemplo, es necesario conocer el voltaje que necesita cada elemento, el consumo de corriente total que puede ocasionar el sistema en el peor de los casos, saber identificar los puertos de entrada y salida, conocer el funcionamiento a bajo nivel de los motores de corriente continua y los servomotores, saber hacer uso correcto de instrumentación de medida como los polímetros y osciloscopios para verificar conexiones y parámetros, etc.

Por lo tanto, se entiende que la plataforma propuesta resulta muy motivadora y útil para los alumnos, puesto que éstos pueden trabajar de una forma

práctica con diferentes cuestiones relacionadas con conceptos teóricos de la electrónica.

5.2 CÁLCULO DE LA CINEMÁTICA INVERSA

El cálculo de la cinemática inversa, es una parte esencial en la robótica relacionada con brazos manipuladores. De esta forma, el brazo robot AL5D que incorpora la plataforma, permite a los alumnos estudiar de una forma práctica la forma de calcular los movimientos necesarios de las articulaciones del robot cuando se desea que la herramienta terminal de éste alcance una posición X-Y-Z.
Para ello se les podría proponer a los alumnos que utilizaran por ejemplo, métodos geométricos o las conocidas matrices de transformación homogénea. La idea es llevar a la práctica las soluciones de este tipo de problemas que se plantean de manera teórica y matemática sobre un papel y comprobar visualmente los resultados sobre una plataforma real.
En la plataforma, la cinemática inversa se ha desarrollado con un bloque *Embedded Function* de Simulink. En dicho bloque es posible verificar de una forma muy simple por ejemplo las dimensiones de los distintos elementos del brazo (base, hombro, codo y herramienta), permitiendo así analizar cómo afecta un cambio de dichas dimensiones en la respuesta del sistema.

5.3 CONTROL DE MOTORES

Los motores son elementos bien conocidos dentro del campo de la Automática y objeto de estudio a fondo en asignaturas relacionadas con el control de procesos, la mecatrónica o la robótica. Mediante la plataforma propuesta es posible realizar una gran variedad de tareas como por ejemplo la identificación de motores, la aplicación de filtros o el control de velocidad y posición basado en la realimentación del valor de los encoders.

5.4 CONTROL CINEMÁTICO

La configuración diferencial definida en la plataforma se considera una de las más sencillas ya que cuenta con dos ruedas de tipo oruga diametralmente opuestas en un eje perpendicular a la dirección del robot. Esta configuración permite realizar giros sobre el propio eje central del robot sin necesidad de avanzar, lo cual es conocido como configuración holonómica.
En una primera etapa se puede proponer a los alumnos que hagan una simulación de la configuración cinemática de este robot, analizando cómo afectan las velocidades de los motores, y por lo tanto su control, en el movimiento de la plataforma.
Una vez comprendido el modelo cinemático se puede proponer a los alumnos tareas relacionadas con la generación de trayectorias de movimiento, y con el control de seguimiento de dichas trayectorias, como por ejemplo los métodos de control de trayectoria por punto descentralizado o el control de camino mediante persecución pura.

5.5 PROGRAMACIÓN MEDIANTE DIAGRAMAS DE BLOQUES

Como ya se ha mencionado anteriormente, se ha realizado una programación mediante bloques de Matlab-Simulink para facilitar la comprensión y el desarrollo de los programas sobre la plataforma que se presenta.
Por supuesto, el nivel de complejidad y el potencial de los mismos dependen únicamente del programador. En este sentido, la Figura 10 muestra por ejemplo un diagrama desarrollado en Matlab-Simulink para la programación de la placa de Arduino encargada del manejo del brazo robótico mediante la recepción de mensajes vía Bluetooth.

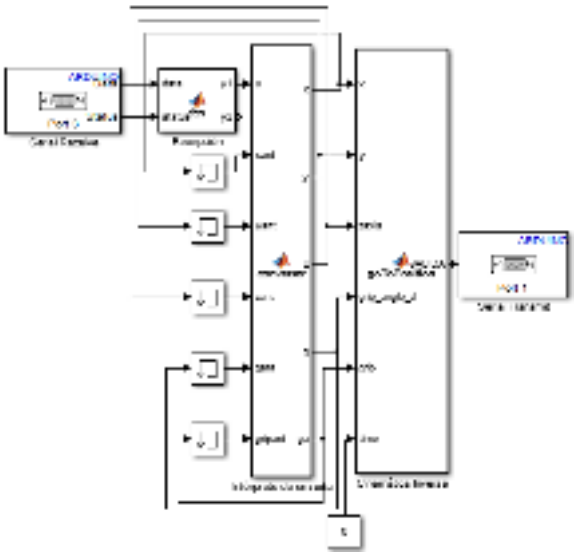


Figura 10: Esquema Simulink para el control del brazo robótico mediante mensajes Bluetooth para Arduino.

En este caso, se puede comprobar como el dispositivo Bluetooth está conectado al puerto serie número 3 de la placa de Arduino. El bloque de *Recepción* realiza una espera activa de la trama de datos y el Bloque del *Intérprete de entrada* es el encargado de desglosar la trama recibida e interpretarla como posiciones x, y, z del brazo robótico donde se desea posicionar, así como la posición de la pinza. Por último, el bloque *Cinemática inversa*, calcula los comandos necesarios para situar al robot en esa posición y se envían por el puerto serie número 1, que es donde se ha conectado la controladora SSC-32.
Para el control de motores mediante comandos vía Bluetooth, la plataforma utiliza una segunda tarjeta Arduino. Para dicho control se ha desarrollado el esquema presentado en la siguiente figura:

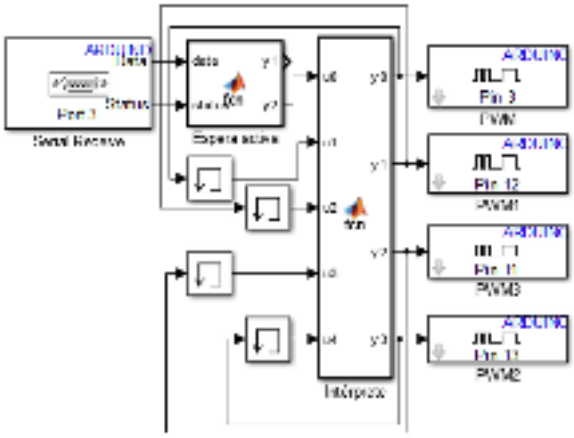


Figura 11: Esquema Matlab-Simulink para Arduino, para el control de motores vía Bluetooth.

Al igual que ocurría con la primera tarjeta, el dispositivo Bluetooth se ha conectado al puerto serie número 3 de Arduino. En este caso el módulo *Intérprete*, traduce la trama recibida a los módulos de salida PWM de Arduino motor Shield. El motor derecho está conectado a los puertos de PWM 3 y 12. El puerto número 3 define la potencia que se desea aplicar al motor y el 12 define el sentido. Del mismo modo, el motor izquierdo está conectado a los pines 11 y 13.

En este sentido, dependiendo del nivel de los alumnos y de los contenidos que se deseen impartir, se pueden proponer diferentes actividades. La más simple sería que los alumnos tuvieran que verificar los distintos módulos y que vieran cómo se puede generar el código y realizar la programación de las tarjetas de control de Arduino.

Por supuesto, se les podrían proponer tareas mucho más complejas, como por ejemplo que fueran ellos los encargados de programar alguna de las funciones embebidas de Matlab para la generación de las señales de PWM de los motores, de las comunicaciones serie, etc.

5.6 DESARROLLO DE UNA APLICACIÓN ANDROID PARA EL CONTROL DE LA PLATAFORMA

Una vez programadas las dos tarjetas de Arduino se podría proponer a los alumnos el desarrollo y la programación de aplicaciones para la interacción vía Bluetooth entre la plataforma robotizada y cualquier dispositivo móvil.

A pesar de que últimamente los dispositivos móviles gozan de gran aceptación y de que hay muchos alumnos interesados en el desarrollo de aplicaciones de dichos dispositivos, su programación no es una tarea simple puesto que, por ejemplo no existe una metodología orientada a la programación mediante organigramas o diagramas de bloques. Por lo tanto, se requiere tener unos conocimientos mínimos de programación mediante código para abordar el desarrollo en este tipo de dispositivos.

No obstante, adquiriendo ciertos conocimientos sobre programación en JAVA, el salto de desarrollar cualquier programa normal en un PC a programar en una tableta no requiere más que conocer los pasos de configuración del entorno de desarrollo.

Para el manejo de la plataforma se ha desarrollado una aplicación en Android a modo de control remoto. La aplicación está basada en un ejemplo proporcionado por Google para la comunicación con dispositivos Bluetooth, en la que un servidor atiende todas las conexiones entrantes por sockets Bluetooth propios del sistema operativo Android.

Como con cualquier dispositivo Bluetooth, lo primero que tendrá que hacer el usuario de la aplicación es dirigirse al menú de opciones de la aplicación, buscar y conectarse al dispositivo Bluetooth receptor de la plataforma. De ésta manera se establecerá la comunicación bidireccional necesaria para el envío de órdenes desde el panel de mandos a la plataforma.



Figura 12: Interfaz de la aplicación Android desarrollada para el control de la plataforma.

Para el control de los motores y del brazo robot de la plataforma móvil, la aplicación dispone de dos mandos de control, izquierdo y derecho, Figura 12. El mando derecho, es el encargado del control de los motores de la plataforma y está compuesto con cuatro botones de dirección intuitivos para el usuario que hacen avanzar, retroceder o girar en los dos sentidos al robot. El mando izquierdo por otra parte, se compone de los controles necesarios para desplazamiento de la herramienta del brazo robot en sus 3 ejes de libertad (x, y, z). Además éste último mando incorpora dos botones para abrir y cerrar la pinza del extremo del brazo, y de ésta manera, poder coger y manipular objetos del entorno.

En este sentido, al igual que en el apartado anterior, dependiendo del nivel de los alumnos, se les podría proponer diferentes actividades, desde la más simple que sería programar una aplicación básica que estableciera simplemente la comunicación Bluetooth entre la tableta y la plataforma robotizada, hasta aplicaciones más avanzadas que permitiera la programación de la trayectoria del robot o el control remoto de éste.



Figura 13: Plataforma manipulando piezas ensamblables.

6 CONCLUSIONES.

Se ha presentado una plataforma robotizada de bajo coste orientada hacia la educación en el campo de la automática en institutos y universidades. El sistema reúne elementos típicos estudiados en problemas de control y automatización como el control de motores, el cableado electrónico, la gestión de puertos serie, la cinemática inversa de un brazo manipulador, la gestión de comunicación inalámbrica…, y es escalable al nivel de dificultad que se desee profundizar.

Se ha detallado el montaje y los conceptos didácticos aplicables a la plataforma y la manera de afrontar su programación mediante organigramas y diagramas de bloques de una manera atractiva e intuitiva para el alumnado.

Se ha incorporado la interacción entre la plataforma y sistemas inalámbricos de uso cotidiano hoy en día como Tabletas o SmartPhones basados en el sistema operativo Android.

En resumen, se ha presentado una plataforma completa de bajo coste basada en parte en software libre con el fin de facilitar la enseñanza de conceptos relacionados con la automática, a través de su desarrollo y programación.

En el siguiente enlace se aloja un vídeo con el montaje y el desarrollo de la plataforma: http://wks.gii.upv.es/cobami/webfm_send/87

Agradecimientos

Los autores desean expresar su agradecimiento al Ministerio de Ciencia e Innovación de España por la financiación parcial de este trabajo bajo los proyectos de investigación DPI2011-28507-C02-01 y DPI2010-20814-C02-02. Además se agradece el soporte financiero por parte de la Universidad de Costa Rica y el apoyo recibido por Instituto U. de Automática e Informática Industrial (ai2) de la U. Politècnica de València con el Plan de Ayudas a la I+D+i 2012.

Referencias

[1] Al-Busaidi, A. M. (2012, November). *Development of an educational environment for online control of a biped robot using MATLAB and Arduino*. Mechatronics 2012. 13th Int'l Workshop on (pp. 337-344).

[2] Android Development Kit. *Developer.android.com*

[3] Busquets, J., Busquets, J. V., Tudela, D., Perez, F., Busquets-Carbonell, J., Barbera, A. y Gilabert, J. (2012). *Low-cost AUV based on Arduino open source*. IEEE/OES (pp. 1-10).

[4] Cooper, S., Dann, W., & Pausch, R. (2000). *Alice: a 3-D tool for introductory programming concepts*. In Journal of Computing Sciences in Colleges (Vol. 15, No. 5, pp. 107-116).

[5] Evans, B. (2012). 3D Printer Toolchain. In *Practical 3D Printers* (pp. 27-47). Apress.

[6] Faludi, R. (2010). *Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing*. O'Reilly Media, Incorporated.

[7] Fisher, D. K., & Gould, P. J. (2012). *Open-source hardware is a low-cost alternative for scientific instrumentation and research*. Modern Instrumentation, 1(2), 8-20.

[8] Google (maintained by Neil Fraser), 2013 "*Blockly - a visual programming editor,"*. Disponible: http://code.google.com/p/blockly/

[9] Henriksen, P., & Kölling, M. (2004, October). *Greenfoot: Combining object visualisation with interaction*. ACM.

[10] Johnson, G. W. (1997). *LabVIEW graphical programming: practical applications in instrumentation and control*. McGraw-Hill School Education Group.

[11] Kadir, W. M. H. W., Samin, R. E., y Ibrahim, B. S. K. (2012). *Internet Controlled Robotic Arm*. Procedia Engineering, 41, 1065-1071..

[12] Kelly, J. F. (2010). *Lego Mindstorms NXT-G Programming Guide*. Apress.

[13] Lim, H., Park, J., Lee, D., y Kim, H. J. (2012). Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *Robotics & Automation Magazine, IEEE*, *19*(3), 33-45.

[14] MathWorks, "Simulink". Disponible en: http://www.mathworks.es/products/simulink/

[15] Piech, C., Sahami, M., Koller, D., Cooper, S., y Blikstein, P. (2012, February). *Modeling how students learn to program*. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 153-160).

[16] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K. y Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67.

[17] Thompson, C. (2008). *Build it. share it. profit. Can open source hardware work*. Wired Magazine, 16(11), 16-11.

[18] Weinberg J. B. y X. Yu, (2003) "*Robotics in education: Low-cost platforms for teaching integrated systems*," IEEE Robotics & Automation Magazine, vol. 10 Issue: 2, pp.4– 6.

Desarrollo de plantas industriales virtuales para el aprendizaje de la automatización

Rodríguez Díaz, Francisco

Moreno Úbeda, José Carlos

Pawlowski, Andrzej

Sánchez Molina, Jorge Antonio

López, Antonio

Ctra. Del Sacramento S/n. ISA, Dpto. Informática Universidad de Almería, frrodrig@ual.es; jcmoreno@ual.es; apawlowski@ual.es; jorgesanchez@ual.es

Resumen

En este trabajo se propone una arquitectura hardware y software para el desarrollo de laboratorios docentes, conectando autómatas programables reales a plantas industriales virtuales. Asimismo, como ejemplo de uso de la arquitectura, se ha desarrollado un laboratorio basado en una planta embotelladora virtual y en los autómatas Modicon M340 de Schneider®.

Palabras Clave: automatización industrial, autómatas programables, laboratorios virtuales, comunicaciones industriales.

1 INTRODUCCIÓN

En la industria actual, la mayor parte de los procesos de fabricación se encuentran automatizados. Generalmente, en los procesos de fabricación no basta que el sistema controle determinadas variables, sino que además se pretende que se establezcan secuencias ordenadas de actuaciones capaces de llevar al proceso, de forma automática, desde un estado de funcionamiento a otro, interesando situaciones como que una válvula de alimentación de un determinado producto se encuentre abierta o cerrada, que un robot realice un determinado programa de instrucciones cuando se detecte la presencia de una pieza en una zona concreta, que se active o se desactive un calefactor, etc.

Para automatizar estos procesos, el sistema debe disponer de unos sensores para detectar eventos o situaciones especiales y unos actuadores para realizar las acciones que tengan que ejecutarse en cada instante, además del controlador correspondiente que gestione todos estos subsistemas.

Los controladores más empleados en este tipo de procesos son los autómatas programables o PLC's (Programmable Logic Controller) que son sistemas electrónicos basados en microprocesador o microcontrolador, con una configuración modular, que pueden programarse para controlar, en tiempo real y en ambiente industrial, procesos que presenten una evolución secuencial.

En el ámbito de casi todos los estudios de ingeniería se estudian este tipo de sistemas y cómo se controlan con autómatas programables, de forma que los alumnos diseñan el sistema de control y lo implementan en los PLC's de los que disponga cada Universidad. El problema radica en que normalmente no se dispone de plantas industriales reales sobre las que probar su funcionamiento ya que son muy costosas y, en caso de disponer de alguna, sólo se podrían realizar prácticas con ese único sistema industrial. Lo que se suele hacer para realizar prácticas en este campo es plantear diversos ejemplos sobre el papel y mediante entrenadores simular el proceso industrial. El gran inconveniente de este planteamiento es que el alumno no tiene una percepción real de una verdadera planta industrial, ya que el propio alumno debe simular el comportamiento real del sistema activando los interruptores correspondientes en el entrenador.

En este trabajo se propone una arquitectura hardware y software (basada en *Labview®*, OPC, *MODBUS*, autómatas *Schneider® Modicon M340* y *SolidWorks®*) que permite diseñar laboratorios docentes en el ámbito de la automatización industrial ([1]), interconectando un PLC real a una planta industrial virtual que irá proporcionando al autómata información de los sensores, y que irá modificando su comportamiento en función del programa que se esté ejecutando en el PLC y de la propia dinámica del proceso. En [3] se desarrolla la misma idea para una planta virtual con tanques de fluidos, haciendo uso de

un PLC S7 300 de *Siemens®, CodeSys®* para programar el SCADA (Supervisory Control And Data Adquisition), PROFIBUS para la comunicación con el PLC y WinMod combinado con Flux Studio para obtener el modelo dinámico 3D de la planta. En [2] se puede encontrar una revisión de una clase especial de plantas industriales virtuales, consistentes en una única máquina herramienta (Virtual Machine Tools), donde se realiza un mecanizado virtual (Virtual Machining) controlado por un PLC. En [4] se presentan dos alternativas para la implementación de estos laboratorios virtuales, una basada en la utilización de una aplicación comercial que permite trabajar con cinco plantas virtuales prediseñadas y otra basada en el conjunto de herramientas de desarrollo de juegos para la videoconsola *XBOX 360* de Microsoft®, permitiendo la definición de plantas completamente concebidas por el usuario. En ambos casos con conexión a PLC, permitiendo también la conexión a un simulador con el fin de facilitar la realización de las prácticas por parte del alumno fuera del laboratorio físico.

En general, la implementación de la interfaz gráfica de las plantas virtuales podría realizarse directamente con herramientas clásicas de desarrollo de SCADAs, sin embargo el objetivo es obtener modelos 3D lo más realistas posible, lo que aconseja hacer uso de herramientas CAD, como por ejemplo *SolidWorks®*, *AutoCAD o XNA Game Studio*, centradas en el desarrollo de escenarios complejos.

El trabajo se ha estructurado en cinco secciones. En la primera se ha presentado una visión general del trabajo, relacionándolo con otros trabajos presentes en la literatura. En la segunda sección se establece el marco docente en el que se encuadra la arquitectura propuesta. En la sección 3 se detalla la estructura de la herramienta, en la 4 se presenta un ejemplo basado en una planta de embotellado, y finalmente, en la sección 5 se presentan las conclusiones y trabajos futuros.

2 MARCO DOCENTE

Actualmente en la Universidad de Almería, se oferta una formación de automatización de procesos industriales en las siguientes cinco titulaciones de grado (planes de 2010):

- Grado en Ingeniería Electrónica Industrial
- Grado en Ingeniería Mecánica
- Grado en Ingeniería Química Industrial
- Grado en Ingeniería Informática
- Grado en Ingeniería Agrícola

En todas ellas, se incluye una Unidad Didáctica dedicada al modelado y control de procesos secuenciales de unos 2 ECTS (15 horas presenciales y 35 horas de trabajo autónomo del alumno), compuesta por dos temas teóricos y una práctica de laboratorio.

El primero de los temas de teoría se centra en el análisis y modelado de los sistemas secuenciales industriales, cuyos objetivos de aprendizaje son:

- Exponer los tipos de procesos que se pueden encontrar en la industria, destacando las diferencias entre los procesos continuos frente a los secuenciales y por lotes en los que es necesario seguir una secuencia de pasos.
- Mostrar los sensores y actuadores que se suelen utilizar en este tipo de sistemas.
- Explicar el concepto formal de Autómata
- Mostrar las distintas técnicas tabulares de modelado de sistemas secuenciales.
- Mostrar las diferentes técnicas gráficas de modelado de sistemas secuenciales, destacando las Redes de Petri por su simplicidad y por facilitar el proceso de programación de controladores de sistemas secuenciales.

El siguiente tema de teoría está dedicado a la implementación de los sistemas de control secuencial con los siguientes objetivos de aprendizaje:

- Mostrar las características principales de los controladores secuenciales y sus tipos, destacando los autómatas programables como controlador básico y más utilizado en este tipo de procesos, frente a la lógica cableada.
- Describir las estructura física externa modular de un autómata programable.
- Mostrar la arquitectura interna y los principios de funcionamiento de un autómata programable.
- Describir la Norma IEC1131-3 donde se muestran los objetos, bloques y diferentes lenguajes de programación de los PLC's.

Como complemento a esta actividad teórica, se propone al alumno la realización de una práctica de laboratorio utilizando autómatas programables *M340* de *Schneider Electric* [5], compuestos por un bastidor de cuatro celdas, la fuente de alimentación CPS 2000, CPU BMX P34 2020, un módulo DDM 16022 con 8 entradas y 8 salidas digitales y un módulo AMM 0600 con 4 entradas y 2 salidas analógicas. Como se puede observar en la figura 1, el puesto de laboratorio dispone además de un entrenador que permite activar/desactivar las entradas del autómata y de un computador personal con sistema operativo Windows 7, en el que se encuentra instalada la herramienta *Schneider Unity Pro* [5] que se utiliza para configurar el autómata, programarlo, transferir el programa y ejecutarlo.



Figura 1. Puesto de laboratorio

En la práctica de esta Unidad Didáctica se revisan conceptos básicos de diseño de sistemas de control procesos secuenciales utilizando autómatas programables, siendo sus objetivos específicos los siguientes:

- Mostrar la estructura física modular de un autómata programable real, así como sus principios de funcionamiento.
- Describir la gestión de entradas y salidas del autómata, así como los fundamentos para la utilización de temporizadores.
- Exponer los pasos que hay que seguir para realizar una aplicación con autómatas: configuración, simbolización de las variables, programación, transferencia del programa al autómata, comprobación del funcionamiento y documentación.
- Describir la programación del algoritmo que soluciona el problema mediante autómatas, utilizando los lenguajes de *Lista de Instrucciones*, *SFC* (Grafcet) y *Diagrama de Bloques Funcionales*, de la Norma IEC-1131.

Para ello, se propone a cada grupo de trabajo, compuesto por dos alumnos, un proyecto de automatización industrial basado, en la mayor parte de los casos, en los supuestos prácticos presentados en [4], modificados convenientemente para que se puedan implementar con el autómata programable disponible en la laboratorio, y para que se trabajen los aspectos fundamentales de este tipo de procesos, como diferentes tipos de entradas y salidas (digitales y analógicas), y uso de temporizadores y contadores. Cada grupo debe realizar las siguientes actividades:

1. Analizar el sistema e indicar todas las variables que lo definen (entradas, salidas y auxiliares).
2. Indicar todas las hipótesis que se han considerado para modelar el proceso. En caso de que se haya realizado alguna modificación al sistema propuesto para mejorar su comportamiento o comprensión, se deben justificar esas decisiones.
3. Modelar el comportamiento del proceso asignado mediante una Red de Petri, justificando la misma.
4. Indicar el cuadro de asignaciones de entradas, salidas y variables auxiliares.
5. Implementar el controlador del proceso industrial asignado, en el autómata programable con los lenguajes de la norma IEC-1131, *Lista de Instrucciones, SFC (Grafcet) y Diagrama de Bloques Funcionales.*

Al finalizar esta Unidad Didáctica, el alumno deberá ser capaz de realizar los pasos básicos necesarios para el diseño completo de un proyecto de automatización de un determinado proceso (modelado e implementación), aplicándolo a sistemas relacionados con el sector industrial de sus titulaciones.

3 ARQUITECTURA DE LA HERRAMIENTA

Como se comentó anteriormente la disponibilidad de procesos industriales posibilita que los estudiantes completen los conocimientos teóricos durante el desarrollo de la asignatura en forma empírica. La principal limitación en este aspecto es la escasa variedad de plantas industriales que pueden ser analizadas y estudiadas. Para hacer frente a este problema, en los últimos años se ha realizado un importante esfuerzo en el desarrollo de procesos y plantas virtuales con fines educativos. Los resultados obtenidos han sido muy satisfactorios, especialmente en áreas de ingeniería donde se destaca la necesidad de realizar los estudios con diferentes grados de abstracción, algo que puede ser fácilmente ajustado gracias a la flexibilidad de la implementación de los procesos virtuales.

La idea de este trabajo es proponer una arquitectura hardware y software que permita, a largo plazo, crear un conjunto de plantas virtuales para la enseñanza de la automatización industrial. La figura 2 muestra la arquitectura propuesta, donde se pueden distinguir los siguientes módulos: visualización y supervisión del proceso, gestión de variables, comunicación y control del proceso. El objetivo del primer módulo es la visualización del estado del proceso basándose en las variables que lo caracterizan. En este trabajo se utiliza el entorno *SolidWorks®*, que permite incorporar elementos 3D con el objetivo de formar componentes y conjuntos de componentes para reflejar las piezas y equipos utilizados en el proceso real equivalente. Este software ha sido seleccionado por su gran flexibilidad de diseño de componentes, y por permitir de forma sencilla la animación de los mismos. De esta forma *SolidWorks®* constituye la

interfaz gráfica del proceso virtual. Para ello es necesario utilizar dos módulos específicos. El primer módulo, *SolidWorks® Simulations*, permite a *SolidWorks®* conectarse al entorno de simulación del proceso y obtener en tiempo real el estado de la planta según su dinámica. Esta información de estado es utilizada por el segundo modulo, *SolidWorks® Motion*, para generar movimientos, transiciones u otros valores que se modifican en la representación gráfica del proceso virtual.
Por otro lado, como es habitual en el ámbito industrial, y dentro del módulo de visualización y supervisión del proceso, se ha considerado la posibilidad de implementar un sistema SCADA (en *LabVIEW®*), con la idea de utilizar esta arquitectura no sólo en prácticas relacionadas con la programación de PLC's, sino también en aquellas relacionadas con el desarrollo de sistemas SCADA.

Por tanto, va a ser necesario definir un conjunto de variables compartidas para monitorizar la planta virtual. *LabVIEW®* se encarga de la gestión de estas variables y de la simulación de la dinámica de los procesos continuos y/o discretos implicados en la planta virtual. Esta parte forma el nodo central de la arquitectura propuesta. Su principal tarea consiste en actualizar el estado de la planta virtual en función de las entradas proporcionadas por la aplicación desarrollada e implementada físicamente en el PLC. De esta forma, *LabVIEW®* se comunica mediante la estructura de datos compartida con *SolidWorks®* para actualizar el estado del proceso virtual.
Por otro lado, *LabVIEW®* se comunica con el autómata para comprobar su estado actual, actualizando en función del mismo el estado del modelo que caracteriza al proceso. Para llevar a cabo estas tareas es necesario utilizar los módulos de National Instruments® para *LabVIEW®*: *Control Design and Simulation, Datalogging and Supervisory Control*, *Real-Time* y *SoftMotion*. La arquitectura propuesta en este trabajo considera dos opciones de implementación para la parte de comunicación entre el autómata y el proceso virtual. Se optó por estas dos posibles vías de desarrollo teniendo en cuenta la diversificación de comunicaciones en los procesos industriales. En la primera configuración, opción I en la figura 2, la comunicación entre el autómata y el proceso virtual se realiza mediante el servidor OPC OFS (OPC Factory Server) de Schneider®. Esta opción proporciona una fácil y rápida configuración de la comunicación entre la parte hardware y software de la arquitectura. En este caso el servidor OPC tiene que estar dentro de la misma red que el autómata y el computador (donde se ejecuta *LabVIEW®*) o directamente ejecutándose en dicho computador. La segunda configuración, opción II en la figura 2, se realiza utilizando el protocolo *MODBUS* implementado sobre TCP/IP. En este esquema se utiliza un estructura maestro-esclavo, donde la parte del maestro está implementada en el autómata. Para realizar esta tarea deben definirse los lugares de memoria que pueden ser accedidos por el nodo de esclavo. En este caso la tarea del esclavo se implementa en el computador (nodo central de la arquitectura propuesta). Para ello se hace uso del módulo *Datalogging and Supervisory Control*, anteriormente mencionado, que tiene incorporadas la rutinas para el protocolo *MODBUS*. En general, este esquema de comunicación permite interconectar un mayor número de nodos, lo que puede ser muy útil en la implementación de procesos distribuidos compuestos por subprocesos más simples controlados de forma remota.

La última parte de la arquitectura propuesta está formada por un PLC donde se implementa el programa que realiza las tareas de control. Las acciones de control se ejecutan en función del estado del proceso virtual que será proporcionado por el nodo central, responsable de gestionar todas las variables representando las entradas y salidas del proceso. Para ello se establecen unas direcciones de memoria del autómata donde se mapean las variables asociadas a los sensores que representan el estado del proceso en cada instante de tiempo. Estas variables son actualizadas por el nodo esclavo de MODBUS (aplicación *LabVIEW®*). Una vez procesadas las entradas se obtiene un conjunto de salidas que se guarda en el nodo maestro de MODBUS (memoria PLC) para compartirlas con el gestor de variables (aplicación *LabVIEW®*) que actualizará el estado del proceso virtual.

De esta forma se obtiene una arquitectura que permite automatizar el funcionamiento de una planta industrial virtual. La arquitectura desarrollada permite visualizar los resultados de control en una forma dinámica y realista, a diferencia de otras arquitecturas (por ejemplo basadas en aplicaciones web aplicando Java). Utilizando esta arquitectura se pueden realizar las tareas de diseño e implementación de forma similar a como se harían para un proceso real. La gran ventaja de los procesos virtuales es el bajo coste de desarrollo en comparación con el de un sistema real. Por otro lado no se requiere ningún tipo de mantenimiento, y permite disponer de una amplia variedad de plantas virtuales que pueden ser orientadas a los aspectos específicos de cada asignatura.

A continuación se muestra como ejemplo una planta embotelladora aplicando la arquitectura propuesta en esta sección.



Figura 2. Arquitectura propuesta para la herramienta de desarrollo de plantas virtuales para el aprendizaje de la automatización.

4 RESULTADOS – UNA PLANTA EMBOTELLADORA

En este apartado se muestra como ejemplo una planta industrial para el llenado de botellas (figura 3), basada en los trabajos del profesor Fernando Castaño de la Universidad de Sevilla (asignatura Informática Industrial).

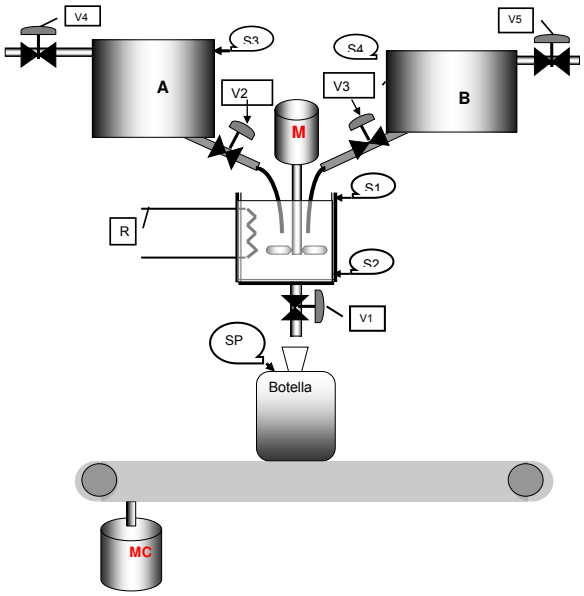


Figura 3. Esquema de la planta embotelladora.

La planta tiene dos depósitos grandes de acumulación (A y B) de dos líquidos distintos. Una vez accionado el interruptor I y gracias a la acción de dos válvulas, una para A (V2) y otra para B (V3), ambos depósitos vierten sobre otro más pequeño al que se le ha instalado un agitador (M) y un calefactor (R). Este depósito tiene capacidad para el llenado de una botella. El llenado se produce hasta que el líquido vertido llega al nivel del sensor S1, que es el encargado de marcar el límite máximo de llenado. En este depósito pequeño los líquidos se mezclan (M) durante un tiempo de 5 segundos (temporizador TM1), luego se calienta la mezcla (R) durante 10 segundos (temporizador TM2), y por último se vierte dentro de una botella hasta que el sensor de nivel S2 se activa advirtiéndose el final de la operación de vaciado y el cerrado de la válvula V1. Las botellas son transportadas por una cinta transportadora (MC) hasta el punto de llenado (SP). Una vez se hayan procesado 4 botellas (contador C1), se procede al llenado de los depósitos grandes (los sensores de nivel S3 y S4 permiten detectar el umbral de llenado máximo).

Una vez estudiado el proceso y reconocidas todas las entradas y salidas del mismo, el alumno debe realizar la Red de Petri correspondiente (figura 4).

El siguiente paso es la programación del autómata, para ello se utiliza la herramienta *Schneider Unity Pro*® que va a permitir a los alumnos trabajar en modo simulación o directamente transfiriendo al autómata el algoritmo elaborado en el transcurso de la práctica. El programa de control de la planta de embotellado se realizará en lenguaje Grafcet y al menos en otro de los lenguajes disponibles en la herramienta: contactos, lista de instrucciones o literal estructurado. Una vez implementado el programa, se procederá a la ejecución. Mediante el diseño 3D de la planta completa realizado en *SolidWorks*® (figura 5), los alumnos podrán comprobar que el funcionamiento de la misma bajo el control del PLC es el correcto.

5 CONCLUSIONES

En este trabajo se ha presentado una arquitectura hardware-software con la que se pretende que el alumno trabaje con autómatas en una simulación de planta industrial real. Este tipo de plantas con muy útiles para asignaturas relacionadas con la automatización industrial, que permite modificar el comportamiento dinámico (continuo/discreto) de una planta industrial virtual mediante un autómata

programable, sin tener que buscar plantas reales que no siempre están disponibles para el ensayo en prácticas y resultan demasiado costosas para una institución docente . Como ejemplo se muestra una planta embotelladora y como trabajo futuro se propone la generación de una batería de plantas de características acordes a la naturaleza de las asignaturas en cuestión, relacionadas con la automatización de máquinas de mecanizado, procesos químicos y plantas industriales en general.

Por otra parte, indicar que para el curso 2013/2014 está previsto probar dos plantas en la asignatura Automatización Industrial, troncal de los estudios de ingeniería industrial, que cuenta con unos 180 alumnos, por lo que se podrá realizar un estudio sobre la respuesta del alumnado ante este tipo de herramientas.

Agradecimientos
Este trabajo se enmarca dentro del Proyecto 12-14-2-09 "Desarrollo de plantas industriales virtuales para el aprendizaje interactivo de la automatización", financiado por la Convocatoria Bienal de Grupos docentes de Innovación para la Creación de Materiales Didácticos en Soporte Informática y Uso de las TICs en la Docencia en la Universidad de Almería (Cursos 2012-13/2013-14)

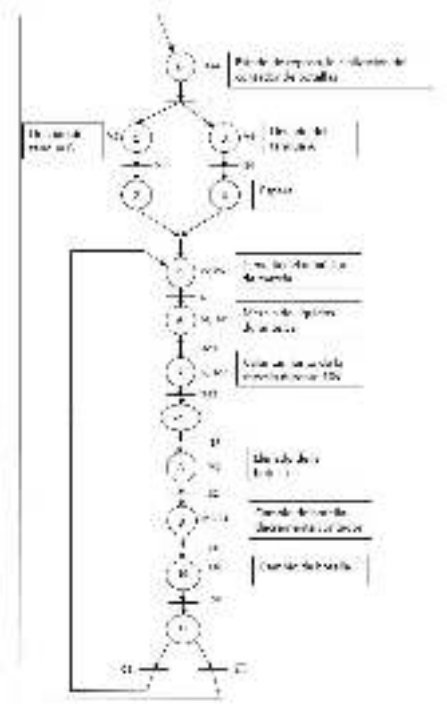


Figura 4. Red de Petri de la planta embotelladora.



Figura 5. Simulación del proceso.

Referencias

[1] Andújar, J. M., Mateo, T. J. (2010) "Diseño de laboratorios virtuales y/o remotos. Un caso práctico", *Revista Iberoamericana de Automática e Instrumentación Industrial*, vol. 7(1), pp. 64-72.

[2] Kadir, A. A., Xun X., Hämmerle E. (2011) "Virtual machine tools and virtual machining – A technical review", *Robotics and Computer-Integrated Manufacturing*, pp. 494-508.

[3] González, F., (2010) Diseño de un sistema virtual de tanques de fluidos (PFC), Escuela Superior de Ingenieros de Sevilla, Universidad de Sevilla.

[4] Sánchez del Pozo, A. J., Escaño, J. M, Muñoz De la Peña, D., Gómez-Estern, F. (2013) "Docencia y evaluación automática mediante simuladores industriales 3D", *Encuentro sobre innovación docente en Automática,* Toledo.

[5] Schneider Electric, Autómatas Programables Modicon M340, (consultada en junio de 2013), http://www.schneiderelectric.es/sites/spain/es/productosservicios/formacion/equipos_didacticos/ed_industry_automatas.page

Herramienta basada en SCORM para la integración automática de Laboratorios Online en LMS

Ildefonso Ruano Ruano, Juan Gómez Ortega, Javier Gámez García, Elisabet Estévez Estévez
Grupo de Robótica, Automática y Visión por Computador, Universidad de Jaén, Jaén, España
{alonso, juango, jggarcia, eestevez}@ujaen.es

Resumen

Los laboratorios online se utilizan como valiosos recursos de apoyo a la docencia desde hace bastantes años. Como recursos docentes con gran potencial podrían enmarcarse dentro de lo que se denomina recurso de e-learning. Una de las últimas tendencias en el desarrollo de estos laboratorios ha sido su integración en los llamados Sistemas de Gestión de Aprendizaje (LMS). En este trabajo se presentan una propuesta basada en el conjunto de normas SCORM- Sharable Content Object Reference Model, estándar de facto de contenidos e-learning que facilita a los desarrolladores Java de laboratorios online la integración con los LMS. Para ello se muestra un ejemplo de laboratorio virtual creado con Easy Java Simulations (EJS) en el que se utiliza esta metodología logrando una integración entre el software del laboratorio virtual y el LMS a través de SCORM.

Palabras Clave: SCORM, Sistemas de Gestión de Aprendizaje, Laboratorios Online.

1 INTRODUCCIÓN

El trabajo de laboratorio es indispensable e indiscutible en la educación universitaria de la mayoría de asignaturas directamente relacionadas con la ingeniería [6]. Los laboratorios remotos, o virtuales, presentan múltiples y variadas ventajas. Entre éstas se puede destacar el ahorro de costes, extender el uso de recursos escasos, compartir equipamiento con otros organismos, seguridad de las personas y ampliar la disponibilidad de recursos en el tiempo y el espacio.

Los laboratorios virtuales/remotos pueden entenderse como una forma de e-learning. Los primeros prototipos de laboratorios remotos para educación fueron realizados a medida con protocolos y servidores también a medida [3], pronto se vió la posibilidad de utilizar la web para dar soporte a los experimentos y surgieron sistemas propios para el desarrollo de laboratorios [9]. Estos sistemas estaban soportados por portales y arquitecturas específicas [4] [8]. El e-learning ha evolucionado desarrollando una serie de herramientas y estándares que facilitan la interoperabilidad de contenidos. A su vez, dentro del e-learning los Sistemas de Gestión de Aprendizaje (LMS, *Learning Management System*) constituyen una herramienta básica común a los alumnos, contienen la mayoría, sino todos los contenidos y herramientas docentes, por eso la evolución lógica de la forma de implementar los laboratorios virtuales y remotos hizo que se tratarán de integrar en los LMS, hecho que se ha realizado en algunas ocasiones de forma básica [13].

Se ha generalizado el uso de LMS como la herramienta básica integradora de recursos de e-learning y SCORM (*Sharable Content Object Reference Model*) como el conjunto de normas y especificaciones de e-learning que permiten el intercambio de contenidos entre LMS, la última versión de SCORM es la cuarta edición de SCORM 2004, lanzada en 2009 [1]. SCORM fue creado para que los desarrolladores pudieran generar contenido reusable e interoperable entre múltiples LMSs. Proporciona las pautas para lanzar y manejar contenidos, un mecanismo común para que los contenidos se comuniquen con el LMS y un lenguaje/vocabulario predefinido que forma la base de esa comunicación.

La integración de laboratorios online y LMS ofrece bastantes ventajas tanto para los desarrolladores de los mismos como para el alumnado. Entre ellas destacan principalmente la posibilidad de que el LMS realice el control de acceso de los usuarios y que se presenten todos los recursos docentes en un mismo entorno.

Las primeras aproximaciones de los laboratorios a los LMSs trataban de usar los modelos de laboratorios ya desarrollados, empleando tecnologías y modelos ya experimentados. Sin embargo, de esta forma se deja de lado al estándar de facto de contenidos e-learning, SCORM, para realizar la integración se desarrollan módulos específicos y extensiones del LMS que atienden a las características particulares del laboratorio en cuestión [7]. De este modo, se pueden conseguir resultados muy prometedores y potentes [14] aunque difícilmente reutilizables ya que no

permiten su uso en laboratorios desarrollados con otras tecnologías ni en un LMS distinto al que se utilizó en principio.

Recientemente se ha desarrollado un sistema que permite que los modelos de laboratorios se integren en los LMS utilizando SCORM. Para ello, se ofrece un portal que actúa como repositorio de laboratorios que se crean mediante la incrustación del software del laboratorio en un formato SCORM [10]. Esta forma de trabajar hace que no existan requerimientos para la tecnología de desarrollo del laboratorio; cualquier tecnología sería válida. Sin embargo, este sistema exige una complicación en las comunicaciones realizadas a la hora de realizar un experimento (en el caso de laboratorios remotos se ven envueltos el sistema servidor remoto de la planta real, el portal específico, el LMS y el PC del usuario que realiza el experimento) y, sobre todo, no ofrece posibilidades de interacción avanzadas entre el software del laboratorio y el LMS. De este modo no se aprovechan las posibilidades de comunicación laboratorio-LMS que se podrían obtener.

En este trabajo se presenta una herramienta para que los desarrolladores de laboratorios virtuales/remotos consigan que su software interactúe de forma completa con un LMS con todas las posibilidades que ofrece SCORM. Para ello se debe utilizar una implementación del interfaz para la tecnología utilizada en el desarrollo del laboratorio, en este caso Java. El trabajo mostrado en esta comunicación realiza un encapsulado de las funciones que ofrece el LMS a los contenidos SCORM (mediante una instancia del *Aplication Program Interface*, API) que permiten la manipulación del modelo de datos del *Run-Time Environment* (RTE) de SCORM [1]. Después será necesario que el software de laboratorio, una vez desarrollado incorporando esta herramienta, se incluya como *Applet* en un formato SCORM. De este modo, las particularidades de las comunicaciones SCORM-LMS son transparentes a los desarrolladores.

A modo de ejemplo se ha desarrollado un laboratorio virtual con la herramienta *Easy Java Simulation*, EJS [5], que utiliza el lenguaje de programación Java para comunicarse con el LMS. Este ejemplo se genera desde EJS como un *Applet* Java que se incluye dentro de un paquete SCORM que puede importarse como contenido SCORM en un LMS compatible [2]. El laboratorio virtual se ha diseñado para asignaturas de grados de la rama de Ingeniería Industrial en las que se está utilizando en módulos de identificación y caracterización de sistemas dinámicos.

El resto del artículo está estructurado de la siguiente forma: en el apartado 2 se explican las características principales de los estándares SCORM y sus 3 componentes, haciendo un mayor hincapié en el RTE y su principio de funcionamiento básico. En el apartado 3 se describe la integración de los laboratorios con los LMS mediante SCORM. En el apartado 4 se muestra un ejemplo de laboratorio virtual desarrollado en EJS e integrado en un módulo SCORM que interactúa con ILIAS, el LMS utilizado como plataforma de docencia virtual en la Universidad de Jaén (UJA), mediante las funciones descritas en el apartado 3. Para finalizar el apartado 5 muestra las conclusiones del trabajo y marca las líneas futuras del mismo.

2 SCORM COMO SOPORTE DE LABORATORIOS ONLINE

Los estándares y especificaciones SCORM constituyen el estándar de facto de docencia virtual. El conjunto de productos que soportan SCORM incluye la mayoría de los LMS del mercado [2]. están documentados y mantenidos por el *Advanced Distributed Learning Iniciative* (ADL) de los Estados Unidos de América, aunque derivan de diversos trabajos realizados por varias organizaciones tecnológicas y de la industria entre las que se encuentra el *IMS Global Learning Consortium* (IMC), el *Aviation Industry CBT Commitee* (AICC) y el *Electrical and Electronics Engineers Learning Technology Standards Commitee* (IEEE LTSC). Existen varias versiones de SCORM (1.0, 1.2, y 2004), utilizando la última de ellas: la edición 4ª de SCORM 2004. SCORM ofrece un marco que se aplica a los contenidos de e-learning en un LMS con el fin de definir su encapsulado, lanzamiento e intercambio de datos. Para ello se definen 3 subespecificaciones:

- Sección de empaquetado de contenidos (*Content Packaging*) [1]. Especifica la forma de describir y empaquetar los contenidos. Está basado en XML (*eXtensible Markup Language*) principalmente.
- Sección de secuenciación (Sequencing) [1]. Especifica la forma en la que el usuario puede navegar entre las distintas partes del contenido SCORM (SCOs o *Sharable Content Object*). Se define como un conjunto de reglas y atributos en formato XML.
- Sección de tiempo de ejecución (RTE) [1]. Especifica la forma en que debe lanzarse el contenido y como se realiza la comunicación con el LMS. Basado en ECMAScript (JavaScript) principalmente.

Para realizar este trabajo se han utilizado las tres especificaciones, aunque el principal objetivo del mismo hace que la subespecificación RTE sea la que se ha tratado de forma más intensa. El RTE requiere unas condiciones para que los objetos de contenido

SCORM puedan lanzarse y gestionar la información del progreso del alumno que puede intercambiarse con el LMS.

En el contexto SCORM existen dos tipos de objetos de contenido: *SCOs* y *Assets*. Los *Assets* no tienen capacidad de comunicarse con un LMS, SCORM sólo requiere que un LMS lance los *Assets* a los alumnos mediante el protocolo HTTP. Un *SCO* es la unidad de información lógica más pequeña que un LMS puede entregar a los alumnos desde la propia plataforma y que puede comunicarse con el LMS. También se puede definir un *SCO* como el único componente de SCORM que usa el API de SCORM, que es el mecanismo de comunicación que existe para obtener y almacenar datos entre el LMS y el *SCO* (p.e. puntuación alcanzada, límites de tiempo, objetivos, etc.) y para informar sobre el estado de comunicación conceptual entre el *SCO* y el LMS (p.e. inicializado, terminado y/o condiciones de error). La información que se pueden intercambiar un *SCO* y un LMS viene definida por el modelo de datos preestablecido y estandarizado. Este modelo de datos se usa para definir elementos del modelo de datos que el *SCO* y el LMS deben conocer y deben ser mantenidos por el LMS de la misma forma independientemente del LMS utilizado.

Cuando un LMS lanza un SCORM debe hacerlo en una ventana de navegador web dependiente de la ventana del LMS que exponga la instancia del API como un objeto del Modelo de Objetos del Documento (DOM). La instancia del API la debe proporcionar el LMS. Un LMS sólo puede lanzar y registrar un *SCO* a la vez (por alumno). El *SCO* debe buscar recursivamente en la jerarquía de ventanas padre hasta encontrar la instancia del API para, una vez conseguido, pueda iniciar una sesión de comunicación con el LMS.

La figura 1 ayuda a entender el modelo temporal del RTE de SCORM.

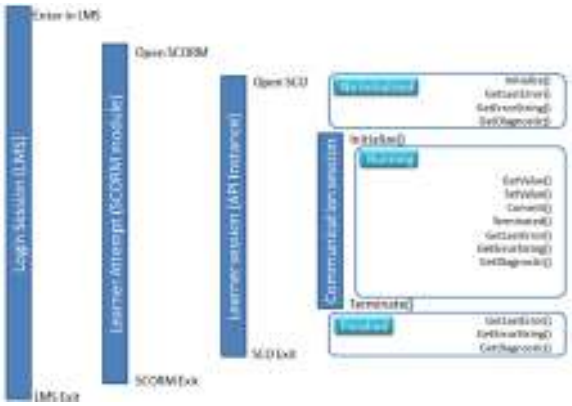


Figura 1. Modelo temporal del RTE.

En la figura 1 se muestra el significado temporal de una sesión de login en el LMS, de un intento del alumno, de una sesión de alumno y de una sesión de comunicación del *SCO* con el LMS. También se muestran las funciones del API que pueden ejecutarse dependiendo del estado en el que se encuentre el SCO.

El API de SCORM consiste básicamente en la definición de sólo 8 funciones que pueden ser utilizadas por un SCO. Estas funciones se pueden estructurar en 3 categorías:

- Funciones de sesión (Initialize() y Terminate()). Se usan para establecer el inicio y el final de una sesión de comunicación.
- Funciones de transferencia de datos (GetValue(), SetValue() y Commit()). Se usan para intercambiar datos del modelo RTE de SCORM entre el LMS y el SCO.
- Funciones de soporte (GestLastError(), GetErrorString() y GetDiagnostic()). Se usan para comunicaciones de soporte entre el *SCO* y el LMS.

Una vez que un LMS lanza un *SCO* puede iniciar una sesión de comunicación con el LMS para almacenar y obtener información del modelo de datos del RTE. Esto lo logra realizando llamadas a las funciones JavaScript que contiene la instancia del SCORM API que facilita el LMS en el que está ubicado el SCO. Aunque el proceso pueda parecer sencillo al existir sólo 8 funciones se puede complicar bastante ya que el modelo de datos RTE de SCORM tiene un número bastante significativo de datos y subconjunto de elementos en muchos de ellos.

3 DESCRIPCIÓN DE LA DE INTEGRACIÓN LAB-LMS CON SCORM

El objetivo de este trabajo es facilitar la integración de los laboratorios online, ya sea virtual o remoto, con los LMS. Para ello se ha tratado de buscar una solución que sea lo más universal y extrapolable posible, por eso se trató de utilizar los estándares SCORM como soporte del SW soporte de los laboratorios.

Las preguntas que hay que responder ahora son ¿cómo incluir un laboratorio online en un SCORM?, ¿cómo lograr la comunicación entre el laboratorio y el LMS?

La respuesta a la primera pregunta es sencilla, SCORM es un formato abierto que se ofrece en forma de página HTML, esto hace que sea totalmente compatible con la mayor parte de los laboratorios

remotos/virtuales, los cuales se suelen mostrar a los alumnos empotrados en páginas web. Este hecho ha sido realizado con éxito en otros trabajos [10], ofreciendo la posibilidad de acceder a un repositorio de laboratorios virtuales y remotos que son incluidos en formato SCORM, sin embargo la respuesta a la segunda pregunta requiere que se produzcan cambios en el software del laboratorio con el fin de acceder a las funciones Javascript del API de SCORM en el que se envuelve el laboratorio.

Esto se puede lograr, facilitando además su uso, desarrollando una herramienta que consiste en un conjunto de funciones nativas a la tecnología de desarrollo del laboratorio que oculten el problema de acceso al modelo de datos del RTE de SCORM. De este modo los desarrolladores de laboratorios virtuales podrán usar estas funciones para integrarlos de forma sencilla con el LMS sin tener que conocer al detalle las características del RTE de SCORM.

Se ha utilizado Java, que es uno de los lenguajes más utilizados en el desarrollo de laboratorios online por tener características muy recomendadas entre las que destacan su amplio uso a nivel mundial, facilidad de manejo, estar generado bajo licencia libre y la posibilidad de generar el SW de laboratorio en formato *Applet* de Java. También se ha usado el paquete *netscape.Javascript.** para poder acceder al modelo de objetos del documento (*Document Object Model*, DOM) de la página en la que se incluye el *Applet* del laboratorio y poder llamar a las funciones Javascript del API de SCORM tal y como recomienda Oracle (la empresa responsable de Java) [11]. La mayoría de las funciones desarrolladas están relacionadas con la transferencia de datos Lab-LMS, estas funciones permiten acceder (y modificar en los casos en los que está permitido) al valor de los datos del modelo de datos del RTE que mantiene el LMS. Estas funciones se pueden agrupar en dos tipos:

- Funciones que permiten obtener valores del modelo de datos de RTE e incorporarlos al SW del laboratorio. A modo de ejemplo, los datos del modelo que se han consultados con estas funciones en el ejemplo descrito en apartados posteriores han sido: el nombre del alumno que el LMS mantiene en su base de datos, la versión del API y el estado de ejecución del SCO en el que se ubica el *Applet* del laboratorio con el fin de mostrarlos en una ventana del *Applet*. Para cada una de estas consultas se ha creado una función específica con el fin de facilitar su uso a los desarrolladores de laboratorios y que no tengan obligación de conocer los detalles del modelo ni de las comunicaciones que se realizan.
- Funciones que modifican algún valor del modelo de datos, y/o crean elementos cuando esto es posible, en función del trabajo realizado en la manipulación del SW de laboratorio. Merece especial mención el conjunto de datos asociados a los comentarios; Desde el laboratorio online (*Applet* de Java) se pueden crear nuevos comentarios para ser almacenados en el LMS que pueden incluir cadenas de datos escritos por el alumno o generados automáticamente por el software del laboratorio indicando además el momento (fecha y hora) y la localización en que se realizaron.

La figura 2 contiene un ejemplo de diagrama UML (*Unified Modified Language*) que muestra las entidades y los métodos necesarios para que se establezca la integración de un laboratorio virtual con un LMS por medio de recursos SCORM. Para ello el alumno (*learner*) debe introducir su nombre de usuario y contraseña para entrar en el LMS (*login LMS*). Después, mediante la navegación por los recursos y estructura del LMS, debe acceder al módulo SCORM y abrirlo, en ese momento comenzará lo que se llama un intento SCORM (*SCORM attempt*), el SCORM puede tener una estructura variada en función del diseño que se la haya dado, en el momento en el que entra en un *SCO* se entra en lo que se llama una sesión de aprendizaje (*learner sesión*) y se dispone de una instancia de la API que proporciona el LMS para acceder a las funciones del RTE, normalmente se suele acceder a la API mediante un software intermedio (*APIWrapper*) que facilita su uso. En este momento ya se pueden utilizar algunas de las funciones que permiten interactuar el *SCO* con el LMS, entre ellas está la función *Initialize()*, que marca el inicio de lo que se conoce como una sesión de comunicación RTE (*communication sesión*). También al iniciar una sesión de comunicación ya se pueden intercambiar datos entre el modelo de datos RTE que mantiene el LMS y el *SCO* o cualquier recurso que éste contenga, como es el caso de los *Applet* Java que constituyen laboratorios online. En el ejemplo de la figura 2 se utiliza la función *rteGetLearnerName()* desde el *Applet* del laboratorio, que se encuentra en el *SCO*, para obtener el valor del nombre del usuario que el LMS posee. Para ello, se debe invocar la función *doGetValue* del *APIWrapper* quien a su vez llamará a la función *GetValue* que ofrece el LMS.

4 EJEMPLO DE USO: DISEÑO DEL SISTEMA DE SUSPENSIÓN DE UNA RUEDA DE AUTOMÓVIL

Para la implementación del caso de estudio se ha creado con EJS un laboratorio virtual (aunque se puede utilizar igualmente con laboratorios remotos) en el que se realiza la simulación del funcionamiento del amortiguador para la rueda de un automóvil con la posibilidad de trabajar en distintos escenarios.

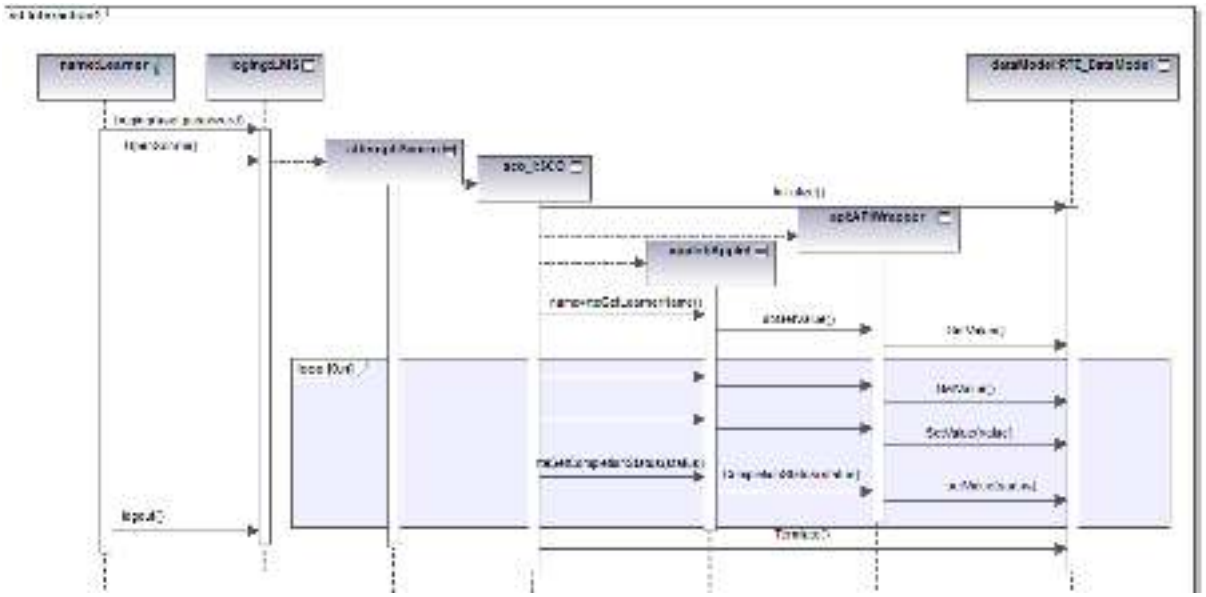


Figura 2. Ejemplo de diagrama de secuencia UML con entidades y métodos relacionados.

Este laboratorio virtual está siendo utilizado en la asignatura Automática Industrial que se imparte en el segundo curso de todos los grados de la rama de Ingeniería industrial ofertados en la Universidad de Jaén (ingeniería eléctrica, electrónica industrial, mecánica y organización industrial). Dicha asignatura tiene como finalidad general proporcionar al alumno una visión global de la aplicabilidad de la automática en entornos industriales y la ofrecer los conocimientos básicos sobre las diferentes tecnologías aplicadas al control de procesos continuos y discretos. Esta práctica se llevaría a cabo en el módulo de identificación y caracterización de sistemas dinámicos. Tiene como objetivos principales:

- Afianzar en el alumno el concepto de sistema dinámico.
- Diferenciar y extraer las características fundamentales de las distintas respuestas temporales. Para ello se le da la opción de ajustar los parámetros del modelo dinámico que caracteriza el sistema.

El laboratorio virtual desarrollado muestra el funcionamiento de un amortiguador de coche permitiendo al alumno modificar sus parámetros (*K*, constante del muelle y *B*, constante de amortiguamiento) y seleccionar distintos escenarios de funcionamiento. La figura 3 muestra una imagen del sistema real, el esquema del sistema de suspensión y el esquema del sistema completo.

Se ha utilizado un ejemplo sencillo que sirva para mostrar el trabajo desarrollado de forma simple, la ecuación del sistema de suspensión se muestra en (1).

$$m\ddot{y} = -mg - K(y - u - L0) - B(\dot{y} - \dot{u}) \qquad (1)$$

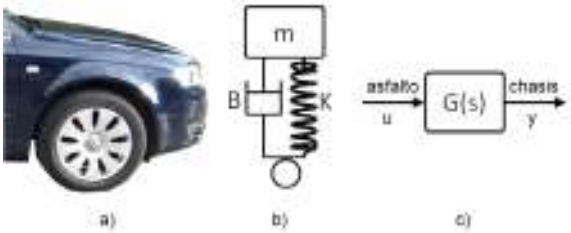


Figura 3. a) Vista delantera de un automóvil donde se aprecia el sistema chasis-rueda. b) Esquema simplificado del sistema de suspensión. c) Esquema del sistema completo.

Donde *m* es la masa del cuerpo a amortiguar (una cuarta parte del peso del vehículo), *y* es la distancia de la masa al punto de referencia, *u* es la distancia del asfalto al punto de referencia, *L0* es la distancia del muelle en reposo y *g* es la aceleración de la gravedad.

En la expresión (2) se muestra la función de transferencia que relaciona la evolución de la altura del chasis, *y(t)*, en función del asfalto, *u(t)*, para el sistema en estudio.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{Bs + K}{ms^2 + Bs + K} \qquad (2)$$

En EJS, el código Java del laboratorio, utiliza las funciones desarrolladas con el fin de interactuar con un LMS. Una vez exportado desde EJS el laboratorio en forma de *Applet* de Java se incluye dentro de una página web que constituye el recurso principal de un SCO. Se ha utilizado un fichero Javascript (*APIwrapper*) que ayuda a usar el API de SCORM. Las funciones desarrolladas en Java han tenido en cuenta el *APIwrapper* para simplificar su creación.

En la figura 4 se puede ver un esquema del laboratorio virtual EJS desarrollado incluido en un paquete SCORM una vez importado e incorporado en un LMS.

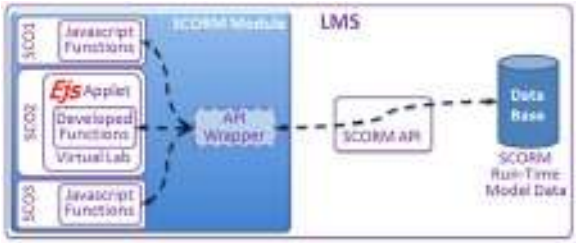


Figura 4. Esquema del ejemplo desarrollado.

Como se puede observar en dicha figura, el paquete SCORM incluye tres *SCO*, en SCO1 se ha incluido teoría sobre el funcionamiento del sistema de suspensión que simula el laboratorio virtual que el alumno debería leer antes de acceder al mismo. La figura 5 muestra una captura de pantalla del SCO1.



Figura 5. Captura de pantalla del SCO1.

En el SCO2 se ha incluido el laboratorio virtual e instrucciones de las prácticas que debe realizar, se describe con más detalle en los próximos párrafos.

Finalmente, en el SCO3 se han incluido pruebas de evaluación de los conocimientos adquiridos.

Se ha tomado esta estructura por pensar que es la más adecuada a la hora de implementar la docencia de laboratorios online. Una vez realizadas todas las modificaciones en la estructura SCORM se ha comprimido en formato zip y desde el LMS de la UJA (ILIAS) se ha importado en una carpeta de acceso público para su libre acceso [12]. Para poder ejecutarse sólo es necesario disponer de un ordenador con acceso a Internet en el que se disponga de un navegador web, la máquina virtual de Java y una configuración que permita la ejecución de *Applet*s Java en el navegador.

Las funciones utilizadas en este ejemplo para interactuar con el LMS permiten obtener del LMS:

- El nombre e identificador del alumno que accede al módulo SCORM.
- La versión de API que se está ejecutando (ofrecida por el LMS).
- El estado de terminación del laboratorio (si está superado).
- La fecha y hora de entrada en el laboratorio.
- Los comentarios guardados por el alumno sobre el laboratorio en el LMS.

Por otro lado se han desarrollado otras funciones que permiten incorporar en el LMS más datos:

- Comentarios del alumno que permanecen de forma persistente en el LMS para su consulta en sesiones e intentos posteriores por parte del laboratorio online.
- Información referente a la puntuación alcanzada por el alumno en el intento de superación de pruebas (para lograr una evaluación del mismo).
- Información referente a la superación del laboratorio.

La figura 6 muestra una captura de pantalla de este ejemplo (SCO2 con el *Applet* EJS del laboratorio virtual).

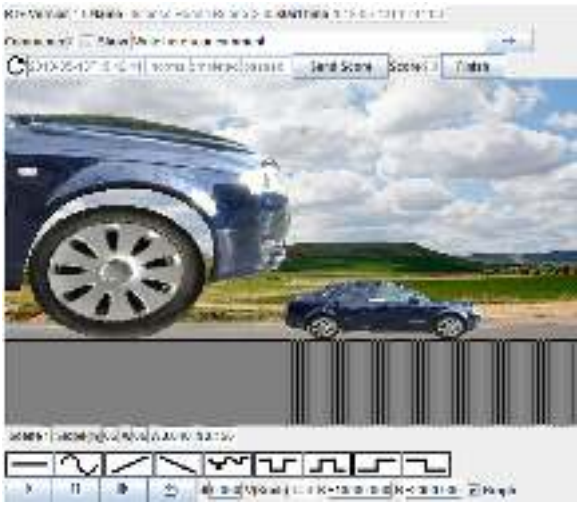


Figura 6. Captura de pantalla del *Applet* EJS del laboratorio virtual en el módulo SCORM.

En la parte superior se muestra un panel en el que se muestra información recogida por el *Applet* del LMS en el momento de su lanzamiento (que puede ser actualizada en cualquier momento) y una zona en la que el alumno que esté trabajando puede guardar los comentarios que estime oportuno en el LMS para recuperarlos cuando desee (para ello debe abrir una ventana supletoria seleccionando la casilla de verificación desde el mismo panel). En la parte central se muestran una serie de imágenes que ayudan al alumno a reconocer la escena y el funcionamiento que se logra en el amortiguador del

coche mientras que en la parte inferior aparecen tres paneles desde los que se pueden configurar las características del experimento:

- Panel superior. Los parámetros de las distintas escenas que se pueden establecer para el funcionamiento del amortiguador.
- Panel medio. Permite seleccionar los escenarios de uso del sistema.
- Panel inferior. Controla la puesta en marcha y desarrollo de la simulación así como cambiar los parámetros de configuración del sistema de amortiguación y la activación de una ventana de gráficas de resultados.

En la figura 7 se muestra una captura de la ventana de gráficos (des/seleccionable desde la ventana principal) que muestra la evolución de la altura del chasis del vehículo (y), de la carretera (*Road*), y de la diferencia entre ambas (y-*Road*).

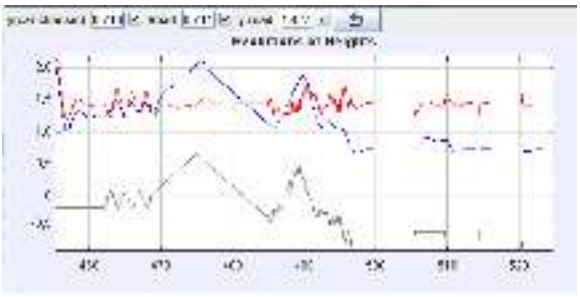


Figura 7. Captura de pantalla de la ventana emergente de gráficos del *Applet*.

En esta figura se puede observar como la evolución del chasis (y) es más "suave" que la que presenta la carretera (*Road*, línea inferior en la gráfica) debido al efecto que realiza el sistema de suspensión. Se pueden seleccionar 9 escenarios diferentes en los que se pueden realizar los experimentos, incluso combinarlos secuencialmente, como se desee. Con estas escenas y la posibilidad de cambiar las características de todas ellas según los parámetros indicados a continuación se pueden diseñar múltiples tipos de prácticas que muestren distintos comportamientos del sistema antes diferentes entradas:

- Firme plano. Permite estudiar el comportamiento del sistema ante una entrada constante. Sirve para comprobar el chasis permanece en su posición de reposo (a una distancia *L0* del asfalto).
- Subida/Bajada de acera (se puede modificar la altura de la acera). Permite estudiar el comportamiento del sistema ante una entrada de tipo escalón positivo/negativo.
- Obstáculo/Hueco en firme (se puede modificar la altura/profundidad). Permite estudiar el comportamiento del sistema ante una entrada de tipo impulso positivo/negativo.
- Firme ascendente/descendente (se puede modificar la pendiente). Permite estudiar el comportamiento del sistema ante una entrada tipo rampa ascendente/descendente.
- Firme senoidal (se puede modificar frecuencia y amplitud de la oscilación del perfil del firme). Permite estudiar el comportamiento del sistema ante una entrada senoidal.
- Firme irregular. Permite estudiar el comportamiento del sistema ante una entrada variable aleatoria.

En el ejemplo de libre acceso [12] se han diseñado dos prácticas muy sencillas. La práctica 1 pide analizar la respuesta en función de los diferentes perfiles. Para ello, se debe ejecutar la simulación, abrir la ventana de gráficos y cambiar entre los diferentes escenarios programados. La evaluación que realiza el laboratorio se basa en la comprobación de la ejecución de la simulación, añadiendo un punto por cada escenario probado. De este modo se tiene la posibilidad de conseguir un máximo de 9 puntos. En la práctica 2 se pide al alumno que asigne valores a las constantes del muelle (K) y de amortiguamiento (B) para que se cumplan ciertas condiciones. En el caso de elegir unos valores dentro de los márgenes establecidos se consiguen 5 puntos adicionales. La calificación final obtenida es comunicada al LMS al pulsar el botón *Finish* para la posible comprobación del tutor.

5 CONCLUSIONES Y LÍNEAS DE FUTURO

La integración de los laboratorios online en los LMS es un proceso que ofrece ventajas a los alumnos y profesores responsables de los mismos. Por ello se trata de una tendencia deseable que se debe desarrollar y potenciar con el fin de que la integración sea plena. En este trabajo se muestra una herramienta que facilita a los desarrolladores Java crear laboratorios online que consigan esta integración prácticamente con todos los LMS del mercado.

Esta herramienta facilita la incorporación de información procedente del LMS en el laboratorio online enriqueciendo al mismo. También facilita que el alumno pueda incorporar datos permanentes sobre el laboratorio online en el LMS que puede recuperar ayudando a la superación de las pruebas que se hayan establecido y la incorporación en el LMS del estado de superación de las prácticas que se hayan programado así como la puntuación conseguida. Con el uso de estas funciones se puede facilitar la integración de los laboratorios online (virtuales o

remotos) desarrollados en lenguaje Java mediante su inclusión como *Applet* en un paquete SCORM que puede ser utilizado por la mayoría de los LMS.

Se ha desarrollado un ejemplo que logra integrarse con el LMS ILIAS de la UJA mediante el uso de estas funciones y muestra parte del potencial de las mismas. En el ejemplo se ha simulado el sistema de suspensión de la rueda de un vehículo. Este ejemplo permite a los alumnos de los grados de ingeniería de la rama industrial de la Universidad de Jaén comprender el funcionamiento de un sistema dinámico y comprobar su comportamiento ante distintas entradas de forma inmediata.

Actualmente se está trabajando en el desarrollo de un paquete Java que incluya funciones que permitan trabajar con todos los datos del modelo SCORM y a más largo plazo se pretende crear una definición multilenguaje de estas funciones que permitan a los desarrolladores de todos los lenguajes crear laboratorios online usando funciones que permiten interactuar con el LMS de forma sencilla y fácil.

Agradecimientos

Este trabajo ha sido financiado parcialmente por los proyectos DPI2011-27284, TEP2009-5363 y AGR-6429

Referencias

[1] Advanced Distributed Learning, (2012) “SCORM 2004 4th Edition Specification”. [Online]. Available: http://www.adlnet.gov/wp-content/uploads/2011/07/SCORM_2004_4ED_v1_1_Doc_Suite.zip

[2] Advanced Distributed Learning, (2013). “SCORM Certified Products”. [Online]. Available: http://www.adlnet.gov/wp-content/uploads/2012/04/SCORMCertifiedProductsLocked.xlsx

[3] Aktan, B., Bohus, C.A., Crowl, L.A. & Shor, M.H. (1996) "Distance learning applied to control engineering laboratories". IEEE Transactions on Education 39 (3) pp.320-326, DOI:10.1109/13.538754.

[4] Dormido, S., Vargas, H., Sánchez, J., Dormido, R., Duro, N. & Morilla, F. (2008) "Developing and implementing virtual and remote labs for control education: The UNED pilot experience". 17th IFAC World Congress 2008, pp. 8159-8164

[5] Esquembre, F. (2003) "Using Easy Java Simulations to create scientific simulations in Java". EUROCON 2003. Computer as a Tool. The IEEE Region 8 Vol:1, 2003, pp.:20-23

[6] Feisel, L. & Rosa, A. (2005) “The role of the laboratory in undergraduate engineering education”. Journal of Engineering Education, 94(1), pp.121-130.

[7] Gutiérrez, E., Trenas, M.A., Ramos, J., Corbera, F., and Romero, S. (2010) "A new moodle module supporting automatic verification of VHDL-based assignments". Computers & Education 54(2), pp.562-577

[8] Harward, V.J., Alamo, J.A. del, Lerman, S.R., Bailey, P.H., Carpenter, J., DeLong, K., Felknor, C., Hardison, J. and Harrison, B. (2008) "The iLab shared architecture: A web services infrastructure to build communities of internet accessible laboratories" Proceedings of the IEEE,Vol:96, Issue:6, pp.931–950, DOI: 10.1109/JPROC.2008.921607

[9] Henry, J. (1996) “Controls laboratory teaching via the World Wide Web” ASEE Annual Meeting, Washington, D.C.

[10] Mateos, V., Gallardo, A., Richter, T., (2011) "LiLa booking system: Architecture and conceptual model of a rig booking system for on-line laboratories". International Journal of Online Engineering (iJOE),2011, 7 (4)

[11] Oracle (2013). "Invoking JavaScript code from an applet". [Online]. Available: http://docs.oracle.com/Javase/tutorial/deployment/applet/invokingJavaScriptFromApplet.html

[12] Ruano, I. (2013) “SMC2013, SCORM module with an example of virtual lab integrated with ILIAS LMS”. [Online]. Available: http://dv.ujaen.es/docencia/goto_docencia_sahs_428996.html

[13] San Cristobal, E., Castro, M., Hardward, J., Baley, P., DeLong, K. and Hardison, J. (2010) "Integration view of web labs and learning management systems". IEEE EDUCON Education Engineering- The future of Global Learning Engineering Education 81, pp.1409-1417

[14] Torre, L. de la, Sánchez, J. P., Heradio, R., Carreras, C., Yuste, M., Sánchez, J. and Dormido, S., (2012) "UNEDLABS: An example of EJS labs integration into moodle", World Conference on Physics Education, Istambul, Turkey.

DESARROLLO DE UN EQUIPO DIDÁCTICO PARA LAS MATERIAS DE CONTROL DE PROCESOS DE LOS GRADOS DE INGENIERÍA DE LA RAMA INDUSTRIAL Y DE SISTEMAS TIC

Teresa Escobet Canal
Departament de Disseny i programació de Sistemes Electrònics
Grupo de Sistemas Avanzados de Control
Universitat Politécnica de Catalunya
teresa.escobet@upc.edu

Antoni Escobet Canal
Departament de Disseny i programació de Sistemes Electrònics
Grupo Soft Computing
Universitat Politécnica de Catalunya
Toni@epsem.upc.edu

Joan Martínez Domene
Departament de Disseny i programació de Sistemes Electrònics
Estudiante de Ingeniería de Sistemas TIC
Universitat Politécnica de Catalunya
juan.martinez.domene@estudiant.upc.edu

Resumen

En este artículo se presenta el desarrollo de un equipo didáctico para la enseñanza práctica de la teoría de control. El equipo permite la realización de experimentos en lazo abierto o cerrado, y puede ser controlado con dispositivos industriales o microprocesadores, adaptando las señales adecuadamente. El trabajo experimental propuesto abarca aspectos de análisis del proceso, diseño de controladores utilizando técnicas de asignación de polos o técnicas de sintonia empírica, y su implementación en base a un microprocesador.

Palabras clave: Equipo didáctico, control de sistemas, sistema microprocesador.

1. INTRODUCCIÓN

El trabajo presentado en este artículo tiene como objetivo el de adaptar el laboratorio de control y automatización industrial a los estudios de grado que actualmente se imparten en la Escola Politècnica Superior d'Engineria de Manresa (EPSEM). Para ello se planteó el desarrollo de un equipo didáctico de bajo coste con el cual ilustrar fácilmente los conceptos relacionados con la teoría de control y de arquitectura de control abierta.

El mercado ofrece actualmente una gran variedad de equipos con los cuales ilustrar los fundamentos de la Ingeniería de Control. Si bién es interesante diponer de dichos equipos en el laboratorio, es económicamente inviable a la hora de plantearse equipar un laboratorio con diez bancos de trabajo. Ante ello y debido a la gran importancia de las enseñanzas prácticas en el ámbito de la ingeniería [2] los docentes han ideado estratégias para facilitar el acceso a bancos de prueba, ya sea con procesos simulados, laboratorios virtuales o la utilización de laboratorios remotos ([5], [4]).

En la EPSEM, además de utilizar herramientas de simulación se ha creido conveniente desarrollar un equipo didáctico con una arquitectura de control abierta para prácticas de las asignaturas de control básico y control digital.

Mencionar que actualmente en la EPSEM se imparten cuatro grados en Ingeniería del ámbito industrial: Eléctrica, Electrónica Industrial y Automática, Mecánica y Química, además el grado en Ingeniería de Sistemea TIC, grado que tiene por objetivo formar profesionales con capacidad de integrar las tecnologías Electrónica, Informática y Comunicaciones [6]. Debido a ello, se requiere que el equipo desarrollado pueda ser fácilmente utilizado por los alumnos que cursan el grado de Ingeniería del ámbito industrial, en el marco de la asignatura de Control y Automatización Industrial, asignatura troncal impartida en el cuarto semestre. En este caso y con el objetivo de limitar los desarrollos prácticos al análisis y diseño de controladores, se dispone de una arquitectura de control

cerrada. Por otro lado, los alumnos del grado de Ingeniería de Sistemas TIC, utilizan el equipo en el marco de la asignatura de Sistemas Electrónicos de Control, trocal del sexto semestre. A diferéncia del primer grupo, los alumnos que cursan el grado en Ingeniería de Sistemas TIC, disponen ya de un amplio conocimiento en electrónica analógica y digital, y de programación de dispositivos, por lo tanto se les proporciona una arquitectura de control abierta que permite analizar, diseñar el sistema de control y realizar su implementación en base a un microprocesador.

El trabajo se ha estructurado de la siguiente forma, la sección 2 describe el equipo desarrollado y un sistema cerrado de monitorización, la sección 3 describe el sistema de control utilizando el microprocesador Arduino y, finalmente, la sección 4 presenta las conclusiones del trabajo.

2. DESCRIPCIÓN DEL EQUIPO DESARROLLADO

2.1. COMPONENTES DEL EQUIPO

La figura 1 muestra el equipo de levitación aerodinámica desarrollado en el laboratorio. El tubo utilizado tiene 6 cm de diámetro y 60 cm de longitud. Dispone de una ranura recubierta con un material plástico que permite visualizar la posición de la bola en su interior. Como sistema de actuación se ha seleccionado un ventilador axial de 24VDC y 0.26A, con una velocidad de rotación máxima de 7350 rpm y un desplazamiento de volumen de aire de 1.05 m^3/min. Para su funcionamiento se requiere un adaptador de potencia, en el equipo diseñado se utiliza el amplificador CC R-24.



Figura 1: Equipo de levitación aerodinámica

El sistema sensorial consiste en un sensor ultrasiónico de características: rango de salida de 0 a 10 V, rango de detección de 30 a 500 mm ajustable, zona ciega de 0 a 30 mm, frecuencia del transductor de aproximadamente 290 kHz y rango de respuesta de 50 ms. El equipo cuenta con una fuente de alimentaciión própia, FE-36, junto con los dispositivos de seguridad adecuados.

Desde el panel frontal del equipo (figura 1) se tiene acceso a la señal proporcionada por el sensor ultrasónico y, mediante un selector, se permite al alumno configurar el modo de operación del equipo. Se dispone de dos modos de operación el interno o el externo. Al configurarse en modo interno se puede variar la tensión aplicada al ventilador utilizando un potenciómetro. Ello permite observar el comportamiento de la bola en el interior del tubo sin necesidad de elementos externos, facilitandose la compresión del objetivo de control. En modo externo, la regulación de la tensión aplicada al ventilador se realiza mediante un dispositivo externo, pudióendose trabajar en lazo abierto o en lazo cerrado según sea menester. La velocidad del ventilador se regula aplicando una tensión entre 0 y 10V.

El diagrama de bloques de la figura 2 muestra, de forma sintetizada, el equipo desarrollado. Se dispone de dos arquitecturas de contol en tiempo real. Una de ellas se basa en el conjunto tarjeta de adquisición de datos NI PIC-6221 y PC, un entorno gráfico desarrollado en LabWindows (figura 3) permite acceder a los canales de entrada y salida de la tarjeta y operar el sistema diseñado. El segundo utiliza el microprocesador Arduino como sistema de control tal y como se describe en la sección 3.

2.2. MODELO DEL SISTEMA

El equipo diseñado tiene por objetivo controlar la posición de una bola en el interior de un tubo mediante el flujo de aire proporcionado por un ventilador. Un objeto en flotación se gobierna mediante fuerzas activas como la fuerza gravitatoria y la fuerza del aire producido por el ventilador. Utilizando los principios fundamentales de la dinámica, la dinámica de la bola en el interior del tubo viene dada por

$$m\frac{d^2y(t)}{dt^2} = mg - F(t) \qquad (1)$$

i la modelización del ventilador accionado mediante un motor cc, del cual se ha despreciado la fricciión, la carga y la impedancia de la parte eléctrica

$$J_m\frac{\omega^2 y(t)}{dt^2} + K_m\omega(t) = K_e u(t) \qquad (2)$$

en donde m es la masa de la bola, y es la posición de la bola en el tubo, g es la gravedad, F es la fuerza de arrastre del flujo de aire, J_m, K_m y K_e son características del motor CC, ω es la velocidad

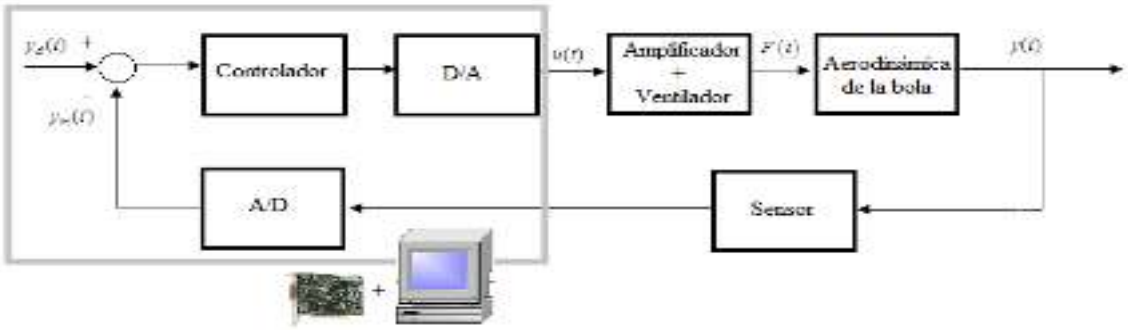


Figura 2: Diagrama de bloques del sistema

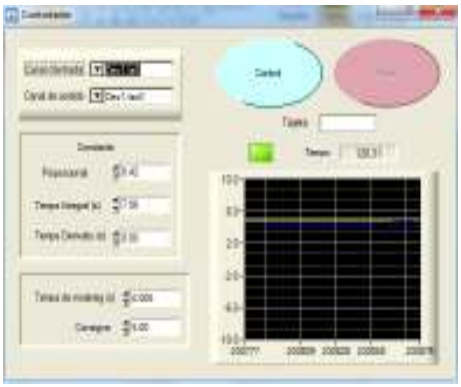


Figura 3: Entorno gráfico de monitorización en tiempo real

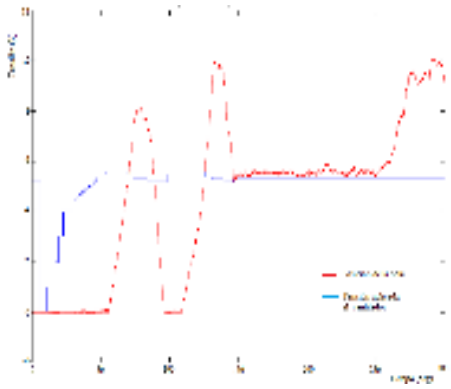


Figura 4: Respuesta del sistema en lazo abierto

de rotación y u la tensión aplicada al motor. La fuerza de arrastre del flujo de aire puede ser modelada utilizando una relación de arrastre estándar [7]

$$F(t) = \frac{1}{2} C_d \rho A \left(v_{aire} - v_{bola}\right)^2 \qquad (3)$$

siendo C_d el coeficiente de arrastre, ρ la velocidad del aire, A el frontal de la área de la bola, v_{aire} velocidad media de la columna de aire y v_{bola} la velocidad absoluta de la bola. La velocidad del aire es función de la velocidad de rotación del ventilador.

2.3. SISTEMA DE MONITORIZACIÓN

El control en tiempo real del sistema se realiza mediante la tarjeta de adquisición de datos NI PIC-6221 junto con un programa de monitorización en tiempo real realizado en LabWindows. Dicho programa dispone de dos menús: respuesta del sistema y control PID. El primero permite visualizar la respuesta del sistema resultado de variar la tensión aplicada al ventilador y el segundo permite controlarlo introduciendo los parámetros del controlador (figura 3). El análisis de los resultados se realiza utilizando Matlab.

La figura 4 muestra el comportamiento de la bola en lazo abierto. En este escenario la tarea planteada consiste en variar manualmente la tensión aplicada al ventilador hasta conseguir que la bola se mantenga en equilibrio a una cierta altura del tubo. En el primer tramo de la figura, se pone el potenciómetro a cero y se va incrementando la tensió aplicada al ventilador. Obsévese que la bola empieza a levitar cuando la tensió aplicada es de unos 6V. A partir de aquí, debe decrementarse la tensió aplida para conseguir el objetivo, mantener la bola en flotación. En los tiempos comprendidos entre 150 y 250 seg, se ha conseguido el objetivo. Evaluando la ecuación (1) se deduce que el equilibrio se consigue cuando $F(t)$ iguala la fuerza gravitacional. El comportamiento inicial del sistema indica la preséncia de una zona muerta, es decir, la bola no empieza a levitar hasta que $F(t)$ no supera la fuerza gravitatoria. También se observa una zona de saturación en y=10 V que coincide con la longitud máxima del tubo. En la figura 4 también se muestra el comportamiento integrador del sistema, tal y como se deduce de la ecuación (1).

El programa de monitorización permite cerrar el lazo mediante un controlador PID. Relacionado con el lazo cerrado y diseño del controlador PID, se proponen tres actividades:

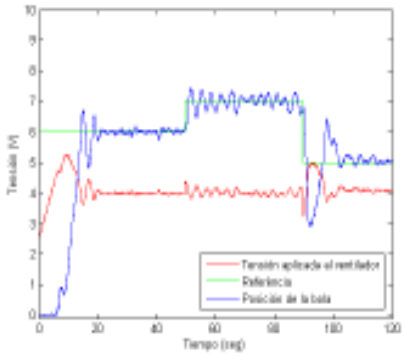


Figura 5: Respuesta del sistema en lazo cerrado

- Utilizando el procedimiento de prueba y error, ajustar un controlador P de forma que ante un cambio escalón la bola tenga una sobre oscilación inferior al 20 %;
- Con dicho ajuste en lazo cerrado, aplicar un cambio escalón del punto de control y estimar los parámetros del sistema en lazo abierto;
- Utilizando el modelo estimado y dadas las especificaciones de funcionamiento, sobre oscilación máxima de 20 % y tiempo de estabilización de 8 seg, diseñar un PI utilizando las técnicas de diseño por asignación de polos.

La función de transferéncia estimada en uno de los equipos es de

$$G_P = \frac{1{,}206}{s\,(s + 0{,}7484)} \tag{4}$$

Aplicando la técnica de asignación de polos se obtien los valores $K_c = 0{,}429$ y $T_i = 7{,}06$, obteniéndose la respuesta de la figura 5. El regulador PI se ha diseñado para estabilizar la bola en levitación, debido a ello en la zona inicial se observa un retardo importante. También se observa que el sistema tiene un comportamiento no lineal debido principalmente a la turbulencia del flujo de aire.

3. DESEÑO DEL SISTEMA DE CONTROL CON ARDUINO

3.1. MICROCONTROLADOR ARDUINO

La plataforma Arduino [1] es una placa de desarrollo hardware de código abierto y software de programación libre, muy versátil y asequible. La parte hardware esta basada en un microprocesador de 8 bits de la familia AVR de Atmel, siendo los microprocesadores ATmega328 y ATmega168 los mas utilizados. El software consiste en un entorno de programación que implementa el lenguaje Processing/Wiring y puede ser ejecutado en sistemas operativos Windows, Macintosh OSX y GNU/Linux. Al ser open-hardware, tanto su diseño como su distribución son libres y pueden utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o se puede conectar a un ordenador. Las placas se pueden montar a mano o adquirirse. La figura 6 muestra una placa con Arduino UNO. Como caracteríticas de Arduino UNO decir que utiliza el microprocesador Atmega328, tiene 14 pines digitales de entrada / salida (de las cuales 6 se puede utilizar como salidas PWM), 6 entradas analógicas (etiquetadas de A0 a A5, con 10 bits de resolución), un oscilador o resonador cerámico de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar al microcontrolador, basta con conectarlo a un ordenador con un cable USB o alimentarlo con un adaptador AC-DC o batería para empezar. ATmega328 dispone de 32 KB (con 0,5 KB utilizados por el gestor de arranque), 2 KB de SRAM y 1 KB de EEPROM (que puede ser leído y escrito con la librería EEPROM).

Cada uno de los 14 pines digitales del UNO se puede utilizar como una entrada o salida, y se accede a ellos utilizando las funciones digitalRead (), pinMode () y digitalWrite (). Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia pull-up interna de 20 a 50 KΩ. Algunos de los pines tienen funciones especializadas, por ejemplo: RX y TX, se utilizan para recibir y transmitir datos TTL en serie, respectivamente; los pines de interrupciones externas pueden ser configurados para activar una interrupción a nivel bajo, por un flanco ascendente o descendente, o por un cambio en el valor; los pines SS, MOSI, MISO y SCK soportan la comunicación SPI usando la librería SPI.



Figura 6: Caracteristicas de la placa Arduino UNO

3.2. ADAPTACIÓN DE LAS ENTRADAS Y SALIDAS

La connexión de Arduino con el equipo de levitación aerodinámica, requiere la adaptación de las entradas y las salidas. Concretamente se debe diseñar un sistema que convierta la salida PWM a una señal de tensión de 0 a 10 VDC, y, por otro lado, se requiere adaptar la salida del sensor (0 a 10 VDC) a la señal analógica de entrada requerida por Arduino de 0 a 5 VDC. En la figura 7 se muestra un esquema del circuito de adaptación diseñado.

Debido a que la impedancia de entrada del conversor ADC es elevada, del orden de 100 MΩ, puede considerarse que el consumo de corriente será despreciable, por lo tanto la conversión de 0-10 V a 0-5V se ha realizado mediante un divisor de tensilón resistivo con dos resistencias de 100 KΩ (ver zona A de la figura 7).

La adaptación de la señal PWM a una tensión de 0 a 10 V se realiza mediante un amplificador operacional Rail-to-Rail, de forma que a la salida del amplificador se aproveche todo el margen de tensión proporcionado por la fuente de alimentacón. Esta etapa amplificadora se visualiza en la zona B de de la figura 7. En JP3 y JP2 se conectan la salida del sensor y la fuente de alimentación respectivamente; en JP5 se obtine la señal de regulación aplicada al ventilador.

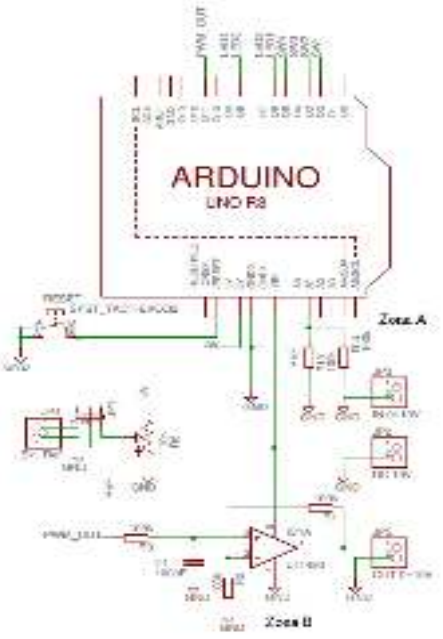


Figura 7: Conexión del sistema con el microprocesador Arduino

3.3. CONTROL DE POSICIÓN DE LA BOLA UTILIZANDO ARDUINO

A diferencia de la técnica propuesta en la sección 2.3, en este caso se pretende sintonizar un regulador PID utilizando técnicas empíricas. Al tratarse de un sistema inestable en lazo abierto, la caracterización del sistema se realizará en lazo cerrado y concretamente utilizando la técnica del relé. Resumidamente se han propuesto dos retos relacionados con el control de la posición de la bola utilizando Arduino: control todo/nada y control PID.

3.3.1. Control todo/nada

Como es sabido, el método del relé, propuesto por Aström y Hägglund en 1984 [3] constituye una forma indirecta de automatizar el método de la oscilación mantenida. El método consiste en provocar un ciclo límite mediante la inclusión de un lazo de control no lineal, este ciclo límite tendrá aproximadamente el mismo período T_u que la oscilación mantenida.

Se propone utilizar el entorno de programación de Arduino para implementar el controlador relé de la forma indicada en la figura 8, observar los resultados y a partir de ellos calcular la ganancia de oscilación, K_u, y el período de oscilación, T_u.

El resultado de aplicar dicha técnica a uno de los equipos desarrolladas se muestra en la figura 9. En ella se determinan las siguientes características: $T_u = 2{,}6seg$, amplitud de la entrada, $d = 2{,}13$ y aplitud de la salida, $a = 1{,}45$. Aplicando la fórmula (5) se calcula que $K_u = 1{,}87$

$$K_u = \frac{4d}{\pi a} \tag{5}$$

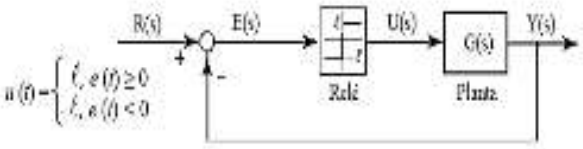


Figura 8: Diagrama de bloques utilizando el método del relé

3.3.2. Control PID

Para el diseño del controlador se proponen aplicar dos técnicas de sintonia empírica, el ajuste de parámetros Ziegler-Nichols y la técnica de Pecharromán [8]. La primera es una técnica muy conocida y pensada para sintonizar controladores tipo PID cuando el sistema puede ser aproximado con un model de primer orden más un retardo puro. En este caso los valores de los parámetros del

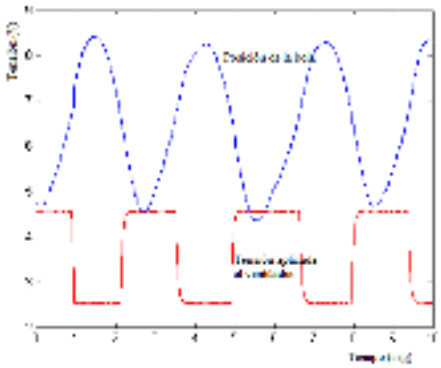


Figura 9: Respuesta al aplicar un control tipo relé

controlador se calculan mediante $K_c = 0{,}4K_u$ y $T_i = 0{,}8T_u$, siendo K_c la ganáncia de l'acción proporcional y T_i el tiempo de acción integral.

La segunda es menos conocida, se trata de una técnica de sintonia que tiene por objetivo ajustar un controlador PI con peso en la referécia que minimiza la integral del error absoluto ponderado en el tiempo (ITAE) cuando el sistema se comporta como un integrador más un retardo puro (figura 10), conocido como modelo IPD. En este caso los parámetros del controlador se calculan mediante las expresiones; $K_c = x_1 K_u$ y $T_i = x_2 T_u$. La tabla 1 muestra alguno de los valores de x_1, x_2 y α en funció de la fase deseada a la frecuencia de cruce de ganancia, ϕ_c.

Una vez calculados los valores del los parámetros del controlador PI, se ha procedido a su implementación en Arduino. El reto consiste en implementar un PI digital con peso en la referéncia y anti windup con integración condicionada (6). La integración condicionada evita que la salida se incremente cuando el controlador está saturado. Las ecuaciones implementadas son,

$$\begin{aligned} P(t_k) &= K_c\left(by_d(t_k) - y(t_k)\right) \\ e(t_k) &= y_d(t_k) - y(t_k) \\ u(t_k) &= P(t_k) + I(t_k) \\ I(t_{k+1}) &= I(t_k) + \frac{K_c h}{T_i} e(t_k) \\ if\left(u(t_k+1) > u_{max}\right), I(t_{k+1}) &= I(t_k) \\ elseif\left(u(t_k+1) < u_{min}\right), I(t_{k+1}) &= I(t_k) \end{aligned} \quad (6)$$

en donde h es el periodo de muestreo; u_{max} y u_{max} son los valores máximos y mínimos de la acción de control; b es el peso de la referéncia, en el caso de ajuste de parámetros con Ziegler-Nichols $b = 1$ y con la técnica de Pecharromán $b = (1 - \alpha)$.

Los resultados obtenidos tras aplicar las dos técnicas de sintonia se muestran en las figuras 11 y 12, respectivamente. Tal como era de prever, la técnica de Ziegler-Nichols ajusta controladores con respuestas rápidas pero más inestables (figura 11). Por contra la técnica de Pecharromán origina respuestas lentas pero más estables. Este comportamiento se obseba en la etapa incial, en el primer caso se requieren 3 seg. para que la bola empieze a levitar y en el segundo caso la espera es de 20 seg.

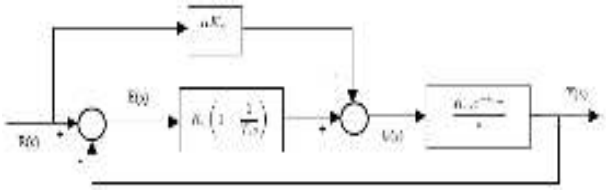


Figura 10: Sintonia de un controlador PI con un modelo IPD

Tabla 1: Valores de los coeficientes en la técnica de Pecharromán.

x_1	x_2	α	ϕ_c
0,049	2.826	0.506	-164°
0.066	2.402	0.512	-160°
0.129	1.716	0.532	-150°
0.189	1.392	0.555	-140°
0.250	1.216	0.573	-130°
0.330	1.114	0.579	-120°

El programa siguiente muestra como implementar las ecuacionses descritas en (6) con Arduino.

```
float Process_PI(mypid_t *MyPI, float
    entrada){
    static float integral = 0;
    static float MAX = 1024;
    static float MIN = 0;
    float error;
    float proporcional;
    float output, pre_int;

    MyPI->valor_entrada = entrada;
    Error = MyPI->valor_consigna -
        entrada;
    MyPI->valor_error = error;
    d_t = MyPI->delta_t;
    Kp =  MyPI->Kp;
    Ki =  MyPI->Ki;
    B =  MyPI->B;
    r = MyPI->valor_referencia;

    pre_I = I;
    I = I+ Ki*(error*d_t) / T_refrs  ;
    Pl = Kp *(B * r - entrada);
    output = pl + i;

     if(output > MAX)  I = pre_int;
     else if(output < MIN)  I =
         pre_int;

    return output;
}
```

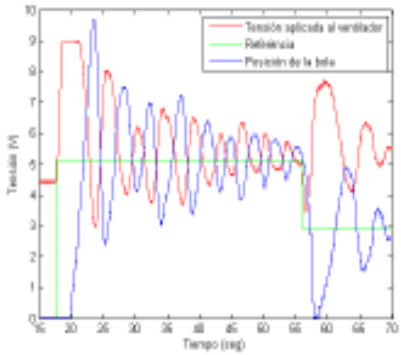


Figura 11: Respuesta del sistema con el controlador sintonizado con las técnicas de Ziegler-Nichols.

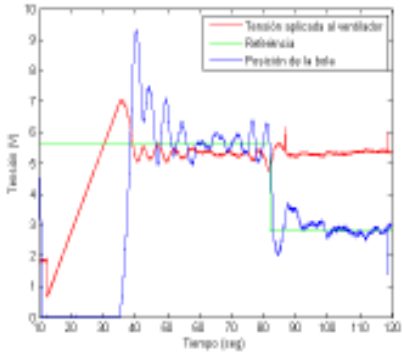


Figura 12: Respuesta del sistema con el controlador sintonizado con las técnicas de Pecharromán.

4. CONCLUSIONES

En este trabajo se ha descrito brevemente el equipo de laboratorio desarrollado para cubrir las necesidades de las asignaturas relacionadas con el control de procesos. Este equipo presenta una arquitectura de control abierta de forma que puede adaptarse fácilmente a los objetivos docentes de cada asignatura. Como ejemplo se han presentado dos desarrollos prácticos, el primero para cubrir las necesidades a la asignatura de Control y Automatización Industrial y el segundo para la asignatura de Sistemes Electrónicos de Control. En este último caso el equipo se ha controlado utilizando el microprocesador Arduino UNO.

Actualmente el laboratorio dispone de diez equipos de levitación areodinámica que han sido utilizados por un total de 150 alumnos, siendo los resultados muy satisfactorios. Se prevé utilizar el equipo desarrollado para experimentar conceptos de control no lineal basado en el diseño de controladores múltiples que contemple, por ejemplo, el modo de arranque y el modo en levitación, y control en tiempo real, diseñando el planificador o utilizando pequeños sistemas operativos de tiempo real como el FreeRTOS o los Protothreads.

Agradecimientos

Los autores desean mostrar su agradecimiento a la sección de Terrassa del Departamento ESAII por la ayuda proporcionada en el desarrollo del equipo y al personal técnico de l'EPSEM por la realización del montaje del equipo.

Referencias

[1] *Arduino Online*, 18 de junio de 2013 (http://www.arduino.cc/).

[2] Abdulwahed, M. and Nagy, Z. K.. (2008). Towards constructivist laboratory education: Case study for process control laboratory, Loughborough University: Institutional Repository. Recuperada en Mayo 3, 2012, del sitio Web temoa : Portal de Recursos Educativos Abiertos (REA) en (http://www.temoa.info/es/node/186787).

[3] Aström, K., and Hägglund, T. (1995), PID Controllers: Theory, Design and Tunig, Instrument Socitey of America, 2nd edition.

[4] De La Cruz, F.; Díaz-Granados, M.; Zerpa, S.; Giménez, D. (2010), Web-LABAI: Laboratorio Remoto de Automatizacióm Industrial, *Revista Iberoamericana de Automatización Industrial*, 7(1), p. 101-106.

[5] Farias, G.; Cervin, A.; Arzen, K-E.; Dormido, S.; and Esquembre, F. (2010), Java Simulations of Embedded Control Systems, *Sensors*, 10, p. 8585-8603.

[6] Giralt, M.R; Palà, P.; Escobet, T.; Vila, S.; y Argelaguet, M. R. (2012), La integración de las tecnologías electrónica, informática y comunicaciones: diseño del nuevo grado en ingeniería de sistemas TIC, A: *XIX Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas*. Barcelona, p. 1088-1099.

[7] Jernigan,S. R., Fahmy, Y., and Buckner, G. D. (2009), Implementing a Remote Laboratory Experience Into a Joint Engineering Degree Program: Aerodynamic Levitation of a Beach Ball, *IEEE transaction on education*, 52(2), p. 205-213.

[8] O'Dwyer, A., (2006), Handbook of PI and PID Controller Tuning Rules, Ed. Imperial College Press.

APLICACIÓN DE CONTROL CON LABVIEW PARA EL SIEMENS S7-1200, EN RED LOCAL O INTERNET

Eduardo J. Moya de la Torre
(ITAP) Instituto de las Tecnologías Avanzadas de la Producción, Ing. de Sistemas y Automática, EII, Paseo del Cauce 59, Valladolid, 47011, España, edumoy@eis.uva.es

Francisco J. García Ruíz,, Israel Surribas Planas
EII, Ing. de Sistemas y Automática, EII, Paseo del Cauce 59, Valladolid, 47011, España, javgar@eii.uva.es,
http://undetantos.webcindario.com

Resumen

Este artículo muestra una aplicación de control con Labview para el Siemens S7-1200, en red local o internet. Surge de la necesidad de medir la temperatura y humedad en una vivienda remota, lugares de ejercicio físico como gimnasios o recintos cerrados, habitáculos con problemas de humedades u otros lugares en el se necesite este tipo de medidas. Imaginemos como ejemplo, un mobiliario antiguo de alto valor económico que se pretende controlar su estado de conservación, para ello se hace indispensable el control de estos parámetros. Así mismo también se integra un control de luminosidad, para ser usado en lugares donde haya que tener sumo cuidado con la luz para no deteriorar elementos susceptibles a ella. Todos ellos son ejemplos que cubren la domótica.

Palabras Clave: Control, Domótica, Labview, OPC, SCADA, Autómata S7-1200, Red Local, Internet.

1 INTRODUCCIÓN

Los sistemas domóticos incorporan a las viviendas las últimas tecnologías informáticas y de comunicaciones, proporcionando cuantiosas ventajas a sus usuarios. Estas tecnologías ayudan a reducir consumos energéticos y a aumentar la seguridad, las posibilidades de comunicaciones y sobre todo, la comodidad. Sin embargo, la situación y cultura actuales conceden poco optimismo a la domótica. En el departamento ISA de la Escuela de Ingenierías Industriales de Valladolid se están desarrollando diversos trabajos sobre tecnologías que se podrían incorporar a los sistemas domóticos para mejorar los servicios ofrecidos y facilitar su penetración en el mercado.

La palabra domótica deriva de la unión de domus e Informática, hace referencia a la incorporación a la vivienda de un conjunto de tecnologías informáticas y de comunicaciones que permiten gestionar y automatizar desde un mismo sistema las diferentes instalaciones de uso cotidiano en una vivienda, proporcionando una mejor calidad de vida de los usuarios de la misma y una mejor conservación y cuidado del edificio.

Las primeras iniciativas en el mundo de la domótica intentaron abrirse paso en el mercado resaltando aspectos superficiales, que realmente no aportaban valor añadido al usuario, como podía ser el mecanizado de ciertos elementos de la vivienda (persianas, riego, iluminación) o la posibilidad de "jugar" con ellos. Actualmente, con una tecnología mucho más madura y con una nueva visión de las necesidades a cubrir, cuatro aspectos fundamentales permiten justificar la inversión en una instalación domótica: Ahorro energético, Seguridad, Comunicaciones, Confort y comodidad, Discapacidad [xx].

En este trabajo se presentan algunos de los campos de aplicación, como puede ser la medición de temperatura para hornos y cámaras frigoríficas, o para la automatización de un hogar incluyendo en esta el control de luminosidad entre otros.

Tal y como hemos comentado uno de los aspectos que más influyen en la realización de un proyecto de automatización es el coste económico y la viabilidad. Por ello la elección de los componentes fue determinante en el desarrollo de la aplicación. El sistema puede dar solución a un claro ejemplo, puede ser aplicable al control de una segunda vivienda. De esta manera, sabremos en qué estado de humedad y temperatura se encuentra, además de controlar la luminosidad de las instancias de manera monitorizada.

Se presenta una solución económica, siendo aplicable a habitáculos que dependan de la activación de distintos elementos actuadores en función de la temperatura o humedad, así como la luminosidad del mismo. Llevando un control vía internet o red local, para así tener el sistema controlado por distintos usuarios, eso sí, sólo uno tendrá el control del mismo en la toma de decisiones.

Mediante un panel frontal se visualizará en tiempo real los valores de dicha estancia además de seleccionar el porcentaje de intensidad que circulará

por la lámpara. Al mismo tiempo, para valores especificados por el usuario podrán entrar en funcionamiento los circuitos conectados al autómata.

Las variables a medir son la temperatura y humedad, y la variable a controlar la intensidad o potencia consumida por la lámpara. Estos parámetros además se registrarán en un documento excel, así puede analizarse los valores obtenidos. La aplicación podría ser aplicable a cualquier tipo de habitáculo o industria.

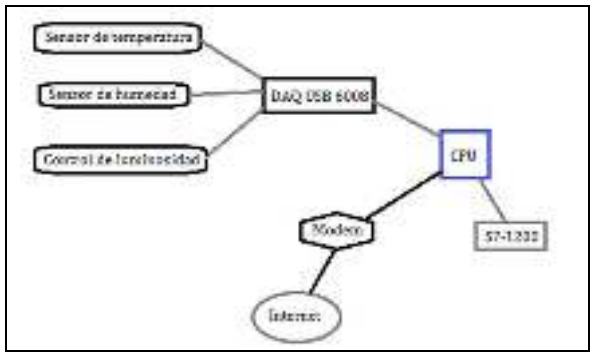


Figura 1. Esquema general del sistema diseñado.

Por lo tanto los objetivos que se pretenden obtener con este trabajo son:

- Evitar la pantalla HMI utilizando un PC en el cual desarrollar con Labview un entorno gráfico bastante adecuado para la aplicación.
- Controlar la aplicación vía internet o red local. Hoy en día, muchas aplicaciones industriales y ya no industriales requieren estas exigencias, con lo que es uno de los objetivos.
- Diseñar y construir los elementos electrónicos necesarios para los distintos componentes (software KICAD). Se elegirán distintos componentes SMD.
- Manejar de la tarjeta de adquisición de datos DAQ USB 6008 de National Instruments. Con ella se adquieren las distintas señales, procedentes de los sistemas diseñados, además de controlar la tensión de salida hacia el control de luminosidad.
- Configurar y programar el autómata S7_1200 con la nueva herramienta TIA PORTAL.
- Realizar las distintas parametrizaciones y programaciones necesarias en el entorno de programación de Labview; así como, la conexión o comunicación entre los distintos programas utilizados.

El artículo está organizado de la manera siguiente Aquí se muestra una breve introducción y objetivos a desarrollar. En la segunda sección se presentan los distintos elementos del sistema de control utilizados, En la siguiente sección se presenta el diseño de las placas utilizadas especialmente en la aplicación, como son el sistema sensor de temperatura y humedad, luminosidad y fuente de alimentación. En la cuarta sección se explica toda la programación realizada y las parametrizaciones necesarias en las comunicaciones para el desarrollo de la aplicación. Finalmente se presentan las conclusiones finales y las líneas de trabajo planteadas en un futuro inmediato, así como las referencias bibliográficas utilizadas.

2 ELEMENTOS DEL SISTEMA DE CONTROL

A continuación vamos a describir los distintos elementos del sistema de control y los equipos que hemos utilizados como son: el SCADA, el OPC, la arquitectura de comunicación, la tarjeta de adquisición de datos, el Autómata, la plataforma de programación.

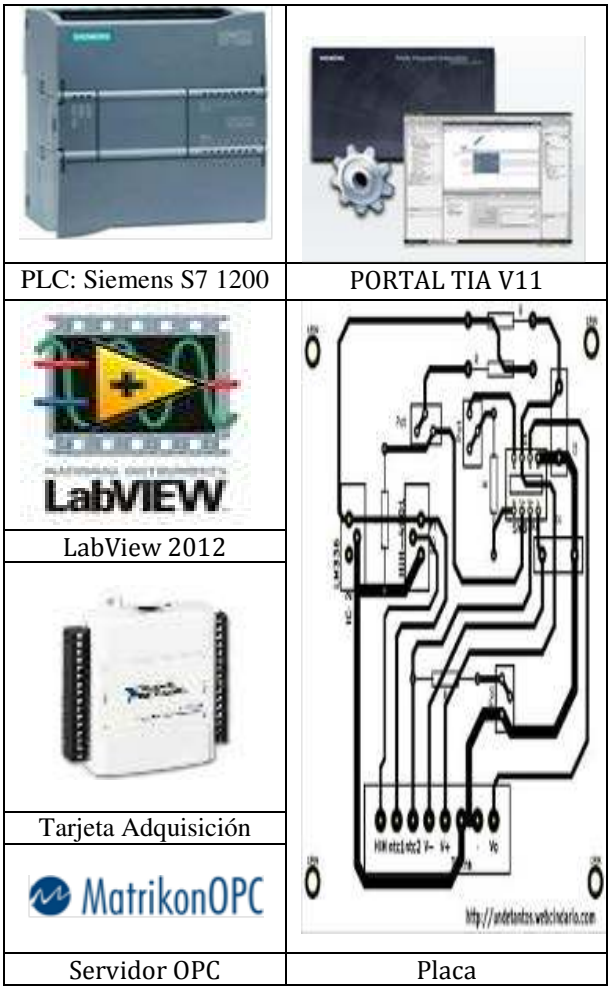


Figura 2. Elementos utilizados.

2.1- TARJETA DE ADQUISICIÓN DE DATOS

Llamaremos sistema de adquisición de datos al hardware que permite capturar y convertir muestras

de las señales medidas para ser tratadas por un procesador digital

Los elementos que forman la cadena de adquisición de datos de una señal son:

- Amplificador de ganancia programable: su misión es adecuar el rango de variación de la señal analógica al margen dinámico del convertidor.
- Circuito de muestreo y retención: su misión es mantener constante el valor adquirido durante el tiempo de conversión.
- Convertidor A/D: su misión es convertir la muestra analógica a formato digital.
- Interface con el procesador digital: su misión es servir de puente entre el convertidor A/D y el procesador.
- Un decodificador de direcciones
- Buffer triestado en la salida

Entre los parámetros a elegir en una adquisición de datos es primordial el número de bits del sistema de adquisición de estos, de él dependerá una buena o mala resolución de los parámetros a medir. La "resolución" de un dispositivo es el mínimo incremento de la entrada que ofrece un cambio medible en la salida. Se suele expresar como un valor en tanto por ciento sobre todo a fondo de escala. Cuando el incremento de la entrada se produce a partir de cero, se habla de "umbral". Sin embargo, en tarjetas de adquisición de datos la resolución suele expresarse como el número de bits del conversor A/D. Como vamos a tener datos de 12 bits de resolución para la configuración diferencial, por ello, se podría considerar que este conversor A/D podría representar 4096 combinaciones binarias. Sin embargo, para la configuración en modo unipolar la tarjeta emplea 11 bits de resolución con lo que podría representar 2048 combinaciones binarias. Con estos datos obtenemos el error teórico de la tarjeta:

Para esta aplicación los parámetros medidos serán la temperatura y humedad. El acondicionamiento de los parámetros a medir determinará una buena medida. El sistema implementado es analógico con componentes 'trough hole device' (THD), pudiendo este ser sustituido directamente en caso de averías.

El equipo de adquisición de datos utilizado y elegido para la aplicación ha sido la DAQ USB 6008.

2.2- CONCEPTO DE SCADA

En éstas últimas décadas se han desarrollado sistemas denominados SCADA, interactúan directamente con el operario a través de una pantalla de ordenador, capaces de controlar distintos hardware de control como los PLC. Actualmente el control automático de cualquier sistema de producción, supervisión o fabricación lleva un sistema SCADA. Utilizándose elementos como autómatas, ordenadores industriales, robots, etc... En este tipo de sistemas, el operario podrá supervisar en tiempo real el control de la planta detectando las alarmas a través del panel, el cual, podrá determinar causas del problema y poder así tratar de solucionarlos en tiempo real.

Un sistema SCADA es una aplicación de software especialmente diseñada para funcionar sobre ordenadores en el control de producción que proporciona comunicación entre los distintos dispositivos de campo, llamados también RTU (Remote Terminal Units o Unidades Remotas), donde se pueden encontrar elementos tales como controladores autónomos o autómatas programables, y un centro de control o Unidad Central (MTU,Master Terminal Unit), donde se controla el proceso de forma automática desde la pantalla de uno o varios ordenadores.

Un sistema SCADA está divido en dos grandes bloques:

- Captadores de datos: recopilan los datos de los elementos de control del sistema (por ejemplo, autómatas, reguladores, registradores) y los procesan para su utilización. Son los servidores del sistema.
- Utilizadores de datos: los que utilizan la información recogida por los anteriores, como pueden ser las herramientas de análisis de datos o los operadores del sistema. Son los clientes.

La Unidad Central (MTU, Master Terminal Unit) centraliza el mando del sistema, en él se realiza principalmente la tarea de recopilación y archivado de datos. Encargándose de gestionar las comunicaciones, recopilar los datos de todas las estaciones remotas (RTU), envío de información, comunicación con los operadores.

La Unidad Remota (RTU, Remote Terminal Unit) es aquel conjunto de elementos dedicados a labores de control y/o supervisión de un sistema, alejados del centro del control, comunicados con éste mediante algún canal de comunicación. Las unidades remotas se encargaban en un principio de recopilar los datos de los elementos de campo (Autómatas reguladores) y transmitidos hacia la Unidad Central, a la vez que enviar los comandos de control a éstos. Hoy día, una estación remota no es necesariamente un autómata con capacidades de comunicación. Se puede tratar de un sistema que forme parte de otro sistema de control donde a su vez controla otro sistema.

Los objetivos para que la instalación de un sistema SCADA sea óptima, entre otros son:

- Funcionalidad completa de manejo y visualización en el sistema operativo, sobre cualquier PC estándar.

- Permitir combinaciones con aplicaciones estándar y de usuario, que permitan a los integradores crear soluciones de mando y supervisión (Active X, OPC, OLE-DB, VB o C,…).
- Permitir la integración con las herramientas ofimáticas y de producción.
- Que sea fácilmente configurable y escalable, debe ser capaz de crecer o adaptarse según las necesidades cambiantes.
- Comunicaciones flexibles para poder comunicarse con total facilidad y de forma transparente al usuario con el equipo de planta y con el resto de la empresa.

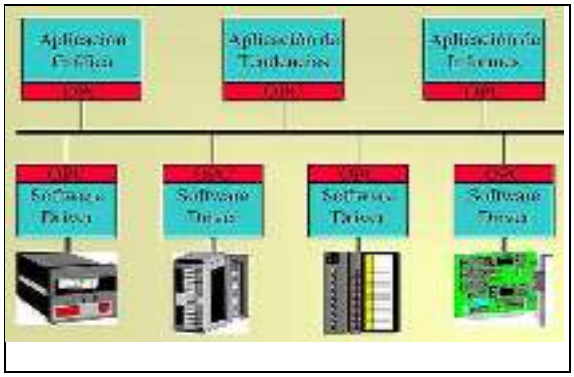


Figura 3. Ejemplo aplicaciones de sistemas OPC

2.3- EL OPC

Para poder interactuar entre el PLC y el software de control, normalmente es necesario proporcionar un software "intermedio", el cual hará de interlocutor entre las distintas variables utilizadas por parte del PLC. En nuestro caso el elegido ha sido el OPC Server de Matrikon.

Este tipo de software llamado OPC (OLE for Process Control) es una aplicación de software (driver) que cumple con una o más especificaciones definidas por la OPC Foundation. El Servidor OPC hace de interfaz comunicando por un lado con una o más fuentes de datos utilizando sus protocolo nativos (típicamente PLCs, DCSs, básculas, Modulos I/O, controladores, etc.) y por el otro lado con Clientes OPC (típicamente SCADAs, HMIs, generadores de informes, generadores de gráficos, aplicaciones de cálculos, etc.). En una arquitectura Cliente OPC/ Servidor OPC, el Servidor OPC es el esclavo mientras que el Cliente OPC es el maestro. Las comunicaciones entre el Cliente OPC y el Servidor OPC son bidireccionales, lo que significa que los Clientes pueden leer y escribir en los dispositivos a través del Servidor OPC.

A pesar de que OPC está diseñado principalmente para acceder a los datos de un servidor de red, los interfaces OPC pueden utilizarse en muchos lugares dentro de una aplicación. En el nivel más bajo que pueden conseguir los datos en bruto de los dispositivos físicos en un sistema SCADA o DCS, o desde el sistema SCADA o DCS en la aplicación. La arquitectura y el diseño hacen que sea posible construir un servidor OPC que permite que una aplicación cliente pueda acceder a los datos de muchos servidores OPC proporcionados por diferentes proveedores de OPC que se ejecutan en los diferentes nodos a través de un único objeto.

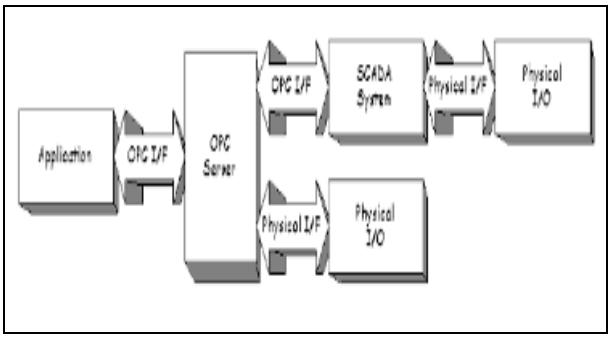


Figura 4: Funcionamiento del OPC

Un servidor OPC se compone de varios elementos, entre ellos: el servidor, el grupo, y el item. El objeto o item del servidor OPC mantiene información sobre el servidor y sirve como un contenedor de grupos de objetos OPC. El grupo de objetos OPC mantiene información sobre si mismo y proporciona el mecanismo para contener y organizar lógicamente los elementos OPC

Los grupos OPC proporcionan una forma para que los clientes organicen los datos. Por ejemplo, el grupo podría representar los elementos de una pantalla de operador en particular o un informe. Los datos se pueden leer y escribir. Los items OPC representan conexiones a fuentes de datos en el servidor. Un item OPC, desde la perspectiva de interfaz personalizada, no es accesible como un objeto por un cliente OPC.

2.4- AUTOMATA S7-1200

El PLC utilizado es el SIEMENS S7-1200 con la CPU 1211 C, aunque tiene dos entradas analógicas estas son de 10 bits (perdiendo resolución en la medida), siendo esta la más compacta de su gama. Ofrece la flexibilidad y potencia necesarias para controlar una gran variedad de dispositivos para las distintas necesidades de automatización.

Gracias a su diseño compacto, configuración flexible y amplio juego de instrucciones, el S7-1200 es idóneo para controlar una gran variedad de aplicaciones. La CPU incorpora un microprocesador, una fuente de alimentación integrada, circuitos de entrada y salida, PROFINET integrado, E/S de control de movimiento de alta velocidad y entradas

analógicas incorporadas, todo ello en una carcasa compacta, conformando así un potente controlador. Una vez cargado el programa en la CPU, ésta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario, que puede incluir lógica booleana, instrucciones de contaje y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes.

2.5- ARQUITECTURA EMPLEADA

La arquitectura de control desde el punto de vista de mi aplicación, se trata de una interconexión de distintos elementos empleados para llevar a cabo el control del sistema. Es de tipo "heterogénea" con morfología "descentralizada".

- Heterogénea: Emplea equipos de distinto tipo e incluso de distinto fabricante. Su principal desventaja es la dificultad de interconexión, su ventaja es la libertad de no estar atado o ceñido a un sólo fabricante.
- Descentralizada: en caso de que las señales no estén próximas al equipo de control (ejemplo una vivienda con varios pisos o líneas de producción) algunos fabricantes ofrecen módulos E/S sin que tengan porque estar físicamente unidos al PLC, sino que es posible alargar el bus del PLC unos cuantos metros.

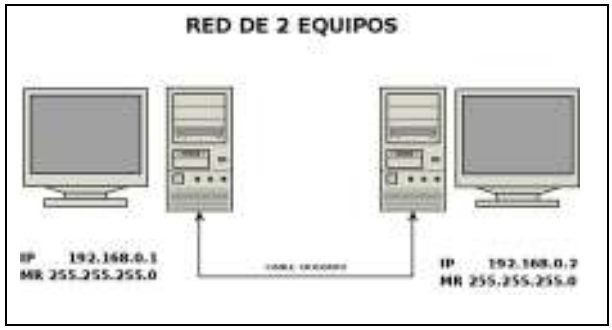


Figura 5: Red formada por los equipos, "punto a punto"

En este trabajo se ha utilizado una topología de red "punto a punto", utilizando una red LAN.

3 DISEÑO DE LAS PLACAS.

Las placas han sido diseñadas por nosotros mismos con el software gratuito KIDCA [XXX], obteniendo por tanto las siguientes ventajas:

- Permiten la producción automática.
- Reduce la mano de obra y por tanto el coste.
- Permite un montaje más compacto, reduciendo considerablemente el espacio y el peso.
- Su exacta reproductibilidad evita errores de cableado.
- Reduce los efectos parásitos inductivos o capacitivos del cableado.
- Permite una sustitución rápida de componentes.

El diseño del circuito impreso consiste en transformar el esquema eléctrico del circuito teórico en un plano real que incluya la colocación y posición de los componentes, el trazado de todas las pistas de conexión eléctrica con sus puntos de contacto e interconexión, las entradas y salidas del circuito, etc. Debido a que los circuitos que he diseñado no requieren grandes prestaciones y además es meramente un prototipo, he escogido baquelita.

Para obtener la anchura de las pistas se ha utilizado la aplicación "Pcb Calculator" que proporciona el propio KICAD. Esta pequeña aplicación aplica la norma internacional para el cálculo de pistas "IPC 2221".

3.1- EQUIPOS DISEÑADOS

Entre los equipos que ha sido necesario diseñar nos encontramos con el sistema de sensorización de temperatura y humedad, el circuito de luminosidad, una fuente de alimentación, y el control PWM [XX]

El sistema de sensorización de temperatura y humedad, está constituido por el acondicionador de la señal de temperatura, adquirida por un termistor NTC y el HIH 4000 que mide la humedad. El rango para el que está calculado el acondicionador de temperatura es de [-10,50]ºC y el sensor de humedad mide para una humedad entre [0,100]HR, dando una tensión variable de [1,4]V. Este es alimentado por una tensión de 5 V,

Una vez escogido el sensor tendremos que linealizarlo en función de la temperatura mediante una gráfica genérica, posteriormente después de diseñar el circuito linealizador (entorno a un punto de funcionamiento) calcularemos un 'sistema de amplificación' que estará ajustado a la señal de entrada del DAQ 6008.

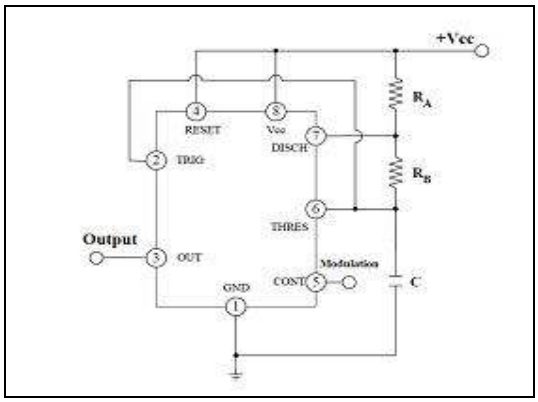


Figura 6: LM 555 conectado como ASTABLE

El circuito de luminosidad será el que controle la luminosidad de la bombilla, este lleva también una fuente de alimentación en la propia placa, ya que la requieren los circuitos asociados. Está basado en dos elementos, el LM 555 y el mosfet de potencia IRF 540. Surgió de la necesidad de controlar una bombilla halógena con una tensión de 0V a 5 V, procedente de la tarjeta de adquisición de datos, aunque al final por precaución es de 0V a 4V. Aprovechando un material sobrante de un laboratorio, en que se disponía de un transformador, unos condensadores, un puente y diversos materiales, entre ellos el LM555 y la bombilla de 200W halógena que trabaja con 12V en alterna, por lo que la intensidad máxima que consume es 1.66A. Para esta intensidad necesitamos entre otros dispositivos el transformador que lo suministre, como es que tenemos es de 3 A, lo suministra de sobra incluso podríamos llegar a controlar dos lámparas en paralelo

Este circuito se fundamenta en el LM 555 y está compuesto por una serie de bloques internos, entre los que constan, dos comparadores de tensión, un flip-flop RS, tres resistencias de 5KΩ conectadas en serie y un transistor entre otros. Puede configurarse de dos maneras, como monoestable (la frecuencia de salida es contante) o como 'astable' (la frecuencia de salida podrá variarse). Este es nuestro caso ya que, según la relación cíclica haremos variar el valor eficaz de la puerta del mosfet.

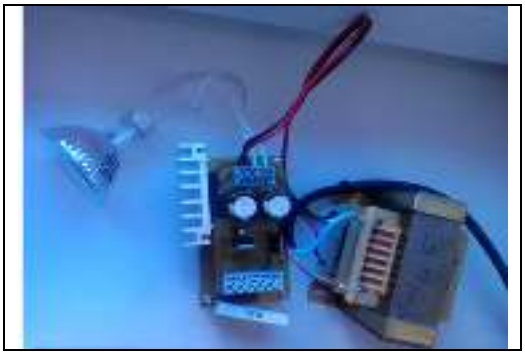


Figura 7. Circuito de control, bombilla y fuente.

La fuente de alimentación dual de [-15,+15] V, que alimenta al sistema de sensorización, ya que, el amplificador de instrumentación es alimentado con dicha tensión. La carga de la fuente de alimentación no tendrá un consumo elevado por ello nos valdrá cualquier transformador comercial. Escogeremos un transformador de 220/24, dual '24+24' para así hacerla simétrica. Con un puente de diodos comercial.

5 PROGRAMACIÓN DE LA APLICACIÓN

Para lograr la programación completa se han tenido que realizar programaciones parciales de los distintos equipos, así como la parametrización de todas las comunicaciones. A continuación pasamos a realizar un breve repaso por cada una de ellas.

5.1- PROGRAMACIÓN AUTÓMATA, SCADA, OPC

Para la programación de la aplicación se ha utilizado el programa de Siemens TIA PORTAL V11.0. Éste se utiliza para introducir un programa simple al autómata, aunque es simplemente activar la variable interna del autómata, también podríamos haber realizado un programa más complejo y con ello sacarle más partido a la programación y al autómata.



Figura 7: Programa en contactos

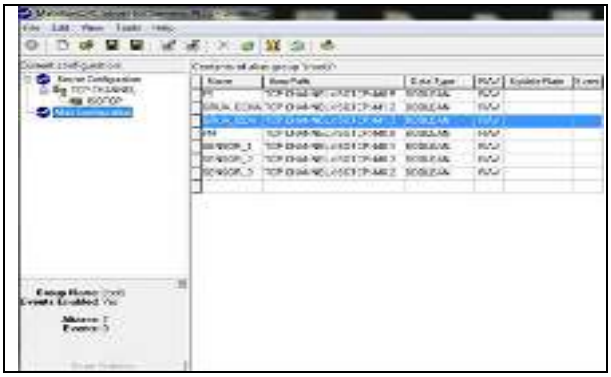


Figura 8: Enlace de Tags

El método escogido para relacionar el Labview y el autómata, como he comentado con anterioridad es mediante el OPC SERVER de MATRIKON. En el entorno gráfico se puede observar la configuración del OPC, en el que deberemos configurar con anterioridad el canal, slot, el tipo de conexión o protocolo de comunicación, además de introducir la IP del autómata y el puerto. Con todo ello quedará bien configurada la conexión. También se deben relacionar los 'Tags' o 'Aliasis', que serán las variables del autómata. Para la programación del SCADA se ha utilizado el software Labview tal y como se muestra en la figura 9 con diversas funciones propias para la aplicación:

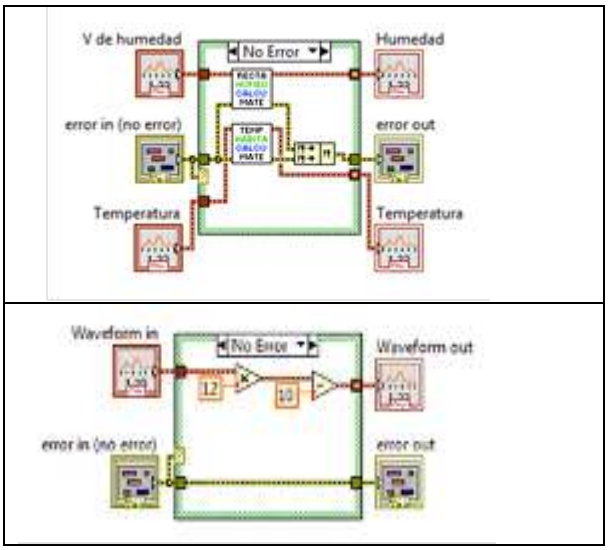


Figura 9: Sensores de Temperatura y Humedad

5.2- CONTROL DE LA CLIMATIZACIÓN

Todas las imágenes del SCADA se deben realizar en lo que Labview denomina 'Block Diagram', es la programación de la aplicación.

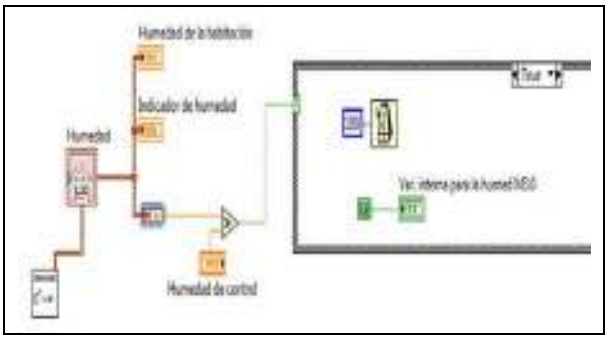


Figura 10: Gestión Programa

En la figura 8, se muestra el programa que gestiona el control de la humedad del habitáculo. Utiliza una estructura "Case", cuando la humedad del lugar a medir supera la humedad de control, se activará un valor "True", que activará la variable interna del autómata activando así el circuito correspondiente para su control. Por contra, cuando esta humedad a medir es menor que la humedad de control, se activará un "False" a su control, y mediante el uso de variables locales será activada la variable booleana, desactivando la acción del autómata.

Otro elemento que también debemos tener en cuenta en el programa de desarrollo Labview, es el 'Front Panel'. En él, visualizaremos la aplicación y cómo el usuario la gestionará, de modo que esta debe ser lo más intuitiva posible para el mismo.

En la figura 11 se refleja el indicador de humedad, la variable de consigna o de control, el indicador gráfico de dicha humedad en el habitáculo que es lo que refleja el programa explicado con anterioridad, el día y la hora de dicha adquisición además de las indicaciones de la activación o desactivación de las variables internas del autómata.

Existe también una función que muestra la gestión y control de la temperatura del lugar, si esta fuese mayor que la temperatura de control, se activará la constante "True" que dará el valor a la variable local que activa la variable interna del autómata. Por contra, si esta es menor que la temperatura de control, la constante "False" desactivará la variable interna del autómata.



Figura 11: Indicadores de Humedad y Temperatura

En el panel también observamos el indicador de temperatura, tanto gráfico como digital además de la propia consigna o variable de control. A mayores esta imagen muestra en donde guardaremos los datos adquiridos, el dial de tensión de control con el que variamos la luminosidad de la lámpara de manera manual ('Elección=Manual'), si 'Elección' estuviese en otra selección esta podría ver que estaría en función de un porcentaje, 25%, 50%, 75% o Máximo entre otras (indican la variación del consumo de la bombilla) y por último 'Tipo de gráfica a visualizar', desplegando el menú obtendremos qué gráfica visualizará la aplicación, teniendo un control absoluto sobre la entrada a visualizar.

Gracias al software de control, se hace fácil implementar una página web a través de distintas funciones utilizadas para ello, por eso deberemos indicar también porque puerto se va a conectar la aplicación.

Se utiliza como servidor el propio ordenador de la aplicación, siendo la dirección propia del equipo la utilizada para 'colgar' la página implementada por Labview para la conexión en 'red local', sin

embargo, para ver la aplicación por Internet y poderla controlar deberemos tener en consideración lo siguiente (http://[IP dinámica pública]: 9080/ proyecto_i.html):

- Tener instalado el "Run Time Engine", tanto en el equipo local como remoto. Este es proporcionado por National Instruments de manera gratuita.
- La aplicación sólo podrá ser controlada por un solo administrador.

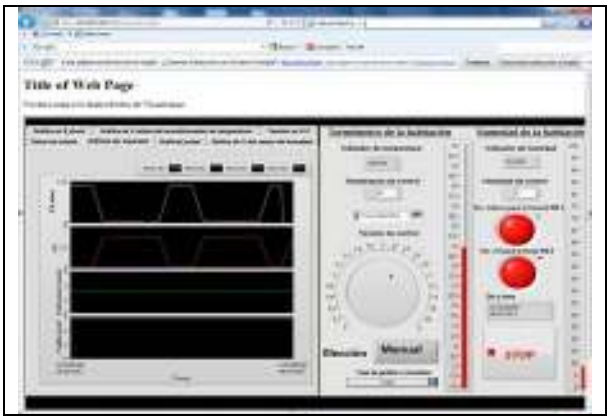


Figura 10: Pagina Web

6 CONCLUSIONES Y LINEAS FUTURAS.

En cuanto a las conclusiones que podemos extraer del presente trabajo y teniendo que ha sido un trabajo multidisciplinar tenemos:

1) Se ha realizado un estudio pormenorizado de los todos los elementos necesarios para el desarrollo de la aplicación tanto a nivel de hardware, de software como de comunicaciones.
2) Se han acondicionado y parametrizado distintas magnitudes físicas, como son la temperatura y humedad. Con circuitos de acondicionamiento se puede trasladar magnitudes físicas a variables como tensión o intensidad midiéndolas a través de tarjetas de adquisición de datos.
3) Se han diseñado, con su correspondiente linealización, circuitos con KICAD. Es un software gratuito, pero bastante potente.
4) Se ha manejado el TIA PORTAL, el Labview y del OPC MATRIKON. Estas herramientas están en continuo desarrollo y creo que es conveniente tenerlas presentes para cualquier aplicación industrial tales como los sistemas SCADA.
5) Se ha controlado y visualizado mediante una web, el control de la aplicación por internet y red local. Con la herramienta para ello, proporcionada desde Labview, no hace falta instalar un servidor como podría ser un servidor como Apache.
6) Se ha realizado la documentación necesaria para otros usuarios. Al realizar la guía de usuario, se debe proponer una mejor compresión cara a su utilización.

En cuanto a las líneas futuras de trabajo, que ya están fijadas, y en algunos casos se están realizando avances interesantes son:

1) Tal vez, un uso muy interesante sería la implementación de una programación en el PLC de cualquier tipo de automatización con lo que la aplicación sería un SCADA a gran escala.
2) La aplicación de utilizar distintas aplicaciones de compartir datos en red con otros ordenadores tanto de dentro de la misma red como de otra.
3) Al mismo tiempo, sería interesante realizar una conexión sin uso del OPC, así evitarías el estar anclado a este tipo de driver y ninguna marca, es decir, desarrollar una conexión desde Labview directa al PLC.

Referencias

[1] Coughlin, R., and F. Driscoll "Amplificadores operacionales y circuitos integrados lineales"; Ed. Prentice Hall, ISBN: 968-880-284-0 (1993)

[2] Lájara Vizcaíno, J. R. and J. Pellegrí Sebastiá "Labview: Entorno gráfico de programación." 2ª Edición Editorial Marcombo ISBN: 978-8426716965 (2011)

[3] Marston, R., "Newnes Electronics Circuits Pocket Book" 2ª Edición; Editorial Butterworth-Heinemann, ISBN: 978-0750630183. (2000)

[4] Ogata, K., (2003) "Ingeniería de Control Moderna", 4ª Edición. Ed. Prentice Hall, Upper Saddle River, ISBN: 84-205-3678-4

[5] Pallás Areny, R. "Sensores y acondicionadores de señal". 4ª Edición, Editorial Marcombo ISBN: 978-8426713445 (2005)

[6] SIMATIC S7-1200 Documentation. s71200_System_Manual_es-ES.pdf

[7] http://iut-tice.ujf-grenoble.fr/kicad/

[8] http://www.buenastareas.com/ensayos/Tutorial-Sensor-De-Humedad/2701366.html

[9] http://www.matrikonopc.es/opc-servidor/index.aspx

[10] http://www.ni.com/labview

[11] http://www.ni.com/support/esa/

[12] http://www.ni.com/white-paper/7906/es; así como el /5407/es, el /6417/en y el /4791/en

SISTEMAS DE CONTROL POR COMPUTADOR UNIFICACIÓN DE CRITERIOS

José Luis Casteleiro-Roca
Dpto. de Ingeniería Industrial, Universidad de La Coruña
jose.luis.casteleiro@udc.es

José Luis Calvo-Rolle
Dpto. de Ingeniería Industrial, Universidad de La Coruña
jlcalvo@udc.es

Ramón Ferreiro García
Dpto. de Ingeniería Industrial, Universidad de La Coruña
ferreiro@udc.es

Resumen

A la hora de estudiar los sistemas de control, es frecuente iniciar el proceso con sistemas continuos; más extendidos y sencillos de comprender para los estudiantes. Al abordar un sistema discreto, la mayor parte de las veces el proceso práctico es discretizar un sistema continuo. En ese proceso es frecuente que los estudiantes se vean "perdidos" al no comprender de antemano donde está la separación entre el sistema discreto y el continuo.
En las aulas a menudo se discretiza toda una planta teórica, y se implementa un control discreto de la misma. Sin embargo, cuando este mismo proceso se realiza de manera práctica, hay que tener bien definidas las diferentes partes del sistema para comprender "donde" empieza el sistema continuo, y "donde" el discreto.
En este trabajo se presenta un esquema para el control por computador. Se pretende unificar criterios y explicar, mediante ejemplos, las funciones concretas de cada parte del sistema de control. Las explicaciones estarán basadas en las plantas disponibles en el Laboratorio de Automática de la Escuela Universitaria Politécnica de Ferrol, y en los sistemas de control implementados en ellas.

Palabras Clave: Sistemas Lineales, Sistemas Discretos, Control por Computador, Prácticas de Laboratorio.

1 INTRODUCCIÓN

Los métodos de control de sistemas han ido variando a lo largo del tiempo debido, entre otros, al abaratamiento de los componentes electrónicos, especialmente en el caso de los componentes digitales. En un inicio el control digital (obviando ya los métodos de control analógicos) se realizaba mediante sistemas de microprocesadores y microcontroladores, diseñados *ex profeso* para el sistema a controlar. Éstos, en ciertas aplicaciones industriales, están dejando (o han dejado) paso a los autómatas programables (PLC), los cuales tienen una versatilidad y compatibilidad mucho mayor, además de ser más fáciles de programar y no tener que ser diseñados para cada sistema.

Los PLCs, pese a ser unos sistemas más cercanos que los sistemas de microprocesador, siguen contando con el gran inconveniente de no tener un poder computacional aceptable. Es por este motivo por el que se está progresando hacia el control por computador.

El control por computador sigue cobrando gran importancia frente a los autómatas, debido a razones como el gran y constante abaratamiento que han sufrido los ordenadores en los últimos años, poseer un poder computacional muy elevado, así como una gran variedad de lenguajes de programación. Es necesario el uso de Tarjetas de Adquisición de Datos (DAQ), como interfaz de entrada/salida, ya sean analógicas o digitales.

Centrándonos en los lenguajes de programación, existen una serie de lenguajes y entornos de programación como podrían ser LabView® o Simulink® que son de una extrema sencillez en programación y no necesitan el uso de código propiamente dicho para la implementación de controladores. Estos lenguajes presentan un claro inconveniente, y es que tienen una deficiencia en cuanto a funciones y posibilidades, pudiendo llegar a resultar más difícil la programación mediante estos entornos que usando otros lenguajes como pudiera hacerse en MatLab®.

MatLab® posee una clara ventaja sobre el resto, se trata de un lenguaje muy potente y con una serie de

toolboxes (herramientas y funciones para la programación) específicas para un gran número de tareas, incluido el control por computador.

El presente trabajo se organiza de la siguiente manera. Después de esta introducción, se expondrán los sistemas de control clásicos, tanto los lineales como los discretos. Seguidamente se presentará el esquema de control sugerido para el control por computador. Se expondrán posteriormente varios ejemplos de plantas de laboratorio desde el punto de vista del esquema presentado; y se finalizará el trabajo presentando las conclusiones del mismo.

2 SISTEMAS DE CONTROL CLÁSICOS

Como sistema clásico de control, definiremos aquellos sistemas que siguen el esquema de control mostrado en la figura 1 [4].

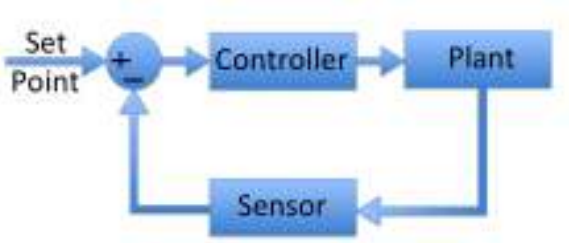


Figura 1 – Esquema clásico de control

En este tipo de sistemas, se pretende que la salida de la planta sea lo más fiel posible a la entrada deseada. Para ello, se suele introducir un regulador en el sistema que recibe como entrada la señal de error entre la salida deseada y la salida real.

2.1 SISTEMAS CONTINUOS

Por sistema continuo se entiende aquel cuyas variables son continuas en todo su rango de funcionamiento [4]. El esquema de un sistema de control continuo se puede ver en la figura 2, que es similar a la explicada en el apartado anterior, con la inclusión de las señales a lo largo del bucle de control.

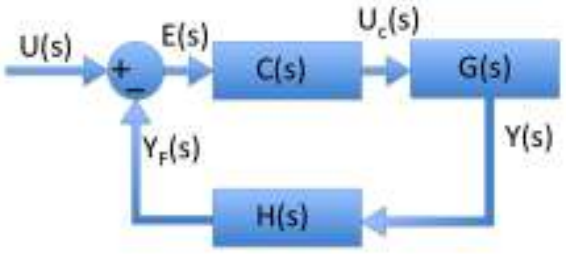


Figura 2 – Bucle de control para un sistema lineal

Al hablar de sistemas continuos, el tratamiento de señales se suele realizar utilizando la transformada de Laplace [3].

Así, la señal de entrada *u(t)*, sería la señal *U(s)*; e igual que esa, ocurriría con el resto. A la hora de tratar el sistema completo, entre la entrada *U(s)* y la salida *Y(s)*, se puede ir operando en el esquema hasta llegar a la ecuación 1.

$$M(s)=\frac{Y(s)}{U(s)}=\frac{C(s)\cdot G(s)}{1+C(s)\cdot H(s)\cdot G(s)} \qquad (1)$$

Existen multitud de herramientas disponibles para el cálculo de un controlador idóneo para un sistema una vez que se conozca la función de transferencia de la planta *G(s)*. El problema en un sistema real, es que esa función de transferencia no es siempre conocida; y aunque se conozcan perfectamente todas las partes del sistema, muchas veces tampoco es factible el calcularla.

En los sistemas reales, el cálculo de estas funciones de transferencias se suele hacer por métodos computacionales, para posteriormente obtener un regulador adecuado al sistema. El problema de los sistemas computacionales es que no son sistemas continuos, sino que se trata de sistemas discretos.

2.2 SISTEMAS DISCRETOS

Un sistema discreto es aquel que tiene señales digitales en alguna de las variables. En la figura 3 puede apreciarse visualmente la diferencia entre una señal continua (línea azul-continua), y una señal discreta (línea roja-a puntos).

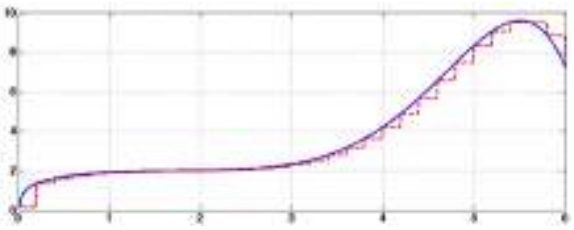


Figura 3 – Señal continua y discreta

A la hora de estudiar un sistema discreto, el procedimiento usual suele ser el de analizar el sistema lineal equivalente, pero aplicando la transformada *Z* a las señales [2]. El resultado de ese proceso puede apreciarse en la figura 4, donde se muestra el mismo sistema lineal de la figura 2, pero con las señales discretizadas.

A la hora de estudiar estos sistemas, la aproximación discreta suele dar buenos resultados. Pueden hacerse las mismas operaciones que para los sistemas lineales; identificar la función de transferencia del

sistema, calcular un regulador, e implementar el control del sistema discreto.

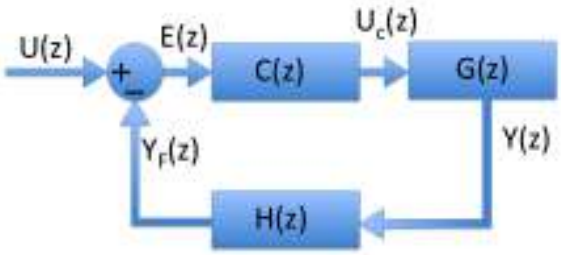


Figura 4 – Bucle de control para un sistema discreto

Para el control de los sistemas discretos, se suelen utilizar sistemas basado en microcontroladores, o en microprocesadores. Las complicaciones suelen aparecer cuando el control discreto se realiza para un sistema continuo. En este caso, hay que diferenciar perfectamente las partes continuas de las discretas, no tanto para que el control sea efectivo, sino para la mejor comprensión de todo el sistema.

3 SISTEMAS DE CONTROL POR COMPUTADOR

Como se ha expuesto en los apartados anteriores, el esquema presentado en este trabajo trata de unificar los criterios para la definición de un sistema de control discreto para una planta continua. Este tipo de sistemas se engloban dentro de los sistemas de control por computador, aunque no son los únicos.

Por control por computador se entiende cualquier tipo de control donde se use un ordenador para realizar los cálculos llevados a cabo por el regulador. No es necesario que el ordenador haga las funciones de regulador, sino que bastaría con que el regulador hiciese uso en un ordenador para variar sus parámetros (por ejemplo), para englobar dicho sistema en el control por computador.

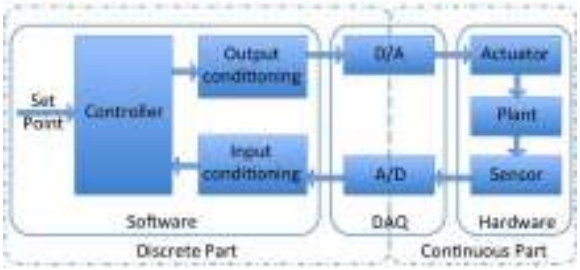


Figura 5 – Bucle de un sistema controlado por computador

El esquema general de un bucle completo de control como el expuesto puede verse en la figura 5. En esta figura se pueden apreciar dos grandes partes, la discreta y la continua; así como los tres sub-bloques principales (*Software*, *DAQ*, *Hardware*).

A continuación se presentarán a grandes rasgos las funciones de cada una de las partes presentadas en este esquema y, posteriormente, se fijarán estos conceptos a través de los ejemplos concretos en las plantas de laboratorio.

3.1 PARTES DISCRETAS DEL SISTEMA

Como partes discretas del sistema, entenderemos aquellas que están internas al ordenador; son las secciones del sistema que más fácilmente se pueden modificar, y podemos hablar principalmente de 3. Dependiendo del sistema, podrían descomponerse en más bloques, o combinar alguno de ellos si son excesivamente sencillos o no necesarios.

3.1.1 Controlador

En los sistemas de control expuestos hasta ahora, el controlador recibía como entrada la diferencia entre la señal de control y la variable medida del sistema (el error del sistema). Por variable medida del sistema se entiende la realimentación que recibe el controlador para conocer el estado de la planta.

Los sistemas reales de control siempre tienen realimentación; en teoría existe otro grupo de sistemas sin realimentación, pero no es normal encontrarse con estos sistemas. La existencia de realimentación produce un efecto directo de mejora en la estabilidad del sistema.

En un sistema controlado por computador, lo normal no es tener una única entrada en el controlador. Esta topología suele remontarse al diseño de los primeros controladores como puede ser el control PID. Sin embargo, aún en este tipo de reguladores, hay algunos que parte de las acciones de control no se realizan sobre la señal de error, sino sobre la propia realimentación. En concreto pueden encontrarse reguladores PID donde la acción proporcional e integral se realizan sobre el error, mientras que la acción derivativa se realiza directamente sobre la realimentación del sistema [1].

Este es el motivo por el que se han expuesto las dos entradas en la figura 5, sin limitarse únicamente al error.

La topología que podemos encontrar para los sistemas de control por computador es tan extensa como se pueda imaginar. Existen desde reguladores clásicos tipo PID (trabajando como reguladores continuos o discretos), hasta los más complejos reguladores inteligentes que combinen (por ejemplo),

algoritmos genéticos con lógica *Fuzzy* de control avanzado.

3.1.2 Escalado/acondicionamiento de entrada

Las señales de entrada al controlador serían capturas por la Tarjeta de Adquisición de Datos. La naturaleza de los sistemas puede ser tan variada, que las variables que queramos procesar podrían ser: longitud, presión, temperatura, presión, viscosidad… Existen en el mercado multitud de DAQs, pero los más extendidos son aquellos que utilizan tensión como magnitud de entrada y salida.

Dicha tensión, como se ha mencionado, puede referirse a cualquier otro tipo de variable, y en el bloque de escalado/acondicionamiento de entrada, se realizaría la conversión entre la tensión y rango de entrada del DAQ, a la magnitud y rango real del sistema.

Un ejemplo: si queremos medir temperatura, podemos usar un sensor LM35, cuyo rango de funcionamiento es de 0 a 100 ºC, y su salida es de 0 a 1 V para el rango mencionado (con salida lineal en todo el rango). Si conectásemos una tarjeta de adquisición de datos directamente a la salida de ese sensor, la tarjeta podría darnos un valor capturado de 0.25 V. Para que el sistema operase con la temperatura real medida, habría que acondicionar la señal capturada según la ecuación 2, que daría una temperatura de 25 ºC.

$$T_{Real} = V_{Capturada} \cdot 100 \tag{2}$$

Además de realizar este "escalado" simple, este bloque se suele utilizar para realizar todos los acondicionamientos necesarios en la señal. Si se quisiera implementar un filtrado en la señal del sensor de temperatura mencionado (por ejemplo), podría introducirse un filtro *software* en este bloque que eliminase el ruido en la señal.

También cabe destacar que en el ejemplo expuesto, el escalado es lineal debido a que la salida del sensor expuesto es lineal. De no serlo, en este mismo bloque se podría implementar un algoritmo concreto que permitiese linealizar la señal y así facilitar el control del sistema.

3.1.3 Escalado/acondicionamiento de salida

Como se mencionó en el apartado anterior, lo normal es que la tarjeta de adquisición de datos trabaje con tensiones, con lo que habría que variar la magnitud de la salida del controlador a un rango válido para el DAQ.

En este módulo, al igual que en el anterior, este acondicionamiento no tiene porqué ser simple. Si por ejemplo el actuador de la planta a controlar no fuese lineal, podría interesar linealizar el mismo en este bloque.

Con esta linealización nos solemos referir a que para que una variación en la salida del controlador, se produzca la misma variación en la señal de entrada a la planta. Si el controlador comandase una variación de 5 unidades en la entrada de la planta, el efecto en la misma fuese el mismo en todo el rango de funcionamiento (de 5 a 10 igual que de 57 a 62).

Este efecto puede apreciarse mejor en la sección de ejemplos, ya que una de las plantas de laboratorio disponibles tiene esta propiedad implementada a propósito para explicar esta posibilidad.

3.2 TARJETA DE ADQUISICIÓN DE DATOS

Las Tarjetas de Adquisición de Datos son los dispositivos usados para la comunicación entre el ordenador y la planta real a controlar. Se pueden definir como los dispositivos que permiten a un ordenador interactuar con el mundo real.

Existen múltiples DAQs, cada una con sus propias características de rangos de funcionamiento. Lo más importante a tener en cuenta es la configuración de las entradas y salidas, y los rangos de funcionamiento que tenga el DAQ elegido para cada aplicación.

En una aplicación concreta es de vital importancia conocer los rangos de los canales de entrada y de salida. Se debería programar correctamente los bloques de acondicionamiento según estos rangos, y se aconseja que en el acondicionamiento se incorpore una protección que evite salirse de dicho rango para evitar posibles fallos en el DAQ.

Dependiendo de la aplicación, uno de los datos a tener en cuenta también es la velocidad de adquisición y generación del DAQ que tengamos. Además, esta velocidad no siempre será la teórica proporcionada por el fabricante; a pesar de que la tarjeta pueda adquirir señales a la velocidad deseada, el bucle de control puede ser excesivamente complejo, y bajar por este motivo la máxima frecuencia de adquisición. Se recomendaría estudiar este fenómeno para aplicaciones de alta velocidad, ya que se han encontrado fallos en el control de sistemas reales debidos a este motivo.

3.3 PARTES CONTINUAS DEL SISTEMA

Las partes continuas del sistemas las definimos como los bloques que no forman parte del ordenador ni del

DAQ. Muchas veces estas partes no son sencillas de diferenciar entre ellas, por ese motivo se expondrán a continuación una definición estándar para cada bloque.

3.3.1 Planta

La planta es el sistema físico indivisible que se pretende controlar. Se remarca la característica de indivisibilidad para indicar que las partes sueltas del sistema, no forman parte de ella.

Por ejemplo, un horno didáctico es un conjunto formado por una caja de metacrilato, con dos bombillas como dispositivo de calentamiento, y un sensor de temperatura para saber el estado del sistema. Las bombillas y el sensor se podrían separar del sistema, pero si se quitase el sensor, ya no se dispondría de realimentación, con lo que el sistema variaría; o si se quitasen las bombillas, el horno no funcionaría, o al menos no de igual manera, con lo que la planta no sería la misma.

El ejemplo anterior, pretende aclarar la definición de planta. En ese caso, la entrada del sistema sería la alimentación de las bombillas, y la salida sería la temperatura leída por el sensor. Si se modificase cualquiera de las partes entre la entrada y la salida mencionada, el sistema variaría.

3.3.2 Actuador

Como actuador se entenderá todo el conjunto de circuitos electrónicos necesarios para adecuar la señal de salida del DAQ para ser válida para la planta.

En el ejemplo mencionado del horno didáctico, la entrada de la planta era la alimentación a unas bombillas para aumentar la temperatura. La potencia de salida de la tarjeta no será suficiente para activar las bombillas directamente, con lo que, como mínimo, habría que introducir un amplificador de potencia. En la realidad, sin embargo, como la alimentación de las bombillas es en alterna, el diseño de este actuador concreto se complicaría con respecto a otros.

Para el caso de la planta de control de nivel, sin embargo, la entrada de la misma sería directamente la señal de control del variador (con un rango de 0 a 10 V). El actuador en este caso puede no ser ninguno, si el rango del DAQ coincide con el de entrada a la planta; o ser un amplificador que adapte los rangos de salida del DAQ a la entrada de la planta. Esta planta se explicará en profundidad más adelante.

El diseño de cada actuador dependerá de la naturaleza de cada planta. Es posible incluir el actuador como parte de la planta, por eso se aconseja definir cada planta indicando lo que se entiende por entrada de control propiamente dicha en cada caso.

3.3.3 Sensor

Al igual que en el caso de la entrada a la planta, ocurriría con la salida. La naturaleza de la señal de salida de la planta real no tiene porqué ser una señal de tensión (la entrada estándar para los DAQs); o, de serlo, no tiene porque coincidir con el rango del canal de entrada del DAQ utilizado.

Como en el ejemplo usado en el resto del trabajo, el sensor de temperatura LM35 proporciona una salida en un rango de 0 a 1 V. Si conectásemos directamente el sensor a una tarjeta con rango de entrada de 0 a 10 V (por ejemplo), estaríamos directamente reduciendo la precisión del sistema. Cada conversor A/D tiene un rango de funcionamiento, y para obtener la mayor precisión en la conversión, sería necesario que el rango de la señal de entrada fuese el mismo que el del conversor. En el caso del LM35, se usaría un amplificador para adaptar el rango de 0-1 V a 0-10 V.

Como ocurre con el actuador, el diseño del módulo sensor depende tanto de la salida de la planta como de la entrada al DAQ. Por este motivo, puede darse el caso de que se tome el bloque sensor como parte de la planta. Como se mencionó, es aconsejable definir perfectamente la planta e indicar fehacientemente sus entradas y salidas en cada caso.

3.3.4 Diferencia entre Actuador y Sensor

A pesar de que aparentemente estos dos bloques pueden resultar muy similares, la gran diferencia entre ellos radica en el sentido de la información. En el caso del actuador, las señales irían del DAQ a la planta; mientras que el bloque sensor se encargaría de las señales procedentes de la planta hacía el DAQ.

4 ESQUEMA DE CONTROL EN PLANTAS DE LABORATORIO

Las plantas de laboratorio disponibles, se han ido nombrando a lo largo del trabajo para aclarar los conceptos. En este apartado se describirán brevemente las plantas, y se explicarán las partes que compondrían el sistema de control en cada caso.

Las plantas disponibles son relativamente sencillas de controlar, no siendo necesario recurrir al control por computador. Con estas plantas se explican a los estudiantes los procesos industriales y las topologías de control que se podrían encontrar en un ambiente laboral. Por ese motivo, se usa el control avanzado en

ellas, consiguiendo de esta manera que los alumnos comprendan el control por computador, a pesar de implementarlo en plantas de laboratorio sencillas.

4.1 CONTROL DE NIVEL

La planta de control de nivel está compuesta por dos depósitos; uno para almacenar el líquido, y otro objetivo donde se controlará el nivel. Para impulsar el agua del depósito almacén al objetivo, se dispone de una bomba centrífuga, pilotada a través de un variador de frecuencia. La lectura del nivel real se hace a través de un sensor de ultrasonidos, que mide la distancia entre el líquido y este dispositivo, pudiendo así calcular el nivel en el depósito objetivo (en la figura 6 puede verse una fotografía de esta planta de laboratorio, junto con la tarjeta de adquisición usada para su control).



Figura 6 – Planta de control de nivel

El volumen de líquido en el depósito no es la variable a controlar; en el caso de esta planta, lo que se pretende controlar es el porcentaje de llenado del depósito.

La tarjeta de adquisición de datos utilizada dispone de rangos diferentes para el canal de entrada (de 0 a 10 V) y salida (de 0 a 5 V). La entrada de control al variador tiene un rango de tensión de 0 a 10 V, al igual que la salida del sensor de ultrasonidos.

4.1.1 Partes discretas

La partes discretas en esta planta incluyen el controlador, y los bloques de escalado correspondientes.

El bloque de escalado de entrada varía la tensión capturada por el DAQ de 0-10 V a un porcentaje de llenado de 0-100%. Se compone de un simple amplificador de ganancia 10. En ocasiones se detectaron múltiples errores de lectura, con lo que se decidió incluir un filtro para reducir estos errores. Estos fallos son debidos a la comunicación serie utilizada entre el PC y el DAQ, y generan lecturas extremas (del 100% de llenado). El problema se solucionó al implementar un filtro paso bajo en el bloque de escalado.

El controlador usado para esta planta por norma general es de tipo PID, pero se ha implementado para pruebas un controlador adaptativo autoajustable y con posibilidad de identificación de la planta.

El bloque de escalado de salida, adapta la señal de salida del controlador, que se supone porcentual por ser las magnitudes de la realimentación y de la consigna, de 0-100% a 0-5 V. Se compone de un simple amplificador de ganancia 0.05.

4.1.2 Partes continuas

Como partes continuas, sólo destacaremos el actuador de salida que realiza la adaptación de la salida de la tarjeta de 0-5 V a 0-10 V (la entrada al variador). Concretamente se trata de un amplificador *hardware* (los de la parte discreta son *software*) de ganancia 2.

Como el rango de salida del sensor de ultrasonidos es el mismo que el de entrada al DAQ, no es necesario implementar un bloque sensor. La salida directa de la planta es la que captura la tarjeta.

A la hora de identificar la planta, hay que tener clara las distinciones de los elementos que la componen; la función de transferencia obtenida sería distinta si se identificase incluyendo el actuador o no.

4.2 HORNO DIDÁCTICO (1)

La maqueta del horno didáctico se ha explicado anteriormente. En este apartado se enumeran simplemente las partes. Se remarca el actuador, que es una de las partes especiales de esta planta. La planta de laboratorio puede apreciarse en la figura 7, junto con una "caja" diseñada al efecto que contiene tanto el bloque actuador como el bloque sensor.

La tarjeta de adquisición usada para el control de esta planta, dispone de un solo rango de funcionamiento de 0 a 10 V tanto para el canal de entrada como el de salida.

4.2.1 Partes discretas

Esta planta incluye los mismos bloques funcionales que la anterior.

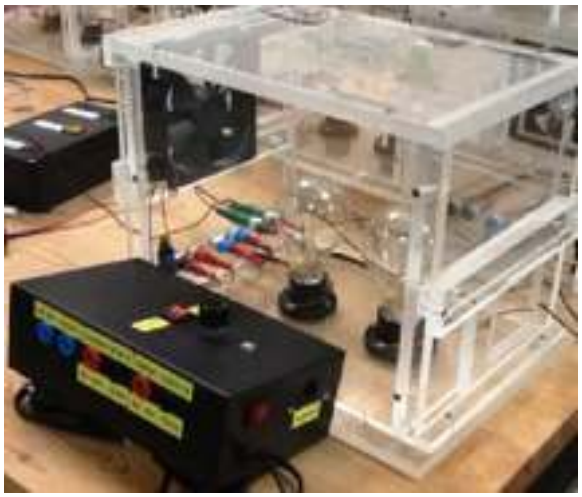


Figura 7 – Horno didáctico

El bloque de escalado de entrada, al disponer de un DAQ con rangos similares tanto para entrada como para salida, de 0 a 10 V, realiza una ampliación similar a la de la planta de control de nivel. Se compone de un amplificador de ganancia 10; pero en este caso el motivo es diferente. Analizando todo el sistema, se obtiene que la salida del sensor de temperatura tiene un rango de 0-1 V; en el bloque sensor, este rango se amplifica a 0-10 V para cubrir todo el margen del DAQ; y el escalado vuelve a ampliarlo a 0-100 para indicar la temperatura real.

La salida del bloque de escalado de entrada es directamente la temperatura en grados centígrados en el interior de la maqueta. La inclusión de un filtro en este caso no es debida a los errores como en la planta de control de nivel. En este caso, es necesario debido al ruido presente en la señal capturada. Con la inclusión de este filtro, se consigue la reducción idónea del ruido y que la salida del bloque "siga" fielmente a la variación real de la temperatura. Se puede apreciar en la figura 8 el contenido de este bloque de escalado diseñado en Simulink® para una de las pruebas de control.

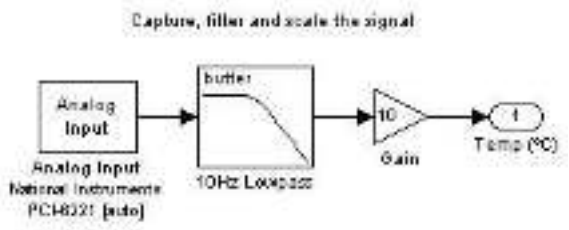


Figura 8 – Bloque de escalado de entrada

El controlador usado en esta planta, al igual que en la anterior, es compatible con múltiples topologías, siendo el PID el elegido en las prácticas para comparar su funcionamiento con controladores industriales similares.

El acondicionamiento de salida en esta planta es similar a la de control de nivel. En este caso se supone un rango de funcionamiento del controlador también de 0 a 100, al igual que el rango de entrada, y como la salida del DAQ sería de 0 a 10 V, se implementaría un amplificador de ganancia 0.1.

4.2.2 Partes continuas

El bloque en el que nos queremos centrar en esta planta de laboratorio es el del actuador. Como se comentó a lo largo del trabajo, la planta dispone de dos bombillas incandescentes como elemento calefactor. Las bombillas están alimentadas con corriente alterna, y para su activación y control se implementó un sistema de potencia a través de un Triac.

Un dispositivo Triac permite el "recorte" de la señal alterna en función de un ángulo de disparo concreto. El actuador diseñado para esta planta realiza la "conversión" de una tensión continua de 0-10 V en potencia entregada de 0-100%.

El actuador es lineal, con lo que una variación de tensión en su entrada de 0.1 a 0.2, implica una variación de potencia entregada a las bombillas del 1 al 2 %. Al igual que una variación de 8.4 a 8.6 voltios, implicaría una variación de potencia del 84 al 86 %. Gracias a disponer de un actuador lineal, el bloque de escalado de salida fue tan sencillo como se mencionó anteriormente. Este actuador lineal se consigue gracias al procesado interno de la señal de control con un microcontrolador que calcula los ángulos de disparo en cada momento.

La planta de laboratorio tiene como entrada la alimentación a las bombillas, y como salida la señal del sensor de temperatura.

Como el sensor de temperatura tiene un rango en su salida de 0-1 V, y la entrada del DAQ de 0-10 V, el bloque sensor se compone de un simple amplificador de ganancia 10. Se podría haber implementado el filtro mencionado en el escalado de entrada de manera *hardware*, pero en esta planta se ha optado por la implementación del mismo tipo *software*.

4.3 HORNO DIDÁCTICO (2)

Esta última planta es una variante de la anterior, donde el actuador no es lineal. Como se explicó, el actuador lineal hace una entrega de potencia lineal con la entrada de tensión; sin embargo en este caso, el ángulo de disparo del Triac es el que es lineal con respecto a la entrada.

Al tener una variación lineal del ángulo de disparo del Triac, la potencia entregada a las bombillas no es

lineal con respecto a la entrada del actuador, como se puede apreciar en la figura 9.



Figura 9 – Salida de potencia con respecto a la entrada del actuador no lineal

Las partes discretas para el control de esta planta son las mismas que para la anterior; sin embargo, el módulo de acondicionamiento de salida, varía sustancialmente.

El actuador de esta planta es diferente, y al no haber una entrega lineal de la potencia, esa no linealidad debería de ser paliada por el acondicionamiento de salida. También podría elegirse no corregir esa no linealidad, y que el controlador implementado sea el que se encargue de corregirla (o la implementación de un regulador no lineal), pero en este caso se optó por implementar un código de programa que realiza una función similar a la del actuador lineal.

La gran ventaja de este método es que el actuador sería más simple, su construcción más sencilla, y se usaría el ordenador de control para paliar la no linealidad. Así se aprovecharía la potencia de cálculo disponible, y se haría que el conjunto global funcionase de manera similar a la planta anterior.

El resto de los bloques del bucle de control son similares al anterior al no haber más variaciones que la mencionada.

5 CONCLUSIONES

Se ha tratado de exponer un esquema común para los sistemas de control por computador, con las pequeñas variaciones que aparecen al estudiar un sistema concreto. Se ha comprobado que una vez que los alumnos comprenden el nuevo esquema, les resulta más sencillo el abordar el estudio de una planta de laboratorio.

Algunos estudiantes se han enfrentado a la tarea de identificar las plantas disponibles, comprobando la diferencia entre funciones de transferencia obtenida según identifiquemos sólo la planta, o el conjunto actuador-planta-sensor.

Para evitar el uso de actuadores y sensores, actualmente se están modificando las plantas de laboratorio para tener entradas y salidas estandarizadas. Concretamente en el Laboratorio de Automática de la Escuela Universitaria Politécnica de Ferrol se ha optado por usar un rango de 0 a 5 V para poder utilizar cualquier dispositivo para realizar el control de las plantas. Inicialmente se había optado por un rango de 0 a 10 V, pero después de estudiar las nuevas placas de microcontroladores Arduino y sus aplicaciones para el control de procesos, se decidió variar ese rango.

En la construcción de las nuevas plantas, tanto el actuador como el sensor se contemplan como elementos de la propia planta, de manera que se conecta directamente las tarjetas de adquisición de datos a las plantas de laboratorio, evitando posibles fallos en el *hardware*, y complicaciones en los bucles de control.

Referencias

[1] Aguado, A., (2003) Identificación y control adaptativos, Prentice Hall.

[2] Ogata, K., (1998) Sistemas discretos de control en tiempo discreto, Prentice Hall.

[3] Ogata, K., (2003) Ingeniería de control moderna, Prentice Hall.

[4] Bolzern, P., (2009) Fundamentos de control automático, McGraw Hill.

CONTROL MULTIVARIABLE QFT PARA LA DINÁMICA LONGITUDINAL DE UN VEHÍCULO AÉREO

Javier Joglar Alcubilla
Dept. de Informática y Automática. UNED. Madrid. "jjoglar@educa.madrid.org"

Joaquín Aranda Almansa
Dept. de Informática y Automática. UNED. Madrid. "jaranda@dia.uned.es"

Resumen

QFT exige trabajar con plantas cuadradas o, en algunos casos aplicando procedimientos de "squaring-down", es posible utilizar más entradas que salidas, aunque siempre en un proceso de reducción de las variables de entrada, que implicará pérdida de características de la planta. El procedimiento de "squaring-up" utilizando compensación MIMO-QFT permite trabajar con plantas con incertidumbre infra-actuadas, es decir, con más salidas que entradas, en un proceso sin supresión de especificaciones del sistema a controlar. Además, en el proceso de diseño se incorpora optimización del esfuerzo de control, tanto para regulación como para seguimiento

Palabras Clave: mimo, qft, infra-actuada, squaring up, optimización del esfuerzo de control , tracking.

1 INTRODUCCIÓN

QFT se caracteriza por su versatilidad sobre plantas con perturbaciones e incertidumbre, para regulación y seguimiento, pudiendo aplicarse sobre sistemas lineales y no lineales, variables e invariantes en el tiempo, continuos y discretos, SISO, MISO y MIMO, sistemas de mínima fase y otros de no mínima fase, [7], [15], [8]. Se han desarrollado metodologías de trabajo directas o DPD y también inversas o IPD [10], compensación diagonal y no-diagonal [5]. Sin embargo, QFT exige trabajar en general con plantas cuadradas o, en algunos casos aplicando procedimientos de "squaring-down" [13], [9], es posible utilizar más entradas que salidas, aunque siempre en un proceso de reducción de las variables de entrada, que supondrá la no consideración de ciertas propiedades de la planta.

"Squaring-up" es un procedimiento de compensación QFT sobre sistemas MIMO en donde existan más salidas que entradas, en particular, de tamaños *[2,n]*, con *n≥2* y paridad *par*, que incorpora optimización (minimización) del esfuerzo de control, en un proceso sin eliminación de características de la planta.

Cuando se trabaja sobre una planta no cuadrada se han desarrollado algunas técnicas para tratar de aplicar control MIMO QFT de forma adecuada, como si se tratara de una planta cuadrada y regular:

1. "Squaring Down": las plantas con más entradas(*m*) que salidas(*n*) ($[m,n], m>n$) son nombradas por [2] como sistemas sobre-actuados, proponiendo un procedimiento de regularización de la función de transferencia del lazo, convirtiéndola en cuadrada $[n,n]$, mediante el uso entre controlador y planta de un precompensador de tamaño $[n,m]$. Aplicable a sistemas con más entradas que salidas, nunca a la inversa.

2. "Non square relative gain array (NRG)": propuesto por [16], es una variante de "Squaring down" que mide las interacciones entre salidas y entradas en sistemas sobre-actuados. Aplicado sobre plantas con más entradas (*m*) que salidas (*n*) ($[m,n], m>n$), consigue transformar el sistema MIMO en *n* sistemas MISO, resolubles de forma independiente. Sin embargo, sigue siendo sólo aplicable a sistemas con más entradas que salidas y no a la inversa. Impide la utilización de ciertas especificaciones no nulas para tracking cruzado.

"Squaring up" consiste en sustituir el *sistema_inicial [2,n]*por *m subsistemas* (*m=n/2*) MIMO *[2,2]*, todos con entradas de control comunes pero, un único compensador en "feedback" *[2,2]* y un prefiltro *[2,n]* independientes para cada uno de ellos. La solución es un controlador diagonal *[2,n]* y un prefiltro No diagonal *[n,n]* con elementos "off-diagonal", capaces de proporcionar robustez en el TD y en el WD [1].

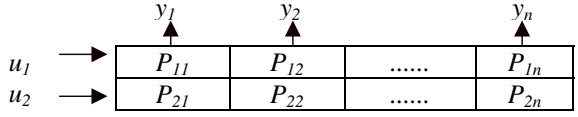


Figura_1: *Composición de la planta* $P[2,n]$, n≥2 *par*

Donde $u=(u_1,u_2)$ son las entradas de control y que es compatible en ser agrupada en subsistemas cuadrados tal que, $Y_1=(y_1,y_2)$, ... , $Y_m=(y_{n-1},y_n)$; es decir, *m subsistemas* de tamaño *[2,2]* cada uno, siendo *m=n/2,* ordenados por :

1. prioridad de especificaciones; esto es, el *subsistema_1* es aquel al que se imponen las especificaciones más restrictivas del problema principal;
2. a partir de un estudio previo de acoplamientos entradas-salidas, según el grado de acoplos-desacoplos detectados; se busca máximo acoplamiento en los elementos diagonales del *subsistema_1*, y a ser posible también máximo desacoplo en sus elementos no diagonales. Expresada la planta como el modelo lineal de espacio de estados siguiente, $\left\{\begin{matrix}\dot{x}=Ax+Bu\\ y=Cx+Du\end{matrix}\right\}$ donde *u* son las entradas de control, *y* las salidas, *x* los estados y *A,B,C,D* las matrices del espacio de estados; el producto de la matriz *C* con la matriz de *eigenvectores* de la matriz *A* proporciona el grado de *acoplamiento salidas-entradas*. Este estudio es aplicable también a la planta de perturbaciones para detectar *acoplamientos salidas-perturbaciones* de entrada. Los *eigenvalores* de la matriz *A* proporcionan además los modos de oscilación característicos de la planta y, de ellos se puede obtener el rango de frecuencias de trabajo de la misma.

Inicialmente se plantea "*squaring up*" en términos de regulación y seguimiento, por separado, para después mezclar los resultados a efectos de obtener la solución al problema combinado tracking-regulación. En una segunda etapa, se incorporan especificaciones para seguimiento y regulación que incluyan minimización del esfuerzo de control (optimización). Para ello, las especificaciones de esfuerzo de control formarán parte de las especificaciones básicas iniciales para seguimiento (tracking) y regulación (sensibilidades).

2 DISEÑO DE CONTROL INICIAL

Problema de control *[2,n]* esquematizado en la *Figura_2* y agrupado por subsistemas cuadrados *[2,2]*, usando controladores diagonales $G_i[2,2]$, prefiltros no diagonales $F_{ij}[2,2]$, subplantas $P_i[2,2]$, referencias de entrada $R_1=(r_1,r_2)$,...., $R_m=(r_{n-1},r_n)$, subplantas de perturbaciones $Pd_i[2,2]$, perturbaciones externas $d=(d_1,d_2)$, entradas de control $u=\sum_{k=1}^{m}G_k\sum_{i=1}^{m}E_{ki}$,con $E_{ki}=F_{ki}R_i-Y_i$

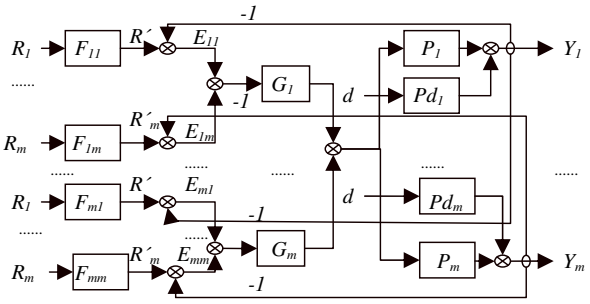


Figura_2: *Problema de tracking/regulación [2,n] agrupado por subsistemas cuadrados [2,2]*

La solución global viene dada por las *m* soluciones de los problemas MIMO $[2,2]$ propuestos a continuación, donde cada compensador G_i y cada prefiltro F_{ii} está especializado en una única planta P_i.

El problema MIMO $[2,n]$ presentado se descompone en *m* problemas (*subsistemas)* MIMO $[2,2]$, tal y como se describen de forma genérica en la *Figura_3,* donde, en términos de regulación, se tiene,

$$Pd_1^{\,i}=-G_1\frac{Y_i}{d}\text{ , con }\ i=2,...,m \qquad (2.1)$$

.....

$$Pd_m^{\,i}=-G_m\frac{Y_i}{d}\text{ , con }\ i=1,...,m-1 \qquad (2.2)$$

$$A_1=-\sum_{k=2}^{m}G_k\sum_{i=1}^{m}\frac{Y_i}{d} \qquad (2.3)$$

.....

$$A_m=-\sum_{k=1}^{m-1}G_k\sum_{i=1}^{m}\frac{Y_i}{d} \qquad (2.4)$$

son funciones que describen la contribución, en forma de perturbaciones de entrada, de los demás *subsistemas* sobre el *subsistema_j* a estudio: $Pd_j^{\,i}$ representa la influencia de la *salida_i* realimentada a través del controlador G_j del *subsistema_j* ; A_i es la influencia de todas las salidas realimentadas a través de todos los controladores, excepto el del *subsistema_j* estudiado. Las relaciones $\frac{Y_i}{d}$ para $i=1,...,m$, se utilizan en forma de imposición de restricciones, a través de las especificaciones de sensibilidad, generalizadas para cada *subsistema_i* como ,

$$|S_i|=\left|\frac{Y_i}{d}\right|\le Wps_i \qquad (2.5)$$

Por tanto, para el *subsistema_j* a estudio, escogiendo la peor situación en la que con perturbaciones externas *d* las salidas Y_i son máximas, se tiene

$$Pd_j^i = -G_j Wps_i \ , i = 1,...,m, i \neq j \quad (2.6)$$

$$A_i = -\sum_{k=1,k\neq j}^{m} G_k \sum_{i=1}^{m} \frac{Y_i}{d} = -\sum_{k=1,k\neq j}^{m} G_k \sum_{i=1}^{m} Wps_i \ , i = 1,...,m, i \neq j \quad (2.7)$$

De esta forma, para el *subsistema_j* genérico, las entradas de control en cuanto a regulación, se pueden describir como,

$$u_{rj} = -G_j Y_j - dG_j \sum_{i=1,i\neq j}^{m} Wps_i - d \sum_{k=1,k\neq j}^{m} G_k \sum_{i=1}^{m} Wps_i \quad (2.8)$$

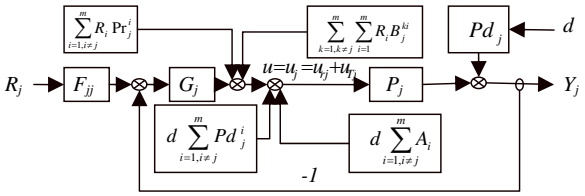


Figura_3: *Subsistema_j genérico MIMO [2,2]. Los m subsistemas_j componen el MIMO [2,n]*

Por otro lado, en términos de tracking tenemos,

$$\Pr_1^i = G_1\left(F_{1i} - \frac{Y_i}{R_i}\right) \text{, con } i = 2,...,m \quad (2.9)$$

.....

$$\Pr_m^i = G_m\left(F_{mi} - \frac{Y_i}{R_i}\right), \text{ con } i = 1,...,m-1 \quad (2.10)$$

$$B_1^{ki} = G_k\left(F_{ki} - \frac{Y_i}{R_i}\right), \text{ con } i = 1,...,m \text{ y } k = 2,...,m \quad (2.11)$$

.....

$$B_m^{ki} = G_k\left(F_{ki} - \frac{Y_i}{R_i}\right), \text{ con } i = 1,...,m \text{ y } k = 1,...,m-1 \quad (2.12)$$

son funciones que describen la contribución, en forma de perturbaciones de entrada, de los demás *subsistemas* sobre el *subsistema_j* a estudio: $\Pr_j^i$ representa la influencia de la *salida_i* realimentada a través del controlador G_j del *subsistema_j* ; B_j^{ki} es la influencia de todas las salidas realimentadas a través de todos los controladores, excepto el del *subsistema_j* estudiado. Las relaciones Y_i / R_j para $i = 1,...,m$ y $j = 1,..,m$, se utilizan en forma de imposición de restricciones, a través de las especificaciones de tracking, generalizadas para cada *subsistema_i* como,

$$\alpha_i \leq |T_i| = \left|\frac{Y_i}{R_j}\right| \leq \beta_i \quad (2.13)$$

Por tanto, para el *subsistema_j* a estudio, escogiendo la peor situación en la que con referencias de entrada R_j las salidas Y_i son máximas, se tiene,

$$\Pr_j^i = G_j\left(F_{ji} - \beta_i\right), \text{ con } i = 1,...,m, i \neq j \quad (2.14)$$

$$B_j^{ki} = G_k\left(F_{ki} - \frac{Y_i}{R_i}\right) = G_k\left(F_{ki} - \beta_i\right), \text{ con } i = 1,...m \text{ y } k = 1,...,m, k \neq j \quad (2.15)$$

De esta forma, para el *subsistema_j* genérico, las entradas de control en cuanto a tracking, se pueden describir como,

$$u_{Tj} = G_j\left(F_{jj} R_j - Y_j\right) + G_j \sum_{i=1,i\neq j}^{m} R_i\left(F_{ji} - \beta_i\right) + \sum_{k=1,k\neq j}^{m} G_k \sum_{i=1}^{m} R_i\left(F_{ki} - \beta_i\right) \quad (2.16)$$

En definitiva, las entradas de control para todo subsistema vendrán dadas por:

$$u_j = u_{rj} + u_{Tj} = u \quad (2.17)$$

Un *subsistema_j* genérico se puede describir en forma matricial como sigue *[15],*

$$Y_j = Pd_j d + P_j u_j \text{ con } j \equiv (k-1, k) \text{ y } k=2,4,..,n \quad (2.18)$$

Para el desarrollo de (2.18) y obtención de especificaciones utilizaremos las siguientes simplificaciones, donde se ha usado la notación $P^{-1} = [q_{ij}]$ y $G = diag(g1, g2)$, en todas ellas con $i = 1,3,5,..., n-1$ y *j=2,4, ..,n :*

$$q_{2i}^1 = q_{2i} - \frac{q_{2i} q_{1i}}{q_{1i} + g1} = \frac{q_{2i} g1}{q_{1i} + g1} \quad (2.19)$$

$$q_{2j}^2 = q_{2j} - \frac{q_{2,j-1} q_{1j}}{q_{1,j-1} + g1} \quad (2.20)$$

$$q_{2,j-1}^{1i} = \frac{q_{2,j-1} g_i}{q_{1,j-1} + g_{j-1}} \quad (2.21)$$

$$q_{2j}^{2i} = q_{2j} - \frac{q_{2,j-1} q_{1j}}{q_{1,j-1} + g_i} \quad (2.22)$$

3 DISEÑO DE CONTROL OPTIMIZADO

Vamos a incorporar en el "diseño de control inicial" una descripción de especificaciones para seguimiento y regulación que incluya minimización del esfuerzo de control (diseño de control optimizado) [15]. Para ello, las especificaciones de esfuerzo de control

optimizado se van a introducir en las especificaciones básicas para seguimiento (tracking) y regulación (sensibilidades) obtenidas con el desarrollo de la ecuación (2.18), de manera que formen parte de las mismas.

3.1 REGULACIÓN OPTIMIZADA

Partiendo de la ecuación, $u_r = -\sum_{k=1}^{m} G_k \sum_{i=1}^{m} Y_i$ (3.1)

Si el esfuerzo de control está optimizado [15], es decir, se utilizan entradas de control optimizadas u_{x1}^{opt} y u_{x2}^{opt}, manteniendo el esfuerzo de control para regulación lo más cerca posible de su valor mínimo, las salidas se anulan y las especificaciones de esfuerzo de control para regulación , vendrán descritas del siguiente modo,

$$\left|Ecr_{lj}^{\ opt}\right| = \left|\frac{u_l^{opt}}{d_j}\right| \le Wecr_{lj} \text{ ,con } l \in [1,2,..,n] \text{ y } j \in [1,2] \quad (3.2)$$

Para el *subsistema_x* genérico optimizado, con *x≡[x1,x2],* siendo *x1=1,3,..,n-1* y *x2=2,4,..,n,* se obtiene,

$$\begin{cases} y_{x1} = 0 = P_{1,x1}u_{x1}^{opt} + P_{1,x2}u_{x2}^{opt} + Pd_{1,x1}d_1 + Pd_{1,x2}d_2 & (3.3) \\ y_{x2} = 0 = P_{2,x1}u_{x1}^{opt} + P_{2,x2}u_{x2}^{opt} + Pd_{2,x1}d_1 + Pd_{2,x2}d_2 & (3.4) \end{cases}$$

3.2 TRACKING OPTIMIZADO

Partiendo de la ecuación, $u_T = \sum_{k=1}^{m} G_k \sum_{i=1}^{m} \left(F_{ki}R_i - Y_i\right)$ (3.5)

Si el **esfuerzo de control** está **optimizado**, es decir, con entradas de control optimizadas u_{x1}^{opt} y u_{x2}^{opt}, manteniendo el esfuerzo de control para seguimiento lo más cerca posible de su valor mínimo, **las salidas seguirán a sus referencias de entrada** y las especificaciones de esfuerzo de control para tracking, vendrán descritas como,

$$\left|Ecs_{lj}^{opt}\right| = \left|\frac{u_l^{opt}}{r_j}\right| \le Wecs_{lj} \text{ ,con } l \in [1,2,..,n] \text{ y } j \in [1,2,..,n] \quad (3.6)$$

Para un *subsistema_x genérico*, optimizando para seguimiento las salidas y_{x1}, y_{x2}, se consigue:

$$\begin{cases} y_{x1} = P_{1,x1}u_{x1}^{opt} + P_{1,x2}u_{x2}^{opt} = r_{x1} & (3.7) \\ y_{x2} = P_{2,x1}u_{x1}^{opt} + P_{2,x2}u_{x2}^{opt} = r_{x2} & (3.8) \end{cases}$$

3.3 DESCRIPCIÓN DE ESPECIFICACIONES OPTIMIZANDO ESFUERZO DE CONTROL

Con las ecuaciones definidas para el problema de control inicial y, a partir de (2.18), se obtienen las especificaciones descritas a continuación, usando los desarrollos optimizados propuestos en 3.1 y 3.2. [3] .

Sensibilidades: Limitaciones impuestas por $\left|\frac{y_i}{d_j}\right| \le Wps_{ij}$, con *i=1,..,n* y *j=1,2*. Para un s*ubsistema_x* genérico optimizado para regulación, con *x≡[x1,x2],* siendo x_1*=1,3,..,n-1* y x_2*=2,4,..,n*, las sensibilidades vienen dadas por,

$$\left|S_{x1,1}\right| = \left|\frac{y_{x1}}{d_1}\right| = \left|\frac{\left|q_{1,x1}Pd_{1,x1}\right| + \left|q_{1,x2}Pd_{2,x1}\right| + \left|q_{1,x2}Wps_{x2,1}\right| + Wecr_{x1,1}}{q_{1,x1} + g_{x1}}\right| \le Wps_{x1,1} \quad (3.9)$$

$$\left|S_{x1,2}\right| = \left|\frac{y_{x1}}{d_2}\right| = \left|\frac{\left|q_{1,x1}Pd_{1,x2}\right| + \left|q_{1,x2}Pd_{2,x2}\right| + \left|q_{1,x2}Wps_{x2,2}\right|}{q_{1,x1} + g_{x1}}\right| \le Wps_{x1,2} \quad (3.10)$$

$$\left|S_{x2,1}\right| = \left|\frac{y_{x2}}{d_1}\right| = \left|\frac{\left|q_{2,x1}^{1,x1}Pd_{1,x1}\right| + \left|q_{2,x2}^{2,x1}Pd_{2,x1}\right| + \left|\frac{q_{2,x1}}{q_{1,x1} + g_{x1}}Wecr_{x1,1}\right|}{q_{2,x2}^{2,x1} + g_{x2}}\right| \le Wps_{x2,1} \quad (3.11)$$

$$\left|S_{x2,2}\right| = \left|\frac{y_{x2}}{d_2}\right| = \left|\frac{\left|q_{2,x1}^{1,x1}Pd_{1,x2}\right| + \left|q_{2,x2}^{2,x1}Pd_{2,x2}\right| + Wecr_{x2,2}}{q_{2,x2}^{2,x1} + g_{x2}}\right| \le Wps_{x2,2} \quad (3.12)$$

NOTA: Observar en las ecuaciones anteriores como las especificaciones de esfuerzo de control para regulación *Wecr* (3.2) forman parte de la estructura de las especificaciones de sensibilidad.

Tracking: Usando entradas de referencia *r* , para un s*ubsistema_x* genérico optimizado para tracking, con *x≡[x1,x2],* siendo *x1=1,3,..,n-1* y *x2=2,4,..,n* , y las limitaciones impuestas por $\alpha_{ij\le}\left|\frac{y_i}{r_j}\right| \le \beta_{ij}$, con *i=1,..,n* y *j=1,...,n* , las ecuaciones de tracking serán,

$$\alpha_{x1,x1} \le \left|T_{x1,x1}\right| = \left|\frac{y_{x1}}{r_{x1}}\right| = \left|\frac{\left|g_{x1}\right| + Wecs_{x1,x1} + \left|q_{1,x2}\beta_{x2,x1}\right|}{q_{1,x1} + g_{x1}}\right| \le \beta_{x1,x1} \quad (3.13)$$

$$\alpha_{x1,x2} \le \left|T_{x1,x2}\right| = \left|\frac{y_{x1}}{r_{x2}}\right| = \left|\frac{Wecs_{x1,x2} + \left|q_{1,x2}\beta_{x2,x2}\right|}{q_{1,x1} + g_{x1}}\right| \le \beta_{x1,x2} \quad (3.14)$$

$$\alpha_{x2,x1} \le \left|T_{x2,x1}\right| = \left|\frac{y_{x2}}{r_{x1}}\right| = \left|\frac{\left|\frac{q_{2,x1}}{q_{1,x1}+g_{x1}}\right|\left(Wecs_{x1,x1}+\left|g_{x1}\right|\right)+Wecs_{x2,x1}}{q_{2,x2}^{2,x1}+g_{x2}}\right| \le \beta_{x2,x1} \quad (3.15)$$

$$\alpha_{x2,x2} \le \left|T_{x2,x2}\right| = \left|\frac{y_{x2}}{r_{x2}}\right| = \left|\frac{\left|\frac{q_{2,x1}}{q_{1,x1}+g_{x1}}\right|Wecs_{x1,x2}+Wecs_{x2,x2}+\left|g_{x2}\right|}{q_{2,x2}^{2,x1}+g_{x2}}\right| \le \beta_{x2,x2} \quad (3.16)$$

NOTA: Observar en las ecuaciones anteriores cómo las especificaciones de esfuerzo de control para seguimiento $Wecs$ (3.6) forman parte de la estructura de las especificaciones de tracking.

Estabilidad Robusta: Expresada como,

$$E = \left(1+PG\right)^{-1} \quad (3.17)$$

donde $P = 1/q$ es la planta efectiva y G el controlador. Las limitaciones de estabilidad se denotan por $\left|Te_{ij}\right| \le We_l$, con $i = 1,2$, $j = 1,..,n$ y $l = 1,..,n$, dando lugar de forma genérica, con $k = 2,4,..,n$, a,

$$\left|Te_{1,k-1}\right| = \left|Te_{1k}\right| = \left|\frac{q_{1,k-1}}{q_{1,k-1}+g_{k-1}}\right| \le We_{k-1} \quad (3.18)$$

$$\left|Te_{2,k-1}\right| = \left|Te_{2k}\right| = \left|\frac{q_{2k}^{2,k-1}}{q_{2k}^{2,k-1}+g_k}\right| \le We_k \quad (3.19)$$

Esfuerzo de Control:

Para regulación, $$ECr = \frac{G}{1+PG} \quad (3.20)$$

Y para seguimiento, $$ECs = F\frac{G}{1+PG} \quad (3.21)$$

Donde *P* es la planta efectiva, *G* el controlador y *F* el prefiltro. Para un *subsistema_i genérico*, siendo $i \equiv (k-1,k)$, con $k = 2,4,..,n$, se tiene,

1.En Términos de Regulación :

$$\left|Ecr_{k-1,1}\right| = q_{1,k-1}\left|S_{k-1,1}\right| \le Wecr_{k-1,1} \quad (3.22)$$

$$\left|Ecr_{k-1,2}\right| = q_{1,k-1}\left|S_{k-1,2}\right| \le Wecr_{k-1,2} \quad (3.23)$$

$$\left|Ecr_{k1}\right| = q_{2k}^{2,k-1}\left|S_{k1}\right| \le Wecr_{k1} \quad (3.24)$$

$$\left|Ecr_{k2}\right| = q_{2k}^{2,k-1}\left|S_{k2}\right| \le Wecr_{k2} \quad (3.25)$$

2.En Términos de Seguimiento :

$$\left|Ecs_{k-1,1}\right| = q_{1,k-1}\left|T_{k-1,1}\right| \le Wecs_{k-1,1} \quad (3.26)$$

.....

$$\left|Ecs_{k-1,k}\right| = q_{1,k-1}\left|T_{k-1,k}\right| \le Wecs_{k-1,k} \quad (3.27)$$

$$\left|Ecs_{k1}\right| = q_{2k}^{2,k-1}\left|T_{k1}\right| \le Wecs_{k1} \quad (3.28)$$

.....

$$\left|Ecs_{kk}\right| = q_{2k}^{2,k-1}\left|T_{kk}\right| \le Wecs_{kk} \quad (3.29)$$

4 CONTROL "SQUARING-UP" PARA LA DINÁMICA LONGITUDINAL DE UNA AERONAVE CIVIL

Se utilizará como problema aplicación del procedimiento de "squaring-up" planteado el modelo de aeronave civil RCAM descrito en el proyecto GARTEUR ("*Group for aeronautical research and technology in europe*") por el grupo FM(AG08) ("*Flight_Mechanics*") [6], [11]. En particular, el problema combinado regulación/seguimiento propuesto se describe bajo los siguientes términos:

- Planta del Canal Longitudinal de la aeronave, $P\left[2,8\right]$ con subplantas $P_i\left[2,2\right]$, tal que $P \equiv \left[P_1, P_2, .., P_4\right]$
- Planta de Perturbaciones, $Pd\ \left[2,8\right]$ con $Pd_i\left[2,2\right]$, tal que $P_d \equiv \left[P_{d1}, P_{d2}, .., P_{d4}\right]$,

donde el vector de perturbaciones externas es $d \equiv \left[w_z, w_{zb}\right]$, las entradas de control $u \equiv \left[DT, TH\right]$ y las salidas $y \equiv \left[z, Va, Wv, V, \alpha, \gamma, n_z, n_x\right]$[6]. El orden seleccionado de las variables en estos vectores *(d,u,y)* se obtiene a partir de un estudio a priori de acoplamientos entradas-salidas, por un lado y, perturbaciones-salidas, por otro.

La solución es un controlador No diagonal [2,8] y un prefiltro diagonal [8,8] con elementos off-diagonal.

Se va a trabajar con 881 escenarios de vuelo distintos (plantillas), incertidumbre de la planta. La banda de frecuencia seleccionada para procesamiento de las plantillas, tras el estudio de rango de frecuencias de los modos de oscilación característicos de la planta previo, es de $\omega(0.5-1.7) rad/s$. A efectos de conseguir estabilidad robusta en el diseño de los

controladores, se amplía la banda de frecuencia hasta los *30rad/s*, suficiente al reducirse con ella de manera importante la ganancia de los distintos subsistemas en lazo abierto [14].

En la aplicación del proceso de diseño, utilizando toolbox QFT de MATLAB [3], se han considerado, en general:

- Criterios de estabilidad basados en el trabajo de [4].
- Las altas frecuencias deben ser atenuadas, a efectos de minimizar el ruido de los sensores [15].

Los resultados obtenidos gráficamente para el primer subsistema son los dados en la Figura_4.1 y Figura_4.2. Los resultados definitivos al problema planteado se describen en la Tabla1, Tabla2 y Tabla3.

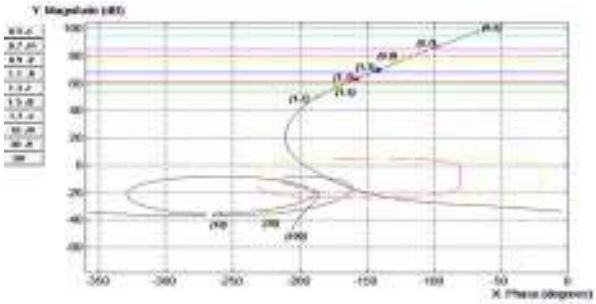


***Figura_4.1**: Obtención del controlador del primer lazo , primer subsistema, g1*

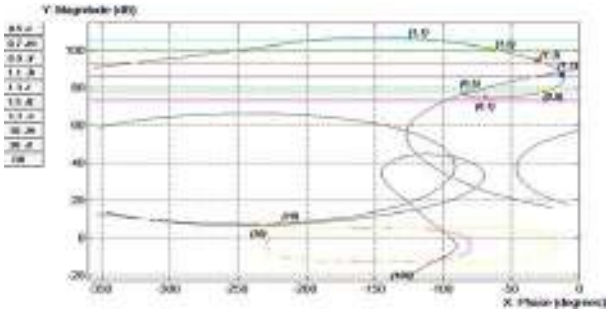


***Figura_4.2**: Obtención del controlador del segundo lazo , primer subsistema, g2*

***Tabla 1**: Estructura del controlador y prefiltro solución al problema RCAM longitudinal [2,8]*

Controlador No diagonal [2,8]	**Prefiltro diagonal [8,8]**
$G=\begin{vmatrix} g_1 & 0 & g_3 & 0 & g_5 & 0 & g_7 & 0 \\ 0 & g_2 & 0 & g_4 & 0 & g_6 & 0 & g_8 \end{vmatrix}$	$F=\begin{vmatrix} F_{11} & F_{12} & F_{13} & 0 & F_{15} & F_{16} & F_{17} & 0 \\ F_{21} & F_{22} & F_{23} & 0 & F_{25} & F_{26} & F_{27} & 0 \\ F_{31} & F_{32} & F_{33} & 0 & F_{35} & F_{36} & F_{37} & 0 \\ F_{41} & F_{42} & F_{43} & F_{44} & F_{45} & F_{46} & F_{47} & 0 \\ F_{51} & F_{52} & F_{53} & 0 & F_{55} & F_{56} & F_{57} & 0 \\ F_{61} & F_{62} & F_{63} & 0 & F_{65} & F_{66} & F_{67} & 0 \\ F_{71} & F_{72} & F_{73} & 0 & F_{75} & F_{76} & F_{77} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & F_{88} \end{vmatrix}$

***Tabla 2**: Funciones de transferencia de los elementos del controlador solución al RCAM longitudinal [2,8]*

$g_1=\frac{0.48s^6+0.84s^5+57.63s^4+91.47s^3+468.58s^2+358.42s+474.38}{0.0002s^8+0.0048s^7+0.1831s^6+0.8514s^5+2.6942s^4+3.1651s^3+3.0995s^2+0.8421s+0.0285}$
$g_2=\frac{339s^6+373s^5+34263s^4+36996s^3+33604s^2+17576s+461}{0.0008s^6+0.0001s^5+0.21s^4+0.2544s^3+1.2117s^2+0.7572s+1.517}$
$g_3=\frac{180s^6+380s^5+19370s^4+46040s^3+103110s^2+28270s+35470}{s^8+8.175s^7+286.2531s^6+292.4713s^5+843.2095s^4+145.8063s^3+328.1161s^2+23.5322}$
$g_4=\frac{213s^6+22s^5+18716s^4+36357s^3+26958s^2+19338s+3056}{0.0095s^7+0.0196s^6+0.1079s^5+0.4009s^4+0.7099s^3+1.8417s^2+1.3412s+1.9074}$
$g_5=\frac{47s^6+67s^5+4831s^4+4979s^3+12725s^2+4293s+4157}{0.001s^9+0.0341s^8+0.2813s^7+0.5284s^6+1.5343s^5+1.4243s^4+2.1206s^3+0.7999s^2+0.5824s+0.0024}$
$g_6=\frac{-10^6\left(4.92s^4+383.86s^2+159.07s+207.23\right)}{0.001s^6+0.0045s^5+0.7702s^4+0.2538s^3+3.972s^2+0.713s+4.9007}$
$g_7=\frac{630s^6+1520s^5+65290s^4+151160s^3+230540s^2+113910s+46180}{0.001s^8+0.0099s^7+0.0013s^6+0.2132s^5+0.4243s^4+1.3201s^3+1.2778s^2+0.8526s+0.2087}$
$g_8=\frac{478.1s^3+2451.2s^2+3295.9s+1559.8}{0.001s^6+0.0196s^5+0.4392s^4+0.5759s^3+8.0492s^2+3.6825s+6.2902}$

***Tabla 3**: Funciones de transferencia de los elementos del prefiltro solución al RCAM longitudinal [2,8]*

$F_{11}=\frac{0.4482s^2+0.4819s+1.2231}{0.001s^5+0.096s^4+0.321s^3+9.216s^2+10.502s+3.169}$
$F_{12}=\frac{0.0096s^4+0.0066s^3+0.1214s^2+0.0436s+0.2214}{10000s^5+13274s^4+6308s^3+7182s^2+786s+751}$
$F_{13}=\frac{0.1636s^4+0.0431s^3+0.2334s^2+0.0096s+0.0774}{100000s^5+143750s^4+64850s^3+116700s^2+9200s+22670}$
$F_{15}=\frac{0.0017s^2+0.0002s+0.0015}{s^5+0.8432s^4+0.953s^3+0.6704s^2+0.1982s+0.1187}$
$F_{16}=\frac{0.0017s^2+0.0002s+0.0015}{s^5+0.8432s^4+0.953s^3+0.6704s^2+0.1982s+0.1187}$
$F_{17}=\frac{0.0003s^2+0.0001s+0.0014}{s^5+0.7283s^4+0.6702s^3+0.4452s^2+0.0971s+0.0594}$
$F_{21}=\frac{10^{-5}(0.5322s^2+0.5699s+0.7132)}{s^3+0.7708s^2+3.741s+0.5034}$
$F_{22}=\frac{0.0002s^3+0.0024s^2+0.2412s+0.6426}{s^3+16.9158s^2+5.6458s+0.7441}$
$F_{23}=\frac{10^{-6}(0.1537s^2+0.211s+0.3297)}{s^3+0.5913s^2+3.6152s+1.5715}$
$F_{25}=\frac{10^{-3}(0.0881s^2+0.1209s+0.1889)}{s^3+0.5913s^2+3.6152s+1.5715}$
$F_{26}=\frac{10^{-3}(0.0881s^2+0.1209s+0.1889)}{s^3+0.5913s^2+3.6152s+1.5715}$
$F_{27}=\frac{10^{-4}(0.0769s^2+0.1055s+0.1648)}{s^3+0.5913s^2+3.6152s+1.5715}$
$F_{31}=\frac{0.0018s^2+0.002}{s^4+0.7156s^3+0.043s+0.6679}$
$F_{32}=\frac{0.0013s^2+0.0016}{s^4+1.4219s^3+0.3971s+0.4313}$

$$F_{33} = \frac{1.3518\,s + 3.5946}{0.01\,s^5 + 0.97\,s^3 + 0.769\,s^2 + 86.31\,s + 4.562}$$

$$F_{35} = \frac{0.0025\,s^2 + 0.0019\,s + 0.0017}{s^4 + 0.1088\,s^3 + 0.7408\,s^2 + 0.0356\,s + 0.1215}$$

$$F_{36} = \frac{0.0025\;\;s^2 + 0.0019\;\;s + 0.0017}{s^4 + 0.1088\;\;s^3 + 0.7408\;\;s^2 + 0.0356\;\;s + 0.1215}$$

$$F_{37} = \frac{0.0009\;\;s^4 + 0.0003\;\;s^3 + 0.0035\;\;s^2 + 0.0002\;\;s + 0.0065}{s^5 + 0.5848\;\;s^3 + 0.1371\;\;s + 0.0218}$$

$$F_{41} = \frac{10^{-5}(0.161s + 0.0886)}{s^7 + 0.193s^6 + 6.0095s^5 + 2.3066s^4 + 10.0051s^3 + 5.5238s^2 + 1.1921s + 0.9601}$$

$$F_{42} = \frac{10^{-5}(0.5269\,s^2 + 0.0123\,s + 0.284)}{s^4 + 0.1286\,s^3 + 3.7468\,s^2 + 0.1995\,s + 1.9301}$$

$$F_{43} = \frac{0.0375\,s^4 + 0.0776\,s^3 + 0.037\,s^2 + 0.1908\,s + 0.1291}{10000\,s^5 + 56061\,s^3 + 562\,s^2 + 86675\,s + 12727}$$

$$F_{44} = \frac{0.0125\,s^4 + 0.1972\,s^3 + 11.9169\,s^2 + 57.1356\,s + 137.9793}{s^5 + 32\,s^4 + 969.3s^3 + 3044.5s^2 + 999.8s + 84.6}$$

$$F_{45} = \frac{0.1147\,s + 0.6118}{1000\,s^4 + 409.8s^3 + 2906.1s^2 + 1228\,s + 1219.6}$$

$$F_{46} = \frac{0.1147\,s + 0.6118}{1000\,s^4 + 409.8s^3 + 2906.1s^2 + 1228\,s + 1219.6}$$

$$F_{47} = \frac{0.1147\,s + 0.6118}{1000\,s^4 + 409.8s^3 + 2906.1s^2 + 1228\,s + 1219.6}$$

$$F_{51} = \frac{0.7534\;\;s^2 + 0.269\;\;s + 0.5844}{100000\;\;s^5 + 41690\;\;s^4 + 553060\;\;s^3 + 160380\;\;s^2 + 398810\;\;s + 164540}$$

$$F_{52} = \frac{0.3599\,s^2 + 0.0645\,s + 0.2422}{10000\,s^3 + 8507\,s + 1902}$$

$$F_{53} = \frac{0.3554\,s^2 + 0.058\,s + 0.2272}{10000\,s^3 + 8511\,s + 1512}$$

$$F_{55} = \frac{0.6914s + 3.8381}{0.0001s^5 + 0.0002s^4 + 0.1001s^3 + 0.1682s^2 + 7.5926s + 0.1947}$$

$$F_{56} = \frac{0.0358\,s^2 + 0.0675}{s^5 + 1.206\,s^4 + 2.1495\,s^3 + 0.8151\,s^2 + 1.4013\,s + 0.0238}$$

$$F_{57} = \frac{0.0076\,s + 0.0328}{s^3 + 0.5542\,s + 0.0605}$$

$$F_{61} = \frac{10^{-7}(0.0972\,s + 0.1052)}{s + 0.3455}$$

$$F_{62} = \frac{10^{-8}(0.5494\,s + 0.286)}{s^3 + 1.4651\,s^2 + 0.1428\,s + 0.1222}$$

$$F_{63} = \frac{10^{-7}(0.0972\,s + 0.1052)}{s + 0.3455}$$

$$F_{65} = \frac{10^{-5}0.2764}{s^2 + 0.9232\,s + 0.0948}$$

$$F_{66} = \frac{0.1569\,s^2 + 1.6696\,s + 180.9912}{s^3 + 27.2279\,s^2 + 102.9s + 178.35}$$

$$F_{67} = \frac{10^{-6}(0.4859\,s + 0.5259)}{s + 0.3455}$$

$$F_{71} = \frac{10^{-6}(0.3333\,s^2 + 0.1321\,s + 0.1785)}{s^3 + 2.2758\,s^2 + 3.0902\,s + 0.6192}$$

$$F_{72} = \frac{10^{-6}(0.3333\,s^2 + 0.1321\,s + 0.1785)}{s^3 + 2.2758\,s^2 + 3.0902\,s + 0.6192}$$

$$F_{73} = \frac{10^{-6}(0.3333\,s^2 + 0.1321\,s + 0.1785)}{s^3 + 2.2758\,s^2 + 3.0902\,s + 0.6192}$$

$$F_{75} = \frac{0.0038\,s^2 + 0.0015\,s + 0.002}{s^3 + 2.2758\,s^2 + 3.0902\,s + 0.6192}$$

$$F_{76} = \frac{0.0038\,s^2 + 0.0015\,s + 0.002}{s^3 + 2.2758\,s^2 + 3.0902\,s + 0.6192}$$

$$F_{77} = \frac{0.8309\,s^2 + 182.9324}{s^3 + 10.1109\,s^2 + 63.1865\,s + 177.2446}$$

$$F_{88} = \frac{0.0298\,s^2 + 1.3902}{0.0084\,s^3 + 0.0969\,s^2 + 0.6153\,s + 1.384}$$

5 CONCLUSIONES

- "Squaring-up" utiliza compensación MIMO-QFT sobre plantas con incertidumbre infra-actuadas, esto es, con más salidas que entradas, de tamaños *[2,n]*, con $n \geq 2$ y paridad *par*, en un proceso sin supresión de especificaciones del sistema a controlar y que, además, incorpora optimización del esfuerzo de control, lo que permite una mejor regulación y seguimiento de todo el conjunto de parámetros considerado, ya sea a efecto de acoplamientos o desacoplamientos.

- La bondad del método se ha puesto a prueba con el ejemplo complejo propuesto, que opera con un rango de incertidumbre elevado, para una gran cantidad de variables de salida y en situación combinada tracking-regulación. Sin embargo, sería interesante ampliar la teoría de "squaring-up" a sistemas de tamaños *[m,n]*, con $m>2$, $n>2$, $n>m$ y ambos con cualquier tipo de paridad. Observar que actualmente el método está restringido en este sentido y no es aplicable a plantas infra-actuadas de cualquier tamaño.

- La aplicación de "squaring up" en sistemas de control de aeronaves, donde es muy habitual encontrar pocas entradas de control sobre muchas variables de salida, podría resultar una solución, en lugar de eliminar parámetros de salida para adaptar su número al de las entradas, como suele hacerse.

REFERENCIAS

[1] Astrom, K.J., Murray, R. M., (2004), "*Feedback systems: an introduction for scientists and engineers*",DRAFT v0.915

[2] Boje E., (2005), "*Squaring down design for over-actuated systems in multivariable QFT*", IFAC.

[3] Borghesani, C., Chait, Y., Yaniv, O.,

(1995). "*Quantitative Feedback Theory Toolbox - for use with MATLAB*". The MathWorks Inc, Natick, MA.

[4] Cohen, N., Chait, Y., Yaniv, O., Borghesani, C., (1994), "*Stability Analysis using Nichols Charts*", International Journal of Robust and Nonlinear Control, 4, pp. 3-20.

[5] Garcia-Sanz, M., Egaña, I., (2002), "*Quantitative non-diagonal controller design for multivariable systems with uncertainty*", International Journal of Robust and Nonlinear Control, 12, pp. 321-333.

[6] GARTEUR Action Group FM(AG08)(1997). "*Robust flight control design challenge. Problem formulation and manual: the research civil aircraft model (RCAM)*", GARTEUR/TP-088-3.

[7] Horowitz, I. M., (1992). "*The Quantitative Feedback Theory, vol.1*", QFT Publishers, Boulder, Colorado

[8] Houpis, C. H., Rasmussen, S. J., (1999), "*Quantitative Feedback Theory: fundamentals and applications*". Marcel Dekker, New York.

[9] Houpis, C. H., (2002)" *Quantitative Feedback Technique (QFT): Bridging the Gap*", RTO-EN-SCI-142

[10] Kerr M., (2004) "*Robust control of an articulating flexible structure using MIMO QFT*", Thesis, University of Queensland.

[11] Magni J.F., Bennani S., Terlouns J. (eds) (1997) "*Robust flight control. A design challenge*". LNCIS 224. Springer 1997.

[12] Muñoz-Mansilla R., Aranda.J, Diaz J.M., Chaos D., (2012), "*Aplicación de técnicas de control robusto QFT a sistemas navales*", Revista Iberoamericana Automática e Informática industrial 9 (2012) 231–243

[13] Philips W.D.,(1989), "*Frequency sensitive QFT weighting matrix*", Thesis, AFIT/GAE/ENG/88D-01

[14] Yaniv, O., Schwartz, B., (1991), "*Criterion for loop stability in MIMO feedback systems having an uncertain plant*", International Journal of Control, 53(3), pp. 527-539.

[15] Yaniv, O., (1999), "*Quantitative feedback design of linear and nonlinear control systems*". Kluwer Academic Publishers, Norwell, Massachusetts

[16] Yousfi N., Melchior P., Rekik C., Derbel N., Oustaloup A.,(2012), "*Design of Centralized CRONE Controller Combined with MIMO-QFT Approach Applied to Non Square Multivariable Systems*", *International Journal of Computer Applications (0975–8887) Vol 45– N.16.*

Control de velocidad mediante paso de pala para un aerogenerador experimental de pequeña potencia

Sergio Fragoso Herrera
Departamento de Informática y Análisis Numérico, Universidad de Córdoba, Campus de Rabanales, Edificio Leonardo Da Vinci, Córdoba, España, e-mail: p52frhes@uco.es

Francisco Vázquez Serrano
Departamento de Informática y Análisis Numérico, Universidad de Córdoba, Campus de Rabanales, Edificio Leonardo Da Vinci, Córdoba, España, e-mail: fvazquez@uco.es

Fernando Morilla García
Departamento de Informática y Automática, ETSI Informática, UNED, C/. Juan del Rosal 16, Madrid, España, e-mail: fmorilla@dia.uned.es

Resumen

El diseño de sistemas de control de velocidad variable representa un reto importante ya que se trata de procesos multivariables no lineales, con fuertes perturbaciones, diversas restricciones y gran interacción entre sus variables. En este trabajo se presenta la problemática que genera el diseño de sistemas de control para aerogeneradores haciendo especial hincapié en las soluciones basadas en técnicas de control multivariable. El objetivo es aplicar una de estas técnicas a un aerogenerador experimental de pequeña potencia de velocidad y paso variable (VS-VP), el cual reúne una tecnología similar a los equipos de mayor potencia. En este trabajo se presenta el análisis y diseño del lazo de control de velocidad actuando sobre el mecanismo servomotor de ángulo de paso de pala.

Palabras Clave: control multivariable**,** velocidad variable, paso variable, aerogenerador experimental.

1. INTRODUCCIÓN

La energía eólica ha experimentado un rápido crecimiento desde finales de la década de los 90. En el año 1997 solamente había instalada a nivel mundial una capacidad de 7480 MW. Inicialmente esta capacidad fue incrementando alrededor de los 2000 MW por año hasta alcanzar los 42 GW instalados en 2011 [16]. En cuanto a generación, en España en 2011 la producción eólica representó un 16.33% de la cobertura de la demanda eléctrica [5]. Estas estadísticas muestran como la energía eólica es una tecnología consolidada y asentada dentro del mix de generación.

El aumento de la generación de energía eléctrica por medio de la fuerza del viento, no solo se asocia a un aumento de la capacidad instalada, sino también a una mejora de su funcionamiento. Los aspectos más importantes a considerar son: la generación eficiente de la potencia y la regulación de la velocidad angular bajo condiciones cambiantes del viento.

Esta mejora de la eficiencia de las plantas de producción eólica constituye unos de los principales problemas a los que se enfrentan las empresas energéticas. Entre los avances encaminados a resolver este problema se encuentra la instalación de aerogeneradores de velocidad variable, con los que se pretende extraer la máxima energía eléctrica posible a bajas velocidades del viento y mantenerla en sus valores nominales cuando se presentan fuertes vientos. Esto aporta muchos beneficios con respecto a los aerogeneradores de velocidad fija, como mejor calidad en la energía eléctrica producida, mayor rendimiento y una considerable reducción en las vibraciones y esfuerzos mecánicos, tanto en la turbina, como en el generador eléctrico [13]. Junto a la velocidad variable, la capacidad de modificación del ángulo de paso de pala aporta mayor flexibilidad y controlabilidad al aerogenerador permitiendo aplicar estrategias de control multivariable, mejorando así el rendimiento [4], [8], [9], [12].

En este trabajo se aborda el diseño de un lazo de control de velocidad sobre el aerogenerador de pequeña potencia, con vista a una futura inclusión en una estrategia de control multivariable. El documento está estructurado como se indica a continuación. En la sección 2 se describe el funcionamiento del aerogenerador y la problemática de control en las diferentes regiones de operación. En la sección 3 se muestra el equipo experimental y sus características principales, así como, el sistema de adquisición y control utilizado. En la sección 4 se analiza el lazo de control de velocidad partiendo de la identificación de

la planta y terminando con la presentación de varios resultados. Finalmente en la sección 5 se presentan las conclusiones y los posibles trabajos futuros.

2. FUNCIONAMIENTO DEL AEROGENERADOR Y PROBLEMÁTICA DE CONTROL

El modo de funcionamiento de todo aerogenerador es dependiente de la intensidad de la velocidad del viento. Fundamentalmente se distinguen tres regiones de operación [4], como se muestra en la figura 1. En la región I (carga parcial) el aerogenerador opera a potencia subnominal y el objetivo es extraer toda la energía posible con objeto de mejorar la eficiencia. En la región III (plena carga) el viento ha alcanzado una velocidad lo suficientemente alta para que la potencia producida se encuentre en niveles nominales. El objetivo en esta zona es mantener ese nivel de producción energética ante cualquier perturbación del viento para evitar posibles sobrecargas y fatigas estructurales. La región II es una región de transición entre ambas zonas.

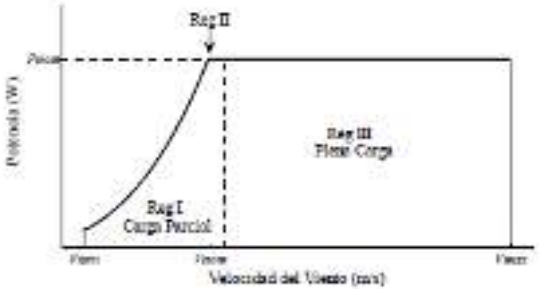


Figura 1: Regiones de operación en función de la velocidad del viento

La solución tradicionalmente adoptada en control de aerogeneradores consiste en el uso de controladores PID independientes en cada región de operación. En la región I los algoritmos MPPT (*Maximum Point Power Tracking*) [1] buscan una optimización del proceso generando puntos de consigna cambiantes para la velocidad ω_r y la potencia P_g, de forma que se consiga la máxima extracción de energía. En esta zona se suele aplicar un control de velocidad a través del par del generador τ_g, manteniendo el valor del ángulo de paso de pala β en el punto más adecuado para la optimización de la extracción de energía, figura 2.

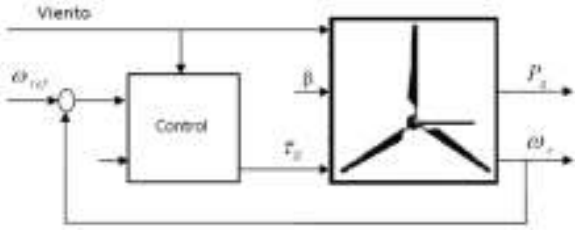


Figura 2: Control de velocidad en la región I

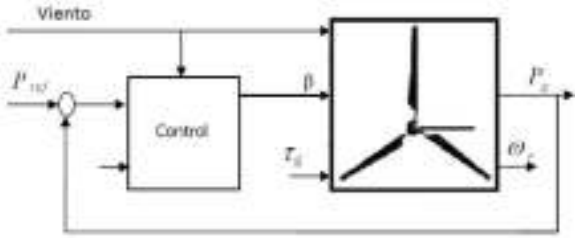


Figura 3: Control de potencia en la región III

En la región III es necesario mantener la potencia y la velocidad en sus valores nominales, mediante un controlador de potencia que modifica el ángulo de paso de pala y mantiene el par del generador en el valor nominal, figura 3.

Esta estrategia monovariable permite mantener al aerogenerador operando en cualquier punto. Sin embargo, ante cambios en la intensidad del viento, la potencia y la velocidad pueden exhibir un comportamiento inadecuado con amplias fluctuaciones, debido principalmente a la interacción existente no corregida con un único controlador.

Por tanto dado el "carácter multivariable" del proceso la línea de trabajo del grupo pretende aproximar un planteamiento de estrategia de control que cumple con las premisas [15]:

- Carácter MIMO, con una estructura similar mostrada por la figura 4, que incluya a la dos variables controladas (potencia y velocidad) así como las dos variables manipuladas (par y paso de pala)
- Puede ser empleado en las tres regiones de operación
- Permite incorporar una capa en un nivel superior, que incluye un algoritmo MPPT en la región I.

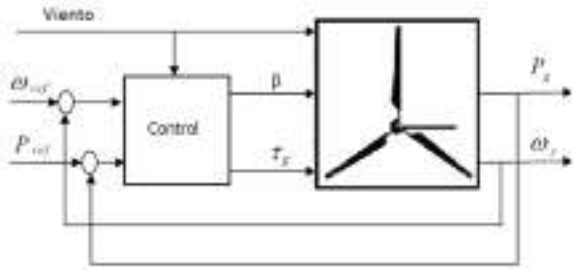


Figura 4: Control Multivariable del aerogenerador

3. EQUIPO EXPERIMENTAL

3.1 AEROGENERADOR DE PEQUEÑA POTENCIA

El aerogenerador de pequeña potencia empleado en este trabajo, figura 5 es el modelo EOLICC del fabricante Astrianne Didact. El equipo se compone de un rotor bihélice con acoplamiento directo a un generador de corriente continua de imanes permanentes, figura 6. Incorpora también un mecanismo de rotación de hélice a través de un servomotor de corriente continua de 10 W. El equipo dispone de un panel de potencia y un panel de control que permiten comandar al aerogenerador tanto en modo local (manual), como en modo exterior (conexión con sistema DAQ).



Figura 5: Imagen del aerogenerador



Figura 6: Detalle del rotor y generador eléctrico del aerogenerador

La tabla 1 recoge los principales parámetros del equipo. La torre junto a la góndola está situada en el interior de un pequeño túnel de viento y dispone de un motor-ventilador comandado por un variador de velocidad, el cual permite establecer las condiciones de viento, figura 6. Justo a la salida del ventilador, el flujo de aire pasa a través de una rejilla de fibra que transforma el caudal de aire turbulento en flujo laminar. Además dispone conjuntamente de unos deflectores laterales que canalizan el flujo de aire hacia el generador eólico, de manera que todo el flujo incide de forma transversal sobre el plano en el que se sitúa el rotor.

Tabla 1: Parámetros del aerogenerador de pequeña potencia

Parámetro	**Unidad**
Tensión de utilización	230 V
Frecuencia de utilización	50 Hz
Consumo eléctrico	2000 VA
Potencia motor-ventilador	1.1 KW
Diámetro del rotor	595 mm
Potencia nominal	80 W
Tensión nominal	24 V
Potencia máxima	94.8 W
Velocidad nominal	2300 rpm
Velocidad máxima	4000 rpm
Resistencia de armadura	1.44 Ω
Inductancia de armadura	0.56 mH
Inercia del rotor	1290 gcm^2
Constante de velocidad	95.3 rpm/V
Constante de par	100 mNm/A
Rango de velocidad de viento	0-42 km/h
Rango de ángulo de paso de pala	-2°-25°
Rendimiento máximo	79 %
Relación de transmisión	Acoplamiento directo

3.2 SISTEMA DE ADQUISICIÓN Y CONTROL

La adquisición y acondicionamiento de la señales de entrada y salida analógicas, así como, el aislamiento eléctrico y el rechazo al ruido electromagnético se efectúa por medio de unos módulos de acondicionamiento de la serie 5B de National Instruments insertados en un backplane 5B08 de 8 canales, del mismo fabricante. Este backplane se comunica con una tarjeta PCI a través de una placa de interfaz SC-2050 de National Instrument.

La arquitectura de control utilizada es la denominada PC-in-the-Loop, donde el ordenador actúa como controlador principal adquiriendo datos mediante una tarjeta serie PCI-6035E de National Instruments. La

tarjeta transmite en tiempo real los datos muestreados de sus canales hacia el software MATLAB/SIMULINK con la toolbox de tiempo real Real-Time Windows Target, que contiene el programa de control, figura 7.

El vector de señales de entrada analógicas se compone de las cuatro medidas que se realizan sobre el equipo: el ángulo de paso de pala (Pitch), la velocidad angular, la tensión generada a la salida del generador eléctrico y la medida de la señal del viento. Estas señales son capturadas con una frecuencia de muestreo de 100 Hz. Adicionalmente y como complemento al acondicionamiento previo que realizan los módulos de adquisición, cada señal de entrada analógica pasa a través de un filtro de primer orden discreto para la reducción del ruido de la señal, y a través de una etapa de adecuación de unidades.

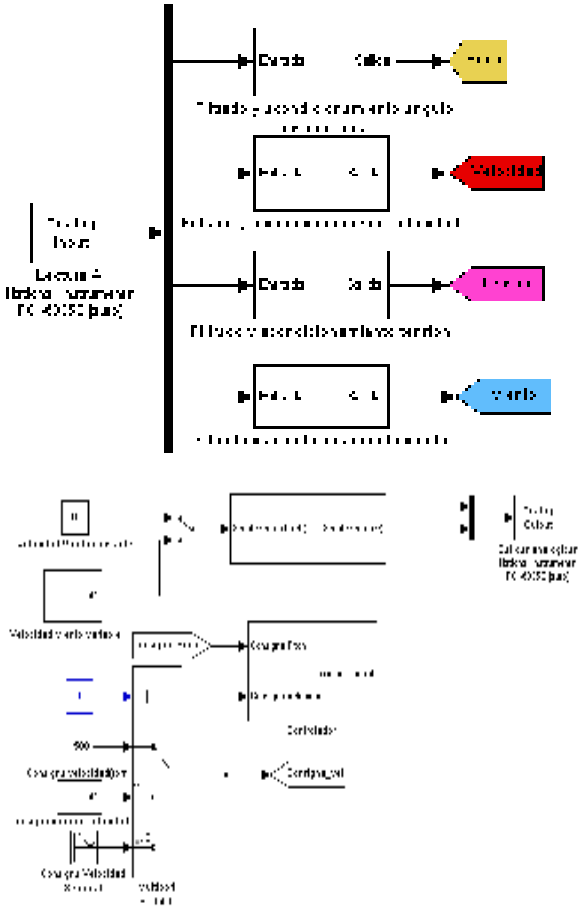


Figura 7: Diagramas de bloques del sistema real en Simulink

Para la captura de la señal de velocidad ha sido necesaria la incorporación de un módulo conversor de frecuencia en tensión, el cual recoge la frecuencia de los pulsos establecidos por el encoder NPN que mide la velocidad del eje de la turbina y lo convierte en una señal de tensión de 0-10V.

En cuanto a salidas analógicas, desde el controlador se proporciona la consigna de velocidad de viento y el ángulo de paso de pala. La modificación de este último puede ser realizada mediante dos vías, en lazo abierto o en lazo cerrad, este último inserto en el lazo de control de velocidad. La señal de velocidad del viento previamente es tratada para la adecuación de unidades y puede presentar una única componente continúa o comportarse como un viento variable. Las salidas analógicas son actualizadas con la misma frecuencia de 100 Hz.

4. CONTROL DE VELOCIDAD

El esquema de control mutlivariable propuesto en la figura 4, requiere del diseño de un lazo de control de velocidad y de un lazo de control de potencia. En esta sección se aborda la identificación y diseño del lazo de velocidad y se presentan los primeros resultados del control, considerando un planteamiento SISO donde no se efectúa ningún tipo de control sobre la potencia.

4.1 IDENTIFICACIÓN DE LA DINÁMICA DE VELOCIDAD

La figura 8 muestra la estructura de control de la velocidad angular ω_r. Existe un lazo exterior encargado de corregir el error de velocidad angular estableciendo consignas para un lazo interior. Este lazo interior regula la posición del ángulo de paso de pala β para influir sobre el comportamiento aerodinámico del equipo. De esta manera se consigue la velocidad angular deseada. Sobre la aerodinámica es donde se manifiesta de manera más abrupta los efectos de la perturbación del viento.

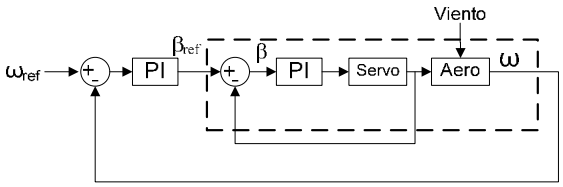


Figura 8: Esquema de control en cascada de la velocidad angular

La fase de identificación estuvo centrada fundamentalmente en la captura de la dinámica del bloque remarcado con línea discontinua en la figura 8. Este bloque es analizado como una caja negra donde se aplica un estímulo en forma de variación del ángulo de paso de pala y se mide el efecto provocado en la señal de velocidad angular. La justificación de esta identificación recae en buena medida en la correcta sintonía del lazo interior de control. Este subsistema capta la dinámica del

servosistema regulador del ángulo de paso de la pala. Algunos autores [2], [3] señalan que el bloque servomotor, generalmente hidráulico o neumático se comporta como un sistema integrador con una ganancia determinada. Tal afirmación fue corroborada en ensayos previos sobre el equipo. Con esta suposición sobre la dinámica del mecanismo servomotor se ha sintonizado un controlador con estructura PI de manera heurística mediante la técnica de Ziegler-Nichols y a través de la herramienta de sintonía de controladores SISO [14]. La sintonía busca una respuesta en lazo cerrado sobreamortiguada. La ganancia proporcional obtenida del controlador interior es $K_{p1} = 6$ grados/V y el tiempo integral $T_{i1} = 5$ s, con una saturación de ±10 V y un slew-rate de ±7 V/s.

La identificación se efectúa en varios puntos de trabajo buscando un modelo representativo aproximado. Los puntos de operación están definidos por la velocidad del viento y por el margen del ángulo de paso de pala que cubre la señal estímulo. El número de identificaciones efectuadas abarca aproximadamente el rango completo de variación del ángulo de paso de pala del equipo (-2° a 25°) para varias velocidades de viento constante. Estas velocidades se seleccionaron para que el aerogenerador siempre trabajara por debajo de su velocidad angular nominal de 2300 rpm. Las figuras 9 y 10 muestran el ajuste de un modelo de primer orden sin retardo y su correspondiente validación en varios puntos de la identificación.

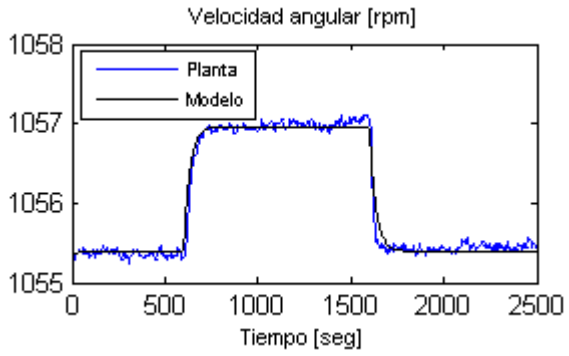


Figura 9: Ajuste y validación de un modelo de primer orden para una velocidad del viento de 8.33 m/s y una zona de ángulo de paso de pala de 17°-14°-17°

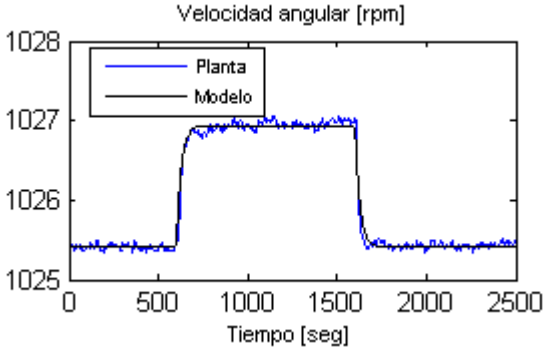


Figura 10**:** Ajuste y validación de un modelo de primer orden para una velocidad del viento de 8.88 m/s y una zona de ángulo de paso de pala de 19°-16°-19°

Mediante el análisis de los diferentes modelos parciales obtenidos, se ha considerado un modelo medio como modelo representativo de la dinámica de velocidad angular. El modelo es de primer orden sin retardo del tipo

$$G(s) = \frac{a}{1 + bs} \qquad (1)$$

Con $a = -0.48$ y $b = 32.38$

4.2 LAZO DE CONTROL DE VELOCIDAD ANGULAR

El tiempo de asentamiento del modelo-sistema en lazo abierto es de aproximadamente 160 s. El sistema en lazo cerrado junto con el controlador ha sido diseñado para obtener un tiempo de asentamiento similar, para impedir que el controlador intente hacer al sistema más rápido frente a un cambio, generando señales de control elevadas que pueden provocar un sobrepaso excesivo no deseado sobre la consigna.

A través de la herramienta software de diseño paramétrico de controladores [10] se ha sintonizado un controlador de estructura PI con la especificación de tiempo de asentamiento requerida. El controlador diseñado presenta una ganancia proporcional de $K_{p2} = -0.81$ Hz/grados y un tiempo integral de $T_{i2} = 11$ s.

Una característica importante de los resultados reales es que el sobrepaso de la señal de velocidad en la dinámica de subida no supone más de un 5% de la referencia. Esto refleja que con este controlador se ha llegado a una situación de compromiso entre la velocidad de respuesta y el tipo de respuesta ofrecida en lazo cerrado.

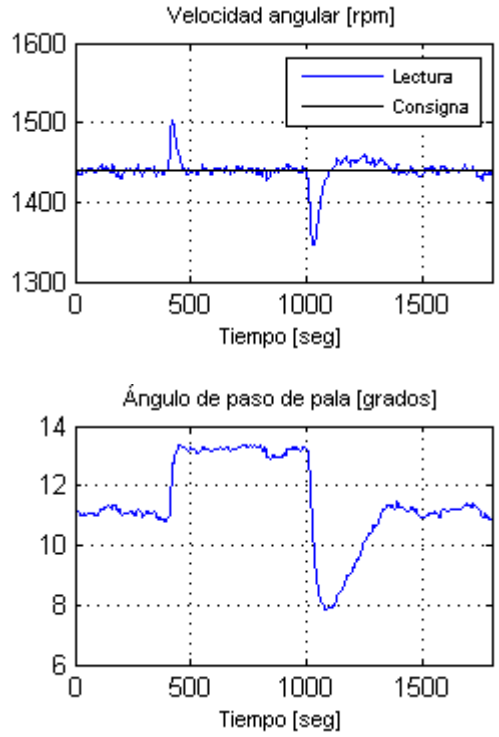
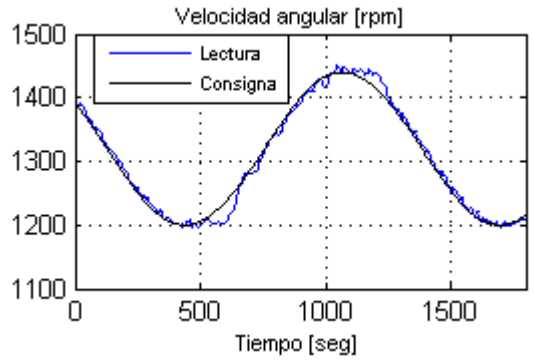


Figura 11: Salida y señal de control del aerogenerador frente a una perturbación provocada por una variación de la velocidad del viento

El sistema conjuntamente con el controlador es de tipo I, contiene un polo en el origen y por tanto es capaz de seguir referencias, figura 11 y rechazar perturbaciones a la salida provocadas por el viento, figura 12. En la figura 11 el aerogenerador es sometido a un viento test con una componente continua de 8.33 m/s, la cual presenta un primer salto escalón positivo de 0.55 m/s de amplitud a los 400 s de ensayo, y un salto negativo de -0.55 m/s en los 1000 s. El controlador intenta mantener una consigna de velocidad de 1440 rpm.



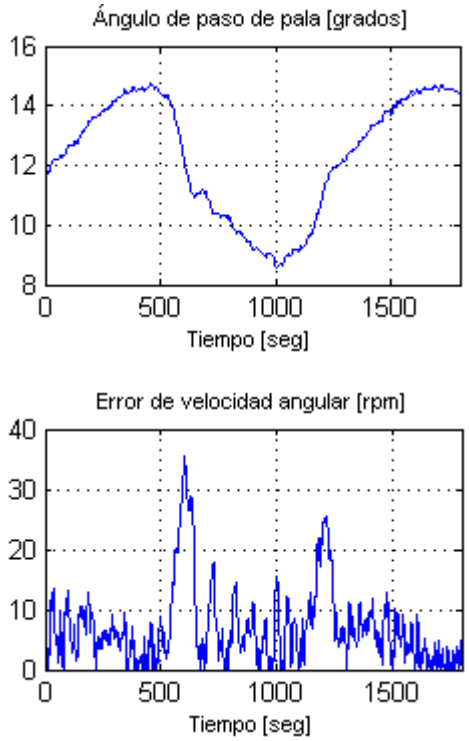


Figura 12: Salida, señal de control y valor absoluto del error de un seguimiento de consigna del aerogenerador

La figura 12 muestra el desempeño del controlador frente a un seguimiento de referencia senoidal oscilante entre 1440 rpm y 1200 rpm, con una componente de viento media de 8.33 m/s. A pesar del continuo cambio de la consigna, la media del error mantiene un valor bajo de 7.2 rpm.

5 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha presentado un aerogenerador VS-VP experimental de tecnología similar a los equipos de mediana y gran potencia, y sobre el cual es posible aplicar diferentes estrategias de control y extrapolar los resultados hacia otros equipos de mayor potencia.

En este trabajo, con el proceso de identificación de la velocidad angular-ángulo de paso de pala se ha llegado a la obtención de un modelo intermedio de primer orden sin retardo que ha permitido diseñar el controlador de velocidad. Tanto en simulación, como en los correspondientes ensayos realizados sobre el equipo experimental ha quedado de manifiesto como se ha conseguido la sintonía de un único controlador capaz de estabilizar al sistema frente a cambios de referencia en zonas de operación subnominal.

El proceso en estudio se considera como un sistema multivariable, por lo que una estrategia de control de estas características puede mejorar el funcionamiento del sistema. Como trabajo futuro se plantea el análisis y aplicación de una estrategia de control multivariable, con el diseño de la electrónica adicional para conseguir un control de potencia. Además se plantea el estudio de técnicas de minimización de interacción entre variables de control a través de redes de desacoplo [6], [7].

Agradecimientos

Este trabajo ha sido financiado por el proyecto de Excelencia P10-TEP-6056 Junta de Andalucía, España.

Referencias

[1] Barakati, S., Kazerani, M., Aplevich, J. (2009). "Maximum Power Tracking for the Wind Turbine Systems including a Matrix Converter". *IEEE Transactions on Energy Conversion*, vol 24, no 3, pp 705-713.

[2] Bianchi, F., D. Battista, H, et al. (2007). *Wind Turbine Control Systems Principles: Modelling and Gain Scheduling Design*. Germany, Springer

[3] Bianchi, F., Mantz R.J., et al. (2004). "Power Regulation in Pitch-Controlled Variable Speed WECS Above Rated Wind Speed". *Renewable Energy*, vol 29,pp 1911-1922.

[4] Boukhezzar, B., Lupu L., et al. (2007). "Multivariable Control Strategy for Variable Speed, Variable Pitch Wind Turbine". *Renewable Energy*, vol 32, pp 1273-1287

[5] European Wind Energy Association (2011). *Pure Power. Wind Energy Targets for 2020 and 2030*

[6] Garrido, J., Vázquez, F., Morilla, F. (2011). "An Extended Approach of Inverted Decoupling". *Journal of Process Control*, vol 21, pp 55-68.

[7] González, M., Garrido, J., et al (2011). "Multivariable Centralized Control with Decoupling Feedforward Compensation for Residential Wind Turbine". *18th World Congress The International Federation of Automatic*. Milan (Italia).

[8] González, M., Vázquez, F., Morilla, F. (2010). "Control Multivariable Centralizado con Desacoplo para Aerogeneradores de Velocidad Variable". *Revista Iberoamericana de Automática e Informática Industrial*, vol 7, pp 53-64

[9] Haque, M., Negnevitsky, M. (2010). "A Novel Control Strategy for a Variable-Speed Wind Turbines with a Permanent Magnet Synchronous Generator". *IEEE Transactions on Industry Applications*, vol 46.

[10] Morilla, F., Vázquez, F., et al (2006). "PID Control Design with Guaranteed Stability". *7th IFAC Symposium on Advances in Control Education*

[11] Muljadi, E., Butterfield, C.P., et al (1996). "Variable Speed Operation of Generators with Rotor-Speed Feedback in Wind Power Applications". *15th ASME Wind Energy Symposium*. Houston (Texas).

[12] Nagai, B., Ameku, K., et al (2009). "Performance of a 3kW Wind Turbine Generator with Pitch Control System". *Applied Energy*, vol 86, pp 1774-1782.

[13] Rodriguez, J.L., Burgos, J.C., et al (2003). *Sistemas Eólicos de Producción de Energía Eléctrica*. Editorial Rueda.

[14] Vázquez, F., Morilla, F., Sanmiguel, E. (2000). "*Entorno Matlab para Diseño de Controladores PID*". XXI Jornadas de Automática. Sevilla (España)

[15] Vázquez, F., González, M., Garrido, J., Morilla, F. (2012). "Control Multivariable: aplicación al control de un aerogenerador". *X Simposio CEA de Ingeniería de Control*. Barcelona (España).

[16] World Wind Energy Association (2011). *World Wind Energy Report 2010*. WWEA Head Office. Bonn (Alemania).

Control multivariable descentralizado de un aerogenerador experimental de pequeña potencia

Sergio Fragoso Herrera
Departamento de Informática y Análisis Numérico, Universidad de Córdoba, Campus de Rabanales, Edificio Leonardo Da Vinci, Córdoba, España, e-mail: p52frhes@uco.es

Francisco Vázquez Serrano
Departamento de Informática y Análisis Numérico, Universidad de Córdoba, Campus de Rabanales, Edificio Leonardo Da Vinci, Córdoba, España, e-mail: fvazquez@uco.es

Fernando Morilla García
Departamento de Informática y Automática, ETSI Informática, UNED, C/. Juan del Rosal 16, Madrid, España, e-mail: fmorilla@dia.uned.es

Resumen

En este trabajo se presentan los primeros resultados de una estrategia de control multivariable descentralizada aplicada sobre un aerogenerador experimental de pequeña potencia. El equipo tecnológicamente pertenece a los sistemas de velocidad variable y paso variable (VS-VP) e incorpora un generador eléctrico de corriente continua de imanes permanentes (PMG). El trabajo plantea la problemática del control del lazo de potencia de generación, y como mediante una extrapolación de los métodos utilizados en equipos de gran potencia se llega a una solución para lograr una regulación eficiente de la misma y conseguir una capacidad de actuación multivariable. En el trabajo también se muestran una serie de resultados de ensayos de la estrategia multivariable diseñada operando con el lazo de potencia conjuntamente con un lazo de control de velocidad.

Palabras Clave: control de potencia, control multivariable, aerogenerador experimental, control PWM.

1. INTRODUCCIÓN

El diseño de sistemas de control para aerogeneradores de velocidad variable presenta una dificultad importante en cuanto a su control, ya que se trata de procesos multivariables con grandes no linealidades, con fuertes perturbaciones, diversas restricciones y gran interacción entre sus variables. Bajo este escenario se debe generar eficientemente la potencia eléctrica P_g y al mismo tiempo regular la velocidad de giro de la turbina ω_r [14]. Las estrategias de control multivariable aportan muchos beneficios con respecto a las estrategias de control convencionales, tales como la reducción de las fluctuaciones en la potencia y velocidad frente a cambios en la intensidad del viento. Los autores en [6] analizan en más detalle la problemática de control de este tipo de sistemas y presentan los objetivos de la línea de investigación enmarcada en el campo del control multivariable de aerogeneradores. Este trabajo representa una primera aproximación de una de estas técnicas de control multivariable sobre un equipo experimental, abordando inicialmente el problema que plantea la necesidad de añadir un lazo de control de potencia.

Los sistemas de conversión de energía eólica presentan dos modos de operación de acuerdo a como se establezca su conexión con la red eléctrica. En los aerogeneradores de velocidad fija, el sistema está directamente conectado a la red y su velocidad rotórica está impuesta por la frecuencia de la red. En los sistemas de velocidad variable un convertidor electrónico se inserta entre el generador y la red [10], [12], por lo que la velocidad rotórica puede cambiar independientemente de la frecuencia de la red.

Los objetivos de regulación de un aerogenerador cambian de acuerdo a la velocidad del viento [5]. A velocidades del viento bajas, se persigue extraer la máxima cantidad de energía posible. Es en este punto donde el modo de velocidad variable es especialmente útil, puesto que el convertidor electrónico ayuda a maximizar la eficiencia de la conversión mediante un control del par del generador [4].

Los algoritmos MPPT (*Maximum Power Point Tracking*) buscan una optimización del proceso de generación eólica generando puntos de consigna cambiantes, que sitúen al aerogenerador en el punto de rendimiento máximo. Estas estrategias de control recurren frecuentemente al uso de variadores de frecuencia, convertidores de corriente continua (buck,

boost, cuk) o montajes más complejos (montajes en tipo puente, back-to-back) [2], [11], [13], que actúan sobre el elemento eléctrico del sistema, dado que el modo de operación a velocidad variable requiere de una velocidad variable del generador eléctrico.

El documento está estructurado de la siguiente manera: en la sección 2 se muestra la estructura de control multivariable planteada. En la sección 3 se presentan las principales características del aerogenerador experimental, así como, del sistema de adquisición y control. En la sección 4 se analiza del tipo de convertidor diseñado para poder modificar el punto de operación del sistema. La sección 5 muestra la topología utilizada para el diseño del circuito modulador de carga. La sección 6 presenta los lazos de control implementados y los primeros resultados de la estrategia de control multivariable aplicada. Finalmente en la sección 7 se presentan las conclusiones y los posibles trabajos futuros.

2. SOLUCIÓN DE CONTROL MULTIVARIABLE

La estrategia de control multivariable sigue la estructura planteada en [6]. Esta estrategia presenta dos salidas a controlar, potencia y velocidad, y dos entradas manipuladas, que son el ángulo de paso de pala y el par generado. La velocidad se controla mediante el ángulo de paso de pala (Pitch), y la potencia a su vez debe ser controlada mediante una modificación del par de salida. En un generador de corriente de corriente continua el par de salida es directamente proporcional a la corriente generada y es esta la que debe ser manipulada en último término. Debido a la simplicidad del generador eléctrico, la modificación de la corriente no resulta tan directa.

Casi la totalidad de los aerogeneradores de mediana y gran potencia disponen de generadores eléctricos de corriente alterna (asíncronos o síncronos). El control de potencia en este tipo de sistemas se lleva a cabo mediante técnicas de control vectorial que permiten modificar la corriente, tensión y frecuencia de salida a través de un conjunto rectificador-convertidor-inversor para que el sistema trabaje sobre el punto de operación deseado [2], [3], [9].

En sintonía con el procedimiento de control de los equipos de gran potencia y extrapolando la problemática de estos al equipo experimental de pequeña potencia, el uso de un convertidor electrónico es una posible solución a la problemática de control de potencia. Este convertidor busca una modificación de la corriente que circula a través de la carga i_a controlando su ciclo de trabajo α. Esta modificación de la corriente conlleva inherentemente una variación de la carga aparente vista por el generador eléctrico y por tanto de su tensión de salida V y en consecuencia la potencia generada [1]. Visto desde el controlador multivariable, la variable α pasa a reflejarse como una variable manipulada en sustitución de la corriente generada.

Con este convertidor basado en la implantación de un circuito electrónico con topología de amplificador en inversión de tensión con semiconductor de potencia MOSFET, es posible realizar una modulación de la carga del generador eléctrico, figura 1. El circuito dispone de un microcontrolador de bajo coste que realiza la función de generación de la señal PWM de conmutación del MOSFET.

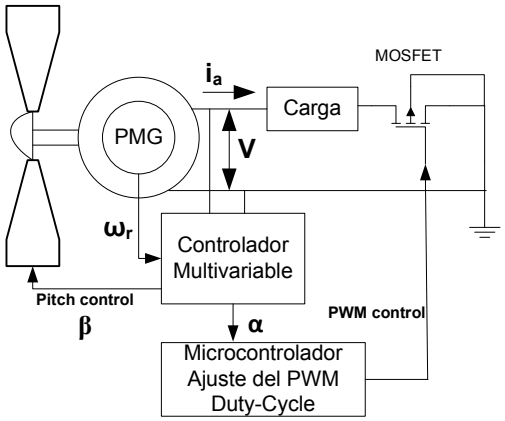


Figura 1: Esquema general de control multivariable

3. EQUIPO EXPERIMENTAL

El aerogenerador de pequeña potencia empleado en este trabajo, figura 2 es el modelo EOLICC de velocidad y paso variable del fabricante Astrianne Didact. El equipo se compone de un rotor bihélice con acoplamiento directo a un generador de corriente continua de imanes permanentes trabajando en modo stand-alone. Incorpora también un mecanismo de rotación de hélice a través de un servomotor de corriente continua. El rotor está situado en el interior de un pequeño túnel de viento que permite canalizar todo el flujo de aire en dirección transversal al plano del rotor. Incorpora un ventilador comandando por un variador de velocidad que permite establecer las condiciones de viento. En [6] se recoge de manera más detallada los parámetros característicos y de diseño de este aerogenerador.



Figura 2: Aerogenerador experimental

Sobre el aerogenerador se miden cuatro señales analógicas: ángulo de paso de pala, velocidad de rotación, tensión generada y velocidad del viento. Las señales son capturadas mediante módulos de entrada analógicos de la clase 5B de National Instruments insertados sobre un backplane analógico del mismo fabricante. Estos módulos transmiten la señal a una tarjeta serie PCI-6035E de National Instruments insertada en un PC. El ordenador actúa de controlador a través de la estrategia de control en tiempo real desarrollada mediante la toolbox de Simulink Real-Time Windows Target. La figura 3 muestra una parte del programa de control desarrollado en Simulink.

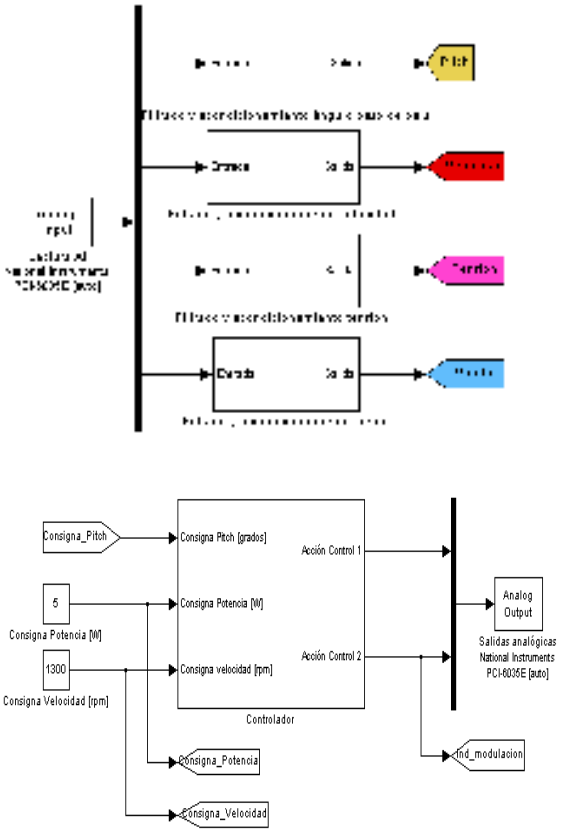


Figura 3: Diagramas de bloques del sistema real en Simulink

4. VARIACIÓN DEL PUNTO DE OPERACIÓN

El generador eólico es capaz de operar en lazo abierto a cualquier velocidad angular constante, siempre y cuando se den unas condiciones de viento favorables. En esta situación de régimen estacionario (1)-(5), por un lado el par aerodinámico τ_a es completamente compensado por el par de fricción mecánica en suma con el par electromagnético generado τ_{em}, y por otro lado la fuerza electromotriz del generador eléctrico es igual a la suma de caídas de tensión en la resistencia de carga R_L y armadura R_a.

$$0 = \tau_a - B_t\omega_r - \tau_{em} \quad (1)$$

$$\tau_a = \frac{1}{2\omega_r}\rho\pi R^2 C_p(\lambda,\beta)v^3 \quad (2)$$

$$\lambda = \frac{\omega_r R}{v} \quad (3)$$

$$0 = k\varphi\omega_r - \alpha R_L i_a - R_a i_a \quad (4)$$

$$\tau_{em} = \frac{V^2}{R_L\omega_r} \quad (5)$$

B_t es el coeficiente de fricción viscosa total, R el radio del rotor, ρ es la densidad del aire, $C_p(\lambda,\beta)$ es el coeficiente de potencia, v es la velocidad del viento, λ es el coeficiente de velocidad específica, k es la constante de proporcionalidad del generador eléctrico, φ es la constante de acoplamiento del flujo magnético del generador eléctrico y V es la tensión de salida del generador eléctrico.

En esta situación existe una condición de ligadura entre el ángulo de paso de pala β, la corriente de armadura i_a y el índice de modulación α para alcanzar una velocidad angular ω_r de equilibrio.

Las estrategias de control implantadas en los sistemas de gran potencia persiguen desplazar el punto de operación del sistema para poder alcanzar un punto de operación a máxima potencia. Alguna de ellas basan su desarrollo en el conocimiento de la característica de rendimiento del sistema (C_p). Existen diversas topologías de control con la misma finalidad [1]:

- Control del coeficiente de velocidad específica λ
- Control feedback de potencia
- Control por perturbación y observación (P&O) o hill-climbing searching (HCS)

Todas ellas al final generan una señal de error que entra al controlador encargado de regular el

convertidor electrónico. La finalidad última de este convertidor es modificar las condiciones eléctricas, para que se produzca una variación del par electromagnético generado, y se alcance un punto diferente estacionario con otra velocidad y otra potencia.

El modelo estacionario no lineal planteado en (1)-(5) no presenta solución analítica. Las curvas de potencia y par de un aerogenerador de características similares al equipo experimental se muestran en las figuras 4 y 5. Estas representan la solución gráfica del modelo para diferentes velocidades de viento Se observa que para cada velocidad del viento existe un punto específico estable donde la potencia de salida es máxima.

La intersección de la curva de par generador y la curva de par aerodinámico determina el punto de operación del aerogenerador. Por tanto para situar al aerogenerador en diferentes puntos de operación estables es necesario insertar un control sobre el generador eléctrico que permita modificar la curva de par de este.

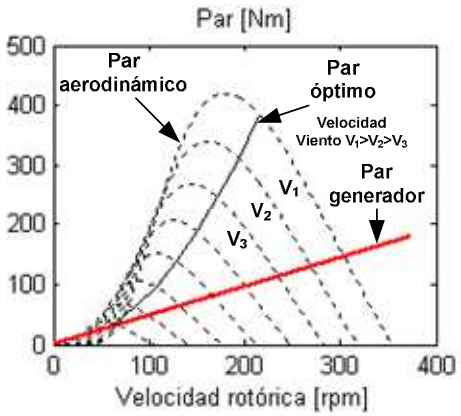


Figura 4: Característica par-velocidad de la turbina y el generador eléctrico para diferentes velocidades del viento

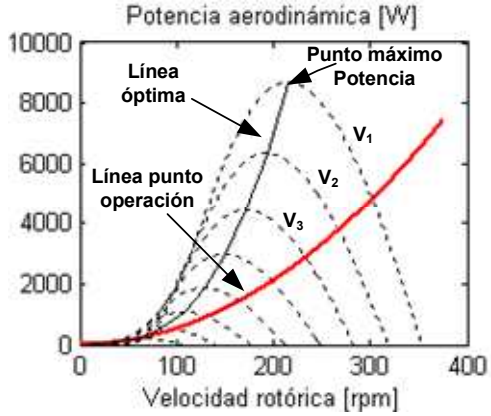


Figura 5: Curvas de potencia a varias velocidades del viento

El aerogenerador experimental, como se ha comentado, incorpora un generador eléctrico de corriente continua de imanes permanentes. Este generador eléctrico tecnológicamente es simple, como se indicaba en la sección 2, lo cual realmente dificulta la capacidad de control sobre el mismo. No existe ninguna variable intrínseca del generador que pueda ser manipulada para producir un cambio en las condiciones de corriente del mismo, como es el caso por ejemplo del flujo de corriente de campo en generadores síncronos.

En este tipo de generador eléctrico la ubicación de la curva de par generado está estrechamente ligada tanto con la corriente generada, como con el valor de la resistencia de carga acoplada. En la figura 6 se observa el efecto que tiene sobre la curva de par generado una variación del valor de resistencia acoplada. Este efecto es similar al producido por una variación de la corriente generada.

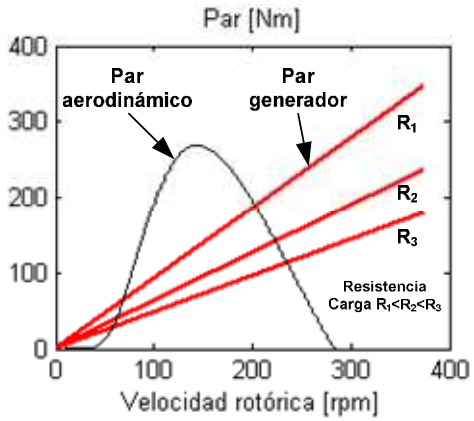


Figura 6: Efecto de la variación de la resistencia de carga

Desde un punto de práctico, la resistencia de carga no puede ser considerada como una variable manipulada en el sistema de control multivariable, puesto que generalmente la carga del generador eléctrico en sistemas conectados a red o en modo stand-alone, es una imposición al sistema. Por este motivo es necesario que el control encargado de desplazar la curva de par sea capaz de modificar las condiciones de carga aparente (virtual) vista desde el generador eólico. La modificación de la carga aparente, siendo la carga real puramente resistiva como es el caso, está ligada, bien a una variación en la tensión aplicada a la resistencia, o bien a una variación de la intensidad que circula por la misma. Es por este motivo por el que se justifica que el circuito diseñado para el control de potencia busca una modificación de la corriente de armadura i_a que circula por la resistencia de carga.

El controlador multivariable genera la señal de control que se ha denominado índice de modulación de carga α. Esta señal como se ha indicado en la sección 2 se corresponde con el ciclo de trabajo del circuito de conmutación, pero como se observa en el modelo no lineal estacionario, conceptualmente y desde un punto de vista matemático es tratada también como un porcentaje de reducción de la corriente de salida del generador eléctrico.

5. CIRCUITO MODULADOR DE CARGA

La topología del circuito modulador implementada se basa en un montaje como amplificador en inversión de tensión a través de transistor MOSFET. Este montaje persigue generar un corte (similar a un montaje chopper) en la intensidad circulante por la resistencia de carga a frecuencias lo suficientemente elevadas para disminuir el rizado que se pudiera ocasionar en la tensión de salida y en la resistencia aparente.

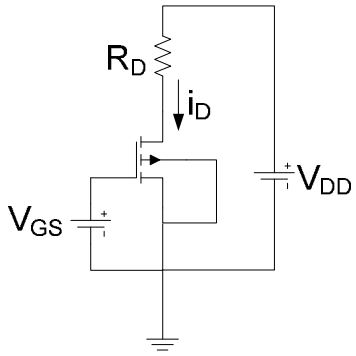


Figura 7: Topología del circuito modulador de carga

El esquema del circuito aparece en la figura 7. En este esquema R_D representa la resistencia de carga acoplada al generador eléctrico, i_D es la corriente de armadura o de salida del generador, V_{DD} es la fuerza electromotriz del generador eléctrico o tensión de salida y V_{GS} es la tensión puerta-fuente del semiconductor.

Un aspecto que es necesario a tener en cuenta en la selección de la topología es la ubicación del semiconductor en el montaje. Es necesario que toda la intensidad disponible circule por la carga para evitar tener un dato erróneo del nivel de generación del sistema. Por este motivo es necesario ubicar el semiconductor en serie con la resistencia de carga.

El semiconductor utilizado es el MOSFET de acumulación IRFZ44N. Es un transistor de potencia con buena respuesta a la conmutación, capaz de trabajar a frecuencias elevadas y con unos límites máximos de funcionamiento muy por encima de los requerimientos del circuito. El MOSFET trabaja en modo conmutación situándose continuamente entre el modo de corte y saturación. La lógica de conmutación se establece a través de la tensión V_{GS} y es una tarjeta de control Arduino la que genera tal señal a través de la técnica PWM. La tarjeta de control recibe por parte de controlador multivariable una señal de control intermedia que establece el ciclo de trabajo de la señal PWM. De esta forma, jugando con los tiempos de corte y saturación se puede lograr una modulación de carga con bastante resolución.

Arduino Duemilanove es una placa con microcontrolador basada en el ATmega168. Dispone de 14 pines con entradas/salidas digitales, 6 de los cuales pueden ser utilizadas como salidas PWM capaces de operar con una resolución de 10 bits, lo que supone alcanzar 1024 puntos en la variable manipulada, a una frecuencia de aproximadamente 1.9 KHz. En la figura 8 se observa el montaje del prototipo del circuito, como banco experimental de ensayos. Destacar la selección de varias resistencias variables de potencia que permiten experimentar muchas configuraciones diferentes de carga.

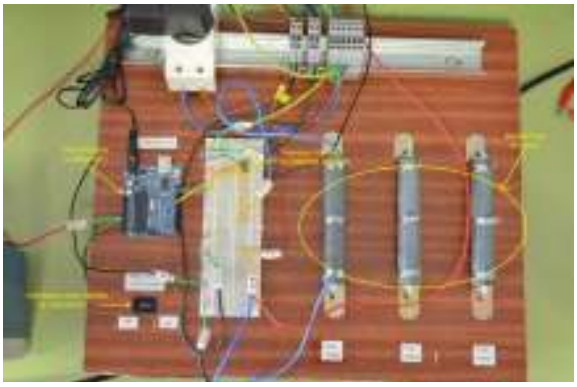


Figura 8: Banco de pruebas experimental

6. RESULTADOS EXPERIMENTALES CON LA ESTRATEGIA MULTIVARIABLE

6.1. CONTROL DE POTENCIA

El esquema de control de potencia P_g se presenta en la figura 9. La potencia generada se mide como potencia calorífica disipada en una carga puramente resistiva (6).

$$P_g = \frac{V^2}{R_L} \qquad (6)$$

El controlador en base al error de potencia cometido genera la señal intermedia de control α dirigida hacia el circuito modulador de carga. Esta señal, como se ha comentado, establece el ciclo de trabajo del

semiconductor generando un efecto de conmutación en la intensidad de salida. Este proceso permite generar una resistencia aparente determinada y por tanto efectuar un desplazamiento indirecto de la curva de par electromagnético que permita modificar el punto de operación del sistema.

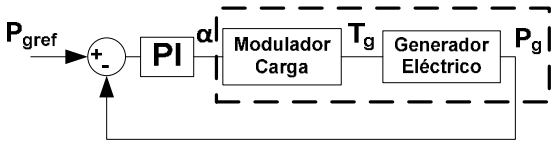


Figura 9: Esquema de control de potencia generada

La sintonía del controlador se ha realizado en base a un modelo identificado en un punto de trabajo de la dinámica enmarcada con línea discontinua en la figura 9. La identificación ha sido realizada sobre un punto de trabajo con una velocidad del viento de 30 Km/h (8.33 m/s) y un ángulo de paso de pala de unos 14°. Inicialmente se estimula al sistema con un índice de modulación α del 70% y se deja evolucionar hacia un estacionario. Seguidamente se incrementa un 10% la variable manipulada hasta alcanzar el nuevo estacionario. Una vez estabilizado se reduce un 10% hasta alcanzar de nuevo el punto de operación inicial.

El modelo se ha aproximado a un modelo de primer orden sin retardo (7). La figura 10 muestra la validación del modelo obtenido en comparación con la salida de la planta.

$$\frac{P_g(s)}{\alpha(s)} = \frac{K_1}{1+\tau_1 s} \tag{7}$$

$K_1 = 0.0371$ y τ_1=1

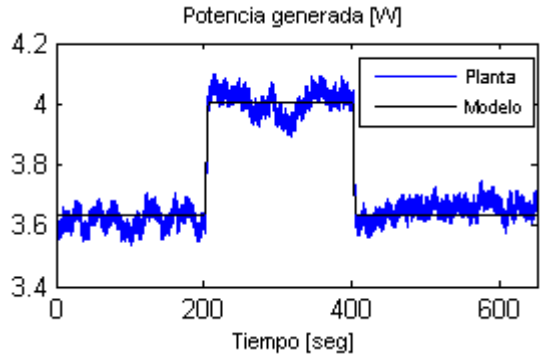


Figura 10: Ajuste y validación del modelo de primer orden para una velocidad del viento de 8.33 m/s, un índice de modulación entre 70%-80%-70% y un pitch de 14°

El controlador ha sido sintonizado para obtener una respuesta en lazo cerrado sobreamortiguada con una ganancia proporcional $K_{p1} = 0.5\,W/\%$ y un tiempo integral $T_{i1} = 1\,s$.

5.2 CONTROL DE VELOCIDAD

La figura 11 muestra la estructura de control de la velocidad angular. Este lazo de regulación ha sido desarrollado en profundidad por los autores en [6]. En él existe un lazo exterior encargado de corregir el error de velocidad angular ω_r estableciendo consignas para un lazo interior. Este lazo interior regula la posición del ángulo de paso de pala β, actuando sobre el servomecanismo de rotación de las palas, para influir sobre el comportamiento aerodinámico del equipo, intentado conseguir la velocidad deseada. Sobre la aerodinámica es donde se manifiesta de manera más abrupta los efectos de la perturbación del viento

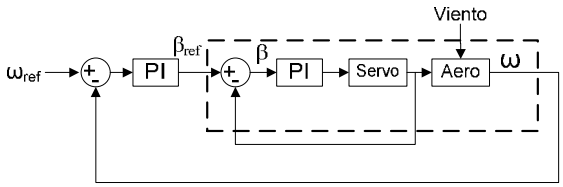


Figura 11: Esquema de control en cascada de la velocidad angular

El modelo obtenido representa la dinámica enmarcada con línea discontinua en la figura 11. Al igual que en el lazo de potencia, se ha considerado un modelo de primer orden sin retardo (8). Este modelo representa una dinámica promedio de diferentes modelos obtenidos en varios puntos de operación.

$$\frac{\omega_r(s)}{\beta_{ref}(s)} = \frac{K_2}{1+\tau_2 s} \tag{8}$$

$K_2 = -0.48$ y $\tau_2 = 32.38$

5.3 RESULTADOS DEL CONTROL MULTIVARIABLE

Con el ensayo planteado se puede observar como la estrategia de control multivariable descentralizada es capaz de estabilizar al sistema en diferentes puntos de operación.

El perfil de viento generado para el ensayo se compone de un viento medio de 30 Km/h (8.33 m/s) con una intensidad de turbulencia de aproximadamente 3 %, figura 12.

El aerogenerador parte de unas condiciones de carga resistiva real fija de 22 Ω y una resistencia aparente (virtual) cercana a los 40 Ω, con un índice de modulación del 63%, figura 13. Inicialmente se deja estabilizar al sistema en un punto de operación de velocidad rotórica inferior a la velocidad nominal (2300 rpm). El aerogenerador en este punto trabaja a

una velocidad rotórica de 1075 rpm y una potencia disipada en la resistencia de 3 W.

En torno a los 100 segundos de ensayo se efectúa un cambio de consigna de velocidad estableciendo el nuevo valor en las 1200 rpm. Se observa como el control de pitch rápidamente actúa disminuyendo el ángulo de paso de pala para alcanzar la nueva consigna. En el lazo de potencia se aprecia una ligera perturbación, que el controlador de par rechaza disminuyendo el índice de modulación.

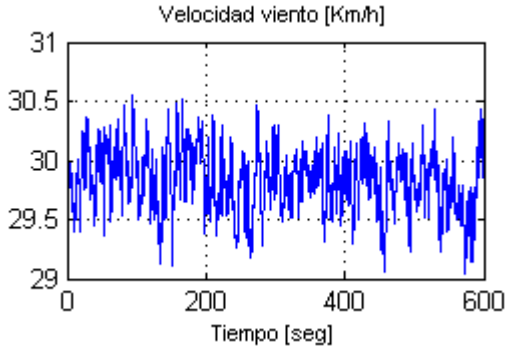
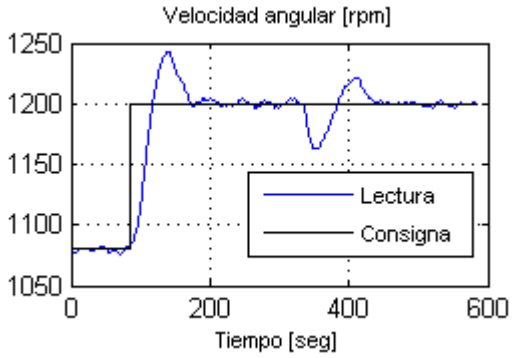


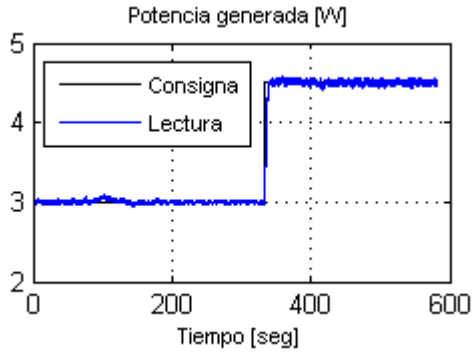
Figura 12: Perfil de velocidad de viento

El cambio de consigna de velocidad obliga a situar al sistema en un punto de operación estable de mayor velocidad, sin perder la consigna de generación de potencia. Para solventar este problema se observa claramente como el circuito modulador modifica la resistencia aparente vista por el generador, aumentándola para contrarrestar el efecto de aumento de la velocidad.

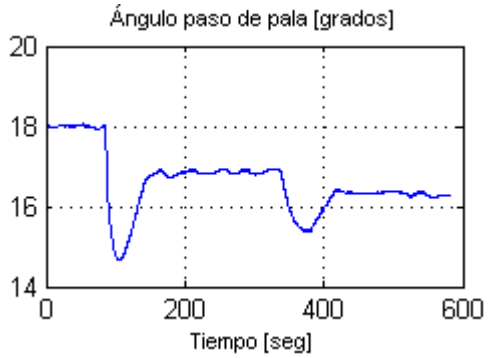
A los 350 segundos de ensayo se modifica la consigna de potencia manteniendo la consigna de velocidad. En este caso se pide al sistema una potencia generada de 4.5 W. El controlador de par actúa aumentando el índice de modulación de carga y por consecuencia disminuyendo la carga aparente acoplada. En este cambio de consigna se aprecia que la perturbación en el lazo de velocidad es mayor que en el caso contrario. El control de pitch tiene que reducir el ángulo de paso de pala para contrarrestar la pérdida de velocidad fruto del desplazamiento de la curva de par del generador.



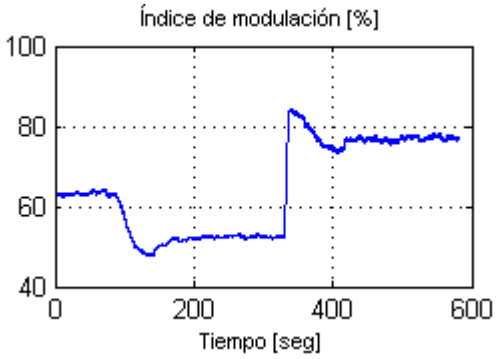
a)



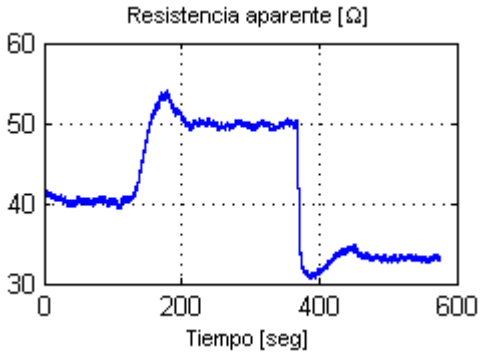
b)



c)



d)



e)

Figura 13: Registro del ensayo de control multivariable descentralizado, a) velocidad rotórica, b) potencia generada, c) ángulo de paso de pala, d) índice de modulación y e) resistencia aparente.

7. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha presentado una primera aproximación de una de las técnicas de control multivariable sobre un sistema de generación de energía eólica. Sobre un equipo experimental de pequeña potencia se ha abordado la problemática de control que presentan estos equipos fundamentalmente en el control de la potencia de generación y se ha demostrado como una estrategia multivariable, en este caso descentralizada, puede estabilizar al sistema en cualquier punto de operación.

En base a un planteamiento teórico previo se ha mostrado el proceso de diseño de la electrónica necesaria para conseguir un control de potencia a través de la modificación de corriente de carga y el desplazamiento de la curva de par generado y de esta forma dotar al sistema experimental de carácter multivariable.

Con los últimos ensayos efectuados ha quedado de manifiesto la interacción que existe entre lazos de control, que en determinadas aplicaciones se podría considerar no deseada. Como trabajo futuro se plantea el estudio de técnicas de minimización de interacción entre variables de control a través de redes de desacoplo [7], [8].

Agradecimientos

Este trabajo ha sido financiado por el proyecto de Excelencia P10-TEP-6056 Junta de Andalucía, España

Referencias

[1] Abdullah, M.A., Yatim, A.H.M., Wei, C. (2011). "*A Study of Maximum Power Point Tracking Algorithms for Wind Energy System*." IEEE First Conference on Clean Energy and Technology CET. Kuala Lumpur (Malasia).

[2] Barakati, S.M., Kazerani, M., Aplevich, J.D. (2008). "*A Mechanical Speed-Sensorless Maximum Power Tracking Control for a Wind Turbine System including a Matrix Converter*". Power and Energy Society General Meeting-Conversion and Delivery of Electrical in the 21st Century. Pittsburgh (PA), EE.UU.

[3] Baroudi, J.A., Dinavahi, V., Knight, M. (2007). "A Review of Power Converter Topologies for Wind Generators". *Renewable Energy,* vol 32, pp 2369-2385.

[4] Bianchi, F., Mantz R.J., et al. (2004). "Power Regulation in Pitch-Controlled Variable Speed WECS Above Rated Wind Speed". *Renewable Energy*, vol 29,pp 1911-1922.

[5] Boukhezzar, B., Lupu L., et al. (2007). "Multivariable Control Strategy for Variable Speed, Variable Pitch Wind Turbine". *Renewable Energy*, vol 32, pp 1273-1287

[6] Fragoso, S., Vázquez, F., Morilla, F. (2013)."*Control de Velocidad mediante de Paso de Pala para un Aerogenerador Experimental de Pequeña Potencia*" presentado para las XXXIV Jornadas de Automática del Comité Español de Automática. Terrassa (España).

[7] Garrido, J., Vázquez, F., Morilla, F. (2011). "An Extended Approach of Inverted Decoupling". *Journal of Process Control*, vol 21, pp 55-68.

[8] González, M., Garrido, J., et al (2011). "Multivariable Centralized Control with Decoupling Feedforward Compensation for Residential Wind Turbine". *18th World*

[9] Haque, Md., Negnevitsky, M., Muttaqui, M. (2010). "A Novel Control Strategy for a Variable-Speed Wind Turbine with a Permanent-Magnet Synchronous Generator". *IEEE Transanctions on Industry Applications*, vol 46, no 1, pp 331-339

[10] Kortabarria, I., Ibarra, E., Martínez de Alegría, I., et al. (2010) "*Power Converters used in Grid Connected Small Wind Turbines: Analisys of Alternatives*". 5th IET International Conference on Power Electronics, Machines and drives. Brighton, UK.

[11] Koutroulis, E., Kalaitzakis, K. (2006). "Design of a Maximum Power Tracking System for Wind-Energy-Conversion Applications". *IEEE Transactions on industrial electronics*, vol 53, no 2, pp 486-494.

[12] Miller, A., Muljadi, E., Zinger, DS. (1997) "A Variable Speed Wind Turbine Power Control". *IEEE Transactions on Energy Conversion*, vol 12, no 2, pp 181-186.

[13] Muljadi, E., Pierce, K., Migliore, P. (2000). "Soft-stall Control for Variable-Speed Stall-Regulated Wind Turbines". *Journal of Wind Engineering and Industrial Aerodynamics* vol 85, pp 277-291.

[14] Vázquez, F., González, M., Garrido, J., Morilla, F. (2012). "Control Multivariable: aplicación al control de un aerogenerador". *X Simposio CEA de Ingeniería de Control.* Barcelona (España).

Modelado y control de un vehículo eléctrico mediante una estrategia de control predictivo basado en modelo

A. Pérez-Castro[a], J. E. Normey-Rico[b], J. L. Guzmán[c], M. Berenguel[c]
[a]Dpto. de Informática y Automática (UNED), [b]Dpto. de Automaçao e Sistemas (UFSC), [c]Dpto. Informática (UAL)
agustinperezcastro@bec.uned.es, julio@das.ufsc.br, joguzman@ual.es, beren@ual.es

Resumen

Este trabajo presenta el diseño de un controlador de la dinámica longitudinal de un vehículo eléctrico, teniendo como objetivo final el optimizar la eficiencia energética del mismo. Debido a las características del sistema a controlar y al objetivo perseguido, se hace uso de un controlador predictivo práctico no lineal basado en modelo (PNMPC). La variación de los parámetros, función de coste y restricciones permite al controlador gestionar el comportamiento del vehículo. Como paso previo al diseño del sistema de control se ha desarrollado un modelo del vehículo que ha sido utilizado tanto en la obtención de predicciones del comportamiento dinámico como en la simulación del sistema real.

Palabras clave: Vehículo Eléctrico, Modelado, Control Predictivo, PNMPC

1. Introducción

El mayor problema al que se enfrentan los vehículos eléctricos es su autonomía. Esto se debe a que, aunque un vehículo eléctrico hace un uso mucho más eficiente de la energía, la cantidad de energía que puede almacenar una batería es muy baja en comparación a la que se puede extraer de los combustibles fósiles [6]. Desde el punto de vista del control se pueden realizar aportaciones encaminadas a que la gestión de energía en el vehículo sea aún más eficiente, evitando frenadas innecesarias o grandes aceleraciones, en las que se malgaste una gran cantidad de energía que luego sea disipada en calor en los frenos. Además, para una mayor aceleración se necesita de una mayor corriente de salida de la batería, lo cual hace que se descargue más rápido. La mejor forma de evitar aceleraciones y frenadas innecesarias, es disponer de una predicción de la velocidad necesaria en cada momento. Por tanto, se requiere de un control que, en primer lugar, pueda anticiparse al comportamiento futuro del vehículo, y segundo, pueda dar soluciones de control que minimicen el consumo. Estas dos características las proporciona el Control Predictivo Basado en Modelo [5], dónde los cálculos de las señales de control actual y futuras se obtienen a través de un proceso de optimización que involucra una función de coste (generalmente cuadrática), un modelo del sistema, las señales de referencia futuras, una estimación de las perturbaciones futuras y las restricciones a las que está sometido el sistema. Se trata por tanto de una técnica ideal para abordar el problema planteado en este trabajo. Existen muchos tipos de técnicas de MPC, en este trabajo, debido a la no linealidad del sistema se hará uso de el algoritmo *Practical Non linear Model Predictive Control* (PNMPC) [9].

2. Modelado de un vehículo eléctrico

Puesto que el objetivo del trabajo es que el vehículo siga unas determinadas consignas (tabuladas) de velocidad, con un comportamiento óptimo en lo que se refiere a la autonomía, el modelo desarrollado contempla únicamente la dinámica longitudinal de un vehículo eléctrico puro. El vehículo se ha modelado de forma simplificada teniendo en consideración los distintos elementos que lo conforman: batería, conversor de continua a continua, motor eléctrico y dinámica lineal. En la Fig. 1 se muestra el esquema que representa a los distintos componentes y la relación entre los mismos a través de sus entradas y salidas.

2.1. Batería

Se ha optado por la realización de un modelo simple de batería basado en un circuito equivalente similar al utilizado en [6], donde se supone que la batería equivale a una fuente de potencial con una resistencia en línea. Dicho circuito está modelado por las ecuaciones (1) y (2). En la ecuación (3) se muestra la diferencia de potencial en los bornes de la batería en función de la corriente de descarga y el estado de carga (SOC), porcentaje de carga que queda en la batería.

$$E_b = nCel \cdot (2.15 - SOD \cdot (2.15 - 2)) \quad (1)$$

$$R_b = nCel \cdot 0.022/C_{10} \quad (2)$$

$$V_b = nCel \cdot (2.15 - 0.15 \cdot SOD - \frac{0.022}{C_{10}} \cdot I_b) \quad (3)$$

donde: E_b: Fuente de potencial del circuito equivalente $[V]$, $nCel$: Número de celdas $[-]$, SOD:

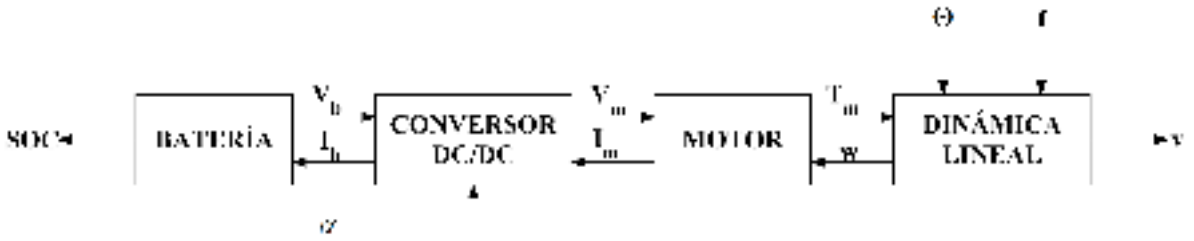


Figura 1: Esquema del modelo completo

Estado de descarga $(1 - SOC)$ $[-]$, R_b: Resistencia interna de la batería $[\Omega]$, C_{10}: Capacidad de la batería a 10 horas de descarga $[A\ h]$, V_b: Voltaje en los bornes de la batería $[V]$, I_b: Corriente de descarga $[A]$.

Además, la batería de un vehículo no se descarga a una tensión constante y la capacidad de las baterías varía en función de la tasa de descarga. Este problema se resuelve, de forma aproximada, utilizando el Coeficiente de Peukert [6]. De esta manera, el SOC de la batería se puede estimar utilizando las ecuaciones (4)-(6).

$$C_{peu} = \left(\frac{C_{10}}{10}\right)^{K_{peu}} \cdot 10 \tag{4}$$

$$\frac{dC_{con}}{dt} = \frac{I_b^{K_{peu}}}{3600} \quad ; \quad \frac{dC_{lib}}{dt} = \frac{I_b}{3600} \tag{5}$$

$$SOD = \frac{C_{con}}{C_{peu}} \quad ; \quad SOC = 1 - SOD \tag{6}$$

donde: C_{peu}: Capacidad de Peukert $[A\ h]$, K_{peu}: Coeficiente de Peukert $[-]$, C_{con}: Carga consumida $[A\ h]$, C_{lib}: Carga liberada $[A\ h]$.

2.2. Conversor de continua a continua

El funcionamiento de un conversor se basa en la conservación de la potencia eléctrica. Además, los valores de salida se ven influenciados por las pérdidas del circuito del conversor. La ecuaciones (7) y (8), similares a las que podemos encontrar en [6], representan el funcionamiento del conversor.

$$V_b \cdot I_b = \eta_c \cdot V_m \cdot I_m \tag{7}$$

$$V_m = V_b \cdot \alpha \quad ; \quad I_b = I_m \cdot \frac{\alpha}{\eta_c} \tag{8}$$

donde: η_c: Eficiencia del conversor $[-]$, V_m: Diferencia de potencial en los bornes del motor $[V]$, I_m: Corriente a través del motor $[A]$, α: Factor de conversión $[-]$.

2.3. Motor eléctrico

Para el modelo del motor se ha elegido un motor de corriente continua con escobillas. La parte mecánica del motor se ha modelado junto con la dinámica lineal del vehículo. Las ecuaciones (9) y (10) muestran la relación entre las entradas y las salidas del motor [11].

$$\frac{dI_m}{dt} = \frac{V_m - K_m \cdot w - R_m \cdot I_m}{L_m} \tag{9}$$

$$T_m = I_m \cdot K_i \tag{10}$$

donde: I_m: Corriente que circula a través del bobinado del motor $[A]$; V_m: Diferencia de potencial en los bornes del motor $[V]$; K_m: Constante de fuerza contraelectromotriz $[V\ rad^{-1}\ s]$; K_i: Constante de par motor $[N\ m\ A^{-1}]$; w: Velocidad angular del motor $[rad\ s^{-1}]$; R_m: Resistencia de la bobina del motor $[\Omega]$; L_m: Inductancia de la bobina del motor $[H]$; T_m: Par motor $[N\ m]$.

2.4. Dinámica lineal

Para poder conocer la velocidad a la que se está desplazando el vehículo en cada instante de tiempo es necesario disponer de un modelo de la dinámica del mismo. Este modelo consiste en la representación de las fuerzas que actúan sobre el vehículo en el eje longitudinal del mismo. En la Fig. 2 se puede observar un esquema, similar al que aparece en [6, 4], de las fuerzas que actúan sobre el vehículo. Dichas fuerzas se describen en las ecuaciones (11)-(19).

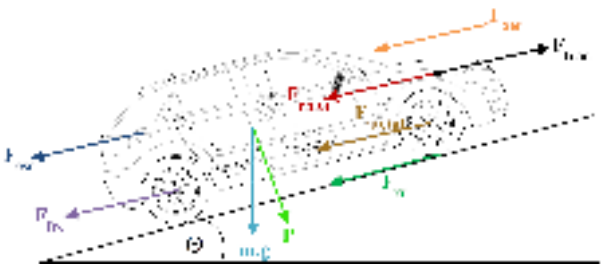


Figura 2: Fuerzas que integran la dinámica lineal

$$F_{trac} = T_m \cdot \frac{G}{r} \quad (11)$$

$$F_{esc} = m \cdot g \cdot \sin(\theta) \quad (12)$$

$$F_{rr} = m \cdot g \cdot \eta_r \cdot \cos(\theta) \quad (13)$$

$$F_{aer} = \frac{1}{2} \cdot \rho \cdot A \cdot C_d \cdot v^2 \quad (14)$$
$$= \frac{1}{2} \cdot \rho \cdot A \cdot C_d \cdot w^2 \cdot (\frac{r}{G})^2$$

$$F_{al} = m \cdot a = m \cdot \frac{dv}{dt} = \quad (15)$$
$$= m \cdot \frac{dw}{dt} \cdot \frac{r}{G}$$

$$F_{awm} = J_m \cdot \frac{dw}{dt} \cdot \frac{G}{r} \quad (16)$$

$$F_{awr} = J_r \cdot \frac{dw}{dt} \cdot \frac{G}{r} \quad (17)$$

$$F_{rmot} = B \cdot \frac{G}{r} \cdot w \quad (18)$$

$$F_{fre} = F_{fmax} \cdot f \quad (19)$$

donde: F_{trac}: Fuerza de tracción $[N]$, G: Relación de transferencia entre el motor y el eje de las ruedas $[-]$, r: Radio de la rueda $[m]$, F_{esc}: Fuerza de escalada $[N]$, m: Masa del vehículo $[kg]$, g: Aceleración de la gravedad $[m\ s^{-2}]$, θ: Ángulo de escalada $[rad]$, F_{rr}: Fuerza de resistencia al rodamiento $[N]$, η_r: Coeficiente de rozamiento $[-]$, F_{aer}: Fuerza aerodinámica $[N]$, ρ: Densidad del aire $[kg\ m^{-3}]$, A: Área frontal del vehículo $[m^2]$, C_d: Coeficiente aerodinámico $[-]$, v: Velocidad lineal $[m\ s^{-1}]$, w: Velocidad angular $[rad\ s^{-1}]$, F_{al}: Fuerza de aceleración lineal $[N]$, F_{awm}: Fuerza de aceleración angular del motor $[N]$, J_m: Inercia del motor $[kg\ m^2]$, F_{awr}: Fuerza de aceleración angular de las ruedas $[N]$, J_r: Inercia de las ruedas y los ejes $[kg\ m^2]$, F_{rmot}: Fuerza de rozamiento en el motor $[N]$, B: Coeficiente viscosidad del motor $[N\ m\ rad^{-1}\ s]$, F_{fre}: Fuerza de frenado $[N]$, F_{fmax}: Fuerza máxima de frenado $[N]$, f: Coeficiente de frenado $[-]$.

Las fuerzas descritas por las ecuaciones (16) y (17) pueden ser consideradas únicamente como un aumento en la masa de la ecuación (15). Esto puede ser útil si no se conocen las inercias correspondientes. La sumatoria de fuerzas resultante da lugar a la ecuación (20)

$$F_{trac} = F_{esc} + F_{aer} + F_{roz} + \quad (20)$$
$$+F_{al} + F_{awm} + F_{awr} + F_{mot} + F_{fre}$$

considerando $F_{atotal} = F_{al} + F_{awm} + F_{awr}$, se tiene que:

$$F_{atotal} = F_{trac} - F_{esc} - F_{aer} - F_{roz} - \quad (21)$$
$$-F_{mot} - F_{fre}$$

Operando en la ecuación (21) se obtiene la relación entre las distintas fuerzas que actúan sobre el vehículo y la velocidad del mismo.

$$\frac{dw}{dt} = \frac{F_{atotal}}{m \cdot \frac{r}{G} + (J_m + J_r) \cdot \frac{G}{r}} \quad (22)$$

Además, para obtener la velocidad lineal del vehículo se hace uso de la ecuación (23).

$$v = w \cdot \frac{r}{G} \quad (23)$$

2.5. Modelo completo

Se considerará que el modelo tiene una entrada y una salida a controlar. La entrada, α, se corresponde con el factor de conversión del convertidor de corriente continua que alimenta al motor, la cual podría ser considerada directamente proporcional a la posición del pedal del acelerador, y la salida a controlar es la velocidad lineal del vehículo. Además, el sistema dispone de otras salidas auxiliares como, por ejemplo, el estado de carga (SOC), el voltaje de la batería y la velocidad angular del motor.

Con el modelo anteriormente expuesto se ha realizado una simulación, donde el vector con los valores de la entrada α se ha generado de forma que el comportamiento en velocidad del vehículo sea similar al del ciclo de conducción FUDS [2], ver sección 4.1. La secuencia de los valores de α utilizada en la simulación se representa en la Fig. 3. En la Fig. 4 se puede observar la evolución de la velocidad lineal del vehículo, y la Fig. 5 muestra la evolución en el SOC de la batería. Respecto a los parámetros utilizados para la simulación del modelo se han utilizado los que se muestran en la Tabla 1.

3. Estrategia de control

Como ya se mencionó anteriormente, el principal objetivo de este trabajo es el desarrollo de un sistema de control que permita mejorar la eficiencia energética de un vehículo eléctrico, lo cual se traduce en disminuir el consumo de batería, permitiendo así una mayor autonomía. Tras un estudio de las distintas alternativas que se podrían utilizar para el desarrollo de dicho sistema de control, se ha decidido usar un sistema de control predictivo basado en modelo (MPC). Más concretamente, debido a las características del sistema a controlar, un sistema no lineal, se ha optado por usar un controlador PNMPC. En la Fig. 6 se muestra el esquema de control propuesto.

Tabla 1: Valores utilizados para la simulación.

Componente	Parámetro	Valor
Batería	$nCel$	24
	C_{10}	210 Ah
	$SOC_{inicial}$	1
	K_{peu}	1.2
Conversor	η_c	0.95
Motor	K_m	1 $V\ rad^{-1}\ s$
	K_i	0.5 $N\ m\ A^{-1}$
	R_m	0.139 Ω
	L_m	0.02 H
	w_{ini}	0 $rad\ s^{-1}$
	I_{ini}	0 A
Dinámica	G	1.1
	r	0.3 m
	m	1200 kg
	g	9.8 $m\ s^{-2}$
	ρ	1.25 $kg\ m^{-3}$
	A	1.8 m^2
	C_d	0.3
	η_r	0.0048
	J_m	3.5 $kg\ m^2$
	J_r	0 $kg\ m^2$
	F_{fmax}	200 N
	B	0.6$N\ m\ rad^{-1}\ s$
Ciclo	v_{ref}	FUDS
Simulación	Ts	0.1 s

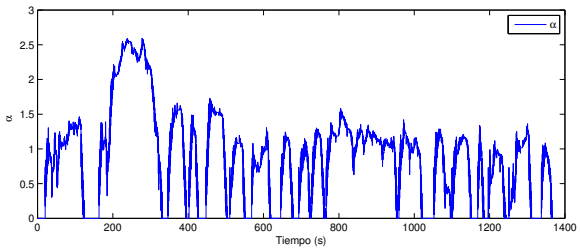


Figura 3: Secuencia de valores de α

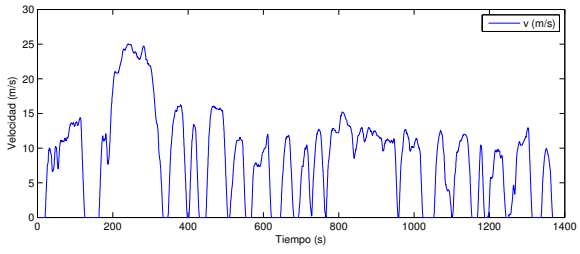


Figura 4: Velocidad del vehículo durante un ciclo

3.1. Control predictivo basado en modelo (MPC)

La posibilidad de calcular consignas que den como resultado un uso más eficiente de los recursos y la facilidad del uso de restricciones son dos de los factores que hacen que este tipo de control sea

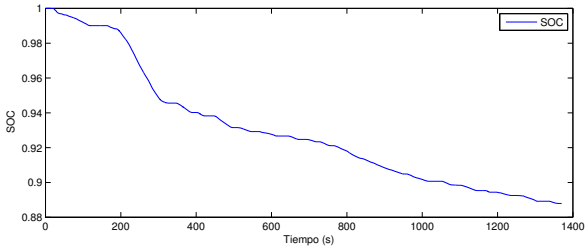


Figura 5: SOC durante un ciclo

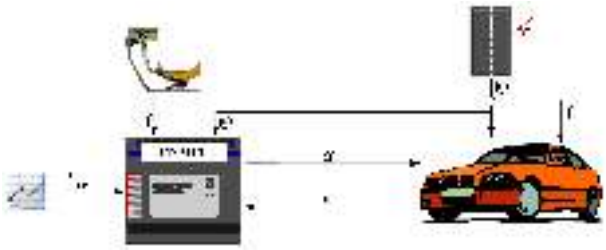


Figura 6: Esquema del control diseñado

ideal para el control de procesos industriales, incluido en el área del control de vehículos eléctricos [3]. En [5] se explican las características, funcionamiento y técnicas más representativas del control MPC.

3.2. Control predictivo práctico basado en modelo no lineal (PNMPC)

El algoritmo de control PNMPC [9] se caracteriza por linealizar el modelo de predicciones en cada uno de los periodos de muestreo. Esto permite una gran simplicidad y un buen ajuste de la linealización, independientemente del punto de trabajo en el que se encuentre el sistema. Por lo tanto, en este caso las predicciones vienen dadas por la ecuación (24). Se puede encontrar un ejemplo del uso de este algoritmo en [8].

$$Y_p = F_{lib} + G_{PNMPC} \cdot \Delta u \tag{24}$$

donde:

- Y_p: Salidas predichas en el horizonte de predicción considerado.
- F_{lib}: Respuesta libre, es decir, las salidas futuras del sistema cuando los incrementos de la señal de control actual y futuros (Δu) son iguales a 0.
- $G_{PNMPC} = \frac{\partial Y_p}{\partial \Delta u}$: Jacobiano de Y_p.

Tanto F_{lib} como G_{PNMPC} se han de calcular numéricamente en cada periodo de muestreo siguiendo el algoritmo propuesto en [9].

3.3. Tratamiento del error de predicción, ruido y perturbaciones

Para tratar con el error de predicción, el ruido y las perturbaciones no medibles, la técnica PNMPC implementa un factor de corrección F_c que se ha de añadir a cada una de las predicciones. Este factor de corrección se calcula en cada instante de muestreo a partir de la integral del error de predicción filtrado, viniendo descrito por un parámetro de diseño a_f:

$$F_c(z^{-1}) = \frac{1}{1-f_d} \cdot \frac{k_i}{1-z^{-1}} \cdot e(z^{-1}) \quad (25)$$

$$f_d = a_f^2; \;\; k_i = 1 + a_f^2 - 2 \cdot a_f \quad (26)$$

3.4. Función de coste

En este trabajo se han realizado simulaciones con dos funciones de coste diferentes. La primera función, J_1, es la función de coste típica de un controlador MPC y vendría representada por los dos primeros sumatorios de la función J_2 que se expone a continuación. La segunda función, representada por la ecuación (27), añade además otra componente para poder minimizar el valor absoluto de la señal de control en cada instante.

$$J_2 = \sum_{j=1}^{N} \|Y_p(k+j|k) - w_{ref}(k+j|k)\|^2_{P_R} + \quad (27)$$

$$+ \sum_{j=1}^{N_u} \|\Delta u(k+j-1)\|^2_{P_Q} + \sum_{j=1}^{N_u} \|u(k+j-1)\|^2_{P_C}$$

donde: $Y_p(k+j|k)$: Salida predicha para el instante $k+j$ calculado en el instante k, w_{ref}: Referencia futura, u: Señal de control, P_R: Ponderación de la referencia, P_Q: Ponderación de la variación del control, P_C: Ponderación de la señal de control, N: Horizonte de predicción, N_u: Horizonte de control.

3.5. Restricciones

En el controlador diseñado se han considerado siete restricciones. Las tres primeras son las restricciones más comúnmente utilizadas en cualquier MPC, estas son restricciones sobre: la variación del control; la señal de control; y la salida. Las otras cuatro, se implementan como mecanismos de protección de los subsistemas del vehículo, y son restricciones sobre: el voltaje de entrada al motor; la corriente en el motor; la corriente en la batería; y la velocidad angular del motor.

$$\Delta u_{min} \leq \Delta u(k+j) \leq \Delta u_{max} \;\; \forall j = 0, \dots N_u - 1 \quad (28)$$

$$u_{min} \leq u(k+j) \leq u_{max} \;\; \forall j = 0, \dots N_u - 1 \quad (29)$$

$$Y_{min} \leq Y_p(k+j|k) \leq Y_{max} \;\; \forall j = 1, \dots N \quad (30)$$

$$V_{m_{min}} \leq V_m(k+j) \leq V_{m_{max}} \;\; \forall j = 0, \dots N_u - 1 \quad (31)$$

$$I_{m_{min}} \leq I_m(k+j) \leq I_{m_{max}} \;\; \forall j = 0, \dots N_u - 1 \quad (32)$$

$$I_{b_{min}} \leq I_b(k+j) \leq I_{b_{max}} \;\; \forall j = 0, \dots N_u - 1 \quad (33)$$

$$w_{min} \leq w(k+j|k) \leq w_{max} \;\; \forall j = 1, \dots N \quad (34)$$

Todas estas restricciones se pueden expresar en función del vector de incrementos actual y futuros de la señal de control Δu [10].

3.6. Mejoras en el algoritmo PNMPC

En este trabajo se han implementado una serie de mejoras sobre la implementación del algoritmo de control PNMPC que mejoran su robustez y tiempo de cómputo [10]. Se ha implementado una variante del algoritmo que permite variar la amplitud del impulso ϵ utilizado para el cálculo de los coeficientes de respuesta a escalón de la matriz G_{PNMPC}. En el caso de los tiempo de cómputo se proponen dos mejoras. La primera solución es modificar el cálculo de GPNMPC propuesto en [9] según el algoritmo siguiente, a través del cual se obtendrá la misma solución pero ejecutando menos veces el modelo de predicciones, reduciendo el tiempo de cómputo. La segunda se basa en la suposición de que en puntos de trabajo cercanos se puede simplificar considerando la misma linealización del sistema y por tanto no siendo necesario recalcularla.

Algoritmo 1 Cálculo mejorado de G_{PNMPC}

1: Se obtiene el vector Y_p^0, de longitud N, ejecutando el modelo de predicciones con las entradas y salidas pasadas y con $\Delta u = [0, 0, \dots, 0]^T$. $F_{lib} = Y_p^0$
2: Se calcula la primera columna de la matriz G_{PNMPC}. Para esto se calcula el vector Y_p^1, de longitud N, ejecutando el modelo de predicciones con las entradas y salidas pasadas y con $\Delta u = [\epsilon, 0, \dots, 0]^T$, donde ϵ será un valor muy pequeño, por ejemplo $\frac{u_{k-1}}{1000}$. $G_{PNMPC_1} = \frac{Y_p^1 - Y_p^0}{\epsilon}$.
3: Se calculan el resto de columnas de la matriz G_{PNMPC}. Para esto se inicializan los k primeros valores con los valores del vector Y_p^k con los k primeros valores de la respuesta libre F_{lib}. El resto de valores de Y_p^k se calculan ejecutando el modelo de predicciones a partir del instante $k+1$. $G_{PNMPC_k} = \frac{Y_p^k - Y_p^0}{\epsilon}$

El paso 3 del algoritmo sustituye dos pasos del algoritmo original [9] (3. Se calcula la segunda columna de la matriz G_{PNMPC}. Para esto se calcula el vector Y_p^2, de longitud N, ejecutando el modelo de predicciones con las entradas y salidas pasadas y con $\Delta u = [0, \epsilon, 0, \dots, 0]^T$. $G_{PNMPC} = \frac{Y_p^2 - Y_p^0}{\epsilon}$. 4. Se calculan el resto de las columnas de G_{PNMPC}).

4. Resultados

4.1. Rutas prefijadas

Existen tablas de velocidades que sirven para probar el comportamiento de un diseño de vehículo dado [6] y comparar distintos vehículos ante un mismo ciclo. Algunos de los ciclos de velocidad más utilizados son: FUDS [2] y SFUDS, útiles para simular un comportamiento en ciudad; FHDS y US06, utilizados para simular el comportamiento en carretera; ECE-15; NEDC, cuyo ejemplo de uso podemos ver en [7]; y otros como ECE-47 o SAE J227a. Para obtener algunos de los ciclos anteriormente mencionados se puede recurrir a [1].

4.2. Resultados de control

Con el fin de comprobar que el controlador funciona en distintos escenarios, se han realizado simulaciones haciendo uso de varios ciclos de conducción. Por motivos de limitación de espacio no se muestran los resultados obtenidos en todos ellos, pero se pueden consultar en [10]. Los parámetros utilizados para realizar dichas simulaciones son los que se muestran en la Tabla 2. Tal y como se puede observar en la Fig. 7 el controlador diseñado proporciona buenos resultados con la modificación propuesta al algoritmo PNMPC.

Tabla 2: Parámetros del controlador PNMPC

Parámetro	Valor	Parámetro	Valor
N	5	N_u	5
J	J_2	P_r	3
P_q	3	P_c	2
$\epsilon_{inicial}$	0.1	a_f	0.3
T_s	0.1 s		

4.3. Cambio en las funciones de coste y los factores de ponderación

En la Tabla 3 se muestran datos de simulación sobre distintos ciclos. Los parámetros comunes son: $N = 5$, $N_u = 5$ y $T_s = 0.1s$. Como se puede observar en dicha tabla, el hacer uso de la función de coste J_2 tiene el efecto de reducir el consumo de batería, aunque como contrapartida el ajuste a la referencia empeora. Respecto a la variación de los factores de ponderación, podemos observar cómo, usando la función J_1, si el peso del seguimiento de referencia es mayor, hay menor error, como era de esperar, y como consecuencia provoca un mayor consumo. Además, de los valores obtenidos en la simulación del ciclo US06, se puede deducir que no basta con actuar sobre los pesos P_R y P_C para mejorar la autonomía, sino que también hay que actuar sobre el parámetro P_Q, el cual pondera el esfuerzo de control. Para el cálculo del error en esta tabla y sucesivas se hace uso del *Root Mean Square Error* (RMSE).

Tabla 3: Efectos del cambio de la función de coste y de los pesos

Ciclo		P_R	P_Q	P_C	SOC	$RMSE$
FUDS						
	J_1	4	1		0.880	0.219
	J_1	2	1		0.882	0.222
	J_2	2	1	1	0.884	0.229
US06						
	J_1	50	1		0.706	0.192
	J_1	2	1		0.706	0.211
	J_2	2	1	1	0.713	0.307
	J_2	2	1	100	0.668	6.579
	J_2	2	50	100	0.898	12.710
NEDC						
	J_1	2	1		0.864	0.117
	J_2	2	1	1	0.866	0.142

4.4. Ruido, error de modelado y perturbaciones

Para simular el error de modelado y el ruido se han cambiado los parámetros del modelo de predicciones para que sean distintos de los del sistema (Tabla 4). Además, se ha añadido un ruido blanco a la salida del sistema. En la Fig. 8 se puede observar cómo actúa el controlador ante un sistema con error de modelado y ruido cuando no se aplica el factor de corrección. Como se puede ver, el controlador ajusta a la referencia bastante bien aunque debido a los errores de modelado incluidos, se observa un error de *offset* permanente. Sin embargo, tal y como podemos ver en la Fig. 9, si se aplica el factor de corrección con parámetro $a_f = 0.3$, el sistema pasa a seguir la velocidad de referencia a pesar de errores de modelado incluidos y del ruido añadido a la simulación.

4.5. Modificación del cálculo de G_{PNMPC}

Respecto a los efectos de implementar una de las mejoras de la eficiencia propuestas, en concreto la que evita calcular G_{PNMPC} en cada periodo de muestreo, la comparativa se puede observar en la Tabla 5. Para las simulaciones de dicha tabla se ha usado el ciclo NEDC, y se han simulado 20 minutos. De los datos de la tabla se puede deducir que la mejora hace que el tiempo de simulación descienda por debajo de la mitad sin hacer que esto afecte a la calidad del control. Además, como se puede observar, el usar porcentajes por encima del 5 % no supondría grandes mejoras en cuanto a lo que tiempo de simulación se refiere.

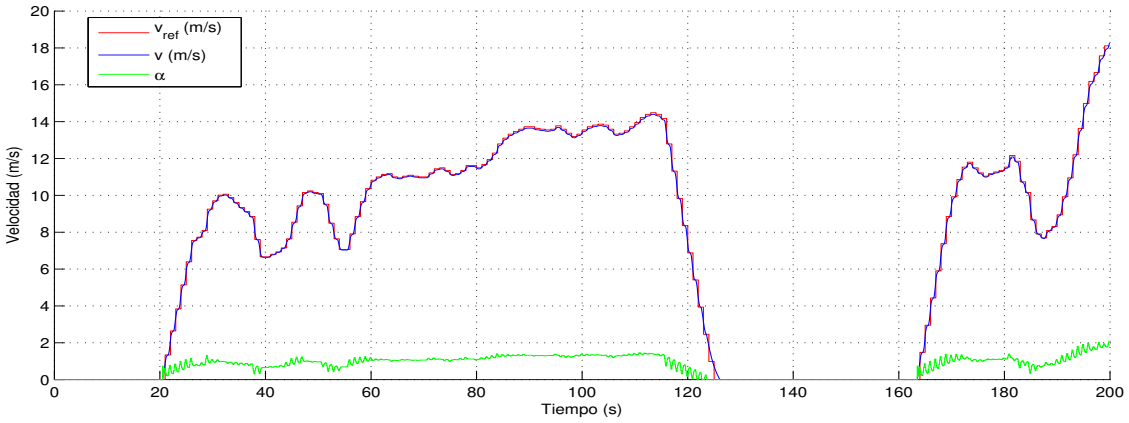


Figura 7: Ciclo de conducción FUDS con PNMPC

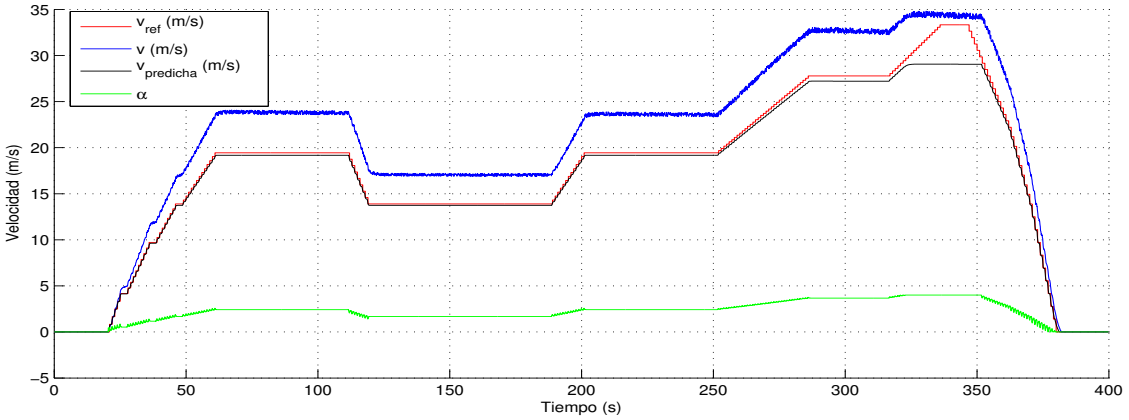


Figura 8: Simulación con error de modelado y ruido, sin factor de corrección

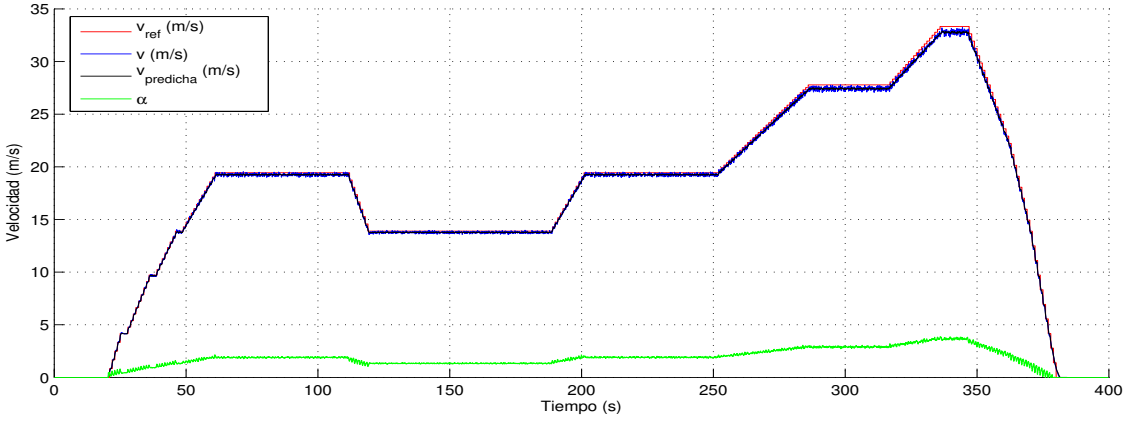


Figura 9: Simulación con error de modelado y ruido, con factor de corrección

Tabla 4: Valores utilizados para simular el error de modelado.

Componente	Parámetro	Valor
Batería	$nCell$	23
	C_{10}	210 Ah
	$SOC_{inicial}$	1
	K_{peu}	1.2
Conversor	η_c	0.95
Motor	K_m	1.2 $V\ rad^{-1}\ s$
	K_i	0.9 $N\ m\ A^{-1}$
	R_m	0.3 Ω
	L_m	0.02 H
	w_{ini}	0 $rad\ s^{-1}$
	I_{ini}	0 A
Dinámica	G	1.1
	r	0.3 m
	m	900 kg
	g	9.8 $m\ s^{-2}$
	ρ	1.25 $kg\ m^{-3}$
	A	2.5 m^2
	C_d	0.3
	η_r	0.0048
	J_m	4 $kg\ m^2$
	J_r	0 $kg\ m^2$
	F_{fmax}	200 N
	B	0.6$N\ m\ rad^{-1}\ s$
Simulación	Ts	0.1 s

Tabla 5: Resultados de la optimización del tiempo de cómputo.

Opt	Error modelado	%	Tiempo (s)	$RMSE$
No	No		203.1	0.50445
No	Si		200.5	0.50207
Si	No	5	87.2	0.50455
Si	Si	5	85.5	0.50200
Si	No	10	81.8	0.50466
Si	No	50	72.8	0.50322

5. Conclusiones, y posibles mejoras

En el presente trabajo se ha mostrado que, el uso de una estrategia de control predictivo basado en modelo no lineal es adecuado para el control de la velocidad de un vehículo eléctrico puro teniendo en cuenta la dinámica longitudinal del mismo. Además, se ha comprobado que el mismo es válido ante un buen rango de entradas, esto es, un conjunto variado de ciclos de velocidad. También se puede concluir que el uso de control predictivo, además de conseguir un buen ajuste a la consigna deseada, permite, mediante las restricciones y la función de coste, ajustar otra serie de parámetros del sistema para que cumplan una restricciones y para que minimice los costes sobre los que se decida actuar. En cuanto a las mejoras o ampliaciones que se pueden realizar sobre el presente trabajo, quizás las dos más importantes sean, la consideración de la frenada regenerativa y el el uso de un esquema de control MIMO que considere la frenada como entrada en lugar de como perturbación. También será importante realizar el modelado de la dinámica lateral del vehículo.

Agradecimientos

Este trabajo se ha desarrollado en el ámbito de los proyectos: DPI2010-21589-C05-04, del Ministerio de Ciencia y Innovación; PHB2009-0008, del Ministerio de Educación y CAPES DGU 220-2010. Los autores también agradecen el apoyo financiero del Ministerio de Economía y Competitividad en virtud de la beca FPI enmarcada en el proyecto DPI2011-27818-C02-02.

Referencias

[1] http://www.epa.gov/nvfel/testing/ dynamometer.htm.

[2] http://www.metricmind.com/data/cycles.pdf.

[3] A. Arce, A. J. del Real, and C. Bordóns. Predictive control for battery performance improvement in hybrid PEM fuel cell vehicles. *17th IEEE International Conference on Control Applications*, 2008.

[4] A. Arce, A. J. del Real, and C. Bordóns. MPC for battery/fuel cell hybrid vehicles including fuel cell dynamics and battery performance improvement. *Journal of Process Control, Volume 19, Issue 8*, 2009.

[5] E. F. Camacho and Carlos Bordóns. *Model predictive control*. Springer, 2004.

[6] J. Larminie and J. Lowry. *Electric vehicle technology explained*. Wiley, 2003.

[7] D. Marcos, C. Bordóns, and M. A. Ridao. Estudio comparativo de sistemas de almacenamiento de energía en un vehículo híbrido con pila de combustible. *XXXI Jornadas de Automática - Jaén*, 2010.

[8] J. E. Normey-Rico, P. R.da Costa Mendes, and R. Le ao Carvalho. *Relatório Controle Separador Trifásico*. Universidade Federal de Santa Catarina, 2011.

[9] A. Plucenio. *Desenvolvimento de Técnicas de Controle Não Linear para Elevaçao de Fluidos Multifásicos*. PhD thesis, Universidade Federal de Santa Catarina, 2010.

[10] A. Pérez-Castro. Modelado y control de un vehículo eléctrico mediante una estrategia de control predictivo basado en modelo. Master's thesis, Universidad de Almería, 2011.

[11] P. Wolm, X.Q. Chen, J.G. Chase, W. Pettigrew, and C.E. Hann. Analysis of a PM DC motor model for application in feedback design for electric powered mobility vehicles. In *Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008. 15th International Conference*, pages 640–645, dec. 2008.

Estrategia para la corrección de modelos no lineales variantes en el tiempo mediante estimación de parámetros: Aplicación al control predictivo y a la diagnosis de sistemas

A. Pérez-Castro[a], J. Sánchez[a], J.L. Guzmán[b]
[a]Dpto. de Informática y Automática (UNED), [b]Dpto. de Informática (UAL)
agustinperezcastro@bec.uned.es, jsanchez@dia.uned.es, joguzman@ual.es

Resumen

En el presente trabajo se presenta una estrategia para la corrección de modelos de predicción no lineales variantes en el tiempo. Dicha corrección se lleva a cabo mediante el recálculo de los parámetros internos que modifican la dinámica del sistema al que representan, así como con la actualización de sus estados. Esta corrección de modelos tiene una doble funcionalidad; por un lado, permite un mejor desempeño en controladores predictivos no lineales que actúen sobre sistemas variantes en el tiempo; por otro lado, sirve como mecanismo de diagnosis de sistemas, ya que proporciona información relevante sobre el estado en el que se encuentran los mismos. Para el desarrollo de esta estrategia se usa un controlador predictivo no lineal, con el objetivo de que la técnica sea aplicable a sistemas no lineales variantes en el tiempo.

Palabras clave: Modelado, Control Predictivo, PNMPC, Diagnosis de sistemas, Modelos no lineales variantes en el tiempo

1. Introducción

Uno de los problemas que se pueden encontrar en los sistemas de control es el de la variación en el comportamiento de los sistemas, ya sea porque el sistema tiene, de forma natural, dinámicas variantes o bien por el deterioro o fallo de alguno de sus componentes. Este cambio en la dinámica o deterioro puede suponer desde pérdidas económicas, por un mal ajuste o por averías, hasta peligro para los operadores, pasando por un mal funcionamiento del sistema. Para solucionar este problema se hace uso del control tolerante a fallos, entendiendo como fallo: todo cambio en el comportamiento de alguno de los componentes del sistema (desviación no permitida de alguna de sus propiedades o parámetros característicos) de manera que éste ya no puede satisfacer la función para la cual ha sido diseñado [2, 8].

Según [8], la metodología del control tolerante a fallos debe seguir cinco etapas: Análisis del sistema, diagnosis, tolerancia a fallos, supervisión y aplicación. Además, divide la diagnosis de fallos en tres etapas: Detección, aislamiento y estimación. Por tanto, siguiendo las definiciones propuestas por [8] la estrategia que en este artículo se presenta se podría utilizar en la etapa de detección, más concretamente como un mecanismo de detección de fallos basada en modelos cuantitativos. Dejando el resto de etapas de diagnosis y del control tolerante a fallos para futuros trabajos.

En [3], tal y como recoge [1], el procedimiento para el diagnóstico y detección de fallos mediante la estimación de parámetros consta de los siguientes pasos:

1. Establecer el modelo del proceso. En el caso de este trabajo se hará uso de un modelo de parámetros como el que se presenta en la Sección 3.2.

2. Determinar las relaciones entre los coeficientes del modelo y los parámetros físicos, que en el caso del modelo utilizado, Sección 4.1, dicha relación es directa.

3. Estimar los coeficientes del modelo a partir de las entradas y salidas del proceso mediante algún método de identificación, para lo que se hace uso del corrector explicado en la Sección 3.1.

4. Calcular los parámetros físicos del modelo nominal. Esto no es necesario ya que la relación es directa entre los parámetros del modelo y los parámetros físicos.

5. Determinar los cambios de los parámetros que ocurren por los diferentes casos de fallo. Esta etapa no se trata ya que no es el objetivo del trabajo, y además tal y como propone [8] sería necesaria una etapa previa de análisis, pero bastaría con establecer unas franjas de funcionamiento en cada uno de los parámetros físicos.

En la Sección 4.5, se combinará la estrategia de corrección de modelos con un controlador predictivo no lineal, obteniendo de esta forma un control

adaptativo que variará los parámetros. Dicha variación, además de proporcionar un mecanismo de detección de fallos, como se ha comentado anteriormente, permitirá una mejora en el desempeño del controlador. Esta mejora tendrá ventajas sobre aquellas estrategias que proponen únicamente una corrección del error, como puede ser el factor de corrección utilizado en la técnica PNMPC [7]; ya que las estrategias basadas en la corrección del error o algunas técnicas robustas, aunque permiten al controlador seguir de forma adecuada las referencias, pueden dar como resultado soluciones sub-óptimas. Sin embargo, cuando lo que se corrige son directamente los parámetros del modelo, se obtendrán resultados más cercanos al óptimo; ya que, aunque el seguimiento de la referencia puede ser igual de bueno, la optimización de otros parámetros internos que modifiquen, por ejemplo, el coste de operación, solo podrá ser óptima si el modelo de predicciones se ajusta en su comportamiento interno al sistema. Además, con la técnica propuesta se evita que el propio lazo de control camufle los fallos, errores de modelado o variaciones en la dinámica del sistema.

2. Control predictivo práctico basado en modelo no lineal (PNMPC)

El algoritmo de control PNMPC [7] es la base del corrector propuesto en este trabajo y se utilizará, además, para obtener una estrategia de control adaptativo.

Esta técnica se caracteriza por linealizar el modelo de predicciones en cada uno de los periodos de muestreo. Esto permite una gran simplicidad y un buen ajuste de la linealización, independientemente del punto de trabajo en el que se encuentre el sistema. Por lo tanto, las predicciones vienen dadas por la siguiente expresión:

$$\vec{y_p} = F_{lib} + G_{PNMPC} \cdot \vec{\Delta u} \tag{1}$$

donde $\vec{y_p}$ son las salidas predichas en el horizonte de predicción considerado, F_{lib} es la espuesta libre, es decir, las salidas futuras del sistema cuando los incrementos de la señal de control actual y futuros ($\vec{\Delta u}$) son iguales a 0 y $G_{PNMPC} = \frac{\partial \vec{Y_p}}{\partial \vec{\Delta u}}$ es el Jacobiano de $\vec{y_p}$. Se pueden encontrar ejemplos del uso de este algoritmo en [5, 9, 10].

Tanto F_{lib} como G_{PNMPC} se han de calcular numéricamente en cada periodo de muestreo siguiendo el algoritmo propuesto en [7] y las modificaciones sobre el mismo propuestas en [6, 9], las cuales mejoran la robustez y la velocidad del algoritmo.

En lo que se refiere al tratamiento del error de predicción, ruido y perturbaciones no medibles, la técnica PNMPC implementa un factor de corrección F_c que se ha de añadir a cada una de las predicciones. Éste se calcula de acuerdo a la ecuaciones (2) y (3), obteniendo, por tanto, la integral del error filtrado en función al parámetro de sintonización a_f.

$$F_c(z^{-1}) = \frac{1}{1-f_d} \cdot \frac{k_i}{1-z^{-1}} \cdot e(z^{-1}) \tag{2}$$

$$f_d = a_f^2; \;\; k_i = 1 + a_f^2 - 2 \cdot a_f \tag{3}$$

En lo que se refiere a la función de coste, la típica y la que se utilizará en la parte de control del presente artículo es la que se muestra en la ecuación (4). Además, dicha función de optimización tendrá restricciones sobre la variación en la señal de control, el valor de la señal de control y el valor de la salida, descritas por las ecuaciones (5), (6) y (7), respectivamente.

$$J = \sum_{j=1}^{N} \|\vec{y_p}(k+j|k) - \vec{w_{ref}}(k+j|k)\|^2_{P_R} + \sum_{j=1}^{N_u} \|\vec{\Delta u}(k+j-1)\|^2_{P_Q} \tag{4}$$

donde $\vec{y_p}(k+j|k)$ son las salidas predichas para el instante $k+j$ calculadas en el instante k, $\vec{w_{ref}}$ son las referencias futuras, $\vec{u}$ son las señales de control, P_R es la ponderación de la referencia, P_Q es la ponderación de la variación del control, N es el horizonte de predicción y N_u es el horizonte de control.

$$\vec{\Delta u_{min}} \leq \Delta \vec{u}(k+j) \leq \vec{\Delta u_{max}} \;\; \forall j = 0, \dots N_u - 1 \tag{5}$$

$$\vec{u_{min}} \leq \vec{u}(k+j) \leq \vec{u_{max}} \;\; \forall j = 0, \dots N_u - 1 \tag{6}$$

$$\vec{y_{min}} \leq \vec{y_p}(k+j|k) \leq \vec{y_{max}} \;\; \forall j = 1, \dots N \tag{7}$$

3. Corrector de modelos

Tal y como se muestra en la Figura 1, el corrector de modelos consta de tres componentes fundamentales: un corrector PNMPC, un modelo de parámetros y un disparador.

3.1. Corrector PNMPC

Mientras que el controlador PNMPC, al igual que otros controladores predictivos [4], trabaja con predicciones del futuro, en el caso del corrector PNMPC se hará uso del pasado. De esta forma se sustituyen los horizontes de predicción N y de control N_u, por los horizontes de corrección N_{cor} y de variación de parámetros N_{par}; siendo N_{cor} el número de periodos de muestreo que se consideran

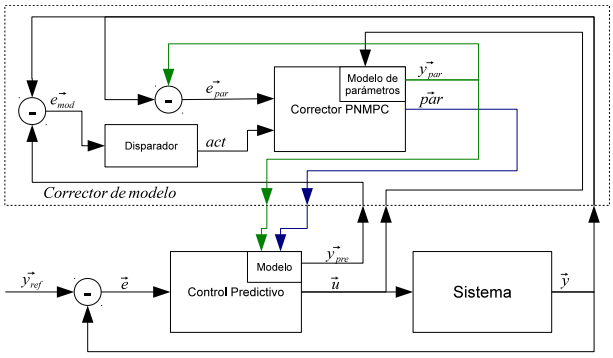


Figura 1: Esquema de control con corrector de modelo

para la corrección de la salida del modelo respecto a la del sistema, y N_{par} el número de periodos de muestreo en los que se permite al controlador realizar variaciones en los parámetros del modelo. La función de coste a minimizar por el corrector PNMPC será la que se muestra en la ecuación (8).

$$J = \sum_{j=N_{cor}-1}^{0} \|y_{\vec{par}}(k-j) - \vec{y}(k-j)\|^2_{P_{cor}} + \sum_{j=N_{par}}^{1} \|\Delta\vec{par}(k-j)\|^2_{P_{par}} \quad (8)$$

donde k es el instante actual de muestreo, $y_{\vec{par}}(k-j)$ son las salidas calculadas por el modelo de parámetros en el instante $k-j$, $\vec{y}(k-j)$ son las salidas reales del sistema en el instante $k-j$, $\Delta\vec{par}(k-j)$ son las variaciones de los parámetros del modelo en el instante $k-j$, P_{cor} son los pesos de corrección (ponderan el orden de magnitud de las distintas salidas y la prioridad en el ajuste entre salidas) y P_{par} son los pesos de variación de parámetros (ponderan el orden de magnitud entre los distintos parámetros y la preferencia en la modificación de los mismos).

La función de coste descrita por la ecuación (8), al igual que ocurre con el control PNMPC, tendrá una serie de restricciones sobre las salidas, los parámetros y las variaciones en los parámetros. Dichas restricciones se describen en las ecuaciones (9), (10) y (11).

$$\vec{y_{min}} \leq y_{\vec{par}}(k-j) \leq \vec{y_{max}} \quad \forall j = N_{cor}-1, \dots 0 \quad (9)$$

$$\vec{par_{min}} \leq \vec{par}(k-j) \leq \vec{par_{max}} \quad \forall j = N_{par}, \dots 1 \quad (10)$$

$$\Delta\vec{par_{min}} \leq \Delta\vec{par}(k-j) \leq \Delta\vec{par_{max}} \quad \forall j = N_{par}, \dots 1 \quad (11)$$

Hay que destacar que la calibración de un modelo respecto a la salida de un sistema no tiene porqué ser única; es decir, para un mismo conjunto de salidas, en un punto de operación concreto, es posible que distintos conjuntos de parámetros den resultados equivalentes. Por ejemplo, si se está modelando el volumen de un recipiente y se tiene como parámetros la altura y la anchura del mismo, estos parámetros podrían intercambiar sus valores dando como resultado la misma salida y siendo calibraciones equivalentes. No obstante, no se trataría del mismo modelo, sino de uno equivalente. Para solucionar este problema se puede hacer uso de las restricciones anteriormente descritas, y así, mediante el uso de las mismas, restringir los rangos en los que se pueden mover los parámetros en valores absolutos y lo que pueden variar respecto al tiempo. Por otro lado, haciendo uso de los pesos de la función de coste se podrá dar prioridad en la variación de un u otro parámetro.

Otra diferencia destacable será la eliminación del filtro de corrección del error de modelado que utiliza el controlador PNMPC. Sin embargo, puesto que el corrector PNMPC solo actúa cuando el error medio supera cierto nivel en un horizonte de corrección N_{cor}, se atenúa la necesidad del recálculo de parámetros por efecto del ruido en la salida.

3.2. Modelo de parámetros

El modelo de parámetros será un modelo equivalente al modelo que describe el sistema, con la diferencia de que los parámetros de configuración del modelo se convertirán en entradas y las entradas se convertirán en perturbaciones, tal y como se muestra en la Figura 2. Así, se obtiene un modelo no lineal que proporciona la salida del sistema en función de sus parámetros, sin necesidad de modificar sus ecuaciones internas.

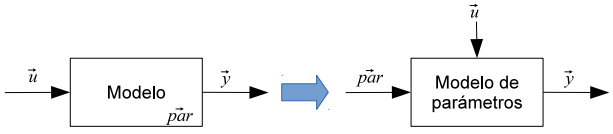


Figura 2: Modelo de parámetros

3.3. Disparador

El disparador tiene la función de activar o desactivar el corrector PNMPC en función del error en las predicciones pasadas del modelo, activando la corrección, $act = 1$, cuando la raíz del error medio cuadrático, $eRMSE$, sobrepasa una sensibilidad sen_{cor} de acuerdo con las ecuaciones (12) y (13). Por lo tanto, cuando sen_{cor} sea nulo el corrector PNMPC se ejecutará de forma continua, pasando a tener una ejecución basada en eventos cuando

sen_{cor} sea mayor de cero.

$$act = \begin{cases} 1 & si \quad eRMSE \geq sen_{cor} \\ -1 & si \quad eRMSE < sen_{cor} \end{cases} \tag{12}$$

$$eRMSE = \sqrt{\frac{\sum_{j=N_{cor}-1}^{0} \|\vec{y}_{pred}(k-j) - \vec{y}(k-j)\|^2}{N_{cor}}} \tag{13}$$

4. Resultados

En la presente sección se presentan ejemplos del desempeño de la estrategia de corrección. Tras mostrar el modelo utilizado para las simulaciones se presentarán, primero, resultados únicamente del funcionamiento como corrector o diagnosticador de modelos, para, más adelante, mostrar los beneficios del uso del corrector en el control predictivo.

4.1. Modelo de un tanque

Para la prueba de la estrategia de modelado y control se hará uso del modelo de un tanque. Dicho sistema, cuya descripción se puede ver en la Figura 3, se trata de un sistema clásico de un tanque, pero con la particularidad de que sus parámetros son variantes en el tiempo. Extrapolando a un sistema real, se podría tratar, por ejemplo, de un tanque intermedio en un sistema de tratamiento de aguas. En dicho tanque, el agua que entra contiene sedimentos, los cuales pueden adherirse a las paredes del tanque y hacer variar el área de la base, A, y/o taponar parcialmente el orificio de salida, variando el área del mismo, a. En la ecuación (14) se representa la dinámica de este sistema.

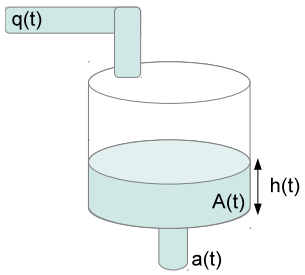


Figura 3: Sistema de un tanque

$$\frac{dh(t)}{dt} = \frac{q(t)}{A(t)} - \frac{a(t)}{A(t)}\sqrt{2gh(t)} \tag{14}$$

donde $A(t)$ es el área de la base del tanque $[m^2]$, $a(t)$ es el área de la apertura de salida del tanque $[m^2]$, $q(t)$ es el caudal de entrada de líquido al tanque $[m^3\ s^{-1}]$, $h(t)$ es la altura de líquido dentro del tanque $[m]$ y g es la aceleración originada por la gravedad $[m\ s^{-2}]$

4.2. Corrección de modelos sin ruido

En la Figura 4 se muestra el efecto de la corrección de parámetros del modelo a través de los valores de su salida, comparando con la salida del sistema y con la salida del modelo sin variar los parámetros iniciales. Los valores de los parámetros iniciales se muestran en la Tabla 1 y los parámetros del corrector se muestran en la Tabla 2. En las Figuras 5.a y 5.b se muestra la variación de los parámetros del modelo y en la Figura 5.c se muestra el error en la salida y los instantes en los cuales el corrector PNMPC se ha ejecutado realizando correcciones. Como se puede observar en la Figura 4, el modelo sigue al sistema con bastante precisión, siendo el error medio de predicción $error_{rmse} = 0{,}0513[m]$.

Tabla 1: Parámetros iniciales

Param.	**Valor**	**Param.**	**Valor**
$h(0)$	$10[m]$	$h_{pre}(0)$	$10,2[m]$
$A(0)$	$5[m^2]$	$A_{pre}(0)$	$5,2[m^2]$
$a(0)$	$0,5[m^2]$	$a_{pre}(0)$	$0,45[m^2]$

donde $h_{pre}(0)$ es la altura del modelo del tanque $[m]$, $A_{pre}(0)$ es el área de la base del modelo del tanque al inicio $[m^2]$ y $a_{pre}(0)$ es el área de la apertura de salida del modelo del tanque $[m^2]$.

Tabla 2: Configuración del corrector PNMPC

Param.	**Valor**	**Param.**	**Valor**
N_{cor}	$3[-]$	N_{par}	$3[-]$
Δpar_{min}	$[-0{,}1; -0{,}01][m^2]$	sen_{cor}	$0,04[m]$
Δpar_{max}	$[0{,}1; 0{,}01][m^2]$	P_{cor}	$1[-]$
par_{max}	$[5{,}3; 0{,}6][m^2]$	P_{par}	$[1; 10][-]$
par_{min}	$[4{,}5; 0{,}4][m^2]$	Y_{min}	$0[m]$
Y_{max}	$30[m]$		

Hay que considerar que la técnica que se usa para la corrección de los modelos está basada en el controlador PNMPC [7]. Este controlador para sistemas no lineales funciona linealizando el modelo de predicciones en cada periodo de muestreo, por lo que, en realidad, no trabaja con el modelo no lineal, sino con las diferentes linealizaciones en los distintos puntos de operación por los que va pasando. Esto implica que el corrector PNMPC en realidad no ajusta los parámetros del modelo no lineal, sino los de un modelo no lineal cuya linealización en un punto de operación coincide con la linealización en el mismo punto de operación que el sistema al que se desea ajustar.

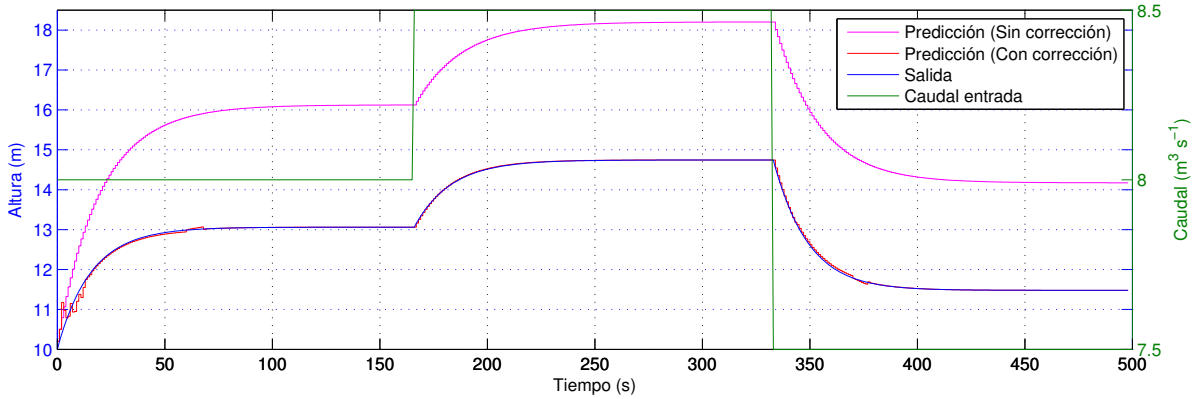


Figura 4: Ajuste de las salidas del modelo sin ruido en la salida real

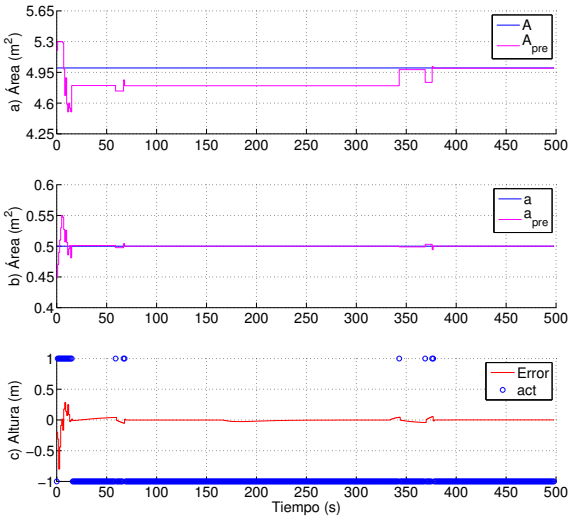


Figura 5: Variación de los parámetros del modelo, error de modelado y activación del disparador

Es por lo anteriormente mencionado que, tal y como se muestra en las figuras 4 y 5, el cambio en el punto de operación, una vez el modelo ya está ajustado, puede provocar, en función del sistema que se este modelando y de la magnitud de la variación que se produzca, que el corrector considere necesario o no el recálculo de los parámetros del modelo. En esto también influirá la sensibilidad de corrección elegida, sen_{cor}, y el horizonte de la misma, N_{cor}.

4.3. Corrección del modelo con ruido en la salida

En lo que se refiere al ruido en la salida, éste podría provocar un recálculo constante e innecesario de los parámetros del modelo. Esto se podría evitar filtrando las mediciones de la salida. Sin embargo, puesto que el corrector solo entra en funcionamiento cuando el error medio $eRMSE$ supera la sensibilidad sen_{cor} en un horizonte de corrección N_{cor}, si el ruido en las mediciones de la salida se trata de un ruido blanco, esto es de media cero, y la sensibilidad y el horizonte de corrección son suficientemente altos, el propio mecanismo de disparo del corrector actuará como un filtro del ruido. De esta forma, como se puede observar en las Figuras 6 y 7, haciendo una adecuada calibración de los parámetros, el sistema de corrección funcionará bien ante mediciones de la salida del sistema real con ruido.

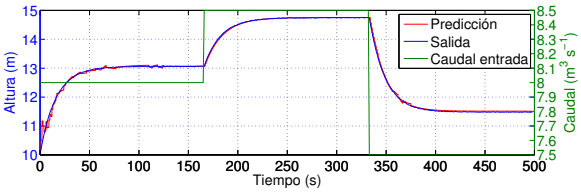


Figura 6: Ajuste de las salidas del modelo con ruido en la salida real

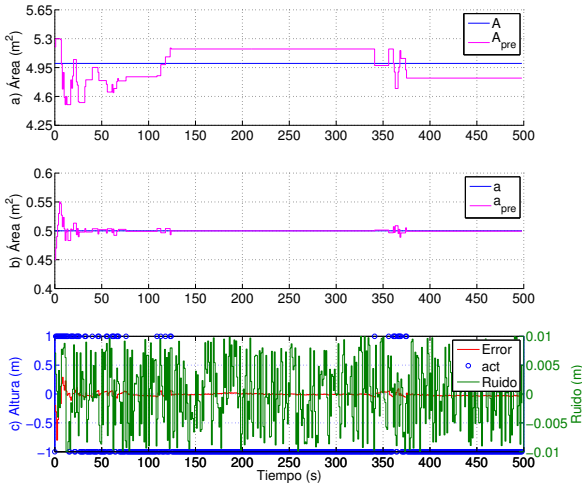


Figura 7: Variación de los parámetros del modelo, error de modelado, activación del disparador y ruido en la salida

4.4. Corrección y diagnosis de sistemas variantes en el tiempo

El verdadero potencial de la técnica descrita en este artículo no está en la corrección de parámetros constantes, ya que para esto sería más útil una buena calibración del modelo, sino la corrección de modelos que representen a sistemas con dinámicas variables en el tiempo; bien sea por la propia naturaleza del sistema, o bien por posibles fallos o perturbaciones no deseadas. Extrapolando al sistema de un tanque utilizado, esto se podría corresponder con una disminución en el área de la base del tanque, A, por una acumulación de sedimentos, o bien por un taponamiento que pudiera producir una reducción en el área del orificio de salida de dicho tanque, a.

Tal y como se puede observar en la Figura 8, el modelo sigue en todo momento la dinámica del sistema, lo cual resulta útil para implementar una estrategia de control sobre el sistema del tanque. Por otro lado, tal y como se muestra en la Figura 9, el corrector PNMPC es capaz de detectar y corregir la variación en el área del orificio de salida del tanque. Esta corrección, además de servir con objetivos de control, ya que, en el caso de usarlo en control predictivo, se tendrá en todo momento un modelo de predicciones adecuado, servirá con objetivos de diagnóstico del sistema y detección de errores. Extrapolando a un sistema real donde el tanque sea un tanque intermedio en un sistema de tratamiento de aguas, el sistema sería capaz de detectar y avisar al operario de la planta de que la salida del tanque ha sido obstruida por algún tipo de residuo.

En la Figura 9 se observa también cómo se modifica el parámetro A_{pre}, sin que esto fuera en realidad necesario. Esto es debido a dos motivos; primero, tal y como se comento anteriormente, se podrían tratar de modelos equivalentes; y en segundo lugar, a la sensibilidad en la corrección sen_{cor}. Además, hay que notar que la influencia del parámetro A_{pre} en la dinámica del modelo es muy pequeña en comparación a la influencia del parámetro a_{pre}.

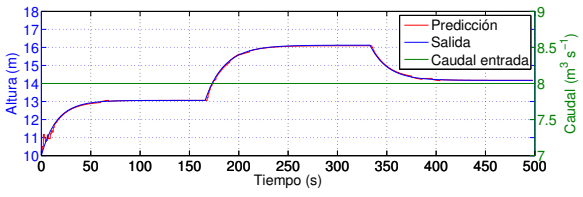


Figura 8: Ajuste de las salidas del modelo ante un sistema variante en el tiempo

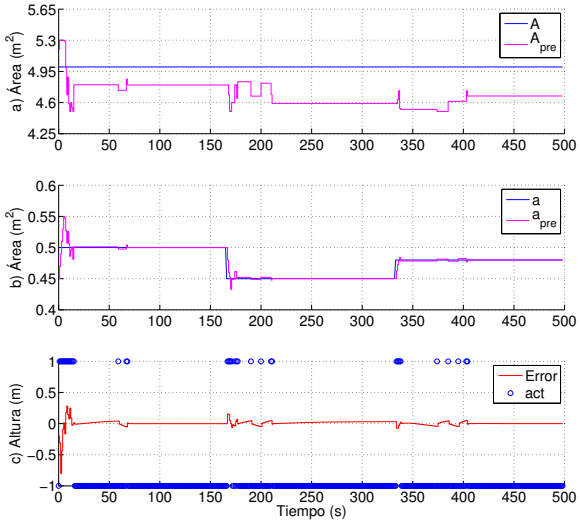


Figura 9: Detección de la variación de los parámetros del sistema

4.5. Aplicación al control predictivo

Una clara aplicación de la estrategia de corrección de modelos presentada es la utilización de la misma en el control predictivo. De esta forma, además de tener un elemento para diagnosticar el comportamiento y el estado del sistema, tendrá utilidad como un elemento para mejorar el desempeño del control. Una posible configuración es la que se muestra en la Figura 1, usando un controlador PNMPC como controlador predictivo. Los parámetros iniciales del modelo y del sistema se muestran en la Tabla 1, los parámetros utilizados para configurar el controlador se muestran en la Tabla 3 y los parámetros del corrector PNMPC son los de la Tabla 2.

Tabla 3: Parámetros del controlador PNMPC

Param.	**Valor**	**Param.**	**Valor**
N	$5[-]$	N_u	$5[-]$
a_f	$0{,}8[-]$	Δu_{min}	$-0{,}1[m^3 s^{-1}]$
Δu_{max}	$0{,}1[m^3\ s^{-1}]$	u_{min}	$0[m^3 s^{-1}]$
u_{max}	$9[m^3\ s^{-1}]$	Y_{min}	$0[m]$
Y_{max}	$30[m]$	P_r	$1[-]$
P_q	$1[-]$		

En la simulación realizada se han considerado variaciones en la referencia, como se puede ver en la Figura 10, y un taponamiento temporal del orificio de salida del tanque, tal y como se muestra en la Figura 11. Con estas condiciones, y haciendo uso del esquema de la Figura 1, se obtienen los resultados que se muestran en la Figura 10, donde se puede observar como la salida sigue bastante bien las referencias marcadas y rechaza las perturbaciones producidas por el taponamiento de la

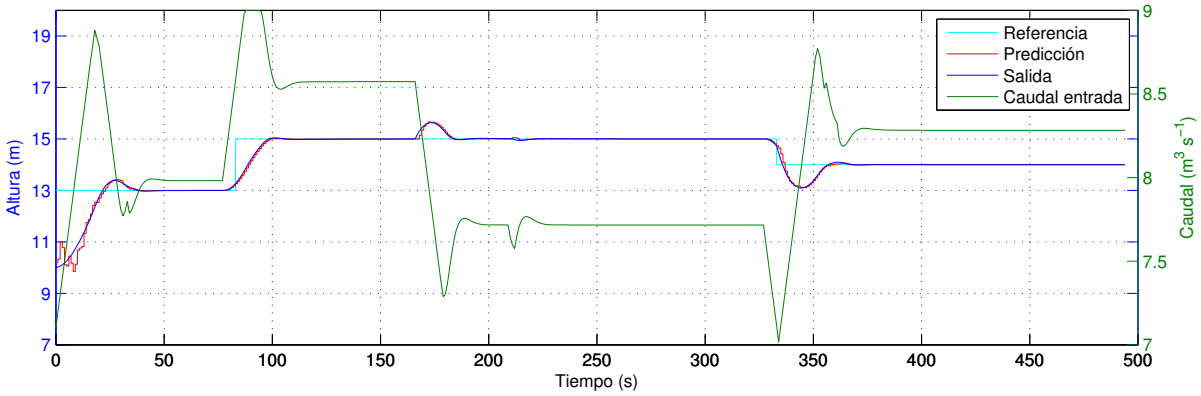


Figura 10: Seguimiento de referencias y señal de control con corrector de modelos

salida. Además, se observa como la predicción de la salida sigue siempre a la salida real, tanto en los instantes en los que se modifica la referencia, y por tanto el punto de operación, como en los instantes en los que aparecen perturbaciones producidas por la variación en la dinámica del sistema.

Por otro lado, en la Figura 11.b se observa como el modelo se corrige casi exclusivamente en los momentos en que el sistema cambia sus parámetros y, por tanto, su dinámica, proporcionando, tal y como se ve en la Figura 11.a, información sobre la diagnosis del estado en el que se encuentra el sistema real y; detectando, en este caso, cuando la salida de líquido se tapona y cuando vuelve a su condición inicial.

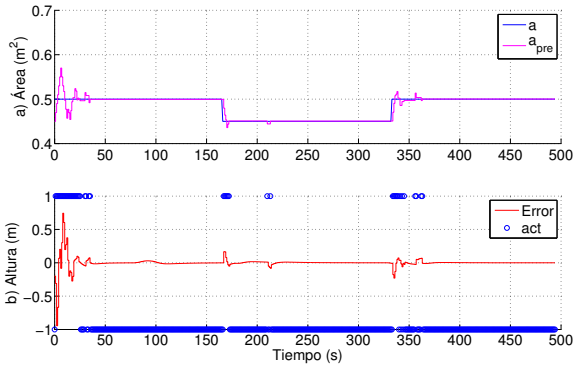


Figura 11: Variación del parámetro a y corrección del modelo

En la Figura 12 se muestran los resultados que se obtienen cuando se usa exactamente el mismo controlador con los mismos parámetros, pero no se hace uso del corrector de modelos. Se puede observar que, gracias a la corrección el error de modelado que hace la técnica PNMPC, es capaz de seguir la consigna; sin embargo, este esquema no permite obtener información sobre el estado en el que se encuentra el sistema y además tiene un peor desempeño, tal y como se puede ver en la Figura 13.

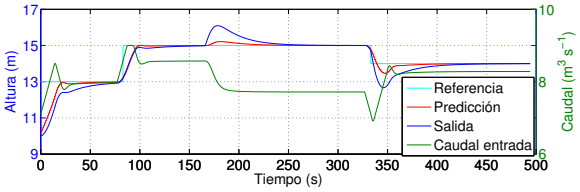


Figura 12: Seguimiento de referencias y señal de control sin corrector de modelos

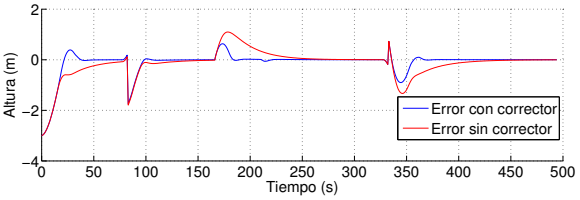


Figura 13: Comparación de los errores de seguimiento

Además, con una buena sintonización, el corrector PNMPC podría ser usado para sustituir al factor de corrección propio de la técnica PNMPC, tal y como se muestra en las Figuras 14 y 15.

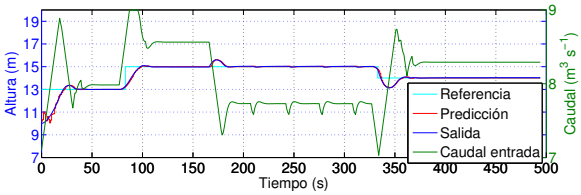


Figura 14: Seguimiento de referencias y señal de control con corrector de modelos y con $a_f = 1$

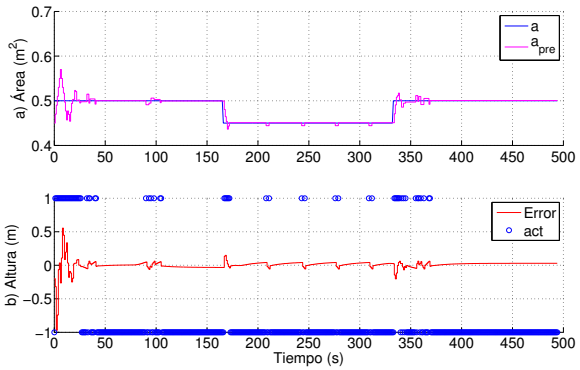


Figura 15: Corrección de parámetros y actuación del corrector de modelos con a_f=1

5. Conclusiones, posibles mejoras y trabajos futuros

Como se ha mostrado, la estrategia de corrección de modelos propuesta permite mejoras en el desempeño de un controlador predictivo además de la diagnosis de sistemas. En lo que se refiere a la diagnosis de sistemas, el poder calcular el valor de los parámetros del sistema cuando estos varían es un paso previo al diagnóstico de errores y solo faltaría hacer un tratamiento apropiado de los residuos. En este caso podría ser adecuado considerar residuos no solo el error en la predicción de las salidas, sino también las variaciones en los parámetros tal y como propone [1]. Por otro lado, las ventajas de tener un modelo de predicciones que ajuste su dinámica a la del sistema van desde un mejor seguimiento de referencias, hasta posibles ahorros en costes debido a un mejor ajuste de los parámetros internos, pasando por la posibilidad de implementar un controlador adaptativo que varíe su comportamiento en función del estado del sistema con fines de robustez. Además, el ajuste de los parámetros del modelo reduciría la necesidad del ajuste de la salida de forma continua por lo que facilitará la implementación de un control predictivo no lineal basado en eventos.

En lo que refiere a posibles mejoras y futuros trabajos se presentan tres líneas principales: la experimentación en plantas reales, la implementación de un mecanismo de diagnóstico de errores y de control tolerante a fallos, y el desarrollo de una técnica de control predictivo no lineal basado en eventos.

Agradecimientos

Los autores agradecen el apoyo financiero del Ministerio de Economía y Competitividad en virtud de la beca FPI enmarcada en los proyectos DPI2011-27818-C02-01 y DPI2011-27818-C02-02.

Referencias

[1] Thamara Villegas Berbesi. *Aplicación de técnicas robustas para detección y diagnóstico de fallos*. PhD thesis, Universidad de Valladolid, 2012.

[2] M. Blanke. What is fault-tolerant control? In *Proceedings of IFAC SAFEPRO-CESS'00*, volume 35, pages 123–126, 2000.

[3] L. F. Blazquez and L. J. de Miguel. *Diagnóstico de fallos basado en el modelo de la planta para sistemas no lineales*. Universidad de León. Secretariado de Publicaciones y Medios audiovisuales., 2003.

[4] E. F. Camacho and Carlos Bordóns. *Model predictive control*. Springer, 2004.

[5] M. Castilla, J.D. Alvarez, J. E. Normey-Rico, and F. Rodriguez. A nonlinear model based predictive control strategy to maintain thermal comfort inside a bioclimatic building. In *Control Automation (MED), 2012 20th Mediterranean Conference on*, pages 665–671, 2012.

[6] Agustín Pérez Castro. Modelado y control de un vehículo eléctrico mediante una estrategia de control predictivo basado en modelo. Master's thesis, Universidad de Almería, 2011.

[7] Agustinho Plucenio. *Desenvolvimento de Técnicas de Controle Não Linear para Elevaçao de Fluidos Multifásicos*. PhD thesis, Universidade Federal de Santa Catarina, 2010.

[8] V. Puig, J. Quevedo, T. Escobet, B. Morcego, and C. Ocampo. Control tolerante a fallos (parte i): Fundamentos y diagnóstico de fallos. *Revista Iberoamericana de Automática e Informática Industrial*, 1(1):15–31, Abril 2004.

[9] A. Pérez-Castro, J. E. Normey-Rico, J. L. Guzmán, and M. Berenguel. Modelado y control de un vehículo eléctrico mediante una estrategia de control predictivo basado en modelo. In *XXXIV Jornadas de Automática - Terrasa, Barcelona*, Junio 2013.

[10] Julio Elias Normey Rico, Paulo Renato da Costa Mendes, and Rodrigo Le ao Carvalho. *Relatório Controle Separador Trifásico*. Universidade Federal de Santa Catarina, 2011.

Sistema experto para la determinación de referencias en el proceso de elaboración de aceite de oliva virgen

P. Cano Marchal, D. Martínez Gila, J. Gámez García, J. Gómez Ortega
Grupo de Robótica, Automática y Visión por Computador
Universidad de Jaén
{pcano,dmgila,jggarcia,juango}@ujaen.es

Resumen

La selección de las referencias de las variables proceso de elaboración de aceite de oliva virgen (PE-AOV) se realiza normalmente de forma manual por parte de los operadores de la almazara. En este trabajo se construye y valida un sistema de apoyo a la decisión que propone valores para dichas referencias. Para ello, el sistema selecciona un objetivo de elaboración, propone valores iniciales para las referencias y los ajusta en función de las salidas del proceso.

Palabras clave: Control de Procesos, Aceite de Oliva Virgen, Sistema Apoyo a la Decisión, Lógica Borrosa.

1. Introducción

El proceso de elaboración de aceite de oliva virgen (PE-AOV) es un proceso industrial relativamente complejo, con diferentes variables implicadas y objetivos de producción contrapuestos [1] [2]. La calidad del aceite producido depende de las características de las aceitunas de entrada y de los valores de diferentes variables de proceso, mientras que algunas de estas variables afectan de forma opuesta a la cantidad de aceite obtenido. Adicionalmente, la máxima calidad potencial de aceite decae con el avance de la campaña de recolección, por tanto, la relevancia de las restricciones impuestas por el objetivo de obtener alta calidad también disminuye. Por todo ello, es importante establecer un buen objetivo de elaboración basado en las características de las aceitunas de entrada y asegurar que el valor de las variables del proceso es consistente con este objetivo.

Actualmente, la selección de las referencias de las distintas variables del proceso se lleva a cabo por parte de los operadores de la almazara, basándose en su experiencia, como es práctica habitual en la industria alimentaria [3]. Por otro lado, en relación al desarrollo de sistemas automáticos capaces de definir o sugerir valores al operador estos valores de las referencias, es relevante citar el trabajo de Bordons y Núñez-Reyes [4], donde los autores propusieron emplear control predictivo (MPC) como controlador de alto nivel de la almazara con el objetivo de maximizar el rendimiento industrial de la planta. Este controlador establecía las referencias de tres variables importantes del proceso, que a su vez eran utilizadas en controladores PID de bajo nivel. La influencia de las variables de proceso en la calidad del aceite producido se resolvió imponiendo restricciones a los valores aceptables para la temperatura de batidora. Este trabajo no consideraba el uso de diferentes valores de referencias para distintos objetivos de elaboración.

Trabajos relacionados que no especifican explícitamente referencias para las variables de proceso, pero que modelan el efecto de las variables en las salidas del mismo, se pueden encontrar en [5] y [6]. En [5], Furferi et al. desarrollaron un red neuronal artificial (ANN) para estimar las características de calidad del aceite elaborado en base a parámetros tanto agronómicos como de proceso. Por su parte, en [6] los autores emplearon variables de proceso y características del fruto de entrada para predecir el contenido graso y la humedad del orujo. Estos trabajos, aunque pueden servir de asistencia a los operadores de la almazara para seleccionar los valores de referencia de las variables del proceso, no proporcionan indicaciones intuitivas y directas. Es más, estos trabajos se apoyan exclusivamente en información captada por sensores, y no incluyen la experiencia y *capacidad sensorial* de operadores expertos.

Finalmente, en la industria alimentaria es frecuente el empleo de técnicas borrosas para el control en lazo cerrado y el control de calidad [7]. La lógica borrosa permite modelar hasta cierto punto el conocimiento experto de los operadores del proceso, derivando acciones de control de la información obtenida de diferentes sensores. Es más, las técnicas borrosas permiten el uso del experto como un sensor cuando no hay otro tipo de sensores disponibles [3].

El objetivo de esta comunicación es prfesentar un sistema de apoyo a la decisión que proponga los valores de las referencias de las variables de proceso e incorpore un mecanismo de realimentación

que los corrija en caso de desviación de las salidas de la planta. Para conseguir este objetivo, el sistema utiliza conocimiento experto para derivar un conjunto de reglas que relacionan las variables de entrada y salida del sistema, y emplea al experto como un sensor para las variables que, bien no existen sensores actualmente para su medición, bien éstos son caros y no se encuentran disponibles habitualmente en las almazaras.

El resto de la comunicación se organiza de la siguiente manera: la próxima Sección expone brevemente el proceso de elaboración de aceite de oliva virgen; la Sección 3 presenta el diseño del sistema experto, y la Sección 4 incluye su validación. Finalmente, la Sección 5 contiene las conclusiones.

2. Proceso de elaboración de aceite de oliva virgen

El PE-AOV comienza con la recepción del fruto en la almazara. Estas aceitunas se lavan para quitar el polvo, las piedras y las hojas que normalmente contienen. Tras esta etapa preliminar, las aceitunas se almacenan en tolvas y se alimentan al molino, donde se muelen para formar la pasta de aceituna. La pasta en este estado no permite una buena separación del aceite, por lo que se bombea a una termobatidora donde se calienta y remueve lentamente. Una vez que la pasta está preparada, se inyecta al decánter donde se produce la separación de las fases y se extraen orujo y aceite. El contenido de humedad e impurezas del aceite es aún elevado, por lo que es necesaria una operación de separación adicional, llevada a cabo bien en centrífuga vertical, bien en depósitos aclaradores. Tras esta operación, el aceite se puede filtrar o bombear directamente a bodega para su conservación.

Este proceso de elaboración se puede dividir en tres operaciones principales: la preparación de la pasta, la separación del aceite del resto de componentes de la pasta y la eliminación del exceso de humedad e impurezas. La preparación de la pasta está compuesta del almacenamiento del fruto, la molienda y el batido, llevado a cabo en la termobatidora. La separación del aceite incluye la operación en el decánter, y la eliminación del exceso de humedad e impurezas el resto de operaciones.

Las dos principales variables de salida del proceso global son la calidad de aceite obtenido (q) y el agotamiento (y). Estas variables presentan una cota superior impuesta por las características de la aceituna que se va a procesar, y el valor efectivamente obtenido depende de las diferentes variables del proceso. La preparación de la pasta determina en gran medida la calidad del aceite, e impone

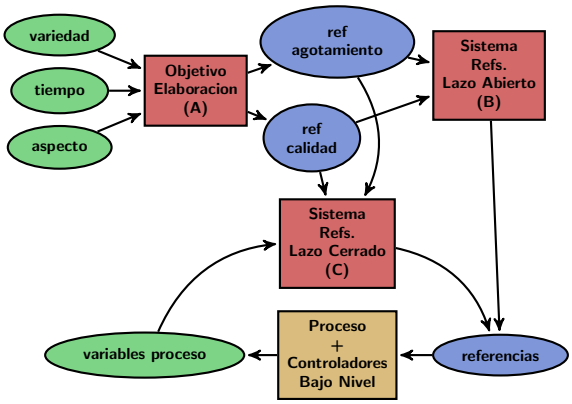


Figura 1: Mapa conceptual del proceso de toma de decisiones de un experto para el establecimiento de referencias del PE-AOV

una cota superior en el agotamiento. Por su parte, la operación de separación del aceite de la pasta influye en el agotamiento obtenido, pudiendo alcanzar valores por debajo del óptimo si la operación no se lleva a cabo de manera adecuada. La eliminación de la humedad e impurezas presenta una influencia menor tanto en la calidad como en el agotamiento [1]. Las principales variables implicadas en cada fase se recogen en la tabla 1, y su influencia en las salidas se trata en más detalle en las siguientes secciones.

3. Diseño del sistema de apoyo a la decisión

La característica clave del PE-AOV es la existencia de un compromiso entre calidad (q) y agotamiento (y), puesto que impone restricciones adicionales a las posibles combinaciones de calidad y agotamiento más allá de las impuestas por las características propias del fruto. Tratar de alcanzar mayor calidad normalmente supone relajar las expectativas sobre el agotamiento, mientras que la obtención de altos agotamientos implica bajar la calidad del aceite obtenido. Este hecho, junto a la evolución en el tiempo de la máxima calidad alcanzable, implica la necesidad de considerar diferentes objetivos de proceso como distintos pares de valores factibles de agotamiento y calidad. Es más, los valores de las referencias del proceso se deberían ajustar en línea para asegurar la consecución del objetivo.

Actualmente, este proceso de decisión se realiza, en general, manualmente; sin embargo, se podría estructurar y controlar automáticamente siguiendo el mapa conceptual presentado en la Figura 1. A partir de este diagrama se puede ver que la determinación de los valores de referencia de las variables del PE-AOV se puede descomponer en

Cuadro 1: Variables implicadas en cada operación

Operación	Símbolo	Variable
Prep. de la pasta	t_s	Tiempo almacenamiento
	C	Tamaño de la criba
	t_b	Tiempo de batido
	T_b	Temperatura de batido
	A_t	Adición de microtalco
Separación	F_p	Ritmo de inyección al decánter
	F_w	Adición de agua
	r_1	Posición de presillas
	$\Delta\omega$	Velocidad diferencial tornillo-bol
Elimin. humedad	t_r	Tiempo de residencia en tanque
	f_p	Frecuencia de purga
	$F_{w,vc}$	Adición de agua a centrífuga vertical
	d_{vc}	Discos utilizados en centrífuga

tres subproblemas distintos:

(A) Determinar el objetivo de elaboración, i.e., seleccionar la calidad y el agotamiento que aspiramos conseguir.

(B) Dado este objetivo de producción, proponer los valores de las referencias que permiten alcanzar dicho objetivo.

(C) Sugerir correcciones a los valores de las referencias utilizando realimentación para compensar desviaciones de los valores de salida deseados.

De acuerdo con esta descomposición del problema general, el sistema de apoyo a la decisión se diseñó como tres subsistemas, cada uno enfocado en un subproblema.

Cada subsistema fue construido utilizando lógica borrosa. Las variables utilizadas y sus funciones de pertenencia fueron inferidas de las respuesta de un cuestionario presentado a operadores expertos del proceso. El cuestionario preguntaba los distintos escenarios de elaboración considerados por los expertos, los valores típicos de la variables en ellos y las correcciones en caso de discrepancia entre el objetivo y la salida real del proceso. Las siguientes Subsecciones presentan los distintos subsistemas y diferentes aspectos de la construcción del sistema de apoyo a la decisión.

3.1. Definición del objetivo de elaboración

Dado que la elaboración de aceite de oliva virgen es una actividad económica, un criterio racional para la selección del objetivo de producción es la maximización del beneficio. Por tanto, el primer subproblema requiere encontrar una combinación factible de calidad (q_r) y agotamiento (y_r) que maximice el retorno económico de la actividad. Esto se puede plantear como un problema de optimización sujeto a restricciones impuestas por las aceitunas a procesar, el proceso y variables propias de la empresa que lleva a cabo la actividad.

Una formulación matemática del problema sería:

$$\begin{aligned}
&\mathbf{Max}: J(q_r, y_r, C_i) && && (1a)\\
&\quad \mathbf{s.t.}\ q_r && \leq q_r^{max} && (1b)\\
&\qquad y_r && \leq y_r^{max} && (1c)\\
&\qquad h_1(q_r, y_r) && = 0 && (1d)\\
&\qquad h_2(q_r, y_r, E, M) && = 0 && (1e)
\end{aligned}$$

C_i representa la cantidad de aceite producido de cada calidad. Las ecuaciones (1b) y (1c) introducen la cota superior en calidad (q_r^{max}) y agotamiento (y_r^{max}) impuesta por las aceitunas a procesar. Esta restricción introduce la noción de *estimación de la máxima calidad alcanzable*, que es un concepto clave en el proceso de toma de decisiones.

Los compromisos y limitaciones impuestos por el proceso se recogen en la ecuación (1d). Las restricciones propias de la empresa pueden incluir límites en la cantidad total de aceite que la compañía es capaz de vender de una determinada calidad, los costes de producción y los precios de venta de las diferentes calidades. Esto es considerado en la restricción (1e), donde E representa las variables relacionadas con la empresa, y M las relacionadas con el mercado, tales como los precios.

En esta comunicación obviamos las restricciones impuestas por la ecuación (1e), suponiendo que es óptimo producir el aceite de la mayor calidad posible, centrándonos, por tanto, en el modelado de las restricciones dadas por las aceitunas y el proceso, i.e., las ecuaciones (1b),(1c) y (1d). Así, la atención está centrada en la obtención de una estimación de la máxima calidad alcanzable y en el modelado de la relación entre calidad y agotamiento esperados.

Bajo estas suposiciones, las variables de entradas consideradas para el sistema son:

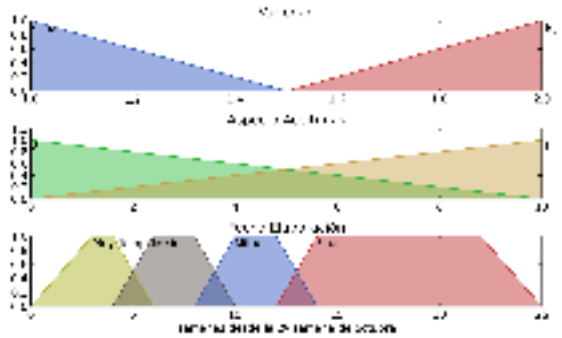


Figura 2: Variables de entradas del subsistema de determinación del objetivo de producción.

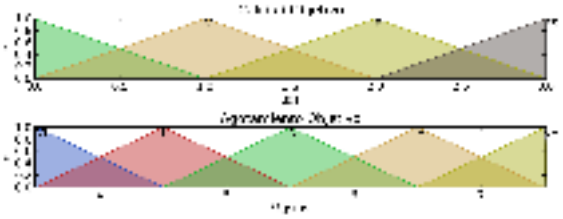


Figura 3: Variables de salida del subsistema de determinación del objetivo de producción.

- Momento de recolección (t), expresado como el número de semanas a partir de una fecha determinada,
- Aspecto de las aceitunas (OA),
- Variedad del fruto (V).

El momento de recolección (t) impone una cota superior a la máxima calidad alcanzable, mientras que el aspecto de las aceitunas trata la posibilidad de que el lote de aceitunas que se va procesar esté dañado, no pudiendo alcanzar la calidad esperada atendiendo solo al momento de recolección. Finalmente, la evolución de la maduración y calidad es diferente para distintas variedades; por tanto, este factor debe ser tenido en cuenta en el sistema experto. A pesar de que es usual que haya distintas variedades en una misma zona geográfica, normalmente se procesan por separado, y este es el enfoque seleccionado para el sistema experto.

Cabe destacar que, en este sistema, el conocimiento del experto se emplea de dos formas distintas:

- Como fuente de conocimiento para la construcción del sistema de reglas borrosas y funciones de pertenencia, y
- Como un sensor, puesto que es necesario su valoración del aspecto de las aceitunas.

Las figuras 2 y 3 muestran las definiciones de las funciones de pertenencia para los valores lingüísticos par cada variable. El uso de funciones triangulares o trapezoidales estuvo determinado por la información suministrada por el experto: si consideraban un único valor como representativo de total pertenencia a un término lingüístico se utilizaba una función triangular, empleando una función trapezoidal si consideraban un intervalo. Este criterio se utilizó también para los otros dos subsistemas.

Las reglas que ligan cada antecedente - tiempo (t) , variedad (OV) y aspecto (OA) — con su consecuente — calidad y agotamiento objetivo (q_r, y_r) — tienen la siguiente sintaxis:

$$R_i: \quad \text{Si } t \text{ es } D_1 \text{ y } OV \text{ es} D_2 \text{ y} OA \text{ es} D_3: \tag{2a}$$
$$\text{Entonces } q_r \text{ es } V_1 \text{ y } y_r \text{ es } V_2 \tag{2b}$$

Para la implementación de sistema, el agotamiento se expresa como *grasa sobre materia seca* ($gsms$), y no depende del contenido total de aceite de las aceitunas. Los resultados obtenidos y la evolución de las salidas con las entradas se presentan en la Sección 4.

3.2. Definición de las referencias

El segundo subproblema toma como entrada el objetivo de elaboración definido por el sistema anterior, junto con variables adicionales que definen las condiciones del fruto que no se han tenido en cuenta en la fase de establecimiento del objetivo de elaboración. Las salidas son los valores iniciales de las referencias de las variables del proceso que permiten obtener dicho objetivo de elaboración. Este subproblema supone obtener algún tipo de modelo invertible de la influencia de las referencias tanto en la calidad como en el agotamiento.

Matemáticamente, la solución a este subproblema debe proveer relaciones de la forma:

$$\mathbf{sp} = \mathbf{f}(\mathbf{o}, \mathbf{e}), \tag{3}$$

siendo $\mathbf{sp}$ las referencias de las diferentes variables de proceso, $\mathbf{o} = [q_r, y_r]^T$ el objetivo de producción y $\mathbf{e}$ las características adicionales del fruto.

Para la construcción de este subsistema, se solicitó a los expertos que definieran los valores de las referencias que seleccionarían inicialmente para cada objetivo de elaboración. Dado que la relación entre agotamiento objetivo (y_r) y calidad objetivo (q_r) está determinada en el subsistema anterior y es unívoca, únicamente se considera q_r como entrada a este subsistema. La variable de entrada adicional es la humedad de las aceitunas (H_o), dado que su influencia se puede obviar para la definición del objetivo de elaboración, pero

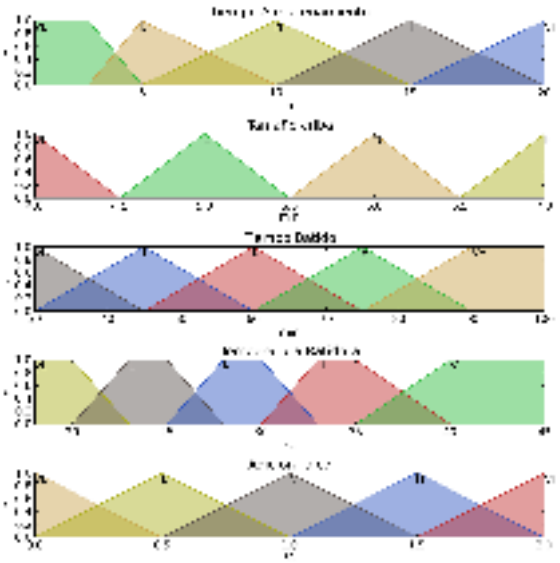


Figura 4: Variables de entradas del subsistema de determinación de referencias en lazo abierto.

debe tenerse en cuenta para la definición de las referencias del proceso.

Dado que la preparación de la pasta y la separación del aceite son procesos hasta cierto punto independientes, en este trabajo, a partir de este punto, nos centramos en la preparación de la pasta. Por tanto, las variables consideradas son $\mathbf{sp} = [t_s, C,\ T_b,\ t_b]^T$. La tabla 1 contiene las definiciones de las variables y la figura 4 muestra las funciones de pertenencia de cada una.

En este subsistema, de nuevo, el experto tiene dos roles: como fuente de conocimiento para construir el sistema y como sensor de contenido de humedad si no hay disponible un sensor en línea en la almazara, como suele ser el caso.

3.3. Ajuste de la referencias

El subsistema anterior propone los valores de las referencias en lazo abierto. Si los valores prescritos no permiten alcanzar el objetivo deseado, este subsistema no aporta indicaciones adicionales que permitan dirigir al sistema al estado deseado.

Por tanto, el subproblema actual requiere encontrar reglas para ajustar estos valores de referencia de forma consistente con los objetivos y las características del proceso. Para la solución de este problema se deben tener en cuenta dos aspectos:

- En general, existe más de una variable de entrada que influencia a la salida, y
- La variable de entrada puede afectar tanto a la calidad como al rendimiento.

De esta forma, la variable de entrada escogida y el valor de su incremento deben ser seleccionados tal que el sistema permanezca lo más cerca posible del objetivo fijado. Es decir, la acción de control se debe seleccionar de la variable que menos perturbe el valor de la variable de salida que no se pretende modificar y teniendo en cuenta efectos de saturación.

Una formulación matemática para la solución de este problema deben ser relaciones del tipo:

$$\mathbf{sp}^{k+1} = \mathbf{f_3}(\mathbf{pv}^k, \mathbf{sp}^k, \mathbf{o}), \tag{4}$$

i.e., los nuevos valores de las referencias ($\mathbf{sp}^{k+1}$) son función de sus valores actuales ($\mathbf{sp}^k$), los objetivos de elaboración ($\mathbf{o}$) y el valor real de las variables de proceso ($\mathbf{pv}^k$). Aquí (k) representa el instante de muestreo.

La Figura 5 ilustra la influencia de las diferentes variables del proceso sobre las salidas. La parte izquierda de la figura contiene las variables implicadas en la preparación de la pasta, y en la parte derecha están las variables que determinan el proceso de separación. Como puede apreciarse en la figura, la influencia de las variables del proceso de preparación de la pasta sobre el agotamiento (y) se ejerce a través de una nueva variable llamada *preparación de la pasta* (P_p), cuyo valor resume los efectos de estas variables.

Un operador experto puede proporcionar valores borrosos para esta variable (P_p) basándose en algunas características de la pasta en la batidora - la limpieza de las palas y la granularidad y el color de la pasta. Por tanto, esta variable se puede considerar como la salida del proceso de preparación de la pasta, y se puede utilizar para cerrar un lazo de control sobre este proceso.

Las entradas a este subsistema son las salidas del anterior, i.e., las referencias de las variables de proceso $[t_s, C,\ T_b,\ t_b,\ A_t]^T$, más:

- Objetivo de elaboración ($[q_r, y_r]^T$), es necesario en este subsistema también, pusto que las modificaciones en las referencias deben ser consistentes con este objetivo
- Calidad obtenida (q), agotamiento (y) y preparación de la pasta (P_p), requeridos para efectivamente comparar la salida con los valores deseados. Nótese que no se incluye un valor deseado de P_p, puesto que está implícito que el objetivo es obtener siempre el mejor valor posible sin violar las restricciones de calidad.

Las funciones de pertenencia para las variables de entrada son las mismas que las definidas en los

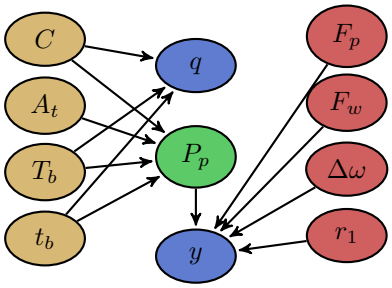


Figura 5: Influencia de las variables de proceso en las salidas. La existencia de un arco entre variables representa que la entrada influye en la salida.

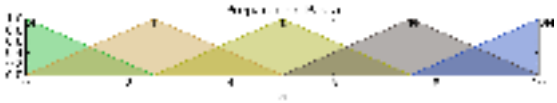


Figura 6: Variables de entrada adicionales del subsistema de determinación de referencias en lazo cerrado.

subsistemas anteriores para las variables comunes, y para las nuevas variables se definen en la Figura 6. Las reglas implementadas en el sistema se centran en la *preparación de la pasta* (P_p). Para cerrar el lazo de control sobre la calidad requiere una consideración más detallada de las características de calidad del aceite elaborado, puesto que existen diferentes desviaciones que se pueden deber a diferentes causas.

Las salidas del sistema son los incrementeos de cada variable del proceso, y sus funciones de pertenencia se recogen en la Figura 7. Un tiempo de muestreo típico para estos lazos es 60-90 min., i.e. alrededor del valor de t_b.

4. Validación del sistema

Una validación preliminar del sistema se llevó a cabo comprobando la concordancia de las salidas con los valores prescritos por los expertos en los cuestionarios. La Fig. 8 muestra la evolución de las salidas con el momento de recolección para la variedad Picual y buen aspecto de las aceitunas, y la Figura 9 muestra la evolución de T_b y A_t para el segundo subsistema. Los valores mostrados en las gráficas son consistentes con los valores indicados por los expertos disponibles.

El mayor número de entradas del tercer subsistema supone una mayor complejidad en la validación, aún así, se adoptó el mismo método. A modo de ejemplo se presentan dos escenarios distintos: la Fig.10 muestra la salida de ΔA_t y Δt_b para el caso $q_r = 3$ (muy alta calidad), como una función

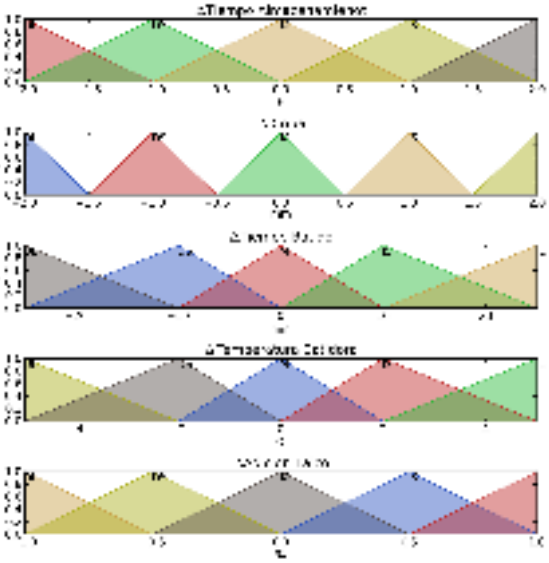


Figura 7: Variables de salida del subsistema de determinación de referencias en lazo cerrado.

del valor actual de A_t. Por su parte, Fig. 11 contiene la evolución de ΔA_t y ΔT_b frente A_t y T_b para $q_r = 1$ (calidad media-baja). De nuevo, los resultados obtenidos están en concordancia con el conocimiento experto.

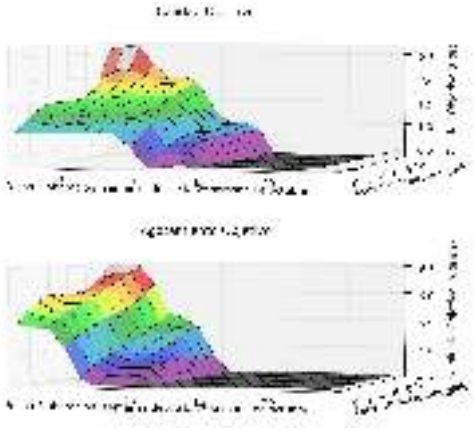


Figura 8: Evolución de las salidas del sistema de determinación del objetivo de elaboración para la variedad Picual.

5. Conclusiones

En esta comunicación se han presentado las bases de un sistema experto de apoyo a la decisión para la determinación de las referencias del PE-AOV. Se han introducido la noción de *máxima calidad alcanzable estimada* y el problema original se ha descompuesto en tres subproblemas, derivando un subsistema para cada uno de ellos. Además, se ha introducido la preparación de la pasta (P_p) como variable de salida del proceso de preparación de

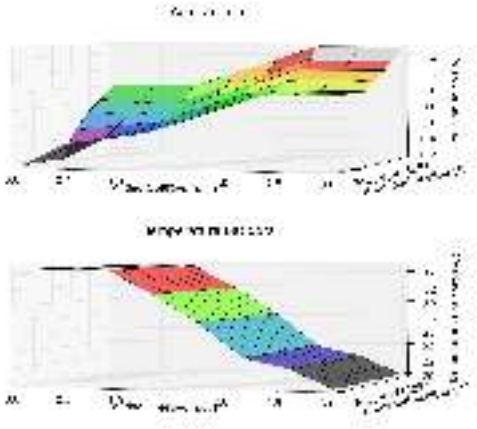


Figura 9: Evolución A_t y T_b, dos de las salidas del sistema de establecimiento de referencias en lazo abierto.

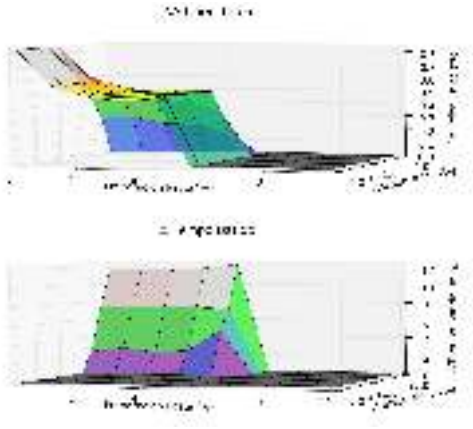


Figura 10: Evolución de ΔA_t y Δt_b con A_t y P_p para $q_r = 1$ y $q_r = 3$ (sistema de modificación de referencias en lazo cerrado).

la pasta, posibilitando cerrar un lazo de control sobre este proceso. Bajo este esquema, un sistema experto se ha construido y validado basado en el conocimiento de expertos en el proceso.

Las líneas de trabajo futuras consisten en la inclusión de los procesos de separación y eliminación de humedad e impurezas y en el cierre de un lazo de control explícito sobre la calidad y la mejora de la estimación del la máxima calidad alcanzable (q_r^{max}). Está prevista, asimismo, la extensión del cuestionario a un número mayor de expertos, con el objetivo de extender la base de conocimiento del sistema y mejorar su validación.

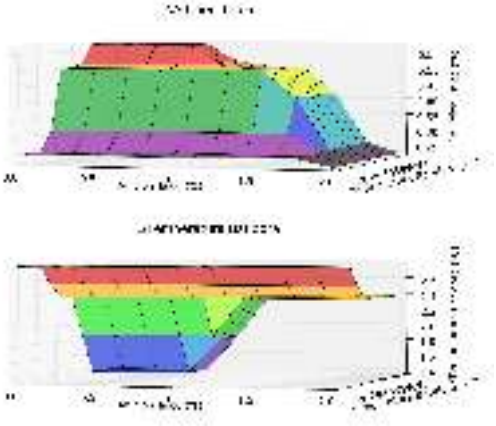


Figura 11: Evolución de ΔA_t y ΔT_b con A_t y T_b for $q_r = 1$ para $P_p = 1$ (sistema de modificación de referencias en lazo cerrado).

Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos DPI2011-27284, TEP2009-5363 and AGR-6429. Pablo Cano Marchal disfruta de una beca FPU del Ministerio de Educación.

Referencias

[1] P. Cano Marchal, J. Gómez Ortega, D. Aguilera Puerto, and J. Gámez García, "Situación actual y perspectivas futuras del control del proceso de elaboración del aceite de oliva virgen," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 8, no. 3, pp. 258–269, Jul. 2011.

[2] L. Civantos, *Obtención del aceite de oliva virgen.* Editorial Agrícola Española, S.A., 1998.

[3] N. Perrot, I. Ioannou, I. Allais, C. Curt, J. Hossenlopp, and G. Trystram, "Fuzzy concepts applied to food product quality control: A review," *Fuzzy Sets and Systems*, vol. 157, no. 9, pp. 1145–1154, May 2006.

[4] C. Bordons and A. Núñez-Reyes, "Model based predictive control of an olive oil mill," *Journal of Food Engineering*, vol. 84, no. 1, pp. 1–11, Jan. 2008.

[5] R. Furferi, M. Carfagni, and M. Daou, "Artificial neural network software for real-time estimation of olive oil qualitative parameters during continuous extraction," *Computers and Electronics in Agriculture*, vol. 55, no. 2, pp. 115–131, Feb. 2007.

[6] A. Jiménez Marquez, M. Aguilera Herrera, M. Uceda Ojeda, and G. Beltrán Maza, "Neural network as tool for virgin olive oil elaboration process optimization," *Journal of Food Engineering*, vol. 95, no. 1, pp. 135–141, Nov. 2009.

[7] S. Birle, M. Hussein, and T. Becker, "Fuzzy logic control and soft sensing applications in food and beverage processes," *Food Control*, vol. 29, no. 1, pp. 254–269, Jan. 2013.

DISEÑO DE LOS CONTROLADORES DE LOS SISTEMAS DE VACÍO Y TEMPERATURA DE UN REACTOR AGITADO

Ferran Babot Branzuela
Clariant Ibérica Producción S.A.
ferranbabot@gmail.com

Ramon Costa Castelló
Universitat Politècnica de Catalunya (UPC)
ramon.costa@upc.edu

Resumen

En este trabajo se presenta el modelado y diseño de los controladores de los sistemas de vacío y temperatura de un reactor agitado.

El trabajo presenta los modelos planteados y los experimentos realizados para validarlos y realizar el ajuste paramétrico correspondiente.

Para el sistema de vacío se presenta un controlador no lineal diseñado en el plano de fase. Mientras que para el sistema de control de temperatura se presenta una linealización por cambio de variable y posteriormente se diseñado un sistema de control por realimentación de estado.

El trabajo presenta también los resultados experimentales obtenidos en la instalación real y demuestra como se puede mejorar el rendimiento de una instalación mediante unos controladores más específicos.

Palabras Clave: Modelo, simulador, control por realimentación de estados.

1 INTRODUCCIÓN Y OBJETIVOS

Una de las aplicaciones básicas que se realizan en la industria química es la destilación. Se utiliza para separar dos productos que se encuentran disueltos, gracias a que tienen temperaturas de ebullición diferentes.

Una destilación se basa en calentar la masa del interior del reactor favoreciendo la evaporación del producto con una temperatura de ebullición inferior.

Es práctica habitual aplicar vacío en el reactor para bajar la temperatura de ebullición, reflejando mejoras de tipo económico y de calidad al no tener que sobrecalentar el reactor.

Para alcanzar este objetivo, se debe regular la temperatura y la presión en el interior del reactor durante el proceso de evaporación. Un control adecuado de las dos variables permite mejorar el rendimiento de la destilación.

El trabajo se estructura de la forma siguiente: la sección 2 describe la instalación y los componentes físicos del proceso, la sección 3 analiza el subsistema de vacío, la sección 4 analiza el subsistema de temperatura y finalmente en la sección 5 se presentan algunas conclusiones.

2 INSTALACIÓN Y PROCESO

La Figura 1 muestra un sinóptico del reactor con todos los elementos de la instalación: sensores (presión y temperatura) y actuadores (válvulas todo nada y de regulación).

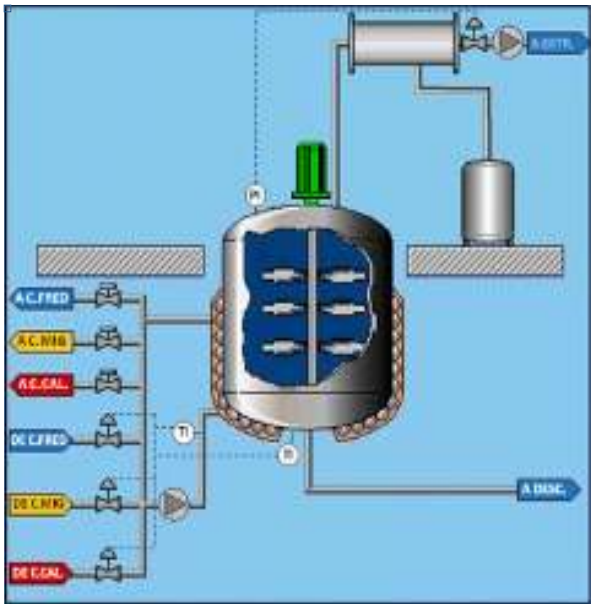


Figura 1 : Sinóptico del reactor.

Como elemento que permite hacer dismunir la presión en el interior del reactor, se utiliza una bomba de alto vacío de tornillo, conectada a la salida del condensador. El control del caudal de vapores se realiza con una válvula de regulación en el tubo que llega a la bomba (vahos) y un transmisor de presión para tener constancia de la presión que tiene en el equipo durante el proceso (Figura 2).

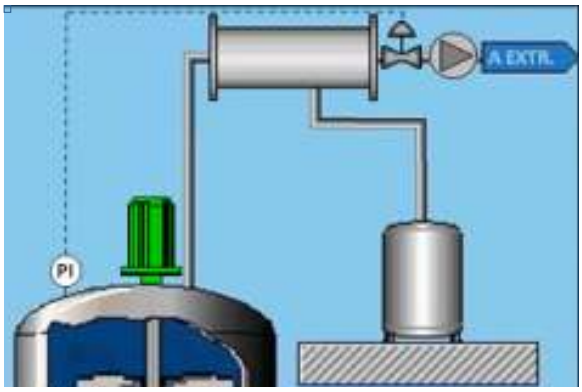


Figura 2 : Elementos de la instalación de vacío.

La temperatura del sistema se regula mediante la circulación de líquido por un serpentín en forma de media caña que envuelve el reactor por su exterior.

Por el circuito de la media caña circula un fluido térmico (en este caso glicol) que mantiene, aproximadamente, constantes sus propiedades dentro de las temperaturas de trabajo. Con la ayuda de una bomba se hace circular el glicol por el circuito de la media caña.

En la empresa donde se ha llevado a cabo éste proyecto, se modifica la temperatura del serpentín empleando glicol procedente de tres circuitos primarios (circuito caliente, circuito frio y circuito a media temperatura). Esto se consigue mediante unas válvulas de regulación en las tuberias que provienen de cada uno de los circuitos primarios y con válvulas todo/nada en los retornos a cada uno de ellos.

Para gobernar de forma óptima el caudal de glicol y determinar cual de los tres circutos hay que escoger, se dispone de dos sondas de temperatura: una en el circuito del serpentín y otra en el interior del reactor (Figura 3).

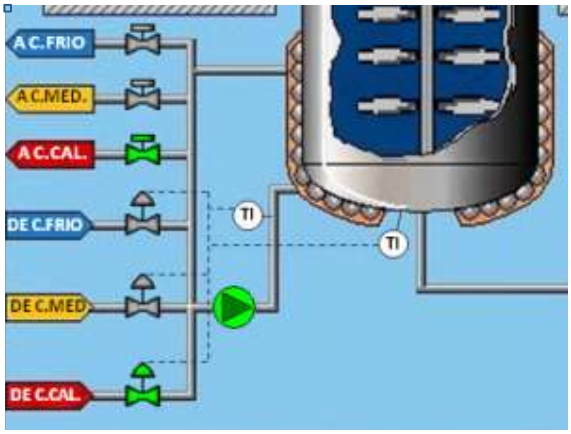


Figura 3 : Elementos de la instalación de temperatura.

3 SISTEMA DE VACÍO

Durante la destilación es necesario bajar la presión del reactor desde la presión atmosférica hasta los -900mbarG de manera escalonada (incrementos de -20, -50 o -100 mbarG como máximo), para evitar la generación de espuma en el interior del equipo. Si se hiciera el vacío bruscamente, el arrastre de producto generaría mucha espuma, cosa que ensuciaría la instalación y reduciría la calidad del producto final.

Es conocido que cuanto más lento es el proceso de reducción de la presión, mayor es el rendimiento en los condensados, favoreciendo, en este caso, su reutilización en el siguiente lote (batch), con el consiguiente ahorro de materia primera.

Para realizar las especificaciones será necesario encontrar un compromiso entre el tiempo que dura la destilación y el volumen condensado.

3.1 MODELADO

Con el fin de generar un modelo del sistema de vacío se realizan diferentes ensayos de respuesta al escalón sobre el sistema real (aberturas constantes sobre la válvula de salida de vahos).

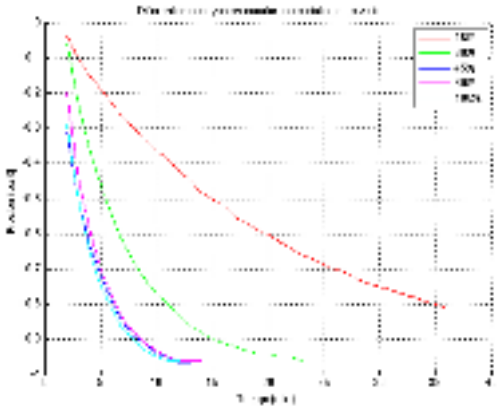


Figura 4 : Respuesta de la planta a diferentes entradas en escalón.

La Figura 4 muestra la respuesta del sistema frente a diferentes aperturas de la válvula (en tanto por ciento). Como se puede observar esta se asemeja a la descarga que presentaría un sistema lineal de primer orden. Dado que la constante de tiempo depende del valor de apertura se concluye que el sistema es no lineal. Con el fin de modelar el comportamiento de la planta, se plantea un modelo de primer orden no lineal de la forma :

$$\frac{dP(t)}{dt} = \frac{K(v(t)) \cdot v(t) - P(t)}{\tau(v(t))} - \alpha \cdot P(t) \qquad (1)$$

dónde $P(t)$ corresponde a la presión del reactor, $v(t)$ al porcentaje de apertura de la válvula, $K(v(t))$ y $\tau(v(t))$ son dos funciones de la apertura de la válvula y α es una constante que representa las pérdidas en la instalación (6mbar/min).
Las funciones $K(v(t))$ y $\tau(v(t))$ se han definido a partir de la observación de los datos experimentales y en toman la forma siguiente:

$$K(v(t)) = \frac{P_{Bomba_Vacío}}{v(t)} \quad (2)$$

$$\tau(v(t)) = \begin{cases} -0{,}73 \cdot v(t) + 27{,}4, & v(t) < V_\tau \\ 2, & v(t) \geq V_\tau \end{cases} \quad (3)$$

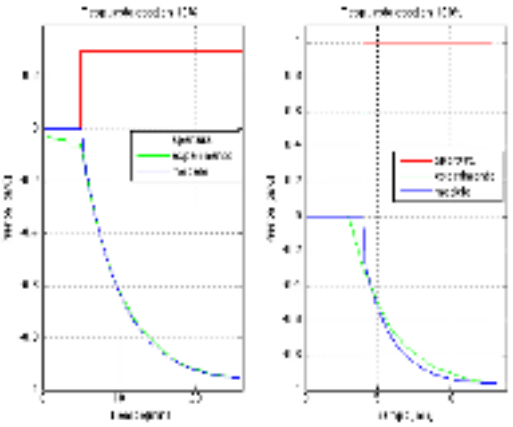


Figura 5 : Validación experimental del modelo del sistema de vacío planteado.

La Figura 5 muestra los resultados experimentales conjuntamente con los obtenidos con el modelo planteado para dos ensayos. Como se puede observar ambas respuestas son muy parecidas cosa que hace dar por bueno el modelo desarrollado.

3.2 CONTROLADOR

Algunos condicionantes que uno debe tener en cuenta, durante el diseño del sistema de control, son

- No se puede recuperar la presión mediante la válvula: la válvula sólo permite bajar la presión (carga). La recuperación (descarga) se produce por la generación de vapores y pérdidas, siendo un proceso muy lento.
- Para evitar la generación de espuma el sistema no debe superar el valor consignado más de 10mbar.
- No conviene realizar aperturas con porcentajes muy altos de válvula durante tiempos largos: ya que el rendimiento y calidad de recogida de condensados disminuye.

Dado que se trata de un sistema de primer orden su comportamiento se puede estudiar en un plano de fase definido por $P(t)$ y $\frac{dP(t)}{dt}$.
Como que el actuador no puede reducir la presión del sistema, en el caso que la presión deseada (SP) sea inferior a la presión actual la válvula se deja cerrada a la espera que la propia dinámica del sistema reduzca la presión.

En el resto se ha realizado un diseño por zonas que permite alcanzar los objetivos diseñados.

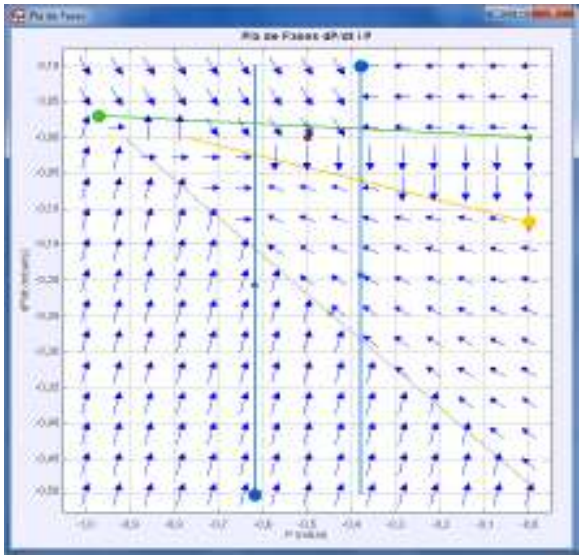


Figura 6 : Plano de fase de lazo cerrado del sistema de control de presión

La Figura 6 muestra el plano de fase de lazo cerrado con el controlador incorporado. Además de las líneas de flujo se visualizan diferentes líneas rectas que corresponden a los parámetros del sistema de control. Todo ello se ha implementado en un herramienta interactiva que permite diseñar el comportamiento de manera gráfica e interactiva. La Tabla 1 muestra los parámetros utilizados en las simulaciones realizadas par la comprobación del controlador con el modelo de la planta:

Parámetro	Valor
SP (Consigna)	0→-0.3→-0.6→-0.9barG
Kp (Constante del controlador)	200 %/bar
α (Generación de vapores y pérdidas)	6mbar/min
Saturación Válvula	Vmin=22% Vmax=50%
ΔpSup	30mbar
Δpinf	10mbar

Tabla 1 : Parámetros utilizados en las simulaciones

El sistema de control propuesto ha sido validado en el sistema real. La Figura 7 muestra la evolución de la presión y la apertura de la válvula a partir de la evolución de la presión de referencia (SP).

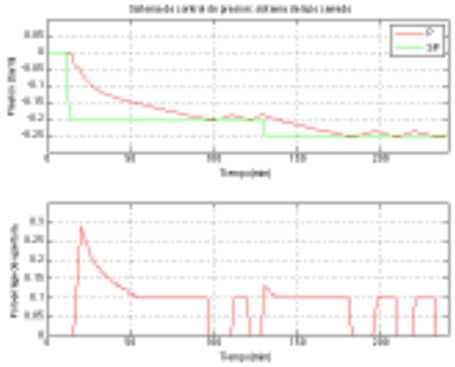


Figura 7 : Sistema de control de vacío, comportamiento de lazo cerrado

Como puede observarse la presión sigue perfectamente la consigna siguiendo una evolución lenta. A cambio de aumentar sensiblemente el tiempo de destilación, se ha conseguido un controlador que permite recoger cerca de un 5% más de condensados.

4 SISTEMA DE TEMPERATURA

En una destilación, se requiere llevar la temperatura del interior del reactor al punto al cual evapore el producto con una temperatura de ebullición más baja.

Interesa también, que no se supere el valor de la temperatura consignado, ya que la energía que se requiere para modificar pocos grados la temperatura de toda la masa del interior del reactor, representa unos costes energéticos anuales considerables. Al igual que no conviene tener oscilaciones en el régimen estacionario, por el mismo motivo. Cada oscilación equivale a un cambio de circuito general de glicol.

Por lo que concierne al controlador de la temperatura del serpentín, tampoco interesa tener oscilaciones en el régimen estacionario, pero no es tan crítico como en el caso del control de la temperatura interior. Es preferible la existencia de un ligero sobrepico (10 o 15%) a cambio que el serpentín alcance la temperatura lo más rápido posible.

4.1 MODELADO

Para este segundo sistema, el modelo se encuentra partiendo de las fórmulas que describen la dinámica del intercambio de las temperaturas del reactor y del serpentín [2]. A partir del balance de energía entre la pared del reactor y el líquido de su interior y el balance entre el fluido del serpentín y la pared exterior del equipo, se llega a las ecuaciones (2) y (3):

$$\frac{dT_r(t)}{dt} = -B \cdot (T_r(t) - T_s(t)) + D(t) \qquad (2)$$

$$\frac{dT_s(t)}{dt} = C \cdot V_{Circ}(t)\big(T_{Circ}(t) - T_s(t)\big) + E \cdot (T_r(t) - T_s(t)) \qquad (3)$$

Donde $T_r(t)$ corresponde a la temperatura del reactor, $T_s(t)$ corresponde a la temperatura del serpentín, $T_{Circ}(t)$ corresponde a la temperatura circuito escogido, $V_{Circ}(t)$ corresponde al porcentaje de apertura de la válvula del circuito escogido y finalmente $D(t)$ corresponde a una perturbación en el interior del reactor.
Las constantes B, C y E corresponden a:

$$C = \frac{K_{Circ}}{V_s} \; ; \; E = \frac{U_s A_s}{V_s \rho_s C_s} \; ; \; B = \frac{U_r A_r}{V_r \rho_r C_r} \qquad (4)$$

Donde los subíndices *s* y *r* hacen referencia al fluido del serpentín y al del interior del reactor respectivamente. Las constantes en (4) corresponden a:

A_i corresponde a la superficie de contacto entre fluido y la pared del equipo, U_i es el coeficiente de transferencia de calor entre fluido y la pared del equipo, V_i equivale a el volumen del fluido, ρ_i la densidad del fluido, C_i es el calor específico del fluido y K_{Circ} la relación entre la apertura de la válvula del circuito escogido y caudal que entra al circuito del serpentín [1].

Según la variación conocida de productos, volúmenes de ocupación del reactor y estado de los circuitos generales, se define a la Tabla 2, en la que se pude observar entre que valores se encuentran acotadas cada una de las tres constantes.

Ct.	Mín.	Máx.
B (1/min)	0,0139	0,0321
C (1/min)	0,00174	0,00883
E (1/min)	0,135	0,473

Tabla 2 : Rango de las constantes.

La ecuación (3), contiene una no linealidad: $V_{Circ}(t)\big(T_{Circ}(t) - T_s(t)\big)$. Para convertir el sistema en lineal, se procede a agrupar estos tres términos por otro: $VT_{dif}(t)$ que, podríamos decir, que se corresponde al flujo de calor que se aporta o se retira del serpentín.

Teniendo ya linealizado el sistema, las funciones de transferencia que describen su comportamiento son:

$$T_R(s) = \frac{B}{s+B} T_s(s) + \frac{1}{s+B} D(s) \qquad (5)$$

$$T_s(s) = \frac{C}{s+E} VT_{dif}(s) + \frac{E}{s+E} T_r(s) \qquad (6)$$

Estos modelos (utilizando para los modelos los valores nominales de B, C y E) se han validado experimentalmente.

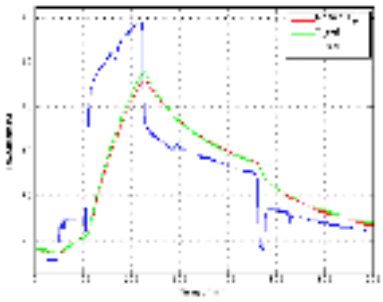


Figura 8 : Validación del modelo de la temperatura del serpentín.

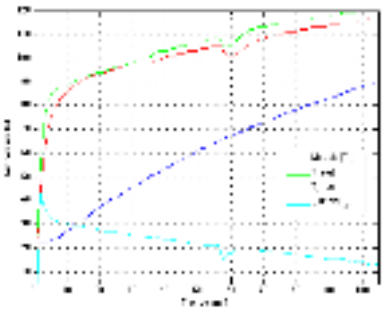


Figura 9 : Validación del modelo de la temperatura del reactor.

La **Figura 8** muestra la evolución de la temperatura del serpentín y la del modelo para un cierto $\boldsymbol{VT_{dif}(t)}$ y una cierta temperatura del reactor. De forma similar la **Figura 9** muestra la evolución de la temperatura del reactor para una cierta evolución de la temperatura del serpentín. Como se puede observar el ajuste del modelo es muy bueno cosa que valida los modelos obtenidos.

4.2 CONTROLADOR

En función de la etapa de la destilación es necesario controlar o bien la temperatura del reactor o bien la del serpentín; por ello se han diseñado dos controladores. Que aunque presentan la misma estructura presentan parámetros diferentes.

Las especificaciones del controlador de la temperatura del reactor son:
- Evitar las oscilaciones.
- No debe superar significativamente el valor consignado.
- El sistema tiene que ser capaz de llegar, o mantener, la consigna tanto con la válvula abierta, como cerrada.

Para el de la temperatura del serpentín, las especificaciones son:
- Evitar las oscilaciones, esta vez sólo en régimen estacionario.
- Puede sobrepasar la consigna y tener sobrepico, siempre que sea de pocos grados.
- Debe ir rápido.

Para poder cumplir con estas especificaciones, se escoge un controlador por realimentación de estados [1], siendo las variables de estado las temperaturas del serpentín ($T_s(t)$) y la del reactor ($T_r(t)$). Para poder conseguir error nulo en régimen estacionario, se añade un tercer estado que corresponde al error integrado de seguimiento entre la consigna y la temperatura que se desea controlar.

A la salida del controlador, en el que se obtiene el $\boldsymbol{VT_{dif}(t)}$ deseado, se añade un bloque encargado de escoger el circuito que mejor conviene en cada caso y de aplicar las saturaciones de las válvulas y temperaturas a que se encuentran cada uno de los tres circuitos generales.

Para evitar problemas con la saturación de $\boldsymbol{VT_{dif}(t)}$, se añade una elemento *Anti-Windup*[3], y una conexión *Bumpless transfer*[3] que permite conmutar entre el control de temperatura del reactor y el control de temperatura del serpentín sin transitorios poco deseables.

El valor de las constantes de los controladores (K_1, K_2 y K_3, para el control de las temperaturas del serpentín y del reactor) se halla mediante asignación de polos.

De los tres polos que tienen los dos sistemas en lazo cerrado se escogen 2 de ellos complejos, tal que cumplan con el sobrepico y el tiempo de establecimiento de las especificaciones, según se observa en la Tabla 3.

Sistema	Especif.	Polos
T_r	MP=0.01 $T_{S2\%}$=25	P_{1r}=(-0.125,0.085) P_{2r}=(-0.125,-0.085)
T_s	MP=0.1 $T_{S2\%}$=12	P_{1s}=(-0.250,0.314) P_{2s}=(-0.250,-0.314)

Tabla 3 : Polos especificados para el sistema de lazo cerrado.

Para elegir el tercer polo es necesario estudiar cada uno de los controladores por separado.

Temperatura del serpentín:
La función de transferencia de la planta es:

$$\frac{T_s(s)}{VT_{dif}(s)} = \frac{C(s+B)}{s(s+(B+E))} \qquad (7)$$

La planta tiene un cero en –B. Esta constante se encuentra acotada entre unos valores más pequeños que los polos deseados (Tabla 2). Tener un cero tan pequeño, comparado con los polos, provoca tener efectos no deseados en régimen transitorio, pudiendo provocar que el sistema llegue muy lentamente a la consigna.

Por este motivo el tercer polo, en el caso del controlador de la temperatura del serpentín, se escoge en el valor medio de la variación de la constante B. Así el efecto del cero se ve fuertemente atenuado.

En la Figura 10, se muestra el diagrama de polos y ceros de la planta y los polos deseados en lazo cerrado.

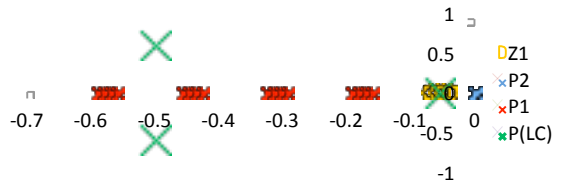


Figura 10 : Diagrama de polos y ceros de Ts.

Temperatura del reactor:
Sabiendo que para esta segunda planta, su función de transferencia es:

$$\frac{T_R(s)}{VT_{dif}(s)} = \frac{CB}{s(s+(B+E))} \qquad (8)$$

Se observa que, en este caso, no existe el cero. Se escoge un tercer polo que no sea dominante, como se muestra en la Figura 11.

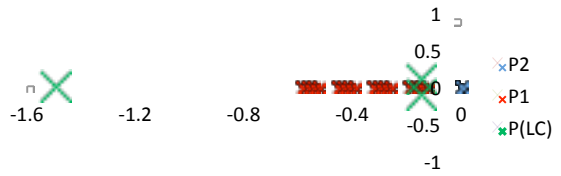


Figura 11 : Diagrama de polos y ceros de Tr.

Así finalmente la tabla de especificación de los polos de lazo cerrado queda como se muestra en la Tabla 4

Sistema	Especif.	Polos
T_r	MP=0.01 $T_{S2\%}$=25	P_{1r}=(-0.125,0.085) P_{2r}=(-0.125,-0.085) P_{3r}=-1,25
T_s	MP=0.1 $T_{S2\%}$=12	P_{1s}=(-0.250,0.314) P_{2s}=(-0.250,-0.314) P_{3s}=-0,023

Tabla 4 : Especificación completa de los polos de lazo cerrado.

A partir de la especificación de los polos de lazo cerrado es posible encontrar las constantes del controlador (K_1, K_2 y K_3).
Dado que los parámetros son inciertos y con el objetivo de obtener un controlar que funcione adecuadamente en todo el espacio de parámetros se ha aplicado el algoritmo siguiente :

1- Discretizar el rango de las tres constantes (B,C,E) en 4 valores, generando así 64 posibles sistemas diferentes.
2- Para cada una de las 64 plantas se obtienen los parámetros del controlador (K_1, K_2 y K_3).
3- Se analizan las prestaciones de cada unos de los controladores en las 64 plantas y se queda con aquel controlador con mejores prestaciones en todos los puntos de operación.

Siguiendo estos pasos, se llega a los siguientes valores del vector K=(K_1, K_2, K_3) para cada una de las dos temperaturas que se desea controlar:

- $K_{serpentín}$ = (-75.39, -241.72, 96.25)
- $K_{reactor}$ = (-48202, -1068, 5044)

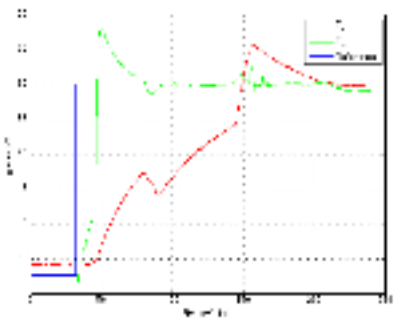


Figura 12 : Evolución del sistema de control de la temperatura del serpentín.

La Figura 12 muestra la evolución de la temperatura del serpentín y el reactor. Hasta el minuto 80 trabaja el lazo de control de la temperatura del serpentín y a partir de ese instante actúa el lazo de control del reactor. Como se puede observar en ambos casos el sistema evoluciona de acuerdo con las especificaciones.

5 CONCLUSIONES

En este trabajo se ha descrito los procesos de modelado y diseño de los controladores de los lazos de control de vacío y temperatura de un reactor agitado. Los diseños analíticos han sido validados experimentalmente en un sistema real.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el MEC mediante el proyecto DPI2010-15110.

Referencias

[1] GARCÍA GUTIÉRREZ, L., (1999) Válvulas de Control. AENOR.
[2] LUYBEN, W.L., (2007) Chemical Reactor Design and Control, pp. 197-201 y 201-205
[3] MIDDLETON, R. H., (1996) The Control Handbook, pp. 337-380.

LA ADMINISTRACIÓN DE RECURSOS APLICADA AL MANTENIMIENTO DE DISPOSITIVOS DE CAMPO

Ramón Ferreiro García
Dpto. de Ingeniería Industrial, Universidad de La Coruña
ferreiro@udc.es

José Luis Calvo-Rolle
Dpto. de Ingeniería Industrial, Universidad de La Coruña
jlcalvo@udc.es

José Luis Casteleiro-Roca
Dpto. de Ingeniería Industrial, Universidad de La Coruña
jose.luis.casteleiro@udc.es

Manuel Romero Gómez
Dpto. de Energía y Propulsión, Universidad de La Coruña
m.romero.gomez@udc.es

Resumen

La gran demanda de información originada por la necesidad de incrementar la eficiencia competitiva, da lugar a la masiva aplicación de medios de supervisión de la instrumentación de campo, lo que hace difícil y tediosa las tareas de mantenimiento de dispositivos integrados a los equipos y procesos controlados automáticamente en todos los sectores de la industria asociada con la producción industrial. Para hacer viable el mantenimiento de tan vasta cantidad y variedad de instrumentas con el objetivo de minimizar las paradas no programadas y alargar los ciclos de vida útil tanto de los equipos como de su instrumentación, se hace imprescindible la integración de herramientas capaces de administrar tales recursos. Este artículo revisa una de estas herramientas destinada al mantenimiento de dispositivos de campo denominada ASM (Asset Management Services).

Palabras Clave: Detección de fallos, Aislamiento de fallos, Administración de recursos, Asset management services, Sistemas basados en reglas.

1 INTRODUCCIÓN

Entre las tareas de mantenimiento y administración de recursos ASM en relación con Fieldbus, conciernen al estado físico de la instrumentación y del hardware asociado de la red, sin involucrar las estrategias de control, los trabajos llevados a cabo en [3-7]. La administración de recursos tiene como objetivo fundamental la explotación eficiente de la red y sus dispositivos. Independientemente del esquema de mantenimiento aplicado a la planta y sus procesos, se puede utilizar la tecnología Fieldbus [1, 2] para obtener información acerca de los dispositivos de campo, los cuales simplificarán el mantenimiento haciéndolo mas eficiente. La administración de recursos sobre Fieldbus conduce a la programación de un mantenimiento proactivo.

El mantenimiento reactivo consiste en un esquema de organización en el cual se actúa sobre algún dispositivo después de observar su inoperancia. Se da la circunstancia de que uno o más dispositivos pueden operar exhibiendo deficiencias en sus medidas sin observar su inoperancia. Esta clase de mantenimiento es poco seguro, peligroso y poco eficiente, a partir de que una planta no puede operar eficientemente si algún dispositivo no lo hace.

El mantenimiento preventivo consiste en una técnica de revisión sistemática a intervalos determinados de tiempo aún sin que sea necesaria la revisión de algún instrumento. Con este método de mantenimiento se logra seguridad y disponibilidad partiendo de que se evitan las paradas no esperadas.

El mantenimiento predictivo utiliza una técnica de revisión sistemática a intervalos determinados de tiempo aún sin que sea necesaria la revisión de algún instrumento (similar al mantenimiento preventivo). Sin embargo, el intervalo de revisión está optimizado en base al análisis estadístico de resultados históricos en base a la fiabilidad. Existe pérdida de recursos pero representa ciertas mejoras respecto al mantenimiento predictivo.

El mantenimiento proactivo es un esquema de organización en el cual la revisión de dispositivos es realizada para cada dispositivo en el instante preciso, en base a la condición de operación de cada instrumento o dispositivo. No se asignan recursos que supongan costes a dispositivos que no necesitan mantenimiento. Este método requiere un mínimo de recursos, tiene el coste posible más bajo y demanda la mínima mano de obra.

1.1 SISTEMA DE ADMINISTRACIÓN DE RECURSOS

La tarea de administración de recursos sobre tecnologías Fieldbus, conocida como "*asset management*" consiste en el manejo eficiente de aplicaciones de software destinadas al mantenimiento y calibración de dispositivos conformes con estándares sobre buses digitales entre los que destacan FOUNDATION^{TM} Fieldbus y HART.

La administración de recursos concierne exclusivamente a los dispositivos. Por esta razón, los bloques de recursos y transductor de cada dispositivo tienen que ser accedidos mediante la correspondiente aplicación software. La figura 1 muestra el esquema que ilustra un entorno propio de la arquitectura típica exhibiendo el principio de utilización del software de mantenimiento AMS (*asset management services*). Esta técnica de mantenimiento de dispositivos de campo opera simultáneamente (en paralelo) con el SCADA, y el software que la soporta es susceptible de ser alojado en el mismo computador o Host que hace las funciones de interfaz del usuario (HMI).

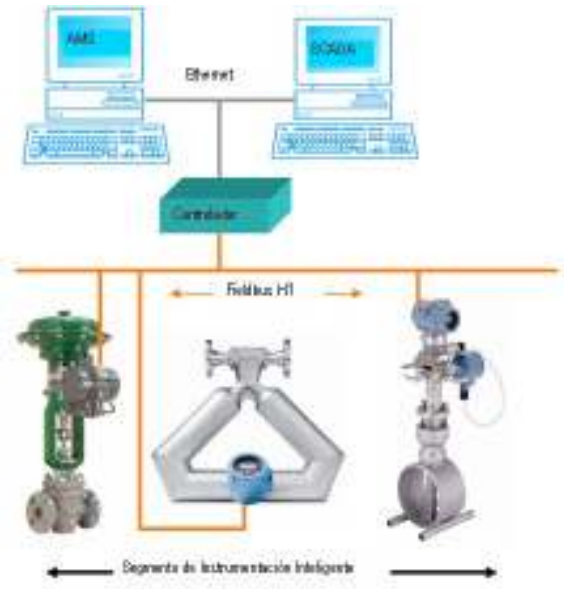


Figura 1 – Esquema de principio de operación del software de administración de recursos

2 RECURSOS UTILIZADOS EN TAREAS DE MANTENIMIENTO DE DISPOSIITIVOS DE CAMPO

Tradicionalmente, una parte significativa del potencial humano destinado al mantenimiento de planta estaba dedicado a la instrumentación de medida, control y actuación, siendo imprescindible cierta especialización en tecnologías de la producción, mecánica, eléctrica, electrónica e informática industrial. En la actualidad, esta tarea es llevada a cabo de manera sistemática por el personal técnico de vigilancia de planta, gracias a las herramientas de mantenimiento de los dispositivos de campo, las cuales realizan una serie de tareas de modo automático o con la ayuda y supervisión del operador de planta. Estas tareas van desde la contabilización de horas de servicio hasta la calibración y certificación con firma electrónica.

2.1 CALIBRACIÓN DE DISPOSITIVOS FIELDBUS

Una primera etapa de calibración es típicamente llevada a cabo en fábrica. Tareas simples de calibración que no demandan el contraste con instrumentos patrón pueden ser ejecutadas en campo. Las tareas de calibración y diagnóstico demandan comunicación entre los dispositivos y el Host posiblemente por medio de los correspondientes dispositivos de enlace, porque no pueden ser realizadas desde un dispositivo en estado de master o administrador activo de enlace (LAS) de modo autónomo en el estado actual de la tecnología basada en el estándar FF (FOUNDATION^{TM} Fielbus).

2.2 CALIBRACIÓN FRENTE A AJUSTE DE RANGO

Las tecnologías Fieldbus no cambian el concepto de calibración. Por ello no debe confundirse los conceptos de calibración y ajuste de rango. En el caso de transmisores analógicos, calibración y ajuste de rango ha sido tradicionalmente llevado a cabo utilizando el mismo juego de potenciómetros. En la instrumentación digital, el concepto de ajuste de rango consiste en el escalamiento de la medida dentro de su rango de trabajo. Por ejemplo, para cancelar la presión húmeda de transductores de nivel o caudal no se utiliza la calibración del bloque transductor, sino que esta operación es realizada con la escala del transductor en el bloque función AI o entrada analógica. El ajuste de rango puede ser ejecutado en modo remoto, pero no ocurre así con la calibración porque por definición esta tarea demanda una conexión procedente de una entrada de referencia estándar. La calibración tienen que ser realizada en el

propio dispositivo con objeto de aplicar diferentes valores conocidos de entrada y comparar la respuesta uno a uno.

La calibración consiste en la corrección de las lecturas del sensor y salida física de manera que satisfagan un estándar. Esto significa que tiene que ser aplicada una señal física de entrada externa estándar y que la salida del transductor tiene que ser medida para su calibración. En resumen, la calibración no puede ser ejecutada en modo remoto porque se necesita una señal externa estándar.

El ajuste de rango es solamente ejecutado para aplicaciones de medida inferida. El rango puede ser ajustado remotamente sin aplicar señal alguna de entrada o midiendo la salida. Se utiliza el ajuste de rango cuando el porcentaje de lectura es incorrecto. El ajuste de rango no afecta al valor primario de lectura.

2.3 EJECUCIÓN DE TAREAS DE CALIBRACIÓN DE DISPOSITIVOS

La calibración es esencialmente una función del dispositivo y es llevada a cabo desde el bloque transductor correspondiente, mientras que el ajuste de rango es una función de la estrategia de control y es ejecutada en el bloque función.

Calibración de entrada

A veces el valor primario de entrada difiere del valor de lectura de la misma variable de entrada proporcionado por el instrumento. Existen diversas razones para que se produzca este desvío, algunas de las cuales se citan como:

- La posición incorrecta en la instalación del transductor.
- Desvío de lectura por sobre-presión.
- Desvío de lectura por sobre-temperatura.
- Deriva.

La calibración es ejecutada para hacer coincidir la lectura con el valor de la señal de entrada aplicada. Aplicando un valor de entrada de un estándar conocido, e informando al transductor cual es el valor de esta señal, se está enseñando al transductor a medir el valor correcto.
La calibración de un transmisor conforme con FOUNDATION™ Fieldbus es llevada a cabo en el bloque transductor asignando los valores bajo y alto de la señal de entrada a los parámetros *CAL_POINT_LO* y *CAL_POINT_HI*. Asimismo el "span" mínimo entre los puntos de calibración superior e inferior puede ser revisado mediante el parámetro *CAL_MIN_SPAN*.

Calibración de salida

La calibración de salida se refiere a transductores que utilizan una señal destinada a la actuación, por ejemplo 20-100 kPa (3-15 psi) o 4-20 mA. Para calibrar una señal de salida se fuerza el bloque transductor a un valor de salida de nivel nominal. Si coincide la señal generada para el transductor con la señal física de salida del mismo, no es necesaria calibración del transductor. En otro caso, el valor actual físico de salida es medido externamente, siendo realimentado sobre el dispositivo transductor por medio del software de calibración corrigiendo la salida en modo adecuado.

Un caso típico de calibración de salida lo constituyen los posicionadores de válvulas de control. Este proceso de calibración dura en torno a un minuto y es realizado en modo automático con el software adecuado. El proceso de calibración de la salida debe ser efectuado fuera de línea debido a que durante este proceso, la válvula o elemento final de control queda fuera de control y el lazo de control quedaría en control manual.

Parámetro	**Información**
PRIMARY_VALUE	
STATUS	Good_NonCascade, NonSpecific, NotLimited
VALUE	23.0922
PRIMARY_VALUE_ RANGE	
CAL_POINT_HI	5068.28
CAL_POINT_LO	0
CAL_MIN_SPAN	127
CAL_UNIT	MmH2O (68ºF)
SENSOR_TYPE	Capacitance
SENSOR_RANGE	
EU_100	5080
EU_0	-5080
UNITS_INDEX	mmH2O (68ºF)
DECIMAL	2
SENSOR_SN	156706
SENSOR_CAL_ METHOD	Factory standard calibration
SENSOR_CAL_LOC	
SENSOR_CAL_ DATE	2001-11-21 09:39:57
SENSOR_CAL_ WHO	

Tabla 1 – Información relacionada con la calibración de un sensor de caudal

Archivos de información de calibración

La información de calibración es archivada por la herramienta software de calibración. La tabla 1 (por ejemplo) muestra la información relacionada con la calibración de un sensor de caudal.

3 DIAGNOSIS DE DISPOSITIVOS FOUNDATION FIELDBUS

Los dispositivos del estándar FOUNDATION™ Fieldbus disponen de varios parámetros dedicados al diagnóstico que simplifican la detección y localización de averías. Utilizando la capacidad de diagnóstico para detectar los fallos de dispositivos rápidamente, se pueden tomar acciones correctoras con prontitud. En otras palabras, la detección prematura de fallos habilita a los operadores de planta para minimizar tiempos de parada. Es una buena idea la de configurar la visualización del proceso para ver el status de parámetros y notificar al operador los fallos detectados en dispositivos.

Los dispositivos basados en tecnologías Fieldbus disponen de una amplia cobertura en diagnosis, utilizando sensores adicionales al principal. Disponen de algoritmos firmware que pueden detectar fallos en base a la experiencia sobre modos de fallo que están restringido al conocimiento de los fabricantes. Un transmisor de estas características detecta los fallos de los sensores externos tales como termopares o termo-resistencias así como defectos de cableado. Las condiciones ambientales tales como el exceso de temperatura que sobrepasa los límites de operación, constituyen una razón para reportar una causa de predisposición al fallo.

3.1 REDUCCIÓN DE MANTENIMIENTO INNECESARIO

Una de las mayores ventajas del diagnóstico continuo de dispositivos consiste en conocer en todo momento que dispositivos funcionan correctamente, aparte de conocer los que funcionan dudosamente o simplemente no funcionan. En definitiva, el diagnóstico proporcionado por los dispositivos Fieldbus, ayuda a realizar una valoración para confirmar que no es necesaria una sesión de detección y localización o búsqueda de averías. Esta técnica contribuye a reducir la frecuencia de revisión de los dispositivos.

El parámetro *BLOCK_ERR* es ideal para obtener un diagnóstico del dispositivo por medio del software de detección de error, de modo que al localizar el tipo de fallo se obtiene suficiente información para adoptar la solución idónea sin pérdidas de tiempo. La tabla 2 muestra el resultado emitido por el software de diagnóstico sobre un dispositivo de captación de temperatura.

Parámetro	**Valor**
MODE_BLK	
BLOCK_ERR	Input Failure
PRIMARY_VALUE_TYPE	Process Temperature
PRIMARY_VALUE	
STATUS	Bad-SensorFailure. NotLimited
VALUE	-177.56
PRIMARY_VALUE_ RANGE	
SENSOR_TYPY	Pt 100 IEC
SENSOR_CONECTION	Three wires
SECONDARY_VALUE	
SECONDARY_VALUE_ UNIT	ºC

Tabla 2 – Resultado del diagnóstico sobre un dispositivo de captación de temperatura.

La elaboración continua de estadísticas de resultados de operación habilita el método de diagnóstico para constituir el esquema de mantenimiento proactivo. La predicción de fallos admite sospechas pero la prevención es posible en base a la estadística que permite determinar el grado de deterioro de un dispositivo. La estadística operacional está salvada en los bloques transductores.

Los servo-posicionadores de válvulas de control recopilan una gran cantidad de información relativa a la estadística de operación como son la totalización del recorrido, que es susceptible de ser asociado a cierto desgaste, o el número de retornos de movimiento. Esta información sirve para predecir con cierta aproximación el tiempo de correcta operación comparando tales estadísticas con las expectativas de vida útil de partes críticas del dispositivo proporcionadas por el fabricante. La tabla 3 muestra la estadística operacional de un servo-posicionador de válvula.

Por ejemplo, si un fabricante recomienda que la empaquetadura del vástago de una válvula debe ser sustituida cada cierto número de retornos de movimiento, este número debe ser introducido en el posicionador como un límite de alarma, de manera que cuando tal cantidad de retornos haya sido alcanzado se ponga de manifiesto por medio de la correspondiente alarma.

Otro beneficio derivado del uso de la estadística operacional reside en el hecho de detectar la calidad

dinámica (pobre o aceptable) de lazos de control. Por ejemplo, un aumento brusco del número de retornos de válvula durante un determinado período de tiempo constituye una indicación clara de la oscilación del lazo de control. Tal oscilación puede ser debida entre otros casos al desajuste del regulador, no precisamente porque hayan cambiado los parámetros del regulador, que no lo han hecho, sino porque han cambiado las condiciones operativas de la planta para ese lazo de control, ya sea por variación de parámetros de la misma o por variación de las condiciones de operación a petición del operador (por ejemplo, otro punto de consigna). Este caso supone la necesidad de revisión sistemática de la estadística a períodos cortos de tiempo.

Parámetro	**Valor**
ORDERING_CODE	FY302-11-053
TRAVEL_ENABLE	Yes
TRAVEL_DEADBAND	2
TRAVEL_LIMIT	100000
TRAVEL	1938.2
REVERSAL_ENABLE	Yes
REVERSAL_DEADBAND	2
REVERSAL_LIMIT	10000
REVERSAL	1
DEVIATION_ENABLE	Yes
DEVIATION_DEADBAND	2
DEVIATION_TIME	60
STROKES	1938.2
TIME_CLOSING	60
TIME_OPENING	60
HIGHEST_TEMPERATURA	45
LOWEST_TEMPERATURE	16
DIAGNOSES_STATUS	OK

Tabla 3 – Estadística operacional de un servo-posicionador de la válvula FY302-11-053

3.2 INTEROPERABILIDAD FRENTE A INTERCAMBIABILIDAD

Cuando se sustituye un dispositivo que falla porque está averiado, por otro modelo idéntico en buen estado, simplemente se carga su configuración. Sin embargo, si el dispositivo averiado es sustituido por otro de diferentes características (tipo, versión) debe tenerse en cuanta la portabilidad de la configuración. Fundamentalmente, cada clase de dispositivo dispone de una extensión específica a los parámetros estándar en el bloque transductor. Si un dispositivo existente es sustituido por un nuevo tipo, debe crearse un nuevo bloque transductor en la configuración. El bloque de recursos (físico) es soportado por todos los dispositivos. Muchos dispositivos disponen de un complemento de bloques función que soportan parámetros diferentes a los parámetros estándar. Si fuese necesario sustituir un dispositivo que soporta bloques función propios de un fabricante por otro dispositivo que no puede soportar tales bloques función (pese a que son intercambiables en base al estándar excepto que son incompatibles en las mejoras adicionales de cada fabricante), se asignarían tales bloques función a otro dispositivo que soporte tales capacidades del mismo fabricante. La implementación de estrategias de control con bloques función del estándar FOUNDATION^{TM} Fieldbus, y no a una extensión de los mismos, asegura la interoperabilidad entre todos los bloques y consiguientemente la intercambiabilidad.

3.3 DETECCIÓN Y LOCALIZACIÓN DE FALLOS (TROUBLESHOOTING)

Existe una larga lista de parámetros pertenecientes a los bloques función, bloque de recursos y bloque transductor, destinados al propósito específico de la detección y localización de fallos. Todos los bloques disponen de un parámetro "mode" (*MODE_BLK*) y varios parámetros con "status" o "condition". Si el modo actual es diferente del modo objetivo, significa que algo va mal. Por consiguiente, el parámetro "mode" es un buen punto de partida para afrontar tareas de localización de averías.

Bloque de recursos ("resource block")

El parámetro "estado de recursos" (*RS_STATE*) refleja el estado de la estrategia de control. En modo "Auto", el estado de recursos está activo ("Online"). En modo fuera de servicio ("OOS"), el estado de recursos está en "Standby". Si el estado de recursos es "Failure", significa que ha fallado la memoria del dispositivo o el hardware.

Bloque transductor ("transducer block")

El parámetro de error del bloque transductor (*XD_ERROR*) se halla en todos los bloques transductores. En general reporta errores relacionados con dispositivo en su conjunto. Estos errores son propios y únicos de cada fabricante del dispositivo. Este parámetro reporta un solo fallo de cada vez, de modo que si existen varios fallos simultáneos, el parámetro *XD_ERROR* reporta el de más alta prioridad o fallo más severo. La interpretación y valoración de la severidad de la avería es realizada con la información del manual de servicio del dispositivo particular.

3.4 COMUNICACIÓN

Los problemas de comunicación debidos a instalación de una red suelen aparecer durante la tarea de comisionado. Una vez hayan sido corregidos no ocurren de nuevo hasta que se realicen cambios en la red. Por ejemplo, añadiendo o sustituyendo un dispositivo, puede introducir un punto de conexión involuntaria a tierra. La expansión o modificación puede introducir problemas de instalación por aumento de carga en la red y consecuente carencia de tensión de alimentación. Son casos comunes por ejemplo, la falta de estanqueidad en las conexiones, permitiendo la entrada de agua en dispositivos o cajas de conexión.

Si existen varios problemas es necesario hallar un patrón para tratar de determinar una causa común. Por ejemplo, los casos en los que el problema afecta a un determinado tipo de dispositivos, o los dispositivos de un segmento de red. En el primer caso, podría deberse a la configuración errónea de los dispositivos y en el segundo caso es sospechosa la instalación del segmento afectado. Partiendo de que un sistema opera correctamente, lo más importante es saber qué modificaciones se han efectuado o simplemente qué se ha hecho antes de aparecer el problema. Si falla la comunicación de un dispositivo en particular, se debe conectar en otro segmento de red y probar la comunicación. Para ello es necesario disponer de repuestos en cable y conexiones adecuadas.

Cuando es detectado un fallo de comunicación, se verifica mediante el uso de la herramienta de diagnóstico para determinar si el fallo afecta a la red completa, a un segmento (utilizando repetidores o barreras de seguridad) o solamente a un dispositivo.

Cualquier dispositivo que falla por comunicación, es dado de baja de la lista viva de dispositivos.

Existe complementariamente la técnica convencional de verificación mediante un simple polímetro. Se puede detectar corto-circuito midiendo entre cable y cable, conductor-pantalla, conductor-tierra y pantalla-tierra. También se puede verificar la capacitancia del cable ya que pese a no ser común, puede haber problemas en los terminadores de red.

3.5 ALIMENTACIÓN

Si la tensión de alimentación en los terminales de un dispositivo es inferior a 9 V, el dispositivo puede operar deficientemente. Algunos dispositivos necesitan valores de alimentación superiores a los 9 V. Bajo valor del voltaje de alimentación es una de las primeras causas de fallo en instalaciones de seguridad intrínseca e incluso en instalaciones no seguras con gran longitud de cables excesivamente delgados. Otra causa potencial de fallo la constituyen los dispositivos de alto consumo de corriente. En las cajas de conexión pueden existir conexiones pobres de contactos, lo que supone otra fuente de fallo. Es necesario asegurarse del estado de los siguientes conceptos:

- Voltaje en los terminales de cada dispositivo mayor de 9 V.
- Suficiente capacidad de la fuente de alimentación, esto es, satisface la demanda de consumo de corriente de los dispositivos de la red en el peor de los casos.
- Verificación de la caída de tensión entre la entrada y salida de las cajas de conexión.

3.6 PERTURBACIONES INHERENTES AL ACOPLAMIENTO MAGNÉTICO

En entornos de intenso campo magnético, la red Fieldbus conjuntamente con su cable de tierra puede formar una bobina en la cual las corrientes alternas son inducidas por el campo magnético. Esta bobina se forma porque los instrumentos de campo están dotados generalmente de filtros que causan cierto acoplamiento capacitivo entre los terminales de la red y la carcasa del dispositivo. La carcasa está conectada a tierra mediante un cable de tierra formando un lazo de corriente. Cuanto mayor es el lazo, mas se favorece la entrada de ruidos al campo magnético creado. En una red bien balanceada, el voltaje entre la tierra del instrumento y la pantalla debe de ser menor de 0.2 V_{rms}. No obstante, los errores de comunicación podrían aparecer si el voltaje se aproximase (o estuviese en torno) a 3 V_{rms}.

Reduciendo el área afectada por la bobina, se reduce la entrada de ruidos y los problemas subsecuentes de comunicación. Se puede reducir la citada área mediante confinamiento del cable dentro del conductor metálico de tierra, el cual mantiene la continuidad eléctrica para todas las secciones del conductor. Como mejora adicional se puede conectar un cable de retorno, preferiblemente de barra, la cual se halla generalmente dentro del conductor.

3.7 INTERFERENCIAS EN EL LAZO A TIERRA

Los capacitores de las conexiones de red de los instrumentos de campo actúan como acoplamientos a tierra en el dispositivo. Si existe una diferencia de potencial de tierra entre el Host y el dispositivo de campo, se forma un lazo cerrado a través de tierra que puede perturbar la comunicación. Un lazo de tierra puede formarse también cuando la pantalla o cables están accidentalmente conectados a tierra en múltiples puntos de la red.

Cuando los dispositivos de campo están correctamente conectados, el potencial de tierra se reduce, disminuyendo la probabilidad de fallos de la comunicación. Para detectar la existencia de múltiples conexiones a tierra se desconecta el Host de tierra (que debería ser la única conexión de tierra en toda la red), y se mide la impedancia entre los conductores y tierra y entre la pantalla y tierra. Si no hay múltiples conexiones a tierra el valor de la impedancia medida debería ser del orden de varios mega Ohms (es decir, un buen aislamiento).

3.8 DIRECCIONAMIENTO

El Host FOUNDATION™ Fieldbus detecta cada dispositivo nuevo conectado a la red, interrogando constantemente las diferentes direcciones de los dispositivos de la red. Los dispositivos detectados por identificación de su "tag" son presentados en la lista viva de dispositivos de la herramienta de configuración.

Estrategia de control

Dos fuentes importantes de información en la tarea de localización o búsqueda de fallos son:

- El parámetro "mode" del bloque.
- El parámetro "status" del bloque.

Un fallo común consiste en que no sean aceptados los cambios de parámetros. Ello puede ser debido a que el bloque está en un modo que prevenga la escritura de los parámetros. Algunos parámetros para ser cambiados requieren que el bloque se halle en modo "Man" u "OOS". Si el valor del parámetro está fuera de escala tampoco deja efectuar cambios.

Si la salida de un bloque permanece inmóvil, puede ser debido a un modo inadecuado, a haber alcanzado un límite o debido a que un parámetro como GAIN ha sido configurado con valor nulo.

Utilización del parámetro "Status"

Observando el status de entradas, salidas y parámetros contenidos en el bloque función, se puede determinar si un problema procede de un bloque particular. El "status" de entrada es uno de los factores que controlan el modo del bloque.

Finalmente, las técnicas convencionales de detección de fallos en base a la redundancia hardware es obviamente posible y recomendable. El único inconveniente lo constituye el incremento en el coste de equipamiento, de modo que esta dotación se aplica principalmente en plantas de proceso crítico, en donde la seguridad de funcionamiento es el criterio de mayor peso.

4 CONCLUSIONES

Se ha revisado una de las herramientas de mantenimiento de dispositivos de campo aplicada en plantas y procesos del sector productivo industrial de forma masiva. La razón del éxito de tales herramientas radica en sus ventajas frente al mantenimiento tradicional: incrementa el determinismo por reducción de ambigüedad en la localización de fallos de la instrumentación, lo que redunda en la reducción de paradas de emergencia o paradas no programadas, con lo que repercute significativamente en la eficiencia productiva.

Referencias

[1] ANSI/ISA-51.1-1979 (R1993), Process Instrumentation Technology, Research Triangle Park, NC:ISA, ISBN 0-87664-390-4, 1993

[2] Jonas Berge, "Fieldbus for process Control: Engineering Operation and Maintenance", ISA-The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, USA, ISA-2002, pp 359-377

[3] Model 3051S FOUNDATION™ Fieldbus Supplement, 00809-0200-4801, Rev AA, April 2002. Reference manual, Model 3051S Series, Pressure Transmitter Family, with FOUNDATION™ Fieldbus protocol, Emerson Process Management. www.rosemount.com

[4] Model 3244MV MultiVariable Temperature Transmitter with FOUNDATION™ Fieldbus, 00809-0100-4769, Rev. BA, 07/00, Emerson Process Management. www.rosemount.com

[5] Model 8742C. Magnetic Flowmeter Transmitter with FOUNDATION™ Fieldbus, 00809-0100-4793, Rev. AA, 8/99. Emerson Process Management. www.rosemount.com

[6] Model 8800C and Model 8800A Vortex Flowmeter with FOUNDATION™ Fieldbus, 00809-0100-4772, Rev. DA, 1/00. Emerson Process Management. www.rosemount.com

[7] Operation & Maintenance Instruction manual. Fieldbus Valve Positioner. Feb/03. FY302. www.smar.com

ESTIMACIÓN GARANTISTA DE LA POSICIÓN DE UN QUADROTOR CON GPS

Ramón A. García, Manuel G. Ortega y Francisco R. Rubio
Depto. de Ingeniería de Sistemas y Automática, Universidad de Sevilla,{ramongr, mortega, rubio}@us.es

Guilherme V. Raffo
Depto. de Engenharia Eletrônica, Universidade Federal de Minas Gerais, raffo@ufmg.br

Resumen

La aplicación de dos métodos de estimación garantistas sobre un quadrotor se presentan en este artículo. Se tiene un modelo del sistema discretizado en el que se calcula su posición cada T segundos, mientras que la medida del GPS es cada t_{sincro} segundos. Se aplicaron los algoritmos garantistas para contemplar las posibles posiciones en las que se puede encontrar el quadrotor y posteriormente, gracias a la medida del sensor corregir y mejorar la estimación realizada.

Palabras clave: Estimación garantista, aritmética intervalar, zonotopos, quadrotor, GPS.

1. Introducción

Un quadrotor es un vehículo aéreo con cuatro motores coplanarios. En la Fig.1 se observa un esquema con las distintas fuerzas y momentos que actúan sobre dicho tipo de vehículo.

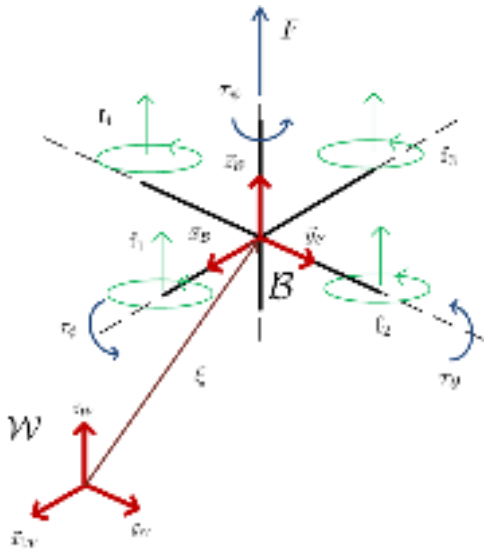


Figura 1: Esquema de las Fuerzas y Pares sobre un Quadrotor.

Se trata pues de un sistema continuo cuya medición de la posición a través de un GPS se realiza de forma discreta. Se tienen dos tiempos de muestreo, uno debido a la implementación del observador discreto, T, y otro con el que se realizan las mediciones del GPS, t_{sincro}, donde $T < t_{sincro}$. En la Fig.2 se puede apreciar el comportamiento que se plantea: en el instante de partida t_1 se tiene tanto la estimación de la posición a través de las ecuaciones del modelo como la medida de la señal GPS, y hasta que no han transcurridos t_{sincro} segundos no se volverá a tener la sincronización con el GPS. Mientras que en los instantes de tiempo intermedios, cada T segundos, se tendrá la estimación basada en las ecuaciones del modelo del sistema. El uso de dos tiempos de muestreo ya ha sido empleado en [11], donde se trabaja con un tiempo para el modelo y otro para la medida del sensor, aunque con fines distintos.

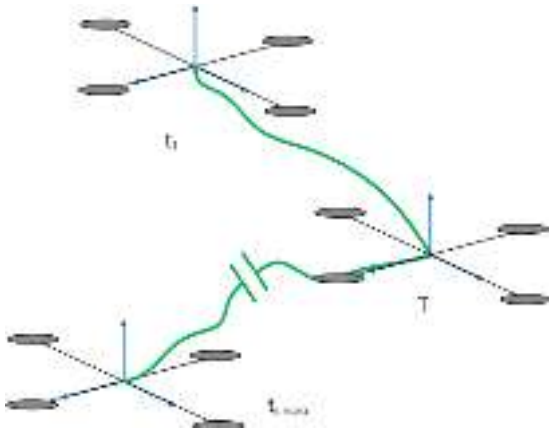


Figura 2: Ejemplo instantes de muestreo utilizados.

Por otro lado, el modelo presenta distintas incertidumbres que pueden provenir de diversas fuentes, tales como error al medir la masa o la distribución de la misma, dinámica del sistema no contemplada, imposibilidad de medir perturbaciones como viento, efecto suelo, etc. También existe la posibilidad de tener incertidumbres a la hora de realizar la medida por parte de los sensores, como muestra la Fig.3. Si se considera una medida totalmente precisa nos encontraremos en el caso (a). Si por el contrario, se considera que la medida no sea perfecta y se asume que puede tener cierta inexactitud, se tiene el caso (b). Más en detalle, en la Fig.3 se representa un proceso de estimación, donde en

un momento o instante determinado se realiza una medición con el GPS o sincronización. Considerando un sistema lineal con una fuerza externa también lineal, se tiene debido a dicho proceso para cada coordenada en el caso (a) un triángulo, ya que al sincronizar nos fiamos completamente de la medida y por tanto al "resetear" tenemos un solo punto, mientras que para el caso (b) lo que se obtiene es un trapecio, ya que al "resetear" no se tiene un punto sino un conjunto de puntos posibles en los que se puede encontrar el estado.

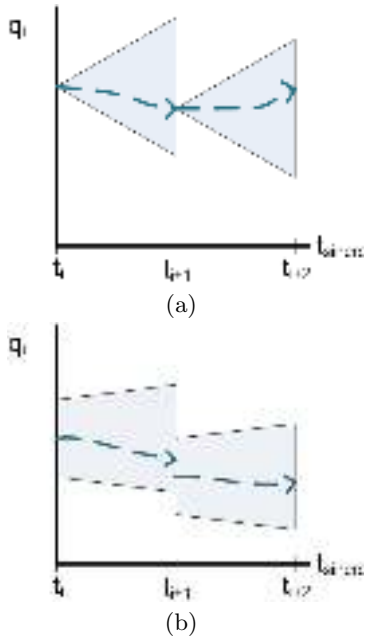


Figura 3: Ejemplos de medida GPS.

A la hora de estimar las variables de estado de un sistema, existen diferentes métodos posibles. Entre ellos, los observadores de estado, usados en diferentes aplicaciones, por ejemplo [1], [12], [8], [14] y [5]. Basándonos en la clasificación realizada en [6] podemos resaltar tres tipos:

1. El observador de Luenberger que no trata explícitamente las incertidumbres.
2. Estimadores basados en el comportamiento estadístico, como el filtro de Kalman que presupone el ruido con cierta función de probabilidad.
3. Estimadores basados en estados garantistas, los cuales no necesitan presuponer una función de probabilidad sino que se basan en que las variables estarán acotadas dentro de un determinado rango, es decir, son conjuntos compactos (en $\mathbb{R}^n$ que sean acotados y cerrados).

Como ejemplo de este último tipo, si se usaran intervalos como método de estimación garantista, sin la necesidad de suponer una función de probabilidad como se hace en el caso del filtro de Kalman, sólo hay que tener en cuenta que el ruido o la incertidumbre están acotados entre un valor máximo y uno mínimo. De este modo, no es necesario recurrir a un estudio estadístico de dicho ruido, lo que en ocasiones puede ser difícil o tedioso de calcular.

Continuando con [6], los estimadores garantistas dan como resultado un estado estimado que es un dominio en el espacio de estados. Dicho domino representa una cota exterior de todos los posibles estados que son consistentes tanto con el modelo incierto (con incertidumbres y/o ruido) como con las medidas inciertas (con incertidumbres y/o ruido).

Tal dominio se puede representar de diversas maneras, ya sea mediante elipsoides, cajas (provenientes del uso de aritmética intervalar), paralelotopos o incluso politopos de complejidad limitada, es decir, con un número de vértices y caras limitado. Según el método empleado para representar el dominio, se tendrá de mayor o menor manera el llamado efecto *wrapping*[1], es decir, será más o menos conservador en función de cómo se acoten las soluciones. No es lo mismo acotarlo por una caja de lados paralelos a los ejes, que por un paralepípedo o por un zonotopo. En la Fig.4 se puede observar el conjunto de estados posibles, la acotación externa mediante una caja cuyos lados son paralelos a los ejes (supuestos ejes en horizontal y vertical), y un paralelepípedo cuyos lados no son paralelos a los ejes y que se intenta adaptar mejor a la forma del conjunto de estados posibles. Se puede apreciar como la solución obtenida por medio de la caja es más conservadora que la obtenida por el paralepípedo.

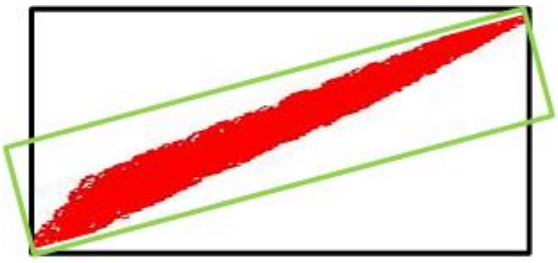


Figura 4: Ejemplo de ajuste del conjunto de posibles estados.

En este artículo se pretende diseñar un observador que estime el estado del sistema haciendo uso de las medidas obtenidas del mismo. El resto del

[1] El efecto *wrapping* es el hecho de añadir más soluciones de las reales al utilizar un método de acotación demasiado conservador y por tanto obtener un conjunto de posibles estados mayor que los posibles estados reales que podría alcanzar el sistema.

trabajo está organizado de la siguiente manera, en la sección 2 se presenta el modelo del sistema. Los dos algoritmos usados serán introducidos en la sección 3 y posteriormente se verán los resultados obtenidos mediante simulación a un caso de estudio en la sección 4. Finalmente, las conclusiones se muestran en la sección 5.

2. Modelado del Sistema

Partiendo del caso general, sea un sistema discreto con incertidumbres, tanto en el modelo como en la medida, dado por:

$$\begin{cases} x_{k+1} = f(x_k, w_k) \\ y_k = g(x_k, v_k) \end{cases} \quad (1)$$

donde $x_k \in \mathbb{R}^n$ es el vector de estado del sistema, $y_k \in \mathbb{R}^p$ es el vector de salida, $w_k \in \mathbb{R}^{n_w}$ representa errores de modelado así como perturbaciones y finalmente $v_k \in \mathbb{R}^{p_v}$ representa el vector de ruidos en las mediciones.

Para los métodos que se van a emplear en la resolución de la estimación, se supone que las incertidumbres están acotadas dentro de unos valores concretos, sin importar la distribución estadística. Por ello se asume que $w_k \in W$, $v_k \in V$ y $x_0 \in X_0$.

Particularizando (1) en el caso de que se desee estimar la posición espacial de un quadrotor, se parte de la segunda Ley de Newton, considerando que todas las fuerzas se agrupan en una sola componente. Siendo análogas las expresiones para los tres ejes, si se considera la expresión sólo para la componente x, se tendrá $\ddot{x} = \frac{F_x}{m}$. Trasladándolo al espacio de estados, se pueden definir dos estados, $x_1 = x$ y $x_2 = \dot{x}$

Se tiene pues un sistema descrito por la posición y la velocidad en $\mathbb{R}^6$:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \frac{F_x}{m} \\ \frac{F_y}{m} \\ \frac{F_z}{m} \end{bmatrix} \quad (2)$$

donde x, y, z son las coordenadas en posición, $\dot{x}, \dot{y}, \dot{z}$ son las velocidades lineales, m es la masa del vehículo y F_x, F_y, F_z son fuerzas aplicadas en cada uno de los ejes respectivamente.

Si se expresa (2) en tiempo discreto, el resultado sería:

$$\begin{bmatrix} x_{1k+1} \\ x_{2k+1} \\ x_{3k+1} \\ x_{4k+1} \\ x_{5k+1} \\ x_{6k+1} \end{bmatrix} = \begin{bmatrix} x_{1k} + Tx_{4k} \\ x_{2k} + Tx_{5k} \\ x_{3k} + Tx_{6k} \\ x_{4k} \\ x_{5k} \\ x_{6k} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2}\frac{F_x}{m} \\ \frac{T^2}{2}\frac{F_y}{m} \\ \frac{T^2}{2}\frac{F_z}{m} \\ T\frac{F_x}{m} \\ T\frac{F_y}{m} \\ T\frac{F_z}{m} \end{bmatrix} \quad (3)$$

donde T es el tiempo de muestreo empleado.

Se han mostrado de forma reducida las fuerzas aplicadas, si se desarrollan incluyendo el subsistema de rotación de un quadrotor, se pueden expresar las fuerzas aplicadas como:

$$\begin{aligned} F_x &= (\cos\phi\cos\psi\sin\theta + \sin\phi\sin\psi)u + A_x \\ F_y &= (\cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi)u + A_y \\ F_z &= (\cos\phi\cos\theta)u + A_z \end{aligned} \quad (4)$$

siendo ϕ, θ, ψ los ángulos de Euler (roll, pitch y yaw respectivamente), u el empuje total y A_i la perturbación en el eje i correspondiente. Sin embargo en (3) se ha empleado una notación simplificada en lugar de hacer referencia a los ángulos de Euler o al empuje, ya que eso es objeto de cálculo del controlador de estabilización el cual no se está abordando en esta parte, es decir, aquí se calculan F_x, F_y, F_z necesarias para el movimiento traslacional y a partir de dichas fuerzas se calcularía en el control rotacional los ángulos y empuje necesarios.

Una vez presentado el sistema, se verá cómo se ha abordado la resolución del problema. Dado el sistema se quiere estimar su posición a través de un modelo discreto con incertidumbres y una medición de la posición del mismo, que podrá ser a su vez incierta o no, según el caso de estudio.

3. Algoritmos utilizados

A continuación se mostrarán los dos casos objetos de estudio: aritmética intervalar y zonotopos. Se ha considerado el movimiento en el plano XY, reordenando los estados de (3) y sin usar las componentes correspondientes al eje z.

3.1. Aritmética Intervalar

La idea de utilizar la aritmética intervalar en estimación garantista surge de la partida del conocimiento de los límites de las incertidumbres, por ejemplo, se conoce que puede haber un error máximo y mínimo a la hora de calcular la masa, pero no se sabe exactamente cuanto es el valor del error.

Visto el motivo de su aplicación, se verá el sistema reordenado a partir de (3):

$$\begin{bmatrix} x_{1k+1} \\ x_{2k+1} \\ x_{3k+1} \\ x_{4k+1} \end{bmatrix} = \begin{bmatrix} x_{1k} + Tx_{3k} \\ x_{2k} + Tx_{4k} \\ x_{3k} \\ x_{4k} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2}\frac{F_x}{m} \\ \frac{T^2}{2}\frac{F_y}{m} \\ T\frac{F_x}{m} \\ T\frac{F_y}{m} \end{bmatrix} \quad (5)$$

donde x_1 es la posición x, x_2 es la posición y, x_3 es la velocidad $\dot{x}$ y x_4 es la velocidad $\dot{y}$. El resto de la notación es la utilizada en (3).

Para abordar el problema, se hace uso de una adaptación del algoritmo presentado en [9], donde se hace uso de la aritmética intervalar para estimar el estado de un sistema autónomo ($x_{k+1} = f(x_k)$). Esquemáticamente, el algoritmo es el siguiente:

- Paso 1: $[\hat{x}](t_2) = [x](t_1) + (t_2 - t_1)\Box f([x](t_1))$
- Paso 2: $[v] = [x](t_1) \cup [\hat{x}](t_2)$
- Paso 3: $[w] = inflado([v], \alpha\omega[v] + \beta)$
- Paso 4: $[x](t_2) = [x](t_1) + (t_2 - t_1)\Box f([w])$

donde $\omega[v]$ es la longitud de su lado mayor y la operación de inflado dado una caja $[x]$ y un escalar ϵ es: $inflado([x], \epsilon) = [\underline{x}_1 - \epsilon, \overline{x}_1 + \epsilon] \times ... \times [\underline{x}_n - \epsilon, \overline{x}_n + \epsilon]$

Como resultado negativo de usar un método tan conservador como el propuesto con la aritmética intervalar es el llamado efecto *wrapping*. En la Fig.4 se puede observar cómo la función f está acotada por la caja de color negro cuando en realidad hay zonas de dicha caja en las que no cae ningún punto.

La fuente de incertidumbre en el modelo se ha supuesto que sea un error a la hora de la medición de la masa, m. Debido a ello se tendrá un intervalo en el que se podrán encontrar los distintos estados.

La función de inclusión[2] $\Box f$ de (5) será (por simplicidad en la notación se han omitido los subíndices k):

$$\Box f([x]) = \begin{bmatrix} [\underline{x}_1, \overline{x}_1] + T[\underline{x}_3, \overline{x}_3] + \frac{T^2}{2}[\frac{1}{\overline{m}}, \frac{1}{\underline{m}}]F_x \\ [\underline{x}_2, \overline{x}_2] + T[\underline{x}_4, \overline{x}_4] + \frac{T^2}{2}[\frac{1}{\overline{m}}, \frac{1}{\underline{m}}]F_y \\ [\underline{x}_3, \overline{x}_3] + T[\frac{1}{\overline{m}}, \frac{1}{\underline{m}}]F_x \\ [\underline{x}_4, \overline{x}_4] + T[\frac{1}{\overline{m}}, \frac{1}{\underline{m}}]F_y \end{bmatrix} \quad (6)$$

[2]Según otros autores se puede expresar como $[f]$

siendo T el tiempo de muestreo. Se han elegido $\alpha = 0{,}1$ y $\beta = 0{,}0001$. A la hora de calcular $\omega[v]$ se han separado los estados en función a la coordenada que representen, es decir, x_1 y x_3 por un lado y x_2 y x_4 por otro. Una vez hecha la distinción, se aplica a cada pareja de forma separada aquel que sea el mayor de cada par.

3.2. Zonotopos

La idea es utilizar otro procedimiento que permita una estimación garantista cuyo resultado sea una acotación menos conservadora. Se vuelve a partir del conocimiento de los límites de la incertidumbre, pero esta vez no se suponen que tengan forma rectangular, o de caja, sino que se permite una forma más compleja, haciendo usos de politopos, concretamente de zonotopos.

En esta ocasión, el sistema reordenado a partir de (3), es el siguiente:

$$\begin{bmatrix} x_{1k+1} \\ x_{2k+1} \\ x_{3k+1} \\ x_{4k+1} \end{bmatrix} = \begin{bmatrix} x_{1k} + Tx_{2k} \\ x_{2k} \\ x_{3k} + Tx_{4k} \\ x_{4k} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2}\frac{F_x}{m} \\ T\frac{F_x}{m} \\ \frac{T^2}{2}\frac{F_y}{m} \\ T\frac{F_y}{m} \end{bmatrix} \quad (7)$$

donde x_1 es la posición x, x_2 es la velocidad $\dot{x}$, x_3 es la posición y y x_4 es la velocidad $\dot{y}$. El resto de notación es la ya utilizada en (3).

Según el algoritmo mostrado en [4]:

- Paso 1: Usar una función de inclusión para acotar la trayectoria del sistema no lineal con incertidumbres.
- Paso 2: Calcular una cota del conjunto de estados consistente.
- Paso 3: Calcular una cota ajustada del conjunto intersección.

Para el paso 1, se empleará el método de Kühn [10], con el cual se consigue un zonotopo que acota los posibles estados del sistema. Para el paso 2, a partir de las medidas obtenidas se estimará una franja del conjunto de variables de estados factibles para dicha medición. Y por último para el paso 3 habrá que calcular la intersección entre zonotopo y franja. Por tanto, lo que se verá en este momento es cómo calcular una franja de posibles estados a partir de una medición.

En [4] viene detallado cómo obtener un conjunto de estados a través de la medida tomada. Para ello se tendrán tantas franjas como componentes

de la salida; siendo la salida $y_k = g(x_k, v_k)$ donde $x_k \in \mathbb{R}^n$ es el estado del sistema y $v_k \in \mathbb{R}^{p_v}$ es el ruido en la medida.

Supongamos que tenemos una medida $y_k \in \mathbb{R}^p$, se tiene como conjunto de estados consistentes con la medida, denotado por X_{y_k}:

$$X_{y_k} = \{x \in \mathbb{R}^n : y_k \in g(x, V)\}$$

Se define $X_{y_k}(i)$ como el conjunto de estados consistentes con la componente i-ésima de y_k:

$$X_{y_k}(i) = \{x \in \mathbb{R}^n : y_k(i) \in g_i(x, V)\}$$

donde $g_i(x, V)$ denota la i-ésima componente de $g(x, V)$.

Con ambas definiciones tendremos que:

$$X_{y_k} \subseteq \bigcap_{i=1}^{p} X_{y_k}(i)$$

En [4] se demuestra por qué una franja sirve de cota externa para el intervalo, en este artículo nos quedaremos con el resultado, y el procedimiento para obtener a partir de una medida una franja. Se supone dado un zonotopo $\hat{X}_k$ y una medida y_k. Se deben obtener mediante aritmética intervalar el vector $c_i \in \mathbb{R}^n$ y los escalares $s_i, \sigma_i \in \mathbb{R}$ tal que:

- $c_i = mid\left(\square \nabla_x g_i(\hat{X}_k, V)\right)$
- $c_i^T \hat{X}_k - g_i(\hat{X}_k, V) \subseteq [s_i - \sigma_i, s_i + \sigma_i]$

entonces, si $X^e_{y_k}(i) = \{x : |c_i^T x - y_k(i) - s_i| \leq \sigma_i\}$ resulta que la intersección entre el zontopo $\hat{X}_k$ y el conjunto de estados consistentes i-ésimo pertenecerá a la intersección entre el zonotopo $\hat{X}_k$ y la franja í-ésima $X^e_{y_k}(i)$:

$$\hat{X}_k \bigcap X_{y_k}(i) \subseteq \hat{X}_k \bigcap X^e_{y_k}(i)$$

4. Simulaciones

Una vez vistos los algoritmos, se mostrarán los resultados obtenidos en simulación.

4.1. Aritmética Intervalar

Una de las ventajas de usar intervalos, es que la intersección de dos intervalos sigue siendo un intervalo. Este hecho, sin embargo, no ocurre con los zonotopos.

Realizando una simulación cuya fuerza aplicada consiste en un escalón de subida y otro de bajada para volver al punto inicial, sin incertidumbres en la medida, se obtiene como resultado el mostrado en la Fig.5. En la simulación se cuenta con dos tiempos distintos, un tiempo de sincronización de 1 segundo (recordar que este tiempo es el que hay entre lectura y lectura del GPS) y uno de muestreo de 20 milisegundos (recordar que este tiempo es el que transcurre entre cálculo y cálculo del vector de estados).

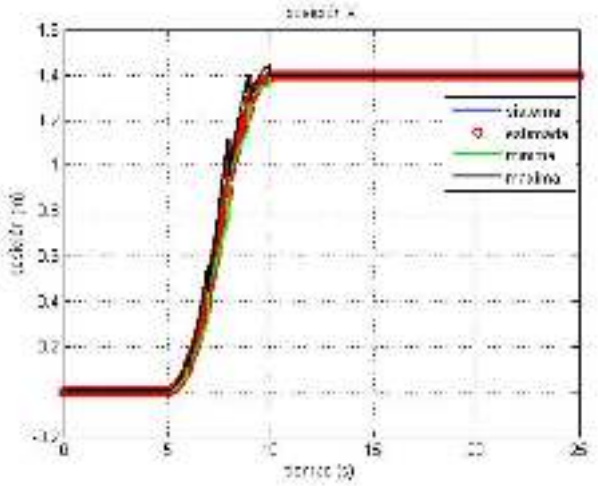


Figura 5: Coordenada X con GPS sin incertidumbres.

Si ahora, la medida del GPS se considerada no "fiable" y se supone un dispositivo diferencial con un error de $\pm 0{,}2m$ por ejemplo, se puede observar en la Fig.6 como varían los resultados. Al igual que en el caso "fiable" el tiempo de sincronización es de 1 segundo y el de muestreo de 20 milisegundos.

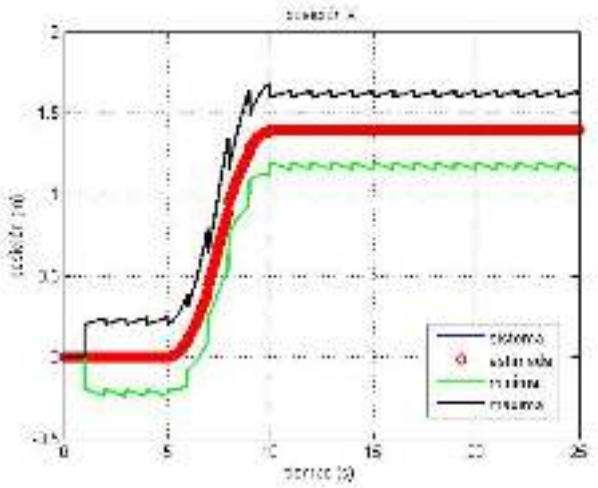


Figura 6: Coordenada X con GPS con incertidumbres.

4.2. Zonotopos

Al igual que con el caso de la aritmética intervalar se mostrarán los resultados sobre el eje x, la fuerza aplicada consiste en un escalón de subida y otro de bajada para volver al punto inicial y que así se llegue a un estado en régimen permanente.

Realizando la simulación bajo dos condiciones,

medida de la señal GPS considerada exacta, y considerada con una desviación de 0.2 metros se obtienen las Fig.7 y 8 respectivamente.

En la Fig.7 se puede apreciar como al principio el área de la región definida por el zonotopo crece hacía un zonotopo mayor, pero conforme se reciben las señales fiables del GPS se consigue reducir el tamaño del zonotopo que relaciona la posición, coordenada x_1, con la velocidad, coordenada x_2.

En la Fig. 8 se puede apreciar como en esta ocasión en la que las medidas del GPS no se consideran exactas, no se tiene una disminución drástica de los zonotopos.

5. Conclusiones

Se ha presentado la aplicación sobre un quadrotor de dos estimadores garantistas, uno basado en aritmética intervalar y otro en el uso de zonotopos.

La aplicación de algoritmos garantistas se llevó a cabo debido a que se tiene un modelo continuo con medidas del posición de forma discreta, en tiempos mayores a los de muestreo del sistema. Por tanto si solo se usara la información del sensor, se tendrá durante instantes de tiempos sin cerrar el lazo de control. Para tener el lazo de control cerrado se tiene el modelo al que se le añaden posibles incertidumbres y se contempla los posibles estados a los que puede llegar el sistema.

Además, permite percatarse de que no solo se necesita el camino que se desee seguir libre de obstáculos, sino que además se necesita una zona donde es posible que se encuentre el quadrotor.

Agradecimientos

Los autores quieren expresar su agradecimiento al MCeI por la financiación de este trabajo, a través de los proyectos DPI2010-19154 y DPI2012-37580-C02-02 así como a FAPEMIG y al *Programa Institucional de Auxílio à Pesquisa de Doutores RecémContratados* de la PRPq/UFMG.

Referencias

[1] Abbott, E. and Powell, D., Land-Vehicle Navigation Using GPS. Proceedings of the IEEE, Vol. 87, No.1, 1999.

[2] Alamo, T., Bravo, J.M. and Camacho,E.F. Guaranteed state estimation by zonotopes. Automatica Vol. 41, Issue 6, pages 1035-1043, 2005.

[3] Bravo, J.M., Alamo, T. y Camacho, E.F., Estimación Garantista de Estados en Sistemas Híbridos.

[4] Bravo, J.M., Control predictivo no lineal robusto basado en técnicas intervalares (Tesis Doctoral)

[5] Brunke, S. and Campbell, M., Estimation Architecture for Future Autonomous Vehicles. Proceedings of the American Control Conference, 2002.

[6] Combastel,C., A State Bounding Observer Based on Zonotopes. Proceedings of European Control Conference, 2003

[7] Combastel, C., A State Bounding Observer for Uncertain Non-linear Continuous-time Sstems based on Zonotopes. Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, 2005.

[8] Farrelly, J. and Wellstead, P., Estimation of Vehicle Lateral Velocity. Proceedings of the 1996 IEEE International Conference on Applications, 1996.

[9] Jaulin, L., Nonlinear bounded-error state estimation of continuous-time systems. Automatica, Vol. 38, Issue 6, pages 1079-1082, 2002.

[10] Kühn, W, Rigorously Computed Orbits of Dynamical Systems without the Wrapping Effect. Computing 61, 47-67, 1998.

[11] Orihuela, L., Rubio, F.R. and Gómez-Estern, F., Model-Based Networked Control Systems under Parametric Uncertainties. 18th IEEE International Conference on Control Applications, 2009.

[12] Raïsi, T., Efimov, D. and Zolghadri, A., Interval State Estimation for a Class of Nonlinear Systems. IEEE Transactions on Automatic Control, Vol. 57, No.1, 2012.

[13] Redondo, M.J., Aplicaciones de Técnicas DC a Identificación Paramétrica, Estimación de Estados y Conjuntos Invariantes, en Sistemas No Lineales (Tesis Doctoral)

[14] Scholte, E. and Campbell, M.E., On-line Nonlinear Guaranteed Estimation with Application to a High Performance Aircraft. Proceedings of the American Control Conference, 2002.

[15] Vu Tuan Hieu LE, Robust predictive control by zonotopic set-membership estimation (Tesis Doctoral)

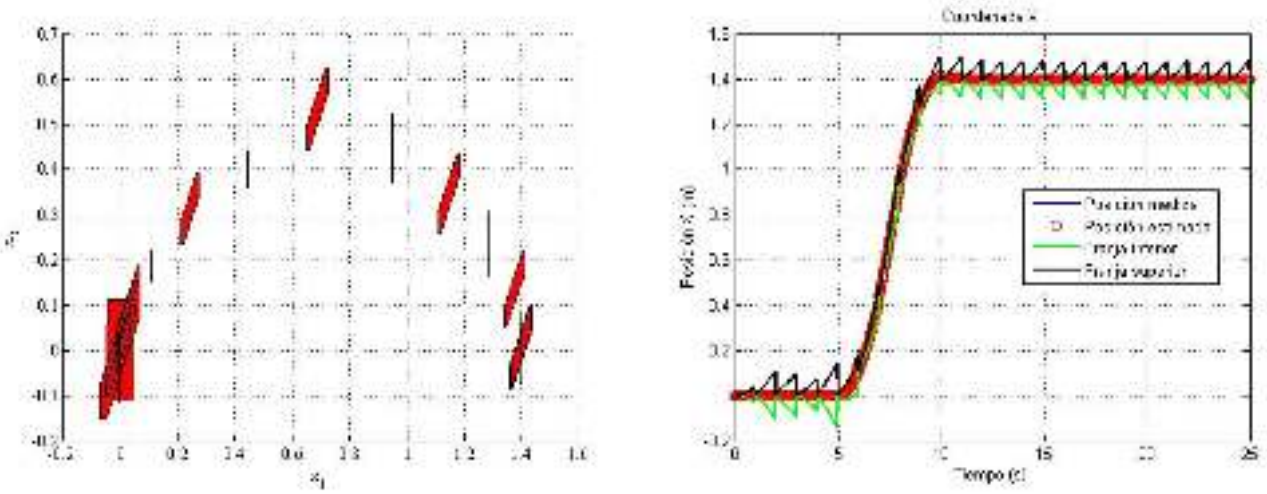


Figura 7: Zonotopo X $\dot{X}$ con GPS sin incertidumbres.

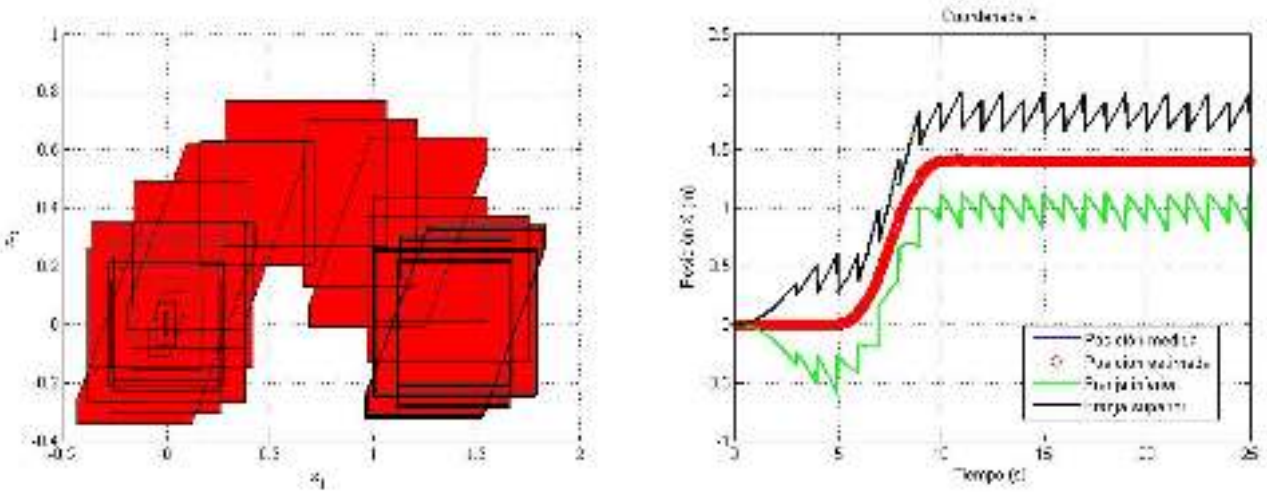


Figura 8: Zonotopo X $\dot{X}$ con GPS con incertidumbres

DISEÑO DE CONTROLADORES POR ADELANTO PARA INVERSIÓN DE RETARDO NO REALIZABLE

C. Rodríguez[1], J. L. Guzmán[1], M. Berenguel[1], T. Hägglund[2] y J. E. Normey-Rico[3]
[1]Departamento de Informática, Universidad de Almería, España {karlos.rc,joseluis.guzman,beren}@ual.es
[2]Department of Automatic Control, Lund University, Suecia, tore.hagglund@control.lth.se
[3]Departamento de Automação e Sistemas, Universidad Federal de Santa Catarina, Brasil, julio@das.ufsc.br

Resumen

Este artículo introduce dos nuevas reglas de diseño de controladores por adelanto cuando la inversión del proceso no es posible debido a un retardo no realizable. La primera regla se deriva del residuo generado en el rechazo a perturbaciones mientras que la segunda se basa en la minimización del error integral cuadrático. Finalmente, se presentan los resultados obtenidos en simulación comparados con otras reglas recientes de sintonía de controladores por adelanto.

Palabras clave: controlador por adelanto, perturbaciones de carga, filtro de atraso-adelanto, control de procesos.

1 INTRODUCCIÓN

Las perturbaciones, junto con la incertidumbre del proceso y las dinámicas inestables, son unas de las razones por las que se requiere el control realimentado. Las perturbaciones de carga son señales de baja frecuencia que afectan al proceso y sacan al sistema del punto de operación deseado. La reducción eficiente del efecto de las perturbaciones de carga es un tema clave en el control de procesos [1]. Las perturbaciones pueden ser compensadas por el controlador de realimentación, pero la principal consecuencia de esta estrategia es que dicho controlador tiene que ser sintonizado para rechazo a perturbaciones y seguimiento a referencia al mismo tiempo. El control por adelanto proporciona una solución para tratar con el rechazo a perturbaciones de forma independiente aportando acciones de control en el mismo momento que el que la perturbación afecta a la salida del proceso [5].

El compensador por adelanto ideal se forma como las dinámicas negativas entre la salida del proceso y la perturbación divididas por las dinámicas entre la salida del proceso y la señal de control. Sin embargo, este compensador ideal puede no ser realizable debido a un retardo negativo, polos en el semiplano derecho, o comportamientos de fase nomínima.

En el control de procesos, la estructura del compensador por adelanto es típicamente una ganancia o un filtro de retraso-adelanto. A veces, es necesario introducir un retardo para asegurar que la compensación no se realiza demasiado pronto. Existen compensadores con estructuras más complicadas pero son muy infrecuentes [5]. El problema de sintonizar un controlador por adelanto cuando el compensador ideal no es realizable es en la actualidad un línea de trabajo abierta. De hecho, existen pocas reglas de diseño para controladores por adelanto en la literatura [4].

La mayoría de los procedimientos de diseño para control por adelanto están basados en reglas de diseño de bucle abierto [3], [9]. En estos casos, en los que la compensación no puede ser realizada perfectamente, el controlador de realimentación contribuye en el rechazo de la perturbación. Por tanto, los resultados pueden no ser los deseados para seguimiento a referencia o rechazo a perturbaciones. La principal consecuencia de no tener en cuenta el controlador de feedback en cuenta durante el diseño en bucle abierto fue observado en [2] y [8].

Recientemente, una nueva regla de sintonía en bucle cerrado donde se tiene en cuenta el controlador de realimentación en el diseño de proceso fue presentada en [6]. Se derivaron reglas sencillas basadas en la minimización de la Integral Absoluta del Error (IAE) con restricciones en la sobreoscilación de la salida del proceso y la ganancia de alta frecuencia.

Todas las reglas de sintonía para controladores por adelanto mejoran los resultados comparados con los obtenidos por el controlador por adelanto de la parte invertible del cociente, pero todavía existen algunos problemas abiertos. El principal inconveniente es que se observan sobreoscilaciones considerables en las respuestas, lo que no es deseable en el proceso de control. Esta sobreoscilación puede ser reducida en algunas versiones de las reglas de sintonía existentes [6, 7], pero la actuación de control se deteriora considerablemente obteniendo respuestas más lentas y grandes picos en la señal de control.

El artículo se organiza como sigue. La sección 2 resume las bases teóricas entorno al control por adelanto y describe la bien conocida estructura de control para el problema de rechazo a perturbaciones. La sección 3 describe las nuevas reglas de diseño para controladores por adelanto. Los resultados de las simulaciones son presentados en la sección 4. Finalmente, se analizan las conclusiones y futuros trabajos en la sección 5.

2 CONTROL POR ADELANTO

Esta sección describre brevemente la estructura de control por adelanto clásica.

En la Figura 1 se muestra el esquema básico del compensador por adelanto. El diagrama de bloques contiene el proceso P_1P_2 el cual determina la influencia de la señal de control u en la salida del sistema y, y el proceso P_3P_2 que relaciona la perturbación medible d con la salida del sistema y. La señal de control u se compone de la acción del controlador de realimentación C y de la del controlador por adelanto C_{ff}. El objetivo es diseñar el compensador por adelanto C_{ff} de tal forma que el efecto de la perturbación d en la salida del proceso y es minimizado.

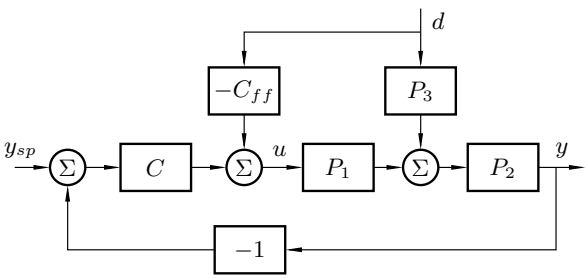


Figura 1: Esquema clásico de control por adelanto.

De la Figura 1 se pueden obtener las siguientes relaciones:

$$\begin{aligned} y &= P_2(P_1u + P_3d) \\ u &= Ce - C_{ff}d \\ e &= y_{sp} - y \end{aligned} \tag{1}$$

donde, para este trabajo, el controlador de realimentación es un PI, $C = K(sT_i + 1)/(sT_i)$, y los tres procesos — P_1, P_2 y P_3 — son funciones de transferencia de primer orden con retardo:

$$P_j = P_j^* e^{-L_j s} = \frac{K_j}{T_j s + 1} e^{-L_j s}, \qquad j = 1..3$$

Hay, por supuesto, procesos que no pueden ser bien descritos por estas simples funciones de transferencia pero para las aplicaciones de control de procesos, esta estructura suele ser suficientemente buena y se ha convertido en un model estándar en este tipo de aplicaciones.

En este trabajo, se considera el compensador por adelanto C_{ff} como un filtro de atraso-adelanto con tiempo retraso (estructuras más complejas son rara vez usadas en las plantas para el control de procesos [4]):

$$C_{ff} = K_{ff} \frac{1 + sT_z}{1 + sT_p} e^{-sL_{ff}} \tag{2}$$

Por tanto, atendiendo a (1), la influencia de la perturbación medible d en la salida del sistema y es eliminada si el compensador por adelanto se define como el cociente entre P_3 y P_1, considerando que dicho compensador sea realizable. En este caso, se puede ver cómo el error de realimentación no es influenciado por la perturbación d:

$$e = \frac{y_{sp}}{1 + P_1P_2C} \tag{3}$$

Sin embargo, el compensador (2) no es siempre realizable, por ejemplo cuando el orden de P_1 es mayor que el de P_3, el retardo de P_1 es mayor que el de P_3 ($L_3 < L_1$) o si P_1 tiene ceros en el semiplano derecho. En este trabajo, el problema en el que el controlador por adelanto ideal no es realizable debido a un tiempo de retardo negativo ($L_3 < L_1$) es analizado.

Cuando $L_3 < L_1$, el compensador por adelanto se suele definir como el cociente entre P_3^* y P_1^*, que describen las dinámicas sin retardo [6]. Sin embargo, si el error de realimentación de (1) es analizado en este caso, el efecto de la perturbación no puede ser eliminado tal y como se muestra a continuación:

$$e = \frac{y_{sp} + P_2P_3^*(e^{-L_1 s} - e^{-L_3 s})d}{1 + P_1P_2C} \tag{4}$$

Se puede comprobar, comparando (3) y (4), que ahora el efecto de la perturbación no es eliminado y es realimentado al lazo de control. En realidad, lo que es realimentado al no realizar la compensación perfecta es una dinámica residual. Sin embargo, a pesar de esto, la acción del controlador por adelanto mejora considerablemente el rechazo a perturbaciones comparado con la realimentación pura [6]. En la siguiente sección se presentan reglas de diseño de bucle abierto para el compensador por adelanto para este caso donde existen problemas de inversión de retardos.

3 DISEÑO DEL CONTROLADOR POR ADELANTO

Esta sección presenta dos nuevas reglas de diseño de compensadores por adelanto de tipo filtro de atraso-adelanto, una para eliminar las sobreoscilaciones de la salida del proceso y otra para minimizar la integral cuadrada del error (ISE), ambas para el caso donde $L_3 < L_1$. Dichas reglas están basadas en perturbaciones que entran al sistema en forma de escalón.

Las reglas se basan en las ideas propuestas en [6], donde se minimiza la IAE en lazo abierto. Además, se supone la ganancia del compensador como $K_{ff} = K_3/K_1$, y el cero del filtro de retraso-adelanto se fija de tal forma que se compense con el polo de P_1, es decir, $T_z = T_1$. De esta forma, se pretenden encontrar reglas óptimas pero a la vez sencillas. El compensador por adelanto resultante puede reescribirse como:

$$C_{ff} = K_{ff}\frac{T_z s + 1}{T_p s + 1} = \frac{K_3}{K_1}\frac{T_1 s + 1}{T_p s + 1} \qquad (5)$$

por lo que únicamente se requiere fijar el valor de T_p.

3.1 Eliminar la sobreoscilación

Esta sección describe cómo el parámetro T_p puede ser sintonizado para eliminar la sobreoscilación en la salida del proceso con la mejor actuación posible. La regla se basa en bucle abierto y por tanto, el efecto del controlador de realimentación es ignorado. De esta forma, se puede reescribir la relación entre la perturbación medible d y la salida del proceso y como:

$$\begin{aligned}\frac{y}{d} &= P_2\left(P_3 - P_{ff}\right)\\ P_{ff} &= P_1 C_{ff}\end{aligned} \qquad (6)$$

Cuando la respuesta de las funciones de transferencia que aparecen en (6) son analizadas, se puede extraer una información muy interesante. La Figura 2 muestra las respuestas ante entrada en escalón para los procesos mencionados con anterioridad variando el parámetro T_p del controlador por adelanto. Se han considerado las siguientes funciones de transferencia:

$$P_1 = \frac{0.5}{s+1}e^{-2.25s} \qquad P_3 = \frac{1}{2s+1}e^{-0.25s} \qquad (7)$$

Nótese que el proceso P_2 ha sido despreciado ya que contiene dinámicas comunes a ambos procesos P_3 y P_1, y por tanto no es relevante en este análisis. De esta forma, que la función de transferencia P_{ff} se viene dada por:

$$P_{ff} = P_1 C_{ff} = \frac{1}{T_p s + 1}e^{-0.25s} \qquad (8)$$

Se puede apreciar cómo la compensación perfecta no puede ser realizada en primer lugar debido a que $L_3 < L_1$, lo que resulta en un error residual inicial — mostrado como el área de color rojo — inherente a las características de los procesos. Además, una vez que ambas respuestas se han cruzado, aparece un error de sobreoscilación — mostrado como el área de color verde — causado por el hecho de que el proceso P_{ff} intenta compensar más rápidamente. Por tanto, si el parámetro del compensador por adelanto T_p es elegido menor, la salida del proceso sufrirá de mayor sobreoscilación, mientras que si se selecciona mayor, la compensación demorará más en rechazar el efecto de la perturbación, es decir, tendrá mayor error residual.

Del análisis previo se puede concluir que para eliminar completamente al sobreoscilación del rechazo a perturbaciones con un filtro de atraso-adelanto, los tiempos de establecimiento de ambas funciones de transferencias en (6), P_3 y P_{ff}, deben de ser iguales. Si se considera que una buena aproximación de tiempo de establecimiento de un sistema de primer orden con retardo es dada por cuatro veces su constante de tiempo más su tiempo de retardo [1], la siguiente relación puede ser obtenida:

$$T_p = \frac{4T_3 + L_3 - L_1}{4} = T_3 - \frac{L_b}{4} \qquad (9)$$

donde $L_b = L_1 - L_3$.

Nótese que esta regla solo tiene aplicación para $L_b/T_3 < 4$ ya que el parámetro T_p debe de ser positivo para alcanzar un controlador estable y distinto de cero para que sea realizable.

3.2 Minimizar el ISE

El problema de la minimización del ISE para la sintonía de un filtro de retraso-adelanto como un compensador por adelanto fue propuesto recientemente en [7]. En dicho trabajo, se realiza la optimización para ambos parámetros del filtro, T_z y T_p. Como resultado, un procedimiento relativamente complejo de sintonía fue descrito que, en la mayoría de las ocasiones, es difícil de implementar debido a que el valor propuesto para el polo requiere que T_p sea igual a cero, y por tanto, se obtiene un controlador no realizable. En este tra-

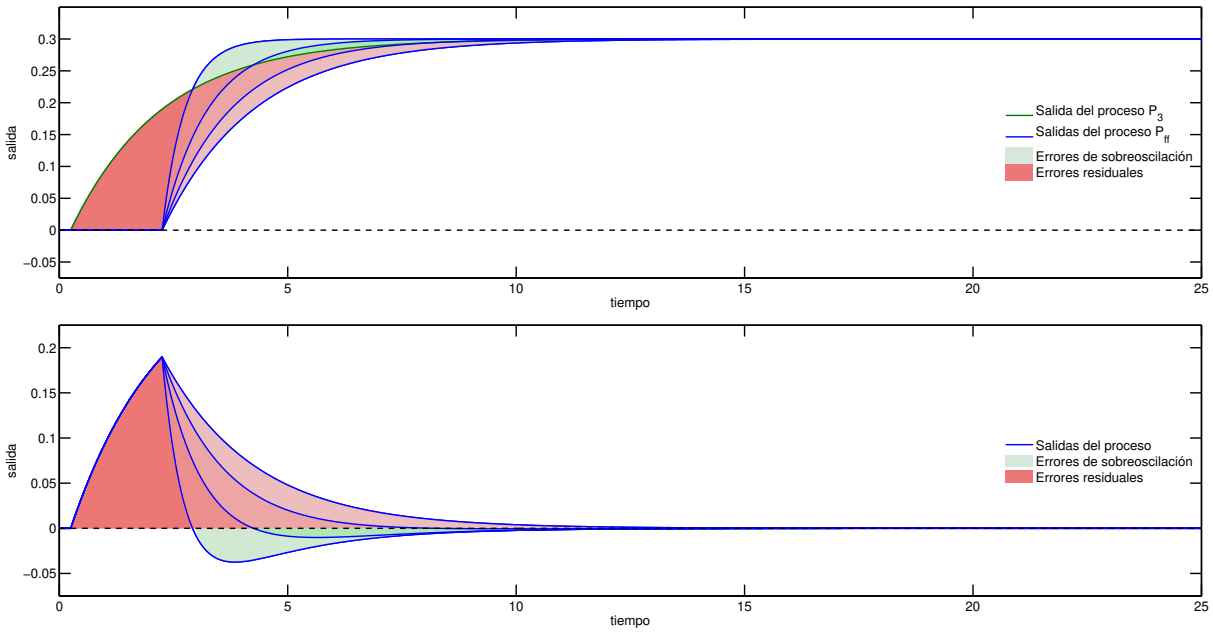


Figura 2: Análisis residual con una perturbación de carga en $t = 0$ con $d = 0.3$ e $y_{sp} = 0$. Se muestran los procesos P_3 y P_{ff} para distintos valores de T_p. Se incluyen los errores, residual y de sobreoscilación, en rojo y verde, respectivamente. También se muestran las salidas del proceso.

bajo se propone una regla de sintonía más sencilla y aplicable basándose en las ideas propuestas en [6].

Aquí, se considera $T_z = T_1$, tal y como se comentó anteriormente, y el problema de la minimización del ISE es únicamente estudiado para la variable T_p suponiendo que d es una perturbación con forma de escalón. Entonces, si el ISE es obtenido a partir de la ecuación (6) y desde el instante en el que la salida del controlador por adelanto difiere de cero, el siguiente análisis puede ser realizado:

$$\begin{aligned}\text{ISE} &= \int_{L_b}^{\infty}\left(e^{-\frac{(t-L_b)}{T_p}} - e^{-\frac{t}{T_3}}\right)^2 dt \\ &= \int_{L_b}^{\infty}\left(e^{-\frac{2(t-L_b)}{T_p}} - 2e^{-\frac{T_3(t-L_b)+T_p t}{T_3 T_p}} + e^{-\frac{2t}{T_3}}\right) dt \\ &= \frac{T_p}{2} - 2T_3\frac{T_p}{T_3+T_p}e^{-\frac{L_b}{T_3}} + \frac{T_3}{2}e^{-\frac{2L_b}{T_3}}\end{aligned} \tag{10}$$

Para minimizar el ISE, la derivada con respecto de T_p se iguala a cero:

$$\begin{aligned}\frac{d\ \text{ISE}}{d\ T_p} &= \frac{1}{2} - 2T_3 e^{-\frac{L_b}{T_3}}\left(\frac{1}{T_3+T_p} + \frac{-T_p}{(T_3+T_p)^2}\right) \\ &= \frac{1}{2} - \frac{2T_3^2}{(T_3+T_p)^2}e^{-\frac{L_b}{T_3}} = 0\end{aligned} \tag{11}$$

Por tanto, T_p es una función cuadrática con la forma:

$$T_p^2 + 2T_3T_p + T_3^2(1 - 4e^{-\frac{L_b}{T_3}}) = 0 \tag{12}$$

Nótese que solo una solución es posible debido a que T_p debe de ser positivo para alcanzar un compensador estable:

$$\begin{aligned}T_p &= \frac{-2T_3 + \sqrt{4T_3^2 - 4T_3^2(1 - 4e^{-\frac{L_b}{T_3}})}}{2} \\ &= T_3\left(2\sqrt{e^{-\frac{L_b}{T_3}}} - 1\right)\end{aligned} \tag{13}$$

Como resultado, se obtiene una regla sencilla que minimiza el ISE. Sin embargo, la aplicación de esta regla queda restringida a unos valores determinados del conciente L_b/T_3, ya que el compensador resultara inestable para valores de T_p menores que cero. Esta desigualdad es desarrollada a continuación:

$$2\sqrt{e^{-\frac{L_b}{T_3}}} - 1 > 0 \Rightarrow \frac{L_b}{T_3} < -ln(0.25) \tag{14}$$

Nótese que T_3 se ha eliminado inicialmente de la expresión anterior ya que se están considerando procesos de primer orden con retardo estables.

4 RESULTADOS

En esta sección se han realizado simulaciones para verificar las reglas de diseño propuestas en el trabajo. Ambas estrategias de diseño serán evaluadas junto con la regla presentada en [6] para la minimización del IAE y la regla de diseño clásica que incluye en el controlador por adelanto como las dinámicas invertibles, es decir, $C_{ff} = P_3^*/P_1^*$.

Se consideran los procesos descritos anteriormente en (7) sin dinámicas comunes, es decir, $P_2 = 1$. Además, se ha considerado como controlador de realimentación un PI sintonizado utilizando el método AMIGO [1], cuyos parámetros para este ejemplo resultan ser $K_c = 0.42 y T_i = 1.25$. Por otro lado, a fin de mejorar la actuación del rechazo a la perturbación, se aplicará a todos los controladores por adelanto la misma reducción de ganancia propuesta también en [6]:

$$K_{ff} = \frac{K_3}{K_1} - K_2 K_3 \Big(L_b + T_1 - T_3 + T_p - T_z \Big) \frac{K}{T_i} \quad (15)$$

En la Tabla 1 se muestran los parámetros de los controladores por adelanto.

Tabla 1: Parámetros del filtro de retraso-avance para el ejemplo en la Figura 3

	K_{ff}	T_z	T_p
Minimizar ISE	1.86	1.00	0.43
Minimizar IAE	1.72	1.00	0.82
Eliminar sobreoscilación	1.50	1.00	1.50
Dinámicas invertibles	1.33	1.00	2.00

En la Figura 3 se muestran los resultados de las simulaciones realizadas. Se puede apreciar cómo el compensador por adelanto definido a partir de la regla de minimización del ISE consigue eliminar el error residual rápidamente en detrimento de obtener mayor error de sobreoscilación. Por otro lado, el compensador ideado para eliminar la sobreoscilación realiza una compensación de la perturbación más suave.

Los valores numéricos de los resultados de la simulación se exponen en la Tabla 2. En primer lugar, se puede apreciar cómo los menores valores de ISE e IAE se obtienen para los compensadores esperados. Por otro lado, también se puede comprobar en la tabla que la estrategia ideada para eliminar la sobreoscilación obtiene resultados considerablemente mejores que en el caso del compensador básico formado por las dinámicas invertibles, obteniendo en ambos casos la misma sobreoscilación.

Tabla 2: Resultados numéricos de la simulación mostrada en la Figura 3

	IAE	*ISE*
Minimizar ISE	0.4827	**0.0427**
Minimizar IAE	**0.4677**	0.0457
Eliminar sobreoscilación	0.5556	0.0633
Dinámicas invertibles	0.6925	0.0825

5 CONCLUSIONES

Los controladores por adelanto requieren de mejores reglas de sintonía cuando el compensador ideal no es realizable. En estas situaciones, el uso de reglas de bucle cerrado mejora los resultados considerablemente.

En este trabajo se presentan dos nuevas reglas para la sintonía de controladores por adelanto. Dichas reglas, junto con la regla de bucle cerrado propuesta en [6], demuestran una mejora considerable en el rechazo a perturbaciones. La elección de la regla adecuada dependerá del problema y las especificaciones para tratar el efecto de las perturbaciones en la salida del sistema.

Finalmente, es remarcable la simplicidad de las reglas desarrolladas desde un punto de vista de aplicación práctica en la industria, si bien es cierto, que existen limitaciones en cuanto a las características de los procesos considerados y al tipo de las perturbaciones. La posibilidad de encontrar reglas generales simples en estos otros casos disminuye al aumentar la complejidad del problema.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los siguientes proyectos: PHB2009-0008 (financiado por el Ministerio Español de Educación, CNPq; CAPES-DGU 220/2010); DPI2010-21589-C05-04 y DPI2011-27818-C02-01 (financiados por el Ministerio de Economía y Competitividad de España y los fondos FEDER).

Referencias

[1] K. J. Åström and T. Hägglund, *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society, Research Triangle Park, NC 27709, 2005.

[2] C. Brosilow and B. Joseph, *Techniques of Model-Based Control*. New Jersey: Prentice-Hall, 2002.

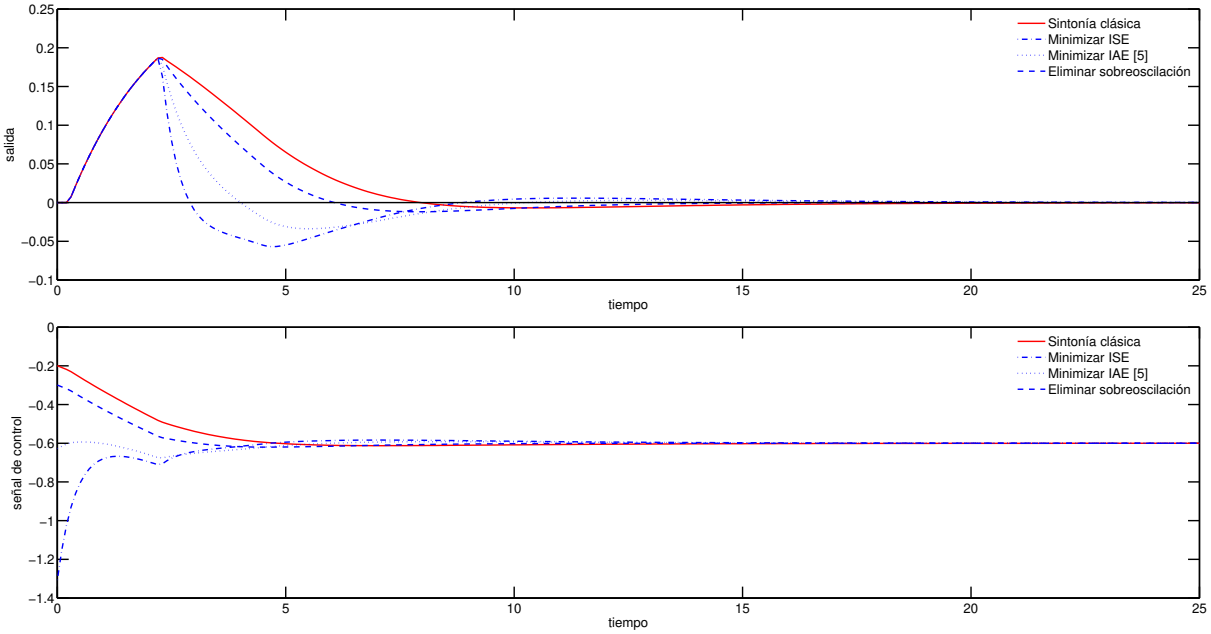


Figura 3: Comparativa de las distintas reglas de sintonía de controladores por adelanto para filtro de retraso-avance con perturbación de carga en el instante $t = 0$, $d = 0.3$, y $y_{sp} = 0$. Las gráficas muestran los resultados para los cuatro compensadores por adelanto con los parámetros mostrados en la Tabla 1. Además, se presentan las señales de control, incluyendo el esfuerzo del compensador por adelanto más el controlador de realimentación.

[3] D. R. Coughanowr, *Process Systems, Analysis and Control*. New York: McGraw-Hill, 1991.

[4] J. L. Guzmán, T. Hägglund and A. Visioli, "Feedforward compensation for PID control loops", in PID Control in the Third Millenium: Lessons Learned and New Approaches. Springer, 2012.

[5] J. L. Guzmán, T. Hägglund, K. J. Åström, S. Dormido, M. Berenguel and Y. Piguet, "Feedforward control concepts through interactive tools", 18th IFAC World Congress, Milano, Italy, 2011.

[6] J. L. Guzmán and T. Hägglund, "Simple tuning rules for feedforward compensators", Journal of Process Control, 21(1), pp. 92-102, 2011.

[7] M. Hast and T. Hägglund, "Design of optimal low-order feedforward controllers", IFAC Conference on Advances in PID Control, Brescia, 2012.

[8] A. Isaksson, M. Molander, P. Modén, T. Matsko and K. Starr, "Low-order feedforward design optimizing the closed-loop response", Preprints, Control Systems Pan Pacific Conference, Vancouver, Canada, 2008.

[9] D. Seborg, T. Edgar and D. Mellichamp, *Process Dynamics and Control*. New York: Wiley, 1989.

RESTRICCIONES EN EL VALOR DE LOS ENLACES DE COMUNICACIÓN EN UN SISTEMA DE CONTROL COALICIONAL

F. J. Muros[†], J. M. Maestre[†], E. Algaba[‡], E. F. Camacho[†]

{franmuros,pepemaestre,ealgaba,efcamacho}@us.es

[†]*Departamento de Ingeniería de Sistemas y Automática, Escuela Superior de Ingenieros, Sevilla*
[‡]*Departamento de Matemática Aplicada II, Escuela Superior de Ingenieros, Sevilla*

Resumen

En este artículo se estudia un esquema de control distribuido para sistemas lineales en el cual los controladores locales trabajan de forma cooperativa entre sí. Se considera una topología de red variante en el tiempo, desconectando en cada instante los enlaces de comunicación que contribuyan menos al beneficio del sistema global. Este esquema de control se modela como un juego cooperativo, mediante el uso de herramientas como el valor de Shapley. En particular, se muestra una forma novedosa de expresar dicho valor, muy apropiada en el contexto del control. Finalmente, se analiza el requerimiento de determinadas restricciones al valor de Shapley en la fase de diseño del controlador.

Palabras clave: Enlaces de Comunicación, Valor de Shapley, Teoría de Juegos, DMPC Distribuido, Control Coalicional.

1 INTRODUCCIÓN

En la última década se ha experimentado un gran avance en el campo del control distribuido. La reducción de recursos necesarios en comparación con el control centralizado, o su inherente modularidad y escalabilidad, son solo algunos ejemplos de los beneficios de esta metodología de control. La idea básica es descomponer el sistema de control global en varios subsistemas más pequeños, regulados cada uno por un controlador local o *agente* que usa la información de su propio estado y que puede compartir o no información con el resto. Existen multitud de aplicaciones de este tipo de esquemas, como redes de tráfico o potencia [9], o cadenas de suministro [6], entre otras.

En este contexto, la literatura está principalmente centrada en algoritmos específicos de control distribuido implementado por controladores locales (en [11] se puede encontrar una amplia recopilación), o trabajos que intentan resolver cuestiones como la dinámica inducida por la red de comunicación [5]. No obstante, en la mayoría de los casos, estos esquemas suelen asumir que el sistema global está dividido en un conjunto estático de vecindades o *coaliciones* prácticamente desacopladas, existiendo un fuerte acoplamiento de los controladores dentro de cada vecindad. En muy pocos trabajos se consideran vecindades no estáticas, como en [14], donde se usa un conjunto de restricciones dinámicas para modificar el conjunto de agentes que deben trabajar juntos.

En este artículo, se extienden los resultados de un trabajo previo [7] que también estudia esta cuestión, mediante un esquema de control distribuido que considera específicamente vecindades variantes en el tiempo, es decir, el sistema de control dirige de forma explícita los enlaces de comunicación usados para conectar los controladores locales, y dichos enlaces pueden estar habilitados o deshabilitados dependiendo de su contribución a las prestaciones del controlador global. En el trabajo mencionado, se aplicaron herramientas de teoría de juegos para tener un conocimiento más amplio de la relevancia de los enlaces de comunicación que interconectan a los controladores locales. La teoría de juegos ha estudiado durante décadas situaciones de interacción mutua, comenzando con los influyentes trabajos de von Neumann [15]. Por esta razón, resulta de enorme interés extrapolar los resultados de este campo al contexto del control distribuido, donde los controladores locales pueden agruparse en coaliciones, al igual que ocurre en los juegos cooperativos.

Más específicamente, en [7] se estudian juegos cooperativos con cooperación restringida [2, 8] para plantear el esquema de control con topología de red cambiante como un juego de enlaces. En este caso, la asignación de los costes globales realizada sobre el conjunto de enlaces fue calculada en base al valor de Shapley [12], una regla de pagos muy utilizada en teoría de juegos cooperativos, que resultó de gran utilidad para proporcionar una medida de la relevancia de cada enlace que dirige el esquema de control. En este trabajo, se han extendido los resultados previos [7] para trabajar en las siguientes direcciones:

- Se formula el valor de Shapley introduciendo una notación matricial que simplifica el cálculo de dicho valor en el contexto del control. Gracias a esta nueva notación será posible encontrar una expresión cerrada que permitirá representar el valor de Shapley como función del estado. Es más, una vez que se fije la ley de control, se podrá obtener el valor de Shapley como función del tiempo, lo que permitirá analizar un problema de control distribuido como un juego cooperativo dinámico.
- En base a la nueva notación, se han desarrollado condiciones que se pueden añadir a la fase de diseño de las leyes de control del esquema propuesto en [7], obteniéndose unas nuevas matrices de control que garantizan que el valor de Shapley satisfaga ciertos requisitos, por ejemplo, que el valor asignado a un cierto jugador (enlace) permanezca siempre por debajo de un umbral dado.

Para alcanzar los objetivos mencionados arriba, se utilizarán conceptos de álgebra matricial y desigualdades lineales matriciales (LMIs). Bajo ciertos supuestos, el uso de LMIs proporciona una poderosa herramienta para resolver problemas de control, alcanzando resultados muy satisfactorios con respecto a estabilidad y tiempo de cálculo. Para más detalles, consultar [1, 3].

El resto del artículo se organiza de la siguiente manera. Primeramente, se plantea el problema de control. En segundo lugar, se proporcionan algunos conceptos de teoría de juegos cooperativos, y se introduce la formulación matricial comentada. A continuación, se presenta una técnica de diseño del controlador basada en LMIs, y que incluye restricciones en el valor de Shapley. Finalmente, aplicando las herramientas que proporciona Matlab®, se ha podido abordar la computación e ilustrar el algoritmo planteado con un ejemplo.

2 PLANTEAMIENTO DEL PROBLEMA

Sea la clase de sistemas distribuidos lineales compuesto por un conjunto de $\mathcal{N} = \{1, 2, \ldots, N\}$ subsistemas interconectados entre sí, acoplados por las entradas y los estados, y cuya dinámica puede ser descrita matemáticamente como

$$\begin{array}{c} \mathbf{x}_i(k+1) = \mathbf{A}_{ii}\mathbf{x}_i(k) + \mathbf{B}_{ii}\mathbf{u}_i(k) + \mathbf{d}_i(k), \\ \mathbf{d}_i(k) = \sum_{j\neq i} \left[\mathbf{A}_{ij}\mathbf{x}_j(k) + \mathbf{B}_{ij}\mathbf{u}_j(k)\right], \end{array} \quad (1)$$

donde $\mathbf{x}_i(k) \in \mathbb{R}^{n_{\mathrm{x}_i}}, i \in \mathcal{N}$ son los estados del subsistema i, $\mathbf{u}_i(k) \in \mathbb{R}^{n_{\mathrm{u}_i}}, i \in \mathcal{N}$ sus respectivas entradas, y $\mathbf{A}_{ij} \in \mathbb{R}^{n_{\mathrm{x}_i} \times n_{\mathrm{x}_j}}, \mathbf{B}_{ij} \in \mathbb{R}^{n_{\mathrm{x}_i} \times n_{\mathrm{u}_j}}, i, j \in \mathcal{N}$ matrices de transición. Se utilizará $\mathbf{d}_i(k)$ para denotar la influencia sobre el subsistema i-ésimo de los subsistemas restantes.

Los estados están confinados en conjuntos independientes definidos por una serie de desigualdades lineales

$$\mathbf{x}_i(k) \in \mathcal{X}_i, \quad \mathcal{X}_i \subseteq \mathbb{R}^{n_{\mathrm{x}_i}}, \quad i \in \mathcal{N}. \quad (2)$$

Se considera que cada subsistema está controlado por un agente diferente que tiene información completa sobre su modelo local y su estado $\mathbf{x}_i(k)$, y es capaz de manipular su acción de control $\mathbf{u}_i(k)$. Los agentes pueden comunicarse a través de una red cuya topología se describe mediante un grafo no dirigido $(\mathcal{N}, \mathcal{E})$, siendo $\mathcal{N}$ el conjunto de agentes, y $\mathcal{E} \subseteq \mathcal{E}^{\mathcal{N}} = \mathcal{N} \times \mathcal{N}$ el conjunto de enlaces correspondiente a los caminos físicos posibles entre los agentes. Cada enlace $l \in \mathcal{E}$ puede estar habilitado o deshabilitado en un determinado paso de control, asumiendo que cada enlace habilitado tendrá un coste fijo $c \in \mathbb{R}^+$.

Definición 1 *Se denota como modo de red $\Lambda(k)$ al conjunto de enlaces habilitados en un determinado paso de control k. Nótese que hay $2^{|\mathcal{E}|}$ modos de red distintos en una red $(\mathcal{N}, \mathcal{E})$, los cuales se representarán como $\Lambda_0(k), \Lambda_1(k), \ldots, \Lambda_{2^{|\mathcal{E}|}-1}(k)$.*

En la Figura 1 se observan dos esquemas $(\mathcal{N}, \mathcal{E})$, con los agentes representados por cajas y números arábicos, y los enlaces por líneas y números romanos. Para cada enlace se ilustran los modos de red que tendrían habilitado dicho enlace, por ejemplo, $\Lambda_7^{\mathrm{a})}(k) = \{\mathcal{E}\}$ o $\Lambda_7^{\mathrm{b})}(k) = \{I, IV\}$.

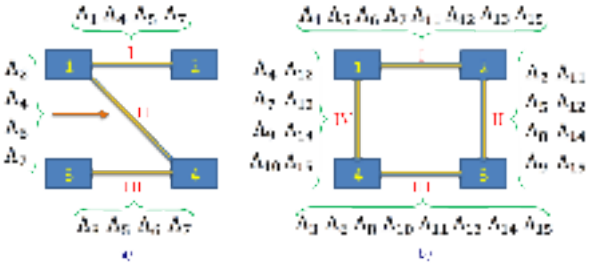


Figura 1: Esquemas con 8 y 16 modos de red $\Lambda(k)$.

El sistema global puede describirse como

$$\begin{array}{c} \mathbf{x}_{\mathcal{N}}(k+1) = \mathbf{A}_{\mathcal{N}}\mathbf{x}_{\mathcal{N}}(k) + \mathbf{B}_{\mathcal{N}}\mathbf{u}_{\mathcal{N}}(k), \\ \mathbf{x}_{\mathcal{N}}(k) \in \mathcal{X}_{\mathcal{N}}, \end{array} \quad (3)$$

donde $\mathbf{x}_{\mathcal{N}}(k) = \left[\mathbf{x}_1(k), \ldots, \mathbf{x}_{|\mathcal{N}|}(k)\right]^{\mathrm{T}} \in \mathbb{R}^{n_{\mathrm{x}_{\mathcal{N}}}}$, $\mathbf{u}_{\mathcal{N}}(k) = \left[\mathbf{u}_1(k), \ldots, \mathbf{u}_{|\mathcal{N}|}(k)\right]^{\mathrm{T}} \in \mathbb{R}^{n_{\mathrm{u}_{\mathcal{N}}}}$ son los vectores de estados y entradas globales, $\mathbf{A}_{\mathcal{N}} = [\mathbf{A}_{ij}]_{i,j\in\mathcal{N}} \in \mathbb{R}^{n_{\mathrm{x}_{\mathcal{N}}} \times n_{\mathrm{x}_{\mathcal{N}}}}$, $\mathbf{B}_{\mathcal{N}} = [\mathbf{B}_{ij}]_{i,j\in\mathcal{N}} \in \mathbb{R}^{n_{\mathrm{x}_{\mathcal{N}}} \times n_{\mathrm{u}_{\mathcal{N}}}}$ las matrices de transición globales, y $\mathcal{X}_{\mathcal{N}} = [\mathcal{X}_i]_{i\in\mathcal{N}} \subseteq \mathbb{R}^{n_{\mathrm{x}_{\mathcal{N}}}}$ el conjunto de restricciones de estado del problema completo.

Definición 2 *Se introduce la función de coste del sistema global como*

$$J = \left(\sum_{j=0}^{\infty} \left[\mathbf{x}_{\mathcal{N}}^{\mathrm{T}}(k+j) \mathbf{Q}_{\mathcal{N}} \mathbf{x}_{\mathcal{N}}(k+j) + \mathbf{u}_{\mathcal{N}}^{\mathrm{T}}(k+j) \mathbf{R}_{\mathcal{N}} \mathbf{u}_{\mathcal{N}}(k+j) \right] \right) + c|\Lambda(k)| = J_s + J_c, \tag{4}$$

con J_s, J_c los costes de control y de comunicación, respectivamente, y $\mathbf{Q}_{\mathcal{N}} \in \mathbb{R}^{n_{\mathbf{x}_{\mathcal{N}}} \times n_{\mathbf{x}_{\mathcal{N}}}}, \mathbf{R}_{\mathcal{N}} \in \mathbb{R}^{n_{\mathbf{u}_{\mathcal{N}}} \times n_{\mathbf{u}_{\mathcal{N}}}}$ matrices de pesos definidas positivas[1].

Suposición 1 *Se establece como objetivo de control la búsqueda de un conjunto de matrices $\mathbf{K}_\Lambda \in \mathbb{R}^{n_{\mathbf{u}_{\mathcal{N}}} \times n_{\mathbf{x}_{\mathcal{N}}}}$ que verifique, $\forall \Lambda \subseteq \mathcal{E}$*

$$\mathbf{u}_{\mathcal{N}} = \mathbf{K}_\Lambda \mathbf{x}_{\mathcal{N}}, \tag{5}$$

con el propósito de regular el sistema al origen, cumpliendo las restricciones (2), y minimizando a su vez la función de coste (4).

Suposición 2 *Existe un conjunto de matrices definidas positivas $\mathbf{P}_\Lambda \in \mathbb{R}^{n_{\mathbf{x}_{\mathcal{N}}} \times n_{\mathbf{x}_{\mathcal{N}}}}$ que proporcionan una función de Lyapunov $f(\mathbf{x}_{\mathcal{N}}) = \mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}}$ del sistema en bucle cerrado, y además verifican, $\forall \Lambda \subseteq \mathcal{E}$*

$$\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}} \geq J_s. \tag{6}$$

Comentario 1 *Nótese que, si no existe camino físico posible entre dos agentes i y j en un modo de red determinado $\Lambda \subseteq \mathcal{E}$, los subbloques correspondientes de las matrices $\mathbf{K}_\Lambda$ y $\mathbf{P}_\Lambda$ que relacionan al agente i con el agente j son nulos.*

Basándose en las suposiciones realizadas, se introduce la siguiente expresión ya definida previamente en [7], que representa una cota superior de la función de coste J, $\forall \Lambda \subseteq \mathcal{E}$

$$r^v(\Lambda, \mathbf{x}_{\mathcal{N}}) = \mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}} + c|\Lambda|. \tag{7}$$

Se usará (7) en la fase de diseño del controlador global, como se verá en el Apartado 5.

3 TEORÍA DE JUEGOS

En este apartado, se estudiará la aplicación de las herramientas de teoría de juegos para averiguar qué enlaces y agentes son más importantes en nuestro esquema distribuido. Se considerará específicamente la teoría de juegos cooperativos, que se centra en situaciones de interacción mutua entre un conjunto de jugadores, y cuyo objetivo es estudiar qué coaliciones de jugadores son las más probables y cómo distribuir los costes o beneficios derivados de la cooperación entre ellos.

Un juego cooperativo de utilidad transferible es la dupla $(\mathcal{R}, \boldsymbol{v})$, donde $\mathcal{R}$ es el conjunto de jugadores y $\boldsymbol{v}$ es la función característica que asigna un valor a cada posible coalición $\mathcal{S} \subseteq \mathcal{R}$ de jugadores, con $v(\emptyset) = 0$. La función $v(\mathcal{S})$ da una medida del coste o beneficio que la coalición $\mathcal{S}$ consigue cuando alcanza el objetivo común sin la colaboración del resto de jugadores.

El punto de vista clásico es identificar al conjunto de jugadores $\mathcal{R}$ con el conjunto de agentes $\mathcal{N}$, asumiendo que cualquier coalición de agentes es factible. En [7], la clave para conectar los campos de teoría de control y juegos cooperativos es interpretar (7) como la función característica de un juego cooperativo donde, en este caso, se considera al conjunto de enlaces $\mathcal{E}$ como el conjunto de jugadores. En otras palabras, cada modo de red Λ se corresponde con una coalición de enlaces, y (7) se utiliza para obtener su valor, mediante el denominado *juego de enlaces* $(\mathcal{E}, \boldsymbol{r^v})$.

Comentario 2 *Nótese que, si se considera la topología de red descentralizada $\Lambda_0 = \{\emptyset\}$, i.e., con todos los subsistemas aislados, se obtiene $r^v(\Lambda_0, \mathbf{x}_{\mathcal{N}}) \neq 0$. Por tanto, para que (7) pueda usarse como función de coste de un juego debe redefinirse, $\forall \Lambda \subseteq \mathcal{E}$, como*

$$r^{v\prime}(\Lambda, \mathbf{x}_{\mathcal{N}}) = r^v(\Lambda, \mathbf{x}_{\mathcal{N}}) - r^v(\Lambda_0, \mathbf{x}_{\mathcal{N}}). \tag{8}$$

3.1 Análisis a nivel de enlaces

En teoría de juegos cooperativos, la distribución del coste o beneficio entre los jugadores se basa en el uso de reglas de asignación, que son herramientas matemáticas que proporcionan un vector de pagos que especifica el beneficio o coste que cada jugador puede razonablemente esperar del juego.

Como nuestra función característica se basa en los costes de control y de comunicación asociados a cada coalición de enlaces, la regla de asignación escogida proporciona el correspondiente coste de cada enlace. En este trabajo, se usará el valor de Shapley [12], que asigna al juego $(\mathcal{E}, \boldsymbol{r^v})$ el vector $\boldsymbol{\phi}(\mathcal{E}, \boldsymbol{r^v})$, que se define $\forall l \in \mathcal{E}$, como

$$\phi_l(\mathcal{E}, \boldsymbol{r^v}) = \sum_{\substack{\Lambda \subseteq \mathcal{E} \\ l \notin \Lambda}} \frac{|\Lambda|!(|\mathcal{E}| - |\Lambda| - 1)!}{|\mathcal{E}|!} [r^v(\Lambda \cup \{l\}) - r^v(\Lambda)]. \tag{9}$$

El valor de Shapley verifica algunas propiedades interesantes como *Aditividad*, *Eficiencia*, *Simetría* o *Tratamiento del jugador pasivo* [12].

3.2 Análisis a nivel de agentes

En [2] se realiza un análisis del juego de enlaces a nivel de agentes para situaciones de comunicación, i.e., la terna $(\mathcal{N}, \boldsymbol{v}, \mathcal{E})$ donde $\mathcal{N}$ es el conjunto de agentes, $(\mathcal{N}, \boldsymbol{v})$ es un juego cooperativo de utilidad

[1] En lo sucesivo se omitirá la dependencia con el paso de control k, por simplicidad de notación.

transferible, y $(\mathcal{N}, \mathcal{E})$ es un grafo no dirigido. El valor de posición genera un vector de pagos para los agentes usando el valor de Shapley de cada enlace del juego. Entonces, asumiendo que cada agente puede usar cualquier enlace que lo conecte, parece razonable dividir el valor de Shapley de un enlace de forma equitativa entre los dos agentes que se encuentran directamente interconectados por dicho enlace. El valor de posición asociado al agente i–ésimo se puede calcular como [2]

$$\pi_i(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) = \sum_{l \in \mathcal{E}_i} \frac{1}{2} \phi_l(\mathcal{E}, \boldsymbol{r^v}), \quad \forall i \in \mathcal{N}, \quad (10)$$

donde $\mathcal{E}_i$ representa el subconjunto de enlaces conectados al agente i, y $\phi_l(\mathcal{E}, \boldsymbol{r^v})$ se refiere al valor de Shapley del enlace l. Nótese que si no hay enlaces conectados a un cierto agente $j \in \mathcal{N}$, se cumple $\pi_j(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) = 0$.

Esta regla de asignación es la única que verifica las propiedades de *Eficiencia por componentes* y *Contribuciones equilibradas* [13].

4 FORMULACIÓN MATRICIAL

El propósito de este apartado es encontrar una expresión matricial que permita representar los valores de Shapley y de posición como función del juego de enlaces. Para este cometido, se introduce una formulación que utiliza dos matrices, $\mathbf{M}$ y $\mathbf{\Pi}$. Por un lado, la matriz $\mathbf{M}$ contiene la información asociada al tamaño de las coaliciones de enlaces y su expresión solo depende del número de enlaces de un juego dado. Por otra parte, en la matriz $\mathbf{\Pi}$ se describe implícitamente la topología de la red. Esta descomposición matricial resultará de mucha utilidad para trabajar con esquemas de control distribuidos.

Definición 3 *Sea la matriz* $\mathbf{M} \in \mathbb{R}^{|\mathcal{E}| \times 2^{|\mathcal{E}|}}$*, donde las filas corresponden a cada enlace* $l \in \mathcal{E}$ *y las columnas a cada uno de los diferentes modos de red* $\Lambda \subseteq \mathcal{E}$*. El elemento* $m_{l\Lambda}$ *de* $\mathbf{M}$ *se define como*

$$m_{l\Lambda} = \begin{cases} \frac{(|\Lambda|-1)!(|\mathcal{E}|-|\Lambda|)!}{|\mathcal{E}|!} & l \in \Lambda, \\ -\frac{|\Lambda|!(|\mathcal{E}|-|\Lambda|-1)!}{|\mathcal{E}|!} & l \notin \Lambda. \end{cases} \quad (11)$$

Proposición 1 *Sea la matriz* $\mathbf{M}$*, con sus elementos* $m_{l\Lambda}$ *definidos mediante (11). Se verifica*

$$\boldsymbol{\phi}(\mathcal{E}, \boldsymbol{r^v}) = \begin{bmatrix} \phi_I \\ \phi_{II} \\ \vdots \\ \phi_{|\mathcal{E}|} \end{bmatrix} = \mathbf{M} \begin{bmatrix} r^v(\Lambda_0, \mathbf{x}_\mathcal{N}) \\ r^v(\Lambda_1, \mathbf{x}_\mathcal{N}) \\ r^v(\Lambda_2, \mathbf{x}_\mathcal{N}) \\ \vdots \\ r^v(\Lambda_{2^{|\mathcal{E}|}-1}, \mathbf{x}_\mathcal{N}) \end{bmatrix} = \mathbf{M} \boldsymbol{r^v}. \quad (12)$$

Comentario 3 *Nótese que la matriz* $\mathbf{M}$ *conecta la función característica del juego de enlaces con su correspondiente valor de Shapley, y solo depende del número de jugadores de dicho juego de enlaces. Por tanto, se tendrá una matriz unívoca* $\mathbf{M}_{|\mathcal{E}|}$ *para todas las posibles combinaciones de los juegos de enlaces con* $|\mathcal{E}|$ *enlaces.*

Utilizando (11) se puede obtener el valor de la matriz $\mathbf{M}$ para juegos con cualquier número de enlaces. Por ejemplo, para el caso de juegos con 3 enlaces

$$\mathbf{M}_3 = \begin{bmatrix} -\frac{1}{3} & \frac{1}{3} & -\frac{1}{6} & -\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & -\frac{1}{3} & \frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} & \frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{6} & -\frac{1}{6} & \frac{1}{3} & -\frac{1}{3} & \frac{1}{6} & \frac{1}{6} & \frac{1}{3} \end{bmatrix}.$$

Con el objetivo de extender el análisis a los agentes, se presenta la siguiente definición

Definición 4 *Sea la matriz* $\mathbf{\Pi} \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{E}|}$*, donde las filas corresponden a cada agente* $i \in \mathcal{N}$ *y las columnas a cada enlace* $l \in \mathcal{E}$*. El elemento* Π_{il} *de* $\mathbf{\Pi}$ *se define como*

$$\Pi_{il} = \begin{cases} \frac{1}{2} & l \in \mathcal{E}_i, \\ 0 & l \notin \mathcal{E}_i. \end{cases} \quad (13)$$

Proposición 2 *Sea la matriz* $\mathbf{\Pi}$*, con sus elementos* Π_{il} *definidos como (13). Se verifica*

$$\boldsymbol{\pi}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{|\mathcal{N}|} \end{bmatrix} = \mathbf{\Pi} \boldsymbol{\phi}(\mathcal{E}, \boldsymbol{r^v}) = \mathbf{\Pi M} \boldsymbol{r^v}. \quad (14)$$

Comentario 4 *Nótese que la matriz* $\mathbf{\Pi}$ *conecta el valor de Shapley del juego de enlaces con el valor de posición del juego original, por tanto depende de la topología de la red* $(\mathcal{N}, \mathcal{E})$ *considerada. Es más, se tendrá una matriz unívoca* $\mathbf{\Pi}_{(\mathcal{N},\mathcal{E})}$ *que define cada red* $(\mathcal{N}, \mathcal{E})$*.*

Por ejemplo, para las diferentes topologías que se observan en la Figura 1, el valor de $\mathbf{\Pi}$ se puede obtener mediante (13)

$$\mathbf{\Pi}_\mathrm{a} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{\Pi}_\mathrm{b} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

5 DISEÑO DEL CONTROLADOR

En este apartado se presenta un método para diseñar las matrices que definen los controladores del sistema. Para ello, se reescribe el problema completo como una serie de LMIs dependientes de un conjunto dado de variables de

decisión matriciales. Dadas las variables matriciales $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$ y la función matricial $\mathbf{H}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m)$, se dice que la desigualdad matricial $\mathbf{H}(\cdot) > 0$ (o análogamente $\mathbf{H}(\cdot) < 0$) es una desigualdad lineal matricial (LMI) en las variables de decisión $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$, si $\mathbf{H}(\cdot)$ es una matriz simétrica para cada $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$ y la dependencia de $\mathbf{H}(\cdot)$ con respecto a $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$ es afín [1].

El conjunto de matrices $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$ que, de forma simultánea, satisfacen todas las desigualdades lineales matriciales es un conjunto convexo. Por tanto, cada LMI impone una restricción convexa sobre las variables de decisión [1]. Es más, se verifica que los algoritmos de punto interior [10] encuentran una solución del problema invirtiendo un tiempo computacional razonable [1].

5.1 Resolución del Problema de Estabilidad

Considérese el problema de obtener una realimentación lineal del estado $\mathbf{u}_\mathcal{N} = \mathbf{K}_\Lambda \mathbf{x}_\mathcal{N}$ para el esquema distribuido objeto de estudio. Se puede probar que el sistema global (3) es estable [1] si se cumple

$$\mathbf{P}_\Lambda - (\mathbf{A}_\mathcal{N} + \mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda)^{\mathrm{T}} \mathbf{P}_\Lambda (\mathbf{A}_\mathcal{N} + \mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda) > 0 \tag{15}$$

con $\mathbf{P}_\Lambda$ definida positiva. En [7], se tiene también en cuenta el requisito (6) en la etapa de diseño, mediante la inclusión de un término adicional en (15), de donde se llega al siguiente sistema a resolver

$$\left.\begin{array}{c} \mathbf{P}_\Lambda > 0, \\ \mathbf{P}_\Lambda - (\mathbf{A}_\mathcal{N} + \mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda)^{\mathrm{T}} \mathbf{P}_\Lambda (\mathbf{A}_\mathcal{N} + \mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda) \cdots \\ \cdots - \mathbf{Q}_\mathcal{N} - \mathbf{K}_\Lambda^{\mathrm{T}} \mathbf{R}_\mathcal{N} \mathbf{K}_\Lambda \end{array}\right\} > 0. \tag{16}$$

Es fácil comprobar que la segunda desigualdad de (16) no cumple la propiedad de afinidad, luego no es posible considerar a dicho sistema como LMI en las variables de decisión $\mathbf{K}_\Lambda$ y $\mathbf{P}_\Lambda$. No obstante, mediante el uso del complemento de Schur [16] y realizando el cambio de variable $\mathbf{W}_\Lambda = \mathbf{P}_\Lambda^{-1}$ y $\mathbf{Y}_\Lambda = \mathbf{K}_\Lambda \mathbf{P}_\Lambda^{-1}$ [1], es posible reescribir (16) como

$$\begin{bmatrix} \mathbf{W}_\Lambda & \mathbf{W}_\Lambda \mathbf{A}_\mathcal{N}^{\mathrm{T}} + \mathbf{Y}_\Lambda^{\mathrm{T}} \mathbf{B}_\mathcal{N}^{\mathrm{T}} & \mathbf{W}_\Lambda \mathbf{Q}_\mathcal{N}^{1\backslash 2} & \mathbf{Y}_\Lambda^{\mathrm{T}} \mathbf{R}_\mathcal{N}^{1\backslash 2} \\ \mathbf{A}_\mathcal{N}\mathbf{W}_\Lambda + \mathbf{B}_\mathcal{N}\mathbf{Y}_\Lambda & \mathbf{W}_\Lambda & 0 & 0 \\ \mathbf{Q}_\mathcal{N}^{1\backslash 2}\mathbf{W}_\Lambda & 0 & \mathbf{I} & 0 \\ \mathbf{R}_\mathcal{N}^{1\backslash 2}\mathbf{Y}_\Lambda & 0 & 0 & \mathbf{I} \end{bmatrix} > 0, \tag{17}$$

en cuyo caso sí proporciona una condición LMI en las variables de decisión $\mathbf{W}_\Lambda$ y $\mathbf{Y}_\Lambda$. Aplicando la condición (17), podemos extraer fácilmente las matrices $\mathbf{K}_\Lambda$ y $\mathbf{P}_\Lambda$, que verifiquen (16) para cada modo de red $\Lambda \subseteq \mathcal{E}$.

Obsérvese que, con la idea de tener en cuenta el Comentario 1, se deben imponer condiciones de diseño adicionales a la estructura de bloques de las matrices $\mathbf{K}_\Lambda$ y $\mathbf{P}_\Lambda$, (o equivalentemente a $\mathbf{W}_\Lambda$ y $\mathbf{Y}_\Lambda$). Para más información, consultar [1].

5.2 Restricciones en el Valor de Shapley

El valor de Shapley del juego de los enlaces proporciona el coste de cada enlace cuando se une de forma aleatoria a una coalición determinada, i.e., distribuye el coste de la gran coalición entre los enlaces que toman parte en el juego. En otras palabras, cuanto mayor sea el valor de Shapley de un enlace, más prescindible para el sistema es dicho enlace. En algunas situaciones, puede ser necesario mantener el valor de Shapley de ciertos enlaces por encima o por debajo de ciertos límites para forzar al sistema a considerarlos críticos o innecesarios.

Llegados a este punto, es importante notar que el valor de Shapley del juego modificado según el Comentario 2, permanece constante, porque ambas funciones de coste solo difieren en un término $\mathbf{x}_\mathcal{N}^{\mathrm{T}} \mathbf{P}_{\Lambda_0} \mathbf{x}_\mathcal{N} > 0$, que no depende de la coalición Λ. Es decir, teniendo en cuenta que

$$\sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} = 0, \quad \forall l \in \mathcal{E}, \tag{18}$$

puede probarse fácilmente

$$\phi_l{}' = \phi_l, \quad \forall l \in \mathcal{E}. \tag{19}$$

A continuación, se definen los diferentes tipos de restricciones en el valor de Shapley que se van a considerar en este estudio. Se distinguirá entre restricciones absolutas y relativas:

- Absoluta: Se impone que el valor de Shapley de un determinado enlace $l \in \mathcal{E}$ esté por encima o por debajo de un valor fijo $\mathcal{V}_l \in \mathbb{R}$ dado

  a) $\boldsymbol{\phi}_l(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}) < \mathcal{V}_l,$ (20)

  b) $\boldsymbol{\phi}_l(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}) > \mathcal{V}_l.$ (21)

- Relativa: Se impone que el valor de Shapley de un determinado enlace $l_p \in \mathcal{E}$ sea mayor que[2] el valor de Shapley de otro enlace $l_q \in \mathcal{E}$

  c) $\boldsymbol{\phi}_{l_p}(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}) > \boldsymbol{\phi}_{l_q}(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}).$ (22)

Seguidamente, se van a calcular condiciones LMI para las restricciones del valor de Shapley presentadas. Si se utiliza la matriz de transformación $\mathbf{M}$ introducida en la Definición 3, y se observa que, $\forall l \in \mathcal{E}$,

$$\sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} c|\Lambda| = c, \tag{23}$$

[2] La desigualdad contraria está trivialmente implícita, realizando un cambio de notación.

se puede expresar el valor de Shapley de cada enlace $l \in \mathcal{E}$ como

$$\phi_l(\mathcal{E}, \boldsymbol{r^v}) = c + \sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \left[\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}} \right]. \quad (24)$$

Por tanto, es posible reescribir (20) como

$$\sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \left[\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}} \right] < \mathcal{V}_l - c. \quad (25)$$

A continuación, si se representa (25) como $\begin{bmatrix} 1 & \mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \end{bmatrix} \mathbf{D_a} \begin{bmatrix} 1 \\ \mathbf{x}_{\mathcal{N}} \end{bmatrix} > 0$, se observa que, resolver (20) es equivalente a hallar una solución de

$$\mathbf{D_a} > 0, \quad \mathbf{D_a} = \begin{bmatrix} \mathcal{V}_l - c & 0 \\ 0 & -\sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}. \quad (26)$$

Si se repite el proceso con la restricción absoluta (21), se llega a la siguiente condición LMI

$$\mathbf{D_b} > 0, \quad \mathbf{D_b} = \begin{bmatrix} c - \mathcal{V}_l & 0 \\ 0 & \sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}. \quad (27)$$

Finalmente, para el caso de la restricción relativa (22) se tiene

$$\sum_{\Lambda \subseteq \mathcal{E}} m_{l_p\Lambda} \left[\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}} \right] > \sum_{\Lambda \subseteq \mathcal{E}} m_{l_q\Lambda} \left[\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}} \right], \quad (28)$$

que se verifica sí y solo sí

$$\mathbf{D_c} > 0, \quad \mathbf{D_c} = \sum_{\Lambda \subseteq \mathcal{E}} \left(m_{l_p\Lambda} - m_{l_q\Lambda} \right) \mathbf{P}_\Lambda. \quad (29)$$

Tras este proceso, se concluye que se han hallado 3 condiciones LMI diferentes, en función de la restricción del valor de Shapley considerada.

Comentario 5 *Nótese que las condiciones LMI planteadas proporcionan matrices de control que satisfacen las restricciones impuestas del valor de Shapley para todo el espacio de estados, i.e., se ha considerado $\mathcal{X}_i = \mathbb{R}^{n_{\mathbf{x}_i}}, \forall i \in \mathcal{N}$ en (2). De todas formas, es posible restringir dicho espacio de estados, mediante el uso de la herramienta de control conocida como s–procedure [3], en situaciones que impliquen resultados demasiado conservadores.*

5.3 Algoritmo

El problema que plantean las condiciones LMI obtenidas (26), (27) y (29), es que dependen de las matrices $\mathbf{P}_\Lambda$, que no son las mismas variables de decisión de las que dependía la condición LMI (17), necesaria para garantizar estabilidad y una cota superior de la función de coste, que en este caso eran las matrices $\mathbf{W}_\Lambda$ and $\mathbf{Y}_\Lambda$. Por tanto, no es posible utilizar todas las condiciones de forma conjunta en el mismo problema de diseño. Para solucionar este inconveniente, se propone el siguiente algoritmo:

1. Resolver un sistema de LMIs que satisfaga el conjunto de condiciones impuesto por (17), teniendo en cuenta a su vez lo reflejado en el Comentario 1. Como resultado es posible obtener unas matrices $\mathbf{K}_\Lambda$ y $\mathbf{P}_\Lambda^{(1)}$ que garanticen estabilidad en bucle cerrado y una cota superior del coste de control.
2. Utilizar el valor obtenido de $\mathbf{K}_\Lambda$ en el paso 1, para considerar (16) como condición LMI en la variable de decisión $\mathbf{P}_\Lambda$. Resolver un sistema de LMIs que satisfaga el conjunto de condiciones impuesto por (16) y, dependiendo de las restricciones del valor de Shapley consideradas, las condiciones (26), (27) y/o (29), con el propósito de obtener $\mathbf{P}_\Lambda^{(2)}$, cumpliendo las especificaciones.

La clave del algoritmo propuesto, es que al considerar como constante $\mathbf{K}_\Lambda$, se fuerza a que la dependencia de (16) con respecto a $\mathbf{P}_\Lambda$ sea afín y, por tanto, pueda utilizarse como condición LMI en dicha variable de decisión. Gracias a este método de cálculo, es posible computar juntas las condiciones LMI necesarias para cumplir todas las especificaciones del problema global.

Con objeto de comprobar que el valor de las matrices $\mathbf{P}_\Lambda^{(2)}$ no ha sufrido una degradación a causa de la aplicación del algoritmo, se introduce el siguiente parámetro.

Definición 5 *Se define el índice de suboptimalidad asociado a una matriz $\mathbf{P}_\Lambda$, para cada modo de red $\Lambda \subseteq \mathcal{E}$, como*

$$\eta(\mathbf{P}_\Lambda) = \left| \frac{\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}}}{\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda^* \mathbf{x}_{\mathcal{N}}} \right|, \quad (30)$$

donde

$$\mathbf{P}_\Lambda^* = \lim_{n \to \infty} \left(\sum_{j=0}^{n} \left(\mathbf{A}_{CL}^T \right)^j \left[\mathbf{Q}_{\mathcal{N}} + \mathbf{K}_\Lambda^T \mathbf{R}_{\mathcal{N}} \mathbf{K}_\Lambda \right] \mathbf{A}_{CL}^j \right), \quad (31)$$

obtenido a partir de (3) y (5), puede considerarse como el valor óptimo de la correspondiente matriz $\mathbf{P}_\Lambda, \forall \Lambda \subseteq \mathcal{E}$, siendo $\mathbf{A}_{CL} = \mathbf{A}_{\mathcal{N}} + \mathbf{B}_{\mathcal{N}} \mathbf{K}_\Lambda$ la matriz de control del bucle cerrado. Cuanto más cercano a la unidad sea el parámetro $\eta(\mathbf{P}_\Lambda)$, mejor estimación representará la correspondiente $\mathbf{P}_\Lambda$.

6 EJEMPLO DE SIMULACIÓN

En este apartado, se implementa una situación con cuatro agentes, $\mathcal{N} = \{1, 2, 3, 4\}$, y cuatro enlaces, $\mathcal{E} = \{I, II, III, IV\}$, con la configuración de la red de la Figura 1b. Las matrices que definen la dinámica del sistema son

$$A_{11} = \begin{bmatrix} 1 & 3.8 \\ 0 & -0.7 \end{bmatrix}, A_{22} = \begin{bmatrix} 1 & 0.8 \\ 0 & 3.8 \end{bmatrix},$$

$$A_{33} = \begin{bmatrix} 1 & 0.9 \\ 0 & -5.6 \end{bmatrix}, A_{44} = \begin{bmatrix} 1 & -2.8 \\ 0 & 0.5 \end{bmatrix}, B_{ii} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$A_{ij} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, B_{ij} = \begin{bmatrix} 0 \\ 0.15 \end{bmatrix}, i \neq j,$$

donde $\mathbf{x}_i \in \mathbb{R}^2$ and $\mathbf{u}_i \in \mathbb{R}$ son, respectivamente, los estados y la entrada de cada subsistema $i \in \mathcal{N}$. El coste de control J_s de todos los subsistemas se define mediante las matrices $\mathbf{Q} = \mathbf{I} \in \mathbb{R}^{8\times8}$ y $\mathbf{R} = \mathbf{I} \in \mathbb{R}^{4\times4}$. Se supondrá un coste común a todos los enlaces $c = 0.5$.

Se consideran las siguientes restricciones del valor de Shapley en el problema global

$$\begin{gathered}\phi_{III}(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}) > \phi_I(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}) > \phi_{II}(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}) > \phi_{IV}(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}), \\ \phi_{III}(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}) > -1, \quad \phi_{IV}(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}}) < 1.\end{gathered} \tag{32}$$

Para cada uno de los modos de red Λ y enlaces l, se ha utilizado el algoritmo descrito en el Subapartado 5.3 para resolver, mediante el uso del paquete Matlab® LMI Control Toolbox [4], un sistema LMI en el que se han utilizado las condiciones (16), (17), (26), (27) y (29), con el propósito de satisfacer (32), asegurando estabilidad del sistema global y una cota superior del coste. Como resultado se han obtenido las matrices $\mathbf{K}_\Lambda$, $\mathbf{P}_\Lambda^{(1)}$ y $\mathbf{P}_\Lambda^{(2)}$. Para este caso numérico concreto, las matrices obtenidas para el modo de red Λ_4 son

$$\mathbf{K}_{\Lambda_4} = \begin{bmatrix} -0.24 & -0.25 & 0 & 0 & 0 & 0 & -0.04 & 0.17 \\ 0 & 0 & -0.24 & -1.56 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.17 & 3.20 & 0 & 0 \\ 0.03 & 0.01 & 0 & 0 & 0 & 0 & 0.28 & -1.24 \end{bmatrix},$$

$$\mathbf{P}_{\Lambda_4}^{(1)} = \begin{bmatrix} 2.27 & 4.49 & 0 & 0 & 0 & 0 & -0.02 & 0.10 \\ 4.49 & 17.75 & 0 & 0 & 0 & 0 & 0.03 & -0.01 \\ 0 & 0 & 4.30 & 3.93 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.93 & 12.22 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.86 & 6.20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6.20 & 35.50 & 0 & 0 \\ -0.02 & 0.03 & 0 & 0 & 0 & 0 & 2.32 & -3.94 \\ 0.10 & -0.01 & 0 & 0 & 0 & 0 & -3.94 & 13.19 \end{bmatrix},$$

$$\mathbf{P}_{\Lambda_4}^{(2)} = \begin{bmatrix} 2.25 & 4.42 & 0 & 0 & 0 & 0 & -0.02 & 0.13 \\ 4.42 & 17.46 & 0 & 0 & 0 & 0 & -0.01 & 0.10 \\ 0 & 0 & 4.27 & 3.85 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.85 & 12.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.85 & 6.11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6.11 & 35.02 & 0 & 0 \\ -0.02 & -0.01 & 0 & 0 & 0 & 0 & 2.29 & -3.87 \\ 0.13 & 0.10 & 0 & 0 & 0 & 0 & -3.87 & 12.98 \end{bmatrix}.$$

Nótese, que de acuerdo con el Comentario 1, las matrices de control satisfacen las restricciones de comunicación impuestas por el modo de red.

Una vez resuelto el problema LMI, si se toma como estado inicial

$$x_1 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, x_2 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, x_3 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, x_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

y se escoge como *modo de red de trabajo* al modo de red óptimo (el que minimiza (7)), recalculado cada tres pasos de control, se obtiene la evolución de los valores de Shapley y del modo de red de trabajo que se observa en la Figura 2, sin considerar las restricciones de (32) (Fig. 2a), asumiendo únicamente las restricciones relativas del valor de Shapley (Fig. 2b), y teniendo en cuenta todas las restricciones de (32) (Fig. 2c). De dichas figuras, podemos extraer los siguientes resultados:

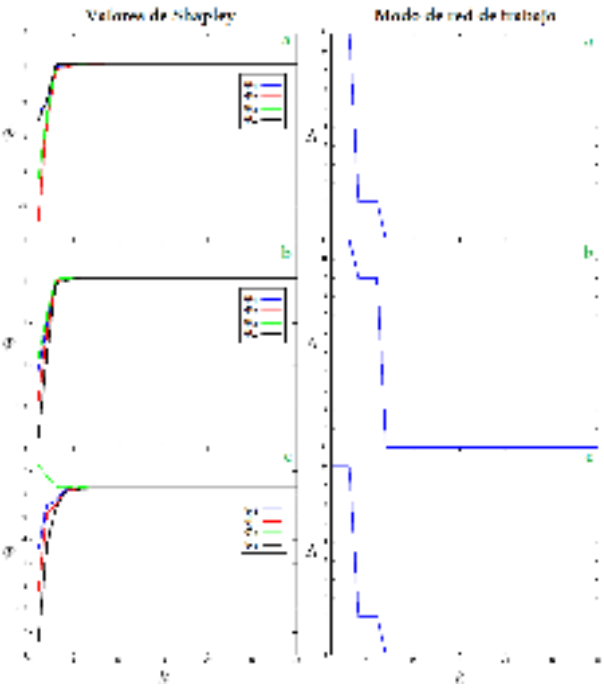


Figura 2: Evolución del valor de Shapley y del modo de red de trabajo para distintos supuestos.

- El valor de Shapley satisface todas las especificaciones (32), en cada paso de control k, cumpliéndose en todos los casos $\eta(\mathbf{P}_\Lambda^{(2)}) \in (1, 1.05)$, lo que muestra la validez del algoritmo.

- El modo de red de trabajo varía dependiendo de las restricciones consideradas, pero en régimen permanente siempre tiende al modo de red descentralizado, lo que es lógico pues es la configuración de reposo una vez alcanzado el origen.

- El valor de Shapley en régimen permanente tiende al coste del enlace $c = 0.5$, $\forall l \in \mathcal{E}$, cuando el sistema global se regula al origen, como se puede extraer de (24). Por tanto, el régimen permanente del valor de Shapley no depende de las restricciones impuestas sobre el mismo.

Como se ha mostrado en este apartado, se puede concluir que los resultados numéricos prueban la viabilidad del método de diseño considerado.

7 CONCLUSIONES

En este artículo, se han extendido trabajos previos para continuar con la línea de integración de teoría de juegos cooperativos y control distribuido. Se ha presentado una novedosa notación matricial para el valor de Shapley y el valor de posición, que simplifica su introducción en un contexto de control, y permite la inclusión de restricciones en el valor de Shapley en la fase de diseño. Adicionalmente, se ha introducido un método de diseño basado en condiciones LMI que nos proporciona un controlador adaptado a la topología de red, estable en bucle cerrado, y que además satisface las restricciones en el valor de Shapley antes mencionadas. Finalmente, el ejemplo planteado y resuelto aplicando el paquete de Matlab® LMI Control Toolbox, ilustra la viabilidad del esquema propuesto. Trabajos futuros podrían incluir un análisis más formal de cómo las restricciones impuestas en el valor de Shapley afectan a la activación o desactivación de un enlace concreto.

Agradecimientos

Se agradece la financiación parcial de este trabajo a los proyectos HYCON2 (FP7-257462) y Control Predictivo en Red (DPI2008-05818).

Referencias

[1] Álamo, T., Normey–Rico, J. E., Arahal M. R., Limón, D., y Camacho, E. F., "Introducing lineal matrix inequalities in a control course," en los *Proceedings del 7º IFAC Symposium on Advances in Control Education*, Madrid, Junio 2006.

[2] Borm, P., Owen, G., y Tijs, S., "On the position value for communication situations," *SIAM Journal on Discrete Mathematics*, vol. 5, no. 3, pp. 305–320, Agosto 1992.

[3] Boyd, S., El Ghaoui, L., Feron, E., y Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, Philadelphia, Pennsylvania, EEUU: Society for Industrial and Applied Mathematics, 1994.

[4] Gahinet, P., Nemirovskii, A. S., Laub, A. J., y Chilali, M., *LMI Control Toolbox for use with MATLAB®*, The MathWorks Inc., 1995.

[5] Kim, W. J., Ji, K., y Srivastava, A., "Network–based control with real–time prediction of delayed/lost sensor data," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 1, pp. 182–185, Enero 2006.

[6] Maestre, J. M., Muñoz de la Peña, D., Camacho, E. F., y Álamo, T., "Distributed model predictive control based on agent negotiation," *Journal of Process Control*, vol. 21, no. 5, pp. 685–697, 2011.

[7] Maestre, J. M., Muñoz de la Peña, D., Jiménez Losada, A., Algaba, E., y Camacho, E. F., "A coalitional control scheme with applications to cooperative game theory," en los *Proceedings del 18º IFAC World Congress*, Milán, Italia, Septiembre 2011, pp. 9121–9126.

[8] Myerson, R. B., "Graphs and cooperation in games," *Mathematics of Operations Research*, vol. 2, no. 3, pp. 225–229, Agosto 1977.

[9] Negenborn, R. R., De Schutter, B., y Hellendoorn, J., "Multi–agent model predictive control of transportation networks," en los *Proceedings de la 2006 IEEE International Conference on Networking, Sensing and Control (ICNSC 2006)*, Ft. Lauderdale, Florida, EEUU, Abril 2006, pp. 296–301.

[10] Nesterov, Y., y Nemirovskii, A. S., *Interior–Point Polynomial Algorithms in Convex Programming*, Philadelphia, Pennsylvania, EEUU: Society for Industrial and Applied Mathematics, 1994.

[11] Scattolini, R., "Architectures for distributed and hierarchical model predictive control – a review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, Mayo 2009.

[12] Shapley, L. S., "A value for n–person games," *Contributions to the Theory of Games II. Editado por Kuhn, H. W., y Tucker, A. W. Annals of Mathematics Studies*, vol. 28, pp. 307–317, 1953.

[13] Slikker, M., "A characterization of the position value," *International Journal of Game Theory*, vol. 33, no. 4, pp. 505–514, Julio 2005.

[14] Trodden, P., y Richards, A. G., "Adaptive cooperation in robust distributed model predictive control," en los *Proceedings del 24º IEEE International Symposium on Intelligent Control*, San Petersburgo, Rusia, Julio 2009, pp. 896–901.

[15] Von Neumann, J., y Morgenstern, O., *Theory of Games and Economic Behaviour*. Princeton, Nueva Jersey, EEUU. 3ª edición: Princeton University Press, 1944.

[16] Zhang, F., *The Schur Complement and Its Applications*, Nueva York, EEUU: Numerical Algorithm Series, Springer Verlag, 2005.

ADMINISTRACIÓN DE RECURSOS REFORZADA MEDIANTE SUPERVISIÓN DE ELEMENTOS FINALES DE CONTROL

Ramón Ferreiro García
Dpto. de Ingeniería Industrial, Universidad de La Coruña
ferreiro@udc.es

José Luis Calvo-Rolle
Dpto. de Ingeniería Industrial, Universidad de La Coruña
jlcalvo@udc.es

José Luis Casteleiro-Roca
Dpto. de Ingeniería Industrial, Universidad de La Coruña
jose.luis.casteleiro@udc.es

Manuel Romero Gómez
Dpto. de Energía y Propulsión, Universidad de La Coruña
m.romero.gomez@udc.es

Alberto Demiguel Catoria
Dpto. de Energía y Propulsión, Universidad de La Coruña
alberto.demiguel@udc.es

Resumen

La gran demanda de información originada por la necesidad de incrementar la eficiencia competitiva, da lugar a la aplicación masiva de instrumentación de campo, lo que hace difícil y tediosa las tareas de mantenimiento de dispositivos integrados a los equipos y procesos controlados automáticamente en todos los procesos asociados con la producción industrial. Para hacer viable el mantenimiento de la inmensa cantidad de instrumentos que constituyen una planta, con el objetivo de minimizar las paradas no programadas y alargar los ciclos de vida útil tanto de los equipos como de su instrumentación, se hace imprescindible la integración de herramientas capaces de administrar con efectividad competitiva tales recursos. En este artículo se propone y analiza una técnica de administración de recursos en base a la supervisión de elementos finales de control (válvulas) por medio del análisis híbrido (analítico, el basado en aproximación funcional y gestionado en base a reglas) de su comportamiento dinámico para detectar fallos en válvulas de control, reforzando la herramienta destinada al mantenimiento de dispositivos de campo denominada AMS (Asset Management Services)

Palabras Clave: Detección de fallos, Aislamiento de fallos, Administración de recursos, Asset management services, Identificación de parámetros, Supervisión.

1 INTRODUCCIÓN

Entre las tareas de mantenimiento y administración de recursos, AMS concierne al estado de la instrumentación y del hardware de la red [11-15]. La administración de recursos tiene como objetivo fundamental la explotación eficiente de la red y sus dispositivos. Independientemente del esquema de mantenimiento aplicado a la planta y sus procesos, se puede utilizar la tecnología Fieldbus [1, 2] para obtener información acerca de los dispositivos de campo, los cuales simplificarán el mantenimiento haciéndolo mas eficiente. La administración de recursos sobre Fieldbus conduce a la programación de mantenimiento proactivo.

El mantenimiento reactivo consiste en un esquema de organización en el cual se actúa sobre algún dispositivo después de observar su inoperancia. Se da la circunstancia de que uno o más dispositivos pueden operar con deficiencias no relevantes sin observar su inoperancia. Esta clase de mantenimiento es poco seguro, peligroso y poco eficiente desde el punto de vista de que una planta no puede operar eficientemente si algún dispositivo no lo hace.

El mantenimiento preventivo consiste en una técnica de revisión sistemática a intervalos determinados de tiempo aún sin que sea necesaria la revisión de algún instrumento. Con este método de mantenimiento se logra seguridad y disponibilidad partiendo de que se evitan las paradas no esperadas.

El mantenimiento predictivo consiste en una técnica de revisión sistemática a intervalos determinados de tiempo aún sin que sea necesaria la revisión de algún instrumento. Sin embargo, el intervalo de revisión está optimizado en base al análisis estadístico de resultados históricos en base a la fiabilidad. Existe pérdida de recursos pero representa ciertas mejoras respecto al mantenimiento predictivo.

El mantenimiento proactivo consiste en un esquema de organización en el cual la revisión de dispositivos es realizada en cada dispositivo en el instante preciso, en base a la condición de operación de cada instrumento o dispositivo. No se asignan recursos a dispositivos que no necesitan mantenimiento y que supongan costes significativos. Este método requiere los recursos mínimos, lo que significa mínimo coste posible y demanda de mano de obra mínima.

La secuencia de sub-tareas a llevar a cabo para asegurar la correcta operación de los procesos constituyen el núcleo de la tarea de supervisión, conocida como monitorización incluyendo diagnosis de fallos (detección e identificación de fallos), reparación, recuperación o reconfiguración mediante intervención [18]. La monitorización de procesos está basada en procedimientos según se muestra en [3, 4]. Con respecto a los métodos analíticos, éstos utilizan los métodos inherentes a los residuos en espacios de paridad y redundancia [9, 10]. Los residuos suponen el resultado de pruebas de consistencia entre las observaciones en tiempo real de la planta o proceso y sus modelos analíticos. Se caracterizan por un valor suficientemente alto para hacer visible los fallos o inconsistencias y suficientemente pequeños en presencia de perturbaciones, ruidos de medida o errores de modelado. Destacan tres métodos utilizados para generar residuos: estimación de parámetros, observadores y relaciones de paridad.

En el caso de estimación de parámetros, los residuos suponen la diferencia entre las salidas proporcionadas con valores nominales del modelo y valores reales [8]. Los desvíos de los parámetros del modelo conforman la base de detección y aislamiento de fallos [16, 17].

En los métodos basados en observadores la salida del sistema es reconstruida a partir de las medidas con la ayuda de observadores. Los residuos se definen como la diferencia entre la salida medida y la estimada.

Las estrategias basadas en relaciones de paridad verifican la consistencia del modelo matemático con medidas en tiempo real. Tales métodos analíticos son efectivos cuando están libres de error.

La tarea de refuerzo propuesta consiste en diagnosticar en línea los elementos finales de control en base al análisis híbrido que consiste en la asociación de la modelización analítica de los elementos a diagnosticar mediante la técnica de identificación de parámetros descrita en [5], y aproximación funcional por técnicas convencionales tales como tablas, regresión polinómica multivariable o redes de neuronas para conseguir la detección y asilamiento de defectos de operación así como la organización de su intervención [6, 7].

La administración de recursos concierne a los dispositivos. Por esta razón, el bloque de recursos y transductor de cada dispositivo tienen que ser accedidos mediante la correspondiente aplicación software.

La figura 1 muestra el esquema que ilustra un entorno propio de la arquitectura típica exhibiendo el principio de utilización del software de mantenimiento AMS. Esta técnica de mantenimiento opera en paralelo simultáneamente con el SCADA, y el software que la soporta es susceptible de ser alojado en el mismo computador o host que hace las funciones de interfaz del usuario (HMI).

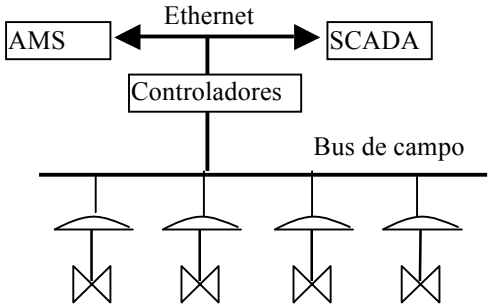


Figura 1: Esquema de principio de operación del software de administración de recursos

2 PROCEDIMIENTO HÍBRIDO DE MODELIZACIÓN

El sistema híbrido de modelización de los elementos finales de control consta de:

a) Modelo analítico. El modelo analítico que define la dinámica de cada elemento final de control basado en una válvula, puede ser definido como se muestra en la ecuación 1.

$$f(t) = C_N \frac{d^N X}{dt^N} + C_{N-1} \frac{d^{N-1} X}{dt^{N-1}} + \cdots + C_1 \frac{dX}{dt} + C_0 X \qquad (1)$$

Donde C_i (i = 0, 1, 2, … N), son los parámetros asociados a la ecuación diferencial (ED) que define la dinámica de la válvula. N es el orden de la ED y X es la salida o posición.

b) Modelos basados en aproximación funcional [6, 7]. Que a su vez se dividen en dos: Fuerzas de excitación (ecuación 2), y variable manipulada (ecuación 3).

$$f(t) = f(U, p_U)) \tag{2}$$

$$\dot{m} = f(X, (p_X)) \tag{3}$$

Donde *f(t)* es la fuerza neta de excitación; la fuerza que corresponde a la diferencia entre la fuerza realizada por el actuador y la fuerza inversa ejercida por la contrapresión de la variable manipulada. Esta fuerza puede ser despreciada en muchos casos al ser de bajo valor con respecto a la fuerza ejercida por el actuador. *U* es la variable de control, $\dot{m}$ es el caudal o variable manipulada, *X* es la posición del vástago de la válvula, y p_u y p_x son los parámetros asociados con la variable de control y la variable manipulada respectivamente.

2.1 PROCESO DE IDENTIFICACIÓN DE LOS PARÁMETROS ASOCIADOS CON EL MODELO ANALÍTICO

El procedimiento propuesto en [5] consiste en realizar un número de medidas en línea en los instantes *t1*, *t2*, *t3*, … *tn*, de las sucesivas derivadas del modelo analítico igual al orden *N* de la ED que define el modelo de cada elemento final de control, resultando las ecuaciones mostradas en 4.

$$\begin{aligned}
f(t1) &= C_N \frac{d^N X(t1)}{dt^N} + C_{N-1} \frac{d^{N-1} X(t1)}{dt^{N-1}} + \cdots + C_1 \frac{dX(t1)}{dt} + C_0 X(t1) \\
f(t2) &= C_N \frac{d^N X(t2)}{dt^N} + C_{N-1} \frac{d^{N-1} X(t2)}{dt^{N-1}} + \cdots + C_1 \frac{dX(t2)}{dt} + C_0 X(t2) \\
f(t3) &= C_N \frac{d^N X(t3)}{dt^N} + C_{N-1} \frac{d^{N-1} X(t3)}{dt^{N-1}} + \cdots + C_1 \frac{dX(t3)}{dt} + C_0 X(t3) \\
&\vdots \\
f(tN) &= C_N \frac{d^N X(tN)}{dt^N} + C_{N-1} \frac{d^{N-1} X(tN)}{dt^{N-1}} + \cdots + C_1 \frac{dX(tN)}{dt} + C_0 X(tN)
\end{aligned} \tag{4}$$

Que en forma matricial se expresarían según la expresión 5.

$$\begin{bmatrix} f(t1) \\ f(t2) \\ f(t3) \\ \vdots \\ f(tn) \end{bmatrix} = \begin{bmatrix} C_N \\ C_{N-1} \\ C_{N-2} \\ \vdots \\ C \end{bmatrix} \begin{bmatrix} \frac{d^N X(t1)}{dt^N} & \frac{d^{N-1} X(t1)}{dt^{N-1}} \cdots \frac{dX(t1)}{dt} & X(t1) \\ \frac{d^N X(t2)}{dt^N} & \frac{d^{N-1} X(t2)}{dt^{N-1}} \cdots \frac{dX(t2)}{dt} & X(t2) \\ \frac{d^N X(t3)}{dt^N} & \frac{d^{N-1} X(t3)}{dt^{N-1}} \cdots \frac{dX(t3)}{dt} & X(t3) \\ \vdots & & \\ \frac{d^N X(tN)}{dt^N} & \frac{d^{N-1} X(tN)}{dt^{N-1}} \cdots \frac{dX(tN)}{dt} & X(tN) \end{bmatrix} \tag{5}$$

Que de forma compacta quedaría como aparece en 6.

$$|f(ti)| = [\mathrm{C}][X] \tag{6}$$

La solución del sistema de ecuaciones que estima los parámetros del modelo obtenido en línea es de la forma mostrada en 7.

$$\begin{bmatrix} C_N \\ C_{N-1} \\ C_{N-2} \\ \vdots \\ C \end{bmatrix} = \begin{bmatrix} \frac{d^N X(t1)}{dt^N} & \frac{d^{N-1} X(t1)}{dt^{N-1}} \cdots \frac{dX(t1)}{dt} & X(t1) \\ \frac{d^N X(t2)}{dt^N} & \frac{d^{N-1} X(t2)}{dt^{N-1}} \cdots \frac{dX(t2)}{dt} & X(t2) \\ \frac{d^N X(t3)}{dt^N} & \frac{d^{N-1} X(t3)}{dt^{N-1}} \cdots \frac{dX(t3)}{dt} & X(t3) \\ \vdots & & \\ \frac{d^N X(tN)}{dt^N} & \frac{d^{N-1} X(tN)}{dt^{N-1}} \cdots \frac{dX(tN)}{dt} & X(tN) \end{bmatrix}^{-1} \begin{bmatrix} f(t1) \\ f(t2) \\ f(t3) \\ \vdots \\ f(tn) \end{bmatrix} \tag{7}$$

Que en su forma general quedaría como 8.

$$|\hat{C}| = [X]^{-1}[f(ti)] \tag{8}$$

2.2 PROCESO DE IDENTIFICACIÓN APLICADO A UNA VÁLVULA DE CONTROL FISHER TIPO DVC 5040F

La servo-válvula ensayada DVC 5040F acciona una válvula PN40, la cual dependiendo del gradiente de presión que experimenta el fluido a través de la válvula requiere la correspondiente acción de la fuerza realizada por el servo-actuador. Los parámetros nominales *M*, *B* y *K* son respectivamente, 3.5 kg, $2 \cdot 10^6$ N/(m/s) y 10^7 N/m.

El procedimiento propuesto exige el conocimiento determinista de la estructura del modelo matemático como condición preliminar. Así, aplicando un balance de fuerzas a la válvula considerada, de las ecuaciones 1, 4 y 6 resulta la expresión 9.

$$\begin{vmatrix} F(t1) \\ F(t2) \\ F(t3) \end{vmatrix} = \begin{bmatrix} \ddot{Y}(t1) & \dot{Y}(t1) & Y(t1) \\ \ddot{Y}(t2) & \dot{Y}(t2) & Y(t2) \\ \ddot{Y}(t3) & \dot{Y}(t3) & Y(t3) \end{bmatrix} \begin{vmatrix} M \\ B \\ K \end{vmatrix} \tag{9}$$

Los parámetros de la válvula son obtenidos por medio de la solución de la ecuación 9 según se muestra en 10.

$$\begin{vmatrix} M \\ B \\ K \end{vmatrix} = \begin{bmatrix} \ddot{Y}(t1) & \dot{Y}(t1) & Y(t1) \\ \ddot{Y}(t2) & \dot{Y}(t2) & Y(t2) \\ \ddot{Y}(t3) & \dot{Y}(t3) & Y(t3) \end{bmatrix}^{-1} \begin{vmatrix} F(t1) \\ F(t2) \\ F(t3) \end{vmatrix} \tag{10}$$

En donde las sucesivas derivadas son obtenidas en modo discreto utilizando diferencias finitas:

- Primera derivada (ecuación 11).

$$\dot{Y}(t1) = \frac{Y(t2) - Y(t1)}{t2 - t1} \tag{11}$$

- Segunda derivada (ecuación 12).

$$\ddot{Y}(t1) = \frac{\dot{Y}(t2) - \dot{Y}(t1)}{t2 - t1} \tag{12}$$

2.3 DETECCIÓN Y AISLAMIENTO DE FALLOS DE LA VÁLVULA DE CONTROL FISHER TIPO DVC 5040F

La tarea de detección es llevada cabo mediante la verificación sistemática de la variación de parámetros efectuada en línea, comparando los parámetros obtenidos en la expresión 10 con los valores nominales de los parámetros, según la ecuación 13.

$$\begin{vmatrix} \Delta M \\ \Delta B \\ \Delta K \end{vmatrix} = \begin{vmatrix} \Delta M_N \\ \Delta B_N \\ \Delta K_N \end{vmatrix} - \begin{vmatrix} M \\ B \\ K \end{vmatrix} \tag{13}$$

La detección y el aislamiento de fallos están asociados por la aplicación de las reglas inherentes a la expresión 13. En 14 se determina de modo determinista qué parámetro de la válvula ha de ser cambiado entre todos los demás mediante la consideración lógica de la asociación de los a través de la función lógica "OR".

$$\begin{aligned} &IF\ |\Delta M| > |\Delta M_L| \Rightarrow fallo\ en\ |M| \\ &IF\ |\Delta B| > |\Delta B_L| \Rightarrow fallo\ en\ |B| \\ &IF\ |\Delta K| > |\Delta K_L| \Rightarrow fallo\ en\ |K| \end{aligned} \tag{14}$$

Donde:

$$\begin{vmatrix} \Delta M_L \\ \Delta B_L \\ \Delta K_L \end{vmatrix} = \begin{vmatrix} M_N \\ B_N \\ K_N \end{vmatrix} \pm \begin{vmatrix} 0.05 \\ 0.1 \\ 0.1 \end{vmatrix} \begin{vmatrix} M_N \\ B_N \\ K_N \end{vmatrix} \tag{15}$$

El valor de los coeficientes atribuidos a la ecuación 15 se deben al margen de desvío exclusivamente aplicable para esta válvula de acuerdo con los requerimientos del usuario. Concretamente equivale al 5%, 10%, 10% respectivamente del valor total de cada parámetro.

Resulta evidente que la variación de la masa de los elementos móviles de una válvula de control, solamente puede ser debida al desprendimiento de la válvula del vástago (caída de válvula) y el correspondiente servo-actuador, y tal fallo se puede detectar por la simple observación de la variable manipulada. Sin embargo el hecho de detectar la variación de su masa permite determinar exactamente la procedencia del fallo. La detección de variaciones en el parámetro que define la fricción está asociada con problemas del sellado del vástago de la válvula, principalmente si el cierre es de tipo prensa-estopas dotado de empaquetaduras. Este tipo de problema suele requerir la sustitución de las empaquetaduras.

3 RESULTADOS

La verificación del método de diagnosis utilizando un SCADA desarrollado con la aplicación DeltaV de Emerson Process Management sobre una válvula Fisher de Rosemount, ha permitido la realización de los ajustes necesarios en la estrategia de acoplar la detección de fallos en válvulas de control al sistema de administración de recursos de DeltaV. De esta forma se ha conseguido un diagnóstico más completo y fiable que el habitual, al disponer de información adicional relevante de los elementos finales de control para cada lazo, entre la que cabe destacar el efecto de agarrotamiento por las variaciones de fricción viscosa, así como la variación de la rigidez mecánica de los resortes como consecuencia de la fatiga generada entre otros problemas comunes. Finalmente la detección de variaciones en la rigidez elástica del resorte es debida a la fatiga acumulada por exceso de tiempo en estado de compresión, lo que determina la necesidad de sustituir el resorte.

Otros fallos comunes como la detección de desvío del valor de la variable manipulada en comparación con la función que define su valor nominal en función de la variable de control según la técnica clásica de aproximación funcional implementada opcionalmente mediante tablas, regresión polinómica por mínimos cuadrados, redes neuronales, otras técnicas o simplemente mediante la ecuación 3, pueden ser debidos (modo no determinista) al deterioro de los componentes asociados con el asiento de la válvula, al posicionador de la válvula o al cambio de las condiciones (presión de entrada, salida o ambas) de la propia variable manipulada. Partiendo de que con cualquiera de estas técnicas es imposible la obtención de soluciones con el requerido determinismo, se hace necesaria la aplicación de redundancia analítica y/o física para ganar fiabilidad en el diagnóstico asociado con esta parte del problema de supervisión.

4 CONCLUSIONES

La estrategia de ayuda al diagnóstico ha sido implementada sobre una servo-válvula de control. Se demuestra que se puede mejorar la tarea de supervisión mediante el acoplamiento de la aplicación propuesta con el AMS destinado a la administración de recursos. De esta forma se ha

logrado reforzar una de las herramientas de mantenimiento de dispositivos de campo aplicadas en plantas y procesos del sector productivo industrial de forma fiable y con un coste efectivo, pero la estrategia se hace extensiva a los demás sistemas destinados al mantenimiento de dispositivos de campo. La ventaja esencial de tal estrategia radica en que incrementa el determinismo por reducción de ambigüedad en la localización de fallos de elementos finales de control, lo que redunda en la reducción de paradas de emergencia o paradas no programadas, con lo que repercute en la eficiencia productiva.

Referencias

[1] ANSI/ISA-51.1-1979 (1993). "Process Instrumentation Technology", Research Triangle Park, NC:ISA, ISBN 0-87664-390-4

[2] Berge, J. (2002). "Fieldbus for process Control: Engineering Operation and Maintenance", ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, USA, ISA, pp 359-377

[3] Blanke M., Kinnaert M.,Lunze J., Staroswiecki (2003). "Diagnosis and Fault-Tolerant Control". Springer-Verlag, Berlin, Heidelberg, pp. 469-496.

[4] Clark, R.N. (1978). "A simplified instrument detection scheme", *IEEE Trans. Aerospace Electron. Syst*., Vo1. 14, pp. 558-563

[5] Ferreiro Garcia, R., Vidal Paz, J. (1994). "Learning task applied to identification of a marine vehicle", *Annual Review in Automatic Programming*, Vol. 19, pp. 197-201

[6] Ferreiro Garcia, R. (2012). "Improving heat exchanger supervision using neural networkd and rule based techniques", *Expert Systems with Applications*, Vol. 39, pp. 3012-3021

[7] Ferreiro Garcia, R. (2013), "Expert condition monitoring on hydrostatic self-levitating bearings", *Expert Systems with Applications*, Vol. 40, pp. 2975-2984

[8] Frank, P.M. (1987). "Fault Diagnosis in Dynamic systems via State Estimation, A Survey", "*In System Fault Diagnostics, Reliability and Related Knowledge-Based Approaches*, Tzafestas et al. (eds.), D. Reidel Publíshing Company, Dordrecht, Vol. 1, pp. 35-98

[9] Gertler, J.J. (1988). "Survey of Model-Based Failure Detection and Isolation in Complex Plants", *IEEE Control Systems Magazine*, Vol. 8 (6), pp. 3-11

[10] Gertler, J.J. (1991). "Analytical Redundancy Methods in Fault Detection and Isolation - Survey and Synthesis", *Preprints of the IFAC/IMACS-Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS '91*, Sept. 10-13, Baden-Baden, FRG, Vol. 1, pp. 9-21

[11] Model 3051S FOUNDATION™ Fieldbus Supplement, 00809-0200-4801, Rev AA, April 2002. Reference manual, Model 3051S Series, Pressure Transmitter Family, with FOUNDATION™ Fieldbus protocol, Emerson Process Management. www.rosemount.com

[12] Model 3244MV MultiVariable Temperature Transmitter with FOUNDATION™ Fieldbus, 00809-0100-4769, Rev. BA, 07/00, Emerson Process Management. www.rosemount.com

[13] Model 8742C. Magnetic Flowmeter Transmitter with FOUNDATION™ Fieldbus, 00809-0100-4793, Rev. AA, 8/99. Emerson Process Management. www.rosemount.com

[14] Model 8800C and Model 8800A Vortex Flowmeter with FOUNDATION™ Fieldbus, 00809-0100-4772, Rev. DA, 1/00. Emerson Process Management. www.rosemount.com

[15] Operation & Maintenance Instruction manual. Fieldbus Valve Positioner. Feb/03. FY302. www.smar.com

[16] Patton, R.J. and Chen, J. (1991). "A review of parity space approaches to fault diagnosis", *IFAC Symposium SAFEPROCE5S'91*, Baden-Baden, Germany, Preprints Vol. 1, pp. 239-256

[17] Patton, R.J. and Chen, J. (1994). "A review of parity space approaches to fault diagnosis for aerospace systems", *Journal of Guidance Control Dynamics*, Vol. 17 (2), pp. 278-285

[18] Willsky, A.S. (1976). "A survey of design methods for failure detection systems", *Automatica*, Vol. 12, pp. 601-611

Control No Lineal Iterativo de Modelos de Maniobra de Vehículos Marinos

Elías Revestido [(a)], M. Tomás-Rodríguez [(b)], Francisco J. Velasco [(a)].
[(a)] Dep. Tecnología Electrónica e Ingeniería de Sistemas y Automática,
Universidad de Cantabria, Spain, e-mail: revestidoe@unican.es
[(b)] School of Engineering and Mathematical Sciences, The City University London, United Kingdom.

Resumen

En este trabajo se aborda el diseño de control para un modelo no lineal de maniobra de 4 GDL. El algoritmo propuesto consiste en la combinación de una técnica de iteración que aproxima el modelo original por una secuencia de sistemas lineales y variantes en el tiempo (LVT) y del diseño de una red de compensación de adelanto para cada uno de los sistemas LVT, el controlador se optimiza en cada instante de tiempo del intervalo. El controlador diseñado para la última iteración se aplica al problema no lineal original. Los resultados de las simulaciones, muestran la exactitud de esta metodología para determinadas maniobras con el controlador de adelanto diseñado. Las principales características obtenidas por el controlador son la reducción del tiempo de establecimiento, la eliminación del sobreimpulso y el error en estado estacionario.

Palabras clave: Ingeniería de control, sistemas no lineales, piloto automático, compensación de adelanto, mantenimiento de rumbo.

1 Introducción

El diseño de pilotos automáticos basados en controladores PID comenzó en los años 20 [12] con la ayuda de un girocompás como elemento de medición, en el lazo de control. Los principales retos en el diseño de pilotos automáticos de buques son las incertidumbres ambientales (olas, viento, corrientes oceánicas) y la dinámica no lineal del buque. Además, la dinámica del timón presenta no linealidades de tipo saturación en el ángulo y velocidad de cambio del timón. En la literatura se encuentran artículos que tratan el diseño y la implementación de pilotos automáticos basados en controladores PID, en el que se realizan linealizaciones del modelo de maniobra del buque, ([12, 15, 16, 20, 21, 27]). En la mayoría de aplicaciones de baja velocidad es aceptable considerar únicamente la dinámica lineal del modelo de maniobra del buque. Sin embargo, para aplicaciones de alta velocidad, giros cerrados, grandes ángulos de deslizamiento lateral o en presencia de corrientes, los efectos no lineales se acentúan considerablemente hasta tal punto de degradar el funcionamiento del controlador. En la literatura, se han utilizado diferentes métodos no lineales [12] para el mantenimiento de rumbo con piloto automático: linealización por realimentación de estado [11], backstepping no lineal [2, 35], control por modo deslizante [18], retroalimentación de salidas [19], control H_∞ [14], enjambre de partículas de optimización [29], algoritmos genéticos [18], métodos de lógica difusa [5], ...etc. Para la mayoría de este tipo de aplicaciones, se consideran modelos de maniobra no lineales de un grado de libertad (GDL), p. ej. [22] o [3], donde el acoplamiento entre las variables no se tiene en cuenta. Debido a la complejidad de algunos de los métodos no lineales citados anteriormente, y a que la implementación puede ser costosa desde el punto de vista computacional, el objetivo que se ha establecido en este trabajo es el de diseñar un método de control para modelo de maniobra no lineal de un buque sin llevar a cabo la simplificación de las no linealidades o acoplamientos del modelo. Proponemos una estrategia de control basada en compensación de avance optimizado combinado con una técnica de iteración que aproxima el sistema no lineal original. Esta técnica de iteración se presentó inicialmente en [30, 33] y se ha utilizado para resolver diversos problemas de control no lineal: control óptimo [32], seguimiento óptimo no lineal [8], asignación de polos en sistemas no lineales [31]...etc. Una de sus ventajas es la de mantener las características no lineales inherentes del sistema, facilitando la implementación de control robusto, eliminando las incertidumbres de modelado. La técnica de iteración se aplica a un modelo de maniobra no lineal de 4 GDL de un buque. Esto permite la novedosa posibilidad de diseñar un control de rumbo con piloto automático basándose en la metodología de compensación de avance aplicado a un modelo no lineal. Este enfoque, se desarrolla sin las limitaciones de los modelos lineales indicados anteriormente, y mantiene la sencillez en el diseño y la implementación de compensación de avance. En base a un estudio preliminar, se ha propuesto la aplicación de un controlador de compensación de avance en lugar

de un PID convencional. Mediante una técnica de optimización, se ha obtenido un compromiso entre el máximo sobreimpulso y el tiempo de respuesta sin error de estado estacionario. El objetivo es diseñar un controlador para sistemas no lineales de la forma:

$$\dot{x} = A(x)x(t) + B(x)u_c(t,\theta_c), \quad x(0) = x_0 \quad (1)$$

$u_c(t,\theta_c)$ es el control, θ_c son los parámetros del controlador, $x(t)$ es el vector de estado, $A(x)$, $B(x)$ son las matrices de dimensiones apropiadas y $x(0)$ las condiciones iniciales. Sustituyendo el sistema no lineal por una secuencia de i sistemas lineales variantes en el tiempo (LVT), se genera la secuencia correspondiente de leyes de control $u_c^{(i)}(t,\theta_c)$ para cada uno de ellos. La respuesta en lazo cerrado para el i sistema LTV en cada instante del intervalo de tiempo es controlado por la red de avance diseñada $u_c^{(i)}(t,\theta_c)$. A partir de la convergencia de la secuencia de soluciones LTV [30], la última ley de control $u_c^{(i)}(t,\theta_c)$, (i), estabilizará el sistema no lineal.

2 Técnica de Iteración

Este método sustituye el problema no lineal original por una secuencia de sistemas LTV, cuyas soluciones convergen en el espacio de las funciones continuas a la solución del sistema no lineal bajo una condición de Lipschitz leve [30]. Cualquier sistema no lineal de la forma:

$$\dot{x}(t) = A[x(t)]x(t) + B[x(t)]u_c(t). \quad (2)$$

donde $x(0) = x_0 \in \mathbb{R}^n$, y $A[x(t)] \in \mathbb{R}^{nxn}$ es localmente Lipschitz, puede aproximarse por una secuencia de ecuaciones LVT donde el vector de estados $x(t) \in \mathbb{R}^n$, en las matrices $A[x(t)]$ y $B[x(t)]$ se sustituyen en cada iteración "i" por los estados obtenidos en la iteración anterior $x^{(i-1)}(t)$:

$$\begin{aligned} \dot{x}^{(1)}(t) &= A[x(0)]x^{(1)}(t) + B[x(0)]u_c^{(1)}(t), \\ &\vdots \\ \dot{x}^{(i)}(t) &= A[x^{(i-1)}(t)]x^{(i)}(t) + B[x^{(i-1)}(t)]u_c^{(i)}(t), \end{aligned} \quad (3)$$

con $x^{(1)}(0) = \cdots = x^{(i)}(0) = x(0)$, para $i \geq 1$ y $\forall t \in [0,\tau]$. Las soluciones $x^{(i)}$ convergen a la solución del sistema no lineal, $x(t)$. El teorema de convergencia global se encuentra en [30]. La aplicación de esta técnica proporciona una representación exacta de la solución no lineal después de pocas iteraciones. Los sistemas no lineales de la forma (2) que satisfacen los requerimientos locales de Lipschitz (suposición muy leve, al ser una condición asumida por la unicidad de la solución) se pueden ahora controlar con métodos lineales clásicos.

3 El Modelo Matemático

El modelo dinámico no lineal de este artículo se conoce como maniobra. El término maniobra trata el movimiento del buque en ausencia de olas [25]. El movimiento, es debido a los dispositivos de control: superficies de control (timones, aletas) y a la propulsión. El movimiento en modelos de maniobra de buques de 4GDL requiere cuatro coordenadas independientes para establecer la posición y orientación del vehículo (se considera un cuerpo rígido). Las coordenadas son las posiciones longitudinales, laterales y velocidades, así como sus derivadas en los respectivos sistemas de coordenadas. Los cuatro grados de libertad describen el movimiento (avance, desp. lateral y guiñada) en el plano horizontal y el balance en el plano vertical. Se utilizan dos sistemas de coordenadas: el sistema de coordenadas n (fijado a tierra, O_n), para definir la posición y el sistema b, (fijado al casco, O_b), ayuda a definir la orientación [25]. Las ecuaciones de cuerpo rígido para un modelo de maniobra de 4GDL vienen dadas por [26]:

$$\begin{aligned} m[\dot{u} - y_g^b\dot{r} - vr - x_g^b r^2 + z_g^b pr] &= \tau_X \\ m[\dot{v} - z_g^b\dot{p} + x_g^b\dot{r} + ur - y_g^b(r^2 + p^2)] &= \tau_Y \\ I_{xx}\dot{p} - mz_g^b\dot{v} + m[y_g^b vp - z_g^b ur] &= \tau_K \\ I_{zz}\dot{r} + mx_g^b\dot{v} - my_g^b\dot{u} + m[x_g^b ur - y_g^b vr] &= \tau_N \end{aligned} \quad (4)$$

El subíndice $_g$ se refiere al centro de gravedad y el superíndice ^{b} al sistema de coordenadas b. Los parámetros de las ecuaciones (4) se encuentran en [26]. Estas ecuaciones se formulan sobre el sistema de coordenadas b, que se fija al punto determinado por la intersección del plano de babor-estribor de simetría, el plano de la línea de flotación y el plano vertical transversal a $L_{pp}/2$ (ver [26] para las dimensiones del casco). Los términos de fuerza del miembro derecho de las ecuaciones (4) se pueden describir como la contribución total de las fuerzas hidrodinámicas, de propulsión y control:

$$\tau = \tau_{hyd} + \tau_p + \tau_c \quad (5)$$

3.1 Fuerzas Hidrodinámicas, τ_{hyd}

Las fuerzas hidrodinámicas surgen como consecuencia del movimiento de la embarcación en aguas tranquilas. Las siguientes ecuaciones corresponden al modelo establecido por [6] que propone una versión simplificada del modelo en [23], preservando de esta manera la coeficientes hidrodinámicos más importantes de modo que el modelo describa una amplia variedad de regímenes de maniobra a pesar de algunas simplificaciones menores. Las fuerzas hidrodinámicas están compuestas principalmente por términos de avance, desp. lateral, balance y guiñada:

Tabla 1: Notación para las variables de desplazamiento del buque.

Movimiento	**Fuerza**	**Velocidad Lineal**	**Posición**
Avance	X	u "b-sist. coord."	x_n "n-sist. coord."
Desp. Lateral	Y	v "b-sist. coord."	y_n "n-sist. coord."
Rotación	**Momento**	**Velocidad Angular**	**Ángulo**
Balance	K	p "b-sist. coord."	ϕ euler
Guiñada	N	r "b-sist. coord."	ψ(rumbo) euler

- *Términos de Avance*

$$\tau^b_{Xhyd} = X_{\dot{u}}\dot{u} + X_{vr}vr + X_{u|u|}u|u| \tag{6}$$

- *Términos de Desp. Lateral*

$$\begin{aligned}\tau^b_{Yhyd} &= Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + Y_{\dot{p}}\dot{p} + Y_{|u|v}|u|v + Y_{ur}ur \\ &+Y_{|v|v}|v|v + Y_{|v|r}|v|r + Y_{|r|v}|r|v + Y_{\phi|uv|}\phi|uv| \\ &+Y_{\phi|ur|}\phi|ur| + Y_{\phi uu}\phi u^2\end{aligned} \tag{7}$$

- *Términos de Balance*

$$\begin{aligned}\tau^b_{Khyd} &= K_{\dot{v}}\dot{v} - K_{\dot{p}}\dot{p} + K_{|u|v}|u|v + K_{ur}ur \\ &+K_{|v|v}|v|v + K_{|v|r}|v|r + K_{|r|v}|r|v + K_{\phi|uv|}\phi|uv| \\ &+K_{\phi|ur|}\phi|ur| + K_{\phi|uu|}\phi u^2 + K_{|u|p}|u|p + K_{p|p|}p|p| \\ &+K_p p - K_{\phi\phi\phi}\phi^3 + \rho g \nabla GMt\phi\end{aligned} \tag{8}$$

- *Términos de Guiñada*

$$\begin{aligned}\tau^b_{Nhyd} &= N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} + N_{|u|v}|u|v + N_{ur}ur \\ &+N_{|v|r}|v|r + N_{|r|v}|r|v + N_{\phi|uv|}\phi|uv| + N_{\phi u|r|}\phi u|r| \\ &+N_{|p|p}|p|p + N_{|u|p}|u|p + N_{\phi u|u|}\phi u|u| + N_{|v|v}|v|v\end{aligned} \tag{9}$$

siendo $\dot{\psi} = r$ y $\dot{\phi} = p$.

3.2 Fuerzas de Propulsión

La dinámica del sistema de propulsión no esta incluida en el modelo [26]. Asumimos que las hélices ofrecen un empuje constante T que compensa la resistencia en aguas tranquilas:

$$T = -X_{u|u|}u^2_{nom} \tag{10}$$

donde u_{nom} es la velocidad de servicio. El vector resultante de fuerzas de propulsión es:

$$\tau_p = [T,\ 0,\ 0,\ 0]^T \tag{11}$$

El movimiento del timón y la contribución de las aletas inducen fuerzas de arrastre que reducen la velocidad del buque.

3.3 Fuerzas de Control: Timón

El buque objeto de estudio de este artículo incorpora dos timones, que junto con la maquinaria de mando constituyen los actuadores del sistema. Con el fin de obtener la expresión de las fuerzas de control, se necesitan las fuerzas de arrastre y sustentación del timón [17]:

$$L = 1/2\rho V_f^2 A_f \bar{C}_L \alpha_e \tag{12}$$

$$D = 1/2\rho V_f^2 A_f (C_{D0} + \frac{(\bar{C}_L\alpha_e)^2}{0.9\pi a}) \tag{13}$$

V_f es la velocidad de flujo, A_f es el área del timón, α_e es el ángulo efectivo de ataque en radianes, y a es la relación de aspecto eficaz. Podemos utilizar la siguiente aproximación lineal para representar el coeficiente de sustentación:

$$\bar{C}_L = \frac{\partial C_L}{\partial \alpha_e}|_{\alpha_e=0} \tag{14}$$

Para calcular la fuerza de sustentación del timón, el ángulo efectivo de ataque, α_e, se aproxima por el ángulo mecánico del timón: $\alpha_e \approx \delta_c$, y la velocidad de flujo local en el timón se considera igual a la velocidad horizontal total de la embarcación, $V_f = \sqrt{u^2 + v^2}$. Aplicamos una corrección global para la fuerza de sustentación y arrastre [4]:

$$\Delta L = T\left[1 + \frac{1}{1 + C_{Th}} sin(\alpha_e)\right] \tag{15}$$

$$\Delta D = T\left[1 + \frac{1}{1 + C_{Th}}(1 - cos\,(\alpha_e))\right] \tag{16}$$

T es el empuje de la hélice, y C_{Th} es el coeficiente de carga hélice:

$$C_{Th} = \frac{2T}{\rho V_f^2 A_p} \tag{17}$$

A_p es el área del disco de la hélice. Las fuerzas de control, τ_c, son generadas por el timón en el sistema de coordenadas b:

$$\tau_c \approx [-D,\ L,\ z^b_{CP}L,\ x^b_{CP}L]^T \tag{18}$$

x^b_{CP} y z^b_{CP} son las coordenadas del centro de presión del timón (CP) con respecto al sistema de referencia . El CP está en la mecha del timón y en

el medio de la superficie del timón. La maquinaria hidráulica (para mover el timón) se implementa siguiendo el modelo de [34] que considera el ángulo y la velocidad máxima del timón. Cuando se trabaja en la zona no saturada, la dinámica del timón se representa por un sistema de primer orden donde $\delta(t)$ es el ángulo del timón, $\delta_c(t)$ el ángulo comandado del timón y T_m la constante de la maquinaria hidráulica:

$$\dot{\delta}(t) = \left[\delta_c(t) - \delta(t)\right]/T_m \tag{19}$$

3.4 Cinemática

La cinemática abarca los aspectos geométricos de desplazamiento del casco sin tener en cuenta la masa y fuerza. La posición del buque se obtiene realizando una transformación entre las velocidades lineales del sistema de coordenadas fijado al casco (b) y la derivada de las posiciones en el sistema (n). Esto es para un modelo de maniobra 6GDL:

$$(\dot{x}_n, \dot{y}_n, \dot{z}_n)^T = R_b^n\,(u, v, w)^T \tag{20}$$

u es la velocidad de avance, v la velocidad de desplazamiento lateral y w es la velocidad de arfada. La matriz de transformación de velocidades R_b^n se puede encontrar en [9,13], para el caso de un modelo de maniobra de 4 GDL como en este trabajo, el movimiento en el eje z no se considera siendo $\theta = 0$. Por lo tanto, la ecuación (20) se simplifica dando como resultado:

$$\begin{aligned} \dot{x}_n &= u \cdot cos(\psi) - v \cdot sin(\psi)cos(\phi) \\ \dot{y}_n &= u \cdot sin(\psi) + v \cdot cos(\psi)cos(\phi) \end{aligned} \tag{21}$$

ψ es el ángulo de guiñada y ϕ es el ángulo de balance, θ es el ángulo de cabezada.

4 Aproximación al Modelo Matemático

En esta sección, se muestra cómo aplicar la técnica de iteración (sección 2) y así aproximar el modelo no lineal del buque dado (sección 3) para el caso particular de una patrullera de costas. Los parámetros y principales características de la patrullera están incluidas en [26]. La patrullera de costas está equipada con dos timones y su velocidad de servicio es $u_{nom} = 15$ nudos $(7.71m/s)$. Las simulaciones se realizaron utilizando Matlab/Simulink y la toolbox GNC [10]. El tiempo de simulación es $t_f = 200$ segundos y el tamaño del paso de integración es $h = 0,1$. El período de muestreo h se selecciona en el rango de 20-40 muestras del tiempo de subida del grado de libertad más rápido. Las ecuaciones de movimiento de este sistema, (4)-(21), son no lineales y se puede reescribir en la forma:

$$\dot{x}(t) = A[x(t)]x(t) + B[x(t)]u_c(t), \quad x(0) = x_0 \in \mathbb{R}^9. \tag{22}$$

$A[x(t)] \in \mathbb{R}^{9x9}$, $B[x(t)] \in \mathbb{R}^{9x2}$, $u_c(t)$ es la señal de control y $x(t)$ es el vector de estados, $x(t) = [u\ v\ p\ r\ \phi\ \psi\ \delta\ x\ y]^T$. Se ha simulado una maniobra estándar zig-zag 20°-20° (ver [17]). Se ha seleccionado una amplitud de 20°con el objeto de excitar la dinámica no lineal y de esta manera mostrar el ajuste de la técnica de iteración al sistema original no lineal. El vector de control para llevar a cabo esta maniobra es $u_c(t) = [\delta_c\ T]^T$, donde T se definió anteriormente en (10) y δ_c es el comando del timón para realizar las fases de la maniobra de zig-zag. A pesar de que no hay diseño de metodología de control, el zig-zag es una maniobra en lazo cerrado donde se mide el rumbo actual $\psi(t)$ y no se cambia el timón de estribor a babor o viceversa hasta que $\psi(t)$ alcanza un valor determinado. La maniobra de zig-zag a de ser completada con al menos cinco fases. Las condiciones iniciales de la maniobra son: $x_0 = [u_{nom}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$, que se sustituyen en el primer sistema lineal aproximado, matrices $A[x_0]$, $B[x_0]$ y, posteriormente, la técnica de iteración da como resultado una secuencia de sistemas (LVT). Para que la técnica de iteración converja al sistema no lineal original se han necesitado 20 iteraciones. La figura (4) muestra la evolución temporal de uno de los estados durante la maniobra de zig zag 20°-20° para las distintas iteraciones y la evolución en el tiempo en el caso del modelo no lineal (línea roja), esto se hace con el fin de ilustrar la convergencia de este método. Se ve cómo la iteración 20 es una representación exacta de la solución del sistema no lineal. Asimismo, se muestra la iteración 40 con el fin de demostrar la convergencia de los estados. Después de la iteración 20, la solución de $x^{(20)}(t)$ converge a la solución no lineal $x(t)$. También se muestra cómo en las siguientes iteraciones hasta $x^{(40)}(t)$ hay una variación mínima.

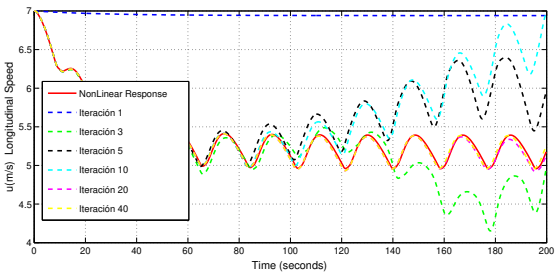


Figura 1: Convergencia del estado $u(t)$ en una maniobra zig-zag.

5 Control No Lineal

5.1 Diseño del Controlador

Un piloto automático debe cumplir dos objetivos: mantener y cambiar el rumbo. En el primer caso, se desea mantener la trayectoria en un rumbo constante deseado, ψ_d. En el segundo caso, el objetivo es cambiar el rumbo sin introducir grandes oscilaciones en la respuesta y en un tiempo mínimo. En ambos casos, el adecuado funcionamiento del sistema debe ser independiente de las perturbaciones producidas por factores externos: viento, olas y corrientes marinas. El rumbo seguido por el barco, $\psi(t)$, se puede representar por un sistema de segundo orden:

$$\ddot{\psi}(t) + 2\zeta w_n \dot{\psi}(t) + w_n^2 \psi(t) = w_n^2 \psi_d \qquad (23)$$

donde w_n es la frecuencia natural y ζ es el coeficiente de amortiguamiento deseado del sistema en lazo cerrado. ζ se elige típicamente dentro de los límites del intervalo $(0.8 \leq \zeta \leq 1)$ con el objetivo de realizar maniobras de manera segura. Esto es, en aguas restringidas y para evitar colisiones, la maniobra de cambio de rumbo a de tener un comienzo claro, con el fin de advertir a los buques cercanos la intención de la maniobra y, por esa razón, esta maniobra debe completarse preferentemente sin máximo sobreimpulso. El siguiente esquema de control PID se utiliza convencionalmente para la aplicación de control de rumbo:

$$U_c(s) = \frac{\delta_c}{E}(s) = \left[k_p + \frac{k_i}{s} + \frac{k_d s}{\alpha T_d s + 1}\right] \qquad (24)$$

donde $k_d = T_d k_p$ and $k_i = k_p/T_i$ siendo T_d el tiempo derivativo, T_i el tiempo integral , $\delta_c(s)$ transformada de Laplace de la posición del timón y $E(s)$ la transformada de Laplace del error, $e(t) = \psi_d - \psi(t)$ y $U_c(s)$ es la transformada de Laplace de la señal de control, $u_c(t, \theta_c)$. El vector $\psi(t)$ se extrae de los estados, siendo $x(t) = [u\ v\ p\ r\ \phi\ \psi\ \delta\ x\ y]^T$ y $x_{(6)}(t) = \psi(t)$.

Los niveles de ruido de la instrumentación estándar a bordo pueden causar problemas de amplificación de ruido por causa del modo derivativo. El esquema PID (24), en el que la acción derivada se filtra mediante un sistema de primer orden $\frac{1}{\alpha T_d s+1}$, evita este problema de amplificación de ruido.

Es muy probable que las saturaciones en velocidad y ángulo de deflexión del timón provoquen el fenómeno llamado windup cuando se aplica la metodología PID (ver [1] para más detalles). Esto es, la contribución del término integral, $(\frac{k_i}{s})$, puede llegar a ser elevada y, como consecuencia, la respuesta del sistema puede adoptar altos niveles de oscilación. Existen varios sistemas anti-windup propuestos en la literatura (véase [1] y las referencias que contiene), pero en lugar de la aplicación de uno de ellos, que supondría una mayor complejidad del controlador diseñado, se ha optado por un método más simple: una estructura de control modificado como el siguiente controlador de red de primer orden, en el que la acción integral se omite:

$$U_c(s) = \frac{\delta_c}{E}(s) = K\left[\frac{s+z}{s+p}\right] \qquad (25)$$

donde $K > 0$ y $p > z$.

La ecuación (25) representa un controlador de compensación de avance [24] que tiene un cero situado más cerca del origen del plano s que el polo. Este cero dominante, mejora la estabilidad del sistema, lo cual es deseable con el fin de satisfacer el objetivo de obtener una respuesta de rumbo sin sobreimpulso.

5.2 Sintonización del Controlador

La sintonización del algoritmo de optimización toma los datos de la salida (rumbo, $\psi(t)$) y de la entrada (rumbo deseado, ψ_d). En la selección del método de optimización se tuvieron en cuenta los objetivos de control de rumbo: reducción al mínimo del sobreimpulso y del tiempo de establecimiento sin error en estado estacionario. Dados estos objetivos, se ha escogido la técnica de optimización minimax, que minimiza el valor máximo de la salida: cuando el valor máximo de la salida se reduce, el sobreimpulso se minimiza. La aplicación del problema minimax para el control de rumbo, consiste en reducir al mínimo el valor máximo de la salida, $\psi(t)$, durante el intervalo de tiempo de simulación $[t_0, t_f]$. La siguiente restricción se impone tal que $\psi(t) \leq \psi_d$, (ψ_d valor de entrada constante),

$$\psi(t) \leq \psi_d, \quad t_r \leq t \leq t_f \qquad (26)$$

siendo t_r el tiempo de subida del sistema. Mediante la imposición de esta restricción, se espera una respuesta plana, sin sobreimpulso y sin error en estado estacionario. El valor de t_r se determina sobre la base de un conocimiento previo de la respuesta del sistema. A continuación, se aplica el problema minimax [7, 28]:

$$\min_{\theta_c^{(i)}} \max_j \{\psi_j(\theta_c^{(i)})\} \equiv \begin{cases} \psi(t) \leq \psi_d, & t_r \leq t \leq t_f \\ lb \leq \theta_c^{(i)} \leq ub \end{cases} \qquad (27)$$

donde $\psi(t)$ es el rumbo, $\theta_c^{(i)}$ son los parámetros del controlador correspondiente a la i aproximación LVT a optimizar, lb es el límite inferior de los parámetros, ub es el límite superior de los parámetros y el subíndice j representa un conjunto de funciones multivariable.

5.3 Técnica de Iteración para Control

La metodología previamente presentada se aplica al caso de control de rumbo. Las ecuaciones de movimiento de este sistema son altamente no lineales y se puede re-escribir:

$$\dot{x}(t) = A[x(t)]x(t) + B[x(t)]u_c(t,\theta_c), \quad x(0) = x_0. \tag{28}$$

donde $A[x(t)] \in \mathbb{R}^{nxn}$, $B[x(t)] \in \mathbb{R}^{nxm}$, $x(t)$ es el vector de estados y el vector de control $u_c(t,\theta_c)$ se diseña mediante la metodología propuesta en 5.1. El sistema (28) se puede aproximar por una secuencia de sistemas LVT de la forma (3). con condiciones iniciales $x^{(1)}(0) = \cdots = x^{(i)}(0) = x(0)$. Para cada una de las "$i$" LVT iteraciones, se diseña una señal de control $u_c^{(i)}(t,\theta_c)$. En la última iteración se obtiene, que la secuencia de soluciones converge a la solución no lineal. La última señal de control diseñada se aplica al problema no lineal original, para lograr el control de los estados:

$$\dot{x}(t) = A[x(t)]x(t)+B[x(t)]u_c^{(i)}(t,\theta_c^{(i)}), \quad x(0) = x_0. \tag{29}$$

6 Simulaciones y Resultados

El escenario de simulación se basa en los datos de la patrullera de costas usados en la sección 2. La maniobra de mantenimiento de rumbo ψ_d=20° sirve para validar y probar el diseño del controlador iterativo implementado en base a la teoría previamente expuesta. La maniobra debe completarse satisfaciendo los objetivos señalados en la sección 5.1.

El modelo definido en la sección 3 se reordena en forma $\dot{x} = A(x)x(t) + B(x)u_c(t,\theta_c)$ donde $x(t,\theta_c) = [u\ v\ p\ r\ \phi\ \psi\ \delta\ x\ y]^T$ y el vector de control es $u_c(t,\theta_c) = [\delta_c\ T]^T$. Las condiciones iniciales son: $x_0 = [u_{nom}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$.

En base a los resultados previos para la maniobra de mantenimiento de rumbo 20° con el método de optimización propuesto, se obtuvo un valor elevado de la constante integral T_i del controlador PID (24). Tal y como se explica en la sección 5.1, la técnica de optimización aplicada para la sintonización, en un intento de reducir la oscilación causada por el problema de windup integral, proporciona un valor alto de T_i para reducir al mínimo la influencia del término integral. Esto sugiere, que la contribución del término integral k_i/s en el controlador PID (24) (siendo $k_i = k_p/T_i$) se puede despreciar. Por todas estas razones, se utiliza un controlador de compensación de avance sin acción integral, en lugar del PID. Se han impuesto restricciones en los parámetros del controlador ($lb = 0$ y $ub = \infty$), con el fin de evitar un comportamiento del controlador inestable. Los parámetros iniciales del controlador de avance se han seleccionado teniendo en cuenta que este tipo de controlador debe tener un cero dominante cerca del origen del plano s.

Las figuras 2 y 3 muestran los resultados para una maniobra de mantenimiento de rumbo 20° (0.349 rad) para cada iteración "i". Después de la iteración 5, el algoritmo converge. Los parámetros de control correspondientes θ_c y la respuesta del rumbo $\psi(t)$ permanecen casi sin cambios. El zoom hecho para el control de rumbo $\psi(t)$ en la parte superior de la figura 2 para las iteraciones 5-8 muestra que la diferencia entre la iteración i y la i-1, es del orden de $\frac{1}{100}$ de grado, que ilustra las propiedades de convergencia del algoritmo presentado. Las figuras 2 y 3 muestran claramente una aproximación exacta para la iteración 5 para el modelo no lineal (véase iteración 5, con la simulación de datos generados con el sistema no lineal original y los parámetros del controlador $\theta_c^{(5)}$). En esta iteración, (i = 5), se reduce el sobreimpulso en la respuesta de rumbo $\psi(t)$ y el tiempo de establecimiento, con respecto a las iteraciones anteriores. Por otra parte, el error en estado estacionario ($e(t) = \psi_d - \psi(t)$) converge a cero después de sólo 30 segundos.

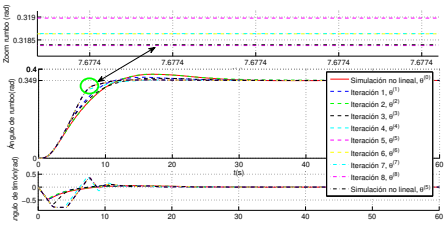


Figura 2: Resultados de convergencia de la variable controlada $\psi(t)$, el actuador y la deflexión del timón, $\delta(t)$, para la patrullera de costas en una maniobra de mantenimiento de rumbo de 20°.

La parte inferior de la figura 2 muestra el desplazamiento del actuador, $\delta(t)$, que representa el valor actual de deflexión del timón. El actuador se satura en la quinta iteración, pero gracias al controlador de compensación de avance, se evita el problema de windup obteniendo una respuesta sin sobreimpulso. El controlador de compensación de avance es una solución más simple que un sistema anti-windup para el controlador PID.

7 Conclusiones

En este trabajo se propone una estrategia de control basada en una metodología de compensación de avance optimizada, combinada con una técnica iterativa basada en aproximaciones LVT a la dinámica no lineal de un buque. La teoría

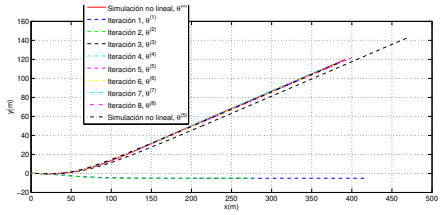


Figura 3: Resultados de convergencia de la posición de la patrullera de costas en una maniobra de mantenimiento de rumbo de 20°.

que aquí se presenta se ha implementado en Matlab/Simulink y se aplica al caso particular de una patrullera de costas en dos escenarios diferentes: en primer lugar, un maniobra de zig zag 20°-20°donde se muestra la convergencia del método de iteración y en segundo lugar, un maniobra de mantenimiento de rumbo de 20° muestra la exactitud de las capacidades de seguimiento del controlador diseñado cuando se aplica a la última iteración de los sistemas LTV.

En el primer caso, los resultados muestran que la aproximaciones LVT a la dinámica no lineal de la patrullera en la maniobra de zig-zag $20° - 20°$ es buena después de un número pequeño de iteraciones, 20 en este caso. Mediante la generación de esta secuencia de ecuaciones LVT que se aproximan a la dinámica no lineal original, se pueden aplicar técnicas de control lineales a la última iteración. Por otro lado, mediante la maniobra de mantenimiento de rumbo de 20°, se pone a prueba la estrategia de control propuesta y la metodología de control de rumbo. En estudios preliminares, la obtención de un valor alto de la constante integral T_i con la estrategia de control optimizada propuesta, indica que la saturación del timón provoca el problema de windup integral cuando se aplica un controlador PID. Por lo tanto, es aconsejable utilizar un controlador sin término integral como es el controlador de compensación avance. Los resultados presentados con el controlador de compensación de avance cumplen con los objetivos planteados para el control de rumbo: la eliminación del máximo sobreimpulso, la reducción del tiempo de establecimiento y la eliminación del error de estado estacionario. Además de esto, el controlador de compensación avance constituye una solución más simple que un sistema de antiwindup para controladores PID.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economia y Competitividad (DPI2011-27990) con fondos FEDER y por el Ministerio de Defensa Español, Programa COINCIDENTE(1003211003100).

Referencias

[1] Karl Johan Astrom and Tore Hagglund. *Advanced PID control*. ISA- The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC, 2006. Karl J. Astrom, Tore Hagglund.

[2] A. Bateman, J. Hull, and Zongli Lin. A backstepping-based low-and-high gain design for marine vehicles. *International Journal of Robust and Nonlinear Control*, 19(4):480 – 93, 2009.

[3] M. I. Bech and L. Wagner Smith. Analogue simulation of ship maneuvres. Hy 14, Hydro-Og and Aerodynamics Laboratory, 1969.

[4] Volker Bertram. *Practical ship hydrodynamics*. Butterworth-Heinemann, Oxford, 2 edition, 2012. Volker Bertram.

[5] S.K. Bhattacharyya, G. Rajesh, and Deepak Kumar Gupta. Fuzzy autopilot for ship maneuvering. *International Shipbuilding Progress*, 58(4):191 – 218, 2011.

[6] M. Blanke. *Ship Propulsion Losses Related to Automated Steeering and Prime Mover Control*. PhD thesis, The University of Denmark Lyngby., 1981.

[7] R.K. Brayton and G.D. Hachtel. A new algorithm for statistical circuit design based on quasi-newton methods and function splitting. *IEEE Transactions on Circuits and Systems*, CAS-26(9):784 – 94, 1979.

[8] T. Cimen and S.P. Banks. Nonlinear optimal tracking control with application to supertankers for autopilot design. *Automatica*, 40(11):1845 – 63, 2004.

[9] T. I. Fossen. *Guidance and Control of Ocean Marine Vehicles*. John Wiley and Sons Ltd, 1994.

[10] T. I. Fossen and T. Perez. Marine systems simulator (mss), 2004.

[11] Thor I. Fossen. High performance ship autopilot with wave filter. In *Proceedings of the 10th International Ship Control Systems Symposium*, pages 2271–2285, 1993.

[12] T.I. Fossen. A survey on nonlinear ship control: from theory to practice. pages 1 – 16, Kidlington, UK, 2000.

[13] T.I. Fossen. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, 2002.

[14] Shr-Shiung Hu, Pao-Hwa Yang, J.Y. Juang, and B.C. Chang. Robust nonlinear ship course-keeping control by h i/o linearization and -synthesis. *International Journal of Robust and Nonlinear Control*, 13(1):55 – 70, 2003.

[15] C.G. Kallstrom, K.J. Astrom, N.E. Thorell, J. Eriksson, and L. Sten. Adaptive autopilots for tankers. *Automatica*, 15(3):241 – 54, 1979.

[16] Claes G. Kallstrom. Autopilot and track-keeping algorithms for high-speed craft. volume 8, pages 185 – 190, 2000.

[17] Edward V. Lewis. *Principles of Naval Arquitecture*. The Society of Naval Architects and Marine Engineers, 1989.

[18] E.W. McGookin, D.J. Murray-Smith, Yun Li, and T.I. Fossen. Ship steering control system optimisation using genetic algorithms. *Control Engineering Practice*, 8(4):429 – 43, 2000.

[19] L. Morawski and J. Pomiriski. Ship track-keeping experiments with a physical tanker model. *Control Engineering Practice*, 1998.

[20] Lucia Moreira, Thor I. Fossen, and C. Guedes Soares. Path following control system for a tanker ship model. *Ocean Engineering*, 34(14-15):2074 – 2085, 2007.

[21] Van-Chung Nguyen, Thi-Oanh Le, Thi-Bich-Su Do, Manh-Minh Nguyen, Thi-Mai Nguyen, Thi-Kim-Tuyen Dinh, and Thanh-Nhan Man. Study on an effective adaptive ship autopilot. In *Advanced Motion Control, 2004. AMC '04. The 8th IEEE International Workshop on*, pages 707 – 710, march 2004.

[22] N. H. Norrbin. On the design and analysis of the zig-zag test on base of quasi linear frequency response. Technical Report 104-3, The Swedish State Shipbuilding Experimental Tank (SSPA), 1963.

[23] N.H. Norrbin. Theory and observation on the use of a mathematical model for ship manoeuvring in deep and confined waters. *8th Symposium on Naval Hydrodynamics, USA*, 1970.

[24] Katsuhiko Ogata. *Modern control engineering*. Prentice-Hall, Boston, MA, 5th edition, 2010.

[25] T. Pérez. *Ship Motion control Course keeping and roll Stabilisation using rudder and fins*. Springer, 2005.

[26] T. Perez, A. Ross, and T. I. Fossen. A 4-dof simulink model of a coastal patrol vessel for manoeuvering in waves. In *Manoeuvring and Control of Marine Craft (MCMC 2006)*, Lisbon, 2006.

[27] R.E. Reid and B.C. Mears. Design of the steering controlled of a supertanker using linear quadratic control theory: a feasibility study. *IEEE Transactions on Automatic Control*, AC-27(4):940 – 2, 1982.

[28] E. Revestido, E. Moyano, F. J. Velasco, and E. López. Tuning heading controllers of an autonomous in-scale fast-ferry model. In *Maritime Transport IV*, 2009.

[29] B. Samanta and C. Nataraj. Design of intelligent ship autopilots using particle swarm optimization. In *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, pages 1 –7, sept. 2008.

[30] M. Tomas-Rodriguez and S.P. Banks. Linear approximations to nonlinear dynamical systems with applications to stability and spectral theory. *IMA Journal of Mathematical Control and Information*, 20(1):89 – 103, 2003.

[31] M. Tomas-Rodriguez and S.P. Banks. An iterative approach to eigenvalue assignment for nonlinear systems. *International Journal of Control*, 20(1):89 – 103, 2013.

[32] M. Tomas-Rodriguez, C. Navarro-Hernande, and S.P. Banks. Parametric approach to optimal nonlinear control problem using orthogonal expansions. volume 16, pages 556 – 561, Prague, Czech republic, 2005.

[33] Maria Tomas-Rodriguez and Stephen P. Banks. *Linear, Time-varying Approximations to Nonlinear Dynamical Systems: with Applications in Control and Optimization*, volume 400. 2010.

[34] J. Van Amerongen. Adaptive steering of ships-a model reference approach. *Automatica*, 20(1):3 – 14, 1984.

[35] A. Witkowska and R. Smierzchalski. Nonlinear backstepping ship course controller. *International Journal of Automation and Computing*, 6(3):277 – 84, 2009.

A ROBUST GRID SIDE CONVERTER CONTROL FOR WIND TURBINE SYSTEM BASED ON DOUBLE FEED INDUCTION GENERATOR

Oscar Barambones, José María González de Durana and Patxi Alkorta
Department of System Engineering and Automatic Control. University of the Basque Country. Spain.
oscar.barambones@ehu.es, josemaria.gonzalezdedurana@ehu.es, patxi.alkorta@ehu.es

Resumen

The performance of the wind power conversions systems can be greatly improved using an appropriate control algorithm. In this work, a robust grid side converter control for wind power generation that incorporates a doubly feed induction generator is described. In the presented design the so called vector control theory is applied. The proposed control scheme uses a grid voltage vector control for the grid side converter bridge control. The proposed robust control law is based on a sliding mode control theory that, as it is well known, presents a good performance under system uncertainties. The stability analysis of the proposed controller under disturbances and parameter uncertainties is provided using the Lyapunov stability theory. Finally simulated results show, on the one hand that the proposed controller provides high-performance dynamic characteristics, and on the other hand that this scheme is robust with respect to the uncertainties that usually appear in the real systems.

Palabras clave: Wind Turbine System, Robust control, Double Feed Induction Generator.

1 INTRODUCTION

The Doubly Feed Induction Generators (DFIG), using a vector control scheme, are widely used in the variable speed operation wind turbine system, owing to their ability to maximize wind power extraction [1]. In these DFIG wind turbines the control system should be designed in order to to achieve the following objectives: regulating the DFIG rotor speed for maximum wind power capture, maintaining the DFIG stator output voltage frequency constant and controlling the DFIG reactive power [2].

The DFIG wind turbine control system are generally composed of two parts: the electrical control on the DFIG and the mechanical control on the wind turbine blade pitch angle. Control of the DFIG is achieved controlling the variable frequency converter (VFC), which includes control of the rotor-side converter (RSC) and control of the grid-side converter (GSC) [3]-[5]. The objective of the RSC is to govern both the stator-side active and reactive powers independently; while the objective of the GSC is to keep the dc-link voltage constant regardless of the magnitude and direction of the rotor power. The GSC control scheme can also be designed to regulate the reactive power or the stator terminal voltage of the DFIG.

This paper investigates a new robust grid side converter control for variable speed wind turbines, in order to maintain the DC link voltage constant regardless of the rotor power flow direction.

Basically, the proposed robust design uses the sliding mode control algorithm to regulate the GSC. In the design it is used a vector oriented control theory in order to align the q-component of the voltage with the supply voltage phasor [6]. This control scheme leads to maintain the DC link constant regardless of the rotor power flow direction and the system uncertainties.

Finally, tests of the proposed method based on a two-bladed horizontal axis wind turbine is conducted using the Matlab/Simulink software. In this test, several operating conditions are simulated and satisfactory results are obtained.

2 GRID SIDE CONVERTER CONTROL

A typical scheme of a DFIG equipped wind turbine is shown in Figure 1. Control of the DFIG is achieved controlling the variable frequency converter (VFC), which includes control of the rotor-side converter (RSC) and control of the grid-side converter (GSC) [7]-[9]. The objective of the RSC is to govern both the stator-side active and reactive powers independently.

The RSC control scheme should be designed in order to regulate the wind turbine speed for maximum wind power capture. Therefore, a suitably designed speed controller is essential to track the optimal wind turbine speed reference w^* for maximum wind power extraction. In this work the RSC control scheme proposed in [10] is used.

On the other hand, the objective of the GSC control is to keep the dc link voltage constant regardless of the direction of rotor power flow. In order to achieve this objective, a vector control approach is used, with

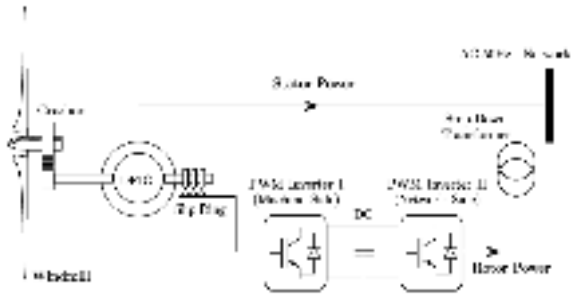


Figura 1: Scheme of a DFIG equipped wind turbine

a reference frame oriented along the stator (or supply) voltage vector position. In such scheme, direct axis current is controlled to keep the dc link voltage constant, and quadrature axis current component can be used to regulate the reactive power flow between the supply side converter and the supply. In this vector control scheme, all voltage and current quantities are transformed to a reference frame that rotates at the same speed as the supply voltage space phase with the real axis (d-axis) of the reference frame aligned to the supply voltage vector. At steady state, the reference frame speed equals the synchronous speed.

The scheme makes use of the supply voltage angle determined dynamically to map the the supply voltage, the converter terminal voltage and the phase currents onto the new reference frame.

In the stator voltage oriented reference frame, the d-axis is aligned with the supply voltage phasor V_s, and then $v_d = V_s$ and $v_q = 0$. Hence, the powers between the grid side converter and the grid are:

$$P = \frac{3}{2}(v_d i_d + v_q i_q) = \frac{3}{2} v_d i_d \quad (1)$$

$$Q = \frac{3}{2}(v_q i_d - v_d i_q) = -\frac{3}{2} v_d i_q \quad (2)$$

where v_d and v_q are the direct and quadrature components of the supply voltages, and i_d and i_q are the direct and quadrature components of the grid side converter input currents.

From the previous equations it is observed that the active and reactive power flow between supply side converter and the supply, will be proportional to i_d and i_q respectively.

The dc power change has to be equal to the active power flowing between the grid and the grid side converter. Thus,

$$E i_{0s} = \frac{3}{2} v_d i_d \quad (3)$$

$$C\frac{dE}{dt} = i_{0s} - i_{0r} \quad (4)$$

where E is the dc link voltage, i_{0r} is the current between the dc link and the rotor and i_{0s} is the current between the stator and the dc link.

From eqn. 3 and 4 it is obtained:

$$\dot{E} = \frac{1}{C}\left(\frac{3}{2}\frac{v_d}{E} i_d - i_{0r}\right) \quad (5)$$

$$\dot{E} = g(t) i_d - \frac{1}{C} i_{0r} \quad (6)$$

where the function $g(t)$ is defined as:

$$g(t) = \frac{1}{C}\frac{3}{2}\frac{v_d}{E} \quad (7)$$

The function $g(t)$ can be split up into two parts:

$$g(t) = g_0 + \triangle g(t) \quad (8)$$

where

$$g_0 = \frac{1}{C}\frac{3}{2}\frac{v_d}{E^*} \quad (9)$$

where the term E^* represents the reference value of E, and the term $\triangle g(t)$ takes into account the deviations from the reference value.

It should be noted that if the controller works appropriately, the term $\triangle g(t)$ will be a small value, because the dc link voltage will be roughly constant.

Then eqn. 6 can be put as,

$$\dot{E} = (g_0 + \triangle g) i_d - \frac{1}{C} i_{0r} \quad (10)$$

$$\dot{E} = g_0 i_d + \frac{1}{C} i_{0r} + d(t) \quad (11)$$

where $d(t) = \triangle g i_d$ is the uncertainty term.

Let us define the dc link voltage error as follows,

$$e(t) = E(t) - E^* \quad (12)$$

Taking the derivative of the previous equation with respect to time yields,

$$\dot{e}(t) = \dot{E}(t) - 0 = g_0 i_d + \frac{1}{C} i_{0r} + d(t) \quad (13)$$

$$= u(t) + d(t) \quad (14)$$

where

$$u(t) = g_0 i_d - \frac{1}{C} i_{0r} \quad (15)$$

Now the sliding variable $S(t)$ is defined with an integral component as:

$$S(t) = e(t) + \int_0^t \lambda\, e(\tau)\, d\tau \quad (16)$$

where λ is a positive constant.

Now, we are going to design a variable structure controller in order to regulate the dc link,

$$u(t) = -\lambda\, e(t) - \gamma\, \mathrm{sgn}(S) \quad (17)$$

where the λ is the constant gain defined previously, γ is the switching gain, S is the sliding variable defined in eqn. (16) and $\mathrm{sgn}(\cdot)$ is the signum function.

As in the case of the wind turbine speed controller, the following assumption should be formulated in order to regulate the dc link:

($\mathcal{A}1$) The gain γ must be chosen so that $\gamma \geq |d(t)|$ for all time.

Note that this condition only implies that the uncertainty term $d(t)$ is a bounded magnitude.

Theorem 1 *Consider the dc-link voltage dynamic equation (11). Then, if assumption ($\mathcal{A}1$) is verified, the control law (21) leads the dc-link voltage $E(t)$, so that the voltage regulation error $e(t) = E(t) - E^*(t)$ tends to zero as the time tends to infinity.*

The proof of this theorem will be carried out using the Lyapunov stability theory.

Proof:

Define the Lyapunov function candidate:

$$V(t) = \frac{1}{2} S(t) S(t) \tag{18}$$

Its time derivative is calculated as:

$$\begin{aligned} \dot{V}(t) = & \quad S(t)\dot{S}(t) \\ & = S \cdot (\dot{e} + \lambda e) \\ & = S \cdot (u + d + \lambda\, e)) \\ & = S \cdot (-\lambda\, e - \gamma\, \mathrm{sgn}(S) + d + \lambda\, e) \\ & = S \cdot (d - \gamma\, \mathrm{sgn}(S)) \\ & \leq -(\gamma - |d|)|S| \\ & \leq 0 \end{aligned} \tag{19}$$

Using the Lyapunov's direct method, since $V(t)$ is clearly positive-definite, $\dot{V}(t)$ is negative definite and $V(t)$ tends to infinity as $S(t)$ tends to infinity, then the equilibrium at the origin $S(t) = 0$ is globally asymptotically stable. Therefore $S(t)$ tends to zero as the time tends to infinity. Moreover, all trajectories starting off the sliding surface $S = 0$ must reach it in finite time and then will remain on this surface.

Then, as $\lambda > 0$, the regulation error $e(t)$ converges to zero exponentially.

Finally, the direct component of the supply current command, $i_d^*(t)$, can be obtained from equations (21) and (15):

$$i_d^*(t) = \frac{1}{g_0} \left[\frac{1}{C}\, i_{0r} - \lambda e - \gamma\, \mathrm{sgn}(S) \right] \tag{20}$$

Therefore, the proposed variable structure voltage control resolves the dc link voltage regulation.

3 CONTINUOUS APPROXIMATION OF SWITCHING CONTROL LAW

A frequently encountered problem in sliding control is that the control signal given by eqn.(20) is not smooth since the sliding control law is discontinuous across the sliding surfaces, which causes the chattering phenomenon. Chattering is undesirable in practice, since it involves high control activity and further may excite high-frequency dynamics. This situation can be avoided by smoothing out the control chattering within a thin boundary layer of thickness $\xi > 0$ neighboring the switching surface [11], [12].

In this way, some modifications should be done in the control law in order to reduce the chattering phenomenon. In this sense, the sign function included the control law (20) is replaced by a saturation function, so that it becomes:

$$i_d^*(t) = \frac{1}{g_0} \left[\frac{1}{C}\, i_{0r} - \lambda e - \gamma\, \mathrm{sat}(S) \right] \tag{21}$$

where the saturation function $sat(\cdot)$ is defined in the usual way:

$$\mathrm{sat}\left(\frac{S}{\xi}\right) = \begin{cases} \mathrm{sgn}(S) & if \quad |S| > \xi \\ \dfrac{S}{\xi} & otherwise. \end{cases}$$

and ξ is a positive constant that represents the thickness of the boundary layer neighboring the switching surface.

4 SIMULATION RESULTS

In this section we will study the variable speed wind turbine regulation performance using the proposed sliding-mode field oriented control scheme. The objective of this regulation is to keep the dc link voltage constant regardless of the direction of the rotor power flow and the system uncertainties.

The simulation are carried out using the Matlab/Simulink software and the turbine model is the one provided in the SimPowerSystems library [13].

In this example simulation a variable speed wind farm with a rated power of 9 MW is used. The farm consists of six 1.5 MW wind turbines connected to a 575 V bus line. The wind turbines use a doubly-fed induction generator (DFIG) consisting of a wound rotor induction generator and an AC/DC/AC IGBT-based PWM converter. The stator winding is connected directly to the 60 Hz grid while the rotor is fed at variable frequency through the AC/DC/AC converter.

The system has the following mechanical parameters. The combined generator and turbine inertia constant is $J = 5.04s$ expressed in seconds, the combined viscous friction factor $B = 0.01pu$ in pu based on the generator rating and there are three pole pairs [13].

In this simulation example it is assumed that there is an uncertainty around 20 % in the system parameters, that will be overcome by the proposed sliding control.

Finally, the following values have been chosen for the controller parameters, $\lambda = 70$, $\gamma = 25$ and $\xi = 0.1$.

In this case study, the rotor is running at subsynchronous speed for wind speeds lower than 10 m/s and it is running at a super-synchronous speed for higher wind speeds. In figure 2, the turbine mechanical power as function of turbine speed is displayed in, for wind speeds ranging from 5 m/s to 16.2 m/s.

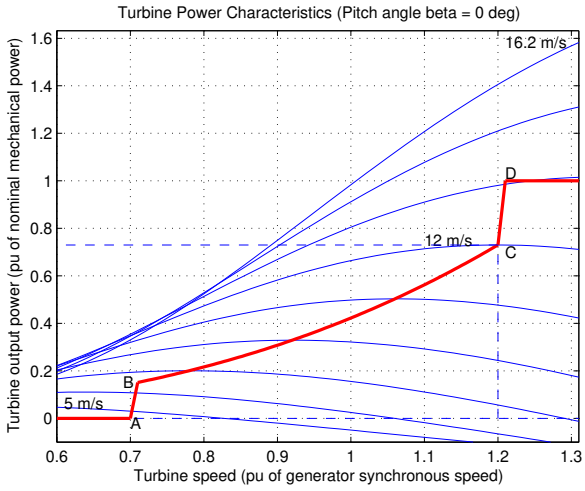


Figura 2: Turbine Power Characteristics

In this simulation it is used a variable wind speed, and figure 3 shows the wind speed used in this simulation. As it can be seen in the figure, the wind speed varies between $0m/s$ and $22m/s$.

Figure 4 shows the generated active. As it can be observed in this figure, at time $12.6s$ the mechanical power (and therefore the generated active power) should be limited by the pitch angle so as not to exceed the rated power of this system in order to prevent damages in the wind turbine.

Figure 5 shows the generated reactive power.

Figure 6 shows the pitch angle. As it can be observed in this figure, at time $12.6s$ the mechanical power (and therefore the generated active power) should be limited by the pitch angle so as not to exceed the rated power of this system.

Finally, figure 7 shows the dc link voltage response obtained using the proposed control scheme. As it can be observed, the DC-link voltage is maintained constant regardless of the direction of the rotor power flow and the system uncertainties. Moreover, the chattering phenomenon has been eliminated because the control signal is smoothed out within a boundary layer by means of a saturation function instead of signum function

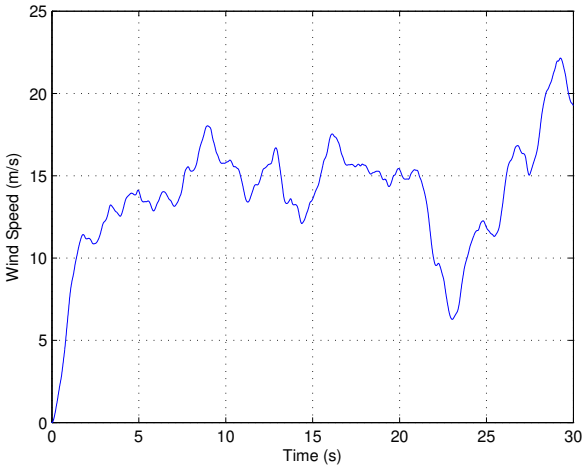


Figura 3: Wind speed

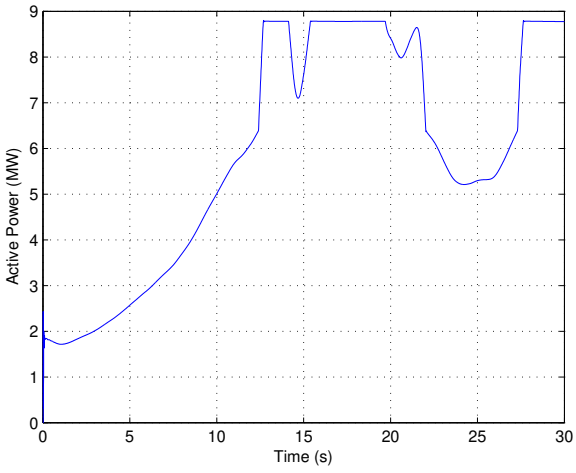


Figura 4: Generated active power

5 Conclusion

In this paper a robust GSC control for a doubly feed induction generator drive, used in variable speed wind power generation, is described. Due to the nature of the sliding control, this control scheme is robust under uncertainties that appear in the real systems. The closed loop stability of the presented design has been proved through Lyapunov stability theory.

The proposed control method keeps the dc link voltage constant regardless of the direction of rotor power flow. In order to achieve this objective, a vector con-

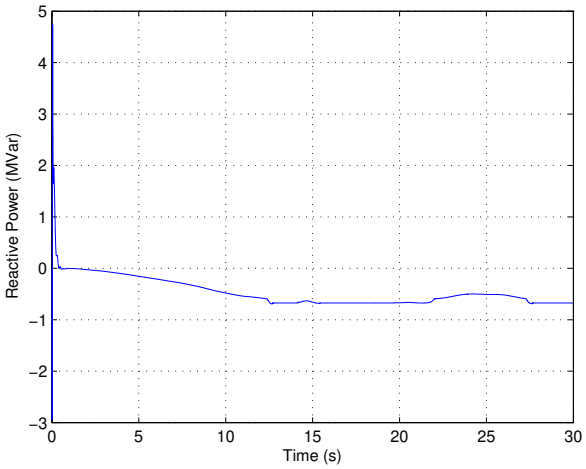


Figura 5: Generated reactive power

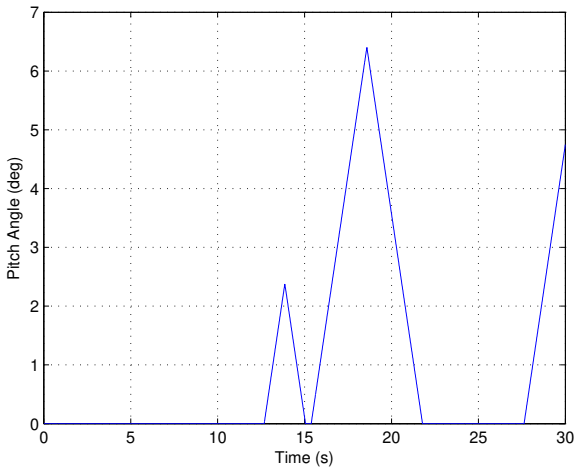


Figura 6: Pitch Angle

trol approach is used, with a reference frame oriented along the stator (or supply) voltage vector position.

Finally, it has been shown that the proposed control scheme performs reasonably well in practice by means of simulation examples. These simulation examples have also proved that the speed tracking objective is achieved in order maintain the DC-link voltage constant regardless of the direction of the rotor power flow and the system uncertainties.

Agradecimientos

The authors are very grateful to the Basque Government by the support of this work through the project S-PE12UN015 and to the UPV/EHU by its support through the projects GIU10/01 and UFI11/07.

Referencias

[1] VLADISLAV AKHMATOV, 2002, Variable-Speed Wind Turbines with Doubly-Fed Induction Generators,*Wind Engineering*, vol. 26, n. 2, pp. 85–108.

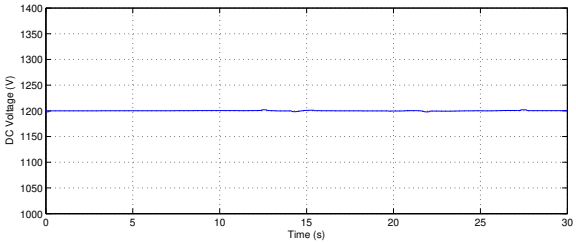


Figura 7: DC Voltage

[2] SIEGFRIED HEIER 1998 *Grid Integration of Wind Energy Conversion Systems* John Wiley & Sons Ltd, 1998, ISBN 0-471-97143-X.

[3] MULLER, S., DEICKE, M., DE DONCKER, R.W., 2002, Doubly fed induction generator system for wind turbines, *IEEE Industry Application Magazine*., **8**, pp. 26-33.

[4] G.M. JOSELIN HERBERT, S. INIYAN, E. SREEVALSAN, S. RAJAPANDIAN 2007, A review of wind energy technologies, *Renewable and Sustainable Energy Reviews*., **11**, pp. 1117–1145.

[5] WEI QIAO, WEI ZHOU, JOSÉ M. ALLER, AND RONALD G. HARLEY 2008, Wind Speed Estimation Based Sensorless Output Maximization Control for a Wind Turbine Driving a DFIG *IEEE Trans. on Power Electronics*, vol. 23, no. 3, pp. 1156-1169.

[6] D. W. NOVOTNY AND T. A. LIPO, 2000 *Vector Control and Dynamics of AC Drives*. Oxford, U.K.: Oxford Univ. Press, 2000.

[7] R. PENA, J. C. CLARE, AND G. M. ASHER 1996 Doubly fed induction generator using back-to-back PWM converters and its application to variablespeed wind-energy generation *Proc. Inst. Elect. Eng.*, vol. 143, no. 3, pp. 231–241.

[8] YAZHOU LEI, ALAN MULLANE, GORDON LIGHTBODY, AND ROBERT YACAMINI 2006, Modeling of the Wind Turbine With a Doubly Fed Induction Generator for Grid Integration Studies, *IEEE Trans. on Energy Conversion*, vol. 21, no. 1, pp. 257-264.

[9] B. Beltran, T. Ahmed-Ali, and M. H. Benbouzid, 2009, High-Order Sliding-Mode Control of Variable-Speed Wind Turbines, *IEEE Trans. Indus. Electro.*, **56**, 3314-3321.

[10] OSCAR BARAMBONES AND JOSE MARIA GONZALEZ DE DURANA 2011, An Adaptive

Sliding Mode Control Law for the Power Maximization of the Wind Turbine System, *IEEE International Conference on Power Engineering, Energy and Electrical Drives*. Torremolinos (Malaga) Spain.

[11] SLOTINE, J.J.E. AND LI, W., 1991, *Applied nonlinear control*. (Englewood Cliffs, New Jersey, USA: Prentice-Hall).

[12] BARAMBONES O. AND ETXEBARRIA V., 2002, Robust neural control for robotic manipulators, *Automatica*, **38**, 235-242.

[13] *SimPowerSystems 5. User's Guide*, The MathWorks

Plataforma para monitorización y control distribuidos basada en 802.15.4

Conrado Arquer, Luis Orihuela, Francisco R. Rubio
Dpto. Ingeniería de Sistemas y Automática, Universidad de Sevilla, España
conarqcob@alum.us.es, orihuela@us.es, rubio@us.es

Resumen

Este artículo presenta una plataforma basada en redes de sensores inalámbricas que se caracteriza por su modularidad y bajo consumo. Para mostrar su utilidad en sistemas de control, se han construido dos aplicaciones diferentes: monitorización remota de sistemas; y control distribuido de flotas de vehículos autónomos. La solución tecnológica utiliza dispositivos Waspmote de Libelium que se comunican a través del protocolo 802.15.4 (ZigBee).

Palabras Clave: Redes de sensores inalámbricas, Monitorización de sistemas, Control distribuido, Flotas de vehículos autónomos.

1. Introducción

Las redes de sensores inalámbricas, también conocidas por sus siglas en inglés WSN –*Wireless Sensor Networks*- están alcanzando en los últimos años una gran aceptación en la comunidad científica y en la sociedad en general [1]. El principal motivo para este mayor alcance es el abaratamiento de los sistemas microelectrónicos y de comunicación.

Aunque las tecnologías derivadas de las normas 802.11x son aplicables en WSN, son los protocolos de la familia 802.15.4 (la base de ZigBee) los que han logrado mejor encuadre, ya que implementan accesos al medio basados en contienda que acotan los retrasos de comunicación. Además, están pensados para reducir el consumo de energía, factor crucial en estos sistemas. Estas características los hacen adecuados para su uso en sistemas de control y en tiempo real.

Dentro de la comunidad del control, las WSN se han utilizado principalmente en dos grandes líneas. En primer lugar, cabe destacar su uso para la monitorización de sistemas [2]. Una línea similar es la que se encarga de la estimación distribuida de sistemas de gran tamaño [3] [4].

Por otro lado, es prolífica también la investigación relacionada con el control distribuido o descentralizado [5] [6]. En ambos casos, el uso de la red se hace necesario porque los diferentes elementos (sensores/actuadores) están localizados en posiciones remotas, y el cableado punto a punto conllevaría grandes costes. Además, algunos sistemas, como las flotas de vehículos autónomos, requieren el uso de este tipo de sistemas por la propia naturaleza de la aplicación [7].

En este trabajo de presenta una plataforma que permite ser utilizada tanto para monitorización de sistemas como para el control distribuido.

Para construir la red de sensores se han empleado las soluciones ofrecidas por Libelium, a través de sus Waspmotes [8]. Esta tecnología se caracteriza por su bajo consumo, facilidad de implementación y desarrollo, así como por su variedad de módulos que permiten añadir grandes capacidades sensoriales utilizando productos un mismo fabricante.

Se proponen dos aplicaciones. En primer lugar, se ha construido un sistema de monitorización remoto que permite acceder a las medidas de los diferentes sensores a través de un teléfono móvil. Este tipo de solución es interesante en sistemas que requieran la respuesta rápida de un operador ante diversas situaciones. Por ejemplo, en aplicaciones sanitarias o en determinados procesos críticos de fabricación.

En segundo lugar, dicha red de sensores se ha acoplado a una flota de vehículos, de forma que se dispone de un banco de trabajo para probar diferentes controladores distribuidos para flotas de vehículos.

El artículo está estructurado de la siguiente manera. A continuación se dan las principales características de los Waspmotes, así como sus posibilidades de creación de redes de sensores. Las secciones 3 y 4 detallan las aplicaciones de monitorización y control, respectivamente. Finalmente, se termina con algunas conclusiones y posibles trabajos de ampliación.

2. Redes de sensores usando Waspmotes

Esta sección presenta en primer lugar una descripción del hardware que se va a emplear. Seguidamente, se

detalla la creación de redes de sensores inalámbricas usando estos dispositivos.

1. Waspmote

Waspmote es una plataforma de código abierto para la construcción de redes de sensores inalámbricas que permite varios protocolos de comunicación (ZigBee, Bluetooth, GPRS) y cuyo objetivo principal es la implementación de modos de bajo consumo que permitan a los nodos sensoriales o *motes* comportarse de forma totalmente autónoma.

Los *motes* pueden operar en cuatro modos, como ilustra la Tabla 1. Cabe señalar el consumo mínimo del dispositivo de 0.06 micro amperios mientras se encuentra en el estado "Hibernate", un consumo mínimo inferior al de la mayoría de dispositivos que realizan funciones similares, véase como ejemplo Mica con un consumo mínimo superior a 1 micro amperio o CM5000 con un consumo mínimo de 1 micro amperio. Gracias a esto, los nodos sensoriales pueden llegar a vivir hasta 5 años de forma ininterrumpida.

Los Waspmotes se alimentan de una pequeña batería de litio que puede recargarse a través de un panel solar, lo que aumenta la autonomía de los dispositivos.

Tabla 1: Modos de operación en los que puede trabajar Waspmote

	Consumo	**Micro**	**Interrupciones aceptadas**
ON	15 mA	ON	Síncronas (Watchdog) y asíncronas
Sleep	55 µA	ON	Síncronas (RTC) y asíncronas
Deep Sleep	55 µA	ON	Síncronas y asíncronas
Hibernate	0.06 µA	OFF	Síncronas

Otra característica a destacar es su arquitectura modular. La placa de Waspmote (Figura 1) proporciona una serie de *sockets* en los cuales se conectan los diferentes módulos disponibles [8], ampliando la funcionalidad del nodo.

Cada una de estas placas incorpora sensores para la medición de parámetros en un ámbito específico. Por ejemplo la placa "GASES" incorpora sensores para medir el nivel de oxígeno, dióxido de carbono, monóxido de carbono, etc. De esta forma se podría controlar el nivel de polución en la ciudad, las emisiones de granjas, viveros y las resultantes de procesos industriales, o detectar incendios forestales.

Waspmote también dispone de dos sensores integrados en la placa base: un sensor de temperatura asociado al RTC (Real-Time Clock) y un acelerómetro de tres ejes (X,Y,Z). Finalmente, cabe señalar que dispone de un bus de comunicaciones I2C.

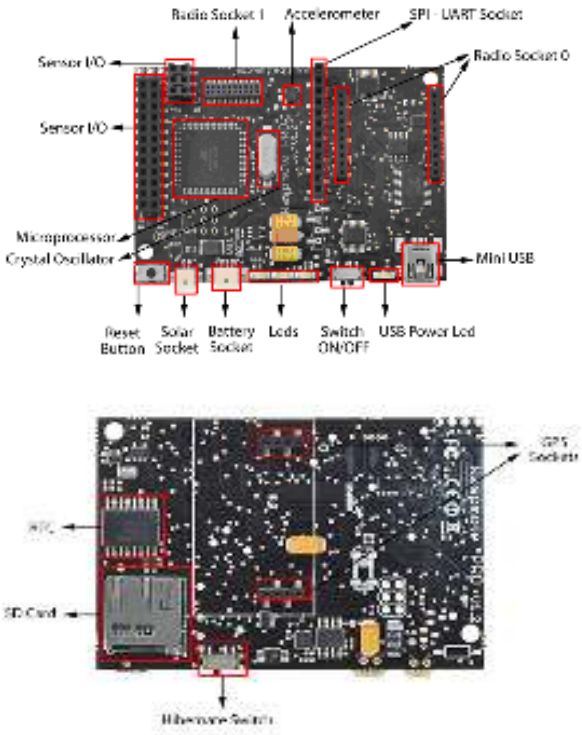


Figura 1: Vista frontal y trasera de la placa de Waspmote

Para su programación, Libelium proporciona un IDE que incorpora un compilador y una serie de librerías con funciones básicas. Este IDE está basado enteramente en el compilador y las librerías que proporciona Arduino en forma de código abierto.

2. Redes de sensores

Lo que hace más interesantes a los Waspmotes son sus capacidades de comunicación para formar redes de sensores inalámbricos [10]. Para que esta comunicación sea posible el Waspmote cuenta con un *socket* para la conexión de módulos XBee que se comunican con el microcontrolador a través de la UART.

Otro de los objetivos que persigue Libelium es el de permitir una comunicación de varios kilómetros de

distancia entre los dispositivos. Para ello se integraron módulos de comunicación de alta sensibilidad que utilizan sistemas inalámbricos de distintas frecuencias (2.4GHz, 868MHz, 900MHz) a fin de evitar interferencias con redes Wifi en zonas urbanas. Hay siete tipos de módulos XBee [8] y todos ellos se caracterizan por tener una alta sensibilidad de recepción y gran potencia de transmisión, con un rango de alcance que varía desde los 500 metros en el caso del módulo XBee-ZigBee (ver Figura 2) hasta los 12 Km en el caso del módulo XBee-868.



Figura 2: Módulo XBee-ZigBee

Cada módulo XBee puede funcionar como "Coordinador", "Router" o "End Device" dentro de la red.

- **Coordinador:** será el encargado de crear la red, esto es buscar un canal de frecuencia libre para la comunicación inalámbrica y establecer un identificador de red denominado PAN ID. Una vez seleccionado el canal y el PAN ID, el Coordinador permitirá a los Routers y End Devices unirse a la red para que puedan transmitir o recibir datos, o encaminarlos.
- **Router:** puede, una vez conectado a la red, permitir a otros Routers y End Devices unirse a la red así como ejercer tareas de encaminamiento, transmisión y recepción de datos.
- **End Device:** necesita de un padre, bien sea un Router o un Coordinador, para poder transmitir y recibir datos, sin embargo, es el único que puede "dormir", es decir, entrar en modo de bajo consumo. El padre deberá por tanto almacenar en un *buffer* los datos destinados al End Device mientras éste se encuentre dormido y entregárselos cuando despierte.

A través de la herramienta X-CTU se pueden programar los módulos para que funcionen de una de estas tres formas.

Con respecto a la topología, existen principalmente dos posibilidades: árbol (Figura 3) y anillo.

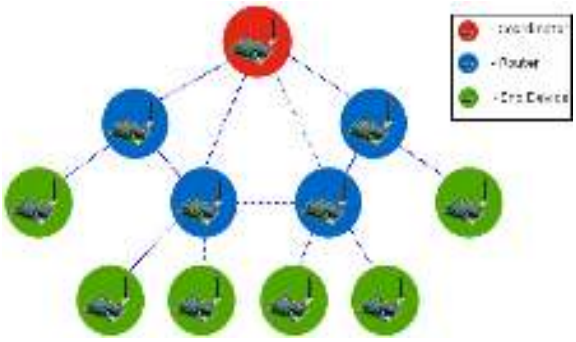


Figura 3: Ejemplo de topología en árbol

3. Aplicación: Monitorización de sistemas remoto

Para mostrar las utilidades que ofrece Waspmote y la posibilidad de integración con otras tecnologías se ha desarrollado una aplicación que consiste en monitorizar el estado de una red de sensores inalámbrica desde un dispositivo móvil. El objetivo es tener acceso desde un teléfono móvil a las medidas de los sensores en tiempo real. Para conectar los *motes* (ZigBee) con el teléfono (Wifi/3G/GPRS) es necesario utilizar un equipo pasarela, concretamente Meshlium [9].

Meshlium es un router que ejecuta un sistema Linux embebido y que cuenta con 5 interfaces de radio diferentes: Wifi 2.4 GHz, Wifi 5GHz, 3G/GPRS, Bluetooth y ZigBee. Además puede integrar un módulo GPS y ser alimentado por luz solar y batería. Meshlium, a diferencia de los routers convencionales, está diseñado para ser resistente al agua y soportar condiciones medioambientales adversas. Por otra parte, la potencia de transmisión de la interfaz Wifi es significativamente mayor que en los routers de uso doméstico pudiendo alcanzar localizaciones muy remotas.

Meshlium utilizará su interfaz ZigBee para convertirse en el nodo Coordinador de una red de Waspmotes. Una vez formada la red, Meshlium podrá recoger datos enviados por los *motes* y almacenarlos en una base local o externa o enviarlos a través de internet usando una conexión Wifi, Ethernet o 3G/GPRS [10].

Aprovechando la capacidad de Meshlium de funcionar simultáneamente como Coordinador de una red de *motes* y como punto de acceso Wifi, se ha decidido implementar una aplicación para un teléfono Android que permita capturar los valores medidos por los sensores del Waspmote y mostrarlos en la pantalla (Figura 4). Haciendo uso de Android SDK, que proporciona las librerías y herramientas de

desarrollo necesarias para la creación de aplicaciones para la plataforma Android, se ha implementado una aplicación que se conecta a Meshlium a través de su interfaz Wifi y le solicita información acerca un determinado nodo de la red de *motes*. Por su parte Meshlium, al recibir la petición de información, la traslada al *Mote* correspondiente haciendo uso de su interfaz ZigBee.

Para que todo esto sea posible, se ha programado una rutina en lenguaje C que el sistema embebido Linux ejecuta en *background.* Esta rutina se encarga de abrir la interfaz Wifi y se mantiene a la espera de peticiones por parte del dispositivo que las solicite, en este caso un Smartphone. Al recibir dicha petición, la procesará y la reenviará a través de la interfaz ZigBee al nodo del que se desee obtener la información. Este nodo obtendrá los valores requeridos de los sensores y se los entregará a Meshlium, el cual a su vez se los proporcionará al Smartphone que los mostrará en la pantalla. Para implementar la comunicación Wifi entre el Smartphone y Meshlium se hace uso del protocolo TCP/IP.

Cada uno de los Waspmotes por su parte tendrá programada una rutina que se mantendrá a la espera de peticiones por parte de Meshlium, leerá los valores de los sensores pertinentes, y los enviará a Meshlium a través de ZigBee.



Figura 4: Esquema de funcionamiento de la aplicación

La Figura 5 muestra algunas capacidades de la aplicación. A la izquierda se observa la pantalla principal que nos da acceso a cada uno de los *motes*. Las medidas que se obtienen de un *mote* en particular se reciben en el teléfono, tal y como se observa en la figura de la derecha.



Figura 5: Aplicación para la monitorización de sistemas remotos

4. Aplicación: Plataforma para el control de flotas de vehículos

La segunda aplicación que se ha realizado utilizando esta tecnología consiste en acoplar los *motes* a una serie de vehículos autónomos, en este caso pequeños coches impulsados por servomotores Pololu (Figuras 6 y 7).



Figura 6: Vista de la parte inferior del vehículo

El movimiento de estos vehículos en el plano XY se describe mediante el conocido modelo del uniciclo:

$$\dot{x} = F(t)\cos(\theta(t)),$$
$$\dot{y} = F(t)sen(\theta(t)),$$
$$\dot{\theta} = T(t),$$

donde x(t), y(t) y θ(t) son, respectivamente, las coordenadas y el ángulo de orientación. Las señales

de entrada F(t) y T(t) representan la velocidad linear y angular. Además de utilizarse para vehículos terrestres, el uniciclo se ha empleado para modelar submarinos autónomos del tipo deslizadores [7].



Figura 7: Vista lateral del vehículo con el *mote*

Las señales de entrada a los servos son generadas por el microcontrolador local mediante PWM, de forma que para realizar giros se les aplica una señal diferente a cada uno de los ejes de giro.

Además, el microcontrolador local debe comunicarse con el Waspmote (ver Figura 7), ya que será éste el encargado de generar las consignas de alto nivel. Para llevar a cabo dicha conexión entre micros, se utiliza el bus I2C de comunicación serie.

Las aplicaciones son variadas. Por ejemplo, haciendo uso del acelerómetro tri-axial incluido en Waspmote, se podrán detectar por ejemplo golpes y caídas, o cuándo el robot está subiendo y bajando cuestas.

El objetivo final que se persigue es coordinar una flota de vehículos para que maniobren en conjunto persiguiendo algún objetivo global (Figura 8).



Figura 8: Flota de vehículos autónomos

En este tipo de aplicaciones es de especial importancia mantener una correcta formación de los vehículos. Para ello, se hace indispensable conocer la posición de los robots, si no de forma absoluta, al menos su posición relativa a sus vecinos. Para resolver este problema, existen las siguientes soluciones:

- Sensores de ultrasonidos: miden la diferencia de tiempos entre la emisión y la recepción de un sonido. Aunque son bastante precisos, hay que tener en cuenta las zonas ciegas y las falsas alarmas.
- Sistemas de localización GPS: miden posiciones absolutas, pero con un margen de error que no es despreciable.
- Sistemas de visión artificial: de gran carga computacional, pueden obtener resultados muy adecuados.

La elección es muy dependiente del tipo de problema que quiera resolverse.

5. Conclusiones

Este artículo ha presentado las posibilidades que ofrece Libelium para desarrollar diversas aplicaciones basadas en redes de sensores. En primer lugar, se ha gestionado el acceso a una red de sensores inalámbricos mediante un *smartphone*. Finalmente, se ha construido una plataforma adecuada para el control de flotas de vehículos. En los siguientes meses, se probarán diversos algoritmos de control, como el propuesto por los autores en [11].

Agradecimientos

Los autores agradecen al Ministerio de Ciencia y Tecnología (DPI2010-19154) y la Junta de Andalucía (P09-AGR-4785) por financiar este trabajo. Igualmente, agradecen a ESIBot su colaboración en el desarrollo de los vehículos.

Referencias

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E. (2002), "Wireless sensor Networks: a survey", Computer Networks, 38, pp: 393-422.

[2] Martínez, M, Blasco, X, Herrero, J. M., Ramos, C., Sanchis, C. (2005), "Monitorización y Control de Procesos. Una Visión Teórico-Práctica Aplicada a Invernaderos", Revista Iberoamericana de Informática y Robótica Industriales, 2(4), pp: 5-24.

[3] Millán, P., Orihuela, L., and Jurado, I., Vivas, C., Rubio, F. R. (2013), "Distributed estimation in networked systems under periodic and event-based

communication policies", International Journal of Systems Science, DOI: 10.1080 /00207721.2013.775387.

[4] Olfati-Saber, R. (2007), "Distributed Kalman filtering for sensor networks", 46th IEEE Conference on Decision and Control, New Orleans, USA, pp: 5492-5498.

[5] Maestre, J. M., Muñoz de la Peña, D., Camacho, E. F. (2011), "Wireless sensor network analysis through a coalitional game: application to a distributed Kalman filter", International Conference on Networking, Sensing and Control, Delft, The Netherlands, pp: 227-232.

[6] Millán, P., Orihuela, L., Vivas, C., Rubio, F. R., Dimarogonas, D. V., Johansson, K. H. (2013), "Sensor-network-based robust distributed control and estimation", Control Engineering Practice, http://dx.doi.org/10.1016/j.conengprac.2013.05.002i

[7] Briñón-Arranz, L., Seuret, A., Canudas de Wit, C. (2009), "Translation control of a fleet circular formation of AUVs under finite communication range", 48th IEEE Conference on Decision and Control y 28th Chinese Control Conference, Shangai, China, pp. 8345–8350.

[8] Libelium Comunicaciones Distribuidas S.L. (2013), "Waspmote Technical Guide".

[9] Libelium Comunicaciones Distribuidas S.L. (2012), "Meshlium Technical Guide".

[10] Libelium Comunicaciones Distribuidas S.L. (2013), "Wireless Sensors Networks with Waspmote and Meshlium".

[11] Millán, P, Orihuela, L., Jurado, I., Rubio, F. R. (2013), "Formation control of autonomous underwater vehicles subject to communication delays", IEEE Transactions on Control Systems Technology, DOI: 10.1109/TCST.2013.2262768.

DETECCIÓN DE FALLOS CON VALIDACIÓN PROBABILÍSTICA

J. Blesa. Eng.Sistemes, Automàtica i Inf.Ind. U. Politecnica de Catalunya (joaquim.blesa@upc.edu)

A. Luque, T. Alamo. Dpto. Ingeniería de Sistemas y Automática. U. Sevilla (amalia,alamo@cartuja.us.es)

F. Dabbene. CNR-IEIIT. Politecnico di Torino. Italia. (fabrizio.dabbene@polito.it)

Resumen

Presentamos una estrategia general para el diseño de un bloque de detección de fallos con validación probabilística (PCV- Procesado, clasificación, validacion). Se propone un esquema general de PCV, que permite diseñar un bloque de detección de fallos con validación probabilística en el porcentaje máximo de fallos no detectados (impuesto como condición de diseño) y en el porcentaje de falsas alarmas (obtenido a posteriori). En cada iteración del algoritmo secuencial, una solución candidata se valida probabilísticamente mediante un conjunto de muestras generadas aleatoriamente. Presentamos un marco general en el que la solución candidata puede violar las restricciones para un reducido número de elementos del conjunto de validación. Este esquema generalizado muestra significativas ventajas, en particular en términos de la obtención de la solución probabilística.

Palabras clave: algoritmos aleatorios, detección de fallos, clasificadores, validación probabilística

1 INTRODUCCIÓN

Se entiende como fallo todo cambio en el comportamiento de alguno de los componentes del sistema (desviación no permitida de alguna de sus propiedades o parámetros característicos) de manera que éste ya no puede satisfacer la función para la cual ha sido diseñado ([4]). Además de los fallos, existen otros factores que alteran el comportamiento normal del sistema, como las perturbaciones y el ruido. Las perturbaciones son entradas no conocidas que pueden manifestarse en el sistema en cualquier momento pero que se han tenido en cuenta a la hora de diseñar el lazo de control convencional. Cualquier perturbación que no se haya tenido en cuenta en este diseño será considerada como un fallo. El ruido también es una entrada no conocida que se manifiesta en el sistema pero, a diferencia de las perturbaciones, tiene media nula y, además, a priori se puede tener conocimiento de cual es su amplitud. Un sistema de detección de fallos ha de reaccionar frente a los fallos y ser inmune (robusto), en la medida de lo posible, a los otros factores presentes en el sistema que generan incertidumbre. Por otro lado, muchos de los métodos de detección de fallos se basan en un modelo (matemático o cuantitativo) del sistema a monitorizar que nunca podrá describir de manera exacta el comportamiento del sistema real y por lo tanto presentará un error de modelado que también se deberá tener en cuenta.

El objetivo de un bloque de detección de fallos es, una vez se ha producido un fallo en un instante T_F, detectarlo en un intervalo de tiempo menor o igual a $T_{D_{max}}$ fijado previamente. Dependiendo de la magnitud e incidencia de los fallos que se deseen detectar y de la presencia de otros factores de incertidumbre en el sistema, no siempre será posible diseñar un bloque de detección que detecte todos los fallos sin que en situaciones de no fallo se activen falsas alarmas. Así que siempre existirá un compromiso entre la proporción de fallos que no se detecten (MF "Missed Faults") y la proporción de veces que se active el bloque detector sin la presencia de fallos debido a los factores de incertidumbre presentes en el sistema (FA "False alarms"). En este compromiso que se deberá tener en cuenta en el proceso de diseño del bloque detector de fallos es lógico priorizar la minimización de fallos no detectados respecto a la minimización de falsas alarmas.

La naturaleza aleatoria de los fallos y las incertidumbres inherentes del sistema convierten el problema de diseño del bloque de detección en un problema de robustez.

Típicamente, para un problema de robustez, los parámetros de diseño, así como diferentes variables auxiliares, son descritos en términos de un vector de variables de decisión θ, que se denota como "parámetro de diseño", y es restringido al conjunto Θ. Por otro lado, la incertidumbre w está acotada en el conjunto $\mathcal{W}$. Es decir, cada elemento $w \in \mathcal{W}$ representa una de las realizaciones admisibles de la incertidumbre, con probabilidad $\Pr_{\mathcal{W}}$. En nuestro contexto de detección de fallos, θ corresponde a las variables de decisión que determinan el bloque de detección de fallos. Dicho bloque permite determinar si hay un fallo o no en

un determinado escenario, por lo tanto tendremos dos conjuntos de incertidumbre $\mathcal{W}_F$ y $\mathcal{W}_N$ que consisten en todos los posibles escenarios de funcionamiento del sistema a monitorizar con fallo y sin fallo respectivamente. Por otro lado, w_F y w_N representan una realización de un escenario con fallo y sin fallo. $\mathcal{W}_F$ y $\mathcal{W}_N$ tienen asociados unos espacios de probabilidad $\Pr_F$ y $\Pr_N$ respectivamente.

Además consideramos también dos funciones binarias medibles:

$$g(\theta, w) := \begin{cases} 0 & \text{si } \theta \text{ detecta fallo} \\ 1 & \text{en otro caso.} \end{cases}$$

$$h(\theta, w) := \begin{cases} 0 & \text{si } \theta \text{ no detecta fallo} \\ 1 & \text{en otro caso.} \end{cases}$$

Al aplicar estas dos funciones sobre los espacios $\mathcal{W}_F$ y $\mathcal{W}_N$ se obtienen las siguientes esperanzas

$$E_g(\theta) := \Pr_F\{w_F \in \mathcal{W}_F : g(\theta, w_F) = 1\}$$

$$E_h(\theta) := \Pr_N\{w_N \in \mathcal{W}_N : h(\theta, w_N) = 1\}.$$

Donde $E_g(\theta)$ y $E_h(\theta)$ son el tanto por uno de fallos no detectados (MF) y falsas alarmas (FA) respectivamente. La utilidad de los algoritmos aleatorios surge del hecho de poder tratar el siguiente problema de diseño

$$\min_{\theta \in \Theta} E_h(\theta) \text{ sujeto a } E_g(\theta) \leq \eta_F \qquad (1)$$

donde η_F es el tanto por uno máximo de fallos no detectados impuesto como requerimiento del bloque detector.

En este marco, se pueden extraer N_N y N_F i.i.d. muestras (independientes e identicamente distribuidas) $\{w_N^{(1)}, \ldots, w_N^{(N_N)}\}$ de $\mathcal{W}_N$ y $\{w_F^{(1)}, \ldots, w_F^{(N_F)}\}$ de $\mathcal{W}_F$ de acuerdo a la probabilidad $\Pr_N$ y $\Pr_F$ respectivamente y con una proporción entre escenarios de fallo y no fallo $F_N = \frac{N_F}{N_N}$ determinada por la probabilidad de fallo del sistema a monitorizar. De esta manera se puede resolver el siguiente problema de optimización muestreado

$$\min_{\theta \in \Theta} \sum_{\ell_N=1}^{N_N} h(\theta, w_N^{(\ell_N)}) \qquad (2)$$

$$\text{sujeto a } \sum_{\ell_F=1}^{N_F} g(\theta, w_F^{(\ell_F)}) \leq \eta_F N_F$$



Figura 1: Esquema del detector de fallos en linea

La idea de permitir algunas violaciones de las restricciones no es nueva y puede encontrarse, por ejemplo, en el contexto de identificación [3].

En este artículo se propone un método de diseño del bloque detector de fallos basado en la utilización de históricos o simulaciones de episodios reales con fallo y sin fallo evitando la dificultad del análisis, que no siempre es posible, debido a la complejidad del problema.

El resultado así obtenido, mediante un test de validacion probabilística, garantiza que la solución propuesta se comporta de la manera deseada con una cierta probabilidad, fijada a priori. Se garantiza asimismo la satisfacción probabilística de las restricciones. Esta técnica resulta muy adecuada para el abordaje de problemas complejos.

Finalmente este artículo ilustra la metodología propuesta con la presentación de una aplicación, en este caso, al diseño de un detector de fallos con garantía probabilística en un depósito virtual que modela el comportamiento de un colector de una red de alcantarillado tal y como se detalla en [8].

2 ESQUEMA DE DISEÑO PROPUESTO

Tal y como se ha comentado anteriormente, el objetivo de este artículo es el de proponer un esquema que permita diseñar un bloque de detección de fallos con validación probabilística en el porcentaje máximo de fallos no detectados (impuesto como condición de diseño) y en el porcentaje de falsas alarmas (obtenido a posteriori). El funcionamiento de este bloque de detección de fallos se describe en el esquema de la Figura 1. De los datos de entrada/salida del sistema a monitorizar cada instante de tiempo j se extraen unos atributos o indicadores $(r_j^{(1)}, r_j^{(2)}, ..., r_j^{(n_r)})$ de éste mediante un bloque de procesado que actúa en horizonte deslizante en el tiempo. Estos indicadores que pueden ser de distinta naturaleza son sensibles a la presencia de fallos pero también al ruido, perturbaciones y otros factores como errores de modelado. Por esta razón son introducidos en un clasificador que determina si se ha producido un fallo o no en el sistema.

Por otro lado, el algoritmo del proceso fuera de linea para lograr diseñar el bloque detector se des-

cribe en el esquema de la Figura 2. En primer lugar se diseña el bloque de procesado o lo que es lo mismo se determina que indicadores serán útiles con el fin de detectar los fallos presentes en el sistema. Para este paso se debe utilizar el conocimiento de la planta y técnicas básicas de detección de fallos [5], [6]. La naturaleza de los indicadores a utilizar puede ser muy diversa : residuos entre medidas y estimaciones, señales características para detectar fallos como vibraciones, señales sonoras, etc..., magnitudes que determinen un cambio de operación de la planta como la temperatura, humedad, etc...

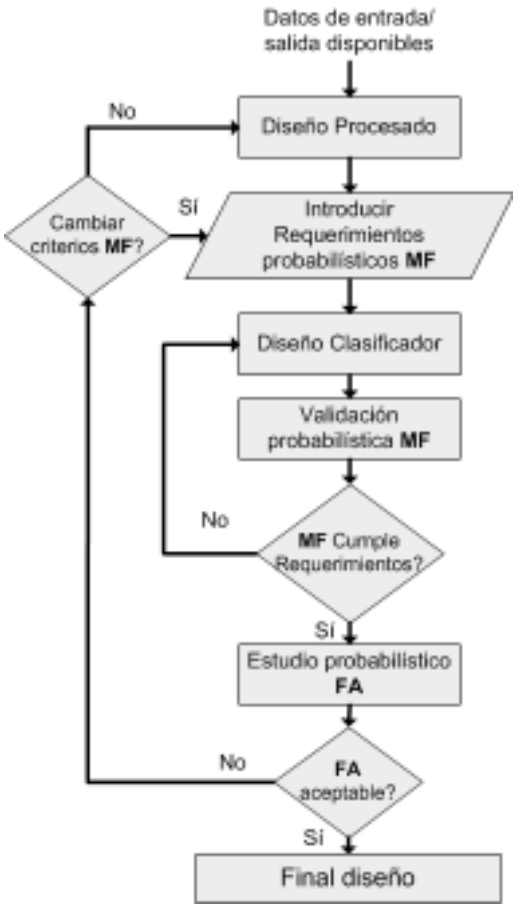


Figura 2: Esquema diseño fuera de linea del bloque de detección de fallos

Una vez determinados los indicadores que se quieren extraer de la información disponible de la planta, se deben especificar los requerimientos probabilísticos de la cota máxima en porcentaje de fallos no detectados (MF) y la garantía probabilística mínima de cumplimiento. A continuación se procede a diseñar un clasificador al que se le suministran datos de la planta en situaciones de funcionamiento con y sin fallo (reales o simuladas). Mediante un proceso de optimización se minimiza el número de falsas alarmas (FA) y se impone (como restricción) un número máximo de fallos no detectados (MF) se calibra adecuadamente el clasificador. A continuación se evalúa el clasificador obtenido mediante un test de validación probabilística con datos diferentes a los utilizados en el diseño de calibración del clasificador. Si no se pasa el test de validación probabilística se vuelve a la etapa de diseño del clasificador. Si se pasa el test de validación probabilística, el clasificador cumple las especificaciones de MF con las garantía probabilísticas impuestas y se procede a un estudio probabilístico del número de falsas alarmas. Si el resultado no és satisfactorio (número de falsas alarmas elevado) hay dos opciones: cambiar (relajar) los requerimientos probabilísticos de MF o mejorar el diseño de procesado (obtener mejores indicadores).

3 DISEÑO DEL PROCESADO

Como se ha comentado anteriormente, los indicadores extraídos del sistema a monitorizar pueden ser de diferente naturaleza. En este apartado se ponen dos ejemplos de posibles indicadores que se utilizan en los métodos de detección de fallos basada en modelos matemáticos. En este tipo de métodos se comprueba la consistencia o no consistencia del modelo del sistema a monitorizar con las medidas obtenidas de dicho sistema.

Por ejemplo, si consideramos que el sistema a monitorizar se puede describir mediante el modelo de regresión en tiempo discreto

$$y(j) = \varphi^T(j)\beta_0 + e(j), \;\; j = 1, \ldots, M \qquad (3)$$

donde

- $y(j)$ es la medida de la salida
- $\varphi(j)$ es el vector de regresión de dimensión n_β función de las entradas $u(j)$ y salidas $y(j)$
- β_0 es el vector de parámetros nominales de dimensión n_β
- $e(j)$ es el error aditivo que contiene el ruido de las medidas y el error de modelado

Entonces esta consistencia se puede evaluar calculando en cada instante j el residuo $\Delta(j)$ entre la salida medida y la estimación dada por el modelo

$$\Delta(j) = y(j) - \varphi^T(j)\beta_0. \qquad (4)$$

En un caso ideal, el residuo debería de ser diferente de cero solo en el caso de que un fallo estuviera

presente en el sistema. Sin embargo, debido a la presencia de ruido y error de modelado el residuo puede ser diferente de cero cuando no haya presencia de fallo y el método de detección debe de ser robusto. Una manera de abordar este problema es aplicar las técnicas Set-Membership que consideran el error $e(j)$ desconocido pero acotado ([7]), esto es

$|e(j)| \leq \sigma \; j = 1, \ldots, M.$

De esta manera se puede aplicar el siguiente test de detección de fallos

$$\begin{cases} \text{Si } |\Delta(j)| \leq \sigma \Rightarrow \text{NoFallo} \\ \text{Si } |\Delta(j)| > \sigma \Rightarrow \text{Fallo.} \end{cases}$$

Y por lo tanto el diseño del diagnosticador consiste en elegir una cota σ adecuada. A este test se le denomina test directo.

Otra manera de mirar la consistencia del modelo con las medidas en el espacio de parámetros, mediante una ventana temporal de N muestras se puede hacer una estimación de parámetros

$$\hat{\beta}(j) = \left(\Phi^T(j)\Phi(j)\right)^{-1}\Phi^T(j)Y(j) \qquad (5)$$

donde $\Phi(j) = \begin{pmatrix} \varphi^T(j-N) \\ \vdots \\ \varphi^T(j) \end{pmatrix}$ y

$Y(j) = \begin{pmatrix} y(j-N) \\ \vdots \\ y(j) \end{pmatrix}$

y de la misma manera que en el residuo temporal, se puede calcular un residuo de parámetros respecto a un modelo nominal

$\Delta\beta(j) = \hat{\beta}(j) - \beta_0$

y así definir el siguiente test de detección

$$\begin{cases} \text{Si } \Delta\beta(j) \in \text{B} \Rightarrow \text{NoFallo} \\ \text{Si } \Delta\beta(j) \notin \text{B} \Rightarrow \text{Fallo.} \end{cases}$$

donde B es el conjunto de incertidumbre de parámetros debido al error aditivo $e(j)$ y a la poca riqueza de los datos usados en la identificación.

Tanto el test directo como el test inverso se pueden incluir en el esquema general 1 incluyendo los términos procesando formular en forma de clasificador elegiendo $r_j^{(1)} = \Delta(j)$ para el test directo y $r_j^{(2)} = \Delta\beta(j)$ para el test inverso. Además tests básicos como valores máximos, fijos o variaciones máximas en los datos obtenidos en los sensores se pueden implementar escogiendo los indicadores adecuados.

4 DISEÑO DEL CLASIFICADOR

Con el fin de conseguir una buena discriminación entre los escenarios sin fallo y los escenarios con fallo, se propone utilizar un clasificador basado en dos etapas: una estática y otra dinámica. Tal y como se muestra en la Figura 3.

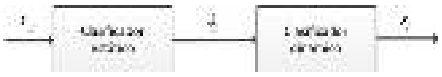


Figura 3: Esquema del clasificador propuesto

El objetivo del clasificador estático es determinar en un instante de tiempo j con el vector de indicadores en ese mismo instante de tiempo r_j si la situación es sintomática de fallo o no mediante una señal binaria ϕ_j. Si la situación es sintomática de fallo $\phi_j = 1$ y si no $\phi_j = 0$. Esta operación se podría realizar mediante una función de entrada vectorial analógica y salida binaria h_{est}

$\phi_j = h_{est}\left(r_j\right)$

donde $r_j^T = (r_j^{(1)}, r_j^{(2)}, ..., r_j^{(n_r)})^T$.

El objetivo del clasificador dinámico de orden T_p es el de determinar en el instante j si se ha producido un fallo o no con las últimas T_p señales sintomáticas. Esto es:

$F_j = h_{din}\left(\phi_j, \phi_{j-1}, \cdots, \phi_{j-T_P+1}\right).$

Donde h_{din} es una función de entrada y salidas binarias y, por la definición de detectabilidad, se cumple que $T_P \leq T_{D_{\max}}$. El motivo de dividir en dos partes el clasificador es el de hacerlo lo más robusto posible respecto a las falsas alarmas sin empeorar las prestaciones de detectabilididad (fallos no detectados).

Para diseñar estos dos clasificadores se dispondrá de escenarios sin fallos y con fallos $\{w_N^{(1)}, \ldots, w_N^{(N_N)}\}$ de $\mathcal{W}_N$ y $\{w_F^{(1)}, \ldots, w_F^{(N_F)}\}$ de $\mathcal{W}_F$ respectivamente. Cada escenario consiste en una secuencia de indicadores ${r_1}^{(i)}, {r_2}^{(i)}, \cdots, {r_{T_{Si}}}^{(i)}$ donde T_{Si} es el número de instantes de tiempo que dura el escenario i. El conjunto de todos indicadores de todos los escenarios se puede dividir en dos suconjuntos: el conjunto de los indicadores que están contaminados por un fallo y el conjunto de indicadores que están libres de fallos $\Re_F$ y $\Re_N$ respectivamente. Se cumple que todos los indicadores que pertenecen a un escenario libre de fallo pertenecen al conjunto $\Re_N$, mientras que en los indicadores que pertenecen a un escenario con fallo parte de ellos están contaminados por el efecto de un fallo (y por lo tanto pertenecen a $\Re_F$) y parte de ellos están libres de efectos de fallo (y por lo tanto pertenecen a $\Re_N$).

El objetivo del clasificador estático es discriminar lo máximo posible entre los indicadores pertenecientes a los dos conjuntos ($\Re_F$ y $\Re_N$). Esto se puede conseguir mediante la elección adecuada de un vector λ que cumpla lo mejor posible las siguientes restricciones

$$f^T(r)\lambda > 0 \Rightarrow r \in \Re_F$$
$$f^T(r)\lambda \leq 0 \Rightarrow r \in \Re_N \ ,$$

donde $f(r)$ es una expresión vectorial de r. El valor de λ se puede hallar mediante el siguiente problema de optimización convexa

$$\min_{\lambda} \quad \left(\sum_{r \in \Re_F} e^{-\tau f^T(r)\lambda} + \sum_{r \in \Re_N} e^{f^T(r)\lambda} \right) \qquad (6)$$

donde $\tau \in (0, \infty)$ es una constante que se ha determinado previamente a la resolución del problema de optimización y permite penalizar en mayor o menor medida los errores de clasificación de los indicadores $r \in \Re_F$ respecto a los errores de clasificación de indicadores $r \in \Re_N$ y por lo tanto priorizar más o menos el comportamiento del sistema respecto a los fallos no detectados o respecto a las falsas alarmas. En concreto, cuanto mayor sea τ más penalización se dará a los indicadores clasificados como situación normal, o sea fallos no detectados, y por lo tanto menos peso dará a las falsas alarmas.

Como se ha comentado anteriormente, en el proceso de diseño del detector de fallos se impondrá un valor máximo de fallos no detectados (MF definido por η_F) y esto se conseguirá eligiendo un τ adecuado. El problema del clasificador estático es que al tomar la decisión en función de los indicadores en un determinado instante, la imposición de bajo número de fallos implique una gran sensibilidad a las incertidumbres del sistema y se traduzca en un elevado número de falsas alarmas. El segundo clasificador tiene como objetivo filtrar el efecto de estas incertidumbres y permitir un diagnóstico lo más robusto posible utilizando las T_P últimas salidas del diagnosticador estático. Como la función del diagnosticador h_{din} és una función de entrada binaria y salida binaria con 2^{T_P} combinaciones, éstas se pueden probar de manera exhaustiva y encontrar cual presenta mejores prestaciones entre todas.

Por lo tanto el diseño del clasificador se reduce en hallar el mínimo valor de τ que cumple con el máximo número de falsas alarmas permitidas. Al bloque de detección de fallos ebtenido lo denominaremos θ.

5 VALIDACIÓN

5.1 Algoritmos secuenciales con validación probabilística en MF

En esta sección se presenta una familia general de algoritmos aleatorios, que llamamos algoritmos SPV (del inglés "**S**equential **P**robabilistic **V**alidation algorithms"), ver [1].

La principal característica de este tipo de algoritmos es que están basados en un paso de validación probabilística.

Cada iteración de un algoritmo SPV se compone del cálculo de una solución candidata para el problema y un paso de validación. Los resultados son fundamentalmente independientes de la estrategia concreta elegida para obtener las soluciones candidatas.

El objetivo de utilizar estos algoritmos en este contexto es el de garantizar un cota máxima en de fallos no detectados (MF) definidos por η_F con una la garantía probabilística mínima de cumplimiento $1 - \delta_F$.

La precisión $\eta_F \in (0, 1)$ y confianza $\delta_F \in (0, 1)$ requeridas para la solución probabilística juegan un papel relevante en la determinación del tamaño muestral (número de escenarios con fallo a evaluar) para cada paso de iteración. Se puede proponer un esquema de validación que garantice que para una precisión η_F y confianza δ_F dadas (impuestas en el esquema de diseño), todas las soluciones probabilísticas obtenidas mediante el algoritmo SPV tienen una probabilidad de violación no mayor que η_F con probabilidad no menor que $1 - \delta_F$.

Cada iteración del algoritmo es enumerada mediante el entero k. Llamamos m_k al número de violaciones permitidas en el paso de validación de iteración k. Asumimos que m_k es dado como función de k, es decir, $m_k = m(k)$ donde la función $m : \mathbb{N} \to \mathbb{N}$ es dada. Denotamos M_k el tamaño muestral para el paso de validación de la iteración k. Asumimos que M_k es dado como función de k, η_F y δ_F. Es decir, $M_k = M(k, \eta_F, \delta_F)$ donde $M : \mathbb{N} \times \mathbb{R} \times \mathbb{R} \to \mathbb{N}$ tiene que ser apropiadamente diseñada para garantizar las propiedades probabilísticas del algoritmo. Denotamos las funciones $m(\cdot)$ y $M(\cdot, \cdot, \cdot)$ como *función de nivel* y *función de cardinalidad* respectivamente.

Estructura de un algoritmo SPV:

(i) Fijar $\eta_F \in (0, 1)$ y $\delta_F \in (0, 1)$ a los valores deseados. Poner $k = 1$.

(ii) Obtener una solución candidata $\hat{\theta}_{F_k}$ al problema de optimización robusta (1).

(iii) $m_k = m(k)$ y $M_k = M(k, \eta_F, \delta_F)$.

(iv) Obtener un conjunto de validación $\mathcal{V}_k = \{v^{(1)}, \ldots, v^{(M_k)}\}$ extrayendo M_k i.i.d muestras de validación de $\mathcal{W}_F$ según la probabilidad $\Pr_{\mathcal{W}_F}$.

(v) Si $\sum_{\ell=1}^{M_k} g(\hat{\theta}_{F_k}, v^{(\ell)}) \leq m_k$, entonces $\hat{\theta}_{F_k}$ es una solución probabilística.

(vi) Salir si se satisface la condición de salida.

(vii) $k = k + 1$. Ir a (ii).

La Figura 4 muestra la parte del esquema de diseño general propuesto que corresponde al algoritmo SPV donde se indica los pasos del algoritmo que corresponden a cada bloque.

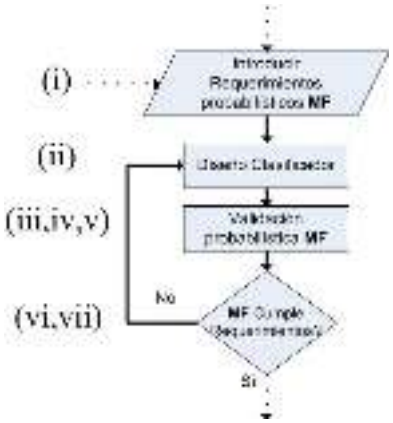


Figura 4: Esquema validación probabilística fallos no detectados

Aunque la condición de salida del algoritmo puede ser bastante general, una estrategia razonable es salir después de un número dado de soluciones candidatas hayan sido clasificadas como soluciones probabilísticas o si se supera un cierto tiempo computacional desde el comienzo del algoritmo. Después de salir se puede elegir entre las soluciones probabilísticas aquella que optimice un determinado índice de desempeño. En la siguiente subsección se propone una estrategia para elegir la cardinalidad del conjunto de validación en la iteración k de modo que, con probabilidad no menor que $1 - \delta_F$ todas las soluciones clasificadas como soluciones probabilísticas por el algoritmo satisfagan la precisión η_F.

5.2 Tamaño muestral

Consideremos un algoritmo SPV con precisión $\eta_F \in (0, 1)$, confianza $\delta_F \in (0, 1)$ y función de nivel $m(\cdot)$ dadas. Entonces, la función de cardinalidad

$$M(k, \eta_F, \delta_F) = \left\lceil \frac{1}{\eta_F} \left(m(k) + \ln \frac{1}{\delta_F \mu(k)} + \sqrt{2m(k) \ln \frac{1}{\delta_F \mu(k)}} \right) \right\rceil,$$

donde

$$\mu(k) = \frac{1}{\xi(\alpha) k^\alpha},$$

donde $\xi(\cdot)$ es la función zeta de Riemann, y $\alpha > 1$,

garantiza que, con probabilidad mayor que $1 - \delta_F$ todas las soluciones probabilísticas obtenidas mediante el algoritmo SPV tienen una probabilidad de violación (fallos no detectados) no mayor que η_F [1]. La función $\mu(k)$ puede adoptar otras expresiones, ver [1].

5.3 Falsas alarmas

Una vez diseñado un clasificador que cumple los criterios probabilísticos respecto a los fallos no detectados, se puede utilizar el resultado obtenido en [2] donde se proporciona la complejidad muestral que caracteriza cómo una media empírica converge en probabilidad a la verdadera probabilidad de violación (en este caso falsas alarmas), para determinar la precisión $\eta_N \in (0, 1)$ y confianza $\delta_N \in (0, 1)$ en términos de falsas alarmas (FA).

Dados $\hat{\theta} \in \Theta$ hallado en el proceso de diseño del bloque de detección y un conjunto de N_N escenarios libres de fallo, se cumple

$$N_N \geq \frac{\ln \frac{1}{\delta_N}}{(\sqrt{\eta_N} - \sqrt{\rho_N})^2}$$

con $0 \leq \rho_N < \eta_N < 1$ donde ρ_N es la proporción de falsas alarmas obtenidas al aplicar el bloque de detección diseñado a los N_N escenarios libres de fallo disponibles

$$\rho_N = \frac{\text{Número de falsas alarmas}}{N_N}.$$

entonces

$\Pr_{\mathcal{W}^N}\{\mathrm{w} \in \mathcal{W}^N \ : \ \hat{E}(\hat{\theta}, \mathrm{w}) \leq \rho_N \text{ y } E(\hat{\theta}) > \eta_N\} \leq \delta_N$.

Por lo tanto, si nuestra estimación de falsas alarmas $\hat{E}(\hat{\theta}, \mathrm{w})$ está dada por el valor ρ_N, se garantiza que la discrepancia entre este valor y la probabilidad de que el número real de falsas alarmas $E(\hat{\theta})$

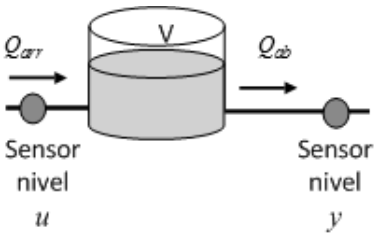


Figura 5: Depósito virtual

sea mayor que otro valor $\eta_N > \rho_N$ fijado a priori, está acotada por δ_N.

Si imponemos una δ_N deseada, la garantía estadística η_N queda determinada por las inecuaciones anteriores. En concreto

$$\eta_N = \left(\left(\frac{\ln(\frac{1}{\delta_N})}{N_N} \right)^{\frac{1}{2}} + \sqrt{\rho_N} \right)^2 \qquad (7)$$

6 RESULTADOS

Para demostrar la eficacia de la metodología presentada en este artículo se ha aplicado al diseño de un bloque de detección de fallos con garantía probabilística en el depósito virtual representado en la Figura 5 que modeliza el comportamiento de un colector de una red de alcantarillado tal y como se detalla en [8].

Se han tenido en cuenta fallos en los sensores de entrada y de salida ($f_u(t)$, $f_y(t)$) y fallos paramétricos ($f_a(t)$, $f_b(t)$) tal y como se indica en la Figura 6. El comportamiento del sistema real considerando estos fallos se puede describir mediante el siguiente modelo en tiempo discreto

$$\tilde{y}(j) = (\tilde{a} + f_a(j-1))\,\tilde{y}(j-1) + \left(\tilde{b} + f_b(j-1)\right)\tilde{u}(j-1) + e_d(j)$$

donde

- $\tilde{a}$, $\tilde{b}$ son los parámetros reales del sistema
- $\tilde{u}(j)$, $\tilde{y}(j)$ son la entrada y salida reales del sistema
- $e_d(j)$ es el error de discretización que dependerá del tiempo de muestreo
- $e_u(j)$, $e_y(j)$ son los errores aditivos introducidos por los sensores

Así pues, con las medidas disponibles

$$y(j) = \tilde{y}(j) + f_y(j) + e_y(j)$$
$$u(j) = \tilde{u}(j) + f_u(j) + e_u(j)$$



Figura 6: Posibles fallos en el sistema depósito virtual

se puede utilizar el siguiente modelo para describir el comportamiento del sistema real

$$y(j) = a_0 y(j-1) + b_0 u(j-1) + e(j) \qquad (8)$$

donde

- a_0, b_0 son los parámetros del modelo obtenidos con datos de entrada salida sin fallo del sistema
- $e(j)$ es el error del modelo que tiene en cuenta los errores $e_d(j)$, $e_u(j)$, $e_y(j)$ y la posible discrepancia entre los parámetros reales ($\tilde{a}$,$\tilde{b}$) y del modelo (a_0,b_0)

Con el fin de obtener los datos del funcionamiento normal y con fallo se ha implementado un simulador en el entorno Matlab-Simulink, donde se han introducido escenarios (con y sin fallo) basados en situaciones reales, teniendo posibles errores entre el modelo y sistema real. El objetivo es diseñar un detector de fallos que cumpla $MF < 1\%$ con una probabilidad no menor del $1-10^{-6}$ ($\eta_F = 0.01$ $\delta_F = 10^{-6}$) con un retardo máximo en la detección del fallo de $T_{D_{max}} = 5$ muestras.

En primer lugar, mediante los datos obtenidos de entrada/salida del proceso se ha estudiado cuales son los indicadores que nos permitirían distinguir entre las situaciones de fallo y de funcionamiento normal. Las Figuras 7 y 8 muestran los indicadores de error en la identificación de parámetros $r_j^{(1)} = (\Delta a(j), \Delta b(j))^T$ ($\Delta a(j) = \hat{a}(j) - a_0$ y $\Delta b(j) = \hat{b}(j) - b_0$) tomando una ventana temporal de 50 muestras, para escenarios con fallo y sin fallo respectivamente. Además, en estas gráficas también se muestra la región $f^T(r)\lambda \leq 0$ obtenida al diseñar un clasificador óptimo. Sólo con este indicador se pueden detectar el 65% de los fallos con un nivel de Falsas alarmas muy bajo (0.001%).

Una de las ventajas del uso de clasificadores es que, con el fin de mejorar la discriminación entre escenarios con y sin fallo, se pueden añadir otros indicadores. En nuestro caso además de indicador descrito anteriormente han añadido otros indicadores como el mismo error en la estimación de

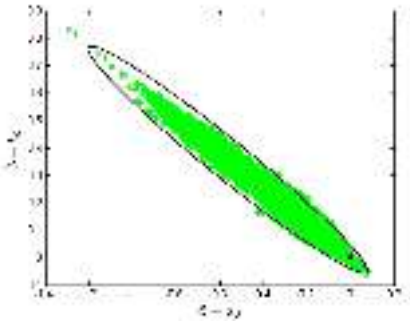


Figura 7: Indicador residuo de parámetros en escenarios sin fallo y conjunto $f^T(r)\lambda \leq 0$

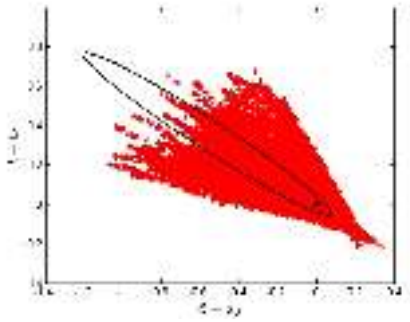


Figura 8: Indicador residuo de parámetros en escenarios con fallo y conjunto $f^T(r)\lambda \leq 0$

parámetros pero con ventanas de 15 y 100 muestras, el residuo temporal (3) utilizando el modelo dado por (6) y acumulando este residuo con ventanas de 15, 50 y 100 muestras. Así como valores máximos en las medidas determinados por el calado máximo admisible en el colector.

Una vez decididos los indicadores a utilizar por el clasificador, se ha utilizado el Algoritmo 2 para hallar el clasificador que cumpla los requisitos probabilísticos en MF definidos anteriormente (a priori) obteniendo $\tau = 7.1$. Al aplicar el clasificador diseñado al conjunto de escenarios sin fallo del sistema se obtiene mediante (7), a posteriori, un nivel de falsas alarmas $FA < 4.5\%$ con una probabilidad no menor del $1-10^{-6}$ ($\eta_N = 0.045\, \delta_N = 10^{-6}$)

7 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha propuesto una metodología general para el diseño de detectores de fallos con garantía probabilística. La gran ventaja de la metodología propuesta es, por un lado su flexibilidad en introducir herramientas de diferente tipo en la detección de fallos y por otro la garantía probabilística certificada del detector propuesto. El funcionamiento de esta metodología se ha ilustrado con su aplicación a un ejemplo de depósito virtual. Como trabajo futuro, siguiendo la filosofía del esquema propuesto, sería interesante abordar el problema de diseñar un diagnosticador que permita determinar, una vez que se ha detectado un fallo, que tipo de fallo se ha producido con una determindad garantía probabilística.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el proyecto DPI-2011-26243. Los autores quieren agradecer a Vicenç Puig y a Roberto Tempo por sus aportaciones a la investigación presentada en este trabajo.

Referencias

[1] T. Alamo, A. Luque, D. R. Ramirez and R. Tempo. Randomized control design through probabilistic validation. *Proceedings of the American Control Conference*, Montreal, Canada, 2012.

[2] T. Alamo, R. Tempo and A. Luque. On the sample complexity of randomized approaches to the analysis and design under uncertainty. *Proceedings of the American Control Conference*, Baltimore, USA, 2010.

[3] E. Bai, H. Cho, R. Tempo and Y. Ye. Optimization with few violated constraints for linear bounded error parameter estimation. *IEEE Transactions on Automatic Control*, 47(7):1067–1077, 2002.

[4] Blanke, M. Fault-tolerant control systems. *New Trends in Advanced Control* Springer-Verlag, 1999

[5] R. Isermann. Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. Berlin, Germany: Springer, 2006

[6] R. Isermann Fault-Diagnosis Applications. Springer-Verlag, Berlin, Heidelberg, 2011

[7] M. Milanese, J. Norton, H. Piet Lahanier, E. Walter. Bounding approaches to system identification. *Plenum Press*. New York, USA, London, UK, 1996

[8] Puig, V.; Blesa, J. Limnimeter and Rain Gauge FDI in Sewer Networks using an Interval Parity Equations based Detection Approach and an Enhanced Isolation Scheme. *Control engineering practice* 21 (2), 146-170.

CONTROL DE ACTITUD DE UN COHETE DE SONDEO ATMOSFERICO

Pau Manent Bistué
pau.manent@bistue.es

Joseba Jokin Quevedo Casín
Joseba.Quevedo@upc.edu

Bernardo Morcego Seix
bernardo.morcego@upc.edu

Resumen

Un cohete necesita de un sistema de control de actitud para superar posibles perturbaciones. Sobre un cohete pueden aparecer fuerzas que lo desestabilicen de una trayectoria idealmente vertical: viento, desalineamiento de las fuerzas de empuje, asimetría de la carga, etc. Éstas y muchas más son las perturbaciones de un cohete, y para evitar que éste se desestabilice y caiga o explote se requiere de un sistema que controle su actitud.

En el presente estudio se opta por un sistema de control activo basado en modificar la dirección del vector empuje simbolizado con TVC (Thrust Vector Control).

El estudio está dividido en dos partes: una primera parte en que se modela matemáticamente el cohete y el sistema de control de empuje y una segunda parte en que se estudia el comportamiento de distintas estructuras de control (PID clásico, control predictivo MPC (Model of Predictive Control) y control auto-sintonizado STR (Self-Tuning Regulator)). Se finaliza con la elección de un sistema de control y sus parámetros. Finalmente, la última sección se dedica al estudio del control discreto y en que se tratan temas relacionados con el tiempo de muestreo, la sensibilidad y la precisión de los sensores, entre otros.

En este estudio se hace especial hincapié en comparar la respuesta de distintos modelos de control de una planta inestable.

Palabras clave: Control de actitud, Control auto-sintonizado, Control predictivo, Control PID, Doble integrador, Planta inestable, Control discreto

1 MODELO MATEMÁTICO DE LA PLANTA

En este estudio es necesario saber a priori que forma tiene la planta a controlar, pues ésta es una planta dinámica con parámetros que pueden cambiar en el tiempo. En este capítulo hay dos apartados: el primero trata de la modelización de la dinámica de un cohete mono-propulsor y el segundo apartado describe la dinámica del sistema de propulsión, TVC

1.1 DINÁMICA DEL COHETE

El cohete se modela como un sólido rígido libre que está sujeto a fuerzas y momentos provenientes del empuje motor (T), el peso (W) y la resistencia aerodinámica (D). En este caso no se han tenido en cuenta las fuerzas de sustentación aerodinámica porque resultan mucho menores que las de resistencia. Las fuerzas de empuje motor se sitúan en un punto de la tobera, para simplificar el estudio se consideran situadas en el centro de la garganta de la tobera. Las fuerzas aerodinámicas, que son fuerzas distribuidas, se pueden simplificar en una fuerza y un momento situado en el centro de presiones del cohete. Finalmente, el peso se sitúa en el centro de gravedad. La figura 1 muestra un esquema de las fuerzas sobre el cohete en una posición arbitraria, donde aparecen las siglas T para empuje, L y D para sustentación y resistencia aerodinámica, W para el peso, v para velocidad, δ para la deflexión del flujo, α para la inclinación del cohete respecto el aire y γ para la inclinación del cohete respecto la vertical.

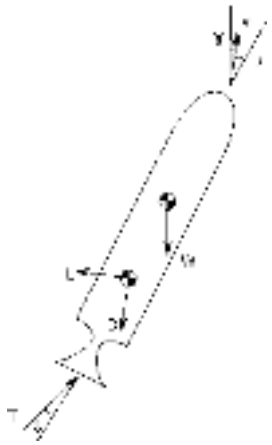


Figura 1: Fuerzas sobre un cohete

Las ecuaciones se estudian en un sistema de referencia situado sobre el cohete donde el centro de coordenadas está situado en el centro de gravedad, el eje 'x' apunta en el eje axial del cohete hacia la cofia, el eje 'y' es perpendicular a 'x' y apunta hacia la posición concreta de un elemento actuador del sistema TVC y el eje 'z' completa el triedro a derechas. Tras proyectar las fuerzas y momentos sobre los tres ejes fijos en la estructura del cohete (ejes *body*) y realizar las correspondientes transformaciones entre los ejes *body* y los ejes *earth* se concluye que las ecuaciones de la dinámica del cohete vistas desde los ejes *body* son:

$$\Sigma\bar{F} = m\begin{pmatrix}a_x\\a_y\\a_z\end{pmatrix} = T\cos\delta\begin{pmatrix}1\\\tan\delta_z\\\tan\delta_y\end{pmatrix} - W\begin{pmatrix}\cos\varphi\cos\psi\\\sin\theta\sin\psi\cos\varphi-\sin\varphi\cos\theta\\\cos\theta\sin\psi\cos\varphi+\sin\varphi\sin\theta\end{pmatrix} - \frac{1}{2}\rho v^2 SC_D\begin{pmatrix}\cos\alpha\\\sin\alpha\cos\beta\\\sin\alpha\sin\beta\end{pmatrix} \quad (1)$$

$$\Sigma\bar{M} = \begin{pmatrix}I_x\ddot{\theta}\\I_y\ddot{\psi}\\I_z\ddot{\varphi}\end{pmatrix} = T\cos\delta\begin{pmatrix}0\\-\tan\delta_y\\-\tan\delta_z\end{pmatrix}l_{0,CG} - \frac{1}{2}\rho v^2 SC_D\begin{pmatrix}0\\\sin\alpha\sin\beta\\-\sin\alpha\cos\beta\end{pmatrix}l_{CP,CG} \quad (2)$$

La ecuación (1) corresponde al modelado de fuerzas mientras que la ecuación (2) corresponde al modelado de momentos. El control de actitud se realizará sobre el segundo conjunto de ecuaciones y determinará que ángulo es necesario desviar el chorro de gases. Efectivamente se observa como el modelo dinámico es realmente complejo: 6 ecuaciones con movimientos acoplados y además algunos parámetros que cambian a lo largo del tiempo.

1.2 LINEALIZACIÓN DE LAS ECUACIONES DE MOVIMIENTO

Existen distintos niveles de linealización de las ecuaciones del movimiento del cohete. El primer nivel de linealización de las ecuaciones se consigue suponiendo los ángulos pequeños. Esta suposición se puede realizar en el caso que nos ocupa porque un ángulo se considera pequeño si está por debajo de los 20º; en este caso el cohete será inestable y no se podrá estabilizar nunca si se alcanzan ángulos girados entorno los 15º. Por este motivo resulta posible linealizar las ecuaciones. Los resultados son:

Tabla 1: Ecuaciones tras la linealización

$\Sigma\bar{F} = m\begin{pmatrix}\ddot{x}\\\ddot{y}\\\ddot{z}\end{pmatrix} = T\begin{pmatrix}1\\\delta_z\\-\delta_y\end{pmatrix} - W\begin{pmatrix}1\\-\varphi\\\psi\end{pmatrix} - \frac{1}{2}\rho SC_D\dot{x}\begin{pmatrix}\dot{x}\\\dot{y}\\\dot{z}\end{pmatrix}$ (8)
$\Sigma\bar{M} = \begin{pmatrix}I_x\ddot{\theta}\\I_y\ddot{\psi}\\I_z\ddot{\varphi}\end{pmatrix} = T\begin{pmatrix}0\\-\delta_y\\-\delta_z\end{pmatrix}l_{0,CG} - \frac{1}{2}\rho SC_D\dot{x}\begin{pmatrix}0\\\dot{z}\\-\dot{y}\end{pmatrix}l_{CP,CG}$ (9)

Obsérvese que las ecuaciones resultantes no aparecen acopladas. Estas ecuaciones se pueden usar para buscar un controlador lineal pero estas ecuaciones aún resultan difíciles de controlar porqué contienen términos no controlables. Es decir, la ecuación de momentos está formada por un término que depende del empuje y un segundo término que depende de la resistencia aerodinámica. Éste segundo término es de difícil cálculo para el regulador, pues es un término que aparece para armonizar las ecuaciones de fuerza con las de momento, pero es un término que no se puede controlar por sí solo. Hay otra razón para eliminar este término, la resistencia aerodinámica representada en este segundo término se puede tratar como una perturbación que ayuda a estabilizar el cohete. La función de la resistencia aerodinámica es ayudar pasivamente en el control de actitud. Si eliminamos este término de las ecuaciones estaremos obligando al controlador a ser lo suficientemente rápido y robusto como para controlar la actitud sin ayuda del *drag*. Es por esto que la segunda aproximación de la ecuación de momentos es precisamente eliminar el término aerodinámico.

$$\Sigma\bar{M} = \begin{pmatrix}I_x(t)\ddot{\theta}\\I_y(t)\ddot{\psi}\\I_z(t)\ddot{\varphi}\end{pmatrix} = T(t)\begin{pmatrix}0\\-\delta_y\\-\delta_z\end{pmatrix}l_{0,CG}(t) \quad (10)$$

Existe una tercera aproximación, la de usar parámetros constantes a lo largo del tiempo para obtener un controlador estático. Aunque esto no parezca una buena idea porque elimina mucha de la

física del cohete, esta aproximación es de gran ayuda al empezar a diseñar u controlador. Luego se pueden cambiar términos en el controlador para que este funcione correctamente con parámetros variables. Finalmente se presentan las ecuaciones del cohete en el dominio de Laplace:

$$\dot{\Theta} = \frac{1}{s}\delta_x \tag{11}$$

$$\begin{pmatrix}\Psi\\ \Phi\end{pmatrix} = -\frac{Tl_{0,CG}}{I_t}\frac{1}{s^2}\begin{pmatrix}\delta_y\\ \delta_z\end{pmatrix} \tag{12}$$

La ecuación (12) presenta la función de transferencia de la inclinación lateral del cohete, representada por un doble integrador. Además, las ecuaciones no están acopladas con lo que se podrán controlar independientemente usando dos controladores SISO iguales. La ecuación (11) muestra la función de transferencia de la velocidad de giro axial del cohete. Es más importante fijar una velocidad de giro axial nula que no determinar una posición concreta que no nos ofrece nada. Es importante limitar la velocidad de giro porque si ésta aumenta mucho, los actuadores que controlan la inclinación del cohete no serán capaces de seguir este giro y no realizarán las deflexiones calculadas en los momentos idóneos y resultará en un empeoramiento de la inclinación.

1.3 MODELO DEL TVC (TOBERA)

Finalmente se precisa obtener un modelo adecuado de *Thrust Vector Control.* Esto es importante porque existirá una relación entre la deflexión real del chorro de gases con el valor que pueda adquirir la variable que se puede manipular. Sin el ánimo de alargarse demasiado en este apartado, se presentan algunos de ejemplos de mecanismos TVC.

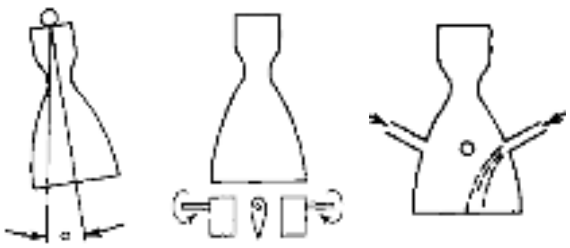


Figura 2: Mecanismos TVC: de iz. a dcha. tobera basculante, deflexión por aletas, deflexión por inyección

Por ejemplo, en una tobera basculante el ángulo desviado por la tobera no tiene que ser el mismo que se desvía el chorro de gases; con el uso de aletas la desviación del chorro tiene una relación con la posición y el giro de las aletas; y finalmente en la inyección, la desviación del chorro dependerá del flujo inyectado y la posición de este. Se requiere un modelo que relacione estas variables.

Tras un estudio profundo sobre los distintos mecanismos de control TVC y con el soporte de otros estudios [3, 4] se concluye que el mejor mecanismo para realizar control TVC con un cohete de pequeñas dimensiones es el método de inyección GITVC (*Gas Injection Thrust Vector Control*). A continuación se presenta el gráfico que relaciona la deflación del chorro con el flujo másico inyectado.

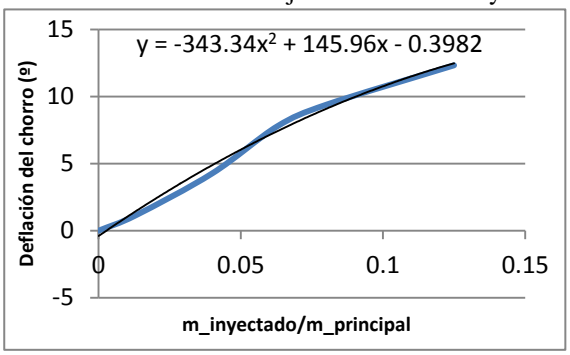


Figura 3: Relación entre la delación del chorro y el flujo másico inyectado

2 ESTRUCTURAS DE CONTROL

Una vez se tiene el modelo dinámico del cohete se procede a diseñar distintos esquemas de control y a compararlos entre ellos para seleccionar el tipo que mejor se adapte a la solución requerida. Para ello se estudian los controles PID, MPC, *Gain Scheduling* y STR. Al final todo regulador deberá ser probado ante el modelo sin linealizar del cohete y de esta forma comprobar que el regulador funciona correctamente. Las especificaciones para el regulador son que la inclinación del cohete sea nula y la velocidad de giro axial también sea nula.

2.1 CONTROL PID

El control PID es el control más clásico y común en la industria. La forma como se ha diseñado el controlador sigue el siguiente esquema:

1. Controlador para una función de transferencia totalmente lineal
2. Modificar el controlador al añadir el término de la resistencia aerodinámica
3. Modificar el controlador al añadir los parámetros variables en el tiempo

4 Modificar el controlador al añadir dinámicas lineales simplificando así el modelo.

Siguiendo este esquema, inicialmente se diseña un controlador muy sencillo que poco a poco se va complicando a causa de las no linealidades.

Para el estudio de la controlabilidad del sistema se emplean principalmente dos tipos de perturbaciones: una consistente en una inclinación inicial de 0.1 radianes y otra consistente en una perturbación en la aceleración angular del cohete de forma cuadrada de $0.2rad/s^2$ y frecuencia de 0.25Hz. Tras numerosas simulaciones del modelo con el control PID se concluye que:

- En el instante inicial el controlador no debe contener ningún parámetro integral. Esto se debe a que la planta es un doble integrador, una función de transferencia muy inestable. Si se usa un integrador, el sistema en lazo cerrado seria de tercer orden y aún sería más difícil de controlar
- El parámetro derivativo debe mantenerse bajo para evitar un excesivo ruido; pero este término ayuda a mejorar la velocidad del controlador.
- El término proporcional actúa como un relajador del control y se mantiene constante siempre a lo largo del tiempo.

Los valores de los términos proporcional, derivativo e integral se han calculado primeramente con sintonía analítica, la cual proporciona unos límites de estos tres parámetros. Muchos han sido los ensayos de controladores pero solo un controlador ha sido elegido. A diferencia de los controladores industriales, los parámetros de este controlador cambian en el tiempo para adaptarse al valor de la planta. Esto se debe a que la planta modifica sus parámetros siguiendo unas leyes que pueden expresarse en función del tiempo. De la misma forma, los parámetros del regulador también se pueden expresar en función del tiempo. Se ha diseñado un controlador adaptando los parámetros de control a cada instante de tiempo. Como disponemos del comportamiento del cohete en cada instante, resulta que se pueden definir los parámetros del regulador PID en función del tiempo. Estos parámetros se han aproximado a rectas para facilitar el cálculo de los parámetros en cada instante de tiempo. A continuación se presenta el regulador PID con parámetros variables a lo largo del tiempo:

$$k_p = -1$$

$$k_i = \begin{cases} -0.02t & si\ t < 5s \\ -0.24t + 1.1 & si\ 5 < t < 15s \\ -2.5 & si\ 15s < t \end{cases}$$

$$k_d = 0.005t - 0.24$$

En la figura 4 se muestra en verde la respuesta del controlador de parámetros variables con el tiempo si el cohete es lanzado con una inclinación inicial de 0.1 radianes. En azul se muestra un primer controlador realizado para el caso de un cohete donde los parámetros de masa, empuje e inercia fueren constantes, donde los parámetros del controlador son:

$$k_p = -1, k_i = -1.71, k_d = -1$$

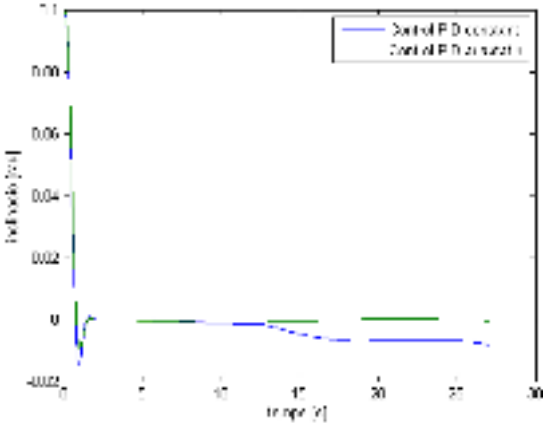


Figura 4: Comparación entre un PID constante (azul) y uno variable en el tiempo (verde)

Se observa como en el caso de usar un controlador con parámetros variables (representado en verde) la actitud del cohete se mantiene alrededor de cero. Por otro lado, el controlador de parámetros constantes (en azul) controla correctamente los instantes iniciales pero a partir de los 13 segundos la inclinación aumenta. La explicación de este fenómeno es que éste controlador solo está adaptado en un punto que ocurrirá en un instante. Alrededor de los 13 segundos, donde la inclinación del cohete es muy pequeña pero no es nula, el cohete pasa de un régimen subsónico a un régimen supersónico. Esto se traduce en que las fuerzas de resistencia aerodinámica, que ayudaban a estabilizar el cohete, disminuyen y recae mayor trabajo sobre el controlador. En cambio, si el controlador se va adaptando punto a punto al modelo a controlar, las

inclinaciones resultantes de eventuales perturbaciones se ven muy reducidas.

El tercer movimiento a controlar es el del giro axial del cohete. No se pretende controlar la posición axial del cohete, pues no modifica para nada la actuación del cohete. La variable a controlar es la velocidad de giro, pues si esta velocidad es demasiado alta los actuadores de corrección de inclinación no responden correctamente. La perturbación de estudio de este caso es la reducción de la velocidad de giro si el cohete es lanzado con una velocidad inicial de giro de 0.1 rad/s, una velocidad que no es muy elevada pero que nos sirve para evaluar el comportamiento del regulador. La función de transferencia de este segundo proceso es un integrador. Se proponen dos controladores para este proceso: un controlador PI y otro P.

Por un lado el controlador P reduce la velocidad de giro siguiendo una respuesta de primer orden. La figura 5 muestra la respuesta del giro del cohete usando reguladores de parámetros proporcionales distintos. Se observa que a medida que el término proporcional aumenta la velocidad de giro disminuye más rápidamente. No obstante, existe un límite en el control porque llega un momento donde los actuadores limitan la velocidad del control. Este punto límite se encuentra entorno a $k_p = 1$.

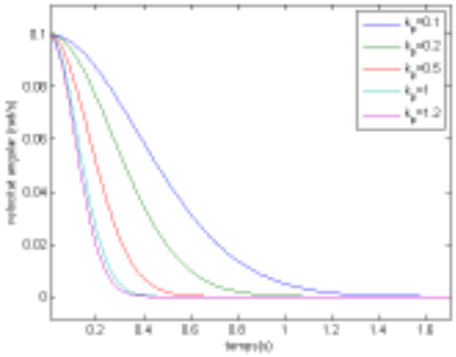


Figura 5: Control de giro por P

Por otro lado el controlador PI incluye un segundo integrador en lazo cerrado. Este regulador, aparte de controlar que la velocidad de giro se anule, también provoca que la posición azimutal se mantenga como la inicial. La figura 6 muestra la velocidad en que el cohete resulta controlado. Se observa una respuesta de segundo orden. A medida que el termino integral aumenta el sobre-pico en la velocidad de giro también aumenta, y a su vez el tiempo necesario para dejar el giro a velocidad nula aumenta. Por este motivo se tiende a dejar el control integral de lado para esta función de transferencia.

El control derivativo desestabiliza la planta y no se tiene en cuenta para este regulador concreto.

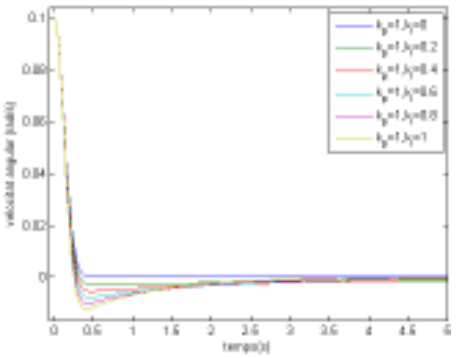


Figura 6: Control de giro por PI

2.2 CONTROL PREDICTIVO MPC

Otra estrategia de control empleada es el control predictivo descrito en [2]. Este consiste en calcular un error futuro y evitarlo con un adelantamiento de las acciones de control. Una de las primeras cuestiones que asaltan cuando se habla de control predictivo es como deben ser los horizontes de control y de predicción. En procesos industriales donde las plantas son estables es común tener un horizonte de predicción de unos 20 intervalos y un control entre 5 y 10 intervalos. En cambio, cuando se habla de un proceso inestable como el que se trata en este artículo, éstos valores cambian y se deben recalcular. Para obtener los valores de estos dos parámetros se ha procedido de la siguiente forma:

1. Se mantienen todos los parámetros constantes y solo se varía uno.
2. Cuando la respuesta del sistema sea diferente (peor control) se para el estudio.
3. Se elige el valor extremo del parámetro que hace que el control sea el óptimo.

Siguiendo este procedimiento y para una planta formada por un doble integrador, se encuentra que tanto el horizonte de predicción como el horizonte de control tienen el valor de 2. Es decir, el control predictivo calculará una previsión del error hasta dos intervalos de tiempo por delante del actual y diseñará una secuencia de control para reducir este error en dos intervalos de tiempo.

Otros parámetros importantes son las funciones de peso que calibran qué parte de la señal de control proviene de la previsión del error y qué parte proviene de las señales de control anteriores. Se ha observado que la función de peso del error desestabiliza mucho el cohete mientras que la respuesta ante las anteriores variables de control es casi uniforme. Por consiguiente se eligen las siguientes funciones de peso:

$$\delta = 0, \lambda = 1$$

Dónde δ es la función de peso del error y λ es la función de peso de las anteriores variables de control.

Por otro lado, en el control MPC también es posible usar modelos de ruido y modelos de perturbación. Estos modelos permiten al controlador conocer la respuesta de la planta ante las perturbaciones. No obstante, el modelo de ruido no se debe emplear sobre el proceso real, pues lo desestabilizaría. Es bueno saber cómo responde el modelo ante una perturbación para así mejorar la respuesta del controlador. No se debe tampoco abusar del modelo de ruido, pues se puede caer en el error de diseñar un control muy efectivo ante un tipo determinado de perturbaciones pero que no funciona bien ante cualquier otra perturbación.

A continuación, la figura 7 muestra las respuestas de la inclinación del cohete cuando este se ha lanzado con una inclinación inicial de 0.1 radianes. En el caso de usar un control MPC (línea roja) la respuesta tiende más rápido hacia cero pero se queda un cierto tiempo oscilando alrededor del punto objetivo. Por el contrario, al usar un control PID (línea azul) la pendiente de aproximación a cero es menor que en el caso del MPC pero no realiza oscilaciones, tan solo existe un sobre-pico. Tanto para el caso del MPC como para el PID los tiempos de estabilización de la inclinación del cohete son similares.

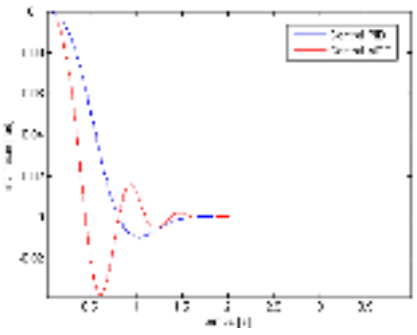


Figura 7: Comparación de control PID vs. MPC

En lo referente al control de la velocidad de giro entorno al eje axial del cohete, éste sigue una aproximación típica de una respuesta de primer orden, como en el caso del PID.

2.3 CONTROL CON PREVISIÓN DE GANANCIA (*GAIN SCHEDULING*)

El control con previsión de ganancia es un control que modifica su respuesta en función de la ganancia predicha de la planta. En el caso que nos ocupa la ganancia del cohete puede ser predicha gracias a que el empuje, la masa, la inercia y la distancia entre el centro de gravedad y un punto fijo en el cohete pueden ser representadas por funciones que dependen del tiempo (ecuaciones 3 a 7). El problema encontrado al estudiar este controlador es que la función en lazo cerrado es una función de transferencia de tercer orden. Es necesario reducir el orden de esta función para poden diseñar un controlador cuya respuesta se asemeje a una respuesta de primer orden. La función de transferencia en lazo cerrado es:

$$G(s) = \frac{k}{(1+\tau s)(1+u\tau s)^2} \qquad (13)$$

Eligiendo correctamente un valor de 'u' se puede hacer que la función de tercer orden se comporte como una de primer orden si existe un polo dominante y los otros dos son no dominantes. También se selecciona un valor para el tiempo de retraso, concretamente:

$$\text{u=0.02 y } \tau = 10$$

Usando la ley de sintonía experimental de Ziegler-Nichols, los parámetros del controlador PID deben ser:

$$k_p = 26, k_i = 2.5\,, k_d = 10.1$$

La figura 8 muestra la respuesta del controlador PID con parámetros no constantes (rojo) y la del controlador PID con previsión de ganancia (azul). Este segundo controlador evita el sobre-pico que realiza el primer controlador, pero en cambio, en la senda de aproximación a cero padece oscilaciones de baja amplitud y baja frecuencia. Aunque estos no sean motivos de descarte del controlador, es necesario comentarlo.

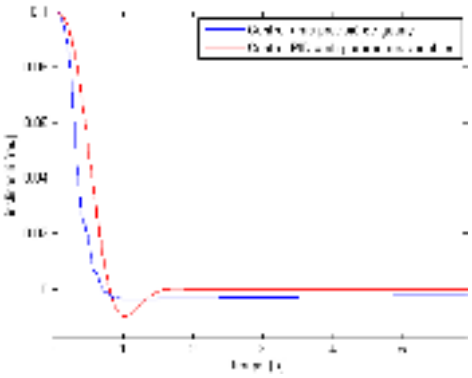


Figura 8: Comparación de controladores PID vs. *Gain Scheduling*

En la figura 9 se muestra la respuesta de ambos controladores cuando el cohete padece una perturbación en su aceleración angular en su inclinación periódica cuadrada de $0.2 rad/s^2$ y frecuencia 0.25Hz (señal verde).

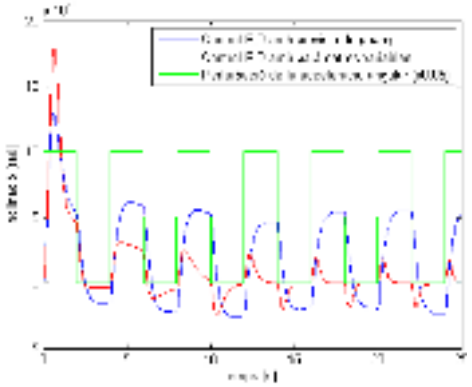


Figura 9: Comparación PID vs. *Gain Scheduling* a perturbaciones periodicas

Se puede observar que tanto la respuesta con previsión de ganancia (azul) y la del PID con parámetros variables (rojo) son estables; pero el PID padece menor perturbaciones en su amplitud y sigue un camino más rápido hacia cero.

2.4 SELF-TUNING REGULATOR (STR)

En cuanto al control auto-sintonizado cabe comentar que realiza cambios en sus parámetros de forma automática según la respuesta de la planta. La respuesta del controlador STR es de la forma:

$$R(s)u(t) = T(s)u_c(t) - S(s)y(t) \qquad (14)$$

Dónde 'u' es la respuesta del controlador (acción de control), 'y' es la respuesta de la planta, 'u_c' es la consigna y R, T y S son polinomios. En este caso la consigna es siempre cero porque se requiere una actitud completamente vertical. Así el polinomio T no se usa. Los otros dos polinomios son polinomios de primer grado (según el método de mínimos polos). De esta forma la ley de control resulta:

$$s^2 \cdot (s + r_0) + g \cdot (s_1 s + s_0) = (s + a_0) \cdot (s^2 + a_{1m} s + a_{2m}) \qquad (15)$$

Donde $(s^2 + a_{1m}s + a_{2m})$ representa la posición de los hipotéticos polos del controlador. Tras una serie de ensayos se determinan los siguientes valores:

$$a_{1m} = 20, a_{0m} = 100, a_0 = 100, r_0 = 120,$$

$$s_1 = \frac{220}{g},\ s_0 = 10000/g.$$

Usando $a_0 = 100$ se elimina la dinámica del polinomio observador. Se observa que los dos parámetros que acompañan al polinomio S están divididos por 'g', que es la ganancia de la función de transferencia. Esto quiere decir que aunque el sistema sea auto-sintonizado, parte de los parámetros tienen una condición exterior impuesta por el valor de la ganancia de la planta, $g = T \cdot L_{cp,cg}/I_t$, que depende del tiempo.

La figura 10 muestra la respuesta de la inclinación del cohete cuando este es lanzado con una inclinación inicial de 0.1 radianes. La respuesta del controlador SRT (en rojo) es muy similar a la del controlador de previsión de ganancia de la figura 8: la respuesta parece ser de primer grado porque no realiza sobre-picos, alcanza el estacionario en el mismo tiempo que el PID de parámetros constantes pero en su aproximación a cero padece una serie de oscilaciones.

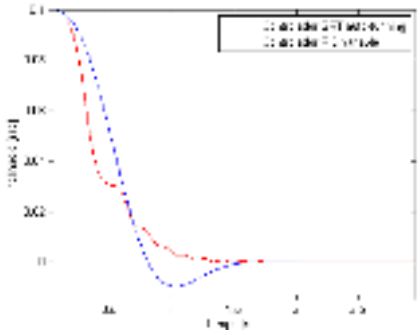


Figura 10: Comparación PID vs. STR

El hecho diferencial entre estos dos reguladores reside en la respuesta cuando el cohete está sometido

a perturbaciones periódicas a lo largo del tiempo. Mientras que en la figura 10 solo se estudia la estabilidad del controlador en el momento del lanzamiento, el cual es un caso muy crítico, en la figura 11 muestra la respuesta del controlador a lo largo de todo el tiempo. La perturbación inducida es periódica cuadrada de amplitud $0.2 rad/s^2$ y frecuencia 25Hz. Tal como se observa en la figura 11, la respuesta del cohete ante estas perturbaciones es más estable que en el caso del PID en los siguientes aspectos: cuando el cohete sufre una perturbación, en el momento inicial de ésta y para el regulador PID el cohete sufre una gran desviación que poco a poco disminuye hasta valores alrededor de cero. En cambio, usando el regulador STR la inclinación sufrida es mucho menor (esto se nota sobre todo en el instante del lanzamiento, donde la función de transferencia es más crítica y la amplitud de la respuesta del cohete varia en un orden de magnitud según se use un PID o un STR); aunque usando el controlador STR el cohete tiende a cero muy lentamente.

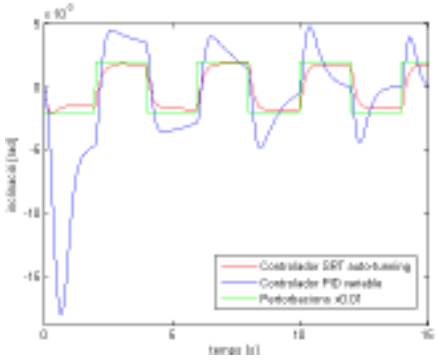


Figura 11: Comparación PID vs. STR con perturbaciones periódicas

En la figura 12 puede observarse el comportamiento del cohete ante una perturbación constante de $0.2 rad/s^2$ en su aceleración angular-

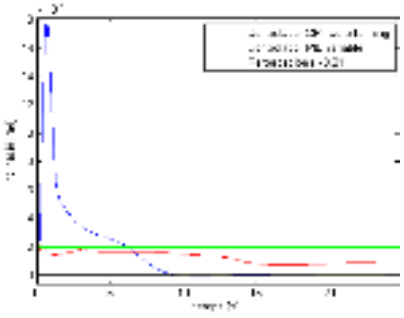


Figura 12: Comparación PID vs. STR con perturbación constante

En este caso la perturbación se mantiene durante mayor tiempo y puede observarse como el regulador STR tiende a estabilizar el cohete entorno a la vertical, pero lo hace muy lentamente. También se nota como en el instante inicial la amplitud de la perturbación es mucho mayor en el caso del PID que en el caso del STR porque el controlador STR es más agresivo.

El otro movimiento a controlar es la velocidad de giro. En este caso la función de transferencia es un integrador y la ley de control usando el método de mínima orden, es:

$$s \cdot r_0 + k \cdot s_0 = a_0 \cdot (s + a_x)$$

Es decir, los polinomios S y R son de primer grado. T es inexistente porque la consigna de velocidad de giro siempre es nula. Tras estudiar distintos valores para los parámetros del regulador, éste queda como:

$$u(t) = -\frac{0.1}{I_x} y(t-1)$$

La figura 13 muestra la velocidad de giro del cohete si se aplica una aceleración periódica cuadrada de $0.2 rad/s^2$ en su aceleración angular axial. Ambos reguladores permiten comportamientos similares pero el controlador STR, al ser más agresivo, reduce mejor la velocidad del giro.

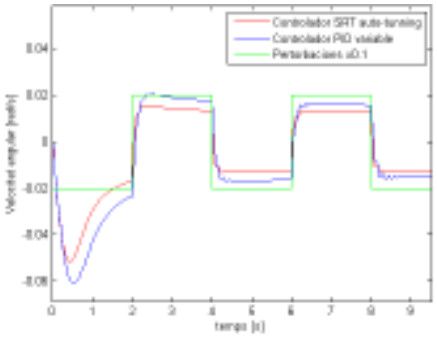


Figura 13: Control de giro, comparación PID vs. STR

3 CONCLUSIONES

Se han presentado hasta cuatro esquemas distintos de control y se ha estudiado su comportamiento ante las mismas perturbaciones. Ahora es momento de seleccionar uno. Para ello se usará el sistema de selección basado en la ponderación del valor técnico de cada solución. Para referenciar los resultados se

toma como valores nominales los ofrecidos por el regulador PID; el resto de valores serán mejores o peores en referencia a éste.

La tabla 2 muestra los valores técnicos calculados para los distintos controladores presentados. El regulador predictivo MPC obtiene una baja puntuación porque este regulador funciona bien para plantas estables y perturbaciones conocidas, pero no es aconsejable para plantas inestables. Los otros tres reguladores obtienen puntuaciones muy similares. Los reguladores PID de parámetros variables y el STR obtienen la misma puntuación. El regulador PID proporciona una corrección del error rápida con pocas oscilaciones mientras que el regulador STR impide picos de error iniciales pero es lento cuando se trata de minimizar el error estático.. Aunque el controlador sea muy estable y no produzca oscilaciones, si en algún momento la inclinación pasa de cierto valor nada podrá hacer el regulador para estabilizar el cohete. El motivo principal para elegir el regulador STR es que, comparando las inclinaciones máximas que permiten estos tres reguladores, la regulador STR obtiene menor inclinación que los otros dos; por otro lado, el motivo para elegir el regulador PID es que es menos oscilatorio y permite el control del error en estático de una mejor forma que otros reguladores..

Tabla 2: Valor técnico de los distintos controladores

Parámetro	Peso	PID		MPC		GS		STR	
		V	VT	V	VT	V	VT	V	VT
Velocidad	3	10	30	10	30	10	30	10	30
Error en estático	-0.5	10	-5	15	-7.5	8	-4	20	-10
Oscilaciones	-0.5	10	-5	20	-10	17	-8.5	20	-10
Pico inicial	-1	10	-10	5	-5	8	-8	0	0
TOTAL	**1**		**10**		**7.5**		**9.5**		**10**

Referencias

[1] **Aström Karl Johan y Wittenmark Björn** Adaptive Control. - USA : Addison Wesley Longman, 1995.

[2] **Camacho E.F. y Bordons C.** Model Predictive Control. - London : Springer_Verlag, 2004.

[3] **Deere K. [y otros]** Computacional study of fluidic thrust vectoring using separation control in a nozzle. - Orlando : AIAA, 2003.

[4] **Deere K. [y otros]** Experimental Study of an Axisymmetric Dual Throat Fluidic Thrust Vectoring Nozzle for Supersonic Aircraft Application. - Hampton : AIAA, 2003.

[5] **Manent Bistué Pau** Control de actitud de un cohete de sondeo atmosférico. Master's Thesis. - Terrassa : [s.n.], 2013.

[6] **Martin C.L. y Powers L.B.** Feasibility study of LITVC for shuttle SBR. - Alabama : NASA, 1981.

[7] **Ogata K.** Ingeniería de control moderna [Libro]. - Madrid : Pearson, 2010.

Desarrollo de una herramienta para el análisis de datos multi-criterio. Aplicación en el ajuste de controladores del tipo PID.

Alberto Pajares Ferrando, Francesc Xavier Blasco Ferragud, Gilberto Reynoso-Meza, Juan Manuel Herrero Dura
alpafer1@upv.es, xblasco@isa.upv.es, gilreyme@upv.es, juaherdu@isa.upv.es

Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València

Resumen

En el presente trabajo, se expone el funcionamiento de una aplicación para el tratamiento de datos en el análisis multicriterio*, en el marco de una estrategia de optimización multiobjetivo. Muchos sistemas requieren optimizar varios objetivos al mismo tiempo, y es por ello, por lo que la estrategia multiobjetivo puede resultar de interés. El algoritmo multiobjetivo calcula un conjunto de soluciones óptimas y es necesario definir un compromiso entre ellas para seleccionar la más idónea en cada caso. Por esta razón, es importante tener una herramienta que ayude al diseñador a realizar esta tarea. La herramienta presentada en este artículo ayudará a alcanzar la solución deseada, como se demuestra en el caso estudio.*

Palabras clave: *Optimización multiobjetivo*, *análisis multicriterio*, *tratamiento de datos*.

1. INTRODUCCIÓN

Es usual plantear un problema de diseño como la optimización de diferentes objetivos. El ingeniero de control (por ejemplo) debe formular el problema de manera que cumpla con una serie de requerimientos y especificaciones. Esto es lo que se conoce como *problema de optimización multiobjetivo* (MOP [1] por sus siglas en inglés).

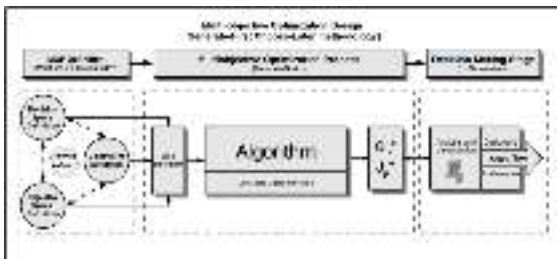


Figura 1. Proceso de *optimización multiobjetivo* para el diseño en ingeniería

Este tipo de optimización difiere de enfoques tradicionales en que calcula un conjunto de soluciones, todas ellas óptimas pero con diferente balance entre las prestaciones de diseño. Lo anterior debido a que usualmente no es posible encontrar una solución que optimice todos los objetivos de diseño. Acorde con la figura número 1, la resolución de un MOP debe considerar por lo menos tres etapas [2,3]: la definición del problema a resolver, el proceso de optimización y la etapa de *selección multicriterio* (MCDM, por sus siglas en inglés).

En la etapa 1, debe definirse el problema multiobjetivo a resolver (definir variables de decisión, y los objetivos del sistema). En la etapa 2, el problema multiobjetivo debe traducirse en un problema de optimización. Así mismo, debe definirse el algoritmo multiobjetivo a emplear. Este debe calcular un conjunto de soluciones óptimas con las características deseadas por el diseñador. Finalmente, en la etapa de *análisis multicriterio*, se suele emplear una herramienta y/o procedimiento que permita involucrar activamente al diseñador. El mismo debe permitir analizar el intercambio entre prestaciones y rendimiento de las diferentes soluciones, con el fin de elegir la solución más idónea a sus especificaciones. Este procedimiento no es trivial y suele llevar más tiempo que la propia fase de búsqueda [2]. Por ello, nuevas herramientas para dicho proceso son valoradas por el diseñador.

La finalidad de este documento, es presentar una herramienta destinada a facilitar al diseñador esta tarea. Él podrá filtrar/descartar soluciones y comparar prestaciones de cada una de ellas, para seleccionar la que más se ajuste a los requerimientos del sistema. Por ello, se ha diseñado una herramienta intuitiva y de fácil disposición. El artículo se organiza de la siguiente forma: en la sección 2 se dará una descripción general del contexto de los MOP; en la sección 3, se explicará la aplicación presentada; en la sección 4, se validará la herramienta; Finalmente, en la sección 5, se presentan las conclusiones.

2. PRELIMINARES

Prácticamente en cualquier área y en una variedad de contextos, se presentan problemas con múltiples

requerimientos. En los últimos años, la optimización de problemas que requieren optimizar más de un indicador ha recibido una gran atención; dicho interés motivado, en gran medida, por la naturaleza multiobjetivo de los problemas del mundo real [4]. De forma general, los optimizadores de los MOP no se restringen a encontrar una única solución dado una función de coste, sino en proporcionar un conjunto de soluciones, conocido como Conjunto de Pareto. Cada solución en este conjunto se dice que es una solución Pareto Óptima, y cuando se representan en el espacio de objetivos, se conocen colectivamente como Frente de Pareto. Una vez obtenido el frente de Pareto, el principal objetivo del diseñador es encontrar la mejor solución, dentro del conjunto aproximado, para el problema teniendo en cuenta sus especificaciones. Por tanto, una fase final de toma de decisiones y MCDM es llevada a cabo.

Debido a que la solución a un MOP no es un solo valor, sino un conjunto de alternativas (Conjunto Pareto), determinar la solución que mejor satisface la preferencia de quien toma las decisiones, es a menudo una tarea difícil [2]. Esta dificultad se debe al número de alternativas y objetivos, además del posible desconocimiento de las herramientas a utilizar y sus limitaciones. Es por ello por lo que se presenta en este trabajo una aplicación destinada a facilitar esta tarea.

Es ampliamente aceptado que las herramientas de visualización son valiosas y proporcionan a los diseñadores un método significativo para analizar el Frente Pareto, y de este modo seleccionar las mejores soluciones en los diferentes contextos [5]. Para los problemas de dos dimensiones (y, a veces para tres dimensiones) suele ser sencillo realizar un análisis gráfico preciso, pero esto se hace más difícil para dimensiones más altas. En esta dirección se desarrolló una alternativa, denominada Level Diagrams (LD) [6]. Esta alternativa, permite un análisis más fácil del conjunto de Pareto y por ello es útil en la toma de decisiones.

LD se puede utilizar en una metodología a priori e interactiva para ayudar a la decisión de la mejor solución. Es una estrategia de visualización geométrica del frente y conjunto de Pareto; cuando se combina con una metodología de coloración para identificar las preferencias, pueden ser una herramienta poderosa para el proceso de MCDM. Éste es el caso de la LD-Tool (Matlab versión)[1].

Pese a estas ventajas, LD-Tool (Matlab versión) tiene algunas limitaciones respecto a su uso. En primer lugar, su ejecución depende del programa MATLAB, una herramienta de software matemático muy extendida en la ingeniería. Esto es una limitación, ya que no permite que su uso se extienda con facilidad. Por otro lado, el acceso a la programación no es sencillo, porque se requieren unos conocimientos previos de programación en el entorno de MATLAB.

Es por ello que se diseñó otra herramienta con este propósito, llamada LD-Tool[2]. Esta aplicación es de libre utilización y tiene mayores prestaciones para el graficado. Además ya no depende del entorno Matlab. Sin embargo, con la LD-Tool se pierden prestaciones de scripting de la versión original.

Por esta razón se buscan nuevos entornos para complementar las prestaciones de dicho visualizador pero fiel a la filosofía de emplear una plataforma de fácil uso e interpretabilidad. Es por ello que se decide realizarla en Microsoft Excel, porque es un programa muy extendido, que permite programar teniendo conocimientos básicos a nivel de usuario. Excel utiliza el lenguaje de programación Visual Basic. Aunque la motivación de la aplicación es la de emplearse en conjunto con el LD-Tool, se puede emplear potencialmente con cualquier otra estrategia de visualización. Por lo tanto uno de los propósitos de la aplicación, es acercar la herramienta al usuario final, y hacerla más accesible. A continuación, se incorpora una detallada explicación de la aplicación diseñada (también disponible junto al LD-Tool).

3. PROPUESTA

La finalidad de esta sección es entrar en detalle en la aplicación, mostrando la interfaz de la misma y comentando cada una de las funciones que realiza.

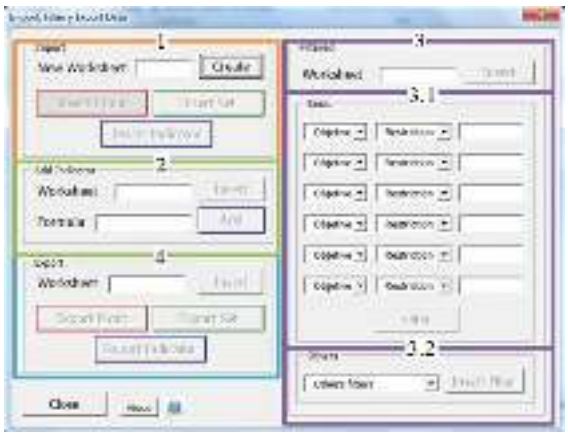


Figura 2. Interfaz Principal

La interfaz principal consta de 4 sectores principales como se observa en la figura número 2. La primera de ellas en la zona 1, llamada “Import”, es en la que el usuario deberá introducir el frente y el conjunto (y los indicadores[3], si lo desea). La zona 2 es la encargada de añadir indicadores, útiles para el diseñador. Estos pueden ser cualquier tipo de fórmula

[1] Disponible en http://www.mathworks.com/matlabcentral/fileexchange/24042
[2] Disponible en http://cpoh.upv.es/es/investigacion/software.html
[3] Indicador es una variable que proporciona una información útil para el diseñador, para la elección de la mejor solución

que admita la herramienta Excel en cualquier hoja de cálculo. La zona 3 es en la que el usuario puede realizar un filtrado. Este, se puede realizar mediante, un filtro básico, que se suele realizar para descartar soluciones en un primer contacto, o un filtro más avanzado, que puede ser por ejemplo un Global Physical Programming [10], una variación del Physical Programming [7]. La zona 4 es en la cual se exportarán los datos, en archivos de texto, listos para ser introducidos en cualquier visualizador.

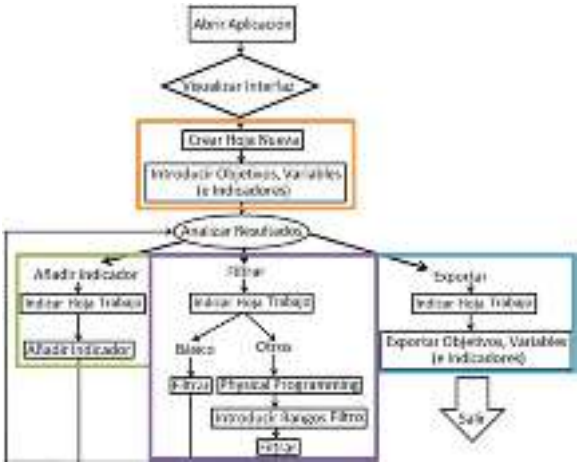


Figura 3. Diagrama de Flujo

Una vez se crea una nueva hoja, se introducen los datos del problema, frente y conjunto. Con este paso realizado se abren dos posibles caminos, en los que se podrá añadir indicadores o filtrar los datos. Para ambos casos se deberá primero introducir la hoja de trabajo, sobre la cual realizar las operaciones.

Para añadir un indicador, habrá que centrarse en la zona 2 de la interfaz. Usualmente se emplean vectores de peso para los objetivos e indicadores existentes, pero puede ser cualquier tipo de fórmula que admita Excel (logarítmicas o exponenciales por ejemplo).

Para realizar el filtrado de datos, en la zona 3. Se puede realizar un filtro básico o un filtro más avanzado. El filtro básico, en la zona 3.1, suele utilizarse para descartar soluciones, para ello se deberá indicar el objetivo o indicador a filtrar, su restricción y el valor al cual restringirlo. El filtro avanzado, puede ser por ejemplo un filtro del tipo Global Physical Programming [10], para ello se deberá seleccionar esta opción en el desplegable de la zona 3.2 de la interfaz. En él se deberá introducir los rangos del filtro (HD, D, T, I y HI) para cada uno de los objetivos e indicadores que tenga el sistema. Se introducirán en una hoja nueva que creará la aplicación, que tendrá una apariencia similar a la figura 4, en la que habrá que introducir un valor en las celdas vacías. Una vez introducidos se visualizará un nuevo indicador con el rendimiento de cada una de las soluciones, ordenadas de menor a mayor, es decir, de más a menos idónea.

En la figura 3 se observa el diagrama de flujo de la aplicación. Como se ve, existen 4 tramos bien diferenciados, cada una de ellos coindice con una zona de la interfaz (se dispone de tutorial en la web).

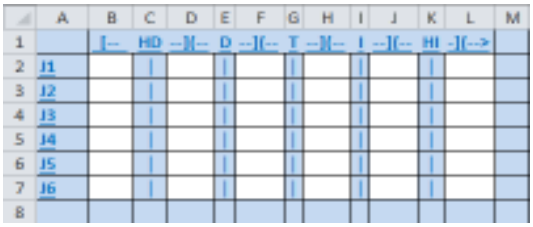


Figura 4. Hoja de datos (Physical Programming).

Por último antes de salir de la aplicación, se podrá exportar los datos en la zona 4. Para ello, se indicará la hoja a exportar, para posteriormente poder exportar los objetivos, variables e indicadores.

4. APLICACIÓN EN EL AJUSTE DE CONTROLADORES DEL TIPO PID

El objetivo de esta sección es validar la propuesta del apartado anterior. Para este propósito, se utilizará como ejemplo el problema mutli-objetivo del ajuste de controladores PID para la regulación del efector final de un manipulador **IRB6600** de **ABB** [8]. En este problema partimos del problema plateado en [9] así como del frente aproximado en dicho trabajo. Este problema tiene 6 objetivos: error máximo; tiempo de establecimiento máximo; Par máximo; Media de la raíz cuadrada del par; Ruido en la acción de control y valor máximo de la función de sensibilidad.

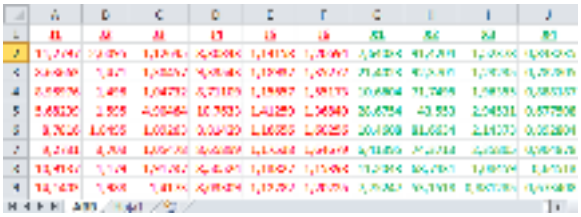


Figura 5. Hoja de cálculo.

En primer lugar se deberán, importar cada uno de los datos (frente, conjunto e indicadores si lo deseamos), en la hoja nueva creada anteriormente, en este caso con el nombre “ABB”. Una vez introducidos, la hoja de cálculo queda tal como se muestra en la figura número 5. Visualizando estos datos iniciales con la LD-Tool, obtenemos la figura 6.

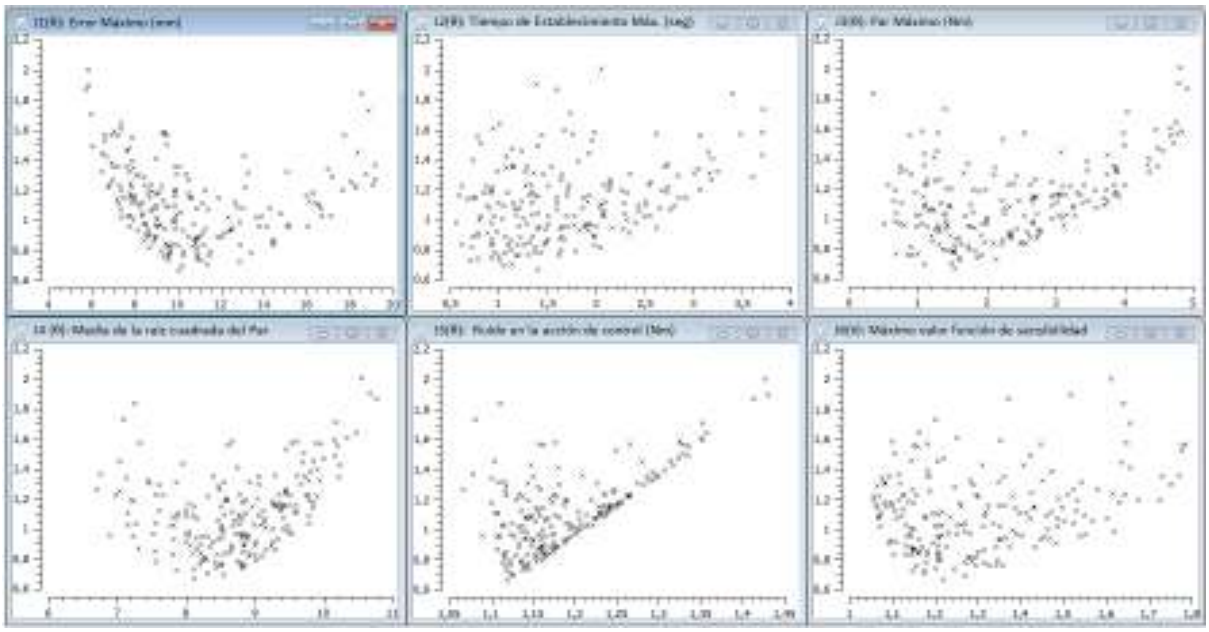


Figura 6. Visualización del frente de Pareto con seis objetivos usando LD-Tool.

Posteriormente se puede por ejemplo introducir un indicador del problema, tal como se explica en la sección 3. En este ejemplo que hemos utilizado, se conoce que un indicador útil para el diseñador [8], es un vector de pesos que multiplique a los objetivos por el vector "[2.8 2.8 3.5 1.4 1.4 0]". Esto se introduce en la zona de añadir indicadores de la interfaz como se ve en la figura 7, mediante la fórmula ("J1* 2.8 + J2*2.8 + J3*3.5 + J4*1.4 + J5*1.4"), obteniendo el indicador mostrado en la figura número 8.

En la zona derecha, se pueden introducir los filtros, tal como se detalla en la sección 3. Se podrá realizar un filtro básico o más avanzado. Los filtros se realizarán en una hoja llamada con el nombre original seguido de "_filter". Además, cada filtro que se realice se mostrará en una hoja diferente, para de este modo nunca perder la información anterior y poder retomarla en cualquier momento.

Para realizar un filtro básico, en este ejemplo interesa descartar los indicadores calculados anteriormente superiores a 50,5. Para ello se introduciría en la zona de filtrado de la interfaz tal como se observa en la figura número 7. Los resultados que se observarían se muestran en la figura número 9. Además como dato informativo para el diseñador, la aplicación nos muestra un mensaje en el que indica el número de datos que se han filtrado. Exportando los datos de la aplicación y visualizando en la LD-Tool obtenemos la figura número 10a. Como se observa las soluciones han disminuido considerablemente, ya que hemos eliminado gran parte de ellas al realizar el filtro anterior.

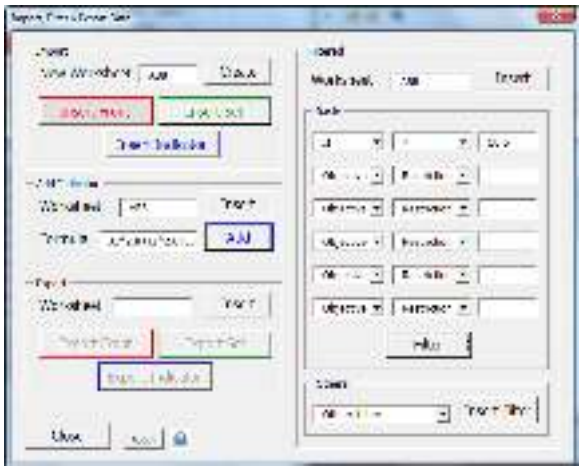


Figura 7. Interfaz Principal.

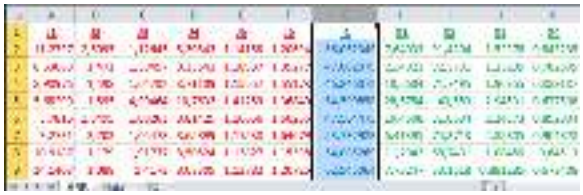


Figura. 8. Hoja de cálculo.



Figura 9. Hoja de cálculo.

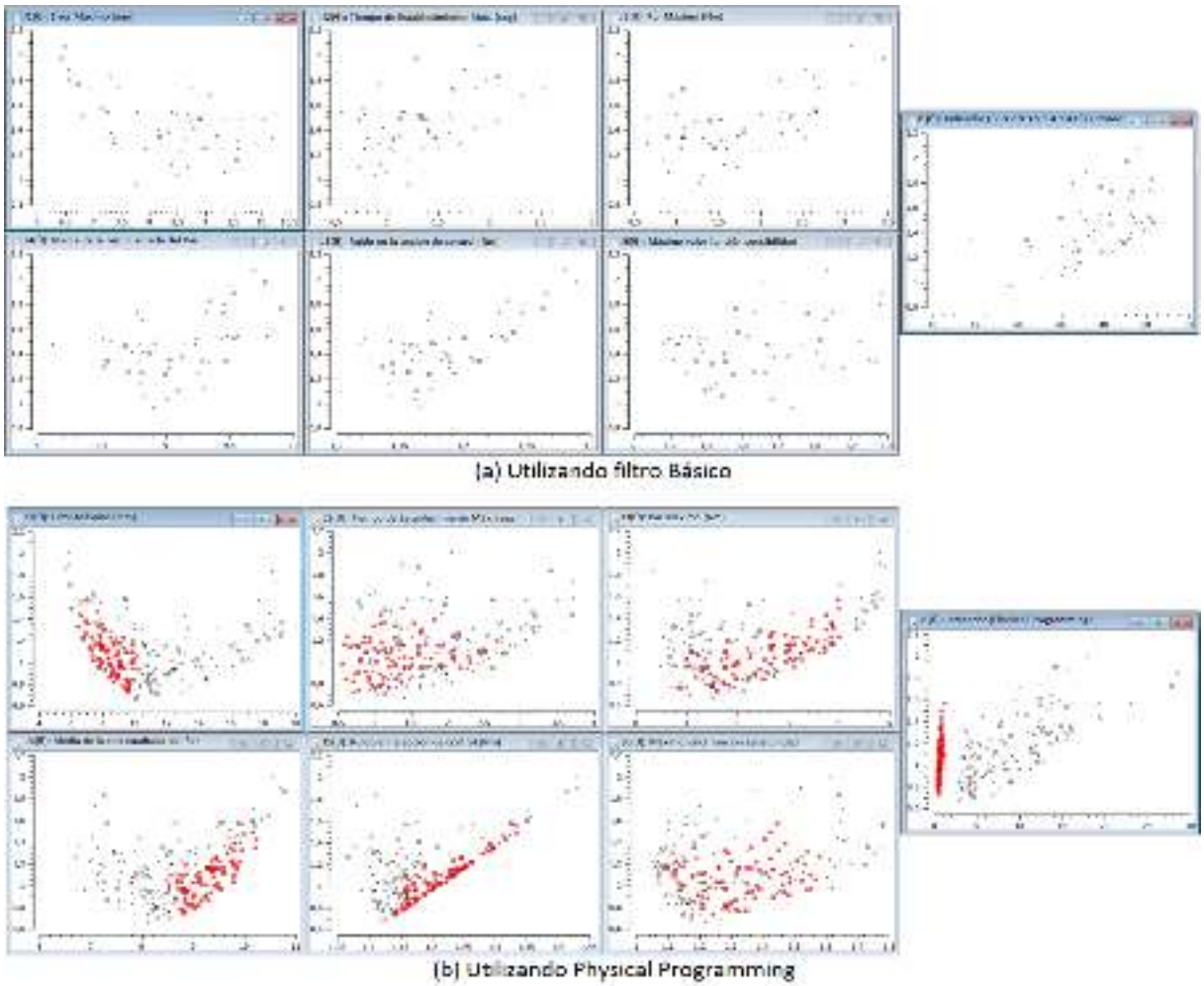


Figura 10. Visualización del frente de Pareto, con seis objetivos, y un indicador, usando LD-Tool.

Además se puede sobre los datos originales, o sobre otros si lo deseamos, realizar otro más avanzado, por ejemplo del tipo Global Physical Programming, tal como se explica en la sección 3. Para realizar este paso, se deberán introducir los rangos del filtro, en este ejemplo se introducen los rangos útiles para el diseñador, tal como se observa en la figura número 11. Una vez calculado, se obtendrán un nuevo indicador, ordenado de menor a mayor, es decir, de solución más a menos idónea, figura número 12. Exportando estos datos desde la aplicación, y visualizándolos en LD-Tool obtenemos la figura número 10b. Como se observa, en el indicador hay una zona de soluciones diferenciada de las demás, son las soluciones de la izquierda que aparecen en color rojo. Estas son las que minimizan el rendimiento calculado anteriormente en el indicador mediante el filtro de la aplicación. Seleccionando la solución más idónea según el filtro Physical Programming, obtenemos como resultados las gráficas mostradas en la figura 13 (condiciones del benchmark).

Por último, se pueden generar tres ficheros de texto, con los objetivos, variables e indicadores, tal como se explica en la sección 3.



Figura 11. Hoja de datos (Physical Programming).



Figura 12. Hoja de datos.

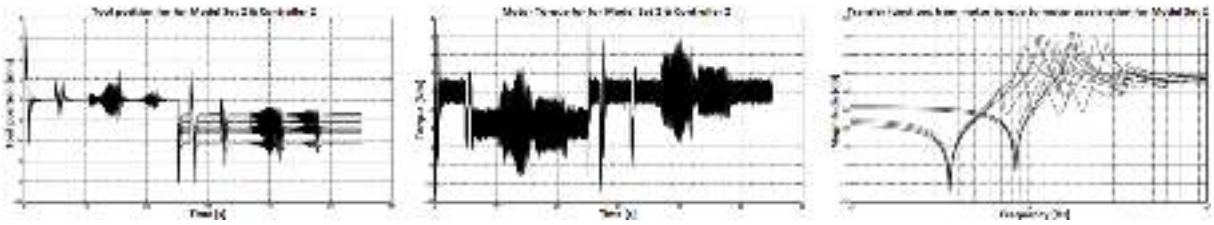


Figura 13. Rendimiento del controlador seleccionado bajo las condiciones del Benchmark [8] (Set II)

5. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo, se ha presentado una aplicación destinada a la etapa de *selección multicriterio*, dentro de la metodología de diseño en ingeniería mediante la *optimización multiobjetivo*. La *optimización multiobjetivo* permite determinar un conjunto de soluciones óptimas, sobre las cuales se seleccionará la más idónea. Elegir una única solución no es trivial y suele llevar más tiempo que el propio proceso de búsqueda. Por ello, nuevas herramientas para dicho proceso son valoradas por el diseñador. Con este propósito se ha diseñado esta aplicación, que proporciona al diseñador una herramienta para comparar prestaciones de cada una de las soluciones, para llegar a la mejor de ellas. Además esta aplicación puede exportar los datos para visualizarlos en cualquier herramienta de visualización.

La aplicación queda abierta a que cualquier usuario externo pueda ampliar sus funciones y añadir algún filtro que ayude al cometido de la misma.

Agradecimientos

El presente trabajo ha sido financiado parcialmente por los proyectos **ENE2011-25900** y **TIN2011-2808**, (Ministerio de Economía y Competitividad, Gobierno de España) y por la Universitat Politècnica de València a través del programa de formación **FPI-UPV-2010/19** y el proyecto **SP20120460** (Acciones PAID-05-12).

Referencias

[1] Reynosa-Meza, G., Blasco, X., Sanchis, J., Martínez, M., -. Algoritmos evolutivos y su empleo en el ajuste de controladores del tipo PID: Estado actual y perspectivas. Revista Iberoamericana de Automática e Informática Industrial – Aceptado.

[2] Coello, C. A. C., Lamont, G. B., Veldhuizen, D. A. V., 2007. Multi-criteria decision making. In: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation Series. Springer US, pp. 515-545.

[3] Coello, C. A. C., Veldhuizen, D. V., Lamont, G., 2002. Evolutionary algorithms for solving multi-objective problems. Kluwer Academic press.

[4] Roy, R., Hinduja, S., Teti, R., 2008. Recent advances in engineering design optimisation: Challenges and future trends. CIRP Annals – Manufacturing Technology 57 (2), 697 – 715

[5] Lotov, A., Miettinen, K., 2008. Visualizing the Pareto frontier. In: Branke, J., Deb, K., Miettinen, K., Slowinski, R. (Eds.), Multiobjective Optimization. Vol. 5252 of Lecture Notes in Computer Science. Springer Berlin / Hiedelberg, pp. 213-243

[6] Blasco, X., Herrero, J., Martínez, M., 2008. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. Information Sciences 178 (20), 3908-3924.

[7] Messac, A., Mattson, C., 2002. Generating well-distributed sets of Pareto points for engineering design using Physical Programming. Optimization and Engineering 3, 431-450, 10.1023/A:1021179727569

[8] Moberg, Öhr, J., Gunnarsson, S., 2009. A Benchmark Problem for Robust Feedback Control of a Flexible Manipulator. IEEE Transactions on control systems technology, Vol. 17, Nº. 6.

[9] Reynoso-Meza G., Guerrero Mario, G., Sanchis, J., Blasco, X., 2011. Análisis multi-objetivo y multi-criterio del Swedish Open Championship in robot control 2004. 10º Congreso Nacional de Mecatrónica.

[10] Blasco, X., García-Nieto, S., Reynoso-Meza, G., 2012. Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor. Simulación y evaluación de propuestas. Revista Iberoamericana de Automática e Informática industrial 9 (2012) 194–199.

CONTROL DE POSICIÓN Y ORIENTACIÓN DE UNA PLATAFORMA OMNIDIRECCIONAL NO-HOLONOMA

Pablo Bengoa Ganado[a], Josu Larrañaga Leturia[b], Eloy Irigoyen[a]
[a]ETSIB, UPV/EHU, Alameda Urquijo s/n. 48013 - Bilbao, {pbengoa002, eloy.iriboyen}@ehu.es
[b]Tecnalia Research & Innovation, Paseo Mikeletegi, 7 E-20009 Donostia - San Sebastián, josu.larrañaga@tecnaila.com

Resumen

En este trabajo se estudia el control de trayectoria de plataformas no-holonomas con un número de ruedas directrices igual o superior a 2. A lo largo del artículo se plantea y analiza un modelo cinemático genérico para dichas plataformas, independiente del número de ruedas. Posteriormente, se desarrolla un controlador de realimentación dinámica capaz de controlar tanto la posición como la orientación de la plataforma. El control de la orientación aporta un factor diferenciador a este trabajo con respecto a planteamientos anteriores. La posibilidad de controlar la orientación en vez de la curvatura de la trayectoria hace que el control de la plataforma sea más intuitivo para el operario destinado al manejo de la plataforma. Finalmente, dicho controlador es validado a través de la simulación en diferentes escenarios.

Palabras Clave: Plataforma no-holonoma, omnidireccional, control no lineal, dynamic feedback linealization, tracking.

1 INTRODUCCIÓN

El comportamiento cinemático y el control de sistemas no-holonomos (como por ejemplo robots móviles con ruedas, satélites, manos multidedo, etc) fueron ampliamente abordados en los años noventa. Esta tendencia vino motivada por la definición de la condición de rango de Brockett sobre la controlabilidad de los sistemas no-holonomos [1]. Pese a los numerosos avances realizados en la época, las conclusiones presentadas no abordaban soluciones más allá de los desarrollos matemáticos y los prototipos de laboratorio [24], [16]. Recientemente, sin embargo, el uso de robots móviles con ruedas, como plataformas onmidireccionales no-holonomas, para el servicio de personas discapacitadas y ancianas y principalmente para fines industriales, ha hecho de esta una temática emergente dentro del campo de la Robótica [21], siendo ejemplo de ello [4], [8], [12].

Los robots móviles con ruedas, independientemente de su configuración física, pueden ser clasificados en base a los tres parámetros que definen la maniobrabilidad de un robot móvil (δ_s, δ_m y δ_M), como se presenta en [19]. El grado de maniobrabilidad (δ_M) de un robot viene definida por la suma de: por un lado, la movilidad definida por las restricciones de deslizamiento, denominado grado de movilidad (δ_m); y por otro, los grados de libertad que ofrecen las ruedas directrices, siendo este el grado de direccionalidad (δ_s).

A lo largo de los años noventa se analizaron muy ampliamente los sistemas $\delta_m = 1$ y $\delta_s = 1$, como son los triciclos o los robots tipo coche [14], y los sistemas $\delta_m = 2$ y $\delta_s = 0$, por ejemplo los robots diferenciales [16], [26]. Siendo las plataformas omnidireccionales, las cuales cumplen que $\delta_M = 3$, estudiadas en menor medida [20], [9].

Las plataformas omnidireccionales pueden estar construidas mediante sistemas con una configuración $\delta_m = 3$ y $\delta_s = 0$, $\delta_m = 2$ y $\delta_s = 1$ ó $\delta_m = 1$ y $\delta_s = 2$. Las dos primeras configuraciones exigen la necesidad de usar ruedas omnidirecciones. El uso de este tipo de ruedas hace que la plataforma pueda moverse en cualquiera de los tres grados de libertad sin realizar ninguna maniobra ni alinear las ruedas, véase Figura 1. Sin embargo, en [25]se enumeran las limitaciones que este tipo de ruedas tienen, no siendo apropiadas para su uso en aplicaciones industriales. La tercera de las configuraciones, las de tipo $\delta_m = 1$ y $\delta_s = 2$, por el contrario, dispone de al menos dos ruedas directrices. Esto permite, pese a ser necesario el alineamiento de las ruedas, posicionar el centro instantáneo de giro (ICR) en cualquier punto del plano de movimiento mediante el uso de ruedas convencionales. Es por ello que, a lo largo de este artículo, se abordará el problema de control de seguimiento sobre una trayectoria (tracking), para una plataforma con esta configuración.

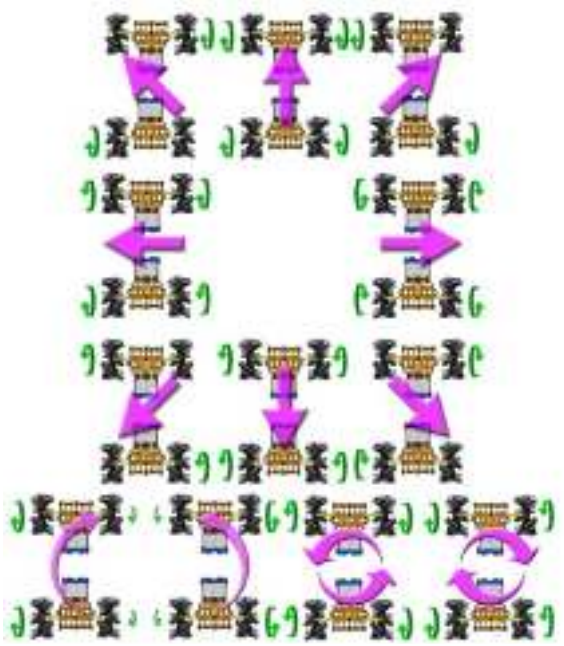


Figura 1. Referencia de las velocidades para poder desplazarse en los tres grados de libertad: eje "x", eje "y", y orientación.

En la literatura existen diferentes artículos donde se aborda esta problemática. Hashimoto et al. en su trabajo [9], por ejemplo, dividen la plataforma en módulos, uno por cada rueda directriz. En primer lugar elabora un controlador de alto nivel que es el responsable de referenciar a los controles de bajo nivel de cada uno de los módulos. En [18] Reister y Unseren por su parte, construyen un controlador en base al análisis de las fuerzas y los pares de las ruedas. En [6] y en [5], Connette et al. definen un espacio tridimensional a partir de las coordenadas "x", "y" y la orientación de la plataforma. A partir de este espacio realizan un controlador en coordenadas esféricas. Por último, en [24] se define un modelo independiente al número de ruedas directrices y se plantea una ley de linealización por realimentación dinámica de estados para el control de la posición "x" e "y" y la curvatura de la referencia.

A partir de la literatura obtenida, en este trabajo se presenta una modificación en la linealización realizada por Thuilot et al. en [24] de forma que pueda ser controlada la posición en los ejes "x" e "y" y la orientación de la plataforma.

Este artículo sigue la siguiente distribución: en primer lugar, en la sección 2, se analiza en detalle el modelo matemático de la plataforma. Seguidamente, la sección 3 aborda el diseño de la ley de linealización entrada-salida dinámica y el controlador. Posteriormente, el artículo analiza los resultados obtenidos en simulación en la sección 4. Las conclusiones y las líneas futuras se desarrollan en la sección 5.

2 MODELO DE LA PLATAFORMA

2.1 DEFINICIÓN DE LAS SUPOSICIONES DE PARTIDA

A lo largo de esta sección se desarrolla un modelo de robot rígido construido a partir de $n \geq 2$ ruedas no deformables. Como se demuestra en [7] y [3] las ruedas auto alineadas, Figura 2.c (ruedas castor, por ejemplo) no afectan en el movimiento del robot, por lo que no se tendrán en cuenta a la hora del modelado de la planta. Por lo tanto, las ruedas a tener en cuenta para el modelado son:

Ruedas directrices: Son aquellas ruedas que son controlables tanto en orientación como en rotación. Se considera que estas ruedas están ancladas mediante un eje vertical rígido a un punto fijo del chasis del robot, véase Figura 2.a.

Grupos diferenciales: Son aquellos grupos de dos ruedas conectadas rígidamente entre ellas y ancladas mediante un eje orientable vertical al chasis del robot. A diferencia de las ruedas directrices este tipo de ruedas solo son controlables en rotación, y a partir de la diferencia de las velocidades de las ruedas es posible controlar la orientación del grupo. Es por ello que, es posible definir una velocidad lineal y una orientación equivalente en el punto de anclaje. A partir de estos parámetros equivalentes, los grupos diferenciales y las ruedas directrices pueden ser modeladas de igual forma, véase Figura 2.b.

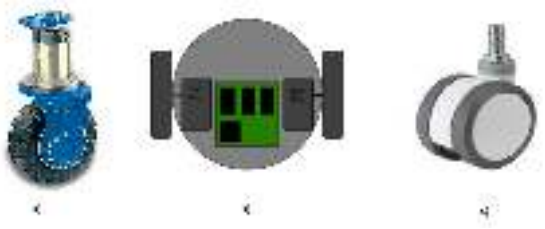


Figura 2. Tipos de ruedas. a) Rueda directriz, b) grupo diferencial, c) rueda auto alineada.

Debido a que a lo largo de este trabajo solo se analiza el comportamiento cinemático de la plataforma, se considerarán las variaciones de las ruedas directrices directamente como variables de entrada al control.

Por otro lado, se asumirá que el robot se desplaza por un plano horizontal, libre de obstáculos. El contacto entre las ruedas y el suelo, por su parte, se considera puntual y que estas giran sin deslizamiento. De esta manera, las restricciones aseguran que la velocidad en el eje de desplazamiento en el punto de contacto es igual a cero (se desplaza sin deslizar) y que en la dirección ortogonal al desplazamiento es también cero (se desplaza sin derrapar).

2.2 DEFINICIÓN DE LAS COORDENADAS A CONTROLAR

Si se define cada uno de los anclajes de los ejes de cada una de las n ruedas como W_i (siendo $1 \leq i \leq n$), se puede colocar el origen del sistema de referencia móvil anclado al robot en W_1, de forma que se crea un sistema de coordenadas ortogonal. Por facilidad de interpretación de los parámetros se definirá el vector expuesto en (1) como vector de salida de la planta.

$$\varepsilon = (x, y, \theta)^T \tag{1}$$

A partir del eje de referencia móvil, la posición de cada uno de los W_i puede ser definida mediante el uso de los parámetros constantes de construcción del robot. Estos parámetros (l_i y α_i) son la posición en coordenadas polares (l_i: distancia y α_i: ángulo) de cada una de las i ruedas directrices que componen la plataforma. Del mismo modo se puede definir el ángulo β_i como el ángulo relativo entre $\overrightarrow{W_1 W_i}$ y la perpendicular de la rueda i. De la misma manera, se define φ como la velocidad angular de la rueda i. Si se agrupan en vectores cada una de las variables, véase (2), el carro queda totalmente definido en cada instante a partir de los vectores ε, β, φ.

$$\begin{aligned} \beta &= (\beta_1, \beta_2, .., \beta_n)' \\ \varphi &= (\varphi_1, \varphi_2, .., \varphi_n)' \end{aligned} \tag{2}$$

2.3 DEFINICIÓN DEL MODELO

El modelo de la plataforma a estudiar, definido en [9], está desarrollado en este trabajo a partir del principio de movimiento instantáneo de Descartes. Este desarrollo, a diferencia de en [23] donde se desarrolla el mismo modelo desde una perspectiva mucho más matemática, está enfocado desde un punto de vista geométrico. Este método permite ver conceptualmente el desarrollo realizado, sin perderse en una matemática de alta complejidad.

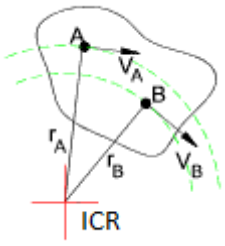


Figura 3. Definición del centro instantáneo de rotación (ICR).

El principio de Descartes afirma que el movimiento instantáneo de cualquier sólido rígido sobre una superficie plana puede definirse como la rotación alrededor de un centro instantáneo de rotación (ICR), véase Figura 3. Tomando la translación pura como un caso límite de la rotación (centro instantáneo de rotación posicionado en el infinito) es fácilmente deducible que dicho principio se cumple constantemente. Por otro lado, es directo, también, concluir que el ICR debe estar localizado en la línea imaginaria que atraviesa perpendicularmente cada una de las ruedas. Por lo tanto, solo será posible obtener un ICR común si y solo si:

Teorema 1: Las líneas perpendiculares de cada rueda se cortan en un mismo punto, o si son todas ellas paralelas (punto de corte en el infinito).

Teniendo esto en cuenta se denominará Configuración Admisible de Ruedas (CAR) a aquellas configuraciones de β que satisfagan el requerimiento definido por el Teorema 1. Así mismo, cada CAR tiene un único ICR. Esta propiedad, cuya demostración se desarrolla en [23], es fácilmente demostrable.

Por lo tanto, si se define $\eta_1$como la velocidad lineal de la plataforma de manera que se cumpla (3) y σ como la inversa de la distancia algebraica del punto W1 al ICR, según (4); entonces el único movimiento asociado a cada CAR está definido en espacio de estados por (5).

$$\eta_1 = R \cdot \dot{\varphi}_1 \tag{3}$$

$$\sigma = \frac{1}{d(\overline{W_1, ICR})} \tag{4}$$

$$\left.\begin{aligned} \dot{x} &= \sin(\theta + \beta_1) \cdot \eta_1 \\ \dot{y} &= -\cos(\theta + \beta_1) \cdot \eta_1 \\ \dot{\theta} &= \sigma \cdot \eta_1 \end{aligned}\right\} \tag{5}$$

Como se puede apreciar en (5), para definir la velocidad lineal del vehículo basta con un grado de libertad, η_1. Para definir el ICR, sin embargo, es necesario utilizar dos, η_2 y η_3. El primero de ellos, η_2, define la velocidad con la que varía el ángulo β_1; en cambio, el segundo, η_3, define la velocidad de variación de la inversa de la distancia, σ. Teniendo en cuenta todo lo expuesto anteriormente, se puede definir completamente una plataforma de n ruedas directrices mediante la expresión (6).

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta}_1 \\ \dot{\sigma} \end{pmatrix} = \begin{pmatrix} \sin(\theta + \beta_1) & 0 & 0 \\ -\cos(\theta + \beta_1) & 0 & 0 \\ \sigma & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{pmatrix} \tag{6}$$

De esta forma se obtiene un modelo que satisface a la expresión descrita en (7), donde x es el vector de salidas y η el vector de entradas.

$$\dot{x} = S(x) \cdot \eta \tag{7}$$

Aplicando el álgebra de Lie, definido en [13], se puede obtener el rango de los corchetes de Lie de la plataforma, (8). A partir del rango, es posible obtener las características de controlabilidad del sistema, [10].

$$rango\{s_1, s_2, s_3, [s_1, s_2], [s_1, s_3]\} = 5 \tag{8}$$

A partir del análisis de la ecuación (8), se demuestra que el sistema es completamente no holonomo, como se afirma en [10]. Campion et al. en [2] demuestra que es posible garantizar mediante leyes suaves de linealización realimentada de estados, la estabilidad marginal global con convergencia a cero de los sistemas con el mismo número de funciones de salida que grados de libertad. Por lo que, en base a los desarrollos realizados en [2] y partiendo de que el rango del modelo desarrollado es cinco (mismo número de grados de libertad que funciones de salida) y de que la técnica de control a utilizar es la linealización realimentada, se concluye que:

1) El sistema es completamente controlable.
2) El sistema no es linealizable mediante leyes de realimentación de estados estáticas.
3) Como consecuencia del principio de Brockett, [1], el modelo no es estabilizable mediante leyes de realimentación suaves invariantes en el tiempo.

3 CONTROL DE TRACKING MEDIANTE LINEALIZACIÓN EXACTA DINÁMICA

En esta sección se aborda el problema de seguimiento de una trayectoria de referencia sin configuración de reposo. En las siguientes líneas se analiza cómo, para un robot móvil con ruedas, el control de linealización de estados dinámica (dynamic state feedback linealization) resuelve apropiadamente este problema. Un desarrollo parecido se realiza en [24], pero a diferencia del trabajo realizado por Thuilot et al., este artículo demuestra que es posible controlar de forma independiente la posición "x" e "y" y la orientación de la plataforma.

Antes de empezar a desarrollar una ley de control específica para el modelo deducido en la sección anterior, se estima conveniente introducir brevemente la teoría de linealización de estados dinámica (análisis más detallados sobre la técnica se pueden encontrar en [10] y en [15]).

Se considera un sistema a controlar que satisfaga la ecuación (9), donde x es el vector l-dimensional de estados y η el vector m-dimensional de entradas.

$$\dot{x} = f_0(x) + f(x) \cdot \eta \tag{9}$$

El diseño de la linealización de estados estática consiste en encontrar una función de salida de igual dimensión que el vector de entradas (m-dimensional) tipo $\varphi = \varphi_1(x), \dots, \varphi_m(x)$ y una función m-dimensional de vectores de integración tipo $\mu = \mu_1(x), \dots, \mu_m(x)$ tal que:

1) la aplicación Φ definida en (10) sea un dioformismo.

$$\begin{aligned} &\Phi: \Re^n \to \Re^n \; x \to \\ &\Phi = \left(\varphi_1(x), \dots, \frac{d^{\mu_1}\varphi_1(x)}{dt^{\mu_1}}, \dots, \right. \\ &\left. \varphi_m(x), \dots, \frac{d^{\mu_m}\varphi_m(x)}{dt^{\mu_m}}\right)^T \end{aligned} \tag{10}$$

2) la derivada $\mu + 1$ de φ, ecuación (11), forme una matriz $\Delta(x)$ que sea invertible.

$$\begin{aligned} &\frac{d^{\mu+1}\varphi(x)}{dt^{\mu+1}} \\ &= \left(\frac{d^{\mu_1+1}\varphi_1(x)}{dt^{\mu_1+1}}, \dots, \frac{d^{\mu_m+1}\varphi_m(x)}{dt^{\mu_m+1}}\right)^T \\ &= f_1(x) + \Delta(x) \cdot \eta \end{aligned} \tag{11}$$

En caso de que se cumpla las condiciones descritas arriba, (10-11), entonces existe una ley de control que satisface la ecuación (12),

$$\eta = \Delta^{-1}(x) \cdot (\eta_a - f_1(x)) \tag{12}$$

donde η_a es un vector m-dimensional de entradas auxiliares. A partir de esta ley de control es claramente deducible que el sistema a controlar mediante el vector de entradas auxiliar es un sistema lineal.

Como se ha afirmado en la sección 2, el sistema definido en (6) es completamente no holonomico, por lo tanto, no es posible aplicarle una linealización de estados estática, ya que esta infringe la condición (11). La explicación física de que la matriz $\Delta(x)$ no sea invertible es que la derivada de orden $\mu + 1$ de la función $\varphi(x)$ está en función de parte de las entradas pero no por el vector de entradas completo.

El algoritmo de extensión dinámico propuesto en [11] y [22] propone, en este caso, incorporar al vector de entradas, las entradas o la combinación de entradas que aparecen en la derivada de orden $\mu + 1$ y considerar su derivada con el nuevo vector de entradas. Por lo tanto, fruto de este cambio, el robot queda representado por el modelo original aumentado con nuevas entradas que son la integración de las entradas iniciales. En el trabajo se referirá a este nuevo modelo como el modelo extendido del robot. Esta extensión del modelo original permite el linealizado de los estados del sistema. Para obtener el vector de entrada del modelo real, simplemente se requiere de integraciones online de parte del nuevo vector de entrada (vector de entradas extendido). Es debido a esta integración online que este algoritmo es conocido como linealización dinámica de estados o dynamic state feedback linealization.

Por lo tanto, siempre y cuando no se trabaje en las configuraciones singulares descritas abajo, es posible aplicar una linealización dinámica completa en cualquier robot con dos o más ruedas directrices. Dichas configuraciones singulares se dan cuando:

1) Se sitúa el ICR en el eje de alguna de las ruedas (singularidades del modelo).
2) Se pasa por una configuración de reposo (singularidad de la ley linealización).

A continuación se procede a desarrollar el algoritmo descrito para la planta modelada en la sección 2. Para ello, y a diferencia de [24], se mantendrá el vector de salida del sistema definido en (13), conteniendo éste las posiciones "x" e "y" y la orientación de la plataforma.

$$\varphi = \varepsilon = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \tag{13}$$

En (5) se puede observar cómo la entrada $\eta_1$multiplica a cada uno de los tres componentes del vector de salida, por lo que no es posible realizar derivadas de mayor orden que uno. Por lo tanto, las condiciones (9) y (10) no se satisfacen, y en consecuencia la linealización estática completa del sistema (6) con respecto al vector de salida (13) no es posible, como se dedujo cuando se ha analizado el orden del sistema (6) (sección 2). A partir del algoritmo de extensión dinámica, se deduce que se debe derivar la entrada η_1, de forma que se obtenga el modelo extendido representado en (14),

$$\begin{aligned} \dot{x} &= \sin(\theta + \beta_1) \cdot \eta_1 \\ \dot{y} &= -\cos(\theta + \beta_1) \cdot \eta_1 \\ \dot{\theta} &= \sigma \cdot \eta_1 \\ \dot{\beta_1} &= \eta_2 \\ \dot{\sigma} &= \eta_3 \\ \dot{\eta_1} &= \eta_4 \end{aligned} \tag{14}$$

donde el vector de estados y el vector de entradas son (15) y (16), respectivamente.

$$x_e = (x, y, \theta, \beta_1, \sigma, \eta_1)^T \tag{15}$$

$$\eta_e = (\eta_1, \eta_2, \eta_3, \eta_4)^T \tag{16}$$

Se puede observar que el sistema extendido definido en (14) es, siempre y cuando se satisfaga el hecho de evitar singularidades, completamente linealizable estáticamente, ya que existe un vector μ que satisface las relaciones (10) y (11).

Partiendo del sistema extendido (14), es directo obtener mediante la derivada de φ el sistema (17), que satisface la ecuación (11).

$$\begin{aligned} \ddot{\varphi} &= \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta + \beta_1) \cdot \sigma \cdot \eta_1^2 \\ \sin(\theta + \beta_1) \cdot \sigma \cdot \eta_1^2 \\ 0 \end{pmatrix} \\ &+ \begin{pmatrix} \cos(\theta + \beta_1) \cdot \eta_1 & 0 & \sin(\theta + \beta_1) \\ \sin(\theta + \beta_1) \cdot \eta_1 & 0 & -\cos(\theta + \beta_1) \\ 0 & \eta_1 & \sigma \end{pmatrix} \\ &\cdot \begin{pmatrix} \eta_2 \\ \eta_3 \\ \eta_4 \end{pmatrix} \end{aligned} \tag{17}$$

Es fácilmente demostrable cómo la matriz $\Delta(x_e)$ del sistema (17) es invertible, siempre y cuando $\eta_1 \neq 0$ y por lo tanto solo falta comprobar si, a partir de la ecuación (10), la aplicación Φ, definida en (18), es un dioformismo.

$$\Phi: \Re^6 \rightarrow \Re^6 \; x_e \rightarrow \Phi = \left(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}\right)^T \tag{18}$$

Tras su desarrollo se deduce que (18) no es un dioformismo ya que para cualquier valor de x_e solo existe un valor Φ posible. Pero sin embargo, para un valor de Φ existen dos posible valores de x_e, véase (19). Esto es debido a que el mismo ICR se puede definir mediante los parámetros (σ, β_1) o mediante $(-\sigma, \beta_1 + \pi)$. En cambio, para poder pasar de uno a otro, es necesario cruzar el punto singular de $\eta_1 = 0$. Como las configuraciones singulares han de ser evitadas, una vez definido el signo inicial de la velocidad del robot, la incertidumbre se despeja,

cumpliéndose por tanto las condiciones del dioformismo.

$$\Phi \rightarrow \begin{cases} \text{if } \eta_1(0) > 0 \\ x_e = \left(x, y, \theta, -\theta - arctan\left(\frac{\dot{x}}{\dot{y}}\right) + \right. \\ \left. f(\dot{y}) \cdot \pi, \frac{\dot{\theta}}{\sqrt{\dot{x}^2 + \dot{y}^2}}, \sqrt{\dot{x}^2 + \dot{y}^2}\right)^T \\ \text{if } \eta_1(0) < 0 \\ \left. [1 - f(\dot{y})] \cdot \pi, \frac{\dot{\theta}}{\sqrt{\dot{x}^2 + \dot{y}^2}}, \sqrt{\dot{x}^2 + \dot{y}^2}\right)^T \end{cases} \quad (19)$$

Aplicando el dioformismo (19) y la ley de control (12), el robot con el modelo extendido puede ser descrito mediante el sistema lineal (20).

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} 0_{3X3} & I_{3X3} \\ 0_{3X3} & 0_{3X3} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0_{3X3} \\ i_{3X3} \end{pmatrix} \cdot \eta_a \quad (20)$$

El controlador a realizar para el control del sistema extendido es, por lo tanto, un controlador lineal. En el caso más sencillo, se propone un PD con referencia de aceleración según se define en (21),

$$\eta_a = H_1 \cdot (\varphi_{ref} - \varphi) + H_2 \cdot (\dot{\varphi}_{ref} - \dot{\varphi}) + \ddot{\varphi}_{ref} \quad (21)$$

siendo φ_{ref} la referencia de la trayectoria a seguir por el vector de salida φ, y H_1 y H_2 dos matrices diagonales positivas y constantes. Aplicando esta ley de control, se obtiene la matriz definida en (22) como matriz A del sistema lineal en lazo cerrado, siendo esta por construcción una matriz estabilizada según los criterios de Lyapunov.

$$A = \begin{pmatrix} 0_{3x3} & I_{3x3} \\ H_1 & H_2 \end{pmatrix} \quad (22)$$

Por último, cabe destacar que como se ha comentado a lo largo de la sección existe una singularidad cuando la velocidad lineal de la plataforma es nula, $\eta_1 = 0$. Esta singularidad, sin embargo, es inevitable ya que la plataforma comenzará y finalizará a ser controlada en reposo. Se han desarrollado diferentes alternativas basadas generalmente en un modo de control híbrido donde se podría utilizar un control para arrancar la plataforma, el control de linealización dinámica desarrollado en este artículo durante todo el trayecto de la plataforma, y finalmente, una nueva estrategia de control para detenerla. Una de las posibilidades barajadas es la descrita en [17], donde se presenta un modelo híbrido que consiste en utilizar un control en lazo abierto para arrancar la plataforma y un control basado en una ley de control variante en el tiempo para detenerla.

4 RESULTADOS DE SIMULACIÓN

La ley de control desarrollada a lo largo de la sección 3 ha sido implementada en simulación a una plataforma de cuatro ruedas directrices. Para ello se ha considerado una plataforma cuadrada de un metro cuadrado. Las cuatro ruedas se han posicionado en los vértices de la plataforma. La simulación se ha realizado en tres escenarios diferentes:

a) Una trayectoria circular sobre ICR situado en el punto cartesiano [x = -1 m, y = 0 m] con una orientación fija de θ = 0 rad.
b) Una trayectoria circular sobre ICR situado en el punto cartesiano [x = -0.5 m, y = -0.5 m] con una orientación de las ruedas constante.
c) Una rotación pura sobre el centro de la plataforma.

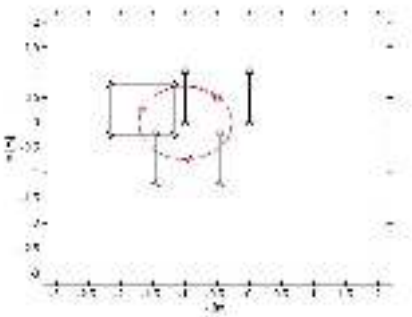


Figura 4. Una trayectoria circular sobre ICR situado en el punto cartesiano x=-1m y=0m con una orientación fija de θ=0rad.

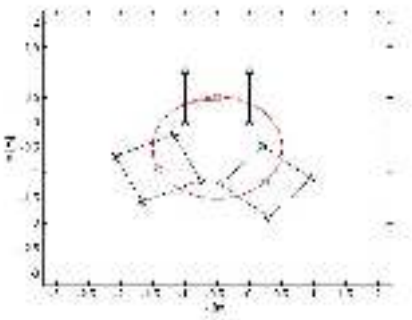


Figura 5. Una trayectoria circular sobre ICR situado en el punto cartesiano x=-0.5m y=-0.5m con una orientación de las ruedas constante.

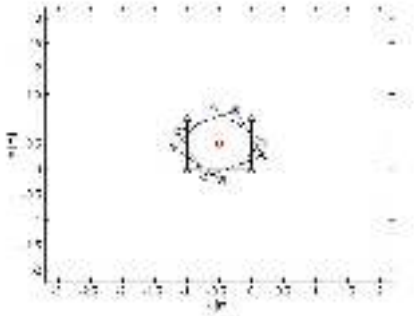


Figura 6. Una rotación pura sobre el centro de la plataforma.

Para la simulación de estas trayectorias se han definido las matrices H_1 y H_2 como $40 \cdot I_{3x3}$ y $11 \cdot I_{3x3}$. De esta manera se obtiene una respuesta con un tiempo de establecimiento de 1 segundo.

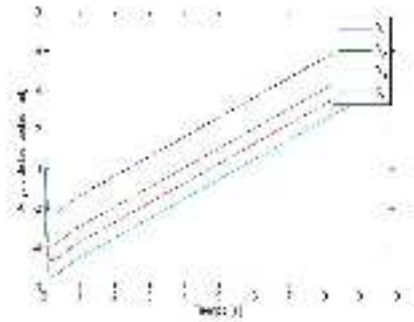


Figura 7. Evolución de las ruedas a lo largo de la trayectoria a)

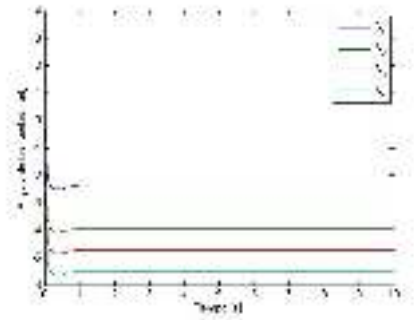


Figura 8. Evolución de las ruedas a lo largo de la trayectoria b)

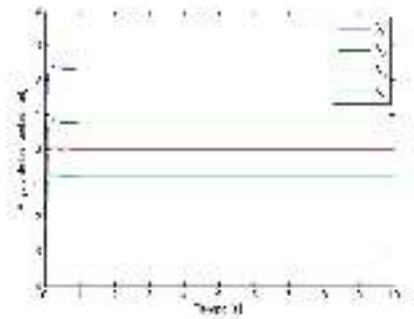


Figura 9. Evolución de las ruedas a lo largo de la trayectoria c)

En la Figura 4, Figura 5 y Figura 6 se observan las trayectorias realizadas por la plataforma móvil para el caso de la trayectoria a), b) y c) respectivamente.
Si se analiza la evolución de las ruedas a lo largo de cada una de las trayectorias; véanse Figura 7, Figura 8 y Figura 9; se puede ver cómo durante los primeros instantes las ruedas tratan de alinearse debidamente para lograr realizar la trayectoria referenciada y cómo posteriormente estas toman una evolución lineal. En el caso de la trayectoria a) el ángulo de las ruedas aumenta de forma constante, mientras que en las trayectorias b) y c) se mantiene en un valor constante.

5 CONCLUSIONES

En este trabajo se ha estudiado el problema del control de trayectoria de un robot móvil equipado con más de dos ruedas directrices. Mediante el desarrollo realizado se ha demostrado que a partir de una correcta elección de variables de estado se puede obtener un modelo cinemático común para cualquier robot con más de dos ruedas directrices, independientemente del número de estas y de su configuración espacial.

A partir de dicho modelo, se ha desarrollado un control de realimentación dinámica en donde, a diferencia de lo desarrollado en la literatura, se mantienen explícitamente controladas las posiciones "x" e "y" y la orientación de la plataforma. A partir de los desarrollos matemáticos se ha visto que no es posible el control de la plataforma cuando la velocidad lineal de esta es nula. Es por ello que es necesario implementar un modelo de control híbrido donde un controlador independiente (un controlador en lazo abierto, por ejemplo) realiza las operaciones de arranque y parada de la plataforma. Mientras que en el resto de la maniobra del robot el controlador desarrollado en el presente artículo es el encargado de dirigir la plataforma a lo largo de la trayectoria.

A partir de las simulaciones obtenidas se aprecia como las diferentes ruedas que componen la plataforma evolucionan de forma continua, evitando así, el desgaste de las ruedas debido a deslizamientos y arrastres no deseados.

Tras la publicación de este artículo, la línea investigadora continúa mediante la implementación del algoritmo desarrollado en una plataforma industrial real donde se validará los resultados obtenidos en simulación.

Agradecimientos

Los autores quieren agradecer la colaboración a los miembros del departamento de Robótica de la División de Transporte de Tecnalia. Del mismo modo, agradecer también la colaboración del Grupo de Investigación de Control Inteligente de la UPV/EHU.

Referencias

[1] Brockett R. Asymptotic stability and feedback stabilization. In: Differential Geometric Control Theory. 1983:181–91.

[2] Campion G. Modelling and state feedback control of nonholonomic mechanical systems. Decision and Control, … 1991;

[3] Campion G. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. Robotics and Automation, … 1996; 12.

[4] Chen G, Zhang W, Zhang X. Speed Tracking Control of a Vehicle Robot Driver System Using Multiple Sliding Surface Control Schemes. International Journal of Advanced Robotic Systems 2013; :1.

[5] Connette C, Parlitz C. Singularity avoidance for over-actuated, pseudo-omnidirectional, wheeled mobile robots. … and Automation, 2009 … 2009; :4124–30.

[6] Connette CP, Pott A, Hagele M, Verl A. Control of an pseudo-omnidirectional, non-holonomic, mobile robot based on an ICM representation in spherical coordinates. 2008 47th IEEE Conference on Decision and Control 2008; :4976–83.

[7] d'Andrea-Novel B. Modelling and control of non-holonomic wheeled mobile robots. Robotics and Automation, … 1991; :1130–5.

[8] Fierro R, Lewis FL. Control of a nonholonomic mobile robot using neural networks. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council 1998; 9:589–600.

[9] Hashimoto M, Fujiwara I, Oba F, Corporation MM. Path Tracking Control of a Non-Holonomic Modular Omnidirectional Vehicle. :637–42.

[10] Isidori A. Nonlinear control systems. Springer; 1995.

[11] J. DESCUSSE, MOOGa CH. Decoupling with dynamic compensation for strong invertible affine non-linear systems. International Journal of Control 1985; :1387–98.

[12] Kang H-S, Kim Y-T, Hyun C-H, Park M. Generalized Extended State Observer Approach to Robust Tracking Control for Wheeled Mobile Robot with Skidding and Slipping. International Journal of Advanced Robotic Systems 2013; :1.

[13] Lie, S. and Engel F. Theorie der Transformationsgruppen. 1893.

[14] Luca A De. Robot motion: planning and control. Laas; 1982. Retrieved 20 June 2013

[15] Nijmeijer H, Schaft A Van Der. Nonlinear dynamical control systems. 1990.

[16] Petrov P. Modeling and adaptive path control of a differential drive mobile robot. … WSEAS international conference on Automatic control, … 2010; :403–8.

[17] Pomet J-B, Thuilot B, Bastin G, Campion G. A hybrid strategy for the feedback stabilization of nonholonomic mobile robots. Proceedings 1992 IEEE International Conference on Robotics and Automation 129–34.

[18] Reister DB, Unseren M a. Position and constraint force control of a vehicle with two or more steerable drive wheels. IEEE Transactions on Robotics and Automation 1993; 9:723–31.

[19] Roland Y. Siegwart IRN. Introduction to Autonomous Mobile Robotos. MIT Press; 2004.

[20] Samson C. Velocity and torque feedback control of a nonholonomic cart. Advanced Robot Control 1991;

[21] Song J-B, Byun K-S. Steering control algorithm for efficient drive of a mobile robot with steerable omni-directional wheels. Journal of Mechanical Science and Technology 2009; 23:2747–56.

[22] Sontag E. Feedback stabilization of nonlinear systems. Robust control of linear systems and nonlinear control 1990;

[23] Thuilot B. Modeling and Asymptotic Stabilization of Mobile Robots Equipped With Two or More Steerittg Wheels. 1992.

[24] Thuilot B. Modeling and feedback control of mobile robots equipped with several steering wheels. Robotics and Automation, … 1996; 12. Retrieved 20 June 2013 from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=499820

[25] Wada M, Mori S. Holonomic and omnidirectional vehicle with conventional tires. Robotics and Automation, 1996 Proceedings … 1996; :3671–6.

[26] Yun X, Yamamoto Y. On feedback linearization of mobile robots. Technical Reports (CIS) 1992;

VALIDACIÓN EXPERIMENTAL DE ESTRATEGIAS DE REDUCCIÓN DEL CONSUMO DE SENSORES INALÁMBRICOS EN SISTEMAS DE CONTROL EN RED.

Ignacio Peñarrocha, Adrian Dinu, Roberto Sanchis
Departament d'Enginyeria de Sistemes Industrials i Disseny
Universitat Jaume I, Castelló, Spain 12071
{ipenarro,dinu,rsanchis}@uji.es

Resumen

En este trabajo se describe una red inalámbrica experimental que se ha desarrollado para la prueba de algoritmos de control. La plataforma dispone de tres nodos; el controlador, el sensor y el actuador. Cada nodo utiliza un microcontrolador en el que se puede implementar diferentes algoritmos de control. La comunicación se basa en una red inalámbrica de área personal de baja tasa de transmisión (IEEE 802.15.4), y se han probado varias estrategias para ahorrar baterías en los sensores inalámbricos, para mitigar el bajo ratio de transmisión o la transmisión con baja potencia. Para aumentar la vida útil de la batería de los sensores inalámbricos, se han explorado dos políticas diferentes de envío: la disminución de la potencia de la transmisión inalámbrica, y el uso de estrategias de muestreo y envío basado en eventos. El trabajo describe en detalle la plataforma experimental que se ha desarrollado y muestra los resultados experimentales obtenidos tras probar las dos estrategias indicadas.

Palabras clave: Control en red, control basado en eventos, redes inalámbricas.

1 INTRODUCCIÓN

En los últimos años, se ha mejorado notablemente la tecnología inalámbrica y han aparecido numerosos dispositivos inalámbricos para su uso en el control industrial [1, 2, 3, 4]. El uso de elementos inalámbricos evita la necesidad de cablear los dispositivos a una unidad central de control, reduciendo así los costes de instalación. Sin embargo, los sensores inalámbricos utilizan una batería que debe reemplazarse de vez en cuando. De hecho, si el sensor se utiliza para controlar un sistema inestable o de dinámica muy rápida, que necesita una frecuencia de muestreo elevada, la vida útil de las baterías podría ser muy baja. Para superar el problema del consumo de potencia, se ha desarrollado el estándar IEEE 802.15.4, pero su uso en el control de procesos todavía es un área en la que hay muchos problemas que resolver [5, 6, 7].

El uso de redes inalámbricas de bajo consumo para el control de procesos también tiene el inconveniente de ser sensible al entorno y a las distancias, lo que puede ocasionar la pérdida de datos entre sensores y controladores, y entre éstos y actuadores. Además, el uso de una red de datos inalámbrica introduce un retardo en la adquisición de datos en el controlador, y en el tiempo que transcurre desde que se calcula una acción de control hasta que se aplica al proceso, lo que contribuye a desestabilizar el lazo de control. Esto significa que el uso de controladores convencionales (como el PID) con los datos adquiridos a través de una red inalámbrica puede llevar el sistema de control a tener un comportamiento no deseado. Uno de los objetivos de este trabajo es estudiar cómo debe ser el diseño en los controladores PID para superar estos problemas de retardos y pérdida de datos en el bucle de control.

El consumo de la batería en sensores inalámbricos se ha estudiado en diferentes trabajos como [8], en los que se muestra que el envío de mensajes es la tarea que conlleva mayor consumo [9, 10, 11]. Para reducir el consumo energético de los sensores inalámbricos se pueden adoptar diferentes estrategias [12]. En este trabajo se consideran las estrategias que se indican a continuación. La primera consiste en enviar los datos a un periodo fijo (el periodo de control), pero con una potencia de transmisión reducida. El inconveniente de este enfoque es que la probabilidad de perder datos puede ser significativa, ya que las transmisiones se hacen más sensibles a factores del entorno. Este enfoque requiere un dispositivo transmisor en el que la potencia se pueda cambiar. La segunda estrategia consiste en enviar mediciones con una potencia adecuada que evite problemas del entorno (es decir, que evite la pérdida incontrolada de datos), pero cuya transmisión se produzca sólo cuando es estrictamente necesario, es decir, cuando el sensor tiene información relevante [13]. Para implementar este enfoque el sensor debe estar equipado con cierta capacidad de cálculo, para así decidir si un dato debería ser enviado o no. Cualquiera de estas dos estrategias implica que el controlador no tenga todas las mediciones de la salida a controlar de forma regular. El otro objetivo de este tra-

bajo es explorar la relación entre las estrategias de envío (mediante el control de potencia o mediante estrategias basadas en eventos), el diseño del controlador (paradigma de codiseño [13]) y las prestaciones que se pueden alcanzar [14].

Diferentes trabajos como [15, 16, 17, 18] analizan modificaciones en la implementación de algoritmos PID para adaptarlos al muestreo basado en eventos. En este trabajo, se analiza el comportamiento de un controlador PID convencional en un lazo cerrado a través de una red de comunicaciones inalámbrica y se relacionan sus prestaciones con la potencia consumida por el nodo sensor y con la estrategia de muestreo y envío utilizada por éste.

La estructura del trabajo es la siguiente. En la sección 2 se presenta el problema del control en red. En la sección 3 se describe la plataforma experimental desarrollada. En la sección 4 se presentan los resultados experimentales obtenidos mediante el uso de la plataforma y de las diferentes estrategias de comunicación. Finalmente, se resumen las conclusiones en la sección 5.

2 MOTIVACIÓN

Sea un proceso continuo definido por las ecuaciones

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t), w(t)),\\ y(t) &= g(x(t)) + v(t),\end{aligned}$$

con $x(t)$ el estado, $u(t)$ la acción de control, $w(t)$ la perturbación del estado, $y(t)$ la salida medida y $v(t)$ el ruido de medida que afecta a $y(t)$. Asúmase que el proceso se controla a través de una red inalámbrica con un nodo sensor, un nodo controlador y un nodo actuador que intercambian su información. El nodo sensor mide la salida del proceso cada T segundos y envía el valor medido a la unidad de control. Llámese a este valor $y_s(t)$. El nodo actuador comprueba constantemente si hay una nueva acción de control a aplicar al proceso (mediante un retenedor de orden cero). Llámese a esta acción de control $u_a(t)$. Se asume que el nodo controlador implementa un controlador PID estándar que recibe la salida medida mediante el canal inalámbrico y lo utiliza también para enviar la acción de control al nodo actuador.

El controlador PID estándar se implementa en tiempo discreto por medio de una rutina que se ejecuta de forma periódica en los instantes de tiempo t_k (siendo el periodo de la rutina $T = t_k - t_{k-1}$). Las principales ecuaciones que implementan el controlador PID son[1]

$$\begin{aligned}I_k &= I_{k-1} + \frac{K_p\, T}{T_i}(r_k - y_k),\\ D_k &= \frac{T_d}{NT + T_d} D_{k-1} +\\ &\quad + \frac{K_p\, T_d N}{NT + T_d}(c\, r_k - y_k - c\, r_{k-1} + y_{k-1}),\\ u_k &= K_p(b\, r_k - y_k) + I_k + D_k,\end{aligned}$$

donde y_k hace referencia a la última salida adquirida del nodo sensor que, idealmente, se obtiene al principio de la tarea, es decir, se supone que $y_k = y_s(t_k)$. u_k hace referencia a la acción de control que se calcula en el instante k y que se envía al nodo actuador e, idealmente, se aplica de forma inmediata al proceso, es decir $u_a(t) = u_k$, para $t_k \leq t < t_{k+1}$. K_p, T_d, T_i y N son las ganancias del controlador, los tiempos derivativo e integral y el factor de filtrado, respectivamente, mientras que b y c son los factores de ponderación de la referencia.

Cuando se utiliza una red inalámbrica de bajo ratio de transmisión, los retardos inducidos por la red (tanto del sensor al controlador como del controlador al actuador) no son despreciables. Esto implica que el valor que se utiliza para calcular la acción de control, y_k, no es $y_s(t_k)$, y el valor calculado para actualizar la entrada del actuador, u_k, no se aplica inmediatamente. Esto puede expresarse como

$$\begin{aligned}y_k &= y_s(t_k - \tau_{s,k})\\ u_a(t) &= u_k,\ \forall t_k + \tau_{a,k} \leq t < t_{k+1} + \tau_{a,k+1}\end{aligned}$$

para ciertos retardos $\tau_{s,k}$, $\tau_{a,k}$. Además, la red está sujeta a ciertas perturbaciones del entorno que pueden ocasionar que ciertas entradas y salidas se pierdan. Otro factor que causa que algunas mediciones se pierdan es la falta de sincronización entre los nodos. El algoritmo anterior se aplica utilizando la salida más reciente adquirida y, si hay una pérdida de dato (es decir, no hay una nueva medición respecto del último periodo), se utiliza la última que se recibió.

Cuando se ajustan controladores PID a partir de un modelo de un proceso, se asumen condiciones de comunicación de datos ideales, ya que las perturbaciones ambientales que afectan a los retardos y a la pérdida de datos en la red inalámbrica no pueden ser reproducidas. Cuando el controlador diseñado se aplica a través una red inalámbrica, su comportamiento final depende del comportamiento de la red, y tanto los retrasos como la

[1] Nótese que se debería implementar también un mecanismo de saturación y anti-windup para evitar comportamientos no deseados debidos a la saturación del actuador.

pérdida de datos pueden causar la degradación o incluso la desestabilización de la planta controlada. El uso de un controlador ajustado con un modelo ideal puede ser inapropiado para estabilizar la planta y para obtener unas propiedades adecuadas en cuanto al seguimiento de referencias o a la atenuación de perturbaciones cuando las condiciones de uso de la red son adversas. Incluso si no se inestabiliza el bucle de control, podría aumentar la sobreoscilación o el tiempo de establecimiento, o el valor de índices de coste del rechazo de perturbaciones (como la integral del error absoluto). Este empeoramiento de prestaciones es mucho más pronunciado cuando se utilizan estrategias de envío de mediciones con baja potencia o mediciones basadas en eventos.

Por esta razón, es interesante el desarrollo de métodos de control que tengan en cuenta las estrategias de envío de datos para alcanzar un compromiso entre potencia consumida (por el sensor) y prestaciones del bucle de control. También es interesante probar dichas estrategias de control en plataformas reales para evaluar los resultados y proponer mejoras realistas sobre las metodologías de diseño ya existentes en la literatura.

En este trabajo se propone diseñar los controladores PID siguiendo el procedimiento descrito en [19]. El método se basa en maximizar la ganancia integral fijando un margen de fase igual al requerido, y un margen de ganancia mayor o igual que un límite inferior prefijado. La maximización de la ganancia integral es próxima a la minimización de la integral del error absoluto (IAE) frente a perturbación si la respuesta no es demasiado oscilatoria, lo cual se garantiza con los márgenes de fase y ganancia. Para resolver la optimización se utiliza un parámetro auxiliar adimensional, que es el cociente entre el cero del controlador y la frecuencia de cruce de ganancia. Para el diseño del PID se necesita un modelo del proceso. Una alternativa es utilizar un modelo identificado en condiciones ideales (sin la red inalámbrica), y tener en cuenta el efecto de la red modificando las especificaciones de robustez en el diseño. Otra opción es identificar el modelo a partir de datos experimentales obtenidos a través de la misma red que se utilizará en el control. De esa forma el modelo utilizado ya incluye ciertos efectos no ideales (como retardos de comunicación).

3 DESARROLLO DE LA PLATAFORMA

En esta sección se describe la plataforma desarrollada, que permite analizar las prestaciones del uso de PIDs convencionales sobre redes inalámbricas, así como probar controladores alternativos con las estrategias de envío descritas anteriormente.

Para el desarrollo de la plataforma, se han utilizado tres microcontroladores dsPIC33FJ64GP206A de Microchip y se han seleccionado unas unidades periféricas MRF24J40MB para la comunicación inalámbrica. Las unidades de comunicación se basan en IEEE 802.15.4 y permiten acceder al registro que controla la potencia utilizada para la transmisión de cada mensaje. Uno de los microcontroladores implementa el algoritmo de control y toma decisiones acerca de cómo actuar sobre el proceso a través de las mediciones de salida adquiridas (es decir, hace de nodo controlador). El segundo microcontrolador es el responsable de muestrear la salida del proceso (nodo sensor) y de enviarla al controlador. Finalmente, un tercer microcontrolador recibe la acción de control que se ha de aplicar al proceso. Los nodos actuador y sensor se han seleccionado con diferentes periféricos para conectarse a una amplia variedad de procesos con la interfaz adecuada.

La plataforma incluye también tres cables de comunicación RS232 que conectan los nodos con un PC para adquirir todos los mensajes intercambiados entre nodos. Esto permite monitorizar también la pérdida de datos entre el nodo controlador y el nodo actuador, y entre el nodo sensor y el nodo controlador debido a los retardos, la falta de sincronización, la transmisión con baja potencia o el muestreo basado en eventos. Este esquema de monitorización serie permite también lanzar la orden de inicio de los experimentos y medir las prestaciones conseguidas con las diferentes condiciones de operación que se quiere explorar. La figura 1 muestra los diferentes elementos de la plataforma experimental.

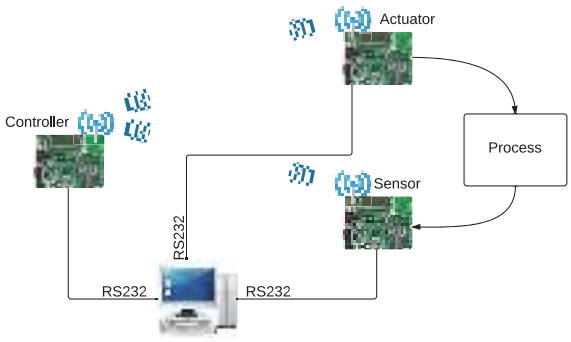


Figura 1: Plataforma experimental.

4 RESULTADOS EXPERIMENTALES

En esta sección se muestran los resultados obtenidos con la plataforma desarrollada. Los experimentos muestran el compromiso existente entre la potencia consumida por el nodo sensor durante el envío de mediciones, y las prestaciones del bucle cerrado. El nodo controlador está configurado para transmitir permanentemente con la máxima potencia y de forma periódica.

Inicialmente, se ha implementado un muestreo y envío constante de mediciones con la máxima potencia en el nodo sensor. Se ha probado la plataforma para obtener los retrasos entre nodos, y se ha obtenido un retardo total medio de $\tau_s + \tau_a = 20$ ms en los canales controlador-actuador y sensor-controlador. Para probar la plataforma y las estrategias de control se ha utilizado un motor de corriente continua y un amplificador en puente en H que recibe señales moduladas por ancho de pulso. El motor de CC tiene un potenciómetro que mide la posición angular. Se ha utilizado el convertidor analógico digital del nodo sensor para adquirir la señal de tensión del potenciómetro (véase figura 2). Considerando el

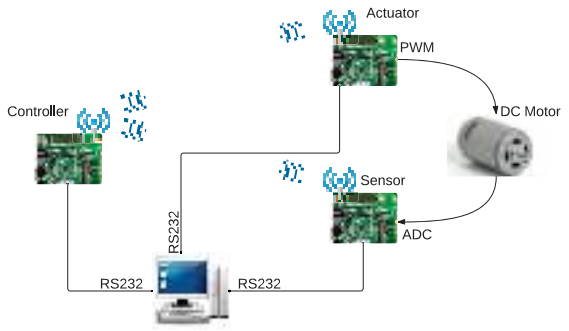


Figura 2: Plataforma experimental con el motor controlado.

retraso medio del bucle de control, se ha utilizado un modelo en función de transferencia de la forma

$$G(s) = \frac{K}{s(1+\tau s)} e^{-ds}.$$

Se ha realizado un experimento de identificación para el motor que, junto con la información del retraso indicada anteriormente, nos lleva a los valores $K = 86.2$ (grados/voltio), $\tau = 34.5$ ms, y $d = 20$ ms. Utilizando este modelo se han calculado controladores PI para diferentes valores del margen de fase, asegurando siempre un margen de ganancia mínimo de 8 dB, y tratando de minimizar el IAE ante perturbación según la técnica presentada en [19]. Los controladores obtenidos se resumen en la tabla 1. Para la implementación

M_F	K_p	T_i	Marca
35°	0.157	0.262	.
40°	0.139	0.320	o
45°	0.122	0.397	x
50°	0.105	0.502	+
55°	0.091	0.665	$\Diamond$
60°	0.077	0.881	∇

Tabla 1: Resumen de los controladores PI diseñados.

de los controladores se ha elegido un periodo de $T = 10$ ms. Se ha observado que la falta de sincronización entre los relojes de los nodos, junto con el retraso indicado anteriormente, da lugar en la práctica a un retraso de uno, dos o tres periodos en cada canal. Para estudiar el comportamiento de cada controlador se ha realizado un experimento como el mostrado en la figura 3, en el que la referencia inicialmente en 0° cambia en forma de escalón en el instante 6 s. Se ha introducido también una perturbación en forma de escalón a partir del instante 12 s, que se ha implementado sumando un valor constante a la acción de control calculada. Para analizar los resultados que se obtienen con diferentes controladores y estrategias de envío de mediciones, se utilizarán como índices a comparar tanto la sobreoscilación como los siguientes tres índices IAE:

1. $IAE_1 = T\sum |r_k - y_{s,k}|$, $k = 600, \ldots, 700$, para el cambio de la referencia en forma de escalón;
2. $IAE_2 = T\sum |r_k - y_{s,k}|$, $k = 1200, \ldots, 1300$, para la atenuación de la perturbación;
3. $IAE_3 = T\sum |r_k - y_{s,k}|$, $k = 1800, \ldots, 1900$, para el comportamiento en régimen permanente, afectado por el ruido de medición.

Para comparaciones posteriores, se define la energía relativa unitaria de transmisión, como la energía necesaria para poder realizar el experimento anterior con el primer controlador presentado, y con envío periódico de mediciones con máxima potencia de transmisión en el nodo sensor.

4.1 ESTRATEGIA BASADA EN EVENTOS

En esta sección se analiza la estrategia de envío basada en un cambio significativo de la salida ("Send On Delta"). El nodo sensor hace una comparación periódica entre la medición actual de la salida $y_{s,k}$ y la que se envió por última vez $y_{s,last}$. Sólo se envía la medición actual si la diferencia entre los dos valores mencionados es mayor que un

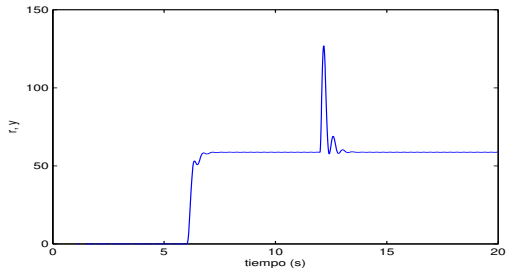


Figura 3: Salida medida en el nodo sensor en un experimento base.

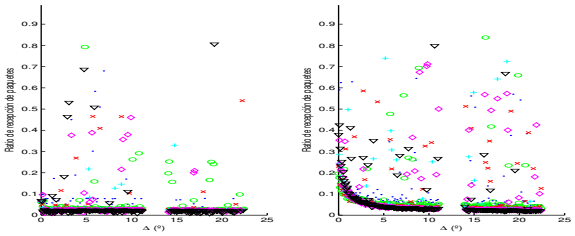


Figura 4: Número relativo de datos enviados con envío basado en eventos.

valor prefijado, es decir, si

$$|y_{s,last} - y_{s,k}| \geq \Delta.$$

La relación entre las prestaciones conseguidas y el valor de Δ se ha analizado mediante experimentos utilizando los distintos controladores PI indicados anteriormente. Cada controlador se ha utilizado con valores de Δ que varían de 0 a 23 grados. En la figura 4 se muestra el número de mediciones enviadas en términos relativos respecto a las enviadas en el escenario ideal (2000 muestras). En la figura de la izquierda se muestra el número de mediciones relativas en el experimento con referencia y perturbación constante, mientras que en la derecha se muestra el experimento en el que se producen los cambios de referencia y perturbación mostrados en la figura 3. Nótese que cuando Δ es grande, el número de envíos es pequeño, incluso en el experimento en el que la salida tiene grandes variaciones.

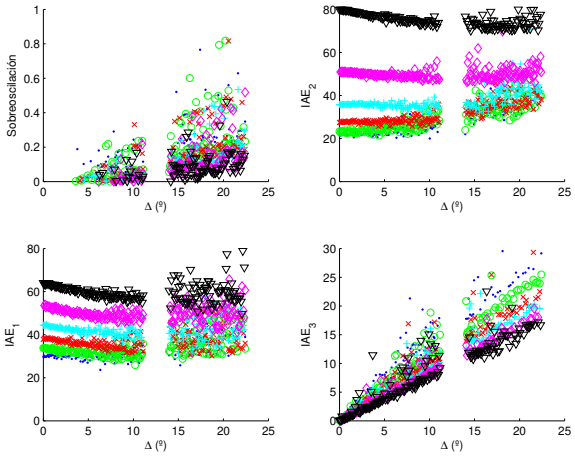


Figura 5: Prestaciones de los controladores PI en función de Δ.

En la figura 5 se muestran las prestaciones obtenidas con cada controlador y con cada valor de Δ. Se observa que incrementar Δ no afecta en exceso a los valores IAE ante cambios de referencia y de perturbación, pero sí que afecta a la sobreoscilación y al IAE de régimen permanente. Se puede observar que incrementar el valor de Δ hasta 5° no afecta en exceso los valores de IAE pero, sin embargo, se requiere para su implementación menos del 10% del número de envíos requeridos con muestreo convencional.

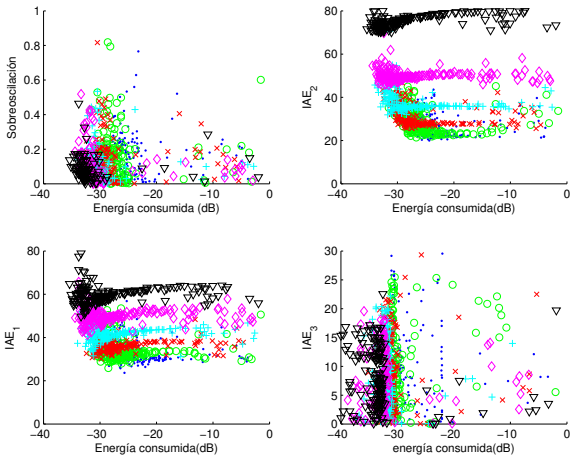


Figura 6: Prestaciones de los controladores PI en función de la energía consumida en la estrategia basada en eventos.

En la figura 6 se muestran las prestaciones obtenidas en función del número relativo de datos enviados, que puede interpretarse como la energía necesaria en la transmisión de datos por parte del sensor con respecto a la energía que necesitaría en un experimento con una potencia de transmisión máxima y envío periódico de datos. Se observa que las prestaciones pueden mantenerse prácticamente constantes hasta una energía consumida de, aproximadamente, 0.1 (-20 dB).

4.2 ESTRATEGIA DE CONTROL DE POTENCIA

En esta sección se analiza el efecto de la potencia de transmisión del nodo sensor sobre el comportamiento del sistema en bucle cerrado. Para cada posible valor de la potencia de transmisión, se han realizado diferentes experimentos utilizando los diferentes controladores PI indicados en la tabla 1. En cuanto la potencia de la transmisión se ha disminuido con respecto al máximo valor posible en la antena disponible, el número de datos que se han perdido durante la transmisión se ha incrementado. En la figura 7 se muestra el efecto de la potencia de transmisión sobre el ratio de disponibilidad de paquetes (número de paquetes recibido con éxito con respecto al número de enviados). Ambas gráficas muestran este ratio para un experimento de duración 20 segundos (2000 muestras), siendo el de la izquierda un experimento con referencia y perturbación constante, y el de la derecha un experimento con cambios en escalón en ambas señales. Se observa cómo el ratio de éxito en el envío de mediciones es independiente del tipo de señal (a diferencia del muestreo basado en eventos del experimento anterior). El eje horizontal muestra la atenuación de la potencia utilizada en la antena en dB con respecto a la máxima potencia disponible en la antena. Se observa que la relación entre la transmisión de potencia y la probabilidad de perder datos no es determinista y que hay una gran sensibilidad y dispersión para potencias por debajo de -30 dB. Durante la experimentación se ha observado que el proceso de pérdida de datos durante la comunicación aparece generalmente como grupos de datos perdidos consecutivos, lo que significa que la pérdida de datos no es un proceso estadísiticamente independiente. Este comportamiento se puede apreciar en la figura 8 para una potencia de transmisión de -20 dB (los ceros significan datos perdidos, y los unos, recibidos con éxito).

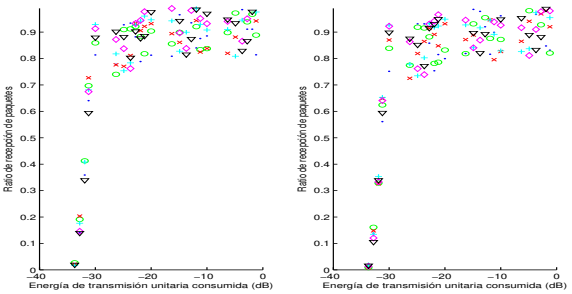


Figura 7: Ratio de recepción de paquetes en función de la potencia de transmisión.

En la figura 9 se observa el efecto que tiene la potencia de transmisión sobre las prestaciones

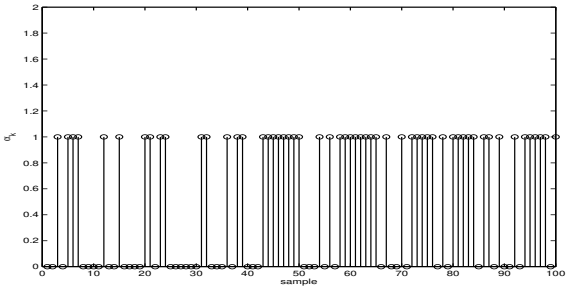


Figura 8: Recepción y pérdida de datos durante 100 muestras para una potencia de transmisión de -20 dB.

conseguidas en bucle cerrado con cada controlador y cada potencia. Es interesante notar que para potencias reducidas hasta -30 dB respecto de la máxima las prestaciones no empeoran, debido a que básicamente el ratio de mensajes recibidos permanece constante. Sin embargo, en cuanto se baja por debajo de -30 dB el empeoramiento es muy brusco (en las gráficas de prestaciones no aparecen los puntos porque el motor se sale del rango de una vuelta del potenciómetro). Este límite de -30 dB indica el punto óptimo de operación para maximizar la duración de las baterías. Sin embargo, este límite depende de las condiciones físicas del entorno (distancia, presencia de obstáculos, etc.), por lo que para una aplicación real habría que determinarlo mediante experimentos. En futuros trabajos se plantea desarrollar estrategias para adaptar la potencia de transmisión a las condiciones del entorno, con el objetivo de minimizar el consumo del sensor garantizando unas prestaciones determinadas del sistema controlado.

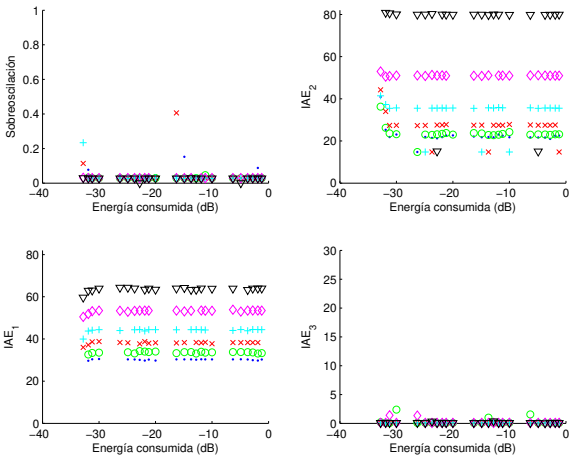


Figura 9: Prestaciones de los controladores PI para diferentes potencias de transmisión.

5 CONCLUSIONES

En este trabajo se describe un esquema de red inalámbrica experimental que se ha desarrollado para probar diferentes algoritmos de control. Se presentan algunos resultados experimentales interesantes, comparando dos estrategias diferentes para reducir la energía consumida de los nodos. La plataforma tiene tres nodos basados en microcontrolador: el controlador, el sensor y el actuador. La comunicación se basa en una red inalámbrica de área personal (IEEE 802.15.4) en la que la potencia de transmisión de cada nodo puede modificarse. Los diferentes nodos se conectan con un PC para capturar todas las señales con el propósito de realizar análisis posteriores. Para poder alargar la vida útil de las baterías de los sensores inalámbricos, se han explorado y comparado dos estrategias diferentes de envío en el nodo sensor: envío basado en eventos con máxima potencia, y envío periódico a baja potencia. La plataforma inalámbrica ha funcionado correctamente y es útil para probar diferentes estrategias de muestreo, envío y control. Con respecto de la estrategia de envío periódico a baja potencia, una conclusión interesante es que para un rango amplio de potencias de transmisión, las prestaciones son iguales que la nominal, dado que prácticamente no se pierden datos. Cuando la potencia se reduce más, el comportamiento de la red inalámbrica es muy sensible a cambios en el entorno, como la presencia de personas, y se tiene un comportamiento incierto en el que algunas veces se pierden muchas muestras de forma consecutiva, y otras veces pocas. Una conclusión que se deriva de la experimentación en cuanto a la estrategia a utilizar en los nodos sensores podría ser la siguiente: para minimizar la potencia consumida del nodo sensor, reducir la potencia de transmisión hasta un valor que garantice un alto grado de recepción satisfactoria de mensajes, y entonces aplicar una estrategia de envío basado en eventos con un valor de Δ que minimiza el número de transmisiones al mismo tiempo que mantiene las prestaciones del bucle. Si las condiciones del entorno son cambiantes, la potencia óptima de transmisión debería ir cambiándose para adaptarla al entorno. Esta idea será objeto de futuros trabajos.

Agradecimientos

Este trabajo ha sido financiado por el proyecto MICINN número DPI2011-27845-C02-0.

Referencias

[1] M. Vieira, C. Coelho, D. da Silva, and J. da Mata, "Survey on wireless sensor network devices," in *EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.03TH8696)*, vol. 1. IEEE, 2003, pp. 537–544.

[2] D. Feng, C. Jiang, G. Lim, L. J. Cimini, G. Feng, and G. Y. Li, "A Survey of Energy-Efficient Wireless Communications," *IEEE Communications Surveys & Tutorials*, no. 1, pp. 167–178, Jan.

[3] R. A. Gupta, "Networked Control System: Overview and Research Trends," *IEEE Transactions on Industrial Electronics*, no. 7, pp. 2527–2535, Jul.

[4] J. A. Giraldo and N. Quijano, "Current results and research trends in networked control systems," in *IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, 2011 IEEE.* IEEE, Oct., pp. 1–6.

[5] L. Zhang, H. Gao, S. Member, and O. Kaynak, "Network-Induced Constraints in Networked Control Systems: A Survey," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 1, pp. 403–416, 2013.

[6] V. Gungor and G. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 10, pp. 4258–4265, 2009.

[7] M. Tabbara, D. NeËť, D. Nesic, and A. R. Teel, "Stability of Wireless and Wireline Networked Control Systems," *Automatic Control, IEEE Transactions on*, no. 9, pp. 1615–1630, Sep.

[8] A. Moller and U. Jonsson, "Input-output analysis of power control in wireless networks," *Automatic Control, IEEE Transactions on*, no. 4, pp. 834–846, Apr.

[9] L. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213).* IEEE, pp. 1548–1557.

[10] A. El Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu, and S. Zahedi, "Energy-efficient scheduling of packet transmissions over wireless networks," in *Proceedings.Twenty-First Annual Joint*

Conference of the IEEE Computer and Communications Societies, no. c. IEEE, pp. 1773–1782.

[11] W. Stark, A. Worthen, S. Lafortune, and D. Teneketzis, "Low-energy wireless communication network design," *IEEE Wireless Communications*, no. 4, pp. 60–72, Aug.

[12] J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch, "Comparative Study of Energy-Efficient Sampling Approaches for Wireless Control Networks," *IEEE Transactions on Industrial Informatics*, no. 3, pp. 416–424, Aug.

[13] W. Heemels, K. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, Dec., pp. 3270–3285.

[14] I. Lestas, "Power Control in Wireless Networks: Stability and Delay Independence for a General Class of Distributed Algorithms," *IEEE Transactions on Automatic Control*, no. 5, pp. 1253–1258, May.

[15] V. Vasyutynskyy and K. Kabitzsch, "First order observers in event-based PID controls," in *2009 IEEE Conference on Emerging Technologies & Factory Automation*. IEEE, Sep., pp. 1–8.

[16] ——, "A comparative study of PID control algorithms adapted to send-on-delta sampling," in *2010 IEEE International Symposium on Industrial Electronics*. IEEE, Jul., pp. 3373–3379.

[17] B. Hensel, J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch, "A simple PI controller tuning rule for sensor energy efficiency with level-crossing sampling," *International Multi-Conference on Systems, Sygnals & Devices*, pp. 1–6, Mar.

[18] J. A. Romero, N. J. N. J. Pascual, I. Peñarrocha, and R. Sanchis, "Event-Based PI controller with adaptive thresholds," in *IV International Congress on Ultra Modern Telecommunications and Control Systems 2012*. St. Petersburg: IEEE, Oct., pp. 219–226.

[19] R. Sanchis, J. Romero, and P. Balaguer, "Tuning of PID controllers based on simplified single parameter optimisation," *International Journal of Control*, vol. 83, no. 9, pp. 1785–1798, 2010.

SISTEMA DE DETECCIÓN DE FALLOS BASADO EN TÉCNICAS DE ERROR ACOTADO Y UMBRAL DINÁMICO

A. Suárez y P. Abad
Departamento de Tecnologías de la Información, Universidad de Huelva email:{asuarez,abadhe.abad}@uhu.es
J.M. Bravo
Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática, Universidad de Huelva
email:caro@uhu.es

Resumen

Este trabajo presenta un nuevo método de detección de fallos aplicable a sistemas dinámicos. El método de detección está basado en datos histórico de entrada-salida del sistema y en la señal denominada residuo. Cada instante de tiempo, el sistema de detección consulta el histórico para aproximar localmente la dinámica del sistema sin fallos y así obtener una predicción de la salida del sistema con la que calcular el residuo. Así mismo, aplicando técnicas de error acotado, se obtiene el umbral máximo permitido para dicho residuo. La expresión que permite obtener dicho umbral es la principal aportación de la comunicación. La detección se basa en comprobar si el residuo supera el umbral máximo marcado por el método.

Palabras clave: Detección de fallos, error acotado, métodos no paramétricos.

1. Introducción

La detección de fallos se puede entender como el proceso mediante el cual se detecta una anomalía de comportamiento o una desviación del funcionamiento correcto del sistema [17]. Existe una amplia bibliografía que cubre el conjunto de técnicas utilizadas en el campo de la detección de fallos. Desde un punto de vista global, estas técnicas se pueden clasificar en dos grandes grupos: aquellas basadas en modelos explícitos de los sistemas [8],[6],[12] y aquellas basadas en datos históricos del sistema [15]. Los métodos de detección basados en modelo suelen utilizar una señal denominada residuo. Varias son las técnicas presentes en la literatura que permiten obtener dicho residuo: observadores [7], relaciones de paridad [9], estimación de parámetros [11] o el filtro de Kalman [16]. Por otro lado, los métodos de detección de fallo basados en datos históricos usan grandes cantidades de datos obtenidos de los procesos industriales. Estos métodos utilizan sistemas de inferencia basados en técnicas estadísticas como pueden ser el análisis de componentes principales (PCA) [15], los mínimos cuadrados parciales (PLS) [15] o el análisis discriminante de Fisher [18].

Las técnicas basadas en error acotado asumen que las medidas obtenidas de un sistema están afectadas por un error acotado, siendo la cota conocida [14]. En identificación de sistemas, el objetivo de estas técnicas es encontrar el conjunto de parámetros que es consistente con las medidas de entrada-salida obtenidas del sistema, la cota del error presente en las medidas y el modelo paramétrico considerado [4] y [5]. Esta clase de métodos de identificación se ha aplicado al desarrollo de varios algoritmos de detección de fallos [10], [3] y [20].

Este trabajo presenta un nuevo método de detección de fallos basado en datos históricos y señal residuo. El método se caracteriza por no utilizar un modelo explícito del sistema para generar el residuo. Cada instante de tiempo, el sistema de detección consulta el histórico de datos con el fin de obtener una predicción de la salida con la que generar el residuo. Dicha predicción se calcula utilizando una aproximación local del comportamiento dinámico del sistema. Así mismo, el sistema de detección genera un umbral máximo para el residuo. Para obtener dicho umbral se utilizan técnicas de error acotado. La expresión explícita que proporciona dicho umbral es la principal aportación del trabajo. El sistema de detección se puede aplicar a sistemas no lineales, que trabajen en un conjunto finito de puntos de operación [1], [2] y [19] y dispongan de un histórico de medidas del sistema obtenidas en ausencia de fallo.

A continuación se describe la organización de este trabajo. En la Sección 2 se realiza una descripción del problema a resolver y la solución propuesta. La Sección 3 aborda la forma de generar el residuo para el método de detección propuesto. El cálculo del umbral máximo permitido para el residuo se trata en la Sección 4. El algoritmo de detección se presenta en la Sección 5. Finalmente en la Sección 6 se presenta una aplicación.

2. Descripción del Problema

Considérese un sistema dinámico descrito por la siguiente ecuación

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ y_k &= g(x_k) \end{aligned} \tag{1}$$

donde k es el instante de tiempo, $x_k \in \mathbb{R}^{n_x}$ son las variables de estado, $u_k \in U^{n_u} \subset \mathbb{R}^{n_u}$ son las variables de entrada, $y_k \in Y^{n_y} \subset \mathbb{R}^{n_y}$ son las variables de salida y U, Y son conjuntos compactos. Por simplicidad se supondrá que $n_y = 1$, aunque las técnicas expuestas en esta comunicación se pueden aplicar al caso generalizado $n_y > 1$ considerando cada componente de y_k como una salida independiente. Se asume que el sistema tiene definido un conjunto discreto de puntos de operación.

Hipótesis 1 *Se asumen múltiples puntos de operación para el sistema. Los puntos de operación se denotan mediante y_e^i con $i = 1, ..., m$. Dichos puntos definen objetivos para la variable de salida y_k. Al conjunto de todos los puntos de operación se le denota mediante Y_e, por lo que $Y_e = \{y_e^1, y_e^2, ..., y_e^m\}$. Con objeto de alcanzar los puntos de operación deseados se asume la siguiente ley de control:*

$$u_k = K(y_k, y_{e_k}) \tag{2}$$

donde $y_{e_k} \in Y_e$ es el punto de operación en tiempo k.

También se asume que se dispone de un sistema de monitorización capaz de leer señales de entrada y salida del sistema.

Hipótesis 2 *Se asume la existencia de un sistema de monitorización con la siguiente funcionalidad:*

- *Capacidad para obtener un histórico, denotado $\mathcal{D}$, con variables medidas y puntos de operación objetivo del sistema, funcionando éste en ausencia de fallos:*

$$\mathcal{D} = \{(y_1, u_1, y_{e_1}), ..., (y_N, u_N, y_{e_N})\}. \tag{3}$$

- *Capacidad para proporcionar en cada instante de tiempo $k > N$ los valores de entrada, salida y puntos de operación incluidos en el vector $\gamma_k = [y_{\mu_y,k}^T \; u_{\mu_u,k}^T \; y_{e_p,k}^T]^T$ siendo*

$$y_{\mu_y,k} = \begin{bmatrix} y_{k-\mu_y+1} \\ \vdots \\ y_{k-1} \\ y_k \end{bmatrix} \quad u_{\mu_u,k} = \begin{bmatrix} u_{k-\mu_u+1} \\ \vdots \\ u_{k-1} \\ u_k \end{bmatrix}$$

$$y_{e_p,k} = \begin{bmatrix} y_{e_{k+1}} \\ \vdots \\ y_{e_{k+p-1}} \\ y_{e_{k+p}} \end{bmatrix} \tag{4}$$

El vector $y_{\mu_y,k} \in Y^{\mu_y}$ contiene μ_y valores de la variable de salida; el vector $u_{\mu_u,k} \in U^{n_u \times \mu_u}$ representa μ_u valores de la variable de entrada; y $y_{e_p,k} \in Y^p$ representa p valores objetivo de la variable de salida.

En este trabajo se propone un nuevo sistema de detección de fallos basado en residuo. Dicho residuo será la diferencia en valor absoluto entre la salida del sistema y_{k+p} y la salida de un predictor $\hat{y}_{k+p|k}$. La predicción de la salida se obtiene con la información de entrada-salida del sistema obtenida hasta el instante k y mediante una aproximación local de la dinámica del sistema. Para calcular dicha aproximación se utilizan los datos recopilados en $\mathcal{D}$. Mediante técnicas de error acotado se obtiene un umbral dinámico que proporciona el valor máximo en cada instante de tiempo que el residuo puede alcanzar en ausencia de fallo. Si el valor del residuo supera el límite marcado por el umbral, el detector proporcionará una detección positiva en caso contrario negativa.

3. Generación del Residuo.

En esta sección se trata la forma de obtener la señal residuo del método propuesto. Para calcular el residuo se utiliza un predictor a p-pasos de la salida del sistema (1). El predictor utiliza los datos históricos disponibles en el conjunto $\mathcal{D}$ para aproximar localmente la dinámica del sistema.

Definición 1 *(Predictor a p-pasos). Para cada instante de tiempo k, dado el vector γ_k y las constantes positivas σ y L, la predicción a p-pasos hacia adelante de la salida del sistema (1) se obtiene mediante las expresiones:*

$$\hat{y}_{k+p|k} = [\gamma_k^\top \; 1]\theta^*(\gamma_k) \tag{5}$$

$$\theta^*(\gamma_k) = (A_{R_k}^T W_k^{-1} A_{R_k})^{-1} A_{R_k}^T W_k^{-1} b_{Y_k} \tag{6}$$

siendo las matrices:

$$A_{R_k}^T = \begin{bmatrix} [\gamma_{s_1}^\top \; 1]^T & [\gamma_{s_2}^\top \; 1]^T & ... & [\gamma_{s_{N_S}}^\top \; 1]^T \end{bmatrix}$$

$$b_{L_k} = \begin{bmatrix} \sigma + L||\gamma_k - \gamma_{s_1}|| \\ \sigma + L||\gamma_k - \gamma_{s_2}|| \\ \vdots \\ \sigma + L||\gamma_k - \gamma_{s_{N_S}}|| \end{bmatrix}$$

$$b_{Y_k} = \begin{bmatrix} y_{s_1+p} & y_{s_2+p} & ... & y_{s_{N_S}+p} \end{bmatrix}^T$$

La matriz W_k es una matriz diagonal. Los elementos de dicha diagonal son los elementos del vector b_{L_k} elevados al cuadrado. El conjunto de índices $\{s_1, s_2, ..., s_{N_S}\}$ es función del vector γ_k considerado, esto es $S(\gamma_k) = \{s_1, s_2, ..., s_{N_S+p}\}$, cumpliendo que $max(\mu_y, \mu_u) \leq s_i \leq N - p$ siendo $i = 1, ..., N_S$.

Para facilitar la notación usaremos S_k en lugar de $S(\gamma_k)$. Notar que S_k suele contener los índice que proporcionan los vectores γ_{s_i} con menor distancia a γ_k. Un elemento importante del predictor y del método de detección son las constantes σ y L. Dichas constantes influyen en la capacidad de detección del método propuesto. En la sección 5.1 se presenta un algoritmo para estimar sus valores. Por otro lado, el vector $\theta^*(\gamma_k)$ incluido en la Definición 1 es la solución óptima de un problema de mínimos cuadrados ponderados. El problema de optimización está formulado para obtener el vector θ que minimiza el error de predicción a p-pasos de una aproximación local del comportamiento del sistema. La aproximación local se obtiene utilizando pesos que ponderan la distancia del vector γ_k a los vectores γ_{s_i} obtenidos a partir del conjunto de datos históricos $\mathcal{D}$. Para realizar la ponderación se utilizan las constantes σ y L.

Propiedad 1 *El vector $\theta^*(\gamma_k)$ resuelve el siguiente problema de mínimos cuadrados ponderados que penaliza de forma cuadrática la predicción del error,*

$$\theta^*(\gamma_k) = arg\min_{\theta} \sum_{i=1}^{N_s} \frac{(y_{s_i+p} - [\gamma_{s_i}^T\ 1]\theta)^2}{(\sigma + L||\gamma_{s_i} - \gamma_k||)^2} \quad (7)$$

Demostración :en efecto, si denotamos

$$\begin{aligned} J(\theta) &= \sum_{i=1}^{N_s} \frac{(y_{s_i+p} - [\gamma_{s_i}^T\ 1]\theta)^2}{(\sigma + L||\gamma_{s_i} - \gamma_k||)^2} \\ &= (b_{Y_k} - A_{R_k}\theta)^T W_k^{-1} (b_{Y_k} - A_{R_k}\theta)^T \end{aligned} \quad (8)$$

entonces su derivada con respecto a θ es

$$\frac{dJ(\theta)}{d\theta} = 2A_{R_k}^T W_k^{-1} A_{R_k}\theta - 2A_{R_k}^T W_k^{-1} b_{Y_k}. \quad (9)$$

igualando la derivada a cero es posible obtener

$$\theta^*(\gamma_k) = (A_{R_k}^T W_k^{-1} A_{R_k})^{-1} A_{R_k}^T W_k^{-1} b_{Y_k} \quad (10)$$

■

Llegados a este punto es posible definir la señal residuo como:

Definición 2 *(Residuo). Se Define el residuo en el instante de tiempo $k+p$ como la diferencia en valor absoluto entre las medida del sistema y_{k+p} y la predicción obtenida $\hat{y}_{k+p|k}$, es decir*

$$res(\gamma_k) = |y_{k+p} - \hat{y}_{k+p|k}| \quad (11)$$

La siguiente sección trata la obtención de un umbral máximo que puede alcanzar la señal de residuo cuando el sistema está funcionando en ausencia de fallos.

4. Umbral Dinámico de la Señal Residuo

En esta sección se propone una expresión que limita el valor máximo que puede alcanzar la señal de residuo en ausencia de fallos. Dicha expresión asume ciertas hipótesis basadas en error acotado que se describen a continuación.

Hipótesis 3 *Se asume que para cualquier vector $\gamma_k \in Y^{\mu_y} \times U^{n_u \times \mu_u} \times Y^p$ existe otro vector $\theta_k \in \mathbb{R}^{\mu_y+\mu_u+p+1}$ y dos constantes positivas $\sigma, L \in \mathbb{R}^+$ tal que se cumple:*

$$y_{k+p} = [\gamma_k^T\ 1]\theta_k + e_k \quad (12)$$

$$y_{i+p} = [\gamma_i^T\ 1]\theta_k + e_{k,i} \quad (13)$$

con $i = max(\mu_y, \mu_y), ..., N-p$. Los términos de error e_k y $e_{k,i}$ están acotados mediante las expresiones $|e_k| \leq \sigma$ y $|e_{k,i}| \leq \sigma + L||\gamma_i - \gamma_k||$.

Una cuestión interesante que se puede plantear en este punto es preguntarse si la hipótesis anteriormente expuesta es razonable. A juicio de los autores la respuesta es que sí. Básicamente lo que se está asumiendo en dicha hipótesis es que el comportamiento del sistema (1) en cada instante de tiempo k se puede aproximar mediante un modelo lineal (ver expresión 12). Así mismo, se asume que si utilizamos este mismo modelo para predecir la salida del sistema en otros instantes temporales recogidos en el conjunto de históricos $\mathcal{D}$ (ver expresión 13), el error de predicción crece con el tamaño de la diferencia $||\gamma_i - \gamma_k||$. Los tamaños de los errores permitidos están definidos en base a las constantes σ y L. A continuación se presenta la principal contribución de esta comunicación. Se muestra que si las hipótesis anteriormente expuestas se cumplen, el residuo generado por la expresión (11) tiene un límite para el caso de funcionamiento en ausencia de fallo.

Teorema 1 *Considerando el sistema (1) funcionando en ausencia de fallo y el residuo obtenido mediante la Definición 2 se cumple la siguiente inecuación,*

$$res(\gamma_k) \leq lim_k = b_{L_k}^T |\lambda_k^*| + \sigma \quad (14)$$

donde

$$\lambda_k^* = W_k^{-1} A_{R_k} (A_{R_k}^T W_k^{-1} A_{R_k})^{-1} [\gamma_k^T\ 1]^T \quad (15)$$

Demostración :

En el Apéndice se incluye una demostración completa del Teorema. ■

Una vez expuesto cómo calcular la señal residuo y cómo obtener un umbral máximo para la misma, la siguiente sección describe el algoritmo de detección de fallos que se propone en este trabajo.

5. Algoritmo de Detección de Fallos Propuesto.

En esta sección se presenta el nuevo algoritmo de detección de fallos denotado Algoritmo I. En cada instante de tiempo $k+p$ el algoritmo de detección de fallos se utiliza con el objetivo de determinar la presencia de fallo. Dicho algoritmo obtiene un residuo $res(\gamma_k)$ en el paso 5, calcula en el paso 6 el umbral dinámico y por último chequea en el paso 7, si dicho residuo sobrepasa el umbral obtenido se detecta un fallo.

Algoritmo I.

1. Obtener las matrices y vectores $A^T_{R_k}$, b_{L_k}, W_k y b_{Y_k} usando $\mathcal{D}$, σ y L.
2. Obtener el vector γ_k
3. Calcular $\theta^*(\gamma_k)$ y $\hat{y}_{k+p|k}$ mediante (6) y (5).
4. Leer y_{k+p}.
5. Calcular $res(\gamma_k)$ usando (11).
6. Calcular lim_k con (14).
7. Si $res(\gamma_k) > lim_k$ fallo detectado.

Los parámetros σ y L definen la sensibilidad del método de detección. A continuación se propone un método para estimar sus valores.

5.1. Ajuste de Parámetros

Para poder realizar el calculo del residuo tal y como se explicó en la Sección 3, es necesario proporcionar unos valores para las constantes σ y L. Estos valores se pueden estimar empíricamente mediante el Algoritmo II. El objetivo de este algoritmo es encontrar los valores de σ y L para los cuales el Algoritmo I no detecta fallos en un conjunto de datos $\mathcal{D}_T = \{(y_{j+p}, \gamma_j)\}$ con $j = 1, ..., N_{\mathcal{D}_T}$ y generan umbrales de detección de valor mínimo. El conjunto de datos $\mathcal{D}_T$ se debe haber obtenido a partir de datos de entrada-salida del sistema funcionando en ausencia de fallo. Dicho conjunto debe contener información suficiente sobre la dinámica del sistema y las posibles perturbaciones. Una posibilidad para construir el conjunto $\mathcal{D}_T$ es utilizar los datos recopilados en $\mathcal{D}$

Por tanto el Algoritmo II es un algoritmo iterativo de búsqueda que localiza aquellas parejas de valores de σ y L que no detectan fallo en el conjunto de datos $\mathcal{D}_T$ y devuelve aquella pareja de valores que hace mínima el área encerrada por los umbrales dinámicos obtenidos.

Algoritmo II. Obtención de Parámetros

1. Inicializar $\sigma = \delta_\sigma,\ L = \delta_L$ con $\delta_\sigma > 0,\ \delta_L > 0$
2. do
3. do
4. $L = L + \delta_L$
5. Testear $\mathcal{D}_T$ con Algoritmo I
6. while se detecte fallo
7. almacenar σ, L y suma de umbrales.
8. $L = \delta_L$
9. $\sigma = \sigma + \delta_\sigma$
10. Testear con Algoritmo I
11. while se detecte fallo
12. seleccionar σ, L con menor suma de umbrales.
13. Devolver σ, L.

6. Ejemplo.

El método de detección de fallos que se propone se ha validado simulando un sistema intercambiador de calor (Figura 1). Los flujos y temperaturas de

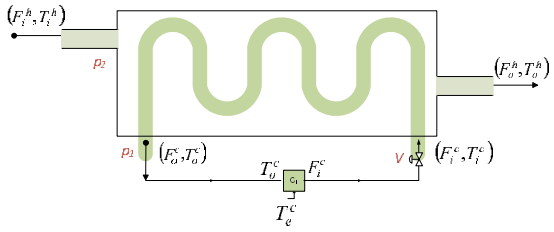


Figura 1: Intercambiador de Calor

los fluidos implicados en la transferencia de calor se denotan como F_i^c, T_i^c,F_o^c, T_o^c, F_i^h, T_i^h y F_o^h, T_o^h. Los superíndices h y c indican el lado caliente y frío y los subíndices i y o representan entradas y salidas, respectivamente.

El sistema dispone de un controlador que actuando sobre el flujo F_i^c intenta llevar la temperatura T_o^c hacia la temperatura de referencia T_e^c. Las magnitudes F_i^h, T_i^h y T_i^c son constantes. Para hacer más realista el escenario del ejemplo se ha añadido una perturbación aleatoria de $\pm 5\,\%$ en el parámetro F_i^h y $\pm 2{,}5\,\%$ en T_i^h y T_i^c.

El sistema se ha simulado en Simulink ® con un tiempo de muestreo de 1 segundo. Se ha utilizado el modelo intercambiador multicelda descrito en [13], con $n = 12$ elementos interconectados. El modelo se basa en las siguientes ecuaciones diferenciales:

$$\frac{dT^h(i)}{dt} = (T^h(i-1) - T^h(i) - \frac{h^h A}{\dot{m}^h c_p^h n}\Delta T^{wh}(i))\frac{\dot{m}^h n}{\rho^h V^h} \quad (16)$$

$$\frac{dT^w(i)}{dt} = (h^h \Delta T^{wh}(i) - h^c \Delta T^{wc}(i))\frac{A}{\rho^w c_p^w V^w} \quad (17)$$

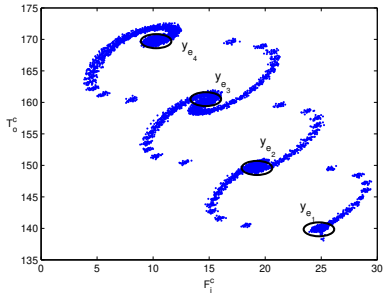


Figura 2: Datos históricos recopilados del sistema

$$\frac{dT^c(n-i+1)}{dt} = (T^c(n-i) - T^c(n-i+1) + \frac{h^c A}{\dot{m}^c c_p^c n}\Delta T^{wc}(n-i+1))\frac{\dot{m}^c n}{\rho^c V^c} \quad (18)$$

donde: $i = 1, ..., n$, $\Delta T^{wc}(i) = T^w(i) - T^c(i)$ y $\Delta T^{wh}(i) = T^h(i) - T^w(n-i+1)$.

Se han omitido los valores de las constantes utilizados por limitaciones de espacio. Se han considerado cuatro puntos de operación distintos definidos como $Y_e = \{y_e^1, y_e^2, y_e^3, y_e^4\} = \{140, 150, 160, 170\}$, donde $y_{e_i} = \{T^c_{e_i}\}$ con $i = 1, .., 4$. La Figura 2 muestra datos de entrada-salida obtenidos del sistema en ausencia de fallos. Además se definen las variables $x_k = [T^c_{o_k}\ ]^T$, $y_k = [T^c_{o_k}]^T$ y $u_k = [F^c_{i_k}]^T$ que formarán el histórico de medidas que viene definido por $\mathcal{D} = \{F^c_{i_j}, T^c_{o_j}, T^c_{e_j}\}$ $j = 1, ..., 17900$. Para la formación del vector γ_k, se ha elegido de forma empírica un horizonte de predicción $p = 5$ y una ventana de datos del pasado definida como $\mu_y = \mu_u = 5$. Con el fin de estimar los parámetros σ y L se aplica el Algoritmo II sobre un conjunto $\mathcal{D}_T \subseteq \mathcal{D}$ obteniéndose los siguientes valores: $\sigma = 0{,}355$ y $L = 0{,}0006$.

Se han considerado cuatro fallos con el objetivo de chequear el método propuesto. El fallo f_{a_1} es un fallo de actuador inducido mediante una zona muerta en la válvula V entre el 5 % y el 15 % de la apertura; el fallo f_{s_1} es un fallo de planta, el cual consiste en una obstrucción de tubería causando la reducción del 20 % en el flujo F^c_o; el fallo f_{s_2} que es un fallo de planta, consiste en una obstrucción de tubería que causa una reducción del 20 % en el flujo F^h_i y el fallo f_{s_3} que es un fallo de planta debido a la suciedad acumulada en el muro que causa una reducción del 20 % en la capacidad calorífica específica del muro c^w_p.

El método desarrollado en este artículo se ha comparado con un método de detección de fallos basado en modelos (denotado MBM) descrito en [12] y otro método basado en técnicas de error acotado publicado en [10] (denotado FDBFSS). En FDBFSS, el algoritmo de detección asume un modelo lineal variante con el tiempo. Utilizando zonotopos y técnicas de error acotado se obtiene un predictor intervar. Definiendo $\breve{\gamma}_k = [F^c_{i_k}\ T^c_{o_k}\ T^c_{e_{k+p}}]$, el algoritmo presentado en FDBFSS considera el siguiente modelo de predicción:

$$\breve{y}_{k+p|k} = \breve{\gamma}_k \breve{\theta}_k + \breve{e}_k$$
$$\breve{\theta}_k = \breve{\theta}_{k-1} + w_k$$

donde k es el tiempo de muestreo, $|\breve{e}_k| <= 0{,}37$ y $|w_k| <= 0{,}0055$. Las cotas de $\breve{e}_k$ y w_k se han calculado siguiendo el procedimiento propuesto en [10].

Por otro lado MBM utiliza un modelo lineal de predicción para obtener una señal residuo. El modelo de predicción se ha calculado mediante técnicas de identificación de parámetros:

$$\breve{y}_{k+p|k} = -0{,}03F^c_{i_k} + 0{,}69T^c_{o_k} + 0{,}31T^c_{e_{k+p}}$$

Para el método MBM se ha definido un umbral estático que coincide con el valor máximo del residuo obtenido en un escenario libre de fallos.

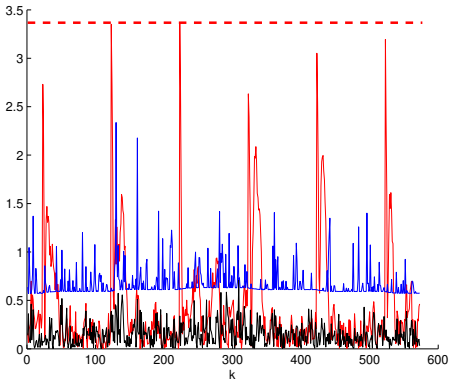


Figura 3: Comparación de residuos y umbrales.

En la Figura 3 puede apreciarse en color rojo el residuo y el umbral obtenidos mediante la aplicación del método MBM a un conjunto de datos libre de fallos. En negro y azul se pueden observar el residuo y el umbral dinámico obtenidos al aplicar el método propuesto en este trabajo al mismo conjunto de datos libre de fallos.

Con el fin de realizar un estudio comparativo, los distintos fallos propuestos se han generado para todos los puntos de operación y distintas transiciones entre ellos. En la Tabla 1 se encuentran marcados con X, los fallos detectados en cada punto de operación por los distintos métodos de detección. Se ha denotado como MDFRT al método desarrollado en este trabajo. De la misma forma en la Tabla 2 se marcan los fallos detectados en las transiciones entre los puntos de operación. Los resultados muestran la mayor capacidad de detección que tiene el método propuesto en este trabajo al aplicarlo a este ejemplo. A juicio de los autores estos resultados se deben la adaptación local y capacidad de representar dinámicas no lineales del método propuesto.

Cuadro 1: Resultados ante fallos en los puntos de operación

Precisión del Método					
		Puntos de Operación			
Fallo	Método	y_{e_1}	y_{e_2}	y_{e_3}	y_{e_4}
f_{a_1}	MDFRT	-	-	-	-
f_{a_1}	FDBFSS	-	-	-	-
f_{a_1}	MBM	-	-	-	-
f_{s_1}	MDFRT	X	X	X	X
f_{s_1}	FDBFSS	-	-	-	-
f_{s_1}	MBM	-	-	-	-
f_{s_2}	MDFRT	X	X	X	X
f_{s_2}	FDBFSS	-	-	-	-
f_{s_2}	MBM	-	-	-	-
f_{s_3}	MDFRT	X	X	X	X
f_{s_3}	FDBFSS	X	X	X	X
f_{s_3}	MBM	X	X	X	X

En la Figura 4 se puede ver la aplicación del Algoritmo I para el fallo f_{s_2} en la transición $y_{e_1} = 140$ a $y_{e_2} = 150$. La línea negra representa el residuo y la línea roja el umbral de detección del fallo. Se ha señalado mediante un circulo el momento en el que el fallo es detectado. Se puede observar que el método propuesto detecta el fallo cuando el residuo, marcado con línea negra, sobrepasa el umbral de detección de fallo, representado por línea roja.

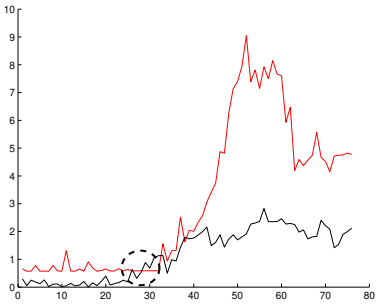


Figura 4: Ejemplo de detección una transición

7. Conclusiones

En este trabajo se ha presentado un nuevo método de detección de fallos basado en datos históricos. El método utiliza un conjunto de datos obtenidos del sistema en ausencia de fallo para generar un residuo y un umbral máximo para este residuo. El residuo se genera asumiendo que en cada instante de tiempo el comportamiento del sistema se puede predecir mediante un modelo lineal local. Dicho modelo local se obtiene consultando el conjunto de datos históricos disponible. La principal aportación de este trabajo es la expresión obtenida aplicando técnicas de error acotado que proporciona un umbral máximo para ese residuo en ausencia de fallo. Dicho umbral es dinámico y por tanto cambia para cada nuevo valor del residuo. El método propuesto se puede aplicar en sistemas no lineales que trabajen en puntos de operación definidos y no necesita asumir consideraciones estadísticas en la señal residuo. La capacidad de detección del método vendrá determinada por la cantidad de información contenida en el conjunto de datos históricos considerado y los parámetros σ y L estimados.

Cuadro 2: Resultados ante fallos en las transiciones entre puntos de operación.

Precisión del Método							
		Transiciones					
Fallos	Método	$y_{e_1} \downarrow y_{e_2}$	$y_{e_2} \downarrow y_{e_3}$	$y_{e_3} \downarrow y_{e_4}$	$y_{e_4} \downarrow y_{e_3}$	$y_{e_3} \downarrow y_{e_2}$	$y_{e_2} \downarrow y_{e_1}$
f_{a_1}	MDFRT	X	X	X	-	-	-
f_{a_1}	FDBFSS	-	-	-	-	-	-
f_{a_1}	MBM	-	-	-	-	-	-
f_{s_1}	MDFRT	X	X	X	X	X	X
f_{s_1}	FDBFSS	-	-	-	-	-	-
f_{s_1}	MBM	X	-	-	-	-	-
f_{s_2}	MDFRT	X	X	X	X	X	X
f_{s_2}	FDBFSS	-	-	-	-	-	-
f_{s_2}	MBM	-	-	-	-	-	-
f_{s_3}	MDFRT	X	X	X	X	X	X
f_{s_3}	FDBFSS	X	X	X	X	X	X
f_{s_3}	MBM	X	X	X	X	X	X

Agradecimientos

Esta investigación ha sido financiada por DPI2012-33309 del MEC de España.

Referencias

[1] M. Basseville and A. B. (Eds). *Detection of Abrupt Changes in Signals and Dynamical Systems.* Lecture Notes in Control and Information Sciences,. Springer-Verlag, Berlin, 1985.

[2] M. Basseville and I. Nikiforov. *Detection of Abrupt Changes - Theory and Application.* Prentice-Hall, Inc., 1993.

[3] J. Blesa, V. Puig, and J. Saludes. Identification for passive robust fault detection using zonotope-based set-membership approaches. *International Journal of Adaptive Control and Signal Processing*, 25(9):788–812, 2011.

[4] J. M. Bravo, T. Alamo, and E. F. Camacho. Bounded error identification of systems

with time-varying parameters. *IEEE Transactions on Automatic Control*, 51-7:1144 – 1150, 2006.

[5] J. M. Bravo, T. Alamo, M. J. Redondo, and E. F. Camacho. An algorithm for bounded-error identification of nonlinear systems based on dc functions. *Automatica*, 44-2:437–444, 2008.

[6] J. Chen and R. J. Patton. *Robust model-based fault diagnosis for dynamic systems.* Kluwer Academic Publishers, 1999.

[7] P. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7(6):403 – 424, 1997.

[8] P. M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica*, 26(3):459 – 474, 1990.

[9] J. Gertler. Fault detection and isolation using parity relations. *Control Engineering Practice*, 5(5):653 – 661, 1997.

[10] A. Ingimundarson, J. M. Bravo, V. Puig, T. Alamo, and P. Guerra. Robust fault detection using zonotope-based set-membership consistency test. *Int. J. Adapt. Control Signal Process*, 23:311–330, 2008.

[11] R. Isermann. Process fault detection based on modeling and estimation methods a survey. *Automatica*, 20(4):387 – 404, 1984.

[12] R. Isermann. *Fault-Diagnosis systems. An Introduction from Fault Detection to Fault Tolerance.* Springer, 2006.

[13] K. W. Mathisen, M. Morari, and S. Skogestad. Dynamic models for heat exchangers and heat exchanger networks. *Computers and Chemical Engineering*, 18, Supplement 1(0):S459 – S463, 1994.

[14] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter. *Bounding Approaches to System Identification.* Plenum Press, New York, 1996.

[15] E. L. Russell, L. H. Chiang, and R. D. Braatz. *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes (Advances in Industrial Control).* Springer-Verlag, 2000.

[16] S. Simani. Identification and fault diagnosis of a simulated model of an industrial gas turbine. *Industrial Informatics, IEEE Transactions on*, 1(3):202 – 216, aug. 2005.

[17] S. Simani, C. Fantuzzi, and R. Patton. *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques.* Advances in Industrial Control. Springer-Verlag, Berlin, 2002.

[18] X.-C. Tang and Y. Li. Monitoring and fault diagnosis using fisher discrimnant analysis. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 2, pages 1100 –1105, aug. 2007.

[19] A. Tartakovsky, I. Nikiforov, and M. Basseville. *Sequential Analysis: Hypothesis Testing and Change-Point Detection.* Chapman zonotopo Hall/CRC Monographs on Statistics zonotopo Applied Probability, Taylor and Francis Group., 2012.

[20] S. Tornil-Sin, C. Ocampo-Martinez, V. Puig, and T. Escobet. Robust fault detection of non-linear systems using set-membership state estimation based on constraint satisfaction. *Engineering Applications of Artificial Intelligence*, 25(1):1 – 10, 2012.

8. Apendice

A continuación se incluye una prueba del Teorema 1. Para comenzar con la demostración es necesario definir el conjunto:

$$\Theta(\gamma_k) = \{\ \theta\ :\ |y_{i+p} - [\gamma_k^T\ 1]\theta| \leq \sigma + L||\gamma_i - \gamma_k||\ ,$$

$$i \in S_k \subseteq \{max(\mu_y, \mu_u), ..., N-p\}\}$$

El conjunto $\Theta(\gamma_k)$ se puede reescribir en notación matricial como

$$\Theta(\gamma_k) = \{\ \theta\ :\ A_{S_k}\theta \leq b_{S_k}\}$$

donde $A_{S_k} = [A_{R_k}^T\ \ -A_{R_k}^T]^T$ y $b_{S_k} = [b_{L_k}^T\ \ b_{L_k}^T]^T + [-b_{Y_k}^T\ \ b_{Y_k}^T]^T$. Una vez definido $\Theta(\gamma_k)$ es necesario definir las cotas $\overline{y}_{k+p}$ y $\underline{y}_{k+p}$ mediante las siguientes expresiones:

$$\begin{aligned} \overline{y}_{k+p} = &\ \max_{\theta} \quad [\gamma_k^T\ 1]\theta \\ &\ s.t. \quad \theta \in \Theta(\gamma_k) \end{aligned}$$

$$\begin{aligned} \underline{y}_{k+p} = &\ \min_{\theta} \quad [\gamma_k^T\ 1]\theta \\ &\ s.t. \quad \theta \in \Theta(\gamma_k) \end{aligned}$$

Teniendo en cuenta la Hipótesis 3, concretamente la expresión (12) se cumplen las siguientes desigualdades:

$$[\gamma_k^T\ 1]\theta_k - \sigma \leq y_{k+p} \leq [\gamma_k^T\ 1]\theta_k + \sigma$$

Es claro que la expresión (13) implica que $\theta_k \in \Theta(\gamma_k)$, por lo que se tiene que

$$\underline{y}_{k+p} - \sigma \leq y_{k+p} \leq \overline{y}_{k+p} + \sigma$$

Dado el vector $\lambda_k^* \in \mathbb{R}^{N_s}$ obtenido de (15), es posible definir los vectores $\underline{\mu}, \overline{\mu} \in \mathbb{R}^{2N_s}$. Aclarar que no se ha incluido la dependencia respecto a k en los vectores $\underline{\mu}$ y $\overline{\mu}$ para simplificar la notación. Los componentes de $\underline{\mu} = [\underline{\mu}_1, ..., \underline{\mu}_{2N_S}]^T$ y $\overline{\mu} = [\overline{\mu}_1, ..., \overline{\mu}_{2N_S}]^T$ se definen mediante:

$$\overline{\mu}_i = \begin{cases} \lambda^*_{k_i} & Si \;\; \lambda^*_{k_i} \geq 0 \\ 0 & Si \;\; \lambda^*_{k_i} < 0 \end{cases} \tag{19}$$

$$\overline{\mu}_{i+N_s} = \begin{cases} 0 & Si \;\; \lambda^*_{k_i} \geq 0 \\ -\lambda^*_{k_i} & Si \;\; \lambda^*_{k_i} < 0 \end{cases} \tag{20}$$

$$\underline{\mu_i} = \overline{\mu}_{i+N_s} \tag{21}$$

$$\underline{\mu}_{i+N_s} = \overline{\mu}_i \tag{22}$$

para $i = 1, ..., N_s$. Es importante resaltar que los componentes $\overline{\mu}_i$ y $\underline{\mu}_i$ tienen valores positivos, es decir, $\overline{\mu}_i, \underline{\mu}_i, \geq 0$ para $i = 1, ..., 2N_s$.

Denotando

$$\underline{y}_{k+p}(\lambda_k^*) = -b_{L_k}^T |\lambda_k^*| + b_{Y_k}^T \lambda_k^* \tag{23}$$

$$\overline{y}_{k+p}(\lambda_k^*) = b_{L_k}^T |\lambda_k^*| + b_{Y_k}^T \lambda_k^* \tag{24}$$

se puede inferir que

$$\underline{y}_{k+p}(\lambda_k^*) = \underline{\mu}^T b_{S_k}$$

$$\overline{y}_{k+p}(\lambda_k^*) = \overline{\mu}^T b_{S_k}$$

Aplicando la expresión (15) se tiene la siguiente igualdad $A_{R_k}^T \lambda_k^* = [\gamma_k^T \; 1]^T$ de lo que se deduce que las igualdades $A_{S_k} \overline{\mu} = [\gamma_k^T \; 1]^T$ y $A_{S_k} \underline{\mu} = -[\gamma_k^T \; 1]^T$ también se cumplen.

Teniendo en cuenta todo lo expuesto anteriormente, la siguiente sucesión de igualdades y desigualdades se cumple:

$$\begin{aligned} \overline{y}_{k+p}(\lambda_k^*) = \; & \overline{\mu}^T b_{S_k} \\ = \; & \max_{\theta} \; ([\gamma_k^T \; 1]^T - A_{S_k}^T \overline{\mu})^T \theta + \overline{\mu}^T b_{S_k} \\ = \; & \max_{\theta} \; [\gamma_k^T \; 1]\theta + \overline{\mu}^T (-A_{S_k}\theta + b_{S_k}) \\ \geq \; & \max_{\theta} \; [\gamma_k^T \; 1]\theta \\ & s.t. \;\; A_{S_k}\theta \leq b_{S_k} \\ = \; & \overline{y}_{k+p} \end{aligned} \tag{25}$$

$$\begin{aligned} \underline{y}_{k+p}(\lambda_k^*) = \; & -\underline{\mu}^T b_{S_k} \\ = \; & \min_{\theta} \; ([\gamma_k^T \; 1]^T + A_{S_k}^T \overline{\mu})^T \theta - \overline{\mu}^T b_{S_k} \\ = \; & \min_{\theta} \; [\gamma_k^T \; 1]\theta + \overline{\mu}^T (A_{S_k}\theta - b_{S_k}) \\ \leq \; & \min_{\theta} \; [\gamma_k^T \; 1]\theta \\ & s.t. \;\; A_{S_k}\theta \leq b_{S_k} \\ = \; & \underline{y}_{k+p} \end{aligned} \tag{26}$$

De lo anterior se deduce que las siguientes desigualdades se cumplen:

$$\underline{y}_{k+p}(\lambda_k^*) - \sigma \leq \underline{y}_{k+p} - \sigma \leq y_{k+p}$$

$$y_{k+p} \leq \overline{y}_{k+p} + \sigma \leq \overline{y}_{k+p}(\lambda_k^*) + \sigma$$

Utilizando las expresiones (5) y (6) de la Definición 1 y las expresiones (15), (23) y (24) se obtiene que

$$\hat{y}_{k+p|k} = [\gamma_k^T \; 1]\theta_k^*(\gamma_k) = b_{Y_k}^T \lambda_k^* = \frac{1}{2}(\overline{y}_{k+p}(\lambda_k^*) + \underline{y}_{k+p}(\lambda_k^*))$$

Finalmente, recordando la definición de la señal residuo $res(\gamma_k) = |y_{k+p} - \hat{y}_{k+p|k}|$ y teniendo en cuenta las dos últimas expresiones, se puede inferir que:

$$res(\gamma_k) \leq \frac{\overline{y}_{k+p}(\lambda_k^*) + \sigma - (\underline{y}_{k+p}(\lambda_k^*) - \sigma)}{2} = b_{L_k}^T |\lambda_k^*| + \sigma = lim_k$$

con lo que queda demostrado el teorema. ■

En la Figura 5 se muestra una interpretación de la señales $res(\gamma_k)$ y lim_k.

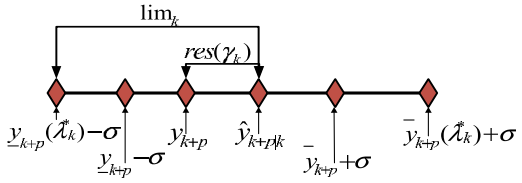


Figura 5: Interpretación de residuo y umbral

Diseño Óptimo Multiobjetivo de PIDs para el Control de Temperatura en Cavidades Microondas

S. Garcia-Nieto, G. Reynoso-Meza, F. Peñaranda-Foix, A. Borrell
sergarro@isa.upv.es, gilreyme@upv.es, fpenaran@dcom.upv.es, aborrell@upvnet.upv.es
Instituto Universitario Ai2, Instituto ITACA, Instituto Tecnológico ITM
Universitat Politècnica de València

Resumen

La sinterización de microondas ha surgido en los últimos años como un método no-convencional para la sinterización de materiales. Dicho método presenta ventajas significativas en comparación con los procedimientos convencionales desde un punto de vista de eficiencia energética y mejora en la calidad del producto. En particular, este novedoso proceso de sinterización es capaz de lograr mayores densidades (hasta un 99 %), durezas (18 GPa) y resistencia a la fractura (6,3 $MPa \cdot m^{1/2}$). Por ello, el diseño de sistemas de microondas, incluidos los sistemas de instrumentación, monitorización y control, se han convertido en un novedoso campo de aplicación e investigación. El presente artículo describe la implementación del sistema de control y monitorización para una cavidad de microondas prototipo desarrollada en la Universidad Politécnica de Valencia (UPV), para la sinterización del nanocompuesto de alúmina-zirconia. La cavidad de microondas multimodo ensayada trabaja a una frecuencia de 2,45 GHz y es capaz de trabajar en un rango de temperaturas de 1200^oC a 1400^oC.

Palabras clave: Cavidades microondas, control adaptativo, nanomateriales, diseño multiobjetivo, control PID.

1. Introducción

La aplicación de energía mediante microondas para el procesamiento de diversos materiales tales como cerámicas, metales y materiales compuestos ofrece diversas ventajas sobre los métodos convencionales de calentamiento. En particular, es posible alcanzar microestructuras únicas cuyas propiedades, tales como el rendimiento, ahorro de energía, coste de fabricación o la síntesis de nuevos materiales [16], se ven todas ellas mejoradas.

La diferencia fundamental entre el proceso convencional de sinterización y el basado en microondas radica en el modo de suministrar la energía térmica, mientras que en los métodos clásicos dicha energía es suministrada a la superficie del material mediante calentamiento radiante, convección, conducción o, incluso, mediante métodos no convencionales como la *Spark Plasma Sintering* (SPS), que emplea la aplicación de pulsos discontinuos de corriente continua, los métodos basados en microondas transfieren la energía térmica mediante la explotación del efecto de resonancia de los materiales. Esta propiedad permite que se *sintonicen* con la emisión de la señal de microondas y se establezca una rápida absorción de la energía contenida en la señal.

En los materiales cerámicos, cuando son sinterizados mediante técnicas tradicionales, las altas temperaturas que se aplican para densificar totalmente las materias primas, provocan como resultado estructuras cristalinas de gran tamaño, debido al efecto denominado *maduración de Ostwald*. Este hecho, dificulta la capacidad de obtener materiales de alta densidad con estructuras cristalinas nanométricas y con tamaños sub-micrométricos [2]. Esta problemática puede ser resuelta de manera eficaz mediante la sinterización por microondas (MW), ya que se comprueba que la utilización de este método permite la reducción de la estructura cristalina de manera radical, así como la producción de una microestructura homogénea. La radiación de microondas para la sinterización de componentes cerámicos ha sido recientemente propuesta en la literatura científica, tal y como se apunta en los trabajos [3, 5]. Asimismo, trabajos recientes han demostrado el rendimiento de los materiales sinterizados mediante cavidades de microondas de rectangular [5] o cilíndricas [7], las cuales proporcionan buenas propiedades mecánicas [15, 5].

Este artículo presenta el diseño del sistema de control y monitorización para un prototipo de cavidad rectangular desarrollado en el Instituto Universitario ITACA de la Universidad Politécnica de Valencia. El objetivo es el diseño de un sistema de control capaz de seguir un perfil de temperatura complejo para la obtención de nano-compuestos de altas prestaciones de tipo $Al_2O_3 - ZrO_2$.

El artículo se encuentra dividido en siete apartados que tratan de resumir el trabajo realizado. Los apartados 2, 3 y 4 describen el proceso de

sinterización mediante cavidades de microondas a través de una breve descripción de los prototipos y materiales empleados. Posteriormente, el apartado 5 describe de manera detallada el proceso de diseño propuesto que permite obtener un conjunto de PIDs óptimos mediante criterios multiobjetivo. Por último, los apartados 6 y 7 muestran los resultados obtenidos al aplicar, sobre el proceso real, el controlador PID óptimo implementado.

2. Prototipo de Cavidad de Microondas

A continuación se muestra la figura 1 que describe de manera sintetizada el prototipo de cavidad de microondas desarrollado por el ITACA. Este prototipo es capaz de llevar a cabo la sinterización de materiales cerámicos con sustrato a altas temperaturas.

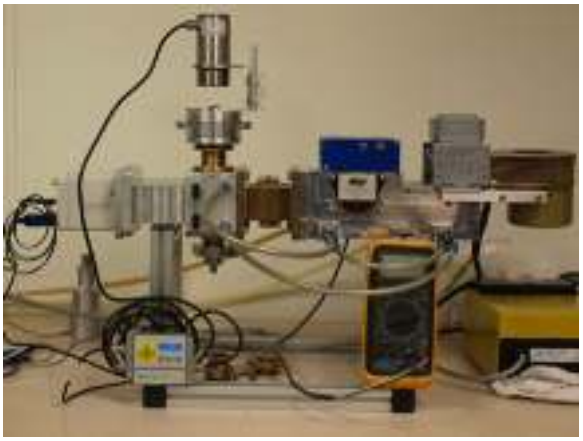


Figura 1: Cavidad de microondas para sinterización.

La figura 1 muestra de derecha a izquierda los siguientes elementos:

- Magnetrón de $1Kw$ de potencia, el cual incluye el sistema de alimentación eléctrica, la bomba de refrigeración y el sistema de acoplamiento para medir la potencia reflejada.
- Cavidad TE_{101}, la cual consiste en una cavidad rectangular para excitar el modo TE_{101}, sobre la base de una guía de onda WR340 y excitado a través de un iris circular para maximizar el componente magnético H_x del modo TE_{101}.
- Sistema motorizado, el cual permite modificar la longitud de la cavidad para permitir la sintonización del material.

El procedimiento de utilización del prototipo comienza con la preparación de la muestra a sinterizar, la cual se introduce en la cavidad a través de un orificio de inserción situado justo en el centro de la cavidad y en la parte superior de la misma. El orificio de inserción tiene un diámetro de $3cm$, mientras que la muestra presenta un diámetro máximo de $1cm$. Esto garantiza que la muestra se encuentre en la zona de sintonización y que, al mismo tiempo, no haya fugas de energía de microondas a través de ella en el modo TE_{11}.

En el centro de la cavidad el modo de excitación tiene máxima componente E_y, un comportamiento de $sin(\cdot)^2$ en la componente axial y horizontal, mientras que no existe ninguna variación en la componente vertical. Esta configuración garantiza la homogeneidad del campo sobre la muestra. La fijación de la muestra en el centro de la cavidad se consigue mediante la utilización de un tubo de cuarzo que es transparente a la energía de microondas.

La monitorización de la temperatura de la muestra se realiza mediante un pirómetro que se encuentra en la parte superior, tal y como se observa en la figura 1. Esto permite que sea posible la introducción de un sistema de monitorización autónomo y, al mismo tiempo, la posibilidad de diseñar un sistema de control que permita automatizar todo el proceso de sinterización.

La figura 2 muestra un esquema simplificado del funcionamiento de la cavidad. Se puede observar la posición central de la muestra dentro de la cavidad, así como el sistema motorizado que permite modificar la longitud de la cavidad con el objetivo de alcanzar el efecto de resonancia sobre el material, permitiendo la máxima absorción de energía.

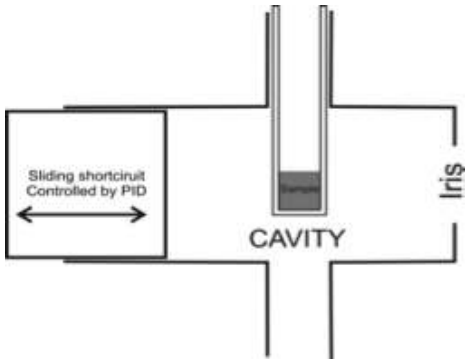


Figura 2: Esquema simplificado de la cavidad.

3. Sistema de control y monitorización

Debido al complejo comportamiento dinámico del proceso de sinterización, resulta imprescindible el diseño de un sistema de control y monitorización que permita controlar de manera precisa el proceso de calentamiento de los materiales. Por ello,

se ha diseñado un nuevo sistema de control para el prototipo de cavidad de microondas TE_{101}. El objetivo principal de este nuevo subsistema es gestionar y optimizar el proceso de sinterización a lo largo de las distintas etapas de calentamiento-enfriamiento que se suceden a en el proceso de transformación del material.

Las principales variables físicas que intervienen en el proceso de sinterización, tales como la temperatura y la energía de microondas, muestran comportamientos dinámicos temporales de carácter continuo y discreto. Por dicha razón, es necesario la utilización de una estructura de control híbrido que permita manejar estas dos realidades [6]. Las estructuras de control híbridas permiten manejar tanto los lazos de control de carácter continuo, en este caso particular mediante el diseño de algoritmos PID, como el comportamiento de las variables discretas que serán gobernadas mediante máquinas de estados secuenciales. La figura 3 muestra de manera esquemática el diagrama funcional del sistema híbrido implementado.

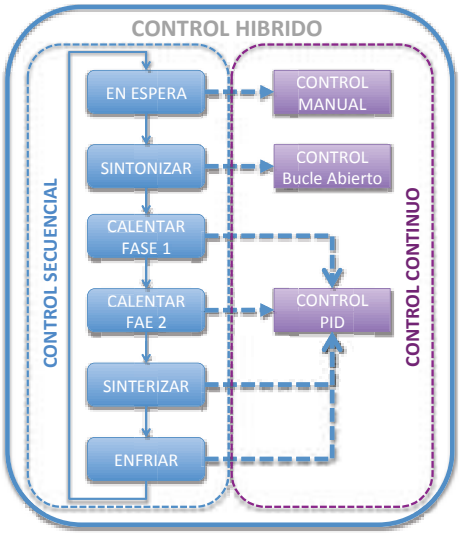


Figura 3: Sistema de control híbrido para la gestión de la sinterización.

La figura 3 muestra el diagrama que describe la relación entre los seis estados de la máquina secuencial discreta y, los lazos de control continuo PID. El ajuste de este sistema de control es una tarea crítica y depende en gran medida de los algoritmos continuos involucrados en el sistema. En este caso, la principal estructura continua que debe ser ajustado es el bien conocido algoritmo PID ISA [1].

$$PID(s) = K_R(1 + sT_D + \frac{1}{sT_I}) \qquad (1)$$

$$PID(t) = K_R\left(e(t) + \frac{de(t)}{dt}T_D + \frac{1}{T_I}\int_{\tau=0}^{t} e(\tau)d\tau\right) \qquad (2)$$

Donde,

	Descripción
s	Operador de Laplace
$e(t)$	Error, calculado como la diferencia entre la referencia deseada y la variable medida
K_R	Término proporcional, ponderación del error *presente* en la acción de control.
T_I	Término integral, ponderación de la acumulación de errores *pasados* en la acción de control.
T_D	Término derivativo, ponderación de la *predicción* del error en la acción de control.

Los parámetros descritos en la tabla anterior (K_R, T_I y T_D) serán ajustado mediante técnicas de diseño basadas en optimización multiobjetivo, tal y como se describe en el apartado 5.

4. Materiales testados en la sinterización

Los materiales testados en la preparación de nanocompuestos mediante sinterización por microondas han sido la alúmina AKP-53, proporcionada por *Sumitomo Co., Japan* (pureza 99,99 %, diámetro medio de partícula de $200nm$), y el óxido de circonio nanométrico (pureza 99,99 %, monoclínico, las partículas primarias de tamaño entre 60 -$100nm$), proporcionado por *Nanoestructurados Materials, Inc.* El procedimiento de dispersión del circonio en el Al_2O_3 ha consistido en la preparación de una suspensión al 15 vol. % de ZrO_2 mediante un proceso de molido estándar [8]. Asimismo, se prepararon muestras cilíndricas de 10mm de diámetro y unos 5mm de altura mediante prensado isostático a $200MPa$.

Las muestras son sinterizadas en la cavidad prototipo mediante la sinterizaron por microondas a 1200^oC, 1300^oC y 1400^oC usando una pendiente de calentamiento de $100frac^oCmin$, estableciendo un tiempo de 60 segundas a máxima temperatura. Dicha temperatura, tal y como ya se ha comentado, es medida mediante un termómetro de radiación infrarroja (Optris CT-laser LT, 8-14 um), que se centra en la muestra de ensayo a través de la pequeña abertura circular en la pared de la cavidad.

5. Ajuste del controlador PID

Para el ajuste del controlador PID se ha optado por una proceso de diseño basada en la optimización multi-objetivo [13] (MOOD por sus siglas en inglés). Se emplea dicha estrategia dada la complejidad del sistema y la flexibilidad que se obtiene con el proceso MOOD para seleccionar alternativas de diseño con el grado de prestaciones deseado por el diseñador. Como consecuencia de ello, se requiere un mayor acercamiento (y tiempo) del diseñador en la fase de selección de la solución final. Este proceso es explicado a continuación.

5.1. Diseño mediante optimización multi-objetivo

La optimización multi-objetivo parte de que es común determinar un conjunto de objetivos, requerimientos y especificaciones para un problema de diseño. Generalmente, los objetivos de diseño especificados entran en conflicto entre sí, *i.e.* la mejora en el rendimiento de uno de ellos es a expensas de empeorar algún otro. Por tanto, se busca un conjunto de soluciones Pareto-óptimas [10], que se diferencian entre ellas por el grado de compromiso entre los objetivos de diseño.

El proceso de diseño mediante la optimización multi-objetivo consta (al menos) de tres etapas primordiales [13]: 1) definición del MOP, 2) la etapa MOO, 3) la selección multi-criterio.

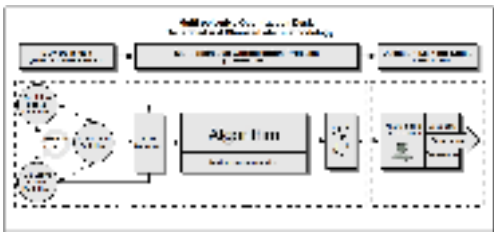


Figura 4: Proceso de diseño mediante optimización multi-objetivo

En el planteamiento del problema multi-objetivo se determinan las variables de diseño (espacio de búsqueda), los objetivos de diseño (medición del rendimiento de una solución) y las restricciones (si las hay). Al plantear un problema de optimización multi-objetivo, no se obtiene (probablemente) una única solución, sino un conjunto de soluciones potenciales, describiendo diferentes grados de compromiso entre los objetivos de diseño. Dicho conjunto de soluciones conforma lo que se conoce como conjunto de Pareto $\boldsymbol{\Theta}_P$, y su imagen en el espacio de objetivos, frente de Pareto $\boldsymbol{J}_P$.

Existe una variedad de alternativas en lo referente a algoritmos para el proceso de búsqueda. Un resumen sobre estrategias utilizadas para tal propósito pueden encontrarse en [9]. Recientemente, se han empleado algoritmos evolutivos multi-objetivo (MOEAs) como herramientas para generar el conjunto de soluciones que describen al frente de Pareto.

La etapa MCDM, consiste en el análisis del conjunto de soluciones y la selección de una alternativa de diseño (la más preferente desde el punto de vista del diseñador). Lo anterior no es trivial, ya que el análisis de un frente de Pareto de 4 o más objetivos lleva a un análisis multi-dimensional de soluciones. Mientras que la visualización podría considerarse resuelta en 2-3 dimensiones para apreciar el intercambio de prestaciones, esto no se logra visualizar adecuadamente en mayores dimensiones. Por ello es crucial contar con mecanismos o herramientas para analizar datos multi-dimensionales y de ser posible, visualizarlos.

5.2. Ajuste del controlador mediante optimización multi-objetivo

El anterior proceso puede ser empleado para el ajuste de controladores PID. A continuación, se describe cada una de las etapas seguidas para ello.

5.2.1. Planteamiento del Problema

Se plantea el siguiente problema de optimización a partir de un modelo nominal del sistema:

$$\min_{\boldsymbol{x}=K_R,T_I,T_D} \boldsymbol{J}(\boldsymbol{x}) = [-K_I(\boldsymbol{x}), M_s(\boldsymbol{x})] \qquad (3)$$

sujeto a:

$$1{,}2 \leq M_s(\boldsymbol{x}) \leq 1{,}8 \qquad (4)$$

$$K_R \in [0,2], T_I \in [0,100], T_D \in [0,100] \qquad (5)$$

Donde $K_I = K_R/T_I$ es la ganancia integral y M_s el valor máximo de la función de sensibilidad del lazo cerrado. Así mismo, se impone como restricción la estabilidad del lazo ante el modelo nominal.

Para la identificación de un modelo nominal, se ha identificado un conjunto de plantas de primer orden con retardo en el rango $500^oC - 1200^oC$. En dicho conjunto de modelos se aprecia la no-linealidad del sistema al registrar diferencias significativas en los parámetros de la planta. Como planta nominal se elige el modelo identificado de los 850^oC : $\frac{272}{17{,}25s+1}e^{-3{,}85s}$.

5.2.2. Proceso de búsqueda-optimización

Cualquier optimizador puede ser empleado para el proceso de búsqueda. En este caso, se ha optado

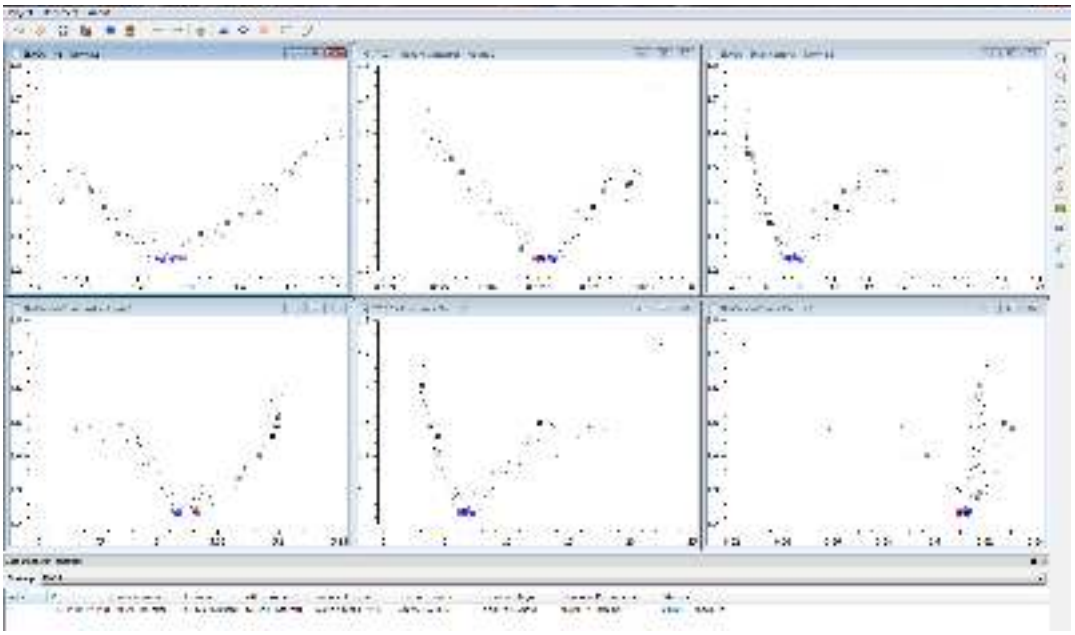


Figura 5: Multi-objective optimization design

por un algoritmo de optimización multi-objetivo evolutivo. Se ha empleado el algoritmo sp-MODE[1] [11] que ha sido empleado con anterioridad en tareas de ajuste de controladores PI/PID [12, 14]. Un total de 84 alternativas de diseño (controladores) han sido calculados, empleando los parámetros por defecto del optimizador.

5.2.3. Proceso de selección

Para visualizar las soluciones y seleccionar la más conveniente para la aplicación de sinterizado de materiales, se han utilizado Level Diagrams[4]. Con objeto de interactuar con la visualización, se emplea el software LD-Tool[2].

Se han incluido nuevos indicadores en la etapa de selección. Dichos indicadores, aunque no han sido considerados en la parte de optimización, se ha determinado que es conveniente valorarlos para elegir una solución final. Sea VT la variación total de la acción de control y Emax el error máximo de seguimiento en un lazo de control. Se definen entonces los indicadores:

$$I_1(\boldsymbol{x}) = Emax|_{g(t)_{Nominal}} \quad (6)$$

$$I_2(\boldsymbol{x}) = TV|_{g(t)_{Nominal}} \quad (7)$$

$$I_3(\boldsymbol{x}) = VAR\left(Emax|_{g'(t)\in SET}\right) \quad (8)$$

$$I_4(\boldsymbol{x}) = VAR\left(TV|_{g'(t)\in SET}\right) \quad (9)$$

[1]Disponible en http://www.mathworks.com/matlabcentral/fileexchange/39215

[2]Disponible en http://cpoh.upv.es/

Donde I_1, I_2 han sido calculados para cada controlador del frente aproximado con la planta nominal. Los indicadores I_3, I_4 han sido calculados con un conjunto de plantas (20) de primer orden con retardo con parámetros aleatorios; dichos parámetros son calculados según los límites registrados en la fase de identificación para la ganancia del sistema, su constante de tiempo y su retardo.

Dichos indicadores no han sido considerados en la etapa de optimización debido al alto coste computacional que conlleva realizar multiples simulaciones para medir el rendimiento de las alternativas de diseño. Por ello se ha optado por el enfoque tradicional de optimizar índices de coste teóricos para medir el rendimiento al seguimiento de trayectoria y robustez del sistema e incluir a posteriori los demás indicadores.

La visualización correspondiente al conjunto de soluciones calculado con anterioridad se observa en la figura 5. Se aprecia un conjunto de controladores PID (mostrados en la figura) como candidatos, siendo que todos tienen norma 2 similar. De entre ellos, se elige el de mayor robustez para ser implementado en el sistema.

6. Resultados

En la figura 6 se muestra el comportamiento logrado por el controlador seleccionado $x = [0{,}0100, 6{,}5887, 2{,}0669]$. Tras el transitorio (alrededor de t=440 segundos), el controlador es capaz de seguir la referencia de temperatura hasta los 1200 ^oC, con un error máximo de 10.9 ^oC y un error medio de 6.69 $^oC/segundo$.

Los datos mostrados en la figura 6 han sido obtenidos mediante experimentación real sobre el prototipo de cavidad TE_{101}. Tal y como se puede apreciar, el sistema de control es capaz de seguir el perfil de temperatura fijado, permitiendo una correcta sinterización del material empleado. Sin embargo, tal y como se puede apreciar en el intervalo de 1100 a 1300 segundos (zona ampliada en la figura 6), el comportamiento dinámico del sistema varía drásticamente en algunas zonas, lo que obliga al controlador a adaptar la señal de control para conseguir seguir la referencia establecida. Este comportamiento dinámico inesperado es debido al cambio abrupto de la estructura molecular de los materiales a lo largo del proceso de sinterización, lo provoca una brusca variación en el comportamiento dinámico. Por tanto, es necesario diseñar sistema de control robusto que garantice un funcionamiento aceptable ante cambios bruscos del comportamiento lo s cuales resultan difíciles de modelar y/o predecir.

7. Conclusiones

El trabajo que se presenta describe de manera resumida una alternativa para el diseño óptimo de controladores robustos destinados a la gestión de la temperatura en procesos de sinterización mediante microondas. La principal dificultad en el diseño del sistema de control radica en la novedad del proceso de sinterización mediante microondas. Por ello, resulta complejo la definición de modelos dinámicos que describan con precisión el comportamiento de la temperatura a la largo del proceso de sinterización. Asimismo, el propio prototipo de cavidad microondas, el TE_{101}, es un dispositivo en fase de test lo que añade un elemento extra de complejidad dentro del diseño del sistema de monitorización y control.

A pesar de la incertidumbre existente en el comportamiento dinámico de los materiales a lo largo del proceso de sinterización, el procedimiento de diseño que se presenta permite establecer el grado de robustez del controlador PID de manera previa. Por ello, tal y como se muestra en el apartado 6, el sistema de control obtenido presenta un comportamiento razonable dentro de los márgenes que se definan. Asimismo, dicho sistema de control permite mantener bajo control todo el sistema, estando siempre muy próximo a los perfiles de referencia de temperatura establecidos, incluso ante cambios bruscos en el comportamiento dinámico, debido a los rápidos cambios estructurales que sufren los materiales a lo largo del proceso de sinterización.

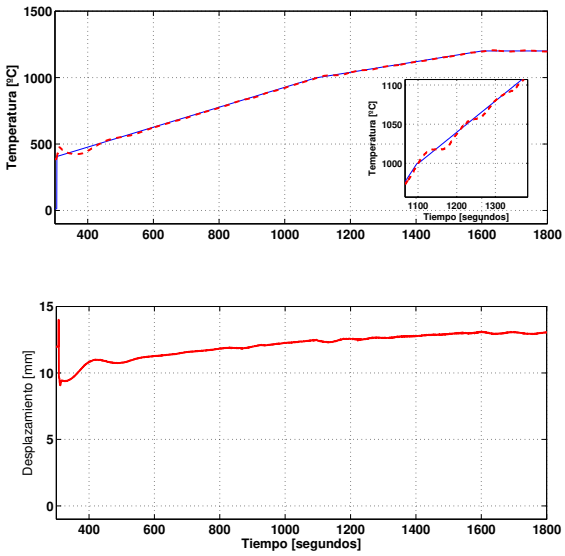


Figura 6: Multi-objective optimization design

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Universitat Politècnica de València mediante los proyectos de investigación PAID/2012/227, SP20120677, SP20120398 y por el Ministerio de Educación mediante el proyecto TEC2012-37532-C02-01.

Referencias

[1] Ang, K. H., Chong, G. C. Y., and Li, Y., (2005) PID control system analysis, design and technology. IEEE. Trans Control Systems Tech., vol. 13, no 4, pp. 559-576.

[2] Anselmi-Tamburini, U., Garay, J. E., and Munir, Z. A., (2006) Scripta Mater., vol. 54, pp. 823-828.

[3] Binner, J., Annapoorani, K., Paul, A., Santacruz, I., and Vaidhyanathan, B., (2008) Dense nanostructured zirconia by two stage conventional/hybrid microwave sintering, J. Eur. Ceram. Soc., vol. 28, pp. 973-977.

[4] Blasco, X., Herrero, J., Sanchis, J., Martínez, M., 2008. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. Information Sciences 178 (20), 3908 – 3924.

[5] Borrell, A., Salvador, M. D., Rayón, E., and Peñaranda-Foix, F. L., (2012) Improvement of microstructural properties of 3Y-TZP materials by conventional and non-conventional

sintering techniques. Ceram. Inter., vol. 38, pp. 39-43.

[6] Goebel, R., Sanfelice, G. R., and Teel, R. A., (2009) Hybrid Dynamical Systems. IEEE. Control Systems Magazine, vol. 29, no. 2, pp. 28-93.

[7] Guyon, A., Charmond, S., Bouvard, D., Carry, C. P., and Chaix, J. M., (2011) Microwave Sintering of Nanometric Ceramic Powders in a Single Mode Resonant Cavity. Proceedings of the 13th Int. Conference on Microwave and RF Heating, Toulouse, pp. 237-240.

[8] Lee, P. Y., Suematsu, H., Jiang, W., Yatsui, K., and Niihara, K., (2006) Synthesis of Al2O3-ZrO2 Nanocomposite Powders by Pulsed Wire Discharge. IEEE Trans. Plasma Sci., vol. 34, pp. 1190-1194.

[9] Marler, R., Arora, J., 2004. Survey of multi-objective optimization methods for engineering. Structural and multidisciplinary optimization (26), 369 – 395.

[10] Miettinen, K. M., 1998. Nonlinear multiobjective optimization. Kluwer Academic Publishers.

[11] Reynoso-Meza, G., Sanchis, J., Blasco, X., Martínez, M., 2010. Multiobjective design of continuous controllers using differential evolution and spherical pruning. In: Chio, C. D. et Al. (Eds.), Applications of Evolutionary Computation, Part I. Vol. LNCS 6024. Springer-Verlag, pp. 532–541.

[12] Reynoso-Meza, G., Sanchis, J., Blasco, X., Herrero, J. M., 2012. Multiobjective evolutionary algortihms for multivariable PI controller tuning. Expert Systems with Applications 39, 7895 – 7907.

[13] Reynoso-Meza, G., Blasco, X., Sanchis, J., March 2012. Optimización evolutiva multiobjetivo y selección multi-criterio para la ingeniería de control. In: X Simposio CEA de Ingeniería de Control.

[14] Reynoso-Meza, G., García-Nieto, S., Sanchis, J., Blasco, X., 2013. Controller tuning using multiobjective optimization algorithms: a global tuning framework. IEEE Transactions on Control Systems Technology 21 (2), 445–458.

[15] Vleugels, J., Dreusen, J. V., Van Roey, O., Vand der Viest, O., and Luypaert, P., (2011) Hybrid Microwave Sintering of Titanium Biomedical Implants. Proceedings of the 13th Int. Conference on Microwave and RF Heating, Toulouse, pp. 305-308.

[16] Sutton, W. H. (1989) Microwave Processing of Ceramic Materials, Am. Ceram. Bull., vol, 68, no. 2, pp. 376-386.

Diseño e implementación en UAVs de un sistema de control de formación de vuelo basado en SMRC.

Ignacio Ribelles, igrifer@etsii.upv.es
Escuela Técnica Superior de Ingenieros Industriales, Universitat Politècnica de València.
Alejandro Vignoni, Sergio García-Nieto, Jesús Picó, {alvig2, sergarro, jpico}@upv.es
Institut d'Automàtica i Informàtica Industrial, Universitat Politècnica de València,
Camino de Vera s/n, 46022 Valencia, España.

Resumen

Trabajando con vehículos aéreos no tripulados (UAVs), surgen entre otros, dos problemas de interés en la gestión de trayectorias, primero conseguir una referencia de trayectoria realizable por el UAV y por otra parte la coordinación de varios UAVs a partir de las referencias de sus trayectorias.

En particular, este trabajo se centra en la implementación en Ar.Drones de algoritmos de control por modos deslizantes, y en particular los de acondicionamiento de referencia por modos deslizantes (SMRC), para la gestión y generación de trayectorias para formaciones de vuelo. Una de las ventajas que presentan estos algoritmos son la propiedades de robustez a perturbaciones externas y a incertidumbre en los parámetros del sistema, que heredan del control por modo deslizante. La propuesta se valida por simulación, y se encuentra a la fecha en fase de implementación física en los Ar.Drones.

Palabras clave: UAV, control de formación, generación de trayectorias, modos deslizantes, acondicionamiento de referencia.

1. Introducción

La gestión de trayectorias es uno de los principales problemas que surgen a la hora de conseguir un sistema autónomo de vuelo [9, 2].

Existen dos estrategias de control básicas con las que se pueden resolver este problema, centralizada y descentralizada [1].

La centralizada consta de un nodo principal donde manda a todos los UAV las diferentes trayectorias que tiene que ir siguiendo cada UAV con la información que va recibiendo de los mismos.

La otra estrategia es la descentralizada, se le manda a uno o todos los drones la misma referencia y todos ellos llegan a un consenso y deciden como modificar sus trayectorias para realizar la formación deseada. Es la más usada en formación de vuelo de UAVs ya que se le envía la referencia al Drone líder y a la vez la posición de este se manda a los demás Drones y con una función geométrica se crea la referencia que va a seguir cada Drone seguidor.

Hay que decir que esta solución es de una rápida implementación pero no tiene en cuenta muchas restricciones que hay en los UAVs ya que simplemente los seguidores siguen al líder con una estrategia geométrica y estática para la formación.

En este trabajo se propone la implementación, en cuadrirotores Ar.Drone, de una estrategia [14, 13] que tiene en cuenta restricciones tanto físicas y constructivas como virtuales originadas por la formación deseada, las diferentes dinámicas que puede tener cada Ar.Drone o perturbaciones externas. La idea básica es que para coordinar los Ar.Drones, la única información que es necesario compartir, ya sea a los otros drones o a un nodo supervisor es la referencia acondicionada. Esta referencia, se construye a partir de la consigna deseada y las restricciones antes mencionadas. De esta manera se logra *esconder* la dinámica de cada Ar.Drone (junto con las posibles diferencias que pueda tener cada uno, como diferentes saturaciones de actuador, potencia de motores o baterías) y transmitir la mínima información por la red.

Como se puede observar para conseguir estas configuraciones es necesario que todos los Drones compartan la misma red de comunicaciones para poder enviar y recibir toda la información necesaria, ya sea al nodo supervisor o directamente entre Ar.Drones.

La implementación se hace sobre un AR.Drone de Parrot con lo que antes de abordar este problema principal tenemos que programar el firmware de los mismos para adecuarlo a los requerimientos del control. Principalmente la modificación, extensión y mejora del sistema de comunicaciones para lograr una eficiente transferencia de información entre los

elementos integrantes del sistema: AR.Drones y nodo centralizado de control.

La gestión de trayectorias consta de la implementación de algoritmos de control y coordinación de formaciones basados en acondicionamiento de referencia por modos deslizantes. Estos algoritmos se basan en la modificación de las referencias de trayectoria de los AR.Drones teniendo en cuenta las restricciones físicas y constructivas de los mismos como así las restricciones virtuales impuestas por la formación.

En la sección 2 se describe brevemente el algoritmo de control, coordinación y generación de formación. A continuación en la sección 3 se detalla la implementación de dichos algoritmos en los AR.Drones, junto con las tareas previas realizadas para acondicionar los AR.Drones al sistema de control. Finalmente, se presentan los resultados obtenidos por simulación y junto con las conclusiones de este trabajo así como trabajo futuro.

2. Marco teórico de Coordinación de sistemas

En esta sección se plantea del problema de coordinación, una reseña del acondicionamiento de referencia por modos deslizantes.

2.1. Presentación del problema

Considere un conjunto de N sistemas, en nuestro caso Ar.Drones pero puede plantearse el problema de forma más general, no necesariamente con la misma dinámica. Asimismo, considere que cada sistema posee un lazo de control estable. Los sistema intervinientes, como se ha dicho, pueden tener diferentes restricciones y capacidades a la hora de seguir su referencia. En este contexto de control con restricciones aparece el concepto de *referencia realizable* [6]: la referencia más rápida que el sistema es capaz de seguir sin violar sus restricciones, manteniéndose siempre en lazo cerrado. Por ejemplo, en el caso de un sistema con restricciones de actuador, una referencia realizable nunca intentará llevar a los actuadores fuera de su rango de operación, ya que esto dejaría al sistema en lazo abierto (pudiendo dar lugar al fenómeno de *windup*).

La *coordinación* será entendida como la acción de lograr un comportamiento colectivo deseado para un conjunto de sistemas considerados. En este trabajo, se aborda el problema actuando sobre las referencias de los sistemas. De esta manera se tienen dos tipos de referencia sobre las que actuar: la referencia local de cada sistema y la referencia global.

Entre los comportamientos colectivos deseados podrían encontrarse:

- Mantener una función χ de las referencias locales lo más cerca posible de la referencia global.
- Mantener una distancia entre las referencias locales de los sistemas, una a una o entre centroides de agrupamientos.
- Lograr sincronización generalizada, como un caso límite de los anteriores.

Cabe destacar que esta definición de coordinación es muy general, pero si particularizamos por ejemplo el segundo caso, y las distancias a las que nos referimos son distancias en el plano euclideo de las trayectorias de los Drones, entonces la coordinación de convierte en generación de formación de vuelo.

2.2. Intercambio de información entre sistemas

Uno de los elementos necesarios para llevar a cabo la coordinación, es el intercambio de información entre ellos. La forma en que se ataca dicho problema en este trabajo es uno de sus principales aportes. La idea principal es que cada sistema envía información de sus restricciones locales a los otros sistemas intervinientes a través de su referencia realizable.

Dependiendo del nivel jerárquico al que se transmite la información (ver Figura 1) resulta:

la topología global cuando la información se transmite a un nivel superior tipo supervisor;

la topología local cuando la información se distribuye en el mismo nivel jerárquico a sistemas vecinos.

En cualquiera de las dos topologías, los sistemas individuales ocultan sus estados y salidas al resto de sistemas, enviando solamente la referencia realizable y minimizando la información transmitida. La referencia realizable refleja en qué situación se encuentra el sistema con respecto a sus restricciones físicas locales.

Aunque la información transmitida se minimice, la comunicación es en general el cuello de botella de las topologías centralizadas, ya que el tiempo para



Figura 1: Topología de conexión en las distintas configuraciones: A. Configuración descentralizada, donde se transmite información a nivel local entre Ar.Drones. B. Configuración centralizada, todos los Ar.Drones transmiten información al nodo supervisor (ordenador).

recoger toda la información de los sistemas depende directamente del número de sistemas y no del diámetro de la red, como en el caso descentralizado.

Asimismo, la topología global presenta los problemas normales de vulnerabilidad, puesto que el nodo supervisor centraliza toda la información, y es por lo tanto el punto débil de la red, ya que un fallo en el supervisor acarrea el fallo de todo el conjunto de sistemas.

2.3. Referencia Realizable y Acondicionamiento por Modos Deslizantes

A continuación se describe la metodología utilizada para obtener la referencia realizable, basada en ideas de invariancia y acondicionamiento de referencia por modos deslizantes.

La idea del acondicionamiento de referencia, basada en el concepto de lograr una referencia realizable, nace originalmente en el contexto del control con restricciones. En concreto, Hanus y Walgama [6, 17] han aplicado este tipo de soluciones para resolver el problema de saturación en los actuadores (windup).

Basándose en este enfoque y aprovechando las posibilidades del control por modos deslizantes, como por ejemplo la robustez a perturbaciones externas y a incertidumbre en los parámetros[10, 11], Mantz y colegas [7] han aplicado acondicionado de la referencia por modos deslizantes (SMRC) para obtener referencias realizables teniendo en cuenta las restricciones, tanto en los actuadores, como en las salidas. En [4, 5] se ha utilizado SMRC para acotar interacciones cruzadas en sistemas lineales MIMO. Luego en [8] se ha utilizado SMRC para búsqueda de consigna en sistemas no lineales con restricciones dependientes del estado. Y en [12, 15, 16] se ha realizado un esquema de coordinación de sistemas dinámicos, en donde se coordinan las referencias de dichos sistemas utilizando SMRC.

El control por modos deslizantes aparece aquí como un lazo adicional que actúa de manera transitoria para generar la referencia realizable, de la siguiente manera: El lazo de acondicionamiento estará *inactivo* hasta que las trayectorias del sistema alcancen por sí mismas la superficie de deslizamiento (intentando violar alguna restricción), pasando el lazo de acondicionamiento al estado *activo*. Una vez que las trayectorias dejen de querer violar dichas restricciones y el lazo principal pueda operar normalmente, entonces el lazo de acondicionamiento pasará nuevamente al estado inactivo.

Es importante señalar que, debido a la característica especial de esta aplicación, los inconvenientes típicos de control de estructura variable y modos deslizantes (es decir, el *chatering* y el modo de alcance) se evitan [3].

3. AR.Drone

El AR.Drone de Parrot es un vehículo aéreo no tripulado radiocontrolado de uso recreativo civil, funciona propulsado por cuatro motores eléctricos en configuración cuadricóptero y es similar en su estructura básica y aerodinámica a otros modelos radiocontrolados, pero se diferencia de todos ellos en que cuenta con un microprocesador y una serie de sensores entre los cuales se incluyen dos cámaras que le permiten captar lo que ocurre a su alrededor, más un conector Wi-Fi integrado que le permite vincularse a dispositivos móviles personales que cuenten con los sistemas operativos iOS, Android o Linux. Esto permite controlar al cuadricóptero directamente desde un dispositivo móvil, mientras se reciben por medio de este dispositivo las imágenes y datos de telemetría de lo que los sensores del drone están captando. AsÃ mismo cuenta de un puerto serie RS-232 embebido el cual es accesible reprogramando el firmware del AR.Drone.



Figura 2: Ar.Drone.

3.1. Hardware

El hardware del que disponemos en el AR.Drone es el siguiente:

- Microprocesador ARM9 RISC de 32 bits @ 468 MHz.
- Memoria DDR SDRAM de 128 MB @ 200 MHz.
- Sistema operativo con núcleo Linux.
- Módem Wi-Fi b / g.
- Conector USB/RS-232.
- Acelerómetro de 3 ejes construido con tecnología MEMS.
- Girómetro de 2 ejes.
- Girómetro de precisión de 1 eje para el control de guiñada.
- Altímetro por ultrasonidos, frecuencia de emisión: 40 kHz con un alcance de 6 metros.
- 4 motores sin escobillas, funcionando a 3.500 rpm con una potencia de 15 W.
- Batería de ion de litio de 3 celdas, capaz de entregar 1000 mA/hora con un voltaje nominal de 11,1 V.
- Cámara frontal con sensor CMOS de tipo gran angular de lente diagonal con una resolución de 640x480 píxeles (VGA).
- Cámara ventral con sensor CMOS de alta velocidad de lente diagonal con una resolución 176x144 píxeles.

3.2. Firmware

En el AR.Drone para que el puerto serie sea accesible tiene que tener la versión 1.3.3 del firmware.

Las dos modificaciones de firmware que vamos a realizar son en la configuración de las comunicaciones del mismo.

1. Por defecto el módem Wi-Fi está configurado como emisor, pero necesitamos que este se conecte a una red que creamos con un punto de acceso para que todos los AR.Drones y el nodo principal compartan la misma red. Para conseguir esto, en el directorio /bin/wifi_setup.sh hay que introducir los siguientes comandos:



Figura 3: Placa madre Ar.Drone.

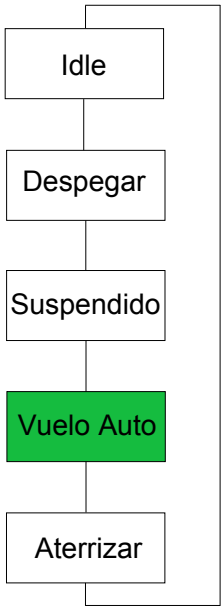


Figura 4: Maquina de estados AR.Drone.

iwconfig ath0 mode managed essid UAVROUTER

ifconfig ath0 192.168.10.51 netmask 255.255.255.0 up

route add default gw 192.168.10.1

Donde en iwconfig, mode managed indica que se

tiene que conectar a la red que indicamos en la essid, en nuestro caso el nombre de la red es UAVROUTER. En ifconfig indicamos la ip que va a tener el dispositivo que en nuestro caso van a ser 192.168.10.51 para el Drone1, 192.168.10.52 para el Drone2 y así consecutivamente para todos los que queramos coordinar. Y por último en route indicamos la ip del router que en nuestro caso será por defecto la 192.168.10.1.

En caso de querer poner nombre en la configuración Wi-Fi a los diferentes AR.Drones en el directorio /data/config.ini podemos indicar el nombre del Drone.

2. Por otra parte el puerto serie que es accesible desde los pines que disponemos en la placa es el ttyPA0. Por defecto este puerto está bloqueado, como necesitamos tener acceso al mismo lo que debemos hacer es entrar al directorio /etc/inittab y comentar la siguiente línea: ttyPA0::askfirst:/bin/sh.

3.3. Lazo de control

El lazo de control implementado en el Ar.Drone consta de una máquina de estados (ver Fig. 4) donde el estado que nos interesa tratar es el de vuelo-automático ya que es el estado en el que implementamos el control por SMRC descrito anteriormente.

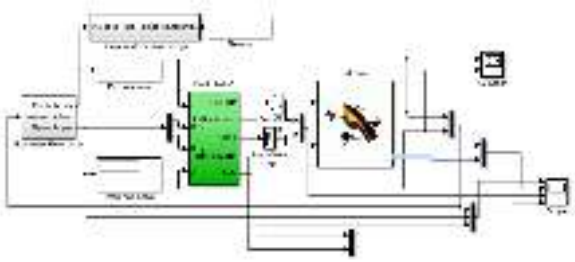


Figura 5: Simulink del lazo de control de un Drone.

En el estado de vuelo automático tenemos el control propiamente dicho (ver Figura 5), en el cual se puede observar perfectamente el controlador y el Modelo del AR.Drone. Dentro del controlador SMRC la parte más importante la podemos ver en el código (ver Figura 6) donde se observa como se calcula la distancia entre los drones y se crea la acción de control discontínua w.

3.4. Comunicación Wi-Fi

Para programar en C las comunicaciones Wi-Fi (ver Figura 7) nos creamos tres funciones: create, sendCommand, recvfrom. Los datos se transfieren en un único paquete con un formato string, una vez enviada o recibida la información se extrae o empaqueta de la forma que nos es más útil.

```
%%%%%%
dif_x = xf_ba_x-xf_ocbaci_x-Rbi;
dif_y = xf_ba_y-xf_ocbaci_y-Rbj;

dist= (dif_x)^2+ (dif_y)^2;

% superficie distancia entre xf_ba, xf_other y R
% menor que \Delta
% luego la region invariante es |xf_ba - xf_other - R| < \Delta
x = dist-Delta^2; %(cuando es menor que cero JUAN)

%implementacion de
%
%         y= signo_f( x )

ZM=1000;
x=0.5*(1/ZM^2).*max(-1/(ZM^2),x);
y=ZM*x;
y=max(1,y);
y=max(-1,y);

%       /
%      |  1      if   x > 0
% z =  |
%      |  0      if   x <= 0
%       \

A = 0.5*(y+1);

w = A;

% Aplicamos acción discontínua en la dirección de l ^ipsi
% En este caso se actúa en la dirección del segmento de distancia

vx3 = w*dif_x/sqrt(dist);
vy3 = w*dif_y/sqrt(dist);
```

Figura 6: Código control del SMRC.



Figura 7: Esquema red Wi-Fi.

Con la función create creamos los puertos, tanto de envío como de recepción, cada uno ligado a un número de puerto y una ip específica para cada AR.Drone.

Las funciones sendCommand y recevfrom nos sirven para poder enviar o recibir respectivamente en el puerto específico al cual queremos acceder.

3.5. Comunicación Serie RS-232

Un trabajo futuro que hay que realizar es introducir un GPS al Ar.Drone con lo que necesitamos acceder al puerto serie RS-232. Para utilizar este puerto utilizamos las librerías Termios.h donde están implementadas todas las funciones necesarias para utilizar este puerto y así podremos leer las coordenadas del GPS.

4. Simulaciones

A continuación se presentan los resultados obtenidos al simular un drone o una formación de tres drones siguiendo una trayectoria determinada. Esta trayectoria a seguir en ambos casos es una sucesión de puntos. Cuando el drone llega a un radio determinado del punto actual se considera que ha llegado al punto de destino y se actualiza el siguiente punto de referencia al control. Los puntos de la trayectoria utilizados en el ejemplo son: [0 0, 10 0, 15 2.5, 10 5, 5 5], siendo el primer valor referencia en coordenadas x y el segundo en coordenadas y.

En la simulación de un solo drone, los puntos consigna son enviados al drone. Luego el algoritmo de control SMRC, al filtrar la referencia, decide cómo realizar la trayectoria para alcanzar el punto deseado.

En el caso de la formación de tres drones, el control se encarga de crear una referencia realizable filtrada que puedan seguir los drones y a la vez modifica las referencias de los mismos para que lograr la formación que se ha determinado. En este ejemplo particular, se han modificado las restricciones físicas del Drone2, reduciendo la ganancia de las acciones de control, como resultado el Drone2 es más lento que los dos restantes, y se puede observar como los dos drones más rápidos modifican la referencia de su trayectoria para adecuarse a la formación deseada.

En la gráfica de las Trayectorias (ver Fig. 8) nos encontramos con tres representaciones en dos dimensiones. Se puede observar la referencia deseada (ref), la referencia acondicionada filtrada (ref. filtrada) es decir, la referencia que el algoritmo de SMRC envía al control proporcional y por último las salidas. Cabe destacar que la referencia acondicionada es tal que los Drones pueden seguirla perfectamente sin violar sus restricciones.

En la Fig. 9 se pueden observar la Referencia acondicionada en x por encima (roja) y en y por debajo (azul), seguidamente la acción de control, se aprecian escalones que van ligados directamente a la selección de los límites del SMRC que se van seleccionando en cada instante. Y por último tenemos la W que es la acción discontinua del SMRC que es la base de todo el control donde se aprecia claramente que actúa en mayor o menor medida en cada referencia cuando es necesario filtrar cada una de ellas.

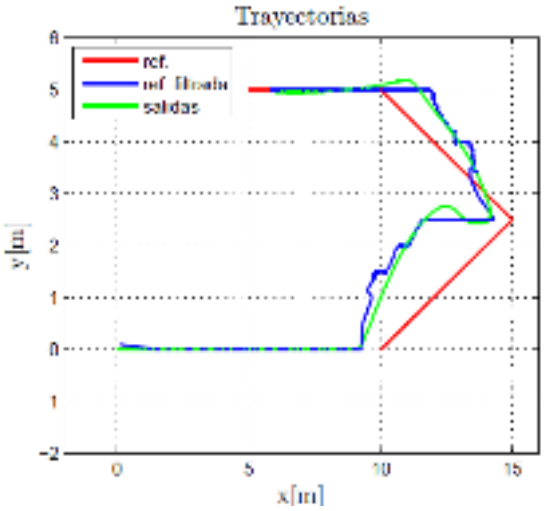


Figura 8: Trayectoria de un Drone.

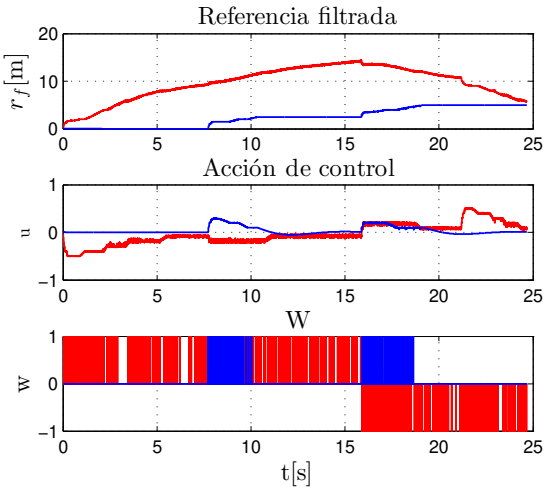


Figura 9: SMRC de un Drone.

En la trayectoria de tres Drones(ver Fig. 10), se puede observar los Drones 1 y 3 modifican las referencias de sus trayectorias para adecuarse a la velocidad del Drone2 manteniendo la formación.

Al igual que pasaba con un Drone (ver Fig. 9) cada referencia filtrada va ligada a una w (ver Fig. 11), siguiendo el mismo orden de colores.

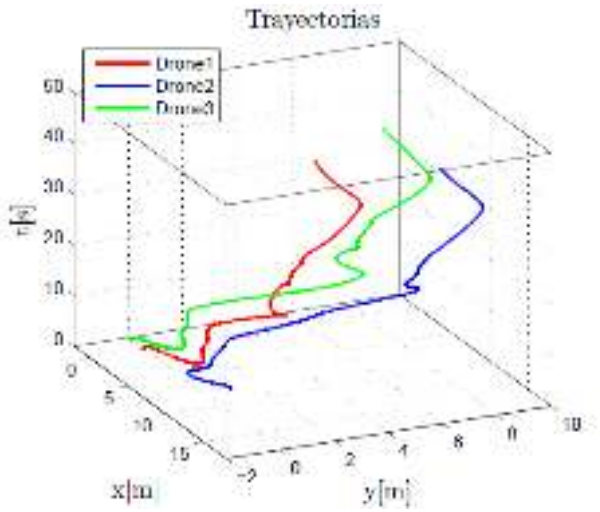


Figura 10: Trayectoria de tres Drones.

5. Conclusiones

En este trabajo, presentamos la implementación de una nueva estrategia con acondicionamiento de referencia por modos deslizantes desarrollada para hacer frente a la coordinación de múltiples agentes y al problema de control de la formación de vuelo de vehículos aéreos no tripulados. Los algoritmos se han implementado en los Ar.Drones de Parrot. Para esto se han extendido las interfaces de comunicaciones presentes en el Drone, para poder hacer frente a los requerimientos del algoritmo de control y generación de formaciones.

La implementación particular realizada considera la comunicación directa entre los Drones intervinientes, pero es fácilmente extensible al caso centralizado donde todos los Drones se comunican con un nodo supervisor que ejecuta las tareas de generación de formación teniendo toda la información que le llega desde los mismos. Dada la flexibilidad de la extensión realizada en el presente trabajo al sistema de comunicación de los Drones, no hay problema en hacer este cambio de configuración de control ya que al tener el sistema de comunicaciones descentralizado podemos optar cualquier configuración de control que queramos.

La implementación se ha validado con *hardware in the loop*, y se encuentra en fase de prueba en los quadrirrotores en este momento, tanto es su configuración descentralizada presentada en este trabajo como así también en la centralizada, para la cual se está desarrollando una interfaz en NI LabVIEW donde se controlen todos los drones.

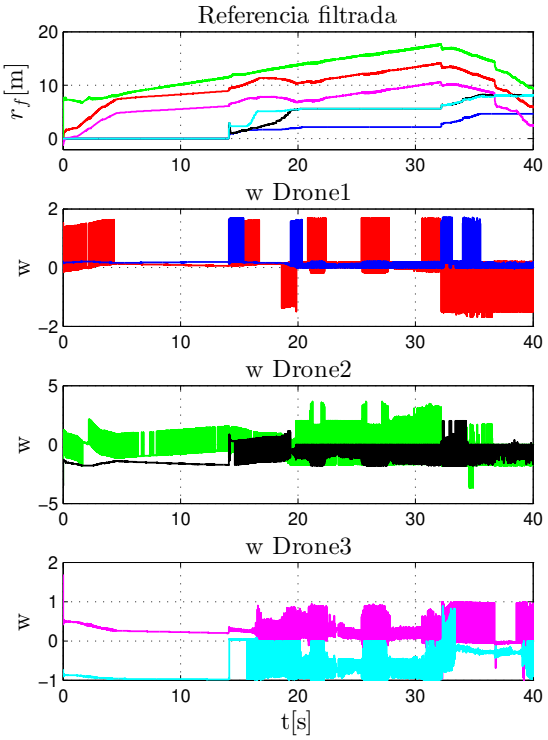


Figura 11: SMRC de tres drones.

Como trabajo posterior, se intentará mejorar el sistema de posicionamiento integrando un GPS en el AR.Drone a través el puerto serie RS-232 y con ellos mejorar el error de posición con el que trabajamos el control.

Agradecimientos

Este trabajo es apoyado parcialmente por el Ministerio de Ciencia e Innovación, Gobierno de España a través de los proyectos FEDER-CYCIT DPI201-28112-C04-01 y TIN2011-28082. Los autores agradecen el apoyo de la Universitat Politécnica de València para el desarrollo de este trabajo mediante la beca FPI/2009-21 y el proyecto PAID-06-11.

Referencias

[1] G. Antonelli. Interconnected dynamic systems: An overview on distributed control. *Control Systems Magazine, IEEE*, 33(1):76–88, 2013.

[2] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordina-

tion. *Industrial Informatics, IEEE Transactions on*, 9(1):427–438, 2013.

[3] F. Garelli. *Sistemas de estructura variable. Aplicación al control con restricciones.* PhD thesis, Facultad de Ingeniería. Universidad Nacional de La Plata., 2007.

[4] F. Garelli, R. Mantz, and H. De Battista. Limiting interactions in decentralized control of MIMO systems. *Journal of Process Control*, 16(5):473–483, 2006.

[5] F. Garelli, R. Mantz, and H. De Battista. Partial decoupling of non-minimum phase processes with bounds on the remaining coupling. *Chemical Engineering Science*, 61:7706–7716, 2006.

[6] R. Hanus, M. Kinnaert, and J. Henrotte. Conditioning technique, a general anti-windup and bumpless transfer method. *Automatica*, 23(6):729–739, 1987.

[7] R. Mantz, H. De Battista, and F. Bianchi. Sliding mode conditioning for constrained processes. *Industrial Engineering and Chemistry Research*, 43:8251–8256, 2004.

[8] J. Picó, F. Garelli, H. De Battista, and R. Mantz. Geometric invariance and reference conditioning ideas for control of overflow metabolism. *Journal of Process Control*, 19:1617 – 1626, 2009.

[9] W. Ren, R. W. Beard, and E. M. Atkins. Information consensus in multivehicle cooperative control. *Control Systems Magazine, IEEE*, 27(2):71–82, 2007.

[10] H. Sira-Ramírez. Sliding regimes in general non-linear systems: a relative degree approach. *International Journal of Control*, 50:1487–1506, 1989.

[11] V.Y. Utkin, J. Guldner, and J. Shi. *Sliding Mode Control in Electromechanical Systems*. Taylor & Francis, London, 1st edition, 1999.

[12] A Vignoni. Coordination in dynamical systems: a sliding mode reference conditioning approach. Master's thesis, Universitat Politècnica de València, http://personales.upv.es/alvig2/MT/vignoni-mthesis.pdf, 2011.

[13] A. Vignoni, F. Garelli, S. García-Nieto, and J. Picó. Uav reference conditioning for formation control via set invariance and sliding modes. In *3rd IFAC Workshop on Estimation and Control in Networked Systems*, 2012.

[14] A. Vignoni, F. Garelli, and J. Picó. Coordinación de sistemas con diferentes dinámicas utilizando conceptos de invarianza geométrica y modos deslizantes. *Revista Iberoamericana de Automática e Informática Industrial*, pages 1–12. En Prensa, 2013.

[15] A. Vignoni, J. Picó, F. Garelli, and H. De Battista. Dynamical systems coordination via sliding mode reference conditioning. In *Proceedings of the 18th IFAC World Congress*, volume 18, pages 11086–11091, 2011.

[16] A. Vignoni, J. Picó, F. Garelli, and H. De Battista. Sliding mode reference conditioning for coordination in swarms of non-identical multiagent systems. In *Variable Structure Systems, 12th IEEE International Workshop on*, pages 231–236, 2012.

[17] K. Walgama, S. Rönnbäck, and J. Sternby. Generalization of conditioning technique for antiwindup compensators. *IEEE Proceedings on Control Theory and Applications*, 139(2):109–118, March 1992.

OBSERVATION OF THE INTERNAL STATES OF A PEMFC ANODE GAS CHANNEL

J. Luna[1]
jluna@iri.upc.edu

C. Batlle[2], C. Kunusch[1], J. Riera[1], M.L. Sarmiento-Carnevali[1], M. Serra[1]
[1]Institut de Robòtica I Informàtica Industrial (UPC-CSIC), C/ Llorens i Artigas 4-6, 08028 Barcelona
[2]Departament de Matemàtica Aplicada IV, Universitat Politècnica de Catalunya, EPSEVG,
Av. V. Balaguer s/n, 08800 Vilanova I la Geltrú

Abstract

In this work a non-linear state observer is developed to estimate the concentrations of the different species inside a Proton Exchange Membrane Fuel Cell (PEMFC) anode gas channel. The implementation of the observer is based on the partial differential equations that define the dynamics of the system, taking into account spatial variations along the anode channel. Forward and backward discretization of the distributed model are employed to take advantage of the boundary conditions of the problem. For comparative reasons, linear and non-linear observability conditions are analysed, and a linear and a non-linear observer are designed. Likewise, two types of Sliding Mode corrective control actions (quasi continuous and super twisting input actions) are implemented in the non-linear state observer. Simulation results are presented to show the performance of the observers and the superiority of the non-linear one over the linear one can be clearly seen.

Keywords: PEMFC, distributed model, non-linear observers, super-twisting, sliding mode

1 INTRODUCTION

Fuel cells are an interesting alternative for clean energy production. Particularly, Proton Exchange Membrane Fuel Cells (PEMFC), with a high power density, are very promising for mass market applications such as automotive and stationary combined heat and power systems. Currently, researchers from all over the world are dedicating a great effort to improve efficiency, reduce degradation and decrease production costs of this technology. In the automatic control field, new estimation, diagnosis and control systems are being developed. In order to design appropriate controllers, different physical variables have to be known. While some of these variables are measurable with the current existing sensor technology, there are parts of the system that are inaccessible. This is the case of the gas channels, where only the inlet and outlet gas flows can be known. Changes in the gases concentrations and temperatures have significant effects on the PEMFC performance and may affect its durability. These variables exhibit a spatial variation in the direction of the channels of the anode and cathode which must be taken into account. The objective of this work is to obtain state observers that provide critical information about the dynamic behavior of the unknown internal states through the reconstruction from the available output measurements of the system. In spite of the industrial interest, research on state estimation in PEMFC is still incipient. Only few works have been published in this topic, mainly addressing the estimation of the oxidant flow and the humidity and water content in membranes and electrodes [2], [7]. In this work, a linear and a non-linear state observer are developed to estimate the concentrations of hydrogen and water vapour in the midpoint of the anode of a PEMFC. The observers are based on a 1D-model with spatial derivatives along the channel [6]. The linear observer is based on a Luenberger state estimation observer structure while the non-linear observer is based on the works of Davila et. al. [1], Levant [3] and Moreno et. al. [5].

In **Section 2** the general system description and statement of the problem is presented. In **Section 3** the mathematical model of the plant based on distributed parameters. In **Section 4** the linear observability is numerically computed for the linearized system and a Luenberger observer is implemented to estimate the states. In **Section 5** the non-linear observability condition is presented and tested for the system and a non-linear observer is developed to reconstruct the hydrogen and water concentrations in the PEMFC anode midpoint. In **Section 6** the results extracted from the simulations are presented and analyzed. Finally, in **Section 7** the conclusions of this work are presented and some future work plans intro-

duced.

2 SYSTEM DESCRIPTION AND PROBLEM STATEMENT

PEM fuel cells produce electrical energy from the consumption of hydrogen and air through an oxidation-reduction chemical reaction. At the anode side the hydrogen is catalytically split into protons and electrons. The protons travel through the electrolyte membrane to react with the air and the returning electrons in the cathode side of the cell, producing water. The electrons travel to the load of the system, producing electrical current through the connection between anode and cathode sides.

The overall system schematic for a single cell PEMFC is represented in Figure 1.

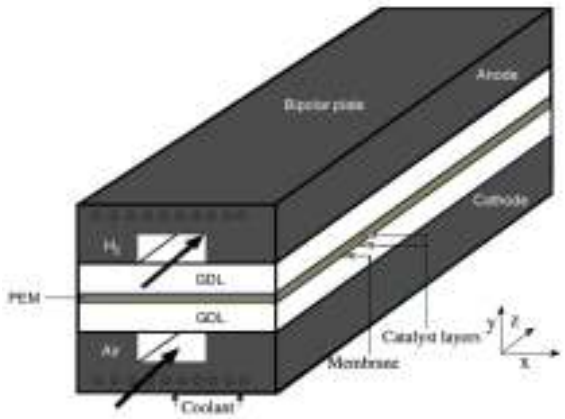


Figure 1: Single PEM cell representation

The work is focused on the anode supply gas channel and the estimation study will be performed in the y-z plane. The spatial derivatives are defined along the z-axis, while the hydrogen and water transport reaction terms are transversal to the supply channel in the y-axis direction. This is showed in the block diagram presented in Figure 2.

It is almost impossible to measure variables inside the gas channels due to the enclosed construction of the PEM fuel cell. Because of this, if we want to design controllers based on the internal states to avoid/reduce degradation in the system, we will need to estimate those values from the available inputs and measurements, which are assumed to be the inlet flow-rates and the outlet concentrations.

The anode gas channel model employed to test the solutions presented in this work has a z-axis length (L) of 0.4 meters and a y-axis channel thickness (δ) of 0.7 mm.

3 MODEL DEVELOPMENT

In this section a mathematical model of the PEMFC anode gas channel will be derived from the partial differential equations of the mass balances, the flow velocities and the pressure drops along the channel's z direction.

$$\frac{\partial c_i}{\partial t} = -\frac{\partial v c_i}{\partial z} - \frac{\dot{n}_i}{\delta} \quad (1)$$

$$v = -K\frac{\partial p}{\partial z} \quad (2)$$

$$p = RT\left(c_{H_2} + c_{H_2O}\right) \quad (3)$$

Table 1 displays all of the variables and parameters used in our study.

Table 1: Nomenclature and units employed

Coefficient	Description, Units
c	Gas concentration, $\frac{mol}{m^3}$
δ	Thickness of the channel, m
K	Pressure drop coefficient, $\frac{m^2}{sPa}$
$\dot{n}$	Molar flux density, $\frac{mol}{m^2 s}$
p	Pressure, Pa
R	Gas constant, $\frac{J}{molK}$
T	Temperature, K
v	Flow velocity, $\frac{m}{s}$
Δz	Discretization length, m

The state variables are the hydrogen (H_2) and water (H_2O) concentrations along the different discretized volumes of the channel. For each gas species, a subscript i will be added to the variables, being $i = 1$ the H_2 index and $i = 2$ the reference for H_2O. A second subscript j will refer to the discretized volume (e.g. $x_{2,2}$ indicates the H_2O concentration in the second volume of discretization).

To make use of the known boundary conditions of the system (input molar fluxes and external ambient pressure), a two-direction discretization will be applied to the spatial partial differential equations that define the dynamics of the system. This discretization,used on equations (1), (2) and (3) yields the following set of equations

$$\dot{x}_{i,j} = \frac{v_{j-1}x_{i,j-1}}{\Delta z} - \frac{v_j x_{i,j}}{\Delta z} - \frac{\dot{n}_{i,j}}{\delta} \quad (4)$$

$$v_j = \frac{K}{\Delta z}\left(p_j - p_{j+1}\right) \quad (5)$$

$$p_j = RT\left(x_{1,j} + x_{2,j}\right) \quad (6)$$

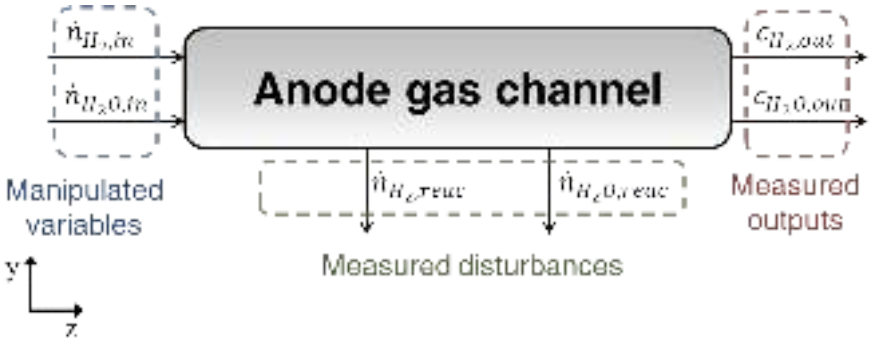


Figure 2: PEMFC anode gas channel model and variables involved in the observation problem

In this work we consider only two discretized volumes along the z axis, and (5) and (6) into (4) result in a 4-dimensional system of the form

$$\begin{aligned}
\dot{x}_{1,1} &= \frac{\dot{n}_{1,in}}{\Delta z} - f_1(\mathbf{x}) - \frac{\dot{n}_{1,1}}{\delta} \\
\dot{x}_{2,1} &= \frac{\dot{n}_{2,in}}{\Delta z} - f_2(\mathbf{x}) - \frac{\dot{n}_{2,1}}{\delta} \\
\dot{x}_{2,1} &= f_1(\mathbf{x}) - f_3(\mathbf{x}) - \frac{\dot{n}_{1,2}}{\delta} \\
\dot{x}_{2,2} &= f_2(\mathbf{x}) - f_4(\mathbf{x}) - \frac{\dot{n}_{2,2}}{\delta}
\end{aligned} \tag{7}$$

where the f_i are state-dependent functions that define the non-linear dynamics of the system, and are given by

$$\begin{aligned}
f_1(\mathbf{x}) &= C_1\left(x_{1,1}^2 + x_{2,1}x_{1,1} - x_{1,2}x_{1,1} - x_{2,2}x_{1,1}\right) \\
f_2(\mathbf{x}) &= C_1\left(x_{1,1}x_{2,1} + x_{2,1}^2 - x_{1,2}x_{2,1} - x_{2,2}x_{2,1}\right) \\
f_3(\mathbf{x}) &= C_1\left(x_{1,2}^2 + x_{2,2}x_{1,2}\right) + C_2 x_{1,2} \\
f_4(\mathbf{x}) &= C_1\left(x_{1,2}x_{2,2} + x_{2,2}^2\right) + C_2 x_{2,2}
\end{aligned} \tag{8}$$

The constants C_1 and C_2 are defined as a combination of the model's parameters that were presented in Table 1

$$C_1 = \frac{K}{\Delta z^2} RT \tag{9}$$

$$C_2 = \frac{K}{\Delta z^2} p^{amb} \tag{10}$$

Some further assumptions have been considered in our study. On one hand, the manipulated variables are the input molar flux densities both for H_2 and H_2O ($\dot{n}_{i,in}$). The transversal molar flux densities for the hydrogen and the water transport through the channel ($\dot{n}_{i,j}$) are included in the model as measured disturbances. Also, the temperature will be considered constant along the channel.

Our study can be generalized to the case of n discretized volumes along the gas channel. In Figure 3 the main structure of the system for this generalization is presented.

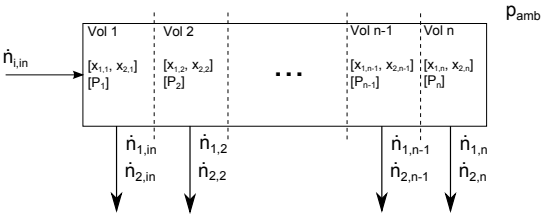


Figure 3: Generalized model of the anode gas channel

4 LINEAR OBSERVABILITY

Before analyzing the non-linear observability of the developed model, a linearization around an operating point $\mathbf{x_0}$ will be performed. From the linearization, the state space representation matrices A, B, C and D will be extracted in order to test the observability of the system around the working point showed in Table 2.

Table 2: Parameters for the linearization

Variable	Value
$\dot{n}_{1,in}$	$35\ \frac{mol}{m^2 s}$
$\dot{n}_{2,in}$	$10\ \frac{mol}{m^2 s}$
$\dot{n}_{1,1}$	$8.1 \cdot 10^{-3}\ \frac{mol}{m^2 s}$
$\dot{n}_{2,1}$	$-1.1243 \cdot 10^{-5}\ \frac{mol}{m^2 s}$
$\dot{n}_{1,2}$	$1.36 \cdot 10^{-2}\ \frac{mol}{m^2 s}$
$\dot{n}_{2,2}$	$-2.5247 \cdot 10^{-6}\ \frac{mol}{m^2 s}$

For linear systems in state space representation, the observability can be tested computing the rank of the observability matrix

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{11}$$

If the rank of (11) is equal to the number of states, it is said that the linear system is **observable** around the working point $\mathbf{x_0}$. Making these computations for the linearization of (7) around the point presented in Table 2 one gets that $rank(\mathcal{O}) = 4$ and hence, since our 2-volume system is 4-dimensional, it is possible to reconstruct the states from the information given by the outputs of the system. A simple Luenberger state observer has been implemented to obtain this reconstruction of $\mathbf{x_0}$ from the outputs of the system.

To test the behavior of the observer, simulation have been performed with input inlet gasses variations at times equal to 1 and 2.5 seconds. Besides that, a sinusoidal perturbation is added at time equal to 4 seconds to one of the measured disturbances signals. Results for the estimation of the linear model non-measured states are presented in Figure 4 and Figure 5.

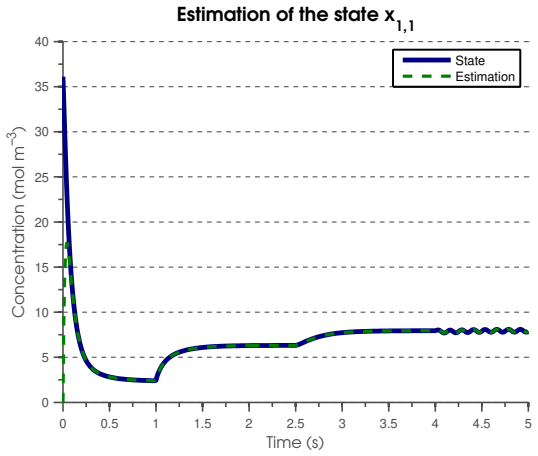


Figure 4: Linear state reconstruction of $x_{1,1}$

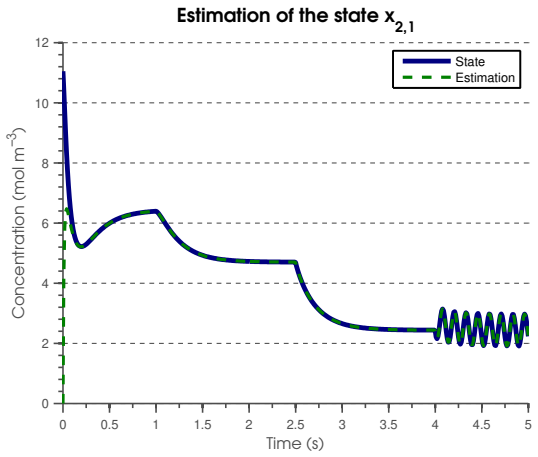


Figure 5: Linear state reconstruction of $x_{2,1}$

As expected from this classic proportional state estimator, the convergence to the unmeasured concentrations occurs very quickly and with no observation error, no matter the inlet gasses variations or the input disturbances applied to the linearized system.

5 NON-LINEAR OBSERVABILITY

Consider a non-linear dynamical system

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}) \end{aligned} \tag{12}$$

with state vector $\mathbf{x} \in \Re^n$, input vector $\mathbf{u} \in \Re^m$ and the scalar output vector $\mathbf{y} \in \Re^p$. The non-linear observer structure takes the known nominal part of (12) and a correction term to converge to the states in finite-time.

The main structure of the observer is defined as follows

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) + \mathbf{g}(\hat{\mathbf{x}})\mathbf{u_o} \\ \hat{\mathbf{y}} &= \mathbf{h}(\hat{\mathbf{x}}) \end{aligned} \tag{13}$$

with observed state vector $\hat{\mathbf{x}} \in \Re^n$ and observed output variable $\hat{\mathbf{y}} \in \Re^p$. The vector function $\mathbf{g}(\cdot)$ and the correction observer input $\mathbf{u_o}$ will be designed following the methodology presented in [1].

5.1 OBSERVABILITY ANALYSIS

In the non-linear case, the observation matrix $\mathcal{O}$ is defined by the space containing all repeated Lie derivatives of the output vector field $\mathbf{y}$

$$\mathcal{O}(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} h(\mathbf{x}) \\ L_{f(\mathbf{x})}h(\mathbf{x}) \\ \vdots \\ L^{n-2}_{f(\mathbf{x})}h(\mathbf{x}) \\ L^{n-1}_{f(\mathbf{x})}h(\mathbf{x}) \end{bmatrix} \tag{14}$$

where the Lie derivate evaluates the mapping between the output vector field and the states of the non-linear system presented in (12), and is given by

$$L_{f(\mathbf{x})}h(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}) \tag{15}$$

To guarantee the observability of (12), the matrix (14) has to be non-singular (full rank). This means that each state variable of the non-linear

system has to have a mapping from the output derivatives to its initial state $x(t_0)$ that distinguishes it from any other initial state x_0.

For the system described in (7), the computation of (14) can be done according to several combinations since there are two measured states. From all these possible combinations, we select the following one, which yields an observation matrix with relative degree equal to 1

$$\mathcal{O}(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} y_1 \\ \dot{y}_1 \\ y_2 \\ \dot{y}_2 \end{bmatrix} = \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} h_1(\mathbf{x}) \\ L_{f(\mathbf{x})}h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ L_{f(\mathbf{x})}h_2(\mathbf{x}) \end{bmatrix} \quad (16)$$

The observability matrix (16) allows to implement corrective control actions for (13) that can also be used to implement closed-loop control on dynamical systems with relative degree equal to 1. Several possibilities are discussed in the literature, and some of them will be compared in this work.

Developing (16) for (7) the following observabiility matrix is obtained

$$\mathcal{O}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ b_1 & b_2 & b_3 & b_4 \\ 0 & 0 & 0 & 1 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix} \quad (17)$$

with

$$\begin{aligned} b_1 &= C_1(2x_{1,1} + x_{2,1} - x_{1,2} - x_{2,2}) \\ b_2 &= C_1 x_{1,1} \\ b_3 &= -C_1 x_{1,1} - C_1(2x_{1,2} + x_{2,2}) + C_2 \\ b_4 &= -C_1 x_{1,1} - C1 x_{1,2} \\ d_1 &= C_1 x_{2,1} \\ d_2 &= C_1(x_{1,1} + 2x_{2,1} - x_{1,2} - x_{2,2}) \\ d_3 &= -C_1 x_{2,1} - C_1 x_{2,2} \\ d_4 &= -C_1 x_{2,1} - C_1(x_{1,2} + 2x_{2,2}) + C_2 \end{aligned} \quad (18)$$

As it was introduced before, the rank of (17) has to be equal to n, in this case 4, since this is the number of states of the studied system. The rank of such matrix only depends on the rank of

$$\mathcal{O}_p(\mathbf{x}) = \begin{bmatrix} b_1 & b_2 \\ d_1 & d_2 \end{bmatrix} \quad (19)$$

The rank of (19) must be 2 in order to have full rank in $\mathcal{O}(\mathbf{x})$. This condition will be met only and only if $det\left(\mathcal{O}_p(\mathbf{x})\right) \neq 0$. It may happen that at a given time the previous condition does not hold, and hence, by continuity, the determinant might be very close to zero for some time, in practice making the system non observable during a finite period of time. The solution is to disconnect the observer while the determinant is zero or very small, and to reconnect it afterwards.

5.2 NON-LINEAR OBSERVER

With the observability matrix computed for the non-linear case, it is possible to obtain the $g(\hat{\mathbf{x}})$ vector field of the observer. This procedure is based on the decoupling of the state reconstruction problem, stated by the expression

$$g(\hat{\mathbf{x}}) = \mathcal{O}^{-1}(\hat{\mathbf{x}})[0, 0, \ldots, 1]^T \quad (20)$$

which in our case is

$$g(\hat{\mathbf{x}}) = \begin{bmatrix} \frac{\hat{x}_{1,1}}{det(\mathcal{O}_p(\hat{\mathbf{x}}))} \\ \frac{2\hat{x}_{1,1} + \hat{x}_{2,1} - \hat{x}_{1,2} - \hat{x}_{2,2}}{det(\mathcal{O}_p(\hat{\mathbf{x}}))} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} g_1(\hat{\mathbf{x}}) \\ g_2(\hat{\mathbf{x}}) \\ 0 \\ 0 \end{bmatrix} \quad (21)$$

This result shows again the importance of having a full rank in the observability matrix. The determinant of $\mathcal{O}_p$ is located in the denominator of the uncoupled vector field $g(\hat{\mathbf{x}})$. If this denominator goes to zero, the observability tracking is lost during an unknown period of time. It can also be noticed that, as expected, there is no need to introduce corrective actions for the observation of the states $x_{1,2}$ and $x_{2,2}$ since they are the measured outputs of the system.

Substituting (21) and the estimation of (8) in the generalized state observer (13), the non-linear observer for each of the states can be defined as

$$\begin{aligned} \hat{x}_{1,1} &= \frac{\dot{n}_{1,in}}{\Delta z} - f_1(\hat{\mathbf{x}}) - \frac{\dot{n}_{1,1}}{\delta} + g_1(\hat{\mathbf{x}})u_{1,1} \\ \hat{x}_{2,1} &= \frac{\dot{n}_{2,in}}{\Delta z} - f_2(\hat{\mathbf{x}}) - \frac{\dot{n}_{2,1}}{\delta} + g_2(\hat{\mathbf{x}})u_{2,1} \\ \hat{x}_{1,2} &= f_1(\hat{\mathbf{x}}) - f_3(\hat{\mathbf{x}}) - \frac{\dot{n}_{1,2}}{\delta} \\ \hat{x}_{2,2} &= f_2(\hat{\mathbf{x}}) - f_4(\hat{\mathbf{x}}) - \frac{\dot{n}_{2,2}}{\delta} \end{aligned} \quad (22)$$

It only remains to design the corrective input actions of $\hat{x}_{1,1}$ and $\hat{x}_{2,1}$. As it was introduced before, the relative degree of the system is equal to 1 for each of the output vector fields ($h_1(\mathbf{x})$ and $h_2(\mathbf{x})$) in the observation matrix $\mathcal{O}(\mathbf{x})$. From this we can implement some suitable control actions

that are being used nowadays in closed-loop control of non-linear systems. In particular, for this work, two approaches will be considered

- Quasi-continuous sliding controller [3]
- Super-Twisting algorithm [5]

Both control actions show sliding dynamics based on sign switching functions. For both of them the output observation error e_y will be selected as the sliding surface.

The dynamics of the second order quasi-continuous controller developed in [3] are

$$u_{1,j} = -\alpha \frac{\dot{e}_y + |e_y|^{\frac{1}{2}} sign(e_y)}{|\dot{e}_y| + |e_y|^{\frac{1}{2}}} \tag{23}$$

Following the work presented in [5], we may introduce also the so called Super-Twisting algorithm, defined by

$$\begin{aligned} u_{1,j} &= -K_1 |e_y|^{\frac{1}{2}} sign(e_y) + v \\ \dot{v} &= -K_2 \frac{1}{2} sign(e_y) \end{aligned} \tag{24}$$

In the next section the non-linear observer (22) will be tested for both corrective input actions (23) and (24) in order to compare the obtained results and extract some conclusions.

6 SIMULATION RESULTS AND DISCUSSION

The designed non-linear observer will be put under the same simulation conditions as the linear case showed in section 4. This means that the input inlet gasses values of the system will be varied at time equal to 1 and 2.5 seconds. On the other hand, a sinusoidal perturbation will be added at time equal to 4 seconds.

The results of the performed simulations are presented in Figure 6 and Figure 7. The simulation has been computed for the two variants of the non-linear state observer. Besides, the linear observer has been put into use to test if it can reconstruct the non-linear states.

In the simulation results the linear states of the system, for the same inputs and disturbances variations, are also showed in order to compare the results with the dynamic behaviour of the non-linear system.

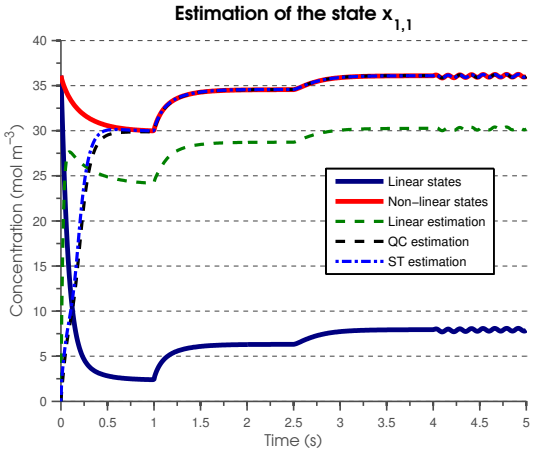


Figure 6: Non-linear state reconstruction of $x_{1,1}$

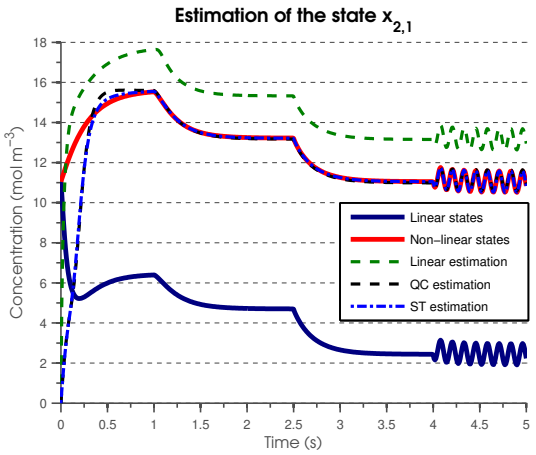


Figure 7: Non-linear state reconstruction of $x_{2,1}$

It can be inferred from Figure 6 and Figure 7 that the linear observer is unable to follow the trajectory of the non-linear dynamics. This justifies the need to develop the non-linear state observer presented in this work. Besides that, both corrective actions for the non-linear observer show similar dynamical responses, and they both converge to the states in finite-time.

The difference between the value of the state and the estimated value is the reconstruction error. For the unmeasured states $x_{1,1}$ and $x_{2,1}$, the state reconstruction error is showed in Figure 8 and Figure 9, where it is showed that the reconstruction error goes to zero in a relatively short time (around 1.5 seconds after the beginning of the simulation).

On the other hand, the simulation results show that for the same values of the inputs and the known disturbances, the behaviour of the linearized system is different from the non-linear model of the plant. In particular, an offset exists

between both dynamical responses.

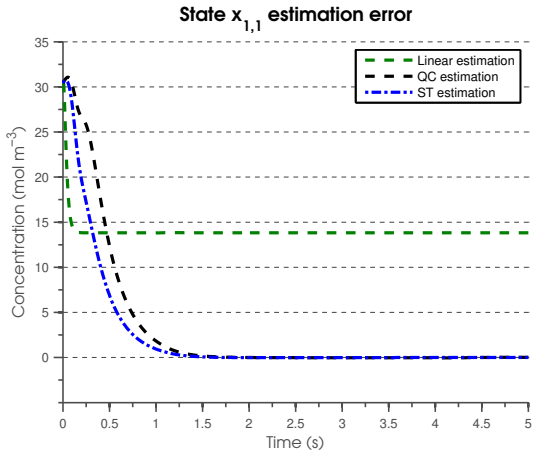


Figure 8: State reconstruction error of $x_{1,1}$

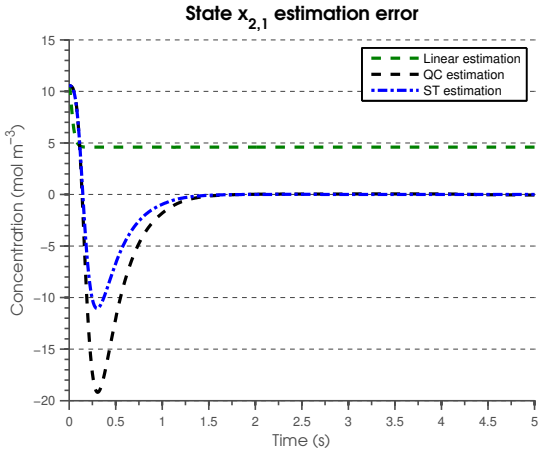


Figure 9: State reconstruction error of $x_{2,1}$

7 CONCLUSIONS

While the linearization of the system may allow us to work at a certain operating point with a linear model, the constantly changing conditions of PEMFC-based systems make necessary to use the full non-linear description. Our work shows that there exists an offset between both models (linear and non-linear), and the complete state representation has to be employed to fully understand the behaviour of the plant.

The physical construction of the fuel cell makes it impossible to know the evolution of the inside concentrations values with time, and a state observer has to be implemented to estimate these concentrations. The linear observer never converges to the non-linear response, making it necessary to implement the non-linear observer model that was developed for this particular application.

The observability matrix has been constructed from the mapping of the outputs with a relative degree equal to 1. This allows the implementation of the studied corrective sliding control actions that have been presented in this work. For higher numbers of discretized volumes, or for other combinations of the observability matrix, new controllers will have to be implemented.

Up until this point, all the disturbances of the system have been considered known beforehand. In reality, the estimation of the water transport inside the channel (and the rest of the PEMFC) is a complex task. In future work, it is planned to analyze the degree of model and input uncertainties that can be included into the design of the non-linear observer.

It is also interesting to remark that, while in the present work only 2 discretized volumes have been considered, the goal is to generalize the approach up to n volumes. This way, a complete and detailed profile of the hydrogen and water concentrations can be extracted with only the output values in the last discretized volume. This is theoretically possible because of the sequential structure of the model but numerical complications arise when the number of discretized volumes increases. This detailed information will be useful to implement controllers in the system in order to maximize efficiency and avoid degradation.

Finally, it has to be clarified that this work presents the approach for the observation in the anode layer of the fuel cell. In the future, the rest of the PEMFC components (anode, membrane, etc.) will be added to the model and more state observers will be designed to extract additional information of the system dynamics.

Acknowledgements

This work has been supported by national project MESPEM with reference DPI2011-25649.

References

[1] Davila J., Fridman L., Pisano A., Usai E., (2008), "Finite-time state observation for non-linear uncertain systems via higher-order sliding modes", *Internation Journal of Control*, vol 82, no 8.

[2] Kazmi I.H., Bhatti A.I. Iqbal S.A., (2009), "A non-linear observer for PEM fuel cell system", *IEEE 13th International Multitopic Conference (INMIC)*

[3] Levant A. (2005), "Quasi-Continuous High-Order Sliding-Mode Controllers", *IEEE*

Transactions on Automatic Control, vol 50, no 11.

[4] Mangold M., Bück A., Hanke-Rauschenbach R., (2010), "Passivity based control of a distributed PEM fuel cell model", *Journal of Process Control*, vol 20, pp 292-313.

[5] Moreno, J. A., Osorio, M. A., (2008). "A lyapunov approach to second-order sliding mode controller and observers", *Proceedings of the 47th IEEE Conference on Decision and Control*

[6] Sarmiento-Carnevali M., Serra M., Batlle C., (2013) "Distributed parameter model simulation tool for PEM fuel cells", *International Journal of Hydrogen Energy*. DOI:10.1016/j.ijhydene.2013.04.015

[7] Thawornkuno C., Panjapornpon C., (2008) "Estimation of Water Content in PEM Fuel Cell", *Chiang Mai J. Sci*, 35(1): 212-220

Modelo para la generación de datos de consumo energético residencial basado en un enfoque probabilístico

J.K. Gruber, M. Prodanovic
Unidad de Sistemas Eléctricos, Instituto IMDEA Energía
Avda. Ramón de la Sagra, 3, 28935 Móstoles (Madrid)
jorn.gruber@imdea.org, milan.prodanovic@imdea.org

Resumen

En el campo de la gestión de demanda, aplicada a usuarios domésticos individuales o agregados, se han observado muchos avances tecnológicos significativos en los últimos años. Recientemente, se han desarrollado diferentes modelos del consumo doméstico con el objetivo de mejorar el modelado de la demanda energética. En este artículo se presenta un nuevo modelo que permite crear datos sintéticos de la demanda de energía para uno o varios hogares usando un enfoque probabilístico. El modelo de la demanda considera la contribución de los diferentes electrodomésticos en un hogar y ofrece un alto grado de flexibilidad por su modular estructura. Los datos de consumo generados artificialmente con el modelo se pueden usar en el desarrollo de algoritmos para la gestión de la demanda doméstica a nivel individual o agregado. El modelo propuesto y sus principales propiedades se ilustran mediante los perfiles de demanda domésticos generados.

Palabras clave: demanda energética doméstica, generación de cargas, modelo probabilístico.

1. Introducción

En los últimos años, la gestión de energía y los contadores inteligentes se han convertido en campos de investigación importantes y han atraído un creciente interés por parte de la comunidad científica y de la industria. En el pasado, los operadores de las redes eléctricas consideraban la gestión de energía principalmente como herramienta para ofrecer al cliente un servicio fiable e ininterrumpido. Recientemente, los consumidores finales han prestado más atención a los temas de gestión de energía a nivel doméstico por la disponibilidad de contadores inteligentes asequibles y su instalación a gran escala en muchos países, incluyendo Italia, Reino Unido y España. Además, los problemas de suministro de energía, los consumidores más sensibilizados ante las cuestiones medioambientales y la subida continua de los precios de energía, convierten la gestión de energía, especialmente la gestión de la demanda, en una herramienta idónea para resolver algunas de las cuestiones más urgentes relacionadas al sector de la energía.

El desarrollo de nuevos productos y servicios basados en la gestión de la energía requiere unos conocimientos detallados del consumo energético del consumidor final. En el caso de no disponer de datos de consumo se usan frecuentemente modelos matemáticos para generar perfiles de demanda. En [10] se presenta un método para predecir la demanda energética combinando los patrones de ocupación del hogar con el consumo de diferentes electrodomésticos, calefacción y agua caliente sanitaria. El modelo de demanda eléctrica doméstica descrito en [8] está basado en perfiles de actividad, patrones de ocupación y datos estadísticos sobre el número de cada tipo de electrodoméstico en un hogar. El modelo usa un enfoque estocástico y un tiempo de muestreo bajo para calcular la demanda doméstica. En [3] se generan datos sintéticos del consumo energético doméstico combinando los datos estadísticos de los electrodomésticos considerados y una clasificación de consumidores basada en factores sociales. En [6] se analiza el consumo eléctrico agregado de muchos consumidores y se presenta un método para determinar la demanda de diferentes componentes (como calefacción, ventilación y aire acondicionado, entre otros) del consumo agregado.

Un campo importante relacionado con el modelado del consumo energético doméstico es la identificación de patrones de uso. La influencia del tipo de vivienda, factores socio-económicos y diferentes electrodomésticos sobre el consumo eléctrico residencial se analiza en [1, 7]. En [1] se ha observado que un grupo de consumidores residenciales muestra picos importantes en la demanda agregada como resultado de altos factores de coincidencia en las actividades de los consumidores considerados. En [7] se ha demostrado la gran influencia de secadoras, lavavajillas y hornos eléctricos sobre la demanda máxima residencial y se menciona el gran potencial de secadoras en el momento de desplazar su demanda en el tiempo (*demand shifting* en inglés). Un enfoque novedoso se usa en [2] para caracterizar la demanda residencial agregada mediante distribuciones probabilísticas. Un análisis

muestra que la desviación estándar de la demanda agregada se reduce con el número de consumidores agregados.

En [5] se han usado patrones de conducta y factores relacionados con la vivienda y características sociales para desarrollar perfiles de consumidores (p.ej. familias, hogares con altos ingresos, jubilados, etc.). Los perfiles de consumidores sirven para el desarrollo de políticas de ahorro energético y los patrones de conducta pueden ser empleados en programas de simulación o cálculos de eficiencia energética. Por último, [9] presenta una estrategia de demanda doméstica basada en la minimización de los gastos de energía de los consumidores, considerando una fijación de precios de electricidad en tiempo real. El estudio muestra que pequeños cambios en la temperatura del termostato pueden reducir considerablemente el consumo de energía y los costes correspondientes sin una pérdida significativa de confort del consumidor.

En este artículo se presenta el desarrollo de un nuevo modelo que permite generar datos detallados del consumo energético de una o varias viviendas. El modelo determina la configuración de cada hogar usando un enfoque probabilístico basado en distribuciones discretas. El cálculo de la demanda energética se lleva a cabo para cada electrodoméstico en los hogares considerados y distingue entre días laborables y festivos.

El resto del trabajo se organiza de la siguiente manera. En la Sección 2 se presenta un esquema de clasificación de los electrodomésticos más comunes. La Sección 3 describe en detalle el modelo de demanda energética residencial desarrollado, incluida la configuración automática de hogares y la generación de datos de consumo. La implementación del modelo y algunos de los resultados obtenidos se muestran en la Sección 4. Finalmente, en la Sección 5 se presentan las conclusiones derivadas del modelo desarrollado.

2. Clasificación de electrodomésticos

El modelo de demanda energética residencial define los diferentes electrodomésticos a través de algunos datos básicos. Las definiciones de los electrodomésticos recogen, entre otros datos característicos, las clasificaciones basadas en la potencia y el uso por parte del consumidor. Los siguientes párrafos describen el esquema de clasificación y su uso en la definición de los electrodomésticos.

2.1. Patrones de potencia

La mayoría de los electrodomésticos poseen una demanda energética característica que permite el desarrollo de un esquema de clasificación basado en patrones de potencia. Los siguientes tres patrones de potencia básicos se han usado para aproximar las curvas de demanda de los diferentes electrodomésticos considerados en el modelo de consumo energético residencial:

- Patrón 1: aparatos en funcionamiento continuo con una demanda energética casi constante[1] (ver la figura 1(a)). Electrodomésticos típicos de este grupo son los frigoríficos y congeladores.
- Patrón 2: aparatos utilizados ocasionalmente con una demanda constante distinta de cero durante el uso y sin ningún consumo de energía en los períodos sin uso[2] (ver la figura 1(b)). Electrodomésticos típicos con este patrón son lavadoras y secadoras.
- Patrón 3: aparatos con variaciones frecuentes en la demanda energética (ver la figura 1(c)). Un típico ejemplo de este grupo es la iluminación del hogar[3].

2.2. Modos de utilización

Otro esquema de clasificación de electrodomésticos está basado en datos acerca de la utilización del aparato por parte del consumidor, incluyendo la frecuencia, duración y hora de uso. Usando estos datos, se han definido los siguientes modos de utilización:

- Modo A: aparato en funcionamiento continuo, nunca apagado, basado en el patrón 1. Aparatos típicos son frigoríficos, detectores de humo y faxes.
- Modo B: aparato usado casi a diario, nunca más de una vez al día, uso muy prolongado (más de 12 horas), basado en el patrón

[1] Algunos electrodomésticos aproximados con el patrón 1 pueden mostrar variaciones de demanda significativas a corto plazo. Sin embargo, estos aparatos poseen un valor medio de demanda constante para períodos prolongados.

[2] Algunos electrodomésticos aproximados mediante el patrón 2 pueden tener cierto consumo de energía durante períodos de no utilización como consecuencia de funcionamiento en espera.

[3] Nótese que se podría aproximar la demanda de una sola bombilla usando el patrón 2. Sin embargo, la demanda energética agregada de todas las bombillas de un hogar muestra cambios frecuentes y corresponde al patrón 3.

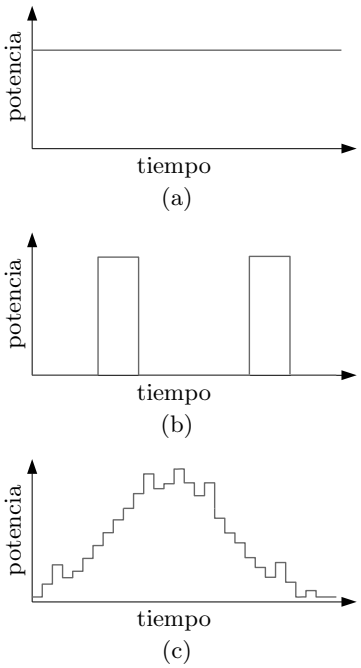


Figura 1: Clasificación de electrodomésticos por el patrón de demanda energética 1 (a), 2 (b) and 3 (c).

2. Ejemplos comunes son decodificadores, enrutadores y módems.

- Modo C: aparato usado varias veces a la semana, nunca más de una vez al día, principalmente por la tarde, uso prolongado (entre 2 y 12 horas), basado en el patrón 2. Aparatos típicos son los dispositivos de ocio (televisores, videoconsolas, etc.) y ordenadores.
- Modo D: aparato usado varias veces a la semana, principalmente por la mañana y por la tarde, períodos cortos a intermedios de uso(entre 0.5 y 3 horas), basado en el patrón 2. Ejemplos típicos son equipos de lavado, calentadores y calefacciones.
- Modo E: grupo de aparatos con frecuencia de uso no homogénea (entre 0.1 y más de 10 veces a la semana), principalmente a la hora de preparar las comidas, basado en el patrón 2. Aparatos típicos son los aparatos de cocción.
- Modo F: aparato con consumo energético variable, basado en el patrón 3. Un ejemplo típico es la iluminación del hogar.
- Modo G: aparato usado una vez al día, uso prolongado, hora de uso variable, basado en el patrón 2. Aparatos de este grupo son los cargadores y teléfonos inalámbricos.
- Modo H: aparato usado una vez al día, uso prolongado y siempre a la misma hora, basado en el patrón 2. Ejemplos típicos son deshumidificadores y acuarios.
- Modo I: aparato no representado por los modos A-H, basado en el patrón II. Aparatos de este grupo son secadores de pelo y mantas eléctricas.

Los electrodomésticos representados por el modo de utilización D ofrecen una gran flexibilidad debido a la posible programación de consumo de energía y atraen el mayor interés con vista a la optimización de la demanda energética. Estos electrodomésticos se pueden usar en combinación con algoritmos de control con el fin de aplanar la curva de demanda. En [4] se presenta una amplia lista de electrodomésticos con sus respectivas clasificaciones basadas en el modo de utilización se puede ver.

2.3. Definición de electrodomésticos

Los electrodomésticos considerados en el modelo de consumo energético residencial deben ser definidos previamente mediante algunos datos básicos. Durante el desarrollo del modelo se han recogido los datos básicos de los 84 electrodomésticos dados en [4]. Estos datos se utilizan tanto en la configuración del hogar como en la generación de la demanda energética.

Gran parte de los datos básicos de los electrodomésticos considerados ha sido extraída de [11], incluyendo el consumo anual, el promedio de aparatos por hogar, los ciclos de operación anuales, la duración de ciclo y la probabilidad de uso en función de la hora y el día (ver un ejemplo de la probabilidad de uso en la figura 2). Se completaron las definiciones con valores medios de demanda (para ciclos de operación y períodos de no utilización) y la clasificación basada en los modos de utilización (ver Sección 2.2). Además se ha añadido información sobre la probabilidad de que un electrodoméstico entre en modo de espera (en lugar de apagarse) después de un ciclo de operación. Por lo tanto, la definición de un electrodoméstico considerado en el modelo de consumo energético residencial comprende los siguientes datos básicos:

- consumo anual [kWh]
- promedio de aparatos por hogar [-]
- demanda en ciclo de operación [W]
- demanda en modo de espera [W]
- duración de ciclo [h]
- ciclos anuales [-]

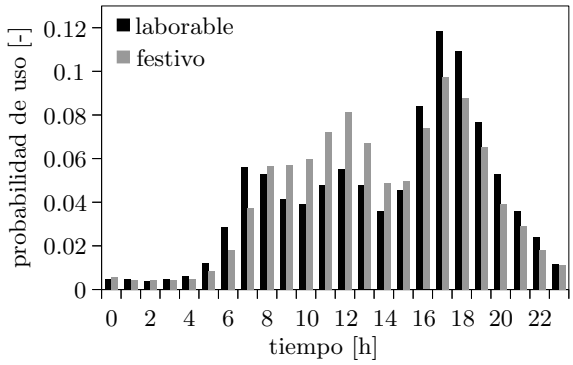


Figura 2: Probabilidad de uso para un hervidor de agua en función de la hora y el tipo de día (laborable o festivo).

- modo de utilización [-]
- probabilidad de la hora de utilización [-]
- probabilidad de modo de espera [-]

3. Modelo de consumo energético residencial

El modelo de consumo energético residencial genera un perfil de demanda doméstica en tres pasos principales. Una vez establecida la configuración inicial del hogar, el modelo permite determinar la frecuencia de uso de cada electrodoméstico a lo largo de todos los días del periodo considerado. Finalmente, es posible obtener la hora exacta de operación de cada uno de los electrodomésticos en dicho periodo.

3.1. Configuración de hogares

El modelo de consumo energético residencial determina en un primer paso la configuración de uno o varios hogares. Para la mayoría de los electrodomésticos se usa un enfoque probabilístico basado en distribuciones binomiales para calcular el número de aparatos en un hogar. Sin embargo, para algunos electrodomésticos se han considerado excepciones.

El enfoque probabilístico usado está basado en la distribución binomial $\mathcal{B}(\eta, \rho)$ donde η representa el número máximo de aparatos de cierto tipo de electrodoméstico y ρ denota el resultado del promedio de aparatos por hogar ϑ dividido por η. Este enfoque basado en una distribución binomial genera sólo valores no negativos enteros con un valor medio de ϑ.

Una de las excepciones consideradas en la configuración de hogares está vinculada a la iluminación. Debido a su importancia, se ha asumido que todos los hogares disponen de iluminación eléctrica. Otra excepción afecta a electrodomésticos que pueden ser operados con gas o electricidad. En este caso, donde se incluyen hornos y cocinas, se ha asumido que todos los aparatos de este tipo de electrodoméstico que bien usan gas, o por el contrario electricidad (p.ej. si un hogar posee varios hornos, todos usan gas o todos usan electricidad). La última excepción limita la suma de calefacciones eléctricas y de gas a un aparato por hogar. Esta limitación se usa también en combinación con calentadores eléctricos y de gas.

Otro aspecto importante es la demanda energética de los electrodomésticos que no se encuentran en uso. Dicha demanda depende principalmente si un dispositivo entra en modo de espera después de su uso o si se apaga por completo. Durante la configuración de los hogares se determina la demanda energética en fases de no utilización para cada uno de los electrodomésticos. El enfoque probabilístico compara un número aleatorio del conjunto $\mathcal{M} = \{0 \leq x \leq 1\}$ con la probabilidad ψ de modo de espera de un electrodoméstico. En el caso $x \leq \psi$, el aparato usa el modo de espera y muestra en las fases de inactividad la demanda en modo de espera definida en la Sección 2.3. De lo contrario ($x > \psi$), el aparato está apagado en los períodos de no utilización y tiene una demanda de 0 W. Es importante subrayar que, dependiendo de la configuración inicial, un aparato siempre entra en modo de espera o siempre está apagado.

3.2. Frecuencia de uso

El modelo de consumo energético calcula para la mayoría de los electrodomésticos el uso diario utilizando un enfoque probabilístico. Adicionalmente se han considerado diferentes excepciones en el cálculo del uso diario de algunos tipos de electrodomésticos.

El uso diario de los electrodomésticos representados por los modos D, E e I (ver Sección 2.2) se calcula mediante la distribución binomial $\mathcal{B}(\eta, \rho)$. El parámetro η es el número máximo de usos diarios de un aparato y ρ representa el resultado del valor medio de ciclos diarios φ de ese aparato dividido por η. La distribución binomial usada garantiza la generación de números enteros con un valor medio de φ.

En el caso de aparatos definidos por los modos B y C, se determina el uso diario mediante un enfoque probabilístico considerando un límite de un ciclo de operación por día. La inclusión de esta limitación fue necesaria debido a las largas duraciones de los ciclos de operación de los electrodomésticos en el caso de los grupos mencionados. El uso diario se calcula usando una distribución de Bernoulli

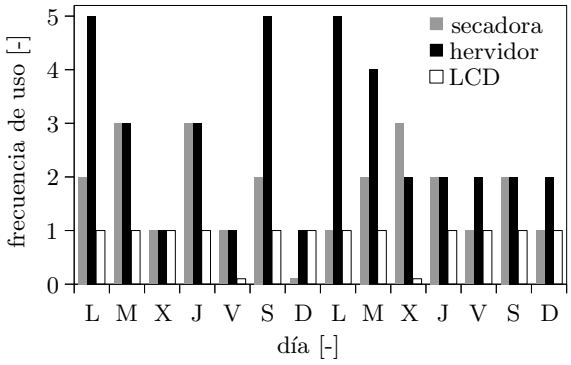


Figura 3: Ejemplo del uso diario para una secadora, un hervidor de agua y un televisor LCD.

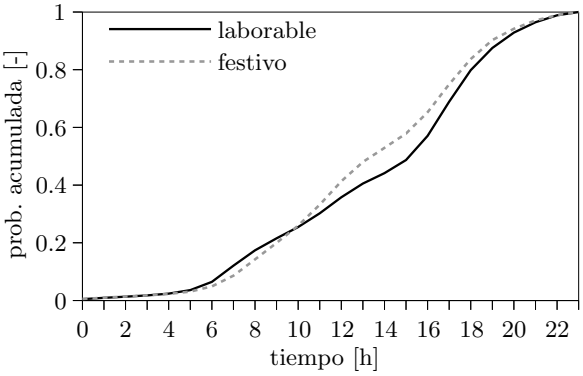


Figura 4: Probabilidad acumulada de la hora de utilización de un hervidor de agua para días laborables (línea negra) y festivos (línea gris).

$\mathcal{B}(\rho)$ donde ρ representa el valor medio de ciclos diarios φ. Para los electrodomésticos representados por los modos de utilización G y H se supone exactamente un ciclo de operación por día (esa regularidad se puede explicar por el probable uso de temporizadores). Por último, los electrodomésticos contenidos en los grupos A y F están usados en operación continua[4].

La figura 3 presenta un ejemplo del uso diario calculado para algunos electrodomésticos. La influencia de los diferentes modos de utilización resulta en el caso del televisor LCD (modo C) en un máximo de un ciclo de operación diario y en la lavadora (modo D) y el hervidor de agua (modo E) en varios usos diarios.

3.3. Serie temporal de la demanda

El modelo de consumo energético residencial determina la hora exacta de operación de cada electrodoméstico usando un enfoque probabilístico y aprovechando los datos sobre el uso diario. Para la generación de la curva de demanda de cada electrodoméstico y la demanda total del hogar se usa un tiempo de muestreo de 15 minutos. El tiempo de muestro elegido representa un compromiso entre la resolución necesaria para la gestión de demanda individual o agregada y la cantidad de datos generada.

Para la mayoría de electrodomésticos (modos de utilización B, C, D, E, G e I) se calcula la hora exacta de operación comparando un valor aleatorio del conjunto $\mathcal{M} = \{0 \leq x \leq 1\}$ con la probabilidad acumulada de la hora de utilización (ver un ejemplo de la probabilidad acumulada en la figura 4). La hora inicial de operación del aparato es en la que el número aleatorio corresponde con el valor de la curva de probabilidad acumulada. Durante la

[4]En el caso de la iluminación la demanda energética varía a lo largo del día y se asume un consumo muy reducido por la noche.

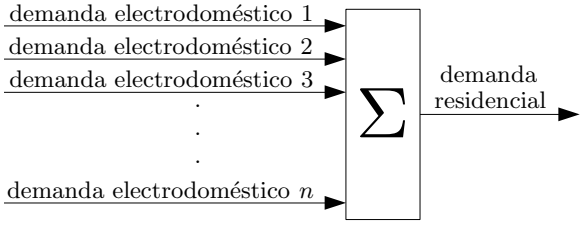


Figura 5: Diagrama del cálculo de la demanda residencial total.

muestra inicial y las siguientes (dependiendo de la duración del ciclo de operación) el aparato está en uso y su consumo corresponde a la demanda en ciclo de operación definida en la Sección 2.3. En el caso de electrodomésticos que se usan varias veces al día, el procedimiento evita el solapamiento de diferentes ciclos de operación.

Para los modos de utilización A, F y H se han considerado excepciones en el cálculo de la hora exacta de uso de los electrodomésticos. Los electrodomésticos definidos por el modo A operan de forma continua con una demanda constante en todas las muestras diarias. En el caso del modo H se usan los electrodomésticos todos los días a la misma hora. Por ello se calcula el tiempo de uso de estos dispositivos sólo una vez, comparando un valor aleatorio del conjunto $\mathcal{M} = \{0 \leq x \leq 1\}$ con la probabilidad acumulada de la hora de utilización. Para los electrodomésticos representados por el modo F el consumo corresponde al producto entre la probabilidad de la hora de utilización y la demanda dada en la información básica del aparato. Finalmente, la demanda total residencial se calcula sumando las demandas generadas por los diferentes aparatos (ver la figura 5).

La figura 6 muestra las series temporales de algunos electrodomésticos de un hogar tipo. En las curvas de demanda de los electrodomésticos se pueden observar grandes diferencias en la duración de

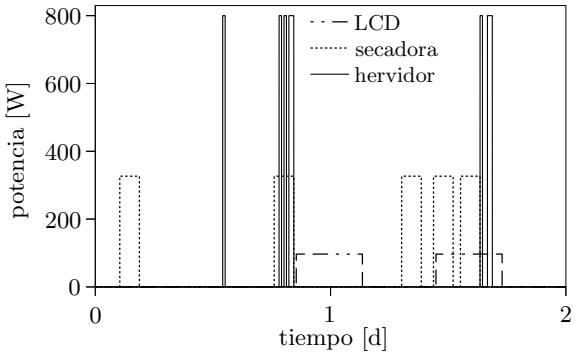


Figura 6: Demanda energética de un televisor LCD, una lavadora y un hervidor de agua durante un período de dos días.

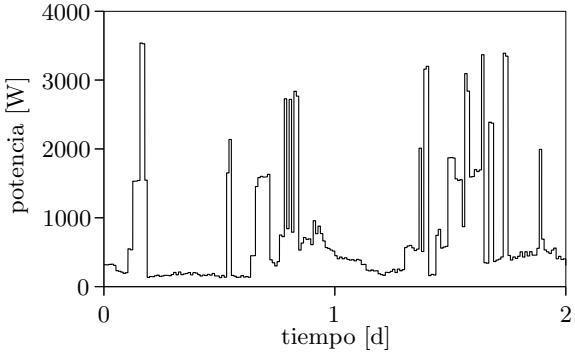


Figura 7: Demanda residencial total de un hogar tipo durante un período de dos días.

los ciclos de operación y en el nivel de potencia[5]. La demanda residencial total del hogar tipo (ver la figura 7) presenta importantes variaciones a lo largo del día mostrando una menor demanda durante las primeras horas del día y algunos picos de consumo causados por el hervidor de agua y otros aparatos de alta potencia.

4. Implementación y resultados

Finalmente, se ha implementado el modelo de consumo energético residencial considerando una amplia gama de diferentes electrodomésticos. Con el modelo implementado se han realizado numerosas simulaciones para validar la configuración de hogares y la generación de la correspondiente demanda energética.

[5]La duración más corta del ciclo de operación es de 15 minutos debido al tiempo de muestreo elegido. En el caso del hervidor de agua, que en general tiene un ciclo de operación más corto, se ha ajustado el nivel de potencia a un ciclo más largo con el fin de mantener constante la cantidad de energía consumida.

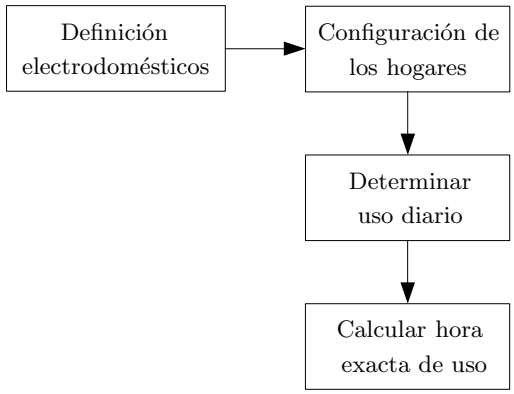


Figura 8: Estructura general del modelo de consumo energético residencial implementado.

4.1. Implementación del modelo

Se ha implementado el modelo de consumo energético residencial en Matlab con la estructura básica que se puede ver en la figura 8. Dicha estructura se corresponde con los procedimientos presentados en las Secciones 3.1, 3.2 y 3.3 en los cuales se utilizan las definiciones básicas de los electrodomésticos descritas en la Sección 2.3. En un primer paso, el modelo implementado lee las definiciones de los electrodomésticos considerados y configura los hogares usando esa información básica. En el segundo paso, el modelo determina cuantas veces se usan los diferentes aparatos de un hogar durante los días en cuestión. Finalmente el modelo calcula la hora exacta de utilización para cada electrodoméstico del hogar considerado.

Una hoja de cálculo contiene la información básica de cada uno de los electrodomésticos. Se pueden agregar fácilmente nuevos electrodomésticos añadiendo los datos básicos a la hoja de cálculo mencionada. El modelo ha sido implementado con la idea de ofrecer una gran flexibilidad en la generación de datos de consumo. Por ello, el modelo permite variar libremente el número de hogares y los períodos considerados en la generación de datos de consumo residencial.

4.2. Resultados de simulación

Con el modelo implementado se han realizado varias simulaciones con el objetivo de generar datos de consumo de los diferentes hogares. La figura 9 muestra la demanda diaria de un hogar tipo calculada a partir de datos de consumo durante un año. En los resultados se ve una diferencia considerable entre la demanda de los días laborables y de los días festivos. Se puede observar que la demanda de 5:00-8:00 es notablemente más elevada en días laborables que en días festivos. Por el contrario, la demanda residencial de 9:00-16:00 es mayor du-

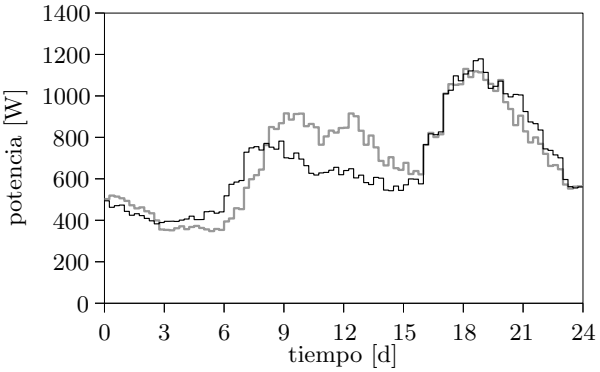


Figura 9: Promedio de la demanda energética diaria de un hogar tipo para días laborables (línea negra) y días festivos (línea gris).

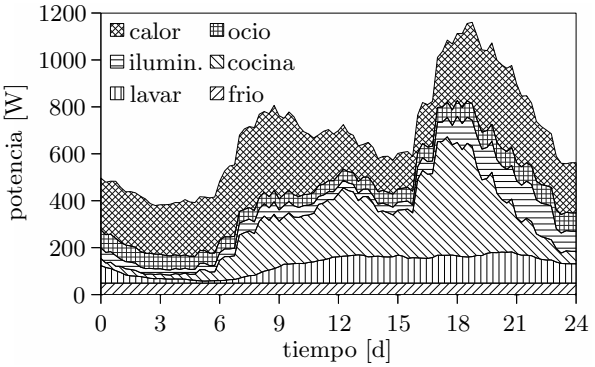


Figura 10: Composición del promedio de la demanda diaria de un hogar tipo.

rante los días festivos que en los días laborables.

Para el mismo hogar tipo, la figura 10 muestra la composición de la demanda diaria sin distinguir entre los días laborables y días festivos. Se puede observar que algunos grupos tienen una demanda más o menos constante (p.ej. equipos de frío o aparatos de ocio) mientras que otros grupos poseen un consumo variable a lo largo del día (p.ej. electrodomésticos de cocina o iluminación).

5. Conclusiones

En este artículo se ha presentado un novedoso modelo de consumo energético residencial para la generación automática de perfiles de demanda. El modelo está basado en un enfoque probabilístico y tiene una estructura modular con la idea de conseguir un alto grado de flexibilidad. El modelo considera 84 de los electrodomésticos más comunes y ofrece la posibilidad de añadir otros dispositivos.

En un primer paso, el modelo lee las definiciones básicas de los electrodomésticos considerados y genera las configuraciones de uno o varios hogares. En el segundo paso, el modelo determina con qué frecuencia se usa cada uno de los aparatos presentes en un hogar. Finalmente, el modelo calcula en el último paso la hora exacta de uso de los electrodomésticos y la demanda energética asociada. Hay que mencionar que el modelo permite omitir el primer paso y definir manualmente los hogares que se quieren considerar en la generación de perfiles de demanda. Además, se puede sustituir el cálculo de la hora exacta de uso de los aparatos por un algoritmo de gestión de la demanda con el fin de optimizar el consumo de energía.

El modelo de consumo energético residencial permite la generación de un gran número de hogares con la correspondiente demanda. Los datos generados pueden ser usados en el desarrollo de técnicas de control y gestión de demanda, tanto a nivel de hogar individual como a escala agregada con muchos consumidores residenciales. Para el futuro se prevé la validación del modelo propuesto mediante datos reales de diferentes distribuidores de energía.

Agradecimientos

Los autores agradecen el apoyo económico recibido de la empresa E.ON AG a través de su programa E.ON International Research Initiative.

Referencias

[1] Asare-Bediako, B., Ramirez Elizondo, L. M., Ribeiro, P. F., Kling, W. L., y Paap, G. C. (2011) Consideration of Electricity and Heat Load Profiles for Intelligent Management Systems. *Proceedings of the 46th International Universities' Power Engineering Conference*, Soest, Alemania, pp. 1–6.

[2] Carpaneto, E. y Chicco, G. (2007) Probabilistic characterisation of the aggregated residential load patterns. *IET Generation, Transmission & Distribution*, **2**, 373–382.

[3] Dickert, J. y Schegner, P. (2011) A time series probabilistic synthetic load curve model for residential customers. *Proceedings of the 2011 IEEE Trondheim PowerTech*, Trondheim, Noruega, pp. 1–6.

[4] Gruber, J. K. y Prodanovic, M. (2012) Residential energy load profile generation using a probabilistic approach. *Proceedings of the 2012 UKSim-AMSS 6th European Modelling Symposium*, La Valeta, Malta, pp. 296–301.

[5] Guerra Santin, O. (2011) Behavioural Patterns and User Profiles related to energy consumption for heating. *Energy and Buildings*, **43**, 2662–2672.

[6] Hobby, J. D. y Tucci, G. H. (2011) Analysis of the residential, commercial and industrial electricity consumption. *Proceedings of the 2011 IEEE PES Innovative Smart Grid Technologies Asia Conference*, Perth, Australia, pp. 1–7.

[7] McLoughlin, F., Duffy, A., y Conlon, M. (2012) Characterising domestic electricity consumption patterns by dwelling and occupant socio-economic variables: An Irish case study. *Energy and Buildings*, **48**, 240–248.

[8] Richardson, I., Thomson, M., Infield, D., y Clifford, C. (2010) Domestic electricity use: A high-resolution energy demand model. *Energy and Buildings*, **42**, 1878–1887.

[9] Tiptipakorn, S. y Lee, W.-J. (2007) A Residential Consumer-Centered Load Control Strategy in Real-Time Electricity Pricing Environment. *Proceedings of the 39th North American Power Symposium*, Las Cruces, EE.UU., pp. 1–6.

[10] Yao, R. and Steemers, K. (2012) A method of formulating energy load profile for domestic buildings in the UK. *Energy and Buildings*, **37**, 663–671.

[11] Zimmermann, J.-P., Evans, M., Griggs, J., King, N., Harding, L., Roberts, P., and Evans, C. (2012) Household Electricity Survey: A study of domestic electrical product usage. Informe técnico, Department for Environment, Food and Rural Affairs - Department of Energy and Climate Change - Energy Saving Trust.

PROGRAMACIÓN DE CONTROLADOR PREDICTIVO EN ECOSIMPRO® E IMPLEMENTACIÓN EN PLANTA DE DISTRIBUCIÓN DE OXÍGENO

D. Tejerina *, R. Martí *, D. Navia **, C. de Prada *

* Departamento de Ingeniería de Sistemas y Automática. Escuela de Ingenierías Industriales.
Universidad de Valladolid. C/Real de Burgos s/n. 47011 Valladolid.
e-mail: {david; ruben; prada}@autom.uva.es

** Departamento de Ingeniería Química y Ambiental, Universidad Técnica Federico Santa María.
Av. Vicuña Mackenna 3939, San Joaquín. Santiago, Chile.
e-mail: daniel.navia@usm.cl

Resumen

Este trabajo aborda la implantación del control predictivo a una planta piloto de distribución de oxígeno a tres unidades consumidoras, que simula el comportamiento de una red de distribución de hidrógeno en una refinería. Para ello, se programa el control predictivo basado en un modelo de conocimiento, obtenido mediante ecuaciones físicas, en EcosimPro® y se establece el esquema de comunicaciones necesario para la implantación del controlador a la planta, empleando servidores OPC creados a partir de la simulación del proceso y del controlador en EcosimPro® y el SCADA EDUSCA para la comunicación con la planta.

Palabras Clave: Control predictivo, optimización no lineal, comunicación en tiempo real, EcosimPro®, OPC, SCADA.

1 INTRODUCCIÓN

El control predictivo basado en modelos (*Model Predictive Control*) es un conjunto de métodos de control desarrollados en torno a determinadas ideas comunes:

- Uso explícito de un modelo para predecir el comportamiento futuro del proceso.
- Cálculo de la secuencia de control mediante la minimización de una función objetivo.
- Estrategia del horizonte móvil.

La estructura básica para implementar el control predictivo viene representada por la figura 1. Se emplea un modelo para predecir la evolución de la salida o estado del proceso a partir de las señales de entrada y salida conocidas. Las acciones de control futuras se calculan con el optimizador, que considera la función del coste y las posibles restricciones [4].

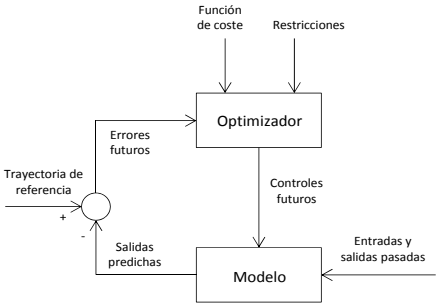


Figura 1: Estructura el MPC

La librería PREDICTIVE_CONTROL de Ecosim Pro® incluye los controladores lineales multivariables DMC y GPC con restricciones, válidos para procesos industriales continuos en condiciones de operación cercanas a las referencias estacionarias. Sin embargo, su aplicación en procesos batch o procesos continuos con cambios frecuentes en el set point es limitada.

Por ello, el objetivo de este artículo es mostrar a los usuarios de EcosimPro® los pasos a seguir para la programación de un controlador no lineal, en el que, una vez realizada su simulación, se aborda su aplicación al proceso identificado, gracias a la posibilidad que ofrece EcosimPro® de generar servidores OPC a partir de los códigos programados, para la comunicación del controlador con la planta vía OPC.

El artículo está organizado de la siguiente manera: En el apartado 2 se describe la planta de distribución de oxígeno; la identificación y validación del modelo se aborda en el apartado 3; la programación del MPC en el entorno de EcosimPro® se recoge en el apartado 4; y el esquema de comunicación del controlador con la planta en el 5 (simulación) y el 6 (proceso real). Los resultados obtenidos se muestran en el apartado 7 y las conclusiones en el 8.

2 DESCRIPCIÓN DE LA PLANTA

La planta piloto está compuesta por dos generadores de oxígeno que aportan flujos de oxígeno de distinta pureza. El generador G1 (de alta pureza) proporciona un flujo con un 62% en volumen de oxígeno, mientras que G2 (de media pureza) aporta oxígeno al 50% en volumen, operando ambos a una presión de trabajo de 0.3 barg, controlada en la purga. El rango de caudal suministrado por ambos generadores se establece entre 0.4 y 4.4 L/min.

Se dispone de dos colectores para la circulación de sendos flujos con tres ramas de aporte por cada colector a las unidades consumidoras, como se muestra en la figura 2.

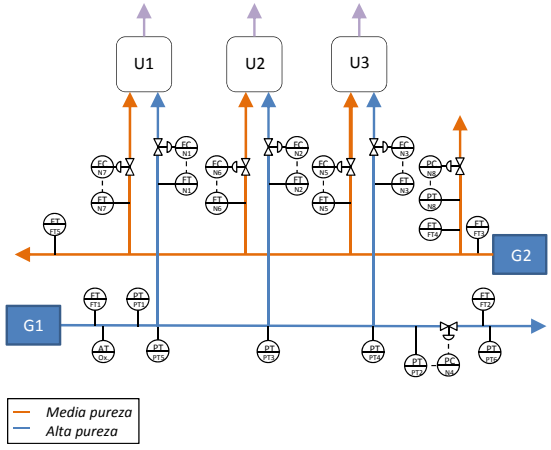


Figura 2: Diagrama de la planta

Las unidades consumidoras son depósitos de tanque agitado por los que circula un flujo continuo de agua impulsado por bombas peristálticas (figura 4), de tal modo que el consumo de oxígeno está basado en el fenómeno de difusión gas-líquido. La entrada de gas a la unidad se lleva a cabo a través de un difusor situado en la parte inferior, para facilitar la transferencia de materia.

Las variables medidas en las unidades consumidoras son la concentración de oxígeno disuelto (variable controlada) y la temperatura (perturbación medible), siendo los flujos de oxígeno en las ramas de aporte a cada unidad las variables manipuladas en el problema de control planteado.

La planta consta de un sistema de control distribuido, compuesto por un autómata programable (PLC) y un sistema de control y adquisición de datos (SCADA). La arquitectura del sistema de control que se ha implementado en la planta piloto se muestra en la figura 3.

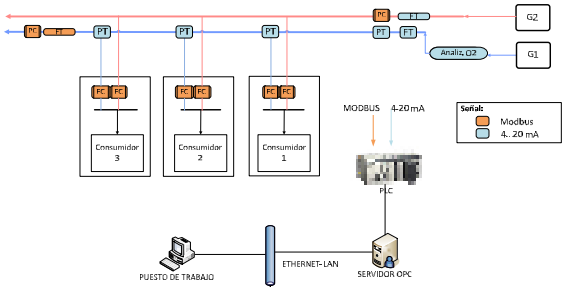


Figura 3: Arquitectura del sistema de control



Figura 4: Fotografía de la primera unidad consumidora

La nomenclatura empleada en el artículo se muestra en la tabla 1.

Nomenclatura		
C_{O2}	Concentración de O_2 disuelto	mg/l
C_{O2}^{sat}	Concentración de O_2 saturación	mg/l
C_{O2}^{in}	Concentración de O_2 entrada	mg/l
T	Temperatura	ºC
P	Presión	bar
F	Flujo de agua	L/min
F_{O2}	Flujo total de gas	L/min
F_{O2C1}	Flujo de gas del colector 1	L/min
F_{O2C2}	Flujo de gas del colector 2	L/min
V_L	Volumen de líquido	L
y_{O2}	Fracción molar de O_2	º/1
y_{O2C1}	Fracción molar de O_2 colector 1	º/1
y_{O2C2}	Fracción molar de O_2 colector 2	º/1
H	Constante de Henry	atm L/mg
k_la	Coeficiente de transferencia	1/min

Tabla 1: Nomenclatura

3 IDENTIFICACIÓN

Se identifica un modelo de primeros principios a partir del balance de materia individual al oxígeno en fase líquida:

$$V_L \cdot \frac{dC_{O2}}{dt} = k_l a \cdot V_L \cdot \left(C_{O2}^{sat} - C_{O2}\right) + F \cdot \left(C_{O2}^{in} - C_{O2}\right) \quad (1)$$

Siendo:

$$C_{O2}^{sat} = \left(\frac{P_{atm}}{H}\right) \cdot y_{O2} \quad (2)$$

$$C_{O2}^{in} = \left(\frac{P_{atm}}{H}\right) \cdot 0.21 \quad (3)$$

$$H = 5 \cdot 10^{-4} \cdot T + 1.315 \cdot 10^{-2} \quad (4)$$

$$k_l a = 0.7432 \cdot \left(T - 7.6670\right)^{0.5} \cdot F_{O2}^{\ 0.6822} \quad (5)$$

$$F_{O2} = F_{O2C1} + F_{O2C2} \quad (6)$$

$$F_{O2} \cdot y_{O2} = F_{O2C1} \cdot y_{O2C1} + F_{O2C2} \cdot y_{O2C2} \quad (7)$$

El rango de validez del modelo es:

- Flujo de gas: 0.4 - 4.4 L/min
- Temperatura: 11 - 17 ºC

La constante de Henry se calcula experimentalmente a partir de la ecuación 2 [8], midiendo la concentración en condiciones de saturación a distintas temperaturas y ajustando los valores alcanzados, obteniendo la expresión 4 (figura 5).

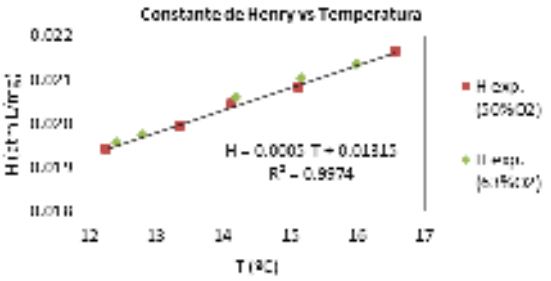


Figura 5: Constante de Henry vs Temperatura

En estado estacionario, el coeficiente de transferencia de materia ($k_l a$) [3] se despeja a partir de la expresión 1 en función de la concentración de oxígeno. De este modo, se identifica el coeficiente de transferencia en función del flujo de gas y de la temperatura, ajustando los valores alcanzados, empleando el comando *sftool* de Matlab [6], obteniendo la expresión 5 (figura 6).

Las ecuaciones 6 y 7 expresan el balance de materia global e individual al oxígeno, respectivamente.

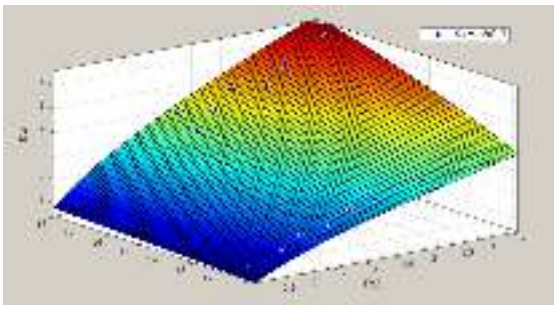


Figura 6: Ajuste del coeficiente de transferencia

Hasta este punto se ha realizado un análisis de los datos obtenidos en distintas condiciones de operación en régimen estacionario, sin embargo para llevar a cabo la identificación es necesario tener en cuenta la dinámica del sistema.

Se realiza la validación a una temperatura intermedia de 15 ºC, obteniendo un error cuadrático medio de 0.43 (figura 7).

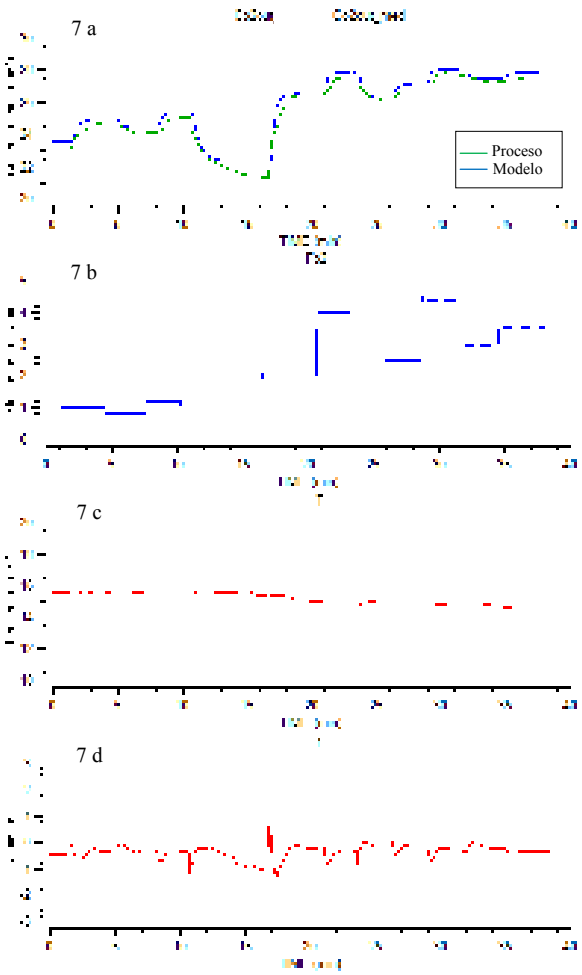


Figura 7: Validación del modelo. (7 a) Evolución de la variable controlada, (7 b) Flujo de gas, (7 c) Temperatura, (7 d) Residuos

4 PROGRAMACIÓN MPC

Para programar el controlador en EcosimPro®, se crea una librería con los archivos coste.el y control.el. El archivo coste.el, recibe los valores de las variables manipuladas (flujos en ambos colectores), la perturbación medible (temperatura) y la variable controlada (concentración de oxígeno) de control.el, calculando la función de coste (J). El archivo control.el recoge el valor de la función objetivo y lleva a cabo la optimización (librerías NAG® [7]), proponiendo las nuevas acciones de control a aplicar (u).

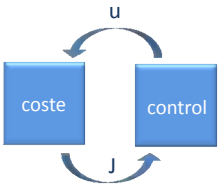


Figura 8: Esquema del MPC

La función de coste viene dada por la expresión:

$$\min_{u_1,u_2} J = \int_0^{t_f} \left(\left(y_{mod}(t) - y_{SP}(t) \right)^2 + \beta_1 \Delta u_1^2(t) + \beta_2 \Delta u_2^2(t) + \gamma \cdot \text{func} \right) \cdot dt$$

$$\text{func} = \max\left(u_1(t) + u_2(t) - F_{O2}^{max}, 0 \right)$$

Sujeto a:

Ecuaciones (1) - (7)

$$0.4 \le u_1(t) \le 4.4; \quad 0.4 \le u_2(t) \le 4.4$$

$$-0.1 \le \Delta u_1(t) \le 0.1; \quad -0.1 \le \Delta u_2(t) \le 0.1$$

Siendo:

y_{mod} = valor predicho por el modelo

$u_1 = F_{O2C1}$; $u_2 = F_{O2C2}$

En la función de coste se incluyen factores de penalización sobre las acciones de control β_1 y β_2, ambos iguales a 0.1, para suavizar la respuesta del sistema. Además, se incluye la restricción de no sobrepasar el flujo máximo en la propia función de coste, con un factor de peso elevado (γ = 500). De este modo, el algoritmo de optimización evita proponer acciones de control que impliquen que se exceda el máximo flujo (F_{O2}^{max} = 4.4 L/min), correspondiente al valor hasta el que se ha identificado el modelo.

Como restricciones del problema de optimización, se limita el flujo proveniente de ambos colectores entre 0.4 y 4.4 L/min y las acciones de control permitidas en cada periodo de muestreo ($\pm$ 0.1 L/min).

La sintonía del MPC viene dada por un horizonte de control N_u = 3 y un horizonte de predicción N_2 = 30. El parámetro α (factor de velocidad de la trayectoria de referencia) se selecciona igual a 0.3. El tiempo de muestreo elegido es de 12 segundos, adecuado para captar la dinámica del sistema.

Sintonía del controlador predictivo	
Horizonte de control	$N_u = 3$
Horizonte de predicción	$N_2 = 30$
Factor de velocidad de la trayectoria de referencia	$\alpha = 0.3$

Tabla 2. Parámetros de sintonía del controlador

La simulación del controlador en EcosimPro® dando varios saltos en el set point, tomando como premisa inicial que el modelo es idéntico al proceso (sin incluir errores de modelado) se muestra en las figuras 9 a y 9 b.

La grafica 9 a incluye los límites superior e inferior relativos a la concentración de saturación y la alcanzada con flujo mínimo, respectivamente.

Se puede observar una respuesta sobreamortiguada, alcanzándose las referencias en tiempos que varían entre 4 y 6 minutos.

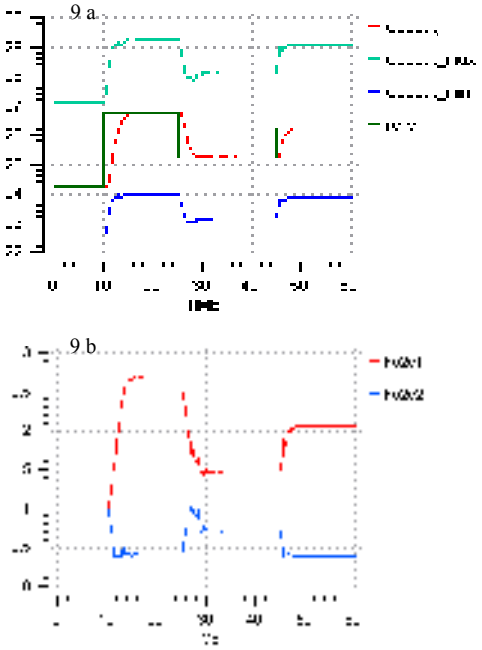


Figura 9: Simulación del controlador en EcosimPro. (a) Variable controlada, (b) Variables manipuladas

2.1 CORRECCIÓN DEL ERROR

No obstante, los modelos siempre incluyen errores, lo que conduce a que el proceso sea ligeramente distinto al modelo identificado. Por ello es necesario introducir un término de corrección del error entre el valor predicho por el modelo y el valor real medido. Dicho error puede sumarse tras la integración de la

función de coste, o bien en el punto inicial de integración. Para comprobar la capacidad de corrección en ambos casos, se provoca un error de modelado, multiplicando la expresión del coeficiente de materia ajustado (expresión 5) por 1.10 (10% de desviación).

Sin introducir término de corrección del error:

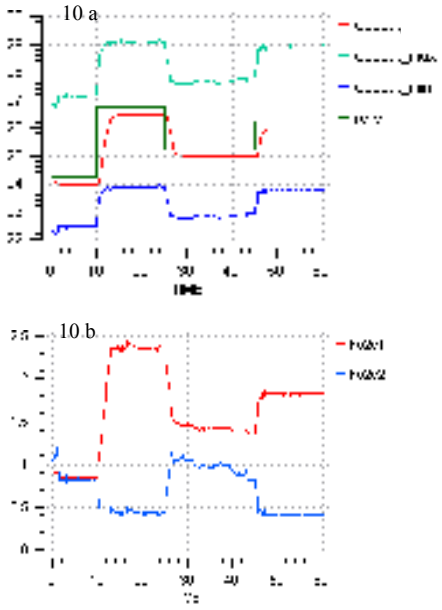


Figura 10: Introducción de un error de modelado, sin corrección del error. (a) Variable controlada, (b) Variables manipuladas

Incluyendo un término de corrección del error:

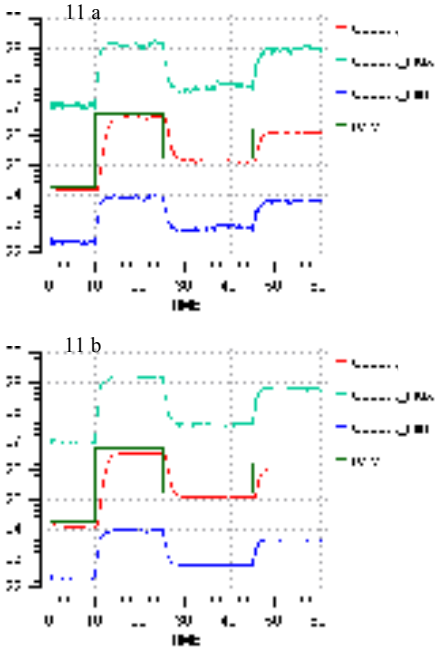


Figura 11: Corrección del error tras la integración (11 a) y en el punto inicial (11 b)

Como puede verse en la figura 10 a, sin corrección del error, los errores estacionarios son en torno al 1%, alcanzándose una concentración en torno a 0.30 mg/l por debajo del set point.

Sumando la corrección del error al valor predicho por el modelo tras la integración de la función de coste (figura 11 a), el error estacionario se reduce en torno a un 50%, mientras que llevando a cabo la corrección en el punto inicial (figura 11 b), se consigue una corrección cercana al 35%. No obstante, el error no se consigue eliminar completamente.

La estrategia para eliminar el error estacionario consiste en plantear un problema de optimización, paralelo al del control predictivo, que minimice el error entre la concentración predicha por el modelo (y_{mod}) y la concentración real medida del proceso (y_{real}) a lo largo de un horizonte móvil (se selecciona un horizonte igual a 5), modificando un parámetro de perturbación del modelo como variable de decisión (d) cada periodo de muestreo (*Moving Horizon Estimation*). Dicho parámetro se multiplica a la ecuación ajustada de k_la:

$$k_la = d \cdot \left\{ m \cdot (T - n)^{0.5} \cdot F_{O2}{}^{a} \right\} \tag{8}$$

$$\min_{d} \sum_{k=1}^{k=N-1} \left(y_{mod}(t_k) - y_{proceso}(t_k) \right)^2 \tag{9}$$

s.a :

Ecuaciones: (1) – (8)

De esta forma, se dispone de un grado de libertad más, permitiendo al modelo ajustarse al proceso real a través de este parámetro. En la figura 12 se muestra la simulación de una perturbación del proceso (modificando el parámetro *m* de la ecuación 8) en el minuto 5, mostrándose cómo la perturbación no medible (d) evoluciona para rechazar la perturbación:

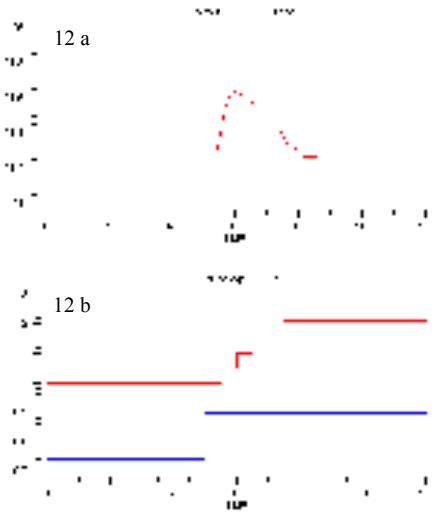


Figura 12: Aplicación MHE. (12 a) Variable controlada. (12 b) Estimación de la perturbación

5 ESQUEMA DE COMUNICACIÓN ECOSIMPRO® - OPC - SCADA

La comunicación entre el controlador y la planta se lleva a cabo vía OPC, gracias a la posibilidad de generar tales servidores a partir de un experimento en EcosimPro®, con el programa CreaOPC 4.8 [1] y configurando el SCADA EDUSCA [2] para el intercambio de variables con el proceso.

OPC (*OLE for Process Control*) es un estándar de comunicación en el campo del control y supervisión de procesos industriales, basado en la tecnología Microsoft OLE/COM/DCOM [9]. Ofrece una interfaz común para comunicación que permite que componentes software individuales interaccionen y compartan datos a través de una arquitectura cliente-servidor.

En primer lugar, se realiza la simulación del control predictivo aplicándolo al modelo de la planta, antes de implementarlo en el proceso real. Para ello se genera un servidor OPC del modelo y otro del controlador, configurando el SCADA para la comunicación entre variables, según el esquema que se muestra en la figura 13.

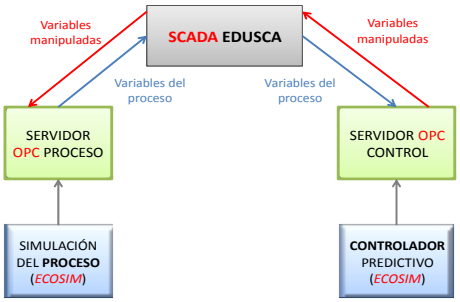


Figura 13: Esquema de comunicaciones

Es necesario sincronizar la lectura de las variables del proceso y la ejecución de las acciones de control. Para ello se emplea una variable entera denominada *activate*. Las acciones a realizar en función del valor de esta variable se especifican en el bloque DISCRETE del código del controlador, mediante el uso de la condición WHEN:

- Cada periodo de muestreo, la variable *activate* toma el valor 0.
- Cuando es igual a 1, se pasa al controlador el estado actual de las variables del proceso, asignando los valores a las variables de intercambio. Después la variable *activate* pasa a 2.
- Cuando toma el valor 3, el controlador calcula la nueva acción de control y el valor de *activate* pasa a 4.

Posteriormente, en la configuración del SCADA, se define el siguiente grafcet para transición entre etapas:

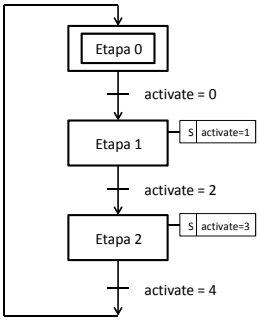


Figura 14: Grafcet para transición entre etapas

- Etapa 0: Espera.
- Etapa 1: El controlador lee las variables actuales vía OPC.
- Etapa 2: El controlador ejecuta las acciones de control.

Una vez guardada la configuración, se ejecuta el SCADA, comprobando que ante los mismos cambios en la referencia de concentración, se produce la misma evolución en las variables manipuladas, alcanzándose el set point de igual manera que en el experimento de EcosimPro®.

6 IMPLANTACIÓN DEL MPC EN PLANTA REAL

En la configuración del SCADA, se sustituyen las referencias al modelo por los valores leídos del proceso. Así, por ejemplo, en las variables de intercambio de los flujos de los colectores, en lugar de tener como variable de salida la correspondiente al modelo, se apunta al valor medido de la planta (figura 16).

Del mismo modo, en el caso de la variable controlada, se toma la medida de la sonda de oxígeno de la planta real (Variable Entrada), no de la simulación del proceso (figura 17).

El esquema de comunicaciones es, por tanto:

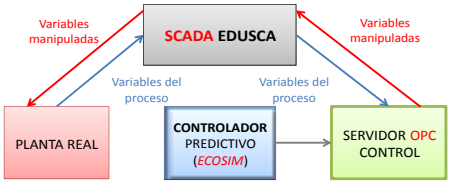


Figura 15: Esquema de comunicaciones del controlador con la planta



Figura 16: Variable de intercambio para el flujo del primer colector



Figura 17: Variable de intercambio para la concentración de oxígeno

7 RESULTADOS Y DISCUSIONES

Como se ha comentado, el objetivo es el control de la concentración de oxígeno disuelto, actuando sobre el flujo de los colectores. Como perturbación medible, inicialmente se planteó únicamente la temperatura en la unidad consumidora, sin embargo, se observó que la pureza de oxígeno en los colectores variaba con el flujo demandado, siendo dicha perturbación mucho más significativa en el segundo colector (entre 45 y 52%), ya que en el primero se mantiene entre 62 y 63% en el rango de flujo de operación. Dado que solo se dispone de un analizador de O_2, se toma como perturbación medible la pureza del segundo colector (de media pureza).

En la figura 19 se presenta un diagrama de aplicación del controlador a una de las unidades consumidoras y en la figura 20 el sinóptico del SCADA para la interacción del usuario con la planta.

La evolución de la variable controlada (figura 18 a), las variables manipuladas (18 b), las perturbaciones medidas (18 c) y la estimación d (18 d) ante varios saltos en la referencia se muestra a continuación.

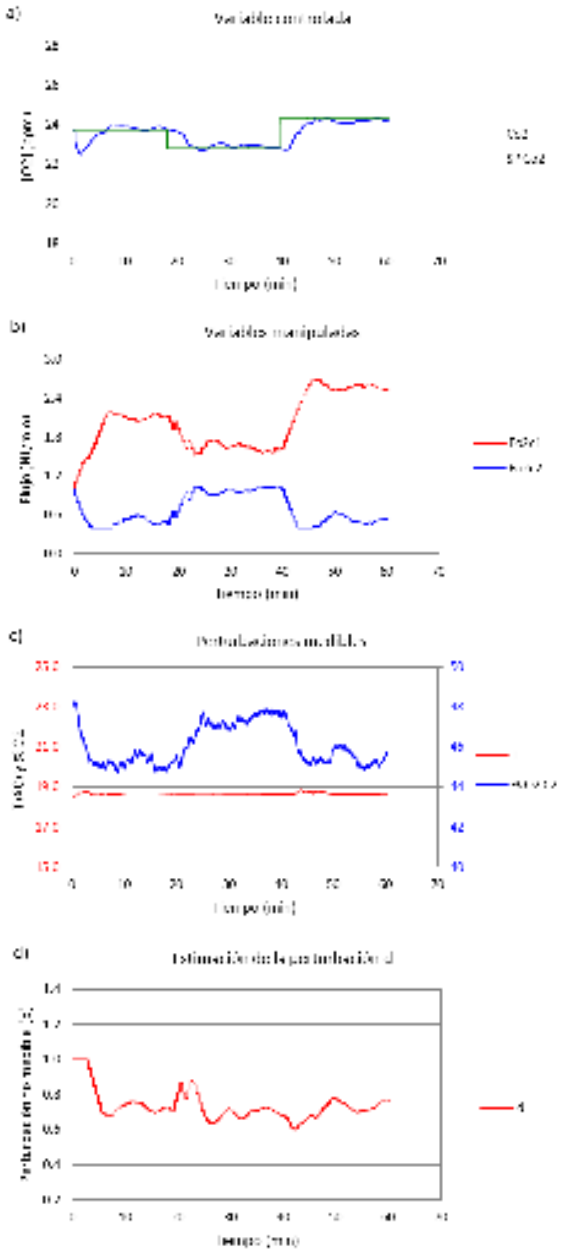


Figura 18: Aplicación del MPC a la planta. (a) Evolución de la variable controlada ante saltos en la referencia, (b) evolución de las variables manipuladas, (c) Perturbaciones medibles, (d) Estimación del parámetro de perturbación

Se observa una respuesta muy similar a la obtenida en la simulación, con tiempos de asentamiento cercanos a 10 minutos. Las pequeñas fluctuaciones de la variable controlada (± 0.2 ppm) en los estados estacionarios pueden deberse a perturbaciones del proceso, como la posible variación de volumen de agua en la unidad consumidora debido al efecto de las burbujas de oxígeno en el seno del líquido.

Inicialmente, se observa que el parámetro d comienza con valor 1, variando a medida que avanza el tiempo

hasta que la variable controlada alcanza la referencia, momento en que la estimación del parámetro d varía en un intervalo más estrecho.

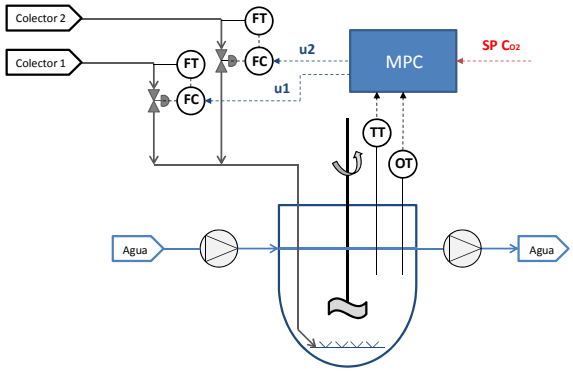


Figura 19: Diagrama de aplicación del MPC



Figura 20: Sinóptico del SCADA

8 CONCLUSIONES

Se han mostrado los pasos a seguir para llevar a cabo la implantación de un controlador predictivo no lineal programado en el entorno de EcosimPro®, estableciendo la comunicación en tiempo real con el proceso vía OPC y configurando el SCADA para la interacción entre el usuario y la planta. Con ello, se propone una nueva funcionalidad de EcosimPro® que, además de simular el comportamiento de un modelo, es capaz de llevar a cabo el control y la supervisión de un proceso real mediante la creación de un servidor OPC a partir del código programado, herramienta que ya incluye la versión de Ecosim 5.0.

Agradecimientos

Los autores desean agradecer al Ministerio de Ciencia y Tecnología la financiación a través del proyecto "Control y optimización de redes de distribución", CICYT (ref. 18ISAZ99) y al Ministerio de Educación la beca de investigación en el Departamento de Ingeniería de Sistemas y Automática, Universidad de Valladolid.

Referencias

[1] Acebes, L. F., Alves, R., Merino, A., (2004), Prada, C., (2004) "Un entorno de modelado inteligente y simulación distribuida de plantas de proceso", *Revista Iberoamericana de Automática e Informática Industrial. Vol. 1, Núm. 2.*

[2] Alves, R., Normey-Rico, J., Merino, A., Acebes, L.F., Prada, C., (2006), EDUSCA (Educational Scada): Features and applications, *7th IFAC Symposium in Advances in Control Education*.

[3] Bird, R., Stewart, W., Lightfoot, E., (2002), "Transport Phenomena", 2nd Edition, John Wiley & Sons.

[4] Camacho, F., Bordons, C., (2004), "Control predictivo: pasado, presente y futuro". *2004 CEA-IFAC.*

[5] EcosimPro by EA Internacional, Dynamic Modeling & Simulation Tool, [Online] http://www.ecosimpro.com/ Último acceso: Mayo de 2013.

[6] MathWorks, Curve Fitting Tool [On line] http://www.mathworks.es/es/help/curvefit/sftool.html Último acceso: Mayo de 2013.

[7] NAG, "Numerical Algorithms Group" [On line] http://www.nag.co.uk/ Último acceso: Mayo de 2013.

[8] Perry, R., Green, D., (1997), "Perry's Chemical Engineers' Handbook", 7th Edition, 1997, Ed. McGraw-Hill.

[9] Zamarreño, J. M. (2001). "Entorno de comunicaciones OPC para Ecosim". *Primeras Jornadas de Usuarios de EcosimPro. Madrid, 2001.*

PREDICTOR NO PARAMÉTRICO BASADO EN TÉCNICAS DE ERROR ACOTADO

J.M. Bravo, A. Suárez y M. Vasallo
Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática, Universidad de Huelva
email:{caro,suarez,manuel.vasallo}@uhu.es
T. Alamo
Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla
email:teodoroalamo@gmail.com

Resumen

En esta comunicación se presenta un nuevo método de predicción intervalar aplicable a sistemas dinámicos. Dado un conjunto de medidas de entrada y salida del sistema, una cota máxima de error permitido y una aproximación local del comportamiento del sistema, el predictor proporciona una predicción intervalar que acota externamente la salida del sistema en cada instante de tiempo. Además, el punto medio de la predicción intervalar se puede utilizar como una predicción nominal de la salida ya que cumple ciertos criterios de optimalidad. La anchura del intervalo de predicción refleja la precisión o fiabilidad de la predicción nominal. La principal contribución del presente artículo es una expresión explícita que proporciona la predicción intervalar y asegura un bajo coste computacional. Finalmente se incluye un ejemplo que ilustra el método presentado incluyendo una comparación con otro método similar presente en al literatura.

Palabras clave: Predicción, identificación de sistemas, métodos no paramétricos.

1 INTRODUCCIÓN

Normalmente, con objeto de predecir la salida de un sistema dinámico se suele utilizar un modelo matemático. Dicho modelo se puede obtener a partir de un conocimiento exhaustivo de los principios físicos que rigen el sistema o bien utilizando técnicas de identificación. Los métodos de identificación de sistemas más extendidos están basados en modelos paramétricos del sistema [9, 15]. Estos métodos asumen que el comportamiento del sistema se puede aproximar mediante una familia de funciones obtenida parametrizando cierta estructura de modelo. Con un conjunto de medidas de entrada y salida del sistema, se formula un problema de optimización que busca minimizar el error de predicción cometido por el modelo. La solución a dicho problema de optimización proporciona los parámetros adecuados y por tanto el modelo buscado.

Otra posibilidad es considerar cierto error aditivo en las medidas obtenidas del sistema. Dicho error se asume acotado, es decir, que es menor a cierto umbral prefijado y puede representar dinámica no modelada, ruido en las medidas o cierta incertidumbre aditiva. El objetivo de los métodos de identificación basados en error acotado [11],[1] es obtener el conjunto de parámetros que es consistente con el conjunto de medidas disponibles, la estructura de modelo usada y la cota de error considerada. A este conjunto se le conoce con el nombre de *Conjunto Solución Factible* y se le denota como CSF.

Cuando un método de identificación basada en error acotado se aplica a un modelo paramétrico lineal, el CSF se puede calcular de forma exacta, y se puede representar mediante un politopo descrito mediante sus caras [5] o vértices [13]. Sin embargo, estos métodos tienen la desventaja de que la complejidad del politopo se incrementa con cada nueva medida. Con el fin de solventar esta dificultad, en la literatura se ha propuesto la utilización de acotaciones externas del CSF. A dichos conjuntos de parámetros se les denomina *Conjuntos Solución Factibles Aproximados* (los denotaremos $CSFA$) y suelen caracterizarse por tener una complejidad limitada. Cajas [10, 7], elipsoides [6, 2], paralelotopos [14] y zonotopos [3] son ejemplos de este tipo de conjuntos. El caso de modelos no lineales en los parámetros también se ha tratado en la literatura (ver [8][4]).

En este artículo se propone una vía diferente a lo expuesto hasta ahora. El método de predicción propuesto no está basado en un modelo paramétrico del sistema. Cada vez que se requiere una predicción, se utiliza el conjunto de medidas disponibles y técnicas de error acotado para obtener una aproximación local del comportamiento del sistema. De esta forma es posible proporcionar una predicción intervalar compuesta por una cota superior e inferior de la salida del sistema. Dicha predicción intervalar se calcula mediante una expresión explícita que utiliza las medidas y las cotas consideradas. Además, el centro de dicha predicción intervalar se puede utilizar como una predicción nominal, ya que coincide

con la predicción obtenida al minimizar el error de predicción de cierto modelo local lineal.

Un predictor intervalar no paramétrico basado en una cota del gradiente de la función que representa al sistema se presentó en [12]. El presente trabajo se diferencia de [12] en las hipótesis de trabajo. En el trabajo que se presenta se asume que el comportamiento del sistema en un punto dado de trabajo se puede aproximar mediante una función afín que se puede calcular mediante los métodos de error acotado antes mencionados. En la sección de ejemplos se incluye una comparación con [12] con el fin de valorar las bondades del nuevo predictor.

2 FORMULACION DEL PROBLEMA

Considérese el siguiente sistema no lineal discreto

$$y_k = f_0(x_k) + e_k \tag{1}$$

donde k es el instante de tiempo discreto, $y_k \in \mathbb{R}$ es la salida del sistema y $x_k \in X \subseteq \mathbb{R}^{n_x}$ son las entradas y salidas pasadas del sistema $x_k = [y_{k-1}, y_{k-2}, ..., y_{k-\mu_y}, u_k, u_{k-1}, ..., u_{k-\mu_u}]$. Notar que $n_x = \mu_y + \mu_u + 1$. Además, X es un conjunto compacto y e_k es un término de error. Este error está acotado, es decir se tiene que $|e_k| \leq \sigma \quad \forall k$ donde $\sigma \in \mathbb{R}^+$ es una constante positiva. Se asume que $f_0(\cdot)$ es desconocido. Sin embargo, se dispone de un conjunto de medidas del sistema denotado $\mathcal{D}$, tal y como se describe en la Hipótesis 1.

Hipótesis 1 *Se dispone de un conjunto de medidas* $\mathcal{D} = \{x_i, y_i\}$ *con* $i = 1, ..., N$.

En los métodos de identificación paramétrica se asume que $f_0(\cdot)$ pertenece a una familia parametrizada de funciones, donde los parámetros son los grados de libertad considerados. La identificación consiste básicamente en estimar dichos parámetros. Sin embargo la propia elección de la familia de funciones ya supone cierto problema al no existir un método sistemático. Con objeto de eludir dicho problema, en esta comunicación se asume una técnica no paramétrica. No se trabaja con la hipótesis de pertenencia a una familia, sino que se asume que la función $f_0(\cdot)$ se puede aproximar localmente mediante una función afín que cumple ciertas condiciones. A continuación se formaliza este concepto.

Hipótesis 2 *Para cualquier vector* $x_k \in X$ *existe* $\theta_k \in \Theta(x_k)$ *tal que* $f_0(x_k) = [x_k^T \ 1]\theta_k$, *siendo* $\Theta(x_k)$ *el conjunto descrito en la Definición 1.*

El conjunto $\Theta(x_k)$ constituye el CSF asociado al vector x_k. Con el fin de limitar los grados de libertad, es necesario acotar el tamaño del conjunto $\Theta(x_k)$. Para ello se utilizan las constantes σ y $L \in \mathbb{R}^+$. A continuación se presenta una definición formal del conjunto $\Theta(x_k)$.

Definición 1 *Para cualquier vector* $x_k \in X$, *el Conjunto Solución Factible* $\Theta(x_k)$ *se define mediante* $\Theta(x_k) = \{ \ \theta \in \mathbb{R}^{n_g} \ : \ |y_{s_j} - x_{s_j}^\top \theta| \leq \sigma + L||x_{s_j} - x_k|| \ con \ (y_{s_j}, x_{s_j}) \in \mathcal{D}(x_k)\}$ *donde* σ *y* $L \in \mathbb{R}^+$ *son constantes positivas, y* $\mathcal{D}(x_k) = \{(y_{s_j}, x_{s_j})\}$ *con* $j = 1, 2, ..., N_S$ *es un subconjunto de* $\mathcal{D}$, *siendo* $\mathcal{D}(x_k) \subseteq \mathcal{D}$.

Como ya se ha comentado antes, σ limita el máximo error considerado en las medidas. Por otra parte, la constante L acota la máxima capacidad de cambio de la función $f_0(\cdot)$. Una consecuencia práctica es que cuanto mayor es la L considerada mayor es el gradiente permitido para $f_0(\cdot)$.

A la hora de definir el predictor intervalar sería útil disponer de una descripción matricial del conjunto $\Theta(x_k)$. Dicha forma matricial se obtiene mediante las expresiones $\Theta(x_k) = \{ \ \theta \in \mathbb{R}^{n_g} \ : A_S\theta \leq b_S\}$ con la matriz $A_S = [-A_R^T \ \ A_R^T]^T$, el vector $b_S = [b_{L_R}^T \ \ b_{L_R}^T]^T + [-b_{Y_R}^T \ \ b_{Y_R}^T]^T$ y

$$A_R^T = \left[\ [x_{s_1}^T \ 1]^T \quad [x_{s_2}^T \ 1]^T \quad ... \quad [x_{s_{N_s}}^T \ 1]^T \ \right] \tag{2}$$

$$b_{L_R} = \begin{bmatrix} \sigma + L||x_k - x_{s_1}|| \\ \sigma + L||x_k - x_{s_2}|| \\ \vdots \\ \sigma + L||x_k - x_{s_{N_s}}|| \end{bmatrix} \tag{3}$$

$$b_{Y_R} = \left[\ y_{s_1} \quad y_{s_2} \quad ... \quad y_{s_{N_s}} \ \right]^T. \tag{4}$$

Una vez contextualizado el problema abordado, es hora de fijar los objetivos del presente trabajo. Un predictor intervalar es un operador que dado un vector $x_k \in X$ proporciona una estimación de los valores máximos y mínimos correspondientes a la salida del sistema. El objetivo de este artículo es proponer un nuevo método de predicción intervalar. A continuación se enumeran las principales características del predictor propuesto.

1. El predictor está basado en un método no paramétrico, por lo que no es necesario considerar una familia parametrizada de modelos.

2. El predictor proporciona una predicción intervalar de la salida del sistema. La anchura de dicha predicción intervalar esta relacionada inversamente con la fiabilidad y precisión conseguidas por el método. Un intervalo estrecho significa alta fiabilidad, mientras que un intervalo ancho implica falta de

información para poder realizar la predicción y, por tanto, menor fiabilidad. Dicha predicción o fiabilidad depende de la cantidad de información que sobre el vector x_k se tiene en el conjunto $\mathcal{D}$.

3. Utilizando el punto central del intervalo proporcionado por el predictor, es posible obtener una predicción nominal de la salida del sistema. Esta predicción central es la solución óptima de un problema de optimización que busca minimizar de forma cuadrática el error de predicción.

4. La predicción intervalar se obtiene de forma eficiente mediante una expresión explícita. No se necesita utilizar un método de búsqueda iterativo.

3 Sistema predictor propuesto

A continuación se aborda la principal aportación de este trabajo, se define el predictor intervalar $f_\lambda(x_k)$ y se indica cómo calcularlo.

Definición 2 *Dado un vector $x_k \in X$, un conjunto de medidas $\mathcal{D}$ y las constantes σ y L, se define el predictor intervalar $f_\lambda(x_k) = [\underline{f}_\lambda(x_k), \overline{f}_\lambda(x_k)]$ mediante las expresiones:*

$$\underline{f}_\lambda(x_k) = -|\lambda|^T b_{L_R} + \lambda^T b_{Y_R} \quad (5)$$

$$\overline{f}_\lambda(x_k) = |\lambda|^T b_{L_R} + \lambda^T b_{Y_R} \quad (6)$$

$$\lambda = W^{-1} A_R (A_R^T W^{-1} A_R)^{-1} [x_k^T \ 1]^T \quad (7)$$

donde $|\lambda|$ denota una vector compuesto por el valor absoluto de los elementos del vector λ. La matriz W es una matriz diagonal. Los elementos de la diagonal son los elementos del vector b_{L_R} elevados al cuadrado.

El predictor $f_\lambda(x_k)$ se compone de dos términos. El elemento $\lambda^T b_{Y_R}$ es una combinación lineal de las salidas del sistema y es compartido por $\underline{f}_\lambda(x_k)$ y $\overline{f}_\lambda(x_k)$. Los término $-|\lambda|^T b_{L_R}$ y $|\lambda|^T b_{L_R}$ son las cotas intervalares respecto el centro $\lambda^T b_{Y_R}$. Es importante señalar que $|\lambda|^T b_{L_R}$ no depende de las salidas del sistema. Este término es función de las constantes σ y L y de la distancia del vector x_k a los vectores $x_i \in \mathcal{D}(x_k)$. Cuanto mayor sea esa distancia, mayor es la anchura de la predicción intervalar pues el método de predicción considera que menor es la cantidad de información sobre x_k incluida en $\mathcal{D}(x_k)$.

El siguiente teorema muestra que el predictor $f_\lambda(x_k)$ proporciona una cota externa de la salida del sistema (1) siempre que las Hipótesis 1 y 2 sean ciertas.

Teorema 1 *Considérese el predictor $f_\lambda(x_k)$ propuesto en la Definición 2. Entonces se cumple que: $\underline{f}_\lambda(x_k) \leq f_0(x_k) \leq \overline{f}_\lambda(x_k) \ \forall x_k \in X$.*

Una prueba del teorema se puede encontrar en el Apéndice. La prueba se basa en explotar las simetrías de la matriz $A_S = [-A_R^T \ \ A_R^T]^T$ y el vector $b_S = [b_{L_R}^T \ \ b_{L_R}^T]^T + [-b_{Y_R}^T \ \ b_{Y_R}^T]^T$.

3.1 Predicción nominal

Como ya se ha comentado anteriormente, el punto central de la predicción intervalar $f_\lambda(x_k)$ coincide con la solución a un problema de mínimos cuadrados formulado para minimizar el error de predicción cometido por un modelo local lineal. En esta sección se muestra dicha propiedad y se analiza sus bondades. A continuación se define el concepto de predicción central.

Definición 3 *Una predicción central $f_\lambda^c(x_k)$ de la predicción intervalar $f_\lambda(x_k)$ se obtiene mediante la expresión $f_\lambda^c(x_k) = \frac{1}{2}(\underline{f}_\lambda(x_k) + \overline{f}_\lambda(x_k)) = \lambda^T b_{Y_R}$.*

Dado el conjunto de datos $\mathcal{D} = \{(y_i, x_i)\}$ con $i = 1, ..., N$, en los métodos de identificación paramétricos se utiliza el término de error $\epsilon_i = y_i - [x_i^T \ 1]\theta$ para obtener el vector de parámetros óptimo θ^*. Con este fin se suele definir una función de coste que penaliza de forma cuadrática el error de predicción. Una posible opción es considerar un modelo local y lineal para obtener la predicción. Una función de coste que cumple dichas premisas podría ser:

$$\begin{aligned} J(\theta) = & \ (b_{Y_R} - A_R\theta)^T W^{-1} (b_{Y_R} - A_R\theta) \\ = & \ \sum_{i=1}^{N_s} \frac{(y_{s_i} - [x_{s_i}^T \ 1]^T\theta)^2}{(\sigma + L||x_{s_i} - x_k||)^2} \end{aligned} \quad (8)$$

Es importante señalar que los términos $\sigma + L||x_{s_i} - x_k||$ constituyen pesos que reflejan la importancia relativa de la correspondiente medida i. Como ya se ha comentado, una estimación del vector de parámetros θ^* se puede obtener minimizando la función (8) (ver [9]).

A continuación se presenta un importante propiedad del punto medio de la predicción intervalar propuesta en este trabajo. La siguiente propiedad muestra que la estimación central $f_\lambda^c(x_k)$ y la estimación basada en mínimos cuadrados $[x_k^T \ 1]\theta^*$ son equivalentes.

Propiedad 1 *Sea θ^* el vector de parámetros que minimiza la función (8),*

$$\theta^* = \min_\theta \ J(\theta) \quad (9)$$

entonces, se cumple la siguiente igualdad:

$$f_\lambda^c(x_k) = [x_k^T\ 1]\theta^*$$

PRUEBA : Derivando $J(\theta)$ respecto θ, igualando a cero y despejando θ, se obtiene que el mínimo se alcanza en

$$\theta^* = (A_R^T W^{-1} A_R)^{-1} A_R^T W^{-1} b_{Y_R}.$$

La predicción con ese mínimo sería

$$[x_k^T\ 1]\theta^* = [x_k^T\ 1](A_R^T W^{-1} A_R)^{-1} A_R^T W^{-1} b_{Y_R} =$$

$$\lambda^T b_{Y_R} = f_\lambda^c(x_k)$$

de donde se deduce que $f_\lambda^c(x_k) = [x_k^T\ 1]^T\theta^*$. ■

Es importante señalar que utilizando la Definición 3 no es necesario resolver el problema de optimización (9) para obtener la solución óptima, debido a la equivalencia demostrada en la Propiedad 1. Este punto queda reflejado en el paso 4 del algoritmo de predicción propuesto.

3.2 Ajuste de los parámetros σ y L

El primer paso para poder aplicar el predictor intervalar propuesto a un sistema concreto es definir los valores σ y L. Con ese fin, en este artículo se propone el Algoritmo 1. Dicho algoritmo utiliza validación cruzada en el conjunto $\mathcal{D}$ con el fin de obtener de forma empírica una estimación heurística de los valores apropiados para σ y L.

Básicamente el algoritmo define un conjunto de valores σ_j con $j = 0, 1, ..., j_{max}$ en los que realizar la búsqueda. Para cada valor σ_j el algoritmo debe calcular el valor mínimo para la constante L_j que admita la validación cruzada. Esto es, para cada pareja (y_i, x_i) perteneciente a $\mathcal{D}$, se utiliza el resto de parejas incluidas en $\mathcal{D}$ y las constantes σ_j y L_j para obtener la predicción intervalar $f_\lambda^j(x_i)$. La pareja σ_j y L_j es consistente si se cumplen las desigualdades $\underline{f}_\lambda^j(x_i) \le y_i \le \overline{f}_\lambda^j(x_i)$ con $i = 1, ..., N$. Posteriormente, el algoritmo devuelve los valores de σ_j y L_j que minimizan la métrica

$$INT_j^D = \sum_{i=1}^{N} \overline{f}_\lambda^j(x_i) - \underline{f}_\lambda^j(x_i).$$

Por tanto se tiene que $\sigma = \sigma_{j^*}$ y $L = L_{j^*}$ con

$$j^* = arg\min_j\ INT_j^D$$

El Algoritmo 1 utiliza los siguientes parámetros de entrada. Las constantes σ_0 y σ_{max} son los límites inferior y superior considerados para la constante σ. El valor ϵ_σ se utiliza para discretizar el espacio de búsqueda en σ. Por tanto, el Algoritmo 1 chequea todos los valores definidos por $\sigma_j = \sigma_0 + j\epsilon_\sigma$ con $j = 0, 1, ..., \epsilon_\sigma^{-1}(\sigma_{max} - \sigma_0)$. La constante L_0 es un valor inicial para L. Se utiliza la entrada ϵ_L para incrementar el valor de L en cada iteración. El conjunto $\mathcal{D}$ es el definido en la Hipótesis 1.

En la iteración i del algoritmo propuesto, se selecciona la pareja (y_i, x_i) de $\mathcal{D}$ y se calcula el conjunto de indices correspondiente a los N_S elementos más cercanos a x_i: $S(x_i) = \{s_1, ..., s_{N_S}\}$ tal que $||x_{s_j} - x_i|| \le ||x_{s_{j+1}} - x_i|| \le ||x_l - x_i||$ donde $x_{s_j} \ne x_i$, $l \notin S(x_i)$ y $j = 1, ..., N_S - 1$. Utilizando $S(x_i)$, σ_j, L_j y $\mathcal{D}$, se calculan las cotas $\underline{f}_\lambda^j(x_i)$ y $\overline{f}_\lambda^j(x_i)$. Si $y_i \notin [\underline{f}_\lambda^j(x_i), \overline{f}_\lambda^j(x_i)]$ se incrementa el valor de L_j y se repite la iteración i. Notar que $N_S \le N - 1$ es una constante positiva definida por el usuario que define el número de pares de $\mathcal{D}$ usado para obtener las cotas de la predicción intervalar.

Algoritmo 1

(σ, L)=*ObtenerParametros*$(\sigma_0, \sigma_{max}, \epsilon_\sigma, L_0, \epsilon_L, \mathcal{D})$
Alg
  $\sigma = \sigma_0$
  $j = 0$
  while $\sigma \le \sigma_{max}$
    $L = L_0$, $i = 1$, $INT = 0$
    while $i \le N$
      cond =1
      Obtener $S(x_i)$
      Obtener A_R y b_{Y_R} usando (2) y (4)
      seguir=1
      while cond
        Obtener b_{L_R} y λ usando (3) y (7)
        Obtener $\underline{f}_\lambda^j(x_i)$ y $\overline{f}_\lambda^j(x_i)$ con (5) y (6)
        if $y_i > \overline{f}_\lambda^j(x_i)$ **o** $y_i < \underline{f}_\lambda^j(x_i)$
          $L = \epsilon_L \cdot L$, seguir=0
        else
          cond=0
          $INT = INT + \overline{f}_\lambda^j(x_i) - \underline{f}_\lambda^j(x_i)$
        end if
      end while
      if seguir==1
        $i = i + 1$
      else
        $i = 1$, $INT = 0$
      end if
    end while
  $INT_j = INT$
  $L_j = L$
  $j = j + 1$
  $\sigma = \sigma_0 + j \cdot \epsilon_\sigma$
  end while
  $j^* = arg\min_j\ INT_j$



Figura 1: Sistema de dos tanques utilizado.

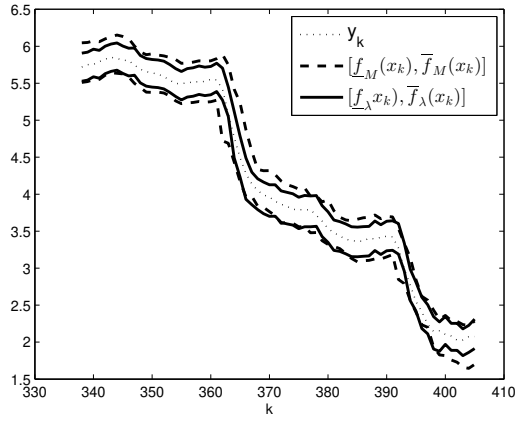


Figura 2: Predicciones intervalares.

devolver $(\sigma_0 + j^* \cdot \epsilon_\sigma, L_{j*})$
End

3.3 Algoritmo de Predicción

En esta subsección se presenta el algoritmo que permite obtener predicciones intervalares de la salida de un sistema. El algoritmo está basado en los resultados obtenidos en las secciones previas. Se asume que en el instante de tiempo k, el conjunto de medidas $\mathcal{D}$ está disponible y se cumplen las Hipótesis de trabajo 1 y 2. A continuación se muestran los pasos del algoritmo de predicción propuesto.

1. Componer los vectores b_{L_R}, b_{Y_R} y las matrices A_R y W^{-1}
2. Calcular el vector
   - $\lambda = W^{-1}A_R(A_R^T W^{-1} A_{R_k})^{-1}x_k$
3. Obtener el intervalo de la predicción $f_\lambda[x_k^T\ 1]^T$
   - $\underline{f}_\lambda(x_k) = -|\lambda|^T b_{L_R} + \lambda_k b_{Y_R}$.
   - $\overline{f}_\lambda(x_k) = |\lambda|^T b_{L_R} + \lambda_k b_{Y_R}$.
4. Obtener la predicción nominal
   - $f_\lambda^c(x_k) = \frac{1}{2}(\underline{f}_\lambda(x_k) + \overline{f}_\lambda(x_k))$.

3.4 Aplicación de ejemplo: Sistema hidráulico de dos tanques

Con el objeto de estudiar la aplicabilidad del predictor intervalar a sistemas reales, se usará una maqueta de dos tanques conectados en cascada (ver Figura 1). Una bomba impulsa el líquido hasta el tanque superior con un flujo que es proporcional a la señal denotada como b_k. Esta es la entrada del sistema. El orificio del tanque superior deposita el líquido en el tanque inferior, el cual vacía su contenido en el tanque de reserva de líquido. La salida del sistema es una señal denotada h_k que se mide cada instante de muestreo. Esta señal es proporcional al nivel de líquido del tanque inferior. El objetivo del ejemplo es predecir el nivel de líquido en el tanque inferior en tiempo de muestreo k, usando las señales b_k, h_{k-1} y h_{k-2}. Por lo tanto, en este caso $y_k = h_k$ y $x_k = [h_{k-1}\ h_{k-2}\ b_k]$. El tiempo de muestreo considerado es 40 segundos. La forma de la señal b_k utilizada como entrada son escalones de 10 minutos de anchura y altura aleatoria comprendida entre 8 y 10. Se han recopilado datos de 616 instantes de muestreo. En la Figura 3 se pueden observar las señales b_k y h_k tomadas del sistema. Los datos correspondientes a los primeros 308 instantes de muestreo constituyen el conjunto $\mathcal{D}$, por tanto, $\mathcal{D} = \{(y_i, x_i)\}$ con $i = 1, ..., 308$. Las medidas obtenidas en los últimos 308 instantes de muestreo se han utilizado como datos de validación.

Dos predictores intervalares se han utilizado para análisis comparativo. El predictor propuesto en este artículo, denotado $f_\lambda(x_k)$, utiliza las expresiones (6) y (7) para obtener las cotas $\underline{f}_\lambda(x_k)$ y $\overline{f}_\lambda(x_k)$. Se denota como $f_M(x_k) = [\underline{f}_M(x_k), \overline{f}_M(x_k)]$ al predictor presentado en [12]. En este caso, la predicción intervalar viene definida mediante las expresiones:

$$\underline{f}_M(x_k) = y_i - \sigma_M - \max_{i=1,...,308} \gamma_M ||x_i - x_k|| \quad (10)$$

$$\overline{f}_M(x_k) = y_i + \sigma_M + \min_{i=1,...,308} \gamma_M ||x_i - x_k|| \quad (11)$$

donde σ_M y γ_M son constantes positivas.

Las constantes σ y L utilizadas en el predictor $f_\lambda(x_k)$ se han obtenido usando el Algoritmo 1

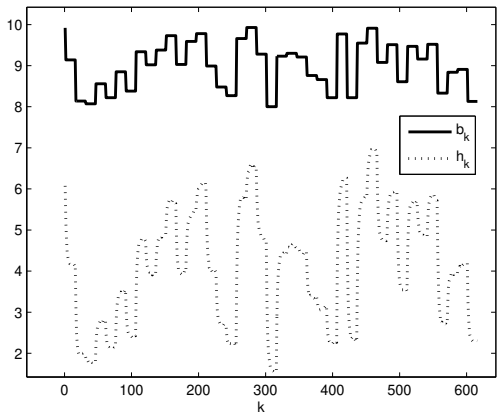


Figura 3: Datos recogidos de la maqueta.

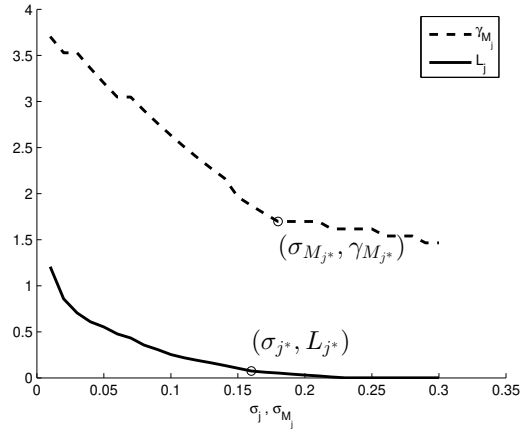


Figura 4: Constantes obtenidas por el Algoritmo 1.

con los siguientes parámetros de entrada: $\sigma_0 = 0.01, \sigma_{max} = 0.3, \epsilon_\sigma = 0.01, L_0 = 1e^{-6}, \epsilon_L = 1.05$ y $\mathcal{D}$. Notar que $\epsilon_\sigma^{-1}(\sigma_{max} - \sigma_0) = 29$. Por tanto, se han chequeado los siguientes valores $\sigma_j = 0.01 + 0.01j$ con $j = 0, ..., 29$. La Figura 4 muestra los valores de σ_j y L_j con j=0,...,29 obtenidos en cada iteración mediante el Algoritmo 1. La Figura 5 muestra la métrica INT_j^D con $j = 0, ..., 29$ obtenida en cada iteración por el Algoritmo 1. Se puede observar que el mínimo se alcanza en $j^* = 15$ y que los valores óptimos devueltos por el Algoritmo 1 son $\sigma = 0.16$ y $L = 0.0747$. Las constantes positivas (σ_M y γ_M) usadas en el predictor $f_M(x_k)$ se obtienen utilizando una versión modificada del Algoritmo 1. En este caso, se han reemplazado las expresiones (5) y (6) del Algoritmo 1 por (10) y (11). El Algoritmo 1 devuelve los valores óptimos $\sigma_M = 0.18$ y $\gamma_M = 1.6969$. Las Figuras 4 y 5 también muestran los parámetros obtenidos en cada iteración y la métrica INT_j^D cuando se emplea el predictor $f_M(x_k)$. Por otra parte, en la Figura 2 se pueden observar las predicciones intervalares obtenidas en datos de validación utilizando las constantes devueltas por el Algoritmo 1.

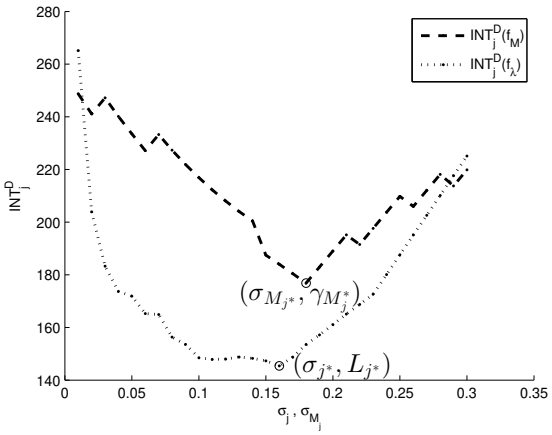


Figura 5: Métrica INT_j^D.

Con objeto de validar los resultados obtenidos por el Algoritmo 1, se considera la siguiente métrica:

$$INT_j^V(f) = \sum_{i=309}^{616} \overline{f}^j(x_i) - \underline{f}^j(x_i),$$

donde el subíndice j indica la dependencia con el par de valores considerado para los parámetros. Esta métrica, ya extendida a ambos predictores, puede generalizarse a cualquier conjunto de datos, por ejemplo $INT_j^D(f)$ se aplicaría al conjunto de datos de entrenamiento $\mathcal{D}$.

La Figura 6 muestra las métricas $INT_j^V(f_\lambda)$ y $INT_j^V(f_M)$ con $j = 0, ..., 29$. Los resultados obtenidos con datos de validación (Figura 6) y datos de entrenamiento (Figura 5) son muy parecidos. Notar que los valores $\sigma_j, L_j, \sigma_{M_j}$ y γ_{M_j} utilizados para obtener $INT_j^V(f_\lambda)$ y $INT_j^V(f_M)$ con $j = 0, ..., 29$, han sido obtenidos previamente mediante el Algoritmo 1 utilizando únicamente el conjunto de entrenamiento $\mathcal{D}$. Se pueden obtener dos conclusiones a partir de la Figura 6. La primera es que el predictor $f_\lambda(x_k)$ propuesto en este artículo mejora los resultados obtenidos por $f_M(x_k)$. La segunda conclusión indica que valores para σ y L devueltos por el Algoritmo 1 son muy cercanos a los que minimizan $INT_j^V(f_\lambda)$.

En la subsección 3.1 se presentó un predictor nominal denotado como $f_\lambda^c(x_k)$. Este predictor es el punto central de la predicción intervalar. Una predicción nominal similar se puede obtener usando $f_M(x_k)$, esto es,

$$f_M^c(x_k) = \frac{1}{2}(\overline{f}_M(x_k) + \underline{f}_M(x_k)).$$

Teniendo en cuenta ambos predictores, la media

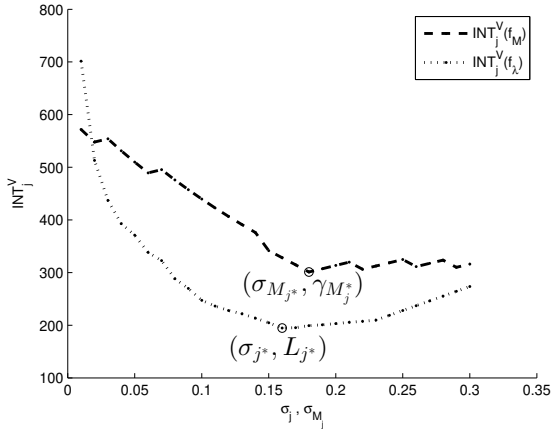


Figura 6: Métrica INT_j^V.

	j^*	σ, σ_M	L, γ_M	$INT_{j^*}^V$	$MAE_{j^*}^V$
$f_\lambda(x_k)$	15	0.16	0.0747	194.66	0.058
$f_M(x_k)$	17	0.18	1.6969	301.19	0.1871

Tabla 1: Tabla de resultados

del error en valor absoluto (denotada MAE) se define mediante la expresión:

$$MAE_j^V(f) = \frac{1}{308} \sum_{i=309}^{616} |f^j(x_i) - y_i|.$$

La Figura 7 muestra las métricas $MAE_j^V(f_\lambda^c)$ y $MAE_j^V(f_M^c)$ con $j = 0, ..., 29$. Se puede observar que el predictor nominal presentado en este trabajo ($f_\lambda^c(x_k)$) obtiene mejores resultado que $f_M^c(x_k)$. Un resumen de los resultados obtenidos se puede consultar en la Tabla 3.4.

4 Conclusiones

En este trabajo se ha presentado un nuevo método para obtener un predictor intervalar aplicable a sistemas dinámicos no necesariamente lineales. El predictor se construye utilizando técnicas no paramétricas basadas en error acotado. Por lo tanto, no se necesita asumir un modelo paramétrico del sistema ni tampoco propiedades estadísticas en el error presente en las medidas o el error de predicción. El predictor utiliza un conjunto de medidas de entrada y salida del sistema y un umbral máximo para el error en medida considerado. Además se asume que el comportamiento del sistema se puede aproximar localmente mediante una función afín que cumple ciertas características. El predictor intervalar proporciona dos valores que constituyen una cota externa de la salida del sistema. La principal aportación del trabajo es la expresión explícita que proporciona dicha predicción. Esta expresión evita la utilización de métodos iterativos de búsqueda. Por otro lado, el punto medio del intervalo predicción se puede utilizar como una predicción nominal de la salida del sistema ya que cumple ciertos criterios de optimalidad. Finalmente se ha incluido una comparación con otro método de predicción intervalar no paramétrico y se han mostrado los buenos resultados obtenidos.

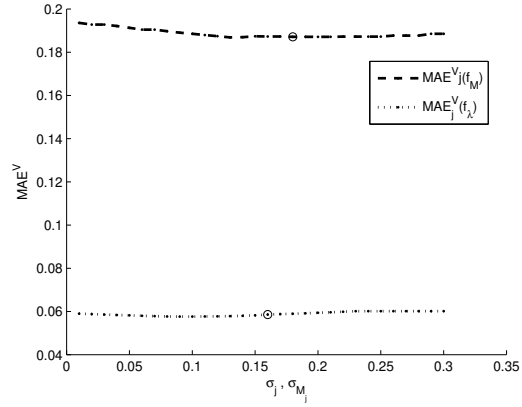


Figura 7: Métrica MAE_j^V.

Agradecimientos

Esta investigación ha sido financiada por DPI2012-33309 del MEC de España.

Referencias

[1] E.W. Bai, Y. Ye, and R. Tempo. Bounded error parameter estimation: a sequential analytic center approach. *IEEE Transactions on Automatic Control*, 44(6):1107–1117, June 1999.

[2] G. Belforte, B. Bona, and V. Cerone. Parameter estimation algorithms for a set-membership description of uncertainty. *Automatica*, 26:887–898, 1990.

[3] J.M. Bravo, T. Alamo, and E.F. Camacho. Bounded error identification of systems with time-varying parameters. *IEEE Transactions on Automatic Control*, 51-7:1144 – 1150, 2006.

[4] J.M. Bravo, T. Alamo, M.J. Redondo, and E.F. Camacho. An algorithm for bounded-error identification of nonlinear systems based on dc functions. *Automatica*, 44-2:437–444, 2008.

[5] V. Broman and M.J. Shensa. A compact algorithm for the intersection and approximation of n-dimensional polytopes. In *Proc. 12th*

IMACS world Congr. on Scientific Computation, volume 2, pages 473–476, Paris, France, July 1988.

[6] E. Fogel and Y.F. Huang. On the value of information in system identification - bounded noise case. *Automatica*, 18:229–238, 1982.

[7] J.L. Goffin and J.P. Vial. Cutting planes and column generation techniques with the projective algorithm. *J. of Optim. Theory and Appl*, 65:409–429, 1990.

[8] L. Jaulin. Interval constraint propagation with application to bounded-error estimation. *Automatica*, 36:1547–1552, 2000.

[9] L. Ljung. *System Identification*. Prentice Hall, Upper Saddle River, NJ, 1999.

[10] M. Milanese and G. Belforte. Estimation theory and uncertainty intervals evaluation in presence of unknown but bounded error: linear families of models and estimators. *IEEE Transactions on Automatic Control*, 27:408–414, 1982.

[11] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter. *Bounding Approaches to System Identification*. Plenum Press, New York, 1996.

[12] M. Milanese and C. Novara. Set membership identification of nonlinear systems. *Automatica*, 40:957–975, 2004.

[13] S.H. Mo and J.P. Norton. Recursive parameter-bounding algorithms which compute polytope bounds. In *Proc. 12th IMACS world Congr. on Scientific Computation*, volume 2, pages 477–480, Paris, France, July 1988.

[14] A. Vicino and G. Zappa. Sequential aproximation of feasible parameter sets for identification with set membership uncertainty. *IEEE Transactions on Automatic Control*, 41(6):774–785, 1996.

[15] E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer-Verlag, 1997.

5 Apéndice

A continuación se incluye una prueba del Teorema 1. Primero es necesario definir las siguientes cotas $\overline{f}_p(x_k)$ y $\underline{f}_p(x_k)$.

$$\begin{aligned}\overline{f}_p(x_k) = \quad & \max_{\theta} \quad [x_k^T\ 1]\theta \\ & s.t. \quad \theta \in \Theta(x_k)\end{aligned}$$

$$\begin{aligned}\underline{f}_p(x_k) = \quad & \min_{\theta} \quad [x_k^T\ 1]\theta \\ & s.t. \quad \theta \in \Theta(x_k)\end{aligned}$$

De la Hipótesis 2 se tiene que para todo x_k existe un $\theta_k \in \Theta(x_k)$ tal que $f_0(x_k) = [x_k^T\ 1]\theta_k$. Teniendo en cuenta lo anterior se infiere que $\underline{f}_p(x_k) \leq f_0(x_k) \leq \overline{f}_p(x_k)$. Teniendo en mente las siguientes igualdades,

- $\Theta(x_k) = \{\ \theta \in \mathbb{R}^{n_g}\ : A_S\theta \leq b_S\} = \{\ \theta \in \mathbb{R}^{n_g}\ : |A_R\theta - b_{Y_R}| \leq b_{L_R}\}$
- $A_R\lambda = A_R W^{-1} A_R (A_R^T W^{-1} A_R)^{-1} [x_k^T\ 1]^T = [x_k^T\ 1]^T$

entonces se puede probar que:

$$\begin{aligned}\overline{f}_p(x_k) &= \max_{\theta}\ [x_k^T\ 1]\theta \\ &\quad s.t.\ A_S\theta \leq b_S \\ &= \max_{\theta}\ \lambda^T A_R\theta \\ &\quad s.t.\ A_S\theta \leq b_S \\ &= \max_{\theta}\ \lambda^T(A_R\theta - b_{Y_R}) + \lambda^T b_{Y_R} \\ &\quad s.t.\ A_S\theta \leq b_S \\ &\leq \max_{\theta}\ |\lambda^T||A_R\theta - b_{Y_R}| + \lambda^T b_{Y_R} \\ &\quad s.t.\ A_S\theta \leq b_S \\ &\leq \max_{\theta}\ |\lambda^T| b_{L_R} + \lambda^T b_{Y_R} \\ &\quad s.t.\ A_S\theta \leq b_S \\ &= |\lambda^T| b_{L_R} + \lambda^T b_{Y_R} = \overline{f}_\lambda(x_k)\end{aligned}$$

Por otro lado:

$$\begin{aligned}\underline{f}_p(x_k) &= \min_{\theta}\ [x_k^T\ 1]\theta \\ &\quad s.t.\ A_S\theta \leq b_S \\ &= \min_{\theta}\ \lambda^T A_R\theta \\ &\quad s.t.\ A_S\theta \leq b_S \\ &= \min_{\theta}\ \lambda^T(A_R\theta - b_{Y_R}) + \lambda^T b_{Y_R} \\ &\quad s.t.\ A_S\theta \leq b_S \\ &\geq \min_{\theta}\ -|\lambda^T||A_R\theta - b_{Y_R}| + \lambda^T b_{Y_R} \\ &\quad s.t.\ A_S\theta \leq b_S \\ &\geq \min_{\theta}\ -|\lambda^T| b_{L_R} + \lambda^T b_{Y_R} \\ &\quad s.t.\ A_S\theta \leq b_S \\ &= -|\lambda^T| b_{L_R} + \lambda^T b_{Y_R} = \underline{f}_\lambda(x_k)\end{aligned}$$

De las expresiones anteriores se infiere que: $\underline{f}_\lambda(x_k) \leq \underline{f}_p(x_k) \leq f_0(x_k) \leq \overline{f}_p(x_k) \leq \overline{f}_\lambda(x_k)\ \forall x_k \in X$ ■

MODELADO SIMPLIFICADO Y ORIENTADO AL CONTROL DE SISTEMAS DE REFRIGERACIÓN

Guillermo Bejarano, Manuel G. Ortega, Francisco R. Rubio
Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla
{gbejarano, mortega, rubio}@us.es

Fernando Morilla
Departamento de Informática y Automática, UNED
fmorilla@dia.uned.es

Resumen

En este trabajo se estudia el modelado dinámico de sistemas de refrigeración por compresión de vapor, desde un punto de vista simplificado y orientado al control. En primer lugar se desarrollan submodelos matemáticos para cada uno de los componentes del sistema: intercambiadores de calor (evaporador y condensador), compresor de velocidad variable y válvula de expansión electrónica. Interconectando dichos modelos de forma adecuada se realizan simulaciones sobre la evolución dinámica de las principales variables del ciclo completo ante cambios en las variables manipulables: la velocidad de giro del compresor y la apertura de la válvula de expansión. El objetivo principal es el desarrollo de modelos básicos útiles para el desarrollo de estrategias de control.

Palabras Clave: Sistemas de refrigeración, Ciclo de compresión de vapor, Modelado, Simulación.

1 INTRODUCCIÓN

El sistema de compresión de vapor, véase la Figura 1, es el método más utilizado a nivel mundial para la generación de frío, ya sea para aplicaciones en refrigeración doméstica, comercial, industrial o climatización [9]. La refrigeración supone un alto porcentaje del consumo energético y su impacto económico y medioambiental está bien caracterizado. Por ejemplo, los supermercados constituyen uno de los mayores consumidores del sector energético. Un supermercado típico consume entre 2 y 3 millones de kWh anualmente [1], y en torno al 50% de esta energía se consume en los procesos de refrigeración. En el caso de edificios de oficinas se ha estimado que el consumo debido al uso de sistemas de acondicionamiento de aire o HVAC (del inglés *Heating, Ventilating, and Air Conditioning*) está en torno al 20-40% del consumo de energía total en países desarrollados [8].

Debido a la creciente necesidad de un consumo más racional de la energía, el interés en mejorar la eficiencia energética y reducir el impacto ambiental de los sistemas actuales de refrigeración se ha visto incrementado en los últimos años. Estas mejoras pasan por las siguientes acciones, entre otras:

- Rediseño de los intercambiadores de calor (evaporador y condensador), buscando un aumento sustancial en su rendimiento.
- Incorporación de elementos electrónicos y sistemas de control para poder operar con presiones flotantes en la condensación y en la evaporación.
- Detección automática de anomalías y degradaciones.
- Búsqueda del máximo coeficiente de comportamiento (COP) mediante la adaptación del sistema a las condiciones de trabajo.
- Uso de refrigerantes seguros para el medio ambiente e inocuos para la capa de ozono.

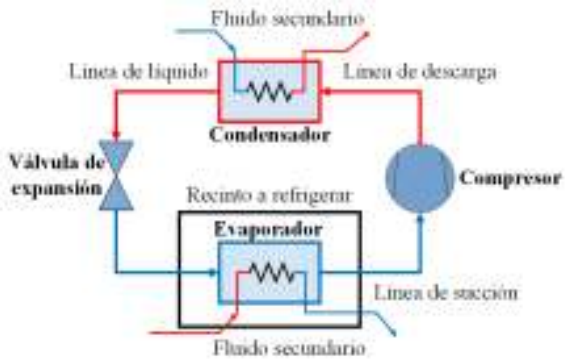


Figura 1: Esquema simple del sistema de compresión de vapor de una etapa

Los sistemas de refrigeración son sistemas multivariables no lineales con un alto grado de acoplamiento [12]. Desde el punto de vista de control, las principales variables manipuladas son la velocidad de giro del compresor, que proporciona el

Tabla 1: Nomenclatura.

Nomenclatura

Símbolos

A	Área transversal de paso del refrigerante en el intercambiador (m^2)
A_{ef}	Área efectiva de paso del refrigerante por la válvula de expansión (m^2)
A_{nom}	Área nominal de paso de la válvula de expansión (m^2)
A_{trnsf}	Área de transferencia de calor entre el refrigerante y el fluido secundario (m^2)
A_v	Apertura de la válvula (%)
a	Parámetro del compresor (W)
b	Parámetro del compresor (adimensional)
COP	Coeficiente de comportamiento (adimensional)
c_{eev}	Coeficiente de apertura de la válvula de expansión (adimensional)
c_p	Calor específico a presión constante (J/kg K)
c_v	Calor específico a volumen constante (J/kg K)
c	Parámetro del compresor (adimensional)
d	Parámetro de la válvula de expansión (1/K)
e	Parámetro de la válvula de expansión (adimensional)
$G(s)$	Función de transferencia
h	Entalpía específica (J/kg)
K	Ganancia de la función de transferencia
L	Longitud del intercambiador (m)
$\dot{m}$	Caudal másico (kg/s)
N	Velocidad de giro del compresor (rpm)
P	Presión (Pa)
$\dot{Q}$	Potencia calorífica intercambiada (W)
S_t	Parámetro del compresor (m^3)
s	Entropía específica (J/kg K), variable de Laplace
T	Temperatura (K)
TSC	Grado de subenfriamiento (K)
TSH	Grado de sobrecalentamiento (K)
t	Tiempo (s)
UA_{comp}	Coeficiente global de pérdidas de calor del compresor (W/K)
$\boldsymbol{u}$	Vector de entradas manipulables
v	Volumen específico (m^3/kg)
$\dot{W}$	Potencia eléctrica demandada (W)
$\boldsymbol{w}$	Vector de perturbaciones al ciclo
$\boldsymbol{x}$	Vector de variables de estado del ciclo
x	Fracción de la longitud total del intercambiador (adimensional)
x_δ	Variable auxiliar para zonas monofásicas (adimensional)
$\boldsymbol{y}$	Vector de salidas del ciclo
α	Coeficiente de transferencia de calor entre el refrigerante y el fluido secundario (W/m^2 K)
γ	Fracción de vacío (adimensional)
ρ	Densidad (kg/m^3)
τ	Constante de tiempo de la función de transferencia (s)

Subíndices

c	condensador
$comp$	compresor
e	evaporador
in	entrada
is	isentrópico
out	salida
$refr$	refrigerante
s	vapor saturado
sc	zona de subenfriamiento
sec	fluido secundario
sh	zona de sobrecalentamiento
$surr$	entorno
w	líquido saturado
ws	zona bifásica

salto de presión entre la salida del evaporador y la entrada del condensador, y la apertura de la válvula de expansión, que permite regular el caudal de refrigerante que circula por el sistema (véase la Figura 1). Con estas variables se pretenden regular, en primer lugar, la capacidad de carga del sistema para responder a la demanda frigorífica y, en segundo lugar, se pretende asegurar un funcionamiento eficiente de la instalación manteniendo para ello un COP máximo. Existen otras entradas al sistema, en forma de perturbaciones, que pueden afectar al funcionamiento del sistema (temperatura exterior, temperatura de los fluidos secundarios en el evaporador y el condensador, etc.).

La bibliografía sobre el modelado de sistemas de refrigeración por compresión de vapor es muy amplia [11]. Se pueden encontrar en la literatura desde modelos muy complejos y detallados, a modelos simplificados orientados al diseño de estrategias de control multivariable. En concreto, los elementos más importantes de cara al modelado dinámico son los intercambiadores de calor, ya que la dinámica del compresor y la válvula de expansión suele ser al menos un orden de magnitud más rápida, de forma que por simplicidad estos elementos acostumbran a modelarse de forma cuasi-estática.

Un modelo detallado de un intercambiador de calor se basa en las ecuaciones de conservación de masa, momento y energía del refrigerante, el fluido secundario y el material que los separa. Esto conduce a una solución numérica de un conjunto de ecuaciones diferenciales discretizadas según el método de diferencias finitas [5]. Este modelo proporciona información muy detallada sobre el sistema, siendo útil para probar el desempeño de diferentes leyes de control, pero debido a su

complejidad no lo es para el diseño de nuevas estrategias de control.

Un modelo más simple puede obtenerse utilizando el llamado *moving boundary model* para el flujo de refrigerante en el intercambiador y asumiendo que las variaciones espaciales en la presión son despreciables. Este modelo es muy general, puede adaptarse a la mayoría de tipos de intercambiadores y describe la dinámica del sistema de forma adecuada [3], [4], [7], [9].

En el presente estudio se simplifica aún más el *moving boundary model*, obteniéndose un modelo no lineal más simple que describe la constante de tiempo dominante y el comportamiento no lineal entre la entrada y la salida.

El resto del artículo se organiza de la siguiente manera: en el Apartado 2 se desarrollan los submodelos de los elementos del ciclo y se describe su interconexión y el algoritmo de resolución del ciclo completo. En el Apartado 3 se presentan las simulaciones realizadas sobre la evolución dinámica de las principales variables del ciclo ante cambios en las variables manipulables y se identifican funciones de transferencia de primer orden para las variables más importantes de cara al control. Finalmente en el Apartado 4 se resumen las conclusiones obtenidas y se expresan los trabajos futuros a realizar.

2 MODELADO DINÁMICO DEL SISTEMA

El modelado se basa en el desarrollo de submodelos para cada uno de los componentes del ciclo, los cuales pueden ser dinámicos (flujo de refrigerante a lo largo de los intercambiadores de calor) o estáticos (compresor, válvula de expansión y fluidos secundarios en evaporador y condensador). En la Tabla 1 se detalla la nomenclatura utilizada.

2.1 MODELO DEL COMPRESOR

El modelado estático del compresor se basa en el procedimiento descrito en [13], [14]. El modelo recibe como entradas las presiones P_c y P_e y la entalpía específica de succión $h_{out,e}$, y proporciona como salidas el caudal de refrigerante succionado desde el evaporador y descargado al condensador, la entalpía específica de descarga y la potencia consumida por el compresor:

$$\dot{m}_{in,c} = \left(S_t - cS_t \left(\left(\frac{P_c}{P_e} \right)^{\frac{c_{v,s,e}}{c_{p,s,e}}} - 1 \right) \right) \frac{N}{v_{out,e}} \quad (1)$$

$$\dot{W}_{comp} = a + b\,\dot{m}_{in,c}(h_{in,c,is} - h_{out,e}) \quad (2)$$

$$T_{in,c,is} = T_c + \frac{h_{in,c,is} - h_{s,c}}{c_{p,s,c}} \quad (3)$$

$$h_{in,c} = h_{out,e} + \frac{\dot{W}_{comp} - UA_{comp}(T_{in,c,is} - T_{surr})}{\dot{m}_{in,c}} \quad (4)$$

donde $h_{in,c,is} = h(P_c, s_{out,e})$ y $s_{out,e} = s(P_e, h_{out,e})$ son funciones termodinámicas para la evaluación de propiedades del refrigerante. Los parámetros a, b, c, S_t y UA_{comp} son característicos del compresor y, o bien se consultan en catálogos, o bien se ajustan en un procedimiento de identificación.

2.2 MODELO DE LA VÁLVULA DE EXPANSIÓN

De la misma forma, el modelado estático de la válvula de expansión se basa en el procedimiento descrito en [13], [14]. El modelo recibe como entradas las presiones P_c y P_e y la entalpía específica a la entrada de la válvula $h_{out,c}$, y proporciona como salida el caudal de refrigerante expandido:

$$\dot{m}_{in,e} = c_{eev} A_{ef} \sqrt{2\rho_{out,c}(P_c - P_e)} \quad (5)$$

$$\rho_{out,c} = \rho(P_c, h_{out,c}) \quad (6)$$

$$A_{ef} = \frac{A_v}{100} A_{nom} \quad (7)$$

$$c_{eev} = d(T_c - T_{out,c})^e \quad (8)$$

Se asume que la expansión es isentálpica ($h_{in,e} = h_{out,c}$). Los parámetros d y e de la ecuación de cálculo de c_{eev} son característicos de la válvula y, o bien se consultan en catálogos, o bien se ajustan en un procedimiento de identificación.

2.3 MODELO DEL EVAPORADOR

El modelado se basa en el procedimiento descrito en [10], en el cual se consideran las siguientes suposiciones y aproximaciones:

- El flujo de refrigerante puede considerarse unidimensional.
- Las variaciones espaciales de la presión en el intercambiador son despreciables.
- La conducción térmica axial es despreciable.
- El área de paso del refrigerante es constante.
- El coeficiente de transferencia de calor entre el fluido secundario y la pared del intercambiador es mucho menor que el existente entre la pared y el refrigerante en forma de vapor sobrecalentado.

- La energía necesaria para sobrecalentar el refrigerante es despreciable en comparación con la necesaria para evaporarlo.
- La capacidad térmica de la pared del intercambiador entre el fluido secundario y el refrigerante se considera despreciable.

Aplicando el llamado *moving boundary model* [3] y realizando el balance de masa y energía en la zona bifásica según se indica en [10], se obtienen las ecuaciones diferenciales (9) y (10):

$$\rho_{w,e}(1-\gamma_e)\left(h_{s,e}-h_{w,e}\right)A_e L_e \frac{dx_e}{dt} = \left(h_{s,e}-h_{in,e}\right)\dot{m}_{in,e} - \alpha_{ws,e}A_{trnsf,e}x_e\left(T_{in,sec,e}-T_e\right) \quad (9)$$

$$\frac{\partial \rho_{s,e}}{\partial P_e}\gamma_e A_e L_e x_e \frac{dP_e}{dt} = \dot{m}_{in,e} - \dot{m}_{out,e} \quad (10)$$

Para la obtención de (9) y (10) se asume asimismo que γ_e es constante e independiente de x_e, que la fase líquida es incompresible $\left(\frac{\partial \rho_{w,e}}{\partial P_e} \approx 0\right)$ y que las variaciones de $h_{w,e}$ y $h_{s,e}$ respecto a las variaciones de presión son despreciables.

Las ecuaciones (9) y (10) conforman un modelo dinámico no lineal con dos variables de estado: x_e y P_e. Las variables de entrada son los caudales de refrigerante a la entrada $\dot{m}_{in,e}$ y a la salida $\dot{m}_{out,e}$, y la entalpía específica a la entrada $h_{in,e}$.

Como variables de salida se definen el grado de sobrecalentamiento del vapor TSH, la potencia frigorífica intercambiada $\dot{Q}_e$ y la temperatura de salida del fluido secundario $T_{out,sec,e}$:

$$TSH = \left(T_{in,sec,e}-T_e\right)\left(1-e^{-\frac{1-x_e}{x_{\delta,sh,e}}}\right) \quad (11)$$

$$x_{\delta,sh,e} = \frac{c_{p,s,e}\dot{m}_{in,e}}{\alpha_{sh,e}A_{trnsf,e}} \quad (12)$$

$$\dot{Q}_e = 0.5\left(\dot{m}_{in,e} + \dot{m}_{out,e}\right)\left(h_{out,e}-h_{in,e}\right) \quad (13)$$

$$T_{out,sec,e} = T_{in,sec,e} - \frac{\dot{Q}_e}{\dot{m}_{sec,e}c_{p,sec,e}} \quad (14)$$

donde $h_{out,e} = h(P_e, T_{out,e})$ y $T_{out,e} = T_e + TSH$. La ecuación (11) se obtiene siempre que las condiciones del fluido secundario sean tales que $\dot{m}_{sec,e}c_{p,sec,e} \gg \dot{m}_{in,e}c_{p,s,e}$.

Este modelo asume que existe una zona bifásica y una zona de sobrecalentamiento en el evaporador. En regímenes transitorios de parada y arranque del sistema el evaporador puede constar solo de la zona bifásica [6]. El modelo simplificado desarrollado no contempla esta situación transitoria.

2.4 MODELO DEL CONDENSADOR

El modelado se basa igualmente en el procedimiento descrito en [10], aunque adaptado para el condensador, en el que en régimen nominal existen tres zonas: de sobrecalentamiento, bifásica y de subenfriamiento. Las suposiciones y aproximaciones realizadas son similares a las tenidas en cuenta en el caso del evaporador.

Aplicando de nuevo el llamado *moving boundary model* [3] y realizando el balance de masa y energía en la zona bifásica según se indica en [10], se obtienen las ecuaciones diferenciales (15) y (16):

$$\rho_{s,c}(1-\gamma_c)\left(h_{s,c}-h_{w,c}\right)A_c L_c \frac{dx_c}{dt} = \left(h_{s,c}-h_{w,c}\right)\dot{m}_{in,c} - \alpha_{ws,c}A_{trnsf,c}x_c\left(T_c - T_{in,sec,c}\right) \quad (15)$$

$$\frac{\partial \rho_{s,c}}{\partial P_c}\gamma_c A_c L_c x_c \frac{dP_c}{dt} = \dot{m}_{in,c} - \dot{m}_{out,c} \quad (16)$$

Las ecuaciones (15) y (16) conforman un modelo dinámico no lineal con dos variables de estado: x_c y P_c. Las variables de entrada son los caudales de refrigerante a la entrada $\dot{m}_{in,c}$ y a la salida $\dot{m}_{out,c}$, y la entalpía específica a la entrada $h_{in,c}$. En este caso, además de la fracción del condensador en zona bifásica x_c, es necesario definir la fracción que está en zona sobrecalentada $x_{sh,c}$:

$$x_{sh,c} = x_{\delta,sh,c} ln\left(\frac{T_{in,c}-T_{in,sec,c}}{T_c - T_{in,sec,c}}\right) \quad (17)$$

donde:

$$x_{\delta,sh,c} = \frac{c_{p,s,c}\dot{m}_{in,c}}{\alpha_{sh,c}A_{trnsf,c}} \quad (18)$$

Como variables de salida se definen el grado de subenfriamiento del refrigerante TSC, la potencia calorífica intercambiada $\dot{Q}_c$ y la temperatura de salida del fluido secundario $T_{out,sec,c}$:

$$TSC = \left(T_c - T_{in,sec,c}\right)\left(1-e^{-\frac{1-x_c-x_{sh,c}}{x_{\delta,sc}}}\right) \quad (19)$$

$$x_{\delta,sc} = \frac{c_{p,w,c}\dot{m}_{in,c}}{\alpha_{sc,c}A_{trnsf,c}} \quad (20)$$

$$\dot{Q}_c = 0.5\left(\dot{m}_{in,c} + \dot{m}_{out,c}\right)\left(h_{in,c}-h_{out,c}\right) \quad (21)$$

$$T_{out,sec,c} = T_{in,sec,c} + \frac{\dot{Q}_c}{\dot{m}_{sec,c}c_{p,sec,c}} \quad (22)$$

donde $h_{out,c} = h(P_c, T_{out,c})$ y $T_{out,c} = T_c - TSC$.

El modelo del condensador asume que existen tres zonas diferenciadas: de sobrecalentamiento, bifásica y de subenfriamiento. En regímenes transitorios de parada y arranque del sistema el condensador puede pasar por hasta 4 situaciones diferentes más (solo sobrecalentamiento, sobrecalentamiento y zona bifásica, solo zona bifásica, o zona bifásica y subenfriamiento) [6]. El modelo simplificado no contempla estas estructuras transitorias.

2.5 ALGORITMO DE RESOLUCIÓN

La Figura 2 muestra la interconexión de los modelos de los elementos del ciclo de vapor, mientras que la Figura 3 muestra la distribución de entradas, salidas y perturbaciones del ciclo completo.

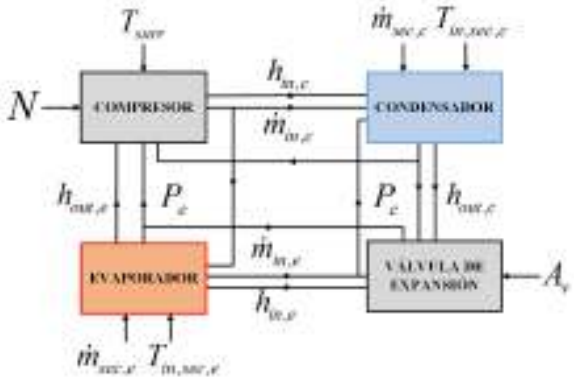


Figura 2: Relaciones de variables de entrada/salida entre componentes del sistema

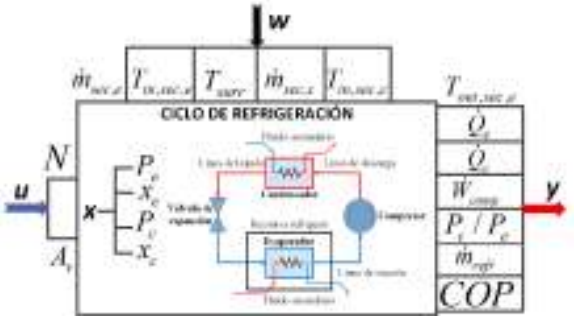


Figura 3: Distribución de variables de entrada, salida, estado y perturbaciones del sistema completo

La simulación del sistema completo se basa en la integración del conjunto de ecuaciones diferenciales formado por los modelos de cada componente. Las condiciones iniciales son las presiones de condensación y evaporación, y las fracciones del evaporador y del condensador en zona bifásica. Estas pueden estimarse mediante la resolución del sistema de ecuaciones estático que se obtiene al anular las derivadas temporales de las variables de estado.

Los coeficientes de transferencia de calor dependen del tipo de intercambiador y se estiman a partir de correlaciones empíricas. Las propiedades termodinámicas del refrigerante y de los fluidos secundarios se calculan mediante la base de datos de libre acceso denominada *CoolProp* [2].

Para la resolución numérica del sistema, en primer lugar se hallan los caudales de refrigerante a partir de las condiciones termodinámicas en el instante anterior, y en segundo lugar se integran las ecuaciones diferenciales de los intercambiadores de calor mediante un método de paso fijo.

3 SIMULACIONES

Se considera un ciclo de vapor con el refrigerante R134a, cuyos componentes tienen las características detalladas en la Tabla 2. El compresor y la válvula de expansión seleccionados son los identificados en [13], mientras que los intercambiadores seleccionados son de tipo carcasa y tubo, con el refrigerante circulando por el interior del tubo y el fluido secundario por la carcasa. El fluido secundario en el evaporador es aire seco con una temperatura de entrada de 1ºC, mientras que el fluido secundario en el condensador es agua líquida a 20ºC.

Tabla 2: Características de los elementos del ciclo.

Parámetro	**Valor**	**Unidad**
Compresor		
a	24.87	W
b	1.613	*adimensional*
c	0.0239	*adimensional*
S_t	$9 \cdot 10^{-6}$	m^3
UA_{comp}	1.87	W/K
Válvula de expansión		
d	0.524	1/K
e	0.186	*adimensional*
A_{nom}	$0.1238 \cdot 10^{-4}$	m^2
Evaporador		
L	5	m
γ	0.8	*adimensional*
A	$7.85 \cdot 10^{-3}$	m^2
A_{trnsf}	1.571	m^2
Condensador		
L	5	m
γ	0.8	*adimensional*
A	$7.85 \cdot 10^{-3}$	m^2
A_{trnsf}	1.571	m^2

Las variables manipulables del sistema son la velocidad de giro del compresor N y la apertura de la válvula de expansión A_v. Se presentan simulaciones con variaciones de cada una de ellas para observar la evolución dinámica de las variables de estado y de salida. Las simulaciones han sido realizadas con el software de cálculo MATLAB.

3.1 VARIACIÓN DE LA VELOCIDAD DE GIRO DEL COMPRESOR

En la Figura 4 se muestra el perfil temporal de las variables manipulables. Se mantiene constante A_v y se varía N respecto a punto de equilibrio. Las Figuras 5 y 6 muestran la evolución dinámica de las variables de estado, mientras que las Figuras 7 y 8 muestran la evolución de algunas de las variables de salida (TSH, TSC, $\dot{m}_{refr}$, $\dot{Q}_e$, $\dot{Q}_c$, $\dot{W}_{comp}$ y COP). En la Figura 9 se muestran en un diagrama *p-h* los distintos puntos de equilibrio alcanzados durante la simulación.

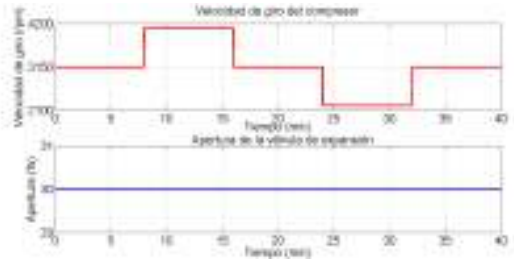


Figura 4: Perfil temporal de variables manipulables, variación de N

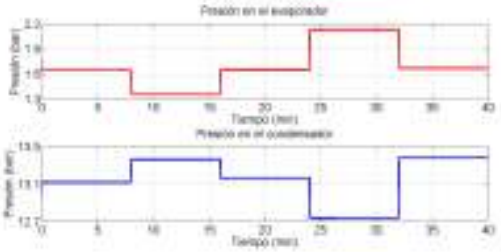


Figura 5: Evolución dinámica de P_c y P_e, variación de N

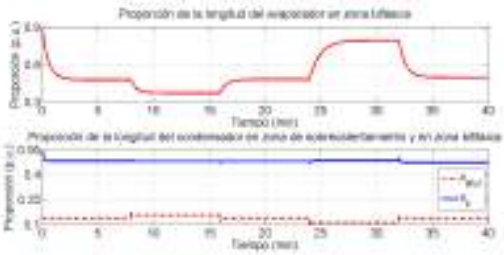


Figura 6: Evolución dinámica de x_c y x_e, variación de N

Se observa que la dinámica de P_c y P_e es mucho más rápida que la de x_e. Esto se debe a que las presiones evolucionan de forma apreciable mientras existe desequilibrio de caudales másicos de refrigerante a la entrada y a la salida de los intercambiadores. Una vez que se equilibran los caudales, la evolución se debe exclusivamente a parámetros termodinámicos, los cuales demuestran generar una dinámica más lenta en el caso del evaporador (a baja presión).

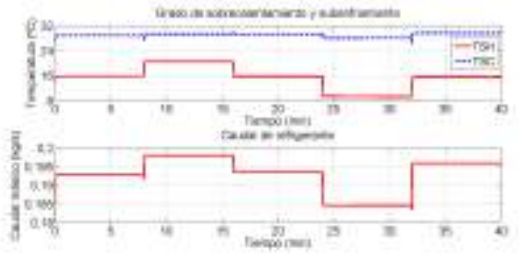


Figura 7: Evolución dinámica de TSH, TSC y $\dot{m}_{refr}$, variación de N

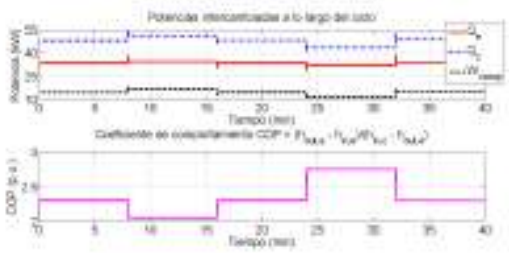


Figura 8: Evolución dinámica de $\dot{Q}_e$, $\dot{Q}_c$, $\dot{W}_{comp}$ y COP, variación de N

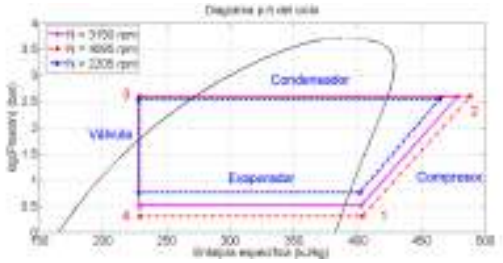


Figura 9: Diagrama *p-h* del ciclo, variación de N

El COP está muy relacionado con el grado de sobrecalentamiento del refrigerante a la salida del evaporador. Para obtener un COP elevado es necesario mantener un bajo TSH, pero siempre positivo, debido a los problemas tecnológicos derivados de la introducción en el compresor de una mezcla líquido/vapor.

En la Figura 9 se observa que si aumenta N disminuye P_c, manteniéndose casi constante P_e. Esto hace que aumente el área interior del ciclo, pero también aumenta el sobrecalentamiento y la potencia consumida por el compresor, de forma que el COP disminuye.

3.2 VARIACIÓN DE LA APERTURA DE LA VÁLVULA DE EXPANSIÓN

En la Figura 10 se muestra el perfil temporal de las variables manipulables. En este caso se mantiene constante N y se varía A_v respecto a punto de equilibrio. Las Figuras 11 y 12 muestran la evolución de las variables de estado, mientras que las Figuras

13 y 14 muestran la evolución de algunas de las variables de salida. En la Figura 15 se muestra el diagrama logarítmico *p-h* del ciclo en los tres puntos de equilibrio diferentes alcanzados.

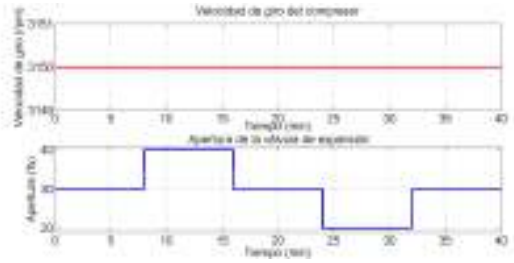


Figura 10: Perfil temporal de variables manipulables, variación de A_v

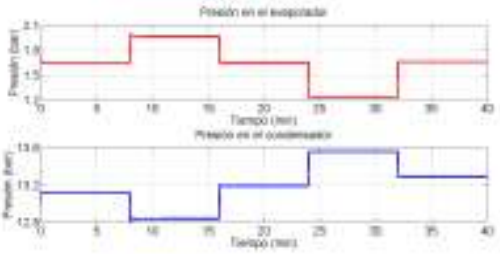


Figura 11: Evolución dinámica de P_c y P_e, variación de A_v

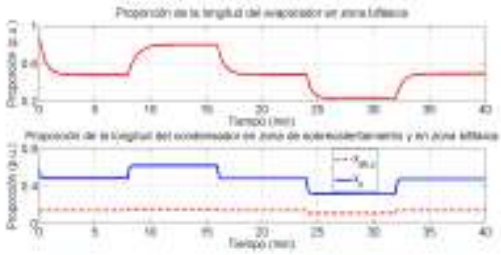


Figura 12: Evolución dinámica de x_c y x_e, variación de A_v

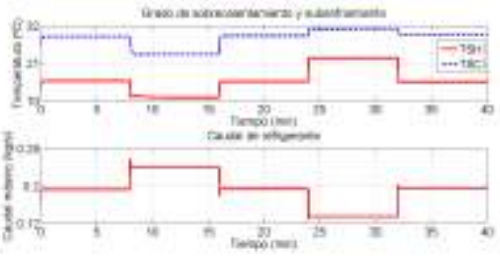


Figura 13: Evolución dinámica de TSH, TSC y $\dot{m}_{refr}$, variación de A_v

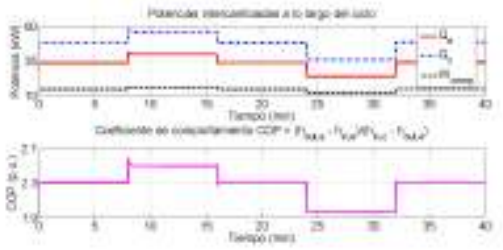


Figura 14: Evolución dinámica de $\dot{Q}_e$, $\dot{Q}_c$, $\dot{W}_{comp}$ y COP, variación de A_v

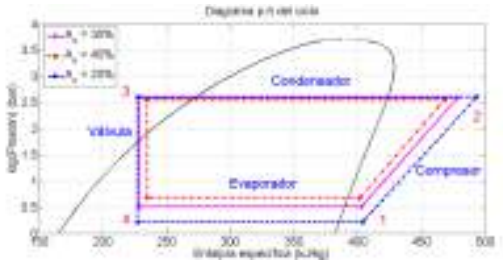


Figura 15: Diagrama *p-h* del ciclo, variación de A_v

La variación de A_v influye de manera más decisiva en el caudal de refrigerante que circula por el ciclo, tal como puede observarse en la Figura 13. En la Figura 15 se observa que al aumentar $\dot{m}_{refr}$ disminuye el sobrecalentamiento a la salida del evaporador, con lo que el COP aumenta. La potencia consumida por el compresor $\dot{W}_{comp}$ aumenta ligeramente, pero también lo hace (y de forma más importante) la potencia calorífica $\dot{Q}_e$ intercambiada en el evaporador.

3.3 IDENTIFICACIÓN DE FUNCIONES DE TRANSFERENCIA

En la Tabla 3 se indican las ganancias y las constantes de tiempo de las funciones de transferencia de primer orden $\left(G(s) = \frac{K}{\tau s+1}\right)$ identificadas a partir de las simulaciones presentadas. Se escogen como variables de salida las más importantes de cara al control: $\dot{Q}_e$ y TSH (medida indirecta del COP).

Tabla 3: Funciones de transferencia identificadas.

Parámetro	Valor	Unidad
$K_{\dot{Q}_e\,N}$	9.4988	W s
$\tau_{\dot{Q}_e\,N}$	4.43	s
$K_{\dot{Q}_e\,A_v}$	637	W / %
$\tau_{\dot{Q}_e\,A_v}$	34.14	s
$K_{TSH\,N}$	0.0464	K s
$\tau_{TSH\,N}$	0.46	s
$K_{TSH\,A_v}$	−0.49	K / %
$\tau_{TSH\,A_v}$	62.23	s

Estas funciones de transferencia representan la dinámica de las variables de salida seleccionadas respecto a las entradas manipulables en el punto de operación estudiado ($N = 3150$ rpm y $A_v = 30$ %).

4 CONCLUSIONES Y TRABAJOS FUTUROS

Se ha desarrollado un modelo dinámico simplificado orientado al control de un ciclo de refrigeración de una etapa, basado en submodelos dinámicos o estáticos de sus componentes. A partir de las simulaciones realizadas variando las entradas manipulables del sistema, se han obtenido modelos en forma de funciones de transferencia de primer orden para las principales variables de salida de cara al control: la potencia frigorífica intercambiada en el evaporador y el grado de sobrecalentamiento. De esta forma se ha caracterizado la dinámica dominante en la respuesta del sistema.

Los submodelos dinámicos de los intercambiadores de calor no contemplan otras situaciones respecto al flujo de refrigerante que pueden darse en régimen transitorio de parada y arranque del sistema. En la actualidad se está trabajando para ampliar las capacidades del modelo simplificado de forma que pueda contemplar todos los regímenes de funcionamiento del ciclo.

Agradecimientos

Los autores quieren expresar su agradecimiento al MCeI por la financiación de este trabajo, a través del proyecto DPI2012-37580-C02-02.

Referencias

[1] Baxter, Van D., (2002) "Advances in supermarket refrigeration systems". Oak Ridge National Laboratory, Oak Ridge, TN 37831-6070, 2002.

[2] Bell, I., Quoilin, S., Wronski, J., (2013) *CoolProp*: *http://coolprop.sf.net/*

[3] Grald, E.W., MacArthur, J.W., (1992) "Moving boundary formulation for modeling time-dependent two-phase flows", *Int. J. Heat and Fluid Flow*, Vol. 13, Iss. 3, pp. 266–272.

[4] He, X. D., Asada, H. H., Liu, S., Itoh, H., (1998) "Multivariable control of vapor compression systems", *HVAC&R Research*, Vol. 4, Iss. 3, pp. 205–230.

[5] Jia, X., Tso, C.P., Jolly, P., Wong, Y.W., (1999) "Distributed steady and dynamic modeling of dry-expansion evaporators", *Int. Journal of Refrigeration*, Vol. 22, Iss. 2, pp. 126–136.

[6] Li, B., Alleyne, A., (2010) "A dynamic model of a vapor compression cycle with shut-down and start-up operations", *International Journal of Refrigeration*, Vol. 33, Iss. 3, pp. 538-552.

[7] Liang, Y., Shao, S., Tain, C., Yan, Y, (2010) "Dynamic simulation of variable capacity refrigeration systems under abnormal conditions", *Applied Thermal Engineering*, Vol. 30, Iss. 10, pp. 1205-1214.

[8] Pérez-Lombard, L., Ortiz, J., Pout, C., (2008) "A review on buildings energy consumption information", *Energy and Buildings*, Vol. 40, Iss. 3, pp. 394-398.

[9] Rasmussen, B. P., (2005) "Dynamic modeling and advanced control of air conditioning and refrigeration systems", Phd Thesis. University of Illinois at Urbana-Champaign.

[10] Rasmussen, H., Larsen, L. F. S., (2011) "Non-linear and adaptive control of a refrigeration system", *IET Control Theory Appl.*, Vol. 5, Iss. 2, pp. 364–378.

[11] Rodríguez, R. D., (2011) "Diseño de planta experimental de refrigeración por compresión de vapor", Trabajo de investigación, Universidad de Córdoba.

[12] Sarabia, D., Capraro, F., Larsen, L.F.S, de Prada, C., (2009) "Hybrid NMPC of supermarket display cases", *Control Engineering Practice*, Vol. 17, Iss. 4, pp. 428–441.

[13] Schurt, L. C., Hermes, C. J. L., Trofino-Neto, A., (2009) "A model-driven multivariable controller for vapor compression refrigeration systems", *International Journal of Refrigeration*, Vol. 32, Iss. 7, pp. 1672-1682.

[14] Schurt, L. C., Hermes, C. J. L., Trofino-Neto, A., (2010) "Assessment of the controlling envelope of a model-based multivariable controller for vapor compression refrigeration systems", *Applied Thermal Engineering*, Vol. 30, Iss. 13, pp. 1538-1546.

Estudio de la interacción del fuselaje y el movimiento de aleteo de las palas del rotor principal en un helicóptero

Salvador Castillo-Rivera
The Engineering and Mathematics Department. City University, Londres, Reino Unido,
email: Salvador.Castillo-Rivera.1@city.ac.uk

María Tomás-Rodríguez
The Engineering and Mathematics Department. City University, Londres, Reino Unido,
email: Maria.Tomas-Rodriguez.1@city.ac.uk

G. Nicolás Marichal Plasencia
Departamento de Ing. de Sistemas y Automática, Arquitectura y Tecnología de Computadores,
Universidad de La Laguna, La Laguna, 38071. Tenerife. España, email: nicomar@ull.edu.es

Ángela Hernández López
Departamento de Ing. de Sistemas y Automática, Arquitectura y Tecnología de Computadores,
Universidad de La Laguna, La Laguna, 38071. Tenerife. España, email: angela@isaatc.ull

Resumen

*En este trabajo se tratan diferentes aspectos del modelado de la dinámica de un helicóptero. El modelo que se presenta tiene configuración Sikorsky, es decir, el modelo está formado por el rotor principal, fuselaje y rotor de cola. El rotor principal es articulado siendo sus palas rígidas. La construcción del rotor principal se ha llevado a cabo teniendo en cuenta los grados de libertad de las palas, aleteo, retraso y ángulo de ataque y el acoplamiento dinámico que existe entre ellos. El modelo ha sido construido utilizando VehicleSim, programa especializado en el modelado de sistemas mecánicos compuestos por cuerpos rí*gidos. *Uno de los aspectos de interés en los helicópteros es la influencia de los acoplamientos que se producen entre las distintas componentes de un helicóptero, en particular, es de gran influencia el existente entre el fuselaje y el movimiento de aleteo de las palas del rotor principal. El artículo presenta el estudio y análisis de la influencia del movimiento de aleteo sobre el fuselaje y viceversa, mediante el uso del análisis modal, que permite identificar los diferentes modos de oscilación que aparecen en el sistema. Los resultados de las simulaciones son analizados y comparados con la teoría existente en la literatura especializada.*

Palabras Clave: Fuselaje, palas, aleteo, acoplamiento.

1 INTRODUCCIÓN

Los helicópteros, son sistemas mecánicamente complejos, formados por varios cuerpos rotatorios, poseen varios grados de libertad y son capaces de desplazarse y ejecutar rotaciones en tres direcciones.
Los estudios realizados en el ámbito de los helicópteros son muy amplios e incluyen, entre otros muchos, métodos avanzados en CFD (computational fluid dynamics) dinámica de cuerpos flexibles [7]...etc. La mayoría de los modelos que existen en la literatura no son demasiado exhaustivos en cuanto a la captura del conjunto completo de las dinámicas que intervienen en el comportamiento de los helicópteros. Gran parte de los resultados que existen hasta ahora están basados en modelos linealizados [7], [12]. La mayoría considera modelos lineales invariantes en el tiempo en el caso de una maniobra de *hover* mientras que en vuelo *forward* las dinámicas son variantes en el tiempo [2], [7] con un periodo igual al periodo de revolución del rotor principal. Por otro lado, existen descripciones no lineales de la dinámica de helicópteros utilizando modelos de orden reducido [11] o sin tener en cuenta la interacción del rotor principal con el fuselaje [22].

La estructura del artículo incluye en las secciones 2 y 3, los aspectos de la arquitectura de los helicópteros actuales. En la sección 4 se describe el entorno de trabajo VehicleSim, utilizado para construir el modelo del helicóptero. En la sección 5 el análisis del acoplamiento entre el aleteo de las palas y el movimiento del fuselaje se presenta. El análisis del

movimiento del fuselaje y su efecto en el aleteo de las palas se muestra en la sección 6. La sección 7 contiene las conclusiones del artículo.

2 MODELO DINÁMICO

2.1 DESCRIPCIÓN DEL MODELO

El modelo a estudiar está constituido por un rotor principal, fuselaje y rotor de cola, tiene 6 grados de libertad (3 translacionales y 3 rotacionales). El rotor principal es articulado, contiene 4 palas unidas por bisagras al eje central.

El modelo desarrollado en este trabajo tiene como punto de partida un modelo previo de rotor principal y fuselaje, donde el rotor es detalladamente modelado y validado [20], [21]. El modelo captura las no-linealidades del sistema y el acoplamiento entre las distintas dinámicas en la misma línea que los modelos desarrollados en [6], [8] o [15], que constituyeron un referente para el desarrollo de un elevado número de simuladores de vuelo de helicópteros, tanto en el ámbito militar como en el académico.

Un helicóptero se puede describir de forma general como un vehículo aéreo que utiliza "alas giratorias" para obtener elevación, propulsión y control. El modelo utilizado está formado por el rotor principal montado encima del fuselaje, y un rotor de cola. Esto se denomina configuración Sikorsky, figura.1:

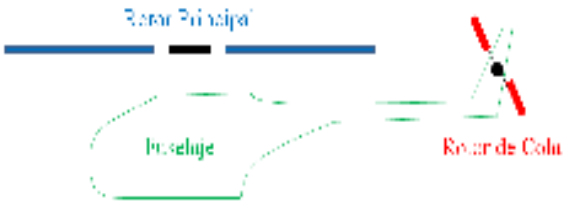


Figura 1: Helicóptero configuración Sikorsky.

2.2 FUSELAJE

Constituye el cuerpo principal del helicóptero que contiene (entre otros componentes) el motor, el piloto y en ocasiones pasajeros y/o carga. Los grados de libertad translacionales (longitudinal y lateral) en el plano horizontal X-Y y traslación vertical en el eje Z. El modelo aquí presentado también tiene tres grados de libertad de rotación en Z (guiñada), Y (alabeo) y X (cabeceo).

2.3 ROTOR PRINCIPAL

Es una de las partes esenciales del sistema ya que genera la sustentación y empuje necesarios para la elevación del vehículo. El rotor principal transfiere las fuerzas y momentos aerodinámicos de las palas en rotación al fuselaje. Un rotor convencional consiste en dos o más palas idénticas igualmente espaciadas que mantienen una velocidad rotacional uniforme Ω gracias al torque del motor transmitido a través del mástil. Una solución común en los rotores de hoy en día es la inclusión de bisagras en la raíz de cada pala para permitir rotaciones alrededor de los ejes normales y paralelos al disco del rotor; las palas están conectadas al mástil con bisagras aleteadoras que permiten el movimiento vertical de las palas. También se incluyen las bisagras de resistencia que permiten el ligero movimiento de la pala en el plano horizontal. La función del movimiento de resistencia es aliviar los excesos de fatiga en la raíz de la pala debido al aleteo y al desequilibrio geométrico que aparece en el vuelo de avance [15].

El cubo que une las bisagras y el mástil; suele ser un anillo o una sección de éste unido al mástil, de manera tal que las palas pueden moverse libremente vertical y horizontalmente (aleteando y resistiendo al avance). Este diseño se conoce como de rotor articulado, ver figura 2.

La más común de todas las bisagras es la de aleteo, que permite a la pala desplazarse verticalmente en una manera similar al batir de las alas de un ave. En algunos casos, la bisagra de aleteo puede ser sustituida por una región de flexibilidad estructural en la raíz de la pala. Normalmente esta bisagra está ubicada a una distancia muy pequeña del cubo, esta distancia se denomina offset.

Una pala en rotación que aletea experimenta momentos de Coriolis en el eje vertical. La bisagra de resistencia contribuye a aliviar la fatiga en la raíz de la pala. El ángulo de resistencia en el plano horizontal se considera positivo cuando es opuesto al sentido de la rotación de rotor.

Por otro lado, el ángulo de ataque de las palas necesita ser controlado para así poder controlar la elevación aerodinámica desarrollada, y en vuelos de avance, permitir que la pala que avanza tenga un ángulo de ataque menor que la pala en retroceso y de esta manera igualar la elevación en el helicóptero. Cuando el ángulo de ataque de todas las palas que conforman el rotor se controla simultáneamente se denomina Control Colectivo, mientras que cambiar el

ángulo de ataque de manera diferente es el llamado Control Cíclico [3], [9]. El control del ángulo de ataque de las palas se ejecuta a través de la conexión de la pala a la placa de unión (figura 2). El piloto controla la altura (colectivo) y ángulo (cíclico) de la placa de unión con un mando *joy-stick*.

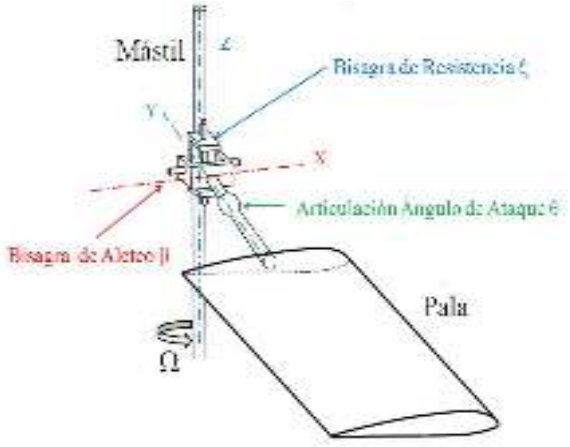


Figura 2: Articulaciones del rotor

2.4 ROTOR DE COLA

La compensación del torque en los helicópteros con un sólo rotor principal es por medio del rotor de cola. Este rotor produce un empuje en sentido contrario al torque generado por el rotor principal, compensando el torque. El rotor de cola gira en un plano vertical alrededor de un eje perpendicular al eje longitudinal del fuselaje. Para que exista equilibrio entre la potencia aplicada al rotor principal durante las fases de vuelo y para poder modificar el giro del helicóptero, el empuje del rotor de cola debe ser variable, esto se logra colocando en él un sistema de variación de paso de las palas, llamado paso colectivo del rotor de cola. Esta variación la comanda el piloto por medio de los pedales.

El rotor de cola también es usado para el control de rumbo en vuelo, además de realizar los cambios en vuelo estacionario.

El ángulo delta-tres se usa a menudo en el rotor de cola para limitar la amplitud con la que la pala asciende/desciende. El ángulo delta-tres consiste en colocar el eje de aleteo con un cierto ángulo que no sea perpendicular al eje longitudinal de la pala. El ángulo es tal que cuando la pala aletea en dirección positiva, el ángulo de paso disminuye, compensando de esta manera la asimetría lateral de sustentación. Por el contrario, cuando el aleteo es en dirección negativa, se evita de este modo un aumento progresivo del ángulo de ataque [4].

3 MODELADO: VEHICLESIM

VehicleSim [9] es un programa de modelado para sistemas compuestos por varios cuerpos. La antigua versión de este programa, AutoSim fue utilizada previamente en numerosos problemas de modelado de dinámica para sistemas mecánicos, especialmente relacionados con la dinámica de vehículos [19], [18], [17] y [14]. Es la base de códigos de simulación comerciales tales como TruckSim, CarSim y BikeSim [9].

VehicleSim es un conjunto de macros LISP que permiten la descripción de sistemas mecánicos de varios cuerpos, con la posible adición de subsistemas no-mecánicos. Las reglas sintácticas de VehicleSim son sencillas. La salida del programa puede obtenerse en varios formatos: (a) fichero .rtf que contiene las ecuaciones simbólicas del sistema descrito, (b) un programa en "C" conteniendo ficheros con los parámetros del modelo y comandos de control de la simulación, o (c) fichero de MATLAB conteniendo las ecuaciones de estado del sistema y la representación matricial A, B, C y D para análisis lineal. Una vez que el modelo se ha construido, el fichero resultante es independiente de VehicleSim y puede ejecutarse tantas veces como se desee sin necesidad de modelar o compilar de nuevo.

4 ESTRUCTURA DEL PROGRAMA

Los comandos de VehicleSim se utilizan para describir los componentes del helicóptero. El sistema modelado consiste en tres subsistemas: rotor principal, fuselaje y rotor de cola. La estructura del programa es "parental" y está definida de acuerdo a las ligaduras físicas y articulaciones existentes entre los cuerpos que conforman el sistema. Existe un cuerpo principal (fuselaje) del que dependen el resto de subsistemas o "hijos", en este caso rotor principal y rotor de cola. Cada uno de los rotores ha sido modelado como un cuerpo rígido dotado de los correspondientes grados de libertad con respecto al fuselaje. Las palas son definidas como componentes "hijos" de los rotores, y a su vez tienen grados de libertad con respecto al "padre" o rotor. Las palas son rígidas y se consideran homogéneas. El caso de palas con flexibilidad no ha sido considerado en este trabajo.

4.1 DESCRIPCIÓN PARAMÉTRICA DEL MODELO

El rotor principal consta de 4 palas igualmente espaciadas entre sí que giran con una velocidad rotacional Ω.

En las simulaciones la velocidad rotacional es $\Omega = 44.4$ rad/s. Las palas del rotor principal tienen una masa $m_p = 31.06$ kg. La longitud de las mismas es $R = 4.91$ m. El centro de masas de la pala es $Y_p = 2.8846$ m. Los momentos de inercia de las palas son $I_{px} = I_{pz} = 62.4$ Kg m^2 y $I_{px} = 0.078864$ Kg m^2. La bisagra de aleteo tiene un muelle cuya constante es $K_a = 46772$ Nm/rad y la bisagra de retardo tiene un muelle de constante $K_r = 314938$ Nm/rad y un amortiguador cuyo coeficiente es $D_r = 349.58$ Nms/rad. La altura a la que el rotor está situado por encima del fuselaje es $h = 1.48$ m. El rotor principal tiene activado el grado de rotación en el eje Z. La masa del helicóptero es $M_h = 2200$ Kg.

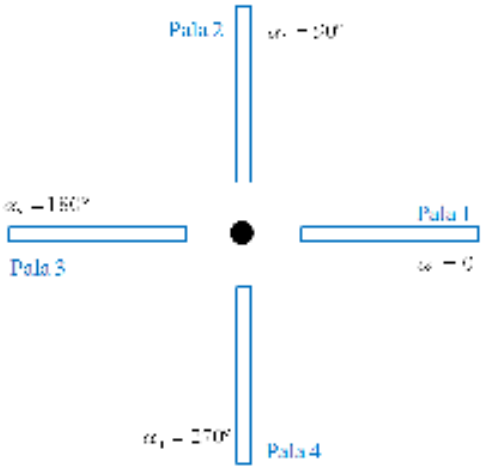


Figura 3: Distribución de las palas del rotor principal

El fuselaje tiene activados 6 grados de libertad, 3 de translación a lo largo de los ejes X, Y, Z y 3 rotaciones alrededor de estos mismos ejes. La aceleración de la gravedad en las simulaciones está desactivada ya que estamos interesados en estudiar únicamente las dinámicas de los subsistemas del helicóptero, así como las fuerzas y las velocidades que aparecen como resultado de las interacciones entre ellos.

5 ANÁLISIS DEL ACOPLAMIENTO ENTRE EL ALETEO DE LAS PALAS Y EL MOVIMIENTO DEL FUSELAJE.

Se sabe que el acoplamiento del movimiento del fuselaje en el eje vertical y el movimiento de aleteo de las palas del rotor principal se produce como consecuencia del muelle que existe en el grado de libertad de aleteo de las palas. Siendo su fin producir una fuerza restauradora. El muelle produce un movimiento oscilatorio a lo largo del eje vertical del helicóptero como consecuencia de la inercia de las palas cuando esta desplazadas de su plano de equilibrio y del momento centrífugo que experimentas las mismas. Si el valor de muelle es cero el fuselaje y el movimiento de aleteo permanecen alineados en su posición original, es decir, sin acoplamiento entre los dos sistemas [15].

Con el objetivo de verificar este acoplamiento en el modelo propuesto, se le permite al fuselaje solamente el grado de libertad a lo largo del eje vertical el resto de los grados de libertad translacionales y rotacionales permanecerán desactivados. El rotor principal tiene los modos de aleteo y de resistencia, aunque este último modo no tiene influencia en el acoplamiento que se va a estudiar. El ángulo de ataque de las palas se mantiene sin efecto sobre el sistema. Por otro lado, el rotor de cola tanto sus propiedades estructurales como cinéticas no están activadas, con el objetivo de central el estudio y los resultados sólo entre el fuselaje y el rotor principal.
Un estudio de la estabilidad del sistema se puede realizar por medio del lugar de las raíces. Las condiciones bajos la cuales la estabilidad es estudiada son en el equilibrio, el resultado se muestra en la figura 4.

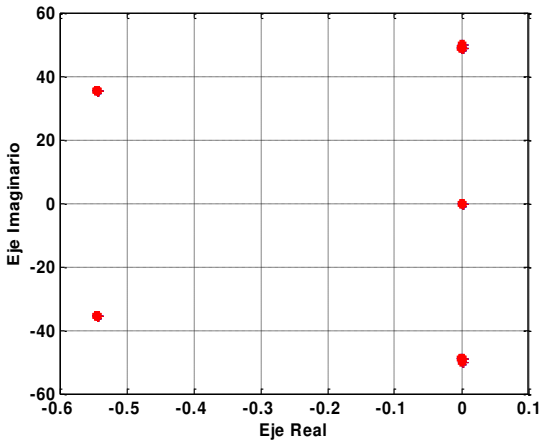


Figura 4: En la figura se muestra el lugar de las raíces para el fuselaje y el rotor principal.

Como se puede ver, hay tres modos de oscilación. El primero claramente diferenciado de los otros dos, situado en la zona de estabilidad del lugar de las raíces, con un valor en el eje real de -0.54. Este está asociado al modo de resistencia de las palas del rotor principal. Los otros dos modos están situados en el eje real sobre el origen del diagrama, el cero. Como se puede ver en la figura 5, donde se muestra estos dos modos de un modo más detallado, los dos modos tienen valores de 50 y 48.8 en el eje imaginario. Estos son modos asociados al movimiento vertical de fuselaje y al movimiento de aleteo de las palas del rotor principal, respectivamente.

Por otro lado, el comportamiento oscilatorio del sistema se puede obtener con una pequeña perturbación sobre el sistema, esta se va realizar

considerando un ángulo inicial de aleteo de 0.0175 rad sobre cada una de las palas del rotor principal, esta condición es suficiente para producir el acoplamiento deseado en el sistema. La gravedad no se considera, ya que estamos interesados en estudiar el movimiento sin ninguna influencia externa sobre el sistema, como se mencionó previamente.

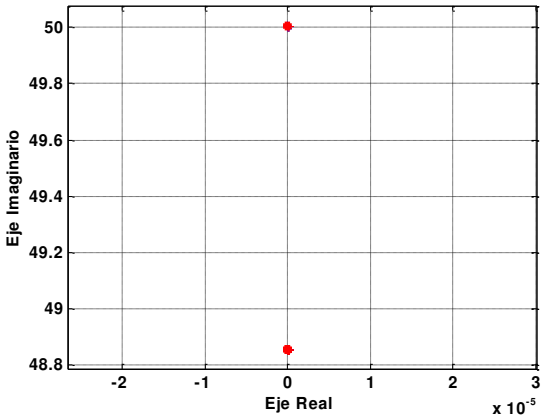


Figura 5: En la figura, se muestra un detalle de la figura previa donde se aprecian mejor los dos modos de oscilación que están en el cero del eje real.

Análisis Modal

Con el objetivo de estudiar los modos de oscilación del sistema se va hacer uso del análisis modal, lo que permitirá entender el acoplamiento dinámico del fuselaje y del rotor principal.

Este método establece que los autovalores de un sistema representan el cuadrado de la frecuencia natural con la que el sistema vibra u oscila, y los autovectores representan la forma del modo, así como las amplitudes relativas de las masas cuando el sistema vibra con una u otra frecuencia. En otras palabras, los autovectores describen como las diferentes partes de un sistema vibran u oscilan unas respecto a otras [13].

El análisis matemático de la dinámica de sistemas lineales permite afirmar que el autovalor λ de la matriz A satisface la ecuación:

$$(A - \lambda I)\phi = 0 \quad (1)$$

La solución viene dada por:

$$\det(A - \lambda I) = 0 \quad (2)$$

Los autovectores son vectores ϕ_i conforme a la ecuación (1) para cualquier autovalor λ_i :

$$A\phi_i = \lambda_i \phi_i \quad (3)$$

Los elementos del autovector dan una medida de la actividad relativa de las variables de estado cuando uno en particular es excitado, es decir, el k-esimo elemento del autovector ϕ_i mide la actividad de la variable x_i en el i-esimo autovalor [1].

En nuestro caso, el modelo matemático asociado al sistema tiene 19 variables de estado, $q(t) = \{x_1\ x_2\ x_3\ ...\}^t$ donde la primera variable x_1 describe el movimiento vertical de translación del fuselaje, x_2 describe la rotación del rotor principal, x_3, x_5, x_7 y x_9 describen el movimiento de aleteo de la primera, segunda, tercera y cuarta pala respectivamente. x_4, x_6, x_8 y x_{10} describen el movimiento de resistencia de la primera, segunda, tercera y cuarta pala respectivamente. x_{11} describe la velocidad a lo largo del eje vertical del fuselaje. x_{12}, x_{14}, x_{16} y x_{18} describen las velocidades del aleteo de la primera, segunda, tercera y cuarta pala. x_{13}, x_{15}, x_{17} y x_{19} describen las velocidades de retraso de la primera, segunda, tercera y cuarta pala.

Los autovalores se muestran en la tabla 1.

Tabla 1: Autovalores del sistema.

Nombre	Autovalor
λ_1	0
λ_2	0.0000 + 50.0070 i
λ_3	0.0000 + 48.8564 i
λ_4	0.0000 + 48.8564 i
λ_5	0.0000 + 48.8564 i
λ_6	-0.5448 + 35.3776 i
λ_7	-0.5448 + 35.3776 i
λ_8	-0.5448 + 35.3776 i
λ_9	-0.5448 + 35.3776 i
λ_{10}	0

Los autovalores complejos conjugados han sido omitidos en la tabla 1.

La matriz de autovectores puede ser calculada, de este modo cada autovalor puede ser asociado con su autovector correspondiente con el objetivo de identificar los diferentes modos de oscilación del sistema. El k-esimo elemento del autovector ϕ_2 mide la actividad de la variable de estado x_i en el autovalor λ_2 [5] [16].

Así, se puede analizar como el fuselaje y el aleteo de las palas vibran unos con respecto a los otros. La

frecuencia natural del autovalor: $\lambda_2 = 0.000 + 50.0070\,i$ es 5.003 rad/s. Si el módulo del autovector asociado a este autovalor se representa, ver figura 7, como se puede ver las variables de estado asociadas al movimiento del fuselaje (x_{11}) y las velocidades de rotación de aleteo (x_{12}, x_{14} , x_{16}, x_{18}) muestran su actividad en este modo de oscilación, exhibiendo el acoplamiento existente entre ambos sistemas.

Por otro lado, la frecuencia natural del autovalor: $\lambda_3 = 0.000 + 48.8564\,i$ es 4.9425 rad/s. Si el módulo del autovector asociado a este autovalor se representa, ver figura 8, como se puede ver las variables de estado asociada con la velocidad de rotación de la segunda pala (x_{14}) tiene una mayor influencia que el resto de los aleteos de las demás palas (x_{12} x_{16}, x_{18}). El fuselaje no muestra ninguna influencia en este modo de oscilación. Así que, este modo tiene su principal contribución en el aleteo de las palas.

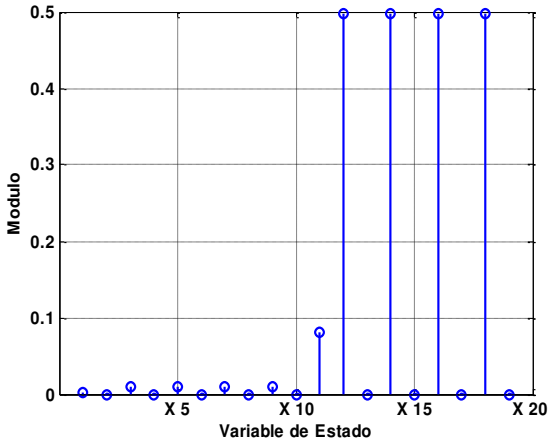


Figura 7: Módulo del autovector asociado al autovalor $\lambda_2 = 0.000 + 50.0070\,i$.

Las frecuencias naturales de los autovalores: λ_4 y λ_5 están también asociadas al aleteo sin mostrar estas ninguna influencia del fuselaje en su oscilación. Finalmente, las frecuencias naturales de los autovalores: λ_6, λ_7, λ_8 y λ_9 con valores $-0.5448 + 35.3776$ son 4.2383 rad/s están asociadas al retraso de las palas. No mostrando en ningún caso acoplamiento con el movimiento vertical del fuselaje

En este punto, conviene considerar que el primer modo asociado al autovector λ_2 mostrado en la figura 7 presenta un movimiento acoplado entre el fuselaje y el aleteo de las palas, mientras que el resto modos están asociados con el movimiento de las palas solamente.

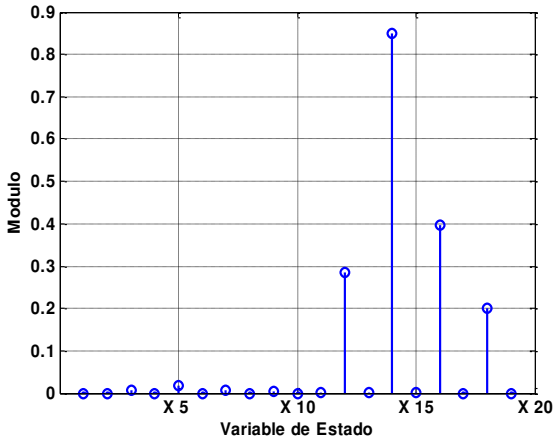


Figura 8: Módulo del autovector asociado a el autovalor $\lambda_3 = 0.000 + 48.8564\,i$.

6 ANÁLISIS DEL MOVIMIENTO DEL FUSELAJE Y SU EFECTO EN EL ALETEO DE LAS PALAS.

Es conocido que cuando el eje del rotor principal esta rotado alrededor de los ejes X e Y, cabeceo y alabeo, las palas experimentan una aceleración giroscópica adicional causada por la velocidades angulares perpendiculares ω_p, ω_r y la velocidad principal del rotor principal Ω. La solución para el movimiento de aleteo viene dada por la expresión:

$$\beta = \beta_{1C}\cos(\Omega t) + \beta_{1S}\sin(\Omega t) \quad (4)$$

donde:

$$\beta_{1C} = \frac{2}{\Omega\left(\lambda_\beta^2 - 1\right)}\,\omega_p \quad (5)$$

$$\beta_{1S} = \frac{-2}{\Omega\left(\lambda_\beta^2 - 1\right)}\,\omega_r \quad (6)$$

Estas soluciones representan la influencia de los efectos giroscópicos cuando cualquier masa esta rotando fuera de su plano. β_{1C} es la inclinación longitudinal en respuesta a la velocidad de cabeceo y β_{1S} es la inclinación lateral a en respuesta a la velocidad de alabeo [15].

Con el objetivo de comprobar este resultado teórico, se va realizar el siguiente estudio, en el cual el retraso y el ángulo de ataque de las palas no se consideran, es decir, las palas sólo tienen permitido el grado de libertad de aleteo con el objetivo de observar la influencia del movimiento del fuselaje.

Se van a considerar solo los grados de libertad de rotación del fuselaje, es decir, cabeceo, alabeo y giñada, los translacionales son desactivados con el fin de poder verificar la teoría. Así de este modo, si se aplica un momento al fuselaje alrededor el eje Y (alabeo) en forma de una función escalón cuyo valor es 1750 Nm, este producirá movimientos de cabeceo y alabeo en el fuselaje, esperando que estos tengan algún tipo de influencia en el movimiento de aleteo de las palas del rotor principal. Esto puede ser comprobado haciendo uso de la expresión (4), que permite obtener los efectos que el cabeceo y el alabeo del fuselaje tienen sobre el movimiento de aleteo. Para el cálculo se ha considerado $\lambda_\beta^2 = 1.2$ siendo este un parámetro estándar existente en la bibliografía. Y aplicable a un modelo de helicóptero como el descrito en este trabajo [15].

Dadas la predicciones que marca la teoría es de esperar que el aleteo de las palas del rotor principal muestren un comportamiento influenciado por el movimiento del fuselaje. En la figura 9, se representa el aleteo de las palas en azul y en rojo el aleteo que se espera obtener como resultado del cabeceo y alabeo del fuselaje sobre las palas.

De este modo, se ha comprobado la influencia que el movimiento del fuselaje tiene sobre el aleteo de las palas del rotor principal, cuando este experimenta movimientos de cabeceo y alabeo.

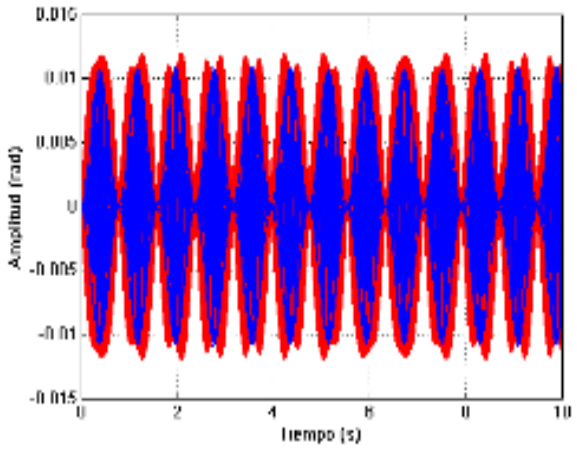


Figura 9: En rojo, se muestra el movimiento de aleteo calculado con la ecuación (4) debido al cabeceo y alabeo del fuselaje. En azul, se muestra el aleteo que muestran las palas.

7 CONCLUSIONES

En este artículo se ha estudiado la mutua interacción que el fuselaje y el aleto de las palas del rotor principal de un helicóptero ejercen entre sí. Esta influencia es importante a la hora de abordar la estabilidad de un sistema tan complejo, y a la vez tan inestable como es el vuelo de un helicóptero. Por este motivo, las pruebas se han realizado sin considerar fuerzas externas sobre el sistema, como por ejemplo la gravedad. Con el fin de hacer un estudio y análisis de las propiedades estructurales del sistema en sí.

Se ha presentado un modelo que representa el comportamiento dinámico de un helicóptero con configuración Sikorsky, y que es capaz de capturar los movimientos transmitidos desde el rotor principal al fuselaje y viceversa. El modelo ha sido implementado en VehicleSim siendo este un programa que permite trabajar y definir los sistemas como composición de varios cuerpos y ligaduras.

La estabilidad del sistema formado por el rotor principal y el fuselaje se ha estudiando haciendo uso del lugar de las raíces, mostrando los resultados la estabilidad en el sistema cuando se consideran condiciones de equilibrio. Además, los modos de oscilación han sido analizados por medio del análisis modal, mostrando estos, un acoplamiento entre el movimiento vertical del fuselaje y el aleteo de las palas del rotor principal.

Por otro lado, la influencia de cabeceo y alabeo del fuselaje sobre el aleteo de las palas ha sido analizado. En este caso, el estudio se ha realizado haciendo uso de resultados teóricos que han sido posteriormente verificados con los resultados de las simulaciones.

Finalmente decir que, un modelo de helicóptero ha modelado con VehicleSim programa especializado en el modelado de sistemas mecánicos. Los estudios realizados de las propiedades dinámicas entre el fuselaje y el rotor principal han permitido comprobar que los resultados están acorde a lo esperado.

Agradecimientos

Este trabajo ha sido financiado por el proyecto del Ministerio de Ciencia e Innovación del Gobierno de España DPI 2010-20751-C02-02, y por la beca de la Agencia Canaria de Investigación, Innovación y Sociedad de la Información del Gobierno de Canarias.

Referencias

[1] Balmes, E. Modes and Regular Shapes. How toExtend Component Mode Synthesys Theory. 2005. Proceedings of the XI DINAME, 28th February-4th March, Ouro Preto- MG- Brazil

[2] Bittanti, S., Lovera, M., On the zero dynamics of helicopter rotor loads, European J. Contr., 2, 1,57–68, 1996.

[3] Bramwell, A, R, S., Helicopter Dymanics. 1976. Butterworth Heinemann. Second Edition.

[4] Brea, G, L., Helicópteros. EAA Argentina.

[5] Calderon-Guizar, J.G. Modal Analysis of Electromechanical Oscillations in Electrical Power Systems. 2008. RIIT Vol.IX. Nº.4. 2008 313-317, ISSN 1 405-7743 FI-UNAM

[6] Decker, W, A., Talbot, P, D., Tinling, B. E., Chen, R, T, N., A mathematical model of a single main rotor helicopter for piloted simulation. Technical memorandum NASA 84281, NASA Ames Research Center, Moffett Field, California, September 1982.

[7] Garrad, W., Low, E. Eigenspace design of helicopter flight control systems. Technical report, Dept. of Aerospace Engineering and Mechanics, Univ. of Minnesota, 1990.

[8] Heffley, R, K., Mnich, M, A., Minimum complexity helicopter simulation math model. Contractor report NASA 177476, Aeroflightdynamics Directorate, U.S. Army Research and Technology Activity (AVSCOM), April 1988.

[9] http://www.carsim.com/products/supporting/vehiclesim/index.php

[10] Johnson, W., Helicopter Theory. Princeton, NJ: Princeton Univ. Press, 1980.

[11] Kaloust, J., Ham, C., Qu, Z., Nonlinear autopilot control design for a 2-DOF helicopter model, Control Theory and Applications, IEE Proceedings 1997, 144, 6, 612-616.

[12] Kienitz, K., Wu, Q., Mansour, M., Robust stabilization of a helicopter model. Proceedings of the 29th CDC, 1990, 2607-2612.

[13] D.V. Martinez & D.R. Vega. Analisis Modal de Sistemas de Potencia ante Variaciones en su Modelado y Operacion. 2008. IEEE Mexican Section. RVP-AI2008. Acapulco. 6-12 of July.

[14] Mousseau, C, W., Sayers, M, W., Fagan, D. J., Symbolic quasi-static and dynamic analyses of complex automobile models, The Dynamics of Vehicles on Roads and on Railway Tracks (G. Sauvage ed.), Swets and Zeitlinger, Lisse, 1992, 446-459.

[15] Padfield, G, D., Helicopter flight dynamics: the theory and application of flying qualities and simulation modeling. Oxford, Blackwell, 1996.

[16] B.J. Schwaz & M. H. Richardson. Experimental Modal Analysis. 1999. CSI Reliability Week, Orlando, FL, October

[17] Sharp, R, S., Evangelou, S., Limebeer, D,J,N., Advances in the modeling of motorcycle dynamics, Multibody System Dynamics 12(3), 2004, 251-283.

[18] Sharp, R, S., Evangelou, S., Limebeer, D,J,N., Multibody aspects of motorcycle modeling with special reference to Autosim, Advances in Computational Multibody Systems, J. G. Ambrosio (Ed.), Springer-Verlag, Dordrecht, The Netherlands, 2005, 45-68.

[19] Steele, Jr., Guy, L., COMMON LISP: The language. Digital Press. Burlington, Mass, 1984.

[20] Tomas-Rodriguez M., Sharp R., Automated Modeling of Rotorcraft Dynamics with Special Reference to Autosim, Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference, 2007, Pages: 974-979.

[21] Tomas-Rodriguez, M., Multibody modeling of Rotorcraft Dynamics, MPhil Thesis, 2008, Aeronautical Engineering Department, Imperial College, London, UK.

[22] Verdult, V., Lovera, M., and Michel Verhaegen. Identification of linear parameter-varying state-space models with application to helicopter rotor dynamics, International Journal of Control, 77, 13, Sep 2004, 1149-1159.

TENSIONES INDUCIDAS EN CABLES DE ALTA TENSIÓN

Miguel Ángel Calafat Torrens
Red Eléctrica de España, mcalafat@ree.es

Matilde Santos Peñas
Universidad Complutense de Madrid, msantos@ucm.es

Resumen

En este trabajo se presenta una aproximación novedosa al modelado de cables subterráneos de alta tensión. Se ha incorporado el cálculo de las tensiones inducidas debidas a las pantallas de los cables, contemplando distintos tipos de conexiones de éstas. También se han tenido en cuenta los cortocircuitos monofásicos que se generan según las distintas configuraciones. Se ha implementado una herramienta de simulación que permite generar de forma automática múltiples circuitos acoplados. Este desarrollo ha sido validado con datos teóricos, siendo los resultados muy parecidos a los esperados.

Palabras Clave: Modelado, simulación, cables subterráneos, acoplamiento múltiple, circuitos, alta tensión.

1 INTRODUCCIÓN

Cada vez son mayores las dificultades que se encuentran para la construcción de líneas aéreas de alta tensión. Varios son los factores que pueden influir en esto, tales como su mayor impacto visual, la fuerte oposición social, la dificultad de llevar a cabo las pertinentes expropiaciones de terreno con las restricciones de tiempo y coste que imponga el proyecto, etc. Por estos motivos en los últimos años se está produciendo una mejora en la tecnología de fabricación e instalación de los cables subterráneos aislados de alta tensión [10].

Sin embargo esta solución también presenta desventajas, como que el coste de una línea subterránea puede ser hasta 20 veces mayor que la aérea para una misma longitud y capacidad de transporte, y también puede presentar algunos problemas ambientales por obstrucción de escorrentías y por afección a animales del hábitat subterráneo.

Además, desde un punto de vista técnico, los cálculos eléctricos en las líneas subterráneas son muy complejos y presentan una serie de características propias que los hacen muy diferentes de aquellos realizados para las líneas aéreas [2, 9].

El objetivo de este trabajo es modelar matemáticamente y simular las tensiones inducidas en las pantallas de cables aislados para cualquier tipo de caso y configuración, con el objetivo de poder visualizar tanto magnitudes como ángulos de desfase entre ellas. Para ello se ha desarrollado una herramienta de simulación que permite desglosar las diferentes componentes de estas tensiones y poder visualizar, en tiempo de diseño, en qué medida afectan los diferentes parámetros (sección de los cables, resistencias de puesta a tierra, configuración de la conexión, longitudes de los tramos, etc.) a las citadas tensiones.

Además, la principal contribución de este trabajo es que permite acoplar múltiples circuitos de forma automática, lo que los pocos programas que lo hacen es a costa de la laboriosa tarea de definir cada acoplamiento manualmente [1].

Para conseguir estos objetivos se ha generado un modelo eléctrico que emula el funcionamiento de los diferentes casos a partir de los datos de la línea introducidos por el usuario. El modelo a simular se ha diseñado y realizado con SPICE [6].

La estructura del artículo es la siguiente. En la sección 2 se describen los componentes de un cable eléctrico aislado de alta tensión. En la sección 3 se exponen los diferentes tipos de posibles conexiones entre pantallas. En las secciones 4 y 5 se presenta el cálculo y modelado de las tensiones inducidas. A continuación se muestra la herramienta de simulación desarrollada y se discuten los resultados obtenidos.

2 DESCRIPCIÓN DEL SISTEMA

Los cables aislados presentan una pantalla envolvente de material conductor que aísla completamente el campo eléctrico, pero que al ser cortado por el campo magnético que generan las corrientes alternas induce tensiones y/o corrientes en las citadas pantallas. Estas tensiones y corrientes deberán ser cuidadosamente calculadas dado que influyen tanto en la capacidad de transporte de la

línea como en el diseño de las protecciones a instalar. Además, dependiendo del tipo de conexión de las pantallas a tierra, se producirán corrientes que a su vez también inducirán tensiones en las pantallas próximas [4].

Puesto que en líneas paralelas cercanas las corrientes de una línea inducen tensiones y/o corrientes en la otra, el modelo a generar deberá ser dinámico; es decir, el modelo para dos líneas paralelas no consiste únicamente en replicar dos veces el modelo de una línea, ya que una línea se comporta de forma diferente si tiene otra al lado, o si tiene dos, o tres, etc.

En resumen, al ser las pantallas de material conductor, y ser atravesadas por campos magnéticos, se produce corriente alterna que genera corrientes y/o tensiones inducidas. Estas tensiones dependen además del tipo de conexión de las pantallas y de la distancia entre las líneas [5].

Los cables eléctricos se pueden clasificar según varios criterios tales como la tensión de servicio, su utilización, el número de fases y el tipo de aislamiento. En este texto se hará especial referencia a cables unipolares con aislamientos de XLPE para tensiones mayores o iguales a 66 kV y para uso subterráneo (en contraposición al uso submarino).

La composición general de estos cables puede verse en la Figura 1.

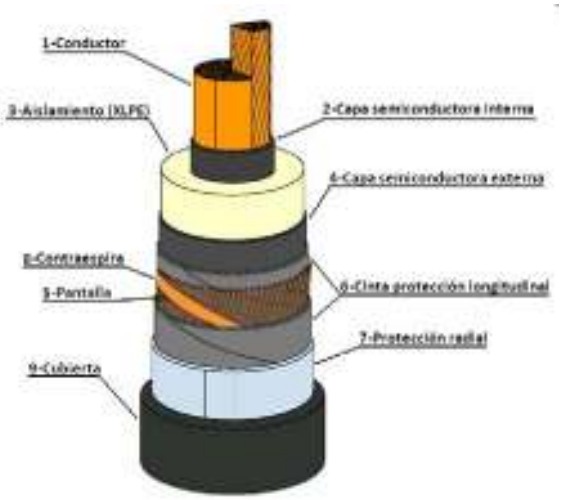


Figura 1: Componentes de un cable eléctrico aislado de alta tensión

El conductor es la parte central de los cables aislados. Puede ser de cobre o de aluminio. Aparte de la evidente función eléctrica de transportar la corriente, también realiza una función mecánica consistente en soportar los esfuerzos longitudinales del tendido. Tres fenómenos eléctricos son especialmente notables en el comportamiento del conductor: las pérdidas óhmicas, el efecto piel y el efecto proximidad.

Aislamiento: Los cables aislados de alta tensión pueden tener dos tipos de aislamiento según sean encintados o extruidos. Su función, como su propio nombre indica, es aislar el conductor (sometido a alta tensión) de la pantalla metálica (puesta a potencial de tierra). El aislamiento debe resistir el campo eléctrico tanto en régimen nominal como en régimen transitorio (sobretensiones transitorias).

La principal característica del aislamiento de un cable es el valor de su rigidez eléctrica, medido en kV/cm. El campo eléctrico máximo al que esté sometido el aislamiento debe ser menor que su rigidez eléctrica para poder soportar la tensión de servicio.

Debido al aislamiento se producen unas pérdidas de la energía transportada por los cables llamadas pérdidas dieléctricas, que dependen de las dimensiones del conductor, del tipo de aislamiento y del cuadrado de la tensión. El valor de estas pérdidas dieléctricas es mucho menor para aislamiento XLPE que para aislamientos de papel.

En este punto cabe destacar, aunque no se trate de una característica propia del aislamiento como tal, que el hecho de tener dos materiales conductores (el conductor y la pantalla) separados por un material dieléctrico (el aislamiento) constituye un condensador eléctrico y deberá ser modelado como tal para considerar este efecto. En la práctica los propios fabricantes ofrecen este dato en sus hojas de características.

Pantalla: es el elemento metálico que se encuentra entre el aislamiento (a través de la capa semiconductora externa) y la cubierta exterior en los cables subterráneos. Suele estar rodeada de otros elementos que sirven para la protección de entrada de agua al aislamiento. Además tiene otras funciones como anular el campo eléctrico en el exterior del cable, con lo cual el campo eléctrico sobre el aislamiento es radial. También es un conductor activo para las corrientes capacitivas que atraviesan el aislamiento, las inducidas por los campos magnéticos de cables cercanos, y la corriente homopolar, drenándolas a tierra. Esta función permite en la práctica dimensionar la pantalla metálica. Además esto implica la necesidad de conectar esta pantalla a tierra, y de estudiar atentamente el diseño de esta conexión.

Por último, otras de sus funcionalidades son contribuir a la protección mecánica ante agresiones externas y, según el diseño, dar estanqueidad radial a la entrada de agua hacia el aislamiento.

3 TIPO DE CONEXIÓN DE PANTALLAS

La corriente alterna que circula por un conductor genera una inducción magnética B variable que es concéntrica al conductor.

Al actuar un campo magnético B sobre una superficie aparece un flujo magnético φ cuya magnitud depende del área de esta superficie y el ángulo de incidencia del campo magnético sobre ella. La Ley de inducción electromagnética establece que el voltaje inducido en el perímetro de una superficie dada es directamente proporcional a la rapidez con que cambia el flujo magnético que atraviesa la citada superficie [3].

Evidentemente, si se induce tensión en un conductor y éste forma parte de un circuito cerrado, entonces circularán corrientes; de ahí que el tipo de conexión de las pantallas de los cables juegue un papel determinante en las tensiones y corrientes que podemos encontrar.

En la industria hay varias maneras normalizadas de realizar las conexiones a tierra de las pantallas [7]. A partir de estos esquemas básicos se pueden formar combinaciones de unos y otros. Se describen a continuación los tipos fundamentales de conexión que se han implementado en el modelo.

3.1 CONEXIÓN TIPO *SOLID BONDING* (SB) O *BOTH ENDS*

Este tipo de conexión de las pantallas a tierra rara vez es usado en líneas de alta tensión. Es más propio de líneas de media tensión (hasta 45 kV) y ocasionalmente puede aparecer en alguna línea corta de 66 kV.

Como su propio nombre indica, *both ends*, se ponen las pantallas a tierra en ambos extremos de la línea o del tramo a considerar.

Atendiendo al hecho de que el circuito formado por las pantallas y tierra es un circuito cerrado, este tipo de conexión de pantallas va a tener corrientes circulando permanentemente por ellas y, por tanto, va a tener que asumir una pérdida de capacidad de transporte que puede llegar a ser importantísima. En líneas de alta tensión estas pérdidas podrían llegar a ser del orden del 50 % de la corriente nominal de la línea.

3.2 CONEXIÓN TIPO *SINGLE POINT* (SP)

Este tipo de conexión de pantallas consiste en conectar directamente a tierra un extremo y dejar el otro abierto. En realidad el extremo abierto no es tal; se conecta a tierra a través de descargadores. Cabe destacar que es importante en este tipo de conexión de pantallas disponer de un cable de conexión equipotencial (ecc) conectado a tierra en los extremos, denominado cable de acompañamiento, con el fin de minimizar corrientes de circulación en régimen permanente y, por tanto, pérdidas.

La principal ventaja de este tipo de conexión de las pantallas a tierra es que no circula corriente por ellas en régimen permanente porque están todas en circuito abierto, lo que hace que no haya pérdida de capacidad de transporte.

Su principal inconveniente es que se genera una tensión inducida en el lado de los descargadores. Esta tensión es proporcional a la distancia de la línea y en régimen permanente podrían ser de magnitud tal que fueran peligrosas para las personas. Los tramos SP pueden tener unas longitudes máximas del orden de 600 m.

En caso de cortocircuito, las tensiones que se inducen pueden llegar a ser peligrosas incluso para la cubierta del cable (que habitualmente suele estar diseñada para soportar 10 kV en cables de transporte de 66, 132 y 220 kV). Este es el motivo por el cual se protege la línea con descargadores [11].

Una variación de la conexión tipo *single point* es la llamada *mid point bonding*. Básicamente es lo mismo que poner dos tramos SP unidos por la conexión a tierra. La ventaja es evidente ya que se puede conseguir el doble de longitud que en un *single point* con la misma magnitud de tensión inducida. Sea como sea, adolecerá de los mismos inconvenientes en cuanto a las altas tensiones inducidas en caso de cortocircuito que limitan la longitud máxima de los tramos.

Es habitual que los tramos SP o *mid point bonding* se pongan en los extremos de líneas largas como continuación de otro tipo de conexión (*cross bonding*) que se verá en el punto siguiente.

3.3 CONEXIÓN TIPO *CROSS BONDING* (CB)

Este tipo de conexión de pantallas puede parecer, por una parte, similar al SB. Se trata de conectar a tierra ambos extremos de un bloque CB. Cada bloque CB está formado por tres tramos elementales de igual longitud (en la medida de lo posible). En el punto de unión de esos tramos se realiza una trasposición de las pantallas de manera que la pantalla que durante el primer tramo elemental estaba con la fase 1 pase a estar con la fase 2; la pantalla que estaba con la fase 2 pase a estar con la fase 3; y la pantalla que estaba con la fase 3 pase a estar con la fase 1 (Figura 2). Es

habitual llamar a las fases R, S y T, o bien 0, 4 y 8, como indicación del desfase en unidades horarias, aunque esto depende del criterio de cada compañía. En este texto se llamarán 1, 2 y 3 para mayor simplicidad.

Esta transposición de pantallas se llevará a cabo en las dos uniones del tramo CB. Si los tramos elementales son iguales (y las corrientes también), entonces la corriente inducida en la pantalla 1 del tramo 1 se compensará con la corriente inducida en esta misma pantalla -recordemos que se traspone- en el tramo 2, y se compensará definitivamente con la tensión inducida en el tramo 3, suponiendo una disposición al tresbolillo.

En la figura 2 puede verse una representación de un bloque *cross bonding* (CB) con sus tres tramos elementales según el modelo implementado.

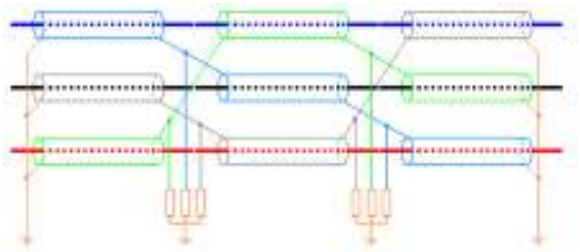


Figura 2: Bloque *cross bonding*

La ventaja del *cross bonding* es evidente si se tiene en cuenta que la tensión resultante es nula (en el caso ideal), de manera que no hay circulación de corriente y por tanto no se pierde capacidad de transporte. Por otra parte, se puede conseguir el triple de longitud que en el caso de una conexión tipo SP con la misma tensión máxima inducida (en régimen permanente).

4 CÁLCULO DE LAS TENSIONES INDUCIDAS

Para el cálculo de las tensiones inducidas se utilizan las ecuaciones empíricas de Carson [5, 8]. En primer lugar hay que conocer las impedancias inducidas, es decir, las impedancias mutuas entre los conductores de fase y las pantallas, y entre las pantallas y la propia de la pantalla. La ecuación de Carson es:

$$Z_{c,p} = \frac{\omega \cdot \mu_0}{2}\left(\frac{1}{4} + j \cdot \frac{1}{\pi} \cdot \ln\left(\frac{D_e}{r_{c,p}}\right)\right) \quad (1)$$

Donde:
$Z_{c,p}$: Impedancia inducida entre conductor y pantalla (Ω/m)
ω : Pulsación (rad./s)
μ_0 : Permeabilidad del vacío ($4 \cdot \pi \cdot 10^{-7}$ H/m)
$r_{c,p}$: Distancia entre conductor y pantalla (m)
D_e : Distancia equivalente del retorno por tierra (m)

Para llevar a cabo estos cálculos es preciso hacer ciertas consideraciones. En primer lugar, el retorno equivalente por tierra, D_e, se calcula mediante la ecuación siguiente.

$$D_e = \frac{1.85}{\sqrt{\frac{\omega \cdot \mu_0}{\rho}}} \quad (2)$$

Siendo ρ la resistividad del terreno (Ω·m)

Por otra parte, $r_{c,p}$ es la distancia entre el centro del conductor y de la pantalla cuya impedancia de acoplamiento queremos calcular. Si se quiere calcular el acoplamiento entre dos pantallas, la fórmula es la misma ya que la distancia entre sus centros es igual que entre el centro de conductor de una y el centro de la pantalla de la otra.

Para el caso en que se quiera calcular la impedancia mutua entre un conductor y su misma pantalla, dado que la distancia entre sus centros es cero, $r_{c,p}$ valdrá el radio medio geométrico de la pantalla. Este radio medio geométrico de una pantalla (de un conductor hueco) vale prácticamente lo mismo que el radio exterior de la pantalla; por tanto puede utilizarse aquel en sustitución de éste cometiendo un error muy leve.

En caso de que se esté calculando la impedancia propia de un cable de acompañamiento (impedancia mutua con sí mismo), entonces $r_{c,p}$ valdrá el radio medio geométrico del cable de acompañamiento. Este dato puede calcularse multiplicando el radio exterior del conductor por el factor e^(-1/4).

$$RMG = r_{ext} \cdot e^{-1/4} \quad (3)$$

r_{ext} : Radio exterior del conductor (m)
RMG : Radio medio geométrico del conductor (m)

Resumiendo, la fórmula que se utiliza en este trabajo de la impedancia inducida es siempre la misma y en ella está implicada la distancia entre los centros de los dos elementos a tener en cuenta. El caso conductor–pantalla es exactamente igual que el caso pantalla–pantalla. Cuando se calcule el caso pantalla–pantalla del mismo elemento consigo mismo (pantalla1–pantalla1, pantalla2–pantalla2, etc.), entonces la distancia a considerar, en lugar de ser cero, será el radio de la pantalla (lo mismo es aplicable al caso del conductor con su propia pantalla). Para el caso de un cable de acompañamiento, como es un conductor macizo, la distancia a considerar será el radio medio geométrico de conductor.

En el modelado hay que tener especial cuidado cuando se considera la impedancia de un elemento dado (pantalla o cable de acompañamiento), ya que a la impedancia inducida calculada habrá que añadirle la impedancia natural del propio elemento que estará indicada en la hoja de características técnicas. A partir de aquí se realiza el cálculo de las tensiones inducidas, que dependerá del tipo de conexión de las pantallas. No se desarrolla, como tampoco el análisis de circuitos monofásicos por motivos de espacio, pero se han implementado en el modelo.

5 MODELADO DE LAS TENSIONES INDUCIDAS EN CABLES AISLADOS

A continuación se desglosa el modelo que se ha implementado, haciendo hincapié sólo en los elementos principales.

El modelo de un conductor es ampliamente conocido y ha sido desarrollado, por ejemplo, por John J. Grainger y William D. Stevenson en base a la inductancia interna del mismo [3]. Sin embargo en este trabajo se ha utilizado otro enfoque ya que el objetivo es el cálculo de las tensiones inducidas en las pantallas de los cables y su relación (desfase) entre unas corrientes/tensiones y otras. Por lo tanto el punto de partida es la corriente que circula por los conductores y no la tensión de la fuente. Dicho de otra manera, puesto que son las corrientes quienes inducen tensión en las pantallas, no influye la tensión de línea ya que el resultado de la tensión inducida para una misma corriente será prácticamente idéntico.

En definitiva, los conductores se modelarán únicamente por su resistencia óhmica. Por ello el modelo elegido para un conductor es de dos resistencias en serie (Figura 3). Se utilizan dos resistencias en lugar de una sola para así poder tener acceso a un punto en el interior del conductor. Este punto se utilizará para el modelado del condensador que representa la capacidad parásita del cable. La fuente de tensión al inicio del conductor representa un sensor de corriente.

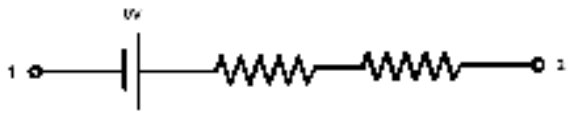


Figura 3: Modelo de conductor utilizado

Para inducir tensión en un conductor dado, sea pantalla o cable de acompañamiento, deberá haberse calculado previamente la impedancia mutua entre el elemento inductor y el elemento que recibe la inducción mediante la fórmula de Carson (1). A partir de aquí se introduce la impedancia calculada y se obtiene, en los bornes de ésta, la tensión inducida.

Finalmente, para inducir la tensión en el elemento receptor bastará con incluir una fuente de tensión dependiente de la tensión en la impedancia mutua calculada (entre nudos 5 y tierra, figura 4) con magnitud de la ganancia igual a la longitud de la línea.

El modelo total del sistema para inducir tensión se muestra en la Figura 4. La fuente de tensión que hace de sensor estaría en un elemento (el conductor), la impedancia mutua con la fuente de intensidad estarían aparte de cualquier otro elemento del circuito, y la fuente de tensión inducida estaría ubicada en un tercer elemento (la pantalla).

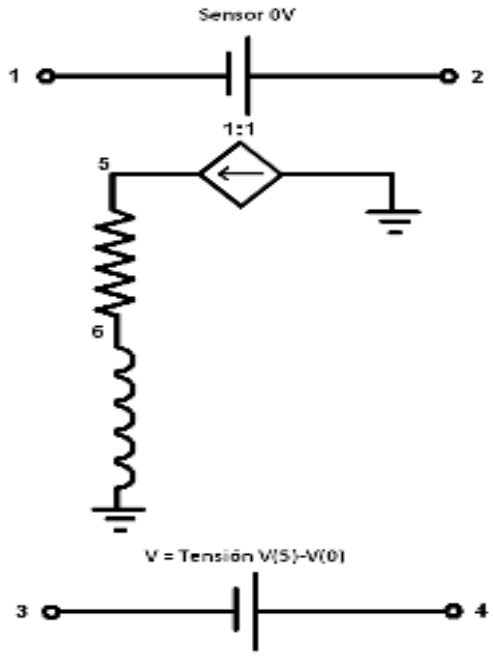


Figura 4 : Modelo eléctrico completo de la tensión inducida en el elemento afectado por la impedancia mutua debido a la corriente que atraviesa el sensor

Los elementos anteriores, conductor y tensión inducida, permiten modelar la pantalla que es una combinación de ellos. Por una parte se utilizarán dos resistencias para modelar el efecto resistivo y dos inductancias para la inductancia propia de la pantalla. Las resistencias a incluir en el modelo deberán ser las resistencias finales; es decir, la resistencia de la pantalla a 50 Hz y a la temperatura de servicio, considerando los factores de corrección que el usuario estime oportuno.

Como en el caso del conductor, se ubicará un sensor en el origen de la pantalla para poder hacer referencia a la corriente que la atraviesa.

Por último habrá que insertar todas las fuentes de tensión que sean necesarias para modelar las tensiones inducidas. En el ejemplo en la Figura 5 se muestra una pantalla de un circuito simple sin cable de acompañamiento. En ella se pueden apreciar las fuentes de tensión que modelan las tensiones inducidas por las corrientes que atraviesan las tres fases (F1, F2 y F3), y las tensiones inducidas por las corrientes que atraviesan las otras dos pantallas (P2 y P3). La inductancia propia ya se modela en las bobinas.



Figura 5: Modelo de una pantalla en un simple circuito sin cable de acompañamiento

En este momento ya estamos en disposición de modelar el cable completo. La capacidad parásita se muestra como un condensador. También hay que determinar el valor de las fuentes de tensión. Esto forma parte del propio modelado y no es, por tanto, un parámetro que se asigne directamente antes de realizar la simulación. Además los circuitos de tensiones inducidas implican ciertas fuentes de corriente de manera que el modelo de un cable depende también de los otros cables que estén cercanos a él.

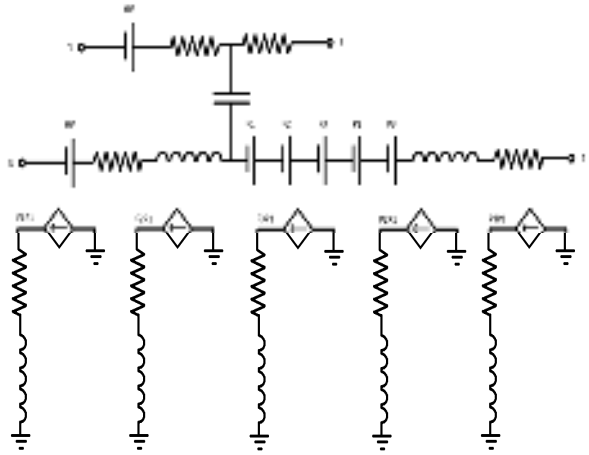


Figura 6: Modelo de cables de potencia en un circuito simple sin cable de acompañamiento

La nomenclatura utilizada representa los dos elementos entre los cuales se ha plasmado la impedancia mutua. Por ejemplo, F1P1 indica la inductancia mutua entre la fase (F) 1 y la pantalla (P) 1. En la Figura 6 se muestran los circuitos correspondientes a las impedancias mutuas entre las fases y las pantallas.

En caso de que en este circuito existiera un cable de acompañamiento, su modelado sería igual al de las pantallas de la figura anterior; es decir, además de tener un sensor, un par de resistencias y un par de inductancias propias, debería tener una fuente de tensión inducida por cada fase y pantalla, por lo tanto seis fuentes de tensión.

Además, todas las pantallas deberían añadir una fuente de tensión más, que modelaría la tensión inducida en la pantalla debida a la corriente por el cable de acompañamiento. Evidentemente, cada fuente de tensión estaría asociada a un pequeño circuito formado por la impedancia mutua entre el cable de acompañamiento y la pantalla en cuestión, y la fuente de corriente dependiente de la corriente por el cable de acompañamiento.

Para un circuito simple con un cable de acompañamiento hay que poner seis fuentes de tensión en cada rama que modela una pantalla. En caso de que tuviéramos un doble circuito, todas las fases, pantallas y cables de acompañamiento del circuito 2 inducirían tensión en todos los elementos (pantallas y cables de acompañamiento) del circuito 1 y viceversa.

Así pues, en el caso de tener dos circuitos con cable de acompañamiento, habría que poner en cada pantalla trece fuentes de tensión (las tres fases de los dos circuitos, las dos pantallas restantes del propio circuito, las tres pantallas del otro circuito y los dos cables de acompañamiento).

En definitiva, el modelado de un circuito en caso de tener un doble circuito no se obtiene sencillamente replicando dos veces el modelo de un simple circuito, sino que cambia y crece con rapidez.

6 HERRAMIENTA DE MODELADO Y SIMULACIÓN

El software desarrollado ha sido generado en VBA para Excel y LT Spice IV [6]. Su uso es muy intuitivo. La pantalla principal se muestra en la Figura 7.

En la parte inferior aparecen las opciones del circuito que se tenga seleccionado; en el panel central están las opciones del tramo (afectando a todos los circuitos del mismo tramo), y en la parte superir de la interface están los parámetros de simulación y cortocircuitos.

Algunos módulos llevan un panel para poder copiar lo que se haya seleccionado en otro tramo y/o circuito. En el botón “Sección circuito” se podrán seleccionar las coordenadas de los cables de potencia y cables de acompañamiento del circuito actual en el tramo actual.

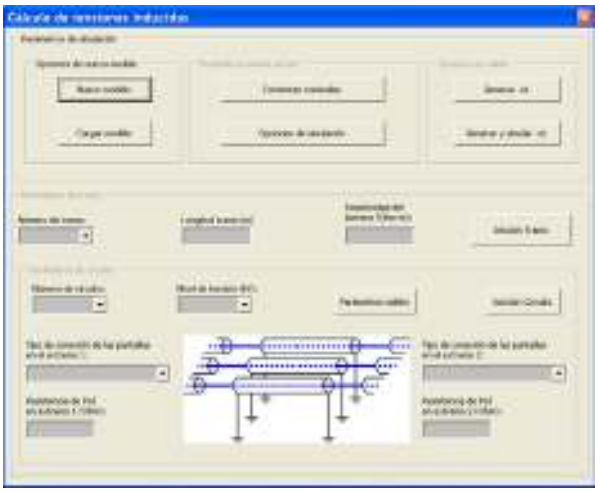


Figura 7: Pantalla de inicio al ejecutar el programa.

También pueden introducirse el número de cables de acompañamiento y el punto en que estos se transponen en tanto por uno (por defecto se trasponen en 0,5; es decir, a la mitad del tramo).

En el botón "Sección tramo" se seleccionan las coordenadas de referencia de los diferentes circuitos dentro de un mismo tramo.

En el botón de "Parámetros de simulación" se puede seleccionar el tipo de simulación (funcionamiento normal, cortocircuito trifásico o monofásico en sus cuatro diferentes modalidades). También puede seleccionarse el circuito que sufre el cortocircuito, la corriente de cortocircuito y los factores de reducción de falta lejana y de efecto sifón.

Por último, aunque resulta obvio, en el botón de "Corrientes nominales" se seleccionarán las intensidades de fase de cada circuito.

Cabe destacar que el documento que se genera en cada simulación presenta un gran número de comentarios que ayudan a seguirlo con facilidad y, si se da el caso, a realizar las modificaciones que uno estime oportunas con rapidez.

Es sencillo visualizar la tensión o corriente en cualquier punto del circuito. Así pues, se puede ver fácilmente la tensión en bornes de un descargador o la corriente que atraviesa una pantalla, o bien la componente de tensión en la pantalla 2 del circuito 1 debido a la corriente que circula por la pantalla 3 del circuito 2, por ejemplo.

7 ANÁLISIS Y DISCUSIÓN DE RESULTADOS

A continuación (Figura 8) se analiza el resultado de simular las tensiones locales inducidas en las pantallas de un simple circuito a 220 kV que consta de un tramo de 500 m con conexión SP con cable de acompañamiento traspuesto a la mitad del recorrido y en funcionamiento normal.

Vemos como dos pantallas tienen la misma tensión inducida en magnitud (aproximadamente) y están desfasadas 120º (líneas azul y magenta). Estas pantallas corresponden a las fases que se sitúan abajo en la disposición al tresbolillo. El cable de acompañamiento está muy cerca de ellas sólo durante la mitad del recorrido.

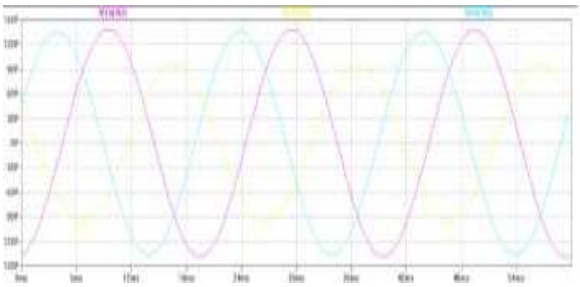


Figura 8: Simulación de tensiones inducidas en dos pantallas

Por otra parte, la tensión inducida en la fase central del tresbolillo (línea amarilla) es menor en magnitud. Esto es debido a que esta fase tiene cerca el cable de acompañamiento durante todo el recorrido, lo que provoca que sea ésta la fase que induce más tensión en el cable de acompañamiento y, por tanto, que la corriente por el cable de acompañamiento esté en oposición de fase respecto a la corriente por el conductor. Así, la componente de la tensión inducida en las pantallas debido a la corriente por el cable de acompañamiento es tal que provoca que la magnitud total de la tensión inducida en esta pantalla sea menor, y en las otras dos pantallas sea mayor de lo que sería si no existiera tal corriente.

Con el fin de validar el correcto funcionamiento del simulador y del modelado subyacente se han realizado una serie de simulaciones sencillas con resultados conocidos de antemano. A modo de ejemplo se comentan los resultados del modelado y simulación de un tramo de simple circuito con conexión de pantallas tipo *solid bonding*. El test se realiza para una distancia de línea de 1 km, una distribución al tresbolillo con distancia de 250 mm entre fases y diferentes combinaciones de valores de resistencia de puesta tierra. Los resultados se ofrecen en V/kA de defecto.

Se han comparado los resultados teóricos y los resultados ofrecidos por el software desarrollado para el circuito anterior en el caso de cortocircuito monofásico (en sus cuatro tipos diferentes).

El error promedio obtenido es del 0,2 % (diferencia entre valor teórico y valor obtenido de la simulación).

Es decir, es muy pequeño para todas las combinaciones de resistencias y en todos los tipos de cortocircuito, comprobándose así la fiabilidad del programa y del método de modelado.

Cabe destacar, además, que el cálculo teórico no tiene en cuenta la existencia de condensadores como sí ocurre en el modelo. Su influencia en el resultado es baja, ya que sólo implica una corriente de pequeño valor; pero es más notable cuanto menores son los resultados de tensión inducida.

8 CONCLUSIONES Y TRABAJOS FUTUROS

El modelado que se realiza en este trabajo es bastante fiel a la realidad, en cuanto a que modela la línea subterránea con todos aquellos parámetros que desempeñan un papel importante en el cálculo de las tensiones inducidas.

A diferencia de otras propuestas [12], esta manera de modelar permite una mayor velocidad de desarrollo porque es dinámica y el modelo en sí no aparece plasmado en lenguaje SPICE hasta el final. Si se realiza un modelado elemento a elemento mediante un entorno gráfico (como en Simulink o en los entornos gráficos basados en SPICE tipo Schematics) no resulta posible vincular una línea con otra de forma eficiente si no se cuenta con un módulo para doble circuito. Evidentemente esto es aplicable a un triple circuito, cuádruple circuito, etc.

Aunque alguna versiones más actuales de algunos simuladores permiten esta funcionalidad de acoplar múltiples circuitos, es a costa de la laboriosa tarea de definir cada acoplamiento manualmente, que es lo que hace nuestro programa de forma automática.

Por este motivo es habitual ver en entornos profesionales cómo se modela una línea de simple circuito en lugar de dos (cuando es el caso) asumiendo que el hecho de que sea un doble circuito no variará mucho las tensiones inducidas en el primero en caso de cortocircuito. Sin duda es así; pero quizás pueda ser de interés el saber qué ocurre exactamente, o bien saber la tensión inducida en las pantallas del circuito que no sufre la falta, lo que sí es posible con el software desarrollado para este trabajo.

Como trabajo futuro se plantea el obtener un modelo para analizar los transitorios.

REFERENCIAS

[1] Ben Amar T.A.F., Abdallah H.H. (2013) Fault prelocalization of underground single-phase cables: Modeling and simulation, Electrical Power and Energy Systems 44, 514–519.

[2] Brookes A.S., (1973) The design of specially bonded cable systems, Electra 28.

[3] Grainger J.J., Stevenson Jr. W.D., (1996) Análisis de Sistemas de Potencia, McGraw-Hill, México.

[4] ICAI (2010) “Máster en Proyectos, Construcción y Mantenimiento de Infraestructuras Eléctricas de Alta Tensió”, Universidad Pontificia de Comillas”.

[5] IEEE (1998) Guide for the application sheath-bonding methods for single conductor cables and calculation of induced voltages and currents in cable sheaths. IEEE Std. 575-1988.

[6] Linear Technology (2011) LT Spice IV User’s Guide.

[7] REE (2010) “Puesta a tierra de pantallas”. Curso de Formación a personal de Red Eléctrica de España S.A.U., UF522A.

[8] Simón Comín P., Garnacho Vecino F., Moreno Mohíno J., González Sanz A. (2011) Cálculo y diseño de líneas eléctricas de alta tensión, Editorial Garceta.

[9] Skipper D.J., (1976) The design of specially bonded cable systems II, Electra 47.

[10] Valle, Y., Hampton, N., Perkel, J., Riley, C. (2013) Underground Cable Systems, in Electrical Transmission Systems and Smart Grids, Springer, 195-237

[11] WG 21.07 (Cigré) (1990) Guide to the protection of specially bonded cable systems against sheath overvoltages, Electra 128.

[12] WG B1.26 (Cigré) (2008) Earth potential rises in specially bonded screen systems. Brochure 347.

SIMULACIÓN Y CONTROL DE COMPRESORES CENTRÍFUGOS USANDO ECOSIMPRO

Tania Rodríguez Blanco *, Daniel Sarabia Ortiz **, José Luis Morales ***, César de Prada Moraga *

*{tania.rodriguez, prada}@autom.uva.es, **dsarabia@ubu.es, ***jlmorales@intergeotecnologia.com

* Dpto. Ingeniería de Sistemas y Automática, Escuela Ingenierías Industriales, C/ Real de Burgos S/N, 47011, Valladolid

** Dpto. Ingeniería Electromecánica (Área de Ingeniería de Sistemas y Automática), Escuela Politécnica Superior, Campus del Rio Vena. C/ Francisco de Vitoria s/n, 09006, Burgos

*** InterGeo Tecnología - c\ Juliana nº 6. Urbanización Parque Real, Bloque 1. 28280, El Escorial - Madrid

Resumen

En este artículo se describe el modelado y simulación de compresores centrífugos accionados por turbinas de gas así como su sistema de control consistente en el control de la presión de impulsión y el control anti-bombeo (anti-surge).
Además, se describe y se simula la estructura de control avanzado necesaria para el control de dos compresores conectados en paralelo.

Palabras Clave: Simulación, control, compresor centrífugo, control anti-bombeo

1 INTRODUCCIÓN

El presente artículo forma parte de un proyecto resultado de la colaboración entre la empresa Intergeo Tecnología S.L y el departamento de Ingeniería de Sistemas y Automática de la Universidad de Valladolid.

Dicho proyecto está centrado en el desarrollo de diversas funcionalidades para incorporar en una herramienta de supervisión y simulación de redes de gaseoductos de gas natural. Estas funcionalidades están orientadas a incrementar la calidad de la gestión y operación de la red de gas a través de la mejora de la información sobre el proceso que puede obtenerse mediante el uso de técnicas de simulación y de algoritmos de supervisión, estimación y optimización.

Para ello se ha desarrollado una librería dinámica de componentes basados en modelos rigurosos y detallados y formada por tuberías de gas, estaciones de medida, estaciones de regulación y medida, estaciones de compresión, válvulas, compresores, turbinas, etc. para la simulación de redes de gas.

Los compresores son ampliamente usados en la industria para aumentar la presión del fluido que bombean. En general son equipos fundamentales cuya operación es compleja y que en el caso particular de las redes de distribución de gas natural parte del gas transportado es quemado para aumentar la presión y velocidad del gas que bombean.

Las estaciones de compresión se utilizan para recuperar la pérdida de presión (debida a la distancia, fricciones, etc.) que se va produciendo en las redes de distribución de gas natural. Normalmente el proceso de compresión se realiza con compresores centrífugos transfiriéndoles la energía mecánica que produce una turbina de gas o motor alternativo.

Los compresores centrífugos pueden ser conectados a través de distintas configuraciones: en serie, cuando se pretende conseguir una alta relación de compresión entre la aspiración y la impulsión, en paralelo cuando lo que se desea es trabajar con una mayor capacidad o como una combinación de ambas.

Es habitual que en las estaciones de compresión de las redes de distribución de gas natural haya instalado más de un compresor en paralelo para aumentar la capacidad de flujo bombeado, lo que implica tener un sistema de control de la presión de descarga más complejo para poder tener en cuenta cambios de carga diferentes en cada compresor.

Por ello en este artículo se detalla el modelo de compresores centrífugos accionados por turbinas de gas y se realiza la simulación de una red de distribución de gas natural a alta presión en la que se instala una estación de compresión. Además se presenta otro ejemplo para comprobar el funcionamiento del sistema de control de dos compresores conectados en paralelo. Todos los modelos han sido desarrollados en la herramienta de modelado y simulación EcosimPro [3]

2 MODELO DINÁMICO COMPRESOR CENTRÍFUGO

Un compresor centrífugo es una turbomáquina que consiste en un rotor (impeller) que gira dentro de una carcasa (spiral casing) provista de aberturas para la circulación del fluido como se observa en la Figura 1.

El rotor (impeller) es el elemento que convierte la energía mecánica del eje en cantidad de movimiento y por tanto en energía cinética del fluido, mientras que en la carcasa se encuentra incorporado el difusor (diffuser), que es el elemento que convierte la energía cinética en energía potencial de presión [4].

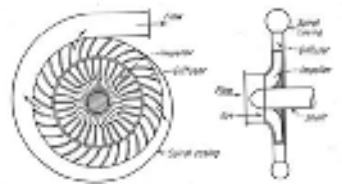


Figura 1: Esquema de las partes de un compresor centrífugo

Se ha implementado un modelo dinámico del compresor basado en ecuaciones diferenciales para describir los balances de materia y de energía, perfil de presiones y variación de la velocidad de giro del compresor.

2.1 CURVA CARACTERÍSTICA

Para cada compresor, el fabricante entrega una curva de operación donde están representadas las líneas de isovelocidad y de isoeficiencia, representando en el eje de ordenadas la diferencia de presión (denominada altura si se expresa en metros de columna de fluido) y el flujo en abcisas. El punto de operación del compresor vendrá dado por la intersección en la curva característica a una determinada velocidad de giro, del incremento de presión en el compresor y el caudal que circula por él. Sin embargo, desde un punto de vista práctico, no todos los puntos de la curva característica pueden ser alcanzados, la Figura 2 muestra los márgenes operativos que deben ser respetados para el correcto funcionamiento del compresor [6] y que son:

1- Línea de bombeo.
2- Línea de máxima velocidad.
3- Línea de mínima velocidad.
4- Línea de caudal máximo.
5- Línea de máxima potencia

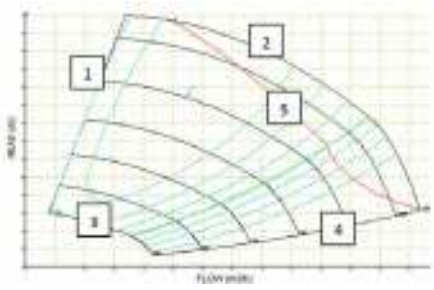


Figura 2: Curva característica típica de un compresor centrífugo

2.2 BALANCE DE MATERIA

El balance de materia en el compresor viene dado por la ecuación (1):

$$\frac{dm}{dt} = F_{in} - F_{out} \qquad (1)$$

y suponiendo que no existe acumulación de materia, la ecuación (1) queda:

$$F_{in} = F_{out} = F \qquad (2)$$

Donde F_{in} y F_{out} (Kg/s) representan los flujos másicos de gas en la aspiración e impulsión del compresor.

2.3 BALANCE DE ENERGÍA

El balance de energía es tal que los cambios de energía cinética y potencial son despreciables, así como las pérdidas de carga.

$$\frac{dE}{dt} = P + F(h_{in} - h_{out})$$

Además se debe tener en cuenta que la energía del compresor E (KJ) es una combinación de energía mecánica y térmica:

$$E = \frac{1}{2}Jw^2 + mU = 2\pi^2 JN^2 + mU$$

Por lo que la ecuación del balance de energía queda definida por la siguiente expresión:

$$4\pi^2 J\frac{dN}{dt} + \frac{d(mU)}{dt} = P + F(h_{in} - h_{out}) \qquad (3)$$

Donde J (Kg m2) es el momento de inercia, U es la energía interna (KJ/Kg), P (KW) es la potencia del compresor y h_{in} y h_{out} (KJ/Kg), las entalpías de entrada y salida.

Finalmente, la velocidad del compresor *N* (rpm) varía en función de la diferencia entre el par motor de la turbina τ_t (N m) y del compresor τ_c (N m):

$$J\frac{dN}{dt} = \tau_t - \tau_c = \frac{P_t}{w} - \frac{P}{N} \quad (4)$$

Donde P_t (KW) es la potencia de la turbina y w (rpm) su velocidad de giro.

2.4 CONCEPTOS TERMODINÁMICOS. MODELO POLITRÓPICO

El trabajo realizado por el compresor W_a (KJ/Kg) y por lo tanto su potencia se calculan a partir del modelo politrópico que se rige por la siguiente expresión:

$$P \cdot V^n = P_{in} \cdot V_{in}^n = cte \quad (5)$$

El modelo politrópico se utiliza en compresores centrífugos ya que todos los procesos de compresión se realizan entre el isoentrópico (no existe intercambio de calor con el exterior) y el isotérmico (hay intercambio de calor con el exterior) [2].

El trabajo politrópico *Wp* (KJ/Kg) absorbido por el compresor se calcula a partir de la ecuación 6:

$$-W_P = Z_m \cdot \frac{R \cdot n \cdot T_{in}}{M \cdot (n-1)} \cdot \left[\left(\frac{P_c}{P_{in}} \right)^{\frac{n-1}{n}} - 1 \right] \quad (6)$$

Donde P_{in}y P_c (Pa) son las presiones de aspiración e impulsión respectivamente.

El factor de compresibilidad Z_m es calculado de forma explícita utilizando el método de Sarem [5].

El trabajo real W_a es función de la eficiencia politrópica η_P del compresor centrífugo:

$$\eta_P = \frac{n \cdot (k-1)}{k \cdot (n-1)} \quad (7)$$

$$-W_a = \frac{-W_P}{\eta_P} \quad (8)$$

Siendo k el coeficiente isoentrópico y n el coeficiente politrópico. Otro parámetro importante para la caracterización del compresor centrífugo es la altura politrópica H_{pol} (J/mol) que se define como la energía acumulada en el fluido como consecuencia del aumento de energía termodinámica y se calcula a partir de la siguiente expresión [7]:

$$H_{pol} = \frac{n}{n-1} \cdot Z_{in} \cdot R \cdot T_{in} \cdot \left[\left(\frac{P_c}{P_{in}} \right)^{\frac{n-1}{n}} - 1 \right] \quad (9)$$

2.5 VARIACIÓN DE PRESIÓN EN EL COMPRESOR

El incremento de presión $incP$ (Pa) producido en el compresor es función de la velocidad del compresor y el caudal Q (m^3 /s) que circula por el compresor y se obtiene interpolando dichos valores en la curva característica del compresor.

$$incP = f(Q, N) \quad (10)$$

Finalmente, la presión de impulsión del compresor se define como:

$$P_c = P_{in} + incP \quad (11)$$

2.6 VALIDACIÓN DEL MODELO

El modelo será validado a partir de datos experimentales como pueden ser la temperatura de descarga del compresor, el incremento de presión y el caudal de aspiración. Se realizará el ajuste de ciertos parámetros, como por ejemplo, la eficiencia politrópica o el momento de inercia del compresor de modo que el modelo represente correctamente el funcionamiento real de un compresor centrífugo.

3 CONTROL COMPRESOR CENTRÍFUGO

3.1 CONTROL ANTIBOMBEO

3.1.1 Bombeo límite

La operación de un compresor centrífugo puede llegar a ser inestable debido a cambios en alguna de las condiciones del proceso (presión, caudal, temperatura) llegando hasta la situación de surge o bombeo límite [1].

Si a cualquier velocidad dada, disminuye el caudal de aspiración, la presión desarrollada por el compresor tiende también a disminuir, pudiendo llegar a un punto en el cual esta presión sea inferior a la existente en la línea de impulsión, resultando como consecuencia una inversión momentánea del sentido de flujo. La inversión del sentido de flujo tiende a bajar la presión en la línea de impulsión, recuperándose la compresión normal, volviéndose a repetir el ciclo continuamente. Esta acción cíclica denominada surge o bombeo límite de la máquina es una condición inestable y puede causar daños importantes en el compresor.

Es por esto que todos los compresores centrífugos incorporan un control antibombeo y por tanto también se implementará en el modelo del compresor en EcosimPro.

3.1.2 Obtención de la línea de bombeo

Existe una capacidad mínima para cada velocidad por debajo de la cual la operación es inestable y se produce el fenómeno de bombeo límite. Estos valores de caudal mínimo se representan en la línea de bombeo.

El antibombeo es un control inferencial, por tanto, el primer paso para diseñar el sistema de control es obtener la ecuación de la línea de bombeo.

La línea de bombeo (12) de pendiente K y término independiente $B0$ se obtiene a partir de calcular la correspondiente recta que pasa por dos puntos de la curva característica próximos a la velocidad de trabajo, por ejemplo 80 y 100 % de velocidad nominal del compresor como se puede observar en la Figura 3.

$$Q = K \cdot incP + B0 \qquad (12)$$

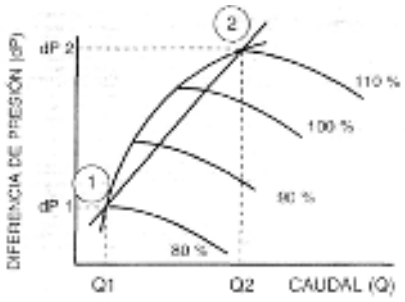


Figura 3: Línea de bombeo

3.1.3 Método de pendiente paralela de la línea de control.

En este método se emplea la ecuación de la recta de bombeo como línea de control. A continuación se suma un margen de seguridad a la línea de bombeo B1 para obtener diferentes rectas como muestra la Figura 4.

$$Q = K \cdot incP + B0 + B1 \qquad (13)$$

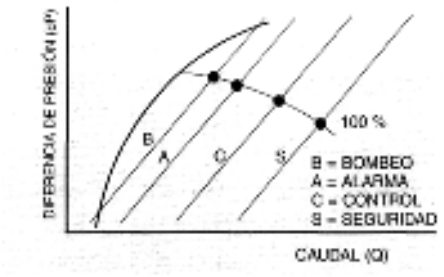


Figura 4: Desplazamiento paralelo de la línea de control

3.1.4 Esquema del sistema de control.

Como puede verse en la figura 5 el sistema de control se comporta como un lazo de control de caudal cuyo punto de consigna está fijado en función del cálculo efectuado para obtener la línea de control.

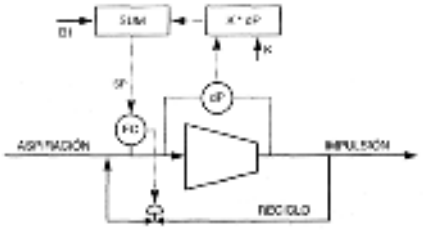


Figura 5: Esquema de control

Para realizar dicho control se utiliza un controlador PID que recibe los valores de la variable a controlar (caudal de entrada al compresor) y del Set Point (caudal mínimo de entrada al compresor) y envía una señal a la válvula de reciclo para variar su apertura y recircular el caudal necesario para garantizar la seguridad del compresor.

3.2 CONTROL PRESIÓN DE IMPULSIÓN

El otro control presente en todo compresor centrífugo es el necesario para controlar la presión de descarga del mismo, ver Figura 6. Lo habitual es regular la presión del compresor a partir de la velocidad de giro de la turbina de gas.

Por otro lado, la velocidad se ajusta modificando la apertura de la válvula que permite la entrada de gases de combustión a la turbina, vapores de alta presión (VAP) en la Figura 6.

Cuando la presión de descarga del compresor disminuya, el sistema de control hará que pase un flujo mayor de gases a la turbina, aumentando la velocidad de ésta y por tanto la del compresor.

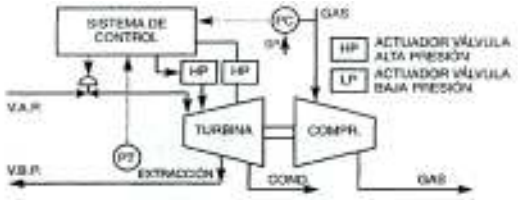


Figura 6: Sistema de control de la presión de impulsión del compresor

4 ESQUEMÁTICO DE UN COMPRESOR CENTRÍFUGO EN ECOSIMPRO

En la Figura 7 se muestra el esquemático de un compresor centrífugo con su estructura de control, implementado en EcosimPro.

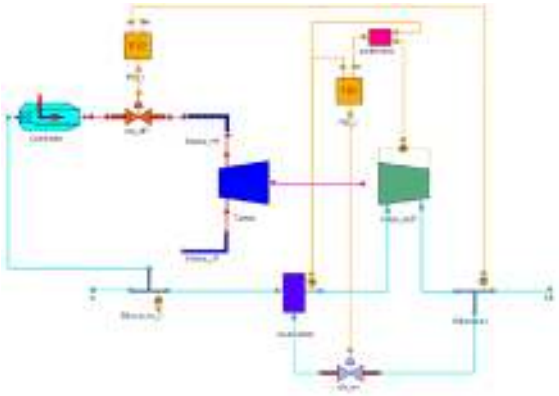


Figura 7: Esquemático de un compresor centrífugo accionado por una turbina de gas

El modelo completo de un compresor centrífugo estará formado por tanto por los siguientes componentes:

- *Compresor centrífugo*
- *Turbina de gas:* Al igual que para los compresores, los fabricantes de turbinas también deben aportar la curva característica en la que se muestre la relación entre el flujo de gas que circula a través de la turbina, su velocidad y la diferencia de presión entre la entrada y la salida. Esta curva permite variar la velocidad de la turbina, y a su vez, la del compresor, regulando el flujo de gas procedente del combustor que se expande en la turbina.
- *Válvulas:* existen dos válvulas, una de reciclo para el control antibombeo y otra que permite cambiar la velocidad de giro de la turbina a partir de la variación del flujo de gases de combustión. El cálculo del flujo másico de gas F_v (Kg /s)que atraviesa cada válvula se realiza a partir de la siguiente expresión:

$$F_v = K_v \cdot u_v \cdot \sqrt{P_{in} \cdot rho_{in} \cdot (1 - P_{in}/P_{out})}$$

  Donde K_v (m^3 s) es el coeficiente de flujo de la válvula, u_v es la apertura y rho_{in}(Kg/m^3) la densidad del gas de entrada a la válvula.
- *Combustor:* es el elemento de la turbina de gas en el cual se realiza la combustión entre gas natural y aire comprimido. La relación entre el flujo de gas natural consumido y el flujo de gases de combustión producidos es 1:17.
- *Acumulador:* depósito que se utiliza para amortiguar los cambios de presión que se producen en la aspiración del compresor.

  El acumulador tiene dos entradas (flujo de gas natural de la red y flujo recirculado) y una salida (flujo de aspiración del compresor). Las ecuaciones 14 y 15 muestran el balance de materia y energía en el acumulador:

$$\frac{dm}{dt} = F_{in1} + F_{in2} - F_{out} \quad (14)$$

$$m \cdot C_{Pout} \cdot \frac{dT_{out}}{dt} = F_{in1}h_{in1} + F_{in2}h_{in2} - F_{out}h_{out} \quad (15)$$

- *Bifurcaciones*
- *Controladores PID*

5 SIMULACIÓN COMPRESOR CENTRÍFUGO CONECTADO A UNA RED DE DISTRIBUCIÓN DE GAS NATURAL

En la figura 8 se muestra una red sencilla de distribución de gas natural formada por dos entradas y una salida en la cual es necesaria una estación de compresión (formada normalmente por compresores centrífugos) para recuperar la pérdida de presión que se va produciendo en las redes de distribución de gas natural.

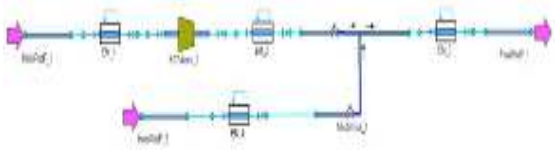


Figura 8: Compresor centrífugo conectado a una red de distribución de gas natural

Los tramos de tubería presentan una longitud de 10 Km y un diámetro de 1.22 m.

Para la simulación de este ejemplo se ha empleado una librería dinámica de componentes también desarrollada en Ecosimpro, en la que se han modelado, además del compresor centrífugo, las tuberías de gas y las estaciones de medida que se utilizan para la conexión entre los distintos elementos. La librería dinámica considera el gas natural formado por 12 componentes.

El modelo presenta un total de 8274 variables, 30 de ellas condiciones de contorno, 51 lazos algebraicos y 940 ecuaciones (diferenciales+algebraicas).

Las condiciones de contorno del experimento son:

- Composición del gas de entrada a la red $y\,\langle CO_2, H_2S, N_2, C1, C2, otros\rangle = \{0.033, 0.0558, 0.0025, 0.82, 0.05, 0.038\}$

- Set Point de presión de descarga del compresor

$$SP(Pa) = 71 \cdot 10^5 + 1 \cdot 10^5 step(TIME, 10) + 2 \cdot 10^5 step(TIME, 1000) + 1 \cdot 10^5 step(TIME, 10000)$$

- Presión del gas en la entrada de la red

$$P_{in}(Pa) = 68 \cdot 10^5$$

- Presión del gas a la salida de la red

$$P_c(Pa) = 68 \cdot 10^5$$

Se realizan saltos en el valor de set point de presión de impulsión para observar como el sistema de control es capaz de seguir la referencia dada.

En la figura 9 se muestra la presión de impulsión y el set point deseado:

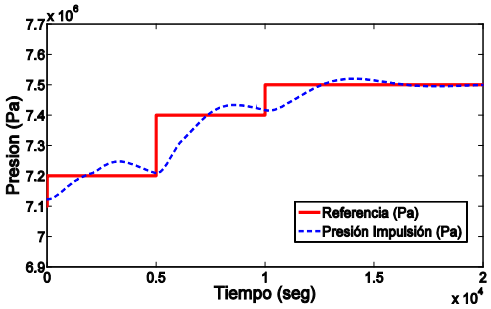


Figura 9: Seguimiento del valor de Set Point de presión de impulsión del compresor

Para conseguir alcanzar el set point la velocidad de giro del compresor varía como se muestra en la figura 10:

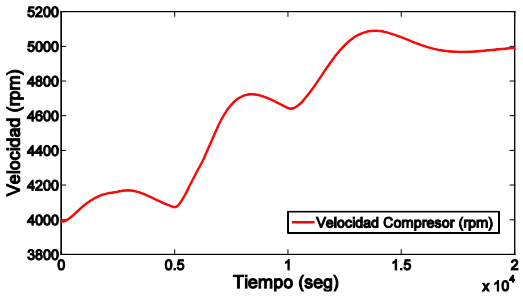


Figura 10: Variación de la velocidad de giro del compresor

En la figura 11 se muestra el perfil de presiones de las tuberías de aspiración y de impulsión del compresor:

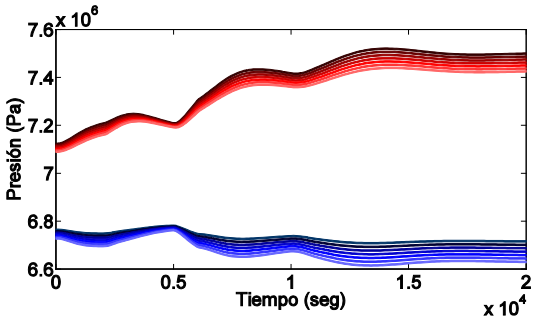


Figura 11: Perfil de presiones en las tuberías de entrada y salida del compresor

6 CONTROL DE COMPRESORES EN PARALELO

La operación de compresores en paralelo permite aumentar la capacidad de flujo de gas a comprimir.
El objetivo principal del control de compresores en paralelo es mantener constante la presión de descarga de los compresores.

Un cambio en la presión de descarga implica una diferencia entre el flujo demandado y el flujo aportado, por lo que la capacidad de los compresores debe ser manipulada para mantener la presión en el valor deseado.

En el caso de los compresores centrífugos la capacidad de los compresores se modificará manipulando la válvula de alimentación de gas a la turbina que implica un cambio en la velocidad de giro del compresor.

El segundo objetivo en el control de compresores en paralelo es proveer una adecuada distribución de la carga entre los compresores que están operando. Esto es necesario aunque los compresores sean idénticos ya que siempre existen diferencias entre el rendimiento de una máquina y otra. El sistema de control descrito en este apartado ha sido obtenido de "Application data 353-123" de SIEMENS [8].

La estrategia de control de la presión de descarga incluye un controlador de posición (ZC) para compensar los cambios producidos en el arranque y parada de compresores y un bloque para ajustar la distribución de la carga gracias a un coeficiente R cuyo valor oscila entre 0 y 2. La suma del coeficiente R de los dos compresores deberá ser 2.

Cuando se trabaja con compresores centrífugos o axiales, es necesario que cada compresor tenga un sistema de protección antisurge propio, como se ha explicado en los apartados anteriores . Esto es necesario para desacoplar el control de la capacidad y el antisurge y así minimizar la interacción entre estas variables.

6.1 DESCRIPCIÓN DE LA ESTRATEGIA DE CONTROL

En esta sección se describe la estrategia de control utilizada para compresores conectados en paralelo. La Figura 12 muestra dicha estructura:

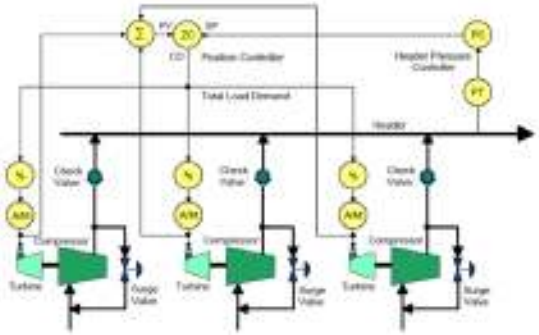


Figura 12: Estrategia de control de compresores en paralelo

El controlador de presión PID manipula el set point del controlador de posición ZC.
El controlador de posición manipula la señal de demanda total que es enviada a los compresores.

El bloque sumatorio calcula la suma de las señales de carga a cada compresor. Estas señales representan la apertura de las válvula de gas que alimentan a las turbinas.
Para el caso de dos compresores el rango de las señales de entrada será entre 0-50%.
Cualquier acción del operador que altere las señales de carga para uno de los compresores (cambio en el coeficiente de distribución de carga R) tendrá un impacto inmediato en la salida calculada por el sumatorio. Si, por ejemplo, el operador disminuye la carga en el compresor 1, el controlador de posición aumentará la señal al otro compresor para incrementar la capacidad.

6.2 SIMULACIÓN DE DOS COMPRESORES CONECTADOS EN PARALELO

En este ejemplo mostrado en la Figura 13, se ha añadido un depósito a la salida de los compresores que determina la cantidad de flujo demandado y cuya presión será la variable controlada por el sistema de control.
Se ha realizado un experimento en el cual se varía el coeficiente de distribución de carga R_1 y R_2 de cada compresor a partir de la siguiente expresión.

$$\zeta \cdot \frac{dR_1}{dt} = R_f - R_1 \tag{16}$$

$$\zeta \cdot \frac{dR_2}{dt} = R_f - R_2 \tag{17}$$

Inicialmente la carga se distribuye uniformemente entre los dos compresores por lo que los coeficientes R_1 y R_2 son igual a 1. Una vez alcanzado el estado estacionario se cambia el valor de Rf de manera que el coeficiente de distribución va variando hasta que R_1 tome el valor de 1.3 y R_2 tome el valor 0.7 (ver Figura 14).. De esta manera la velocidad de la turbina del compresor 1 aumenta y por lo tanto la del propio compresor. El incremento de presión del compresor 1 será mayor que el incremento en el compresor 2 (ver Figura 15), es decir, el compresor con mayor carga aporta un mayor incremento de presión, pero ambos incrementos se compensan de modo que en el depósito de salida la presión se mantenga en el valor de set point deseado.

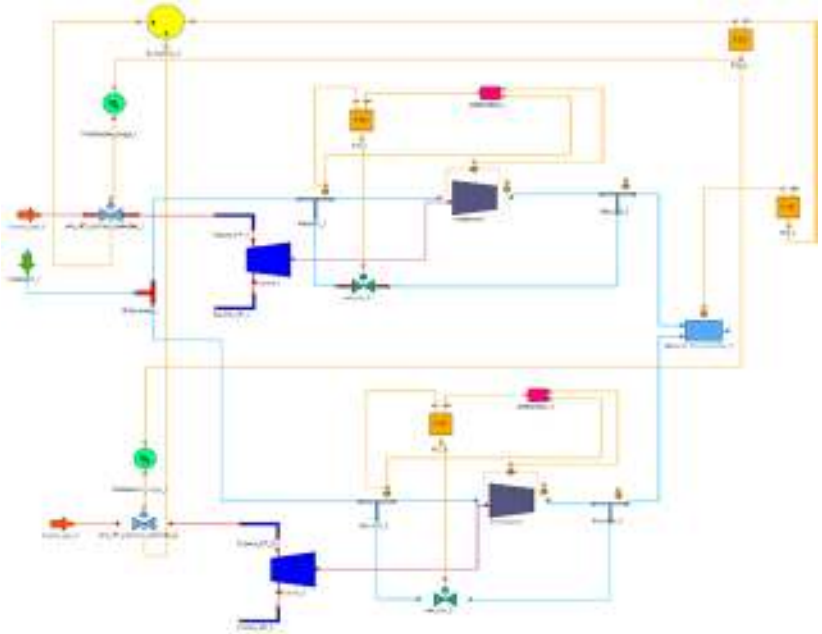


Figura 13: Esquemático en EcosimPro del modelo de compresores en paralelo.

Finalmente, en tiempo igual a 100 s se ha realizado de nuevo un cambio en los coeficientes de distribución de modo que toda la carga la realice el compresor 1, es decir, su coeficiente tome el valor 2, siendo 0 la carga del compresor 2. Aunque no se haya asignado ninguna carga al compresor 2, éste, por el hecho de estar conectado, girará a la mínima velocidad de giro y dará el incremento de presión mínimo determinado por su curva característica.

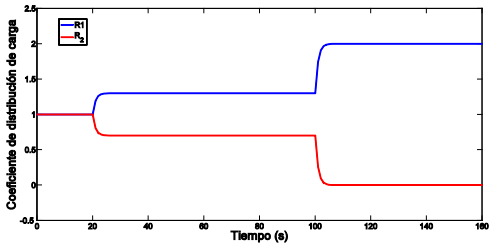


Figura 14: Variación del coeficiente de distribución de carga de los compresores. La suma debe ser siempre 2.

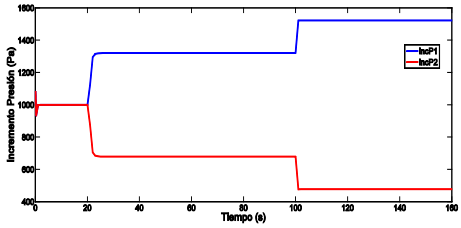


Figura 15: Incremento de presión en cada compresor. La suma debe ser el valor de la presión de descarga en el depósito.

Para un caudal de gas determinado, un aumento del incremento de presión de cada compresor implica un aumento de la velocidad de giro (y viceversa) como se muestra en la Figura 16.

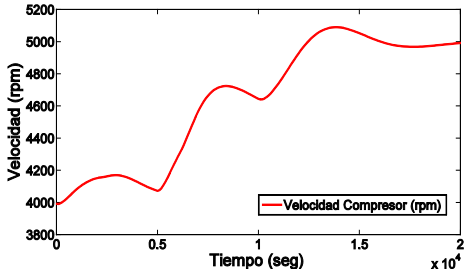


Figura 16: Velocidad de giro de cada compresor

7 CONCLUSIONES

Los compresores son un elemento fundamental en cualquier proceso industrial, en particular en las redes de distribución de gas natural, en donde se utilizan para recuperar la presión del gas que va perdiendo a lo largo de la distancia.

En este artículo se ha descrito el modelo dinámico y detallado de un compresor centrífugo y todos los elementos y componentes auxiliares, turbina de gas, válvulas, etc. Así como el sistema de control de la presión de descarga y del sistema de control anti-bombeo necesarios para poder operar el compresor de manera segura. También se ha descrito el sistema de control para regular la presión de descarga cuando han sido instalados más de un compresor en paralelo permitiendo asignar una carga diferente a cada compresor.

Además se han mostrado dos ejemplos de uso de los compresores centrífugos: a) simulación de una red sencilla de distribución de gas natural que incorpora un compresor y b) simulación de dos compresores en paralelo para gestionar diferentes cargas a cada uno.

Todos los modelos y las simulaciones forman parte de una librería de modelos más compleja para modelar redes de gas natural y han sido desarrollados en la herramienta de simulación y modelado EcosimPro.

Agradecimientos

Este trabajo ha sido desarrollado en el marco del proyecto "Modelado y supervisión de redes de distribución de gas natural" financiado por la empresa Intergeo Tecnología S.L.

Referencias

[1] Acedo Sánchez, J., (2003) "Instrumentación y control avanzado de procesos (Teoría y Práctica)".
[2] Bathod P. & all. "Process component design". Capítulo 7: pumps and compressors. Universidad de Tulsa. Oklahoma
[3] EcosimPro. (2013). Empresarios Agrupados. Versión 5.0.6, http://www.ecosimpro.com/
[4] http://materias.fi.uba.ar/6720/unidad3.PDF
[5] Pitzer, K.S. y otros: The Volumetric and Thermodynamic Properties of Fluids, Journal of American Chemical Society, Vol 77, No. 13, July 1955, p. 3427-3440
[6] Protocolo de detalle PD-10: "Cálculo de capacidades de las instalaciones". Ministerio de industria, turismo y comercio. Dirección general de política energética y minas.
[7] Procedimiento para el Cálculo de la Capacidad de la Estación de Compresión de Córdoba. PF-105-DGOS 07/2006. Enagás.
[8] SIEMENS. Application data 353-123: Procidia control solutions Parallel Compressor control

CONCURRENT AND DISTRIBUTED SYSTEMS ANALYSIS USING COLORED PETRI NETS

Jenaro Nosedal Sánchez
Autonomous University of Barcelona, Sabadell, Spain, jenaro.nosedal@uab.cat

Olatunde Baruwa
Autonomous University of Barcelona, Sabadell, Spain, olatunde.baruwa@uab.cat

Miquel Ángel Piera Eroles
Autonomous University of Barcelona, Sabadell, Spain, miquelangel.piera@uab.cat

Abstract

Distributed and concurrent systems are, by nature, complex and difficult to design and test. Colored Petri Nets (CPNs) have been used extensively for the performance evaluation of systems since the model contains all the events and their interactions. The graphical and mathematical meaning supports their wide application for modeling, simulating and analyzing the evolution of complex systems (building partially or fully state space graphs also known as coverability tree). In this paper a benchmark problem is modeled and analyzed using CPNs and in specific by coverability tree revision.

Key Words: Colored Petri Nets (CPNs), Reachability graphs, Simulation.

1 INTRODUCTION

A typical distributed or concurrent system consists of a number of independent but communicating processes. This means that the execution of such systems may proceed in many different ways as a result, distributed and concurrent systems are, by nature, complex and difficult to design and test.

This has motivated the development of methods which support computer-aided analysis, validation, and verification of the behavior of concurrent and distributed systems. State space methods are some of the most prominent approaches in this field. The basic idea underlying state spaces is (in its simplest form) to compute all reachable states and state changes of the system, and represent these as a directed graph. [2]

Petri Nets are an extension of state graphs which provide a compact and effective representation of concurrent systems [1] (with parallel and synchronized changes), besides mutual exclusions and conflicts.

The graphical meaning by directed bipartite schemas and the mathematical expression by state matrix or state equation, supports their wide application for modeling, simulating and analyzing the evolution of complex systems (building partially or fully state space graphs also known as coverability tree).

CPNs allow a higher level of modeling by using colors that allow the representation of entity attributes, both Petri nets (PNs) and CPNs have been used extensively for the performance evaluation of systems since the model contains all the events and their interactions. [7]

This paper presents the application of CPNs and in specific the usefulness of coverability tree for simulating and analyzing dynamic and concurrent systems. As a benchmark problem, is modeled and analyzed the Sudoku numerical puzzle, which has been widely revisited with different approaches like linear and constraint programming, genetic algorithms, simulated annealing, among orders [8].

The paper is organized has follows, in the next section description and properties of CPNs are presented and a more detailed description of the coverability tree is provided. The benchmark problem is presented in section 3 and some results from another approaches. In section 4, is presented our model in CPNs. Section 5 includes some experimental results obtained and the section 6 our main conclusions.

2 COLORED PETRI NETS (CPNs) AND REACHABILITY GRAPH

2.1 CPNs

The main characteristics of CPNs that offer a suitable formalism to describe discrete event–oriented simulation models are the following [7]:

a. All the events that could appear according to each particular system state can be easily determined reachability graph or coverability tree).

b. All the events that can set off the firing of a particular event can be detected visually.

A modeling methodology that could support both characteristics for any type of discrete event–oriented system is essential to tackle the performance improvement of complex systems, from the conceptual model that describes all the event relationships to the codification of a simulation model that could support the decision task of optimization routines at any moment of the evaluation process.

Other CPNs characteristics empower the use of this formalism to specify logistic or manufacturing systems:

c. They allow the specification of a system at different abstraction levels, according to the modeling objectives.
d. They allow the specification of a complex system by means of bottom-up techniques or more advanced software engineering techniques, such as an iterative and incremental development process instead of a waterfall cycle and the promotion of a component-based architecture.

They can be considered as a graphical modeling tool with a few syntactic rules.

The main CPN components that fulfill the modeling requirements are [6]:

•Places: These are very useful to specify both queues and logical conditions, represented by circles.
•Transitions: These represent the events of the system, depicted by rectangles.
•Input arc expressions and guards: These are used to indicate which type of tokens can be used to fire a transition.
•Output arc expressions: These are used to indicate the system state change that appears as a result of firing a transition.
•Color sets: Determine the types, operations and functions that can be used by the elements of the CPN model. Token colors can be seen as entity attributes of commercial simulation software packages.
•State vector: The smallest piece of information needed to predict the events that can appear. The state vector represents the number of tokens in each place, and the colors of each token.

The color sets will allow the modeler to specify the entity attributes. The output arc expressions make it possible to define which actions should be coded in the event routines associated with each event (transition). The input arc expressions, in turn, make it possible to see when and why an event can appear, and consequently introduce new pre-conditions (or removing them) in the model, or alternatively change some variable or attribute values in the event routines to disable active events.

From the Operational Research (OR) point of view, the CPN model provides the following mathematical structures:
• Variables: A variable can be identified for each color specified in every place node.
• Domains: The domains of the variables can be easily determined by enumerating all the tokens specified in the initial state.
• Constraints: Can be obtained straightforwardly from the arc and guard expressions. Arc expressions can contain constant values, color variables or mathematical expressions.

2.2 REACHABILITY GRAPH

The reachability graph or coverability tree of the untimed CPNs can be generated calculating all the states that can be reached from the initial one taking into account the restrictions imposed by the net structure (arc expressions, input arcs and guards) that is why it is also known as the reachability graph.[5]

Each node in the graph represents a feasible marking and each arc the transition or event which allows the system evolution from the initial state to a new (from a father node to a son node). In the first level is placed the initial marking and the tree grows horizontally, until any other states can be reached from the father node, when all the descendents are obtained this process continues until there is no more descendents nodes to obtain (the system can´t reach any different state and any transitions are enabled) which means that coverability tree has been obtained. [10]

From the Artificial Intelligence (AI) point of view, the coverability tree of a CPN model makes it possible to determine [6]:

•All the events that could appear according to a particular system state.

•All the events that can set off the firing of a particular event.

•All the system states (markings) that can be reached starting from a certain set of initial system operating conditions M0.

•The transition sequence to be fired to drive the system from a certain initial state to a desired end-state. [10]

3 BENCHMARK PROBLEM

Sudoku is a logical puzzle that has achieved international popularity. Given this, there have been a number of computer solvers developed for this puzzle. Various methods including genetic algorithms, simulated annealing, particle swarm optimization and harmony search have been evaluated for this purpose. [8]

Accordingly with [11] while many attempts have been made to solve the sudoku problem, most fall short as they do not solve the problem efficiently though some techniques, such as linear programming and constraint satisfaction, solve the problem extremely quickly and efficiently. Though the effort of solving sudoku puzzles with artificial intelligence techniques is not in vain as these investigations sometimes divulge helpful information with regards to solving other NP complete problems.

The Sudoku problem may be established under the next specifications [3]:

Definition 1 A Sudoku puzzle is represented by a 9x9 grid, which comprises nine 3x3 sub-grids (also called boxes). Some of the entries in the grid are filled with numbers from 1 to 9, whereas other entries are left blank.

Definition 2 A Sudoku puzzle is solved by assigning numbers from 1 to 9 to the blank entries such that every row, every column, and every 3x3 sub-grid contains each of the nine possible numbers.

Property 1 Sudoku puzzles that have only one solution.
Property 2 Sudoku puzzles that can be solved with only reasoning.

Having only one solution means that all the numbers to be assigned to the blank entries are necessary assignments.

The second property requires in addition that, at any stage in the course of solving the puzzle, there should always be at least one blank entry that can be assigned to merely by considering what is immediately implied by the set of non-blank entries. Hence, reasoning consists in using inference rules in such a way that all of the assignments are found.
The difficulty of a Sudoku puzzle can be measured by the positioning of the numbers on the grid and not the number of givens on the grid [4].

In Table 1 are presented time performance data reported in [11] for the search solution of easy Sudoku puzzles using genetic algorithm and Steepest Hill Ascent.

Table 1: Searching solution time performance.

Algorithm Approach	**Level**	**Clues**	**Avg Time to solve (s)**
Steepest Hill Ascent	Easy	31	1443.99656
Genetic Algotihms	Easy	30	11226.8

4 MODEL IN CPNs

The model presented in this section has been prototyped in CPNTools 3.4.0. CPNTools is a mathematical tool for editing, simulating, and analyzing Colored Petri nets. A fast simulator efficiently handles untimed and timed nets. Full and partial state spaces can be generated and analyzed, and a standard state space report containing information, such as boundedness properties and liveness properties. [9]

For a better understanding of the model, in Figure1 is presented the number of each cell within the grid, and the numbers in left indicate the row and top number indicate the column of each cell, besides is presented a sub grid grouping each set of 3x3 cells.

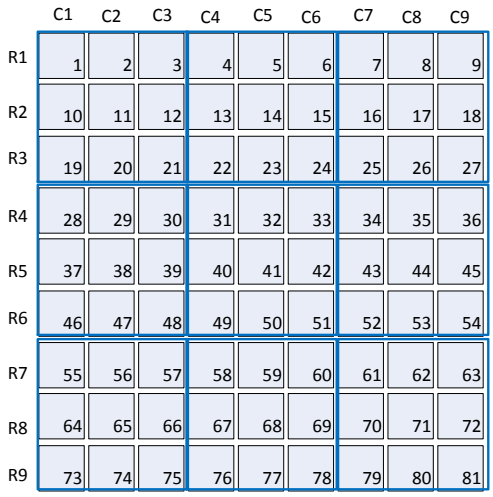


Figure 1: Sudoku general grid

Each empty cell represents an assignment event and thus is modeled by one transition. The model comprises 81 transitions, if a cell has been given as a clue (initial marking) guard expression disables the transition.

In Figure 2 is presented the transition 1 which corresponds to the assignment for cell number 1.

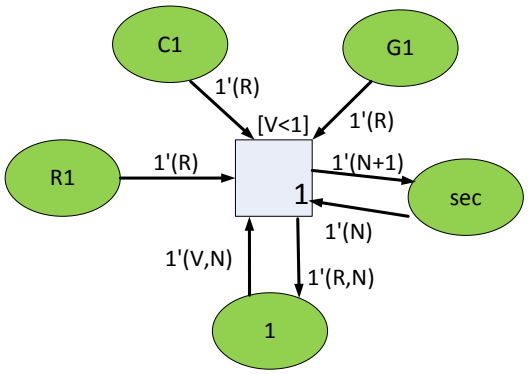


Figure 2: Transition number 1

In Figure 3 is presented the transition 81 for the assignment in cell 81.

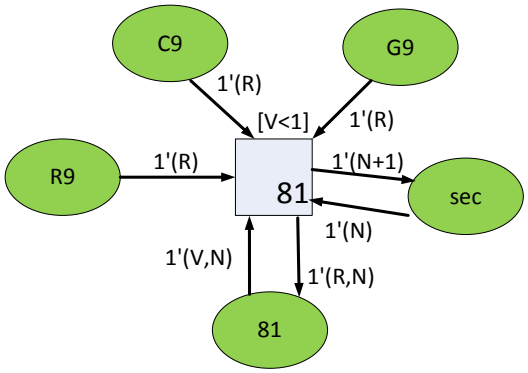


Figure 3: Transition number 81

The meaning of places and colors and transitions is presented next in Tables 2 and 3 and 4.

Table 2: Places.

Place	Color	Meaning
1 TO 81	1'(V,N)	Each cell in the grid for the Sudoku puzzle. If is a given clue initial marking is the given value otherwise the initial marking is cero "0".
R1 TO R9	1'(R)	Available number to assign in each row
C1 TO C9	1'(R)	Available numbers to assign in each column
G1 TO G9	1'(R)	Available numbers to assign in each group or block of 3x3 cells
sec	1'(N)	A process sequence number (avoids unnecessary combinatorial)

Table 3: Colors

Color	Definition	Meaning
R	Int 1…9	Available number to be assigned by row, column and block.
V	Int 0…9	Initial marking in the cell
N	Int 1..N	Turn for assignation

Table 4: Transitions.

Transition	Description
1 TO 81	Filled of the cell n (between 1 and 81 if is empty) with a feasible number (available in Cn&Rn&Gn)

5 EXPERIMENTAL RESULTS

The model was tested for a set of Sudoku puzzles with 31given clues and normal difficult level. The coverability trees were obtained using Radius (Radius is a prototype tools for exploration of reachability from CPNs, developed by UAB).

In figures 4 to 8 are presented the Sudoku puzzles A to E, the bold and underlined denote the given clues in each case and the rest the assignment solution obtained.

	C1	C2	C3	C4	C5	C6	C7	C8	C9
R1	7	9	2	6	1	5	3	**8**	**4**
R2	**5**	8	3	7	**4**	**2**	**6**	9	1
R3	1	6	**4**	3	9	8	5	**2**	7
R4	9	**4**	8	2	**6**	**3**	**7**	1	5
R5	2	7	5	4	8	**1**	9	6	**3**
R6	**6**	**3**	1	**9**	**5**	**7**	**2**	4	8
R7	8	**5**	7	1	2	**9**	4	3	**6**
R8	**3**	**2**	6	**8**	7	4	**1**	5	**9**
R9	4	1	**9**	**5**	3	6	**8**	7	2

Figure 4: Sudoku "A"

	C1	C2	C3	C4	C5	C6	C7	C8	C9
R1	3	6	**8**	**9**	5	4	**7**	**2**	1
R2	5	2	**9**	7	**1**	8	6	**4**	3
R3	7	**1**	**4**	3	6	**2**	**5**	8	9
R4	9	5	1	2	8	**6**	4	3	7
R5	2	**4**	**7**	1	3	9	8	5	6
R6	**6**	8	**3**	**5**	**4**	7	**9**	1	**2**
R7	1	7	**5**	4	**9**	**3**	**2**	6	8
R8	4	9	**6**	8	**2**	1	3	**7**	5
R9	8	3	2	**6**	**7**	**5**	1	9	**4**

Figure 5: Sudoku "B"

	C1	C2	C3	C4	C5	C6	C7	C8	C9
R1	4	1	6	**3**	2	**7**	5	9	8
R2	2	**8**	**3**	4	**5**	9	6	1	7
R3	**9**	7	5	6	1	8	**3**	**2**	**4**
R4	1	9	**7**	**8**	**6**	**2**	4	**3**	5
R5	6	**4**	8	**9**	3	5	1	7	2
R6	**3**	**5**	**2**	1	7	4	9	8	6
R7	7	2	4	5	**9**	**3**	**8**	**6**	1
R8	**5**	3	1	**7**	**8**	6	2	**4**	9
R9	8	**6**	9	2	**4**	1	7	5	**3**

Figure 6: Sudoku "C"

	C1	C2	C3	C4	C5	C6	C7	C8	C9
R1	1	**3**	**6**	7	4	5	**2**	9	**8**
R2	9	5	8	6	3	2	7	4	1
R3	**7**	**4**	**2**	8	1	**9**	**3**	**5**	**6**
R4	**4**	7	9	1	6	3	**5**	**8**	2
R5	3	2	1	5	7	8	4	**6**	9
R6	8	**6**	5	**2**	**9**	**4**	1	7	3
R7	**2**	**9**	3	4	**8**	7	6	1	**5**
R8	6	**8**	4	**3**	5	1	9	**2**	**7**
R9	5	1	**7**	9	2	**6**	8	**3**	**4**

Figure 7: Sudoku "D"

	C1	C2	C3	C4	C5	C6	C7	C8	C9
R1	1	**3**	**2**	**7**	6	4	**8**	5	9
R2	4	9	5	**2**	**8**	**3**	1	**7**	**6**
R3	8	6	7	1	**9**	**5**	2	3	4
R4	9	7	**6**	4	1	8	**5**	**2**	3
R5	2	**8**	3	5	**7**	**6**	**4**	9	1
R6	5	1	**4**	9	**3**	**2**	7	6	8
R7	6	**2**	9	8	5	1	**3**	4	7
R8	**3**	**4**	1	6	2	**7**	9	**8**	5
R9	7	**5**	**8**	3	**4**	9	6	1	**2**

Figure 8: Sudoku "E"

In Table 5 is presented the time to obtain the coverability tree for some Sudoku puzzles and also the number of nodes by level of the tree.

Table 5: Reachability tree size by level and time performance in Radius.

Level	**A**	**B**	**C**	**D**	**E**
1	4	2	4	3	5
2	13	4	6	10	8
3	45	4	16	24	9
4	91	4	8	26	16
5	116	4	14	22	14
6	136	10	24	66	44
7	60	20	20	72	48
8	152	40	44	72	38
9	316	30	128	128	8
10	396	32	334	368	24
11	542	12	444	544	32
12	156	12	192	1136	10
13	130	12	16	1280	10
14	78	12	16	96	12
15	78	24	16	192	22
16	126	24	32	96	20
17	78	96	16	256	48
18	94	240	16	624	52
19	74	324	16	628	120
20	176	452	16	918	20
21	480	588	24	458	12
22	295	880	12	234	6
23	293	692	18	656	16
24	89	320	12	1208	28
25	132	486	24	2352	44
26	98	716	48	1302	29
27	129	470	96	312	6
28	129	574	36	72	10
29	20	738	24	92	2

30	17	891	24	12	2
31	17	288	24	18	4
32	11	204	24	12	2
33	15	132	16	12	2
34	7	36	8	12	4
35	14	84	12	12	8
36	14	84	12	18	12
37	7	96	16	8	12
38	14	59	16	8	12
39	6	2	2	8	12
40	1	4	2	1	4
41	1	8	3	2	4
42	1	4	5	2	4
43	1	4	6	1	4
44	1	2	1	1	2
45	1	1	1	1	1
46	1	1	1	1	1
47	1	1	1	1	1
48	1	1	1	1	1
49	1	1	1	1	1
50	1	1	1	1	1
Total Nodes	4659	8726	1849	13380	807
Time (s)	40.3	60.7	21.6	136.2	6.9

The proposed model, avoids the well known state space explosion by means of a smart search dealing only with feasible assignments, and blocking the exploration of non- feasible assignments.

By the complete reachability graph the solutions were obtained and besides can be confirmed for each case that the solution is unique.

Accordingly with the size of the coverability tree may be measured the complexity of a given Sudoku, regardless the number of given clues.

6 CONCLUSIONS

In this paper was introduced CPN and reachability graph for concurrent and distributed systems analysis.

The Sudoku puzzle was presented as a benchmark problem and some result from artificial intelligence approaches were presented.

A CPN model for the Sudoku problem was introduced and some experimental results about the efficiency and effectiveness for the construction of reachability graph were presented.

The CPN model approach presented is effective to find solution and is extensible for the verification and generation of new Sudoku puzzles.

Based on the reachability graph nodes total number may be ranked each Sudoku, being less difficult those with small graphs, the size of the graph represents a quantitative measure of the complexity for the search solution process.

Further research may be conducted to tackle the minimum givens Sudoku problem stated in [12] by McGuire and Tugemann.

Acknowledgements

This work was supported by CONACYT and ICyTDF through the Scholarship program.

References

[1] Jimenez E. (2005) “Modelado y simulación de sistemas logísticos y de producción mediante redes de Petri”,Revista Iberoamericana de Automática e Informática Industrial, Spain.

[2] Kristensen L.M. (2000) “State space methods for coloured Petri nets”, PhD Dissertation, University of Aarhus Denmark.

[3] Lynce I., (2006) “Sudoku as a SAT Problem”, Proceedings of the 9 th International Symposium on Artificial Intelligence and Mathematics, AIMATH, Fort Lauderdale,U.S.

[4] Moraglio, A., Togelius, J. (2007). “Geometric Particle Swarm Optimization for the Sudoku Puzzle”. In Proceedings of GECCO, London, England.

[5] Mújica M. (2010) “State Space Quantitative Analysis of Timed Coloured Petri Nets for Scheduling Improvements”, PhD Thesis, UAB, Barcelona, Spain.

[6] Nosedal J. (2012) “Causal Decision Support Tools for Strategic Trajectory De-confliction to Enable Seamless Aircraft Conflict Management (STREAM).”. In Proceeding of Second SESAR Innovation Days, Braunschweig, Germany.

[7] Piera M.A. (2004) “Optimization of Logistic and Manufacturing Systems through Simulation: A Colored Petri Net–Based Methodology”. SIMULATION, Vol. 80, Issue 3, The Society for Modeling and Simulation International.

[8] Pillay N. (2012) “Finding Solutions to Sudoku Puzzles Using Human Intuitive Heuristics”, Research Article — SACJ. South Africa.

[9] Ratzer A.V. (2003) "CPN Tools for editing, simulating, and analysing coloured Petri nets". Applications and Theory of Petri Nets 2003: 24th International Conference, ICATPN, Denmark.

[10] Revero A. (1998), "Análisis de redes de Petri mediante su árbol de cobertura", Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería, Madrid, Spain.

[11] Serpen G. (2009) "Survey of the Applications of Artificial Intelligence Techniques to the Sudoku Puzzle", Fundamentals of Intelligence Systems, Technical Report.

[12] Tugemann B. (2012) "There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem", Dublin, Ireland.

DESARROLLO Y VALIDACIÓN EXPERIMENTAL DE UN MODELO DINÁMICO PARA UN VEHÍCULO ELÉCTRICO CON MOTORES EN LAS RUEDAS

David Marcos[a], Carlos Bordons[a], Johan Wideberg[b], Daniel A. Mántaras[c] y Pablo Luque[c]

[a]Ingeniería de Sistemas y Automática – Universidad de Sevilla {dmarcos, bordons@us.es}

[b]Ingeniería e Infraestructura de los Transportes – Universidad de Sevilla, wideberg@us.es

[c]Ingeniería e Infraestructura de los Transportes – Universidad de Oviedo, luque@uniovi.es

Resumen

Este trabajo presenta el desarrollo de un modelo dinámico de un vehículo eléctrico con motores en las cuatro ruedas. El modelo detallado, que se ha realizado en la plataforma ADAMS ® , tiene un total de 14 grados de libertad y en él se han incluido todos los elementos del vehículo real, como muelles helicoidales, amortiguadores telescópicos y ruedas neumáticas. El modelo ha sido validado experimentalmente en el vehículo real, mostrando la bondad del ajuste. Este simulador dinámico servirá como banco de pruebas para mejora del vehículo y para el ensayo previo de controladores de tracción y estabilidad.

Palabras Clave: modelado, vehículo eléctrico, control de tracción, pilas de combustible.

1 INTRODUCCIÓN

Desde la invención del automóvil, los vehículos con motor de combustión interna (VMCI) han sido, con diferencia, los que han dominado el mercado. Aunque siempre ha habido alternativas, éstas nunca han podido compararse en calidad, prestaciones y precio a los VMCI.

Sin embargo, en los últimos años una serie de factores está haciendo que crezca el interés por otro tipo de automóviles. La subida de precios del combustible, así como una mayor concienciación ecológica están llevando a una nueva mentalidad en el consumidor, y en consecuencia a un cambio en el sector. A estos hechos se une el desarrollo de la tecnología en las baterías (LiPo, Li-ion, etc.), y la posibilidad de combinarlas con otro tipo de fuentes como supercondensadores o pilas de combustible [2, 3], reduciendo de esta forma la mayor limitación de los vehículos eléctricos (VE), que es la autonomía.

Es, por tanto, la fuente de energía y la gestión de la misma la principal línea de investigación para los VE; bien sean eléctricos puros, o híbridos. Sin embargo, el uso de motores eléctricos en lugar de motores de combustión interna (MCI) abre una serie de posibilidades en el control del vehículo inéditas hasta el momento. Así, los VMCI, disponen de un único motor, que, mediante una serie de dispositivos mecánicos, aplica un par en las ruedas delanteras, en las traseras o en las cuatro, dependiendo del tipo de vehículo. Este par es repartido por el diferencial de manera igualitaria para todas las ruedas motrices [9].

A diferencia de los MCI, los motores eléctricos no necesitan una velocidad mínima para funcionar (ralentí), entre otras ventajas. Esto, y la sencillez de los propios motores, lleva la posibilidad de utilizar varios en un mismo vehículo, utilizando uno por cada rueda motriz. De esta manera se puede controlar el par cedido por cada eje, consiguiendo un mejor comportamiento dinámico del vehículo, tanto en conducción normal (actuando como un diferencial) como en situaciones de emergencia (control de tracción, estabilidad, etc.) [5, 6, 7]. Incluso este control de par se puede combinar con otros actuadores que optimicen otros parámetros, como los ángulos de caída, avance o la rigidez de la suspensión, como en el caso del vehículo a escala descrito en [7].

En este artículo se presenta el vehículo *FOX,* fabricado específicamente como plataforma de ensayo para distintos controladores tanto de gestión

de potencia como de control del par en cada rueda. A su vez, se presenta un modelo dinámico del vehículo, imprescindible para el desarrollo de los diferentes controladores.

2 DESCRIPCIÓN DEL VEHÍCULO

2.1 VEHÍCULO.

El vehículo FOX está montado sobre el chasis de un coche de competición *Silver Car S2*, ligeramente modificado para la colocación de los nuevos dispositivos. Además, se le ha instalado un segundo asiento, y los elementos de la suspensión han sido reemplazados por otros hechos a medida para los motores en rueda.
El carenado es el mismo que monta el modelo original. Se puede ver una imagen del vehículo, ya montado, en la fig. 1.



Figura 1. Vehículo eléctrico FOX.

El vehículo está propulsado por cuatro motores en rueda *brushless DC*, de 7 kW cada uno (fig. 2). La alimentación de los mismos se hace a partir de seis módulos de baterías LiFeMnPO4, de 12'8 V y 100 Ah, conectados en serie. Entre el conjunto de las baterías y los cuatro motores hay sendos convertidores de potencia. A estos llegará la consigna de par o velocidad (según el modo de funcionamiento) para cada motor.



Figura 2. Motor en rueda.

2.2 SENSORES.

Como se ha explicado, en este trabajo se hace un modelo del vehículo FOX. Para el ajuste de parámetros y la validación del mismo se necesitan una serie de sensores. Estos medirán el comportamiento del vehículo, y de algunos elementos en particular. Además es conveniente medir las consignas del conductor, pues de esta forma será más fácil predecir el comportamiento del vehículo. En el caso del acelerador, su medida es necesaria, ya que, al tratarse de un vehículo eléctrico, esta señal, procesada, será la que se envíe a los controladores de los motores.

2.2.1 Sensores del comportamiento del vehículo.

El vehículo dispone de una serie de sensores que por un lado, permiten conocer el estado de distintas variables internas del mismo y por otro, proporcionan información de las acciones del conductor.

- *Sensores del recorrido de la suspensión* (fig. 3). Se colocan en paralelo a los amortiguadores, midiendo de esta forma la compresión de los mismos en cada momento. De esta manera se puede estimar la fuerza vertical en cada rueda, imprescindible para modelar los esfuerzos en las ruedas.

Características principales:

- Tipo de medición: divisor de tensiones.
- Resistencia nominal: 6kΩ (trasero), 1 kΩ (delantero).
- Fuerza de actuación: 2'45 N (horizontal)
- Alcance: hasta 150 mm
- Máx. velocidad de trabajo: 10 m/s (trasero), 1 m/s (delantero)



Figura 3. Sensor de recorrido.

· *Unidad de Medición Inercial (IMU).* La IMU utilizada es del modelo *3DM-GX3-35* del fabricante *Microstrain*. Mide las aceleraciones lineales y angulares para los tres ejes, así como el campo magnético. Incluye además un receptor *GPS*, con el que se tendrá una medida externa de la posición.
Características principales:

- Rangos: Acelerómetros, ±5g; Giróscopos, ±300º/s; magnetómetro, ±2'5 Gauss; GPS, ±4g, 500 m/s.
- Error de alineación, ±0.05°.
- Precisión GPS (velocidad) 0'1 m/s.

2.2.2 Sensores de los dispositivos del conductor.

- *Lectura del acelerador.*

Señal de 0 a 5 voltios, proporcional a la posición del pedal.

- *Potenciómetro de lectura del freno.*

Características principales:

- Resistencia nominal: 5 kΩ
- Alcance: 10mm

- *Potenciómetro de lectura del giro del volante.*
- *Sensores del recorrido de la suspensión.*

Características principales:

- Resistencia nominal: 10 kΩ
- Precisión: 0'034%

2.2.3 Sensores externos.

- *Básculas por rueda.* Estos sensores se han usado únicamente durante la validación del modelo, para obtener el peso que soporta cada rueda.

3 DESARROLLO DEL MODELO DE SIMULACIÓN POR ORDENADOR DE LA DINÁMICA VEHICULAR

El modelo numérico del comportamiento dinámico del vehículo ha sido desarrollado e implementado en la plataforma ADAMS ®, de MSC Software [10]. Para ello se ha hecho un modelo en CAD del vehículo, utilizando el software SolidWorks, a partir del diseño en tres dimensiones del chasis, aportado por el fabricante.

El modelo desarrollado se muestra en la figura 4. La configuración cinemática del modelo consta de un total de:

- 40 cuerpos
- 15 juntas de revolución
- 16 juntas esféricas
- 3 juntas de traslación
- 4 juntas homocinéticas
- 13 juntas fijas

Se han incluido unas masas de ajuste de los parámetros inerciales para ajustar los valores reales del vehículo, incluyendo los elementos de carrocería no considerados en el modelo CAD y el resto de elementos, sensores y sistemas embarcados en el vehículo.

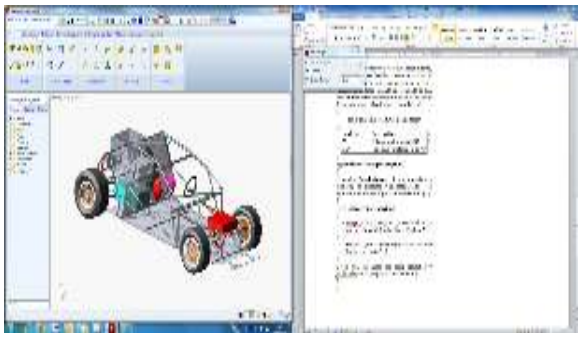


Figura 4. Modelo completo desarrollado en ADAMS

El modelo final tiene un total de 14 grados de libertad. La caracterización dinámica se caracteriza por la modelización de diversos elementos incluidos en el vehículo real, como son los muelles helicoidales, amortiguadores telescópicos y ruedas neumáticas. Para la simulación de las ruedas neumáticas se ha caracterizado el comportamiento de los neumáticos utilizando la denominada Magic Formula de Pacejka [4]. Los parámetros identificativos han sido obtenidos por ajuste de las curvas dinámicas procedentes del fabricante Avon [1].

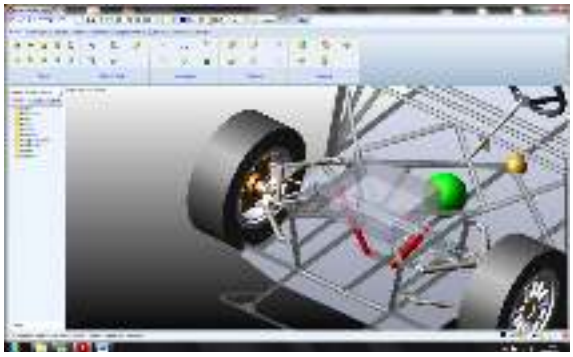


Figura 5. Detalle modelo de suspensión delantera

Los brazos de la suspensión han sido considerados como rígidos y unidos por juntas esféricas al bastidor multitubular del vehículo, considerando que, debido al estado del vehículo, no existen holguras ni fricciones adicionales, con lo que las juntas han sido consideradas como ideales.

El sistema de suspensión delantera (figura 5) modelizado es un mecanismo de doble brazo por rueda, denominado de paralelogramo deformable, con elementos elásticos (muelles helicoidales) y disipativos (amortiguadores hidráulicos telescópicos) accionados por unas bieletas biarticuladas. Los muelles y amortiguadores se consideran lineales en la zona de trabajo habitual y el proceso de ajuste estático y dinámico permitirá determinar las precargas estáticas de los elementos elásticos. La dirección se realiza mediante una cremallera accionada por una timonería unida al volante de dirección.

La configuración cinemática de la suspensión trasera está compuesta por dos brazos, articulados sobre el bastidor. El brazo superior de cada paralelogramo está conectado adicionalmente a las torretas de suspensión fijas al bastidor por un conjunto muelle-amortiguador concéntricos, con características dinámicas inicialmente desconocidas, que deberán ser determinadas en las fases de ajuste.

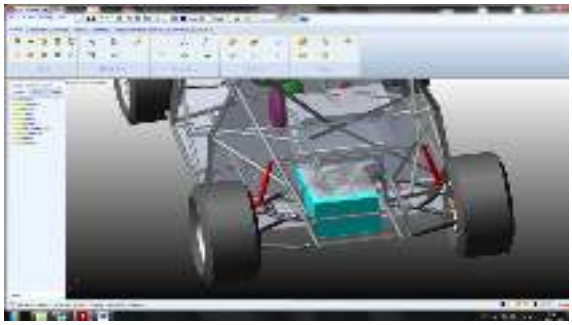


Figura 6. Detalle modelo de suspensión trasera

Para la simulación del modelo se han incluido diversas entradas, tanto del control cinemático como del control dinámico del vehículo. La cinemática está controlada esencialmente por el movimiento de la dirección. El control viene impuesto por el movimiento del volante de dirección.

El control dinámico del vehículo, en lo relativo a la motricidad (tracción/frenado) viene regulado por el par en cada una de las ruedas. En función del control real implementado en el vehículo, se ha optado por un reparto igual del par en las cuatro ruedas. Para imponer el par se implementa un regulador tipo PID, en el que se introducen los valores de velocidad y aceleración longitudinal.

Para garantizar la fidelidad a la realidad de la respuesta dinámica del vehículo se han incluido unas masas representativas de los ocupantes en los puestos de conductor y acompañante, con los valores másicos correspondientes.

4 ENSAYOS Y VALIDACIÓN EXPERIMENTAL

Para la validación del modelo implmentado en ADAMS, se plantea un programa experimental basado en la realización de ensayos estáticos y dinámicos con el vehículo real [11]. La realización del programa experimental implica la utilización de una instrumentación descrita en la sección 2.2.

El programa experimental realizado se basa en la realización de dos tipo de pruebas:

1. Estáticas: Mediante la aplicación de cargas y/o desplazamientos controlados se obtiene información para el ajuste de diversos parámetros del modelo de simulación implementado:
    a. Reparto de peso por rueda y posicionamiento del centro de gravedad
    b. Posición inicial en orden de marcha (alturas, cabeceo y balanceo) de vehículo
    c. Precargas de muelles helicoidales
    d. Desmultiplicación de sistema dirección.
    e. Rigidez de suspensión equivalente en rueda.
2. Dinámicas: La realización de ensayos dinámicos con el vehículo instrumentado, se obtiene información que permite el ajuste y validación del modelo desarrollado. Los ensayos realizados
    a. Aceleración en recta y curva
    b. Frenada en recta y curva
    c. Slalom
    d. Curvas circulares con velocidad constante/creciente
    e. Trayectorias combinadas

A modo de ejemplo, se muestra la simulación de una maniobra combinada de slalom +curvas_circulares + slalom + curva_frenado.

Tras el proceso de ajuste final del modelo, se constata que se reproduce adecuadamente las variables más representativas del movimiento vehicular (trayectoria, velocidades y aceleraciones), así como la respuesta dinámica de elementos

significativos como son la suspensión, dirección y neumáticos.



Figura 7. Maniobra combinada

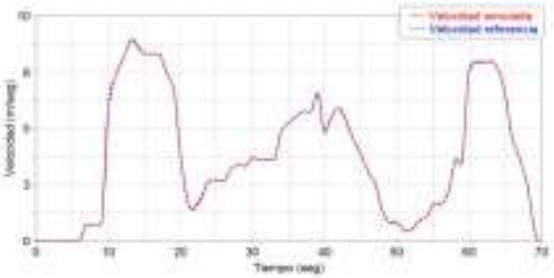


Figura 8. Validación de velocidad simulación frente a ensayo real tomada como referencia.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 CONCLUSIONES

Se ha desarrollado un modelo dinámico de un vehículo eléctrico con motores en las cuatro ruedas, en la plataforma ADAMS®. En el modelo se han incluido todos los elementos del vehículo real, como muelles helicoidales, amortiguadores telescópicos y ruedas neumáticas. El modelo ha sido validado experimentalmente en el vehículo real, mostrando la bondad del ajuste. Se dispone por tanto de un simulador dinámico detallado que puede ser usado como banco de pruebas para mejora del vehículo y para el ensayo previo de controladores de tracción y estabilidad.

5.2 LÍNEAS FUTURAS.

El modelo en ADAMS será empotrado en el entorno *Matlab-Simulink*, y a partir del mismo se desarrollarán controladores para el reparto de par en los cuatro motores. Estos controladores pueden diseñarse teniendo en cuenta diversos criterios, como la eficiencia energética, la seguridad o las prestaciones. Asimismo, se le añadirán los modelos de los dispositivos de potencia, de manera que se puedan modelar distintas estrategias de gestión de potencia. Finalmente estos algoritmos serán utilizados en el vehículo FOX.

Agradecimientos

Este trabajo ha sido financiado en parte por el Ministerio de Economía y Competitividad a través del proyecto DPI2010-21589-C05-01. Los autores desean agradecer a Elena González, Carlos Montero, Alejandro Oliva y Juan Cabrera su inestimable trabajo en la puesta en marcha del vehículo y la realización de las pruebas.

Referencias

[1] AVON (2013), on-line reference http://www.avontyres.es/

[2] C. Bordons, M.A. Ridao, A. Pérez, A. Arce, D. Marcos, (2010), Model Predictive Control for Power Management in Hybrid Fuel Cell Vehicles, IEEE Vehicle Power and Propulsion Conference.

[3] D. Marcos, C. Bordons, M.A. Ridao, (2010), Estudio Comparativo de Sistemas de Almacenamiento de Energía en un Vehículo Híbrido con Pila de Combustible. Jornadas de Automática XXXI.

[4] H.B. Pacejka, (2002) Tyre and Vehicle Dynamics, Butterworth-Heinemann, Oxford, UK.

[5] H. Qian, G. Xu, J. Yan, T.L. Lam, Y. Xu, K. Xu, (2010), Energy Management for Four-Wheel Independent Driving Vehicle. International Conference on Intelligent Robots and Systems

[6] J. Gutiérrez, J. Romo, M.I. González, E. Cañibano, J.C. Merino, (2011), Control Algorithm Development for Independent Wheel Torque Distribution with 4 In-wheel Electric Motors. Fifth UKSim European Symposium on Computer Modeling and Simulation (EMS).

[7] J. Edrén, (2011), Exploring force allocation control of over actuated vehicles. Licentiate thesis, KTH Royal Institute of Technology

[8] K. N. Spentzas, I. Alkhazali, (2001), Generalisation of the concept of electronic differential, Forschung im Ingenieurwesen, Volume 66, Issue 6, pp 273-278, Springer-Verlag.

[9] P. Luque, D.A. Mántaras y C. Vera (2004), Ingeniería del Automóvil. Sistemas y comportamiento dinámico, Thomson, Madrid.

[10] MSC Software, (2013), on-line reference http://www.mscsoftware.com/product/adams.

[11] D.A. Mántaras y P. Luque, (2012), Virtual test rig to improve the design and optimization process of the vehicle steering and suspension systems, Vehicle System Dynamics, vol. 50, nº 10, pág. 1563-1584.

MODELO NARMAX DE PH EN UN FOTOBIORREACTOR TUBULAR DE MICROALGAS

I. Fernández*, G.A. Andrade**, J.L. Guzmán*, M. Berenguel*, D.J. Pagano**
*Departamento de Informática, Universidad de Almería, Ctra. Sacramento s/n 04120 Spain, {ifernandez, joguzman, beren}@ual.es
** Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, 88040-900 Florianópolis, SC, Brazil {gandrade, daniel}@das.ufsc.br

Resumen

Este artículo describe el proceso de identificación y validación de un modelo no lineal NARMAX de pH para un fotobiorreactor tubular. El modelo captura la dinámica de pH para un sistema de producción de microalgas. Estos sistemas regulan habitualmente el estado de pH del cultivo por medio de inyecciones de CO_2 para proporcionar una alta tasa de producción de microalgas. El modelo identificado tiene en cuenta dichas inyecciones, al igual que la influencia de radiación solar como principal perturbación del sistema. Un modelo no lineal del sistema permite una mejor representación de la dinámica del proceso para diferentes puntos de operación, posibilitando la representación de comportamientos dinámicos que no son descritos mediante modelos lineales. El modelo NARMAX es identificado y validado, con el objetivo de utilizarlo en trabajos futuros en el desarrollo de diferentes estrategias avanzadas de control para la regulación de pH.

Palabras clave: Producción de microalgas, Fotobiorreactor tubular, Identificación de procesos, Modelos NARMAX, Mínimos cuadrados ortogonales, Criterio de información Akaike.

1. INTRODUCCIÓN

En la actualidad la mayoría de sistemas que están presentes en una amplia variedad de plantas industriales son no lineales [12]. Un ejemplo de estos sistemas puede encontrarse en los sistemas de producción de microalgas, en los cuales una gran diversidad de productos químicos son obtenidos a partir de la biomasa microalgal [14]. Esta biomasa está compuesta por el carbono que es fijado por las microalgas durante el proceso de fotosíntesis. En los sistemas de producción las microalgas son cultivadas en reactores a gran escala bajo condiciones controladas, obteniendo grandes productividades gracias a una elevada tasa de crecimiento [13].

Para mantener una alta velocidad de crecimiento es necesario controlar ciertas variables de estado del cultivo. Una de las variables más importantes para estos tipos de sistemas es la variable de pH. Su valor es crítico y por ello es necesario mantenerlo en torno a un cierto rango caracterizado por el tipo de microalga a cultivar. El pH es habitualmente controlado mediante estrategias clásicas de control todo/nada, por medio de inyecciones de dióxido de carbono en forma gaseosa, lo que permite aumentar la disponibilidad de carbono inorgánico para el crecimiento microalgal, evitando así la limitación del mismo. Las pérdidas de CO_2 en este tipo de sistemas son superiores al 75 %, con lo que el consumo de dióxido de carbono con el fin de controlar el pH del cultivo representa hasta el 30 % de los costes de producción [3]. Dichas pérdidas pueden ser reducidas por debajo del 30 % mediante un adecuado diseño y operación del reactor [7]. Sin embargo, se ha demostrado que es posible reducirlas aún más mediante el diseño de estrategias avanzadas de control que tengan en cuenta los fenómenos de mezcla y transferencia que tienen lugar en el sistema [4], [9].

Para el análisis y diseño de estrategias avanzadas de control es necesario obtener un modelo que represente con la mayor exactitud posible los fenómenos que tienen lugar en el interior del reactor. Existe una amplia variedad de modelos físicos no lineales en la literatura basado en primeros principios que permiten capturar las dinámicas que tiene lugar en estos tipos de sistemas [2], [7], [8]. Estos modelos físicos son habitualmente complejos, albergando ecuaciones diferenciales difíciles de manipular. Desde un punto de vista de control es necesario desarrollar modelos simplificados que puedan ser utilizados en estrategias avanzadas de control [4]. Una alternativa a los modelos físicos puede ser el uso de técnicas de identificación, mediante las cuales se pueden determinar relaciones matemáticas entre las entradas y las salidas del sistema, sin un conocimiento a priori del mismo.

Este artículo describe el proceso de identificación y validación de un modelo no lineal con el objetivo de capturar las dinámicas no lineales presentes en las mayoría de sistemas biológicos. Para ello se optó por la identificación de un modelo polinomial NARMAX [6]. El modelo identificado captura la

dinámica que relaciona el pH del cultivo con la inyección de CO_2 junto con la radiación, donde dicho modelo pertenece a la familia de modelos multivariables de tipo MISO. El proceso de identificación se ha llevado a cabo mediante el método de mínimos cuadrados ortogonales, utilizando la tasa de reducción del error como criterio para seleccionar la estructura del modelo a identificar, y el criterio Akaike como métrica de la identificación, de forma que el modelo resultante reproduzca exactamente la complejidad estructural del sistema con un menor número de términos posibles. Nótese que como etapa previa a la realización de ensayos de identificación en la planta real, en este trabajo el modelo no lineal basado en principios físicos desarrollo en [8] es utilizado como sistema real.

Este artículo esta organizado de la siguiente forma. En la segunda sección se realiza una breve descripción de los procesos relacionados con la regulación de pH en un sistema de producción de microalgas, junto con una descripción de la planta cuyo modelo se ha utilizado como sistema real para la identificación. En la siguiente sección se explica en detalle el algoritmo utilizado para la identificación del modelo NARMAX. En la cuarta sección se presentan los resultados obtenidos, describiendo los diferentes índices que demuestran la bondad de ajuste del modelo conseguido. Finalmente, en la última sección se muestran las conclusiones obtenidas y trabajos futuros.

2. PRODUCCIÓN DE MICROALGAS

En esta sección se describirán los procesos involucrados en los sistemas dedicados a la producción de microalgas, junto a una breve descripción del tipo de fotobiorreactor utilizado en el modelo basado en principios físicos, que se ha utilizado en este trabajo para realizar los ensayos que han sido llevados a cabo en el proceso de identificación del modelo NARMAX.

2.1. DESCRIPCIÓN DEL PROCESO

Para maximizar el crecimiento global del cultivo y por lo tanto maximizar el rendimiento de cualquier microalga se requiere que las variables de operación sean mantenidas en sus valores óptimos. Entre ellas, el pH es una de las variables críticas, la cual tiene que ser regulada adecuadamente. El pH de un cultivo de microalgas se ve influenciado principalmente por dos fenómenos. Por un lado el aporte del CO_2 como nutriente provoca la formación de ácido carbónico y éste un descenso de pH en el cultivo. Por otro lado, las microalgas realizan la fotosíntesis en presencia de radiación solar consumiendo el CO_2 y generando O_2, lo que provoca una subida paulatina del pH. El CO_2 aportado se transfiere al medio en función del coeficiente de transferencia de materia en el sistema. Además, la variación de la radiación solar disponible, motivada por el ciclo solar diario y la presencia de nubes, produce cambios en la velocidad de fotosíntesis y por tanto en la velocidad de ascenso del pH.

2.2. MODELO DE UN FOTOBIORREACTOR TUBULAR

El modelo utilizado como sistema real, identificado en este artículo, describe las dinámicas no lineales que tienen lugar en un fotobiorreactor tubular de microalgas, donde dicho modelo fue desarrollado y validado en [8]. La identificación realizada sobre un modelo del sistema real es debida a la sensibilidad del cultivo frente a cambios de pH. Por ello en este trabajo describe una primera etapa de identificación sin perjudicar la producción del sistema real. Una vez comprobado los resultados, estos se llevarán a cabo en la planta real.

En la figura 1 se muestra una imagen de los reactores utilizados para el desarrollo del modelo que se utilizó como sistema real, donde dicho modelo ha sido validado bajo condiciones reales de producción. Este reactor tiene, en su conjunto, una capacidad de 2600 L, una longitud de 19.0 m y una anchura de 0.7 m. Se puede dividir en dos partes principalmente, un lazo externo y una columna de burbujeo. El lazo externo está formado por 400 m lineales de tubos de metacrilato de 84 mm de diámetro externo, con una capacidad de 2200 L, unidos mediante conexiones y codos de PVC. El objetivo principal del lazo externo es aumentar la superficie expuesta al sol para que las microalgas puedan captar la mayor cantidad de luz posible cuando realicen la fotosíntesis. La columna de burbujeo tiene 3.25 m de alto por 0.4 m de diámetro, con una capacidad de 400 L, y cumple varias funciones. Por un lado, se utiliza para mezclar el cultivo mediante el burbujeo de aire a un caudal aproximado de 80 L/min, y a su vez, este burbujeo permite la desorción del O_2 producido durante la fotosíntesis por las microalgas en su recorrido por el lazo externo. Por otro lado, en la columna también se lleva a cabo el control de la temperatura del cultivo, mediante un intercambiador de calor tubular. Además de todo esto, en la columna se añaden los nutrientes (medio) y se obtiene el cosechado por un rebosadero situado en la parte superior. El cultivo es recirculado continuamente entre el lazo y la columna mediante una bomba de impulsión situada en la parte inferior de la columna.



Figura 1: Fotobiorreactores Tubulares de la Estación Experimental "Las Palmerillas"de la Fundación Cajamar

3. IDENTIFICACIÓN UTILIZANDO MODELOS NARMAX

En esta sección se describirá el proceso de identificación de un modelo NARMAX por medio de un algoritmo de mínimos cuadrados ortogonales, utilizando el criterio de la tasa de reducción del error para escoger la estructura del modelo NARMAX. Para determinar el número de parámetros a utilizar que mejor represente las dinámicas del proceso se utiliza el criterio de información Akaike. Además se presenta un test de significación estadística para eliminar posibles parámetros que no contengan información relevante del proceso en una etapa final de identificación.

3.1. MODELO NARMAX

Para representar los fenómenos no lineales que se producen en el interior de un fotobiorreactor destinado a la producción de microalgas se ha escogido los denominados modelos NARMAX (*Non-linear AutoRegressive Moving Average model with eXogenous inputs*). Los modelos NARMAX son modelos discretos que predicen la salida del proceso en función de valores previos de la propia salida, de la entrada, y del ruido, pudiendo representar sistemas multivariables de la siguiente forma [11]:

$$\begin{aligned} &y_i(k) = F^i[y_1(k-1),\ldots,y_1(k-n^i_{y_1}),\ldots,\\ &y_m(k-1),\ldots,y_m(k-n^i_{y_m}),u_1(k-1),\ldots,\\ &u_1(k-n^i_{u_1}),\ldots,u_r(k-1),\ldots,u_r(k-n^i_{u_r}),\\ &e_1(k-1),\ldots,e_1(k-n^i_{e_1}),\ldots,e_m(k-1),\ldots,\\ &e_m(k-n^i_{e_m})] + e_i(k) \quad i=1,\ldots m \end{aligned}$$

donde F^i es la i-ésima función no lineal para un sistema con m salidas y r entradas, $y_i(k)$ es la salida i-ésima, $u_i(k)$ es la entrada i-ésima, $e_i(k)$ es el ruido correspondiente a la salida i-ésima, n^i_y, n^i_u y n^i_e son los retardos máximos para la salida, entrada y ruido respectivamente, de la i-ésima función no lineal.

Un modelo NARMAX multivariable es una representación de las funciones F^i en una sumatoria de términos, formando polinomios de grado l^i para cada una de las salidas del sistema. Esta clase de modelos están formados por una parte determinista, y una parte de media móvil que contiene el error. Considerando que el sistema no contiene retardo ni ruido, una aproximación polinomial para un sistema con m salidas y r entradas puede ser escrita por:

$$\begin{aligned} y^i(k) &= \theta^i_0 + \sum_{i_1=1}^{n^i} \theta^i_{i_1} x_{i_1}(k) \\ &+ \sum_{i_1=1}^{n^i}\sum_{i_1=i_2}^{n^i} \theta^i_{i_1 i_2} x_{i_1}(k)x_{i_2}(k) + \ldots \\ &+ \sum_{i_1=1}^{n^i}\ldots\sum_{i_l=i_{l-1}}^{n^i} \theta^i_{i_1\ldots i_l} x_{i_1}(k)\ldots x_{i_l}(k) \\ &+ e^i(k) \end{aligned} \tag{1}$$

siendo

$$\begin{aligned} x_1 = y_1(k-1) \qquad & x_{m\times n^i_y+1} = u_1(k-1) \\ \vdots \qquad & \vdots \\ x_{m\times n^i_y} = y_m(k-n^i_y) \qquad & x_n = u(k-n^i_u) \end{aligned} \tag{2}$$

donde el número de términos, n^i, para representar la i-ésima salida es igual a $m\times n^i_y + r\times n^i_u$ y θ^i los parámetros del modelo.

La utilización de modelos NARMAX para representar modelos no lineales puede ser justificada por diversas razones:

- Es una representación global, permitiendo representar una dinámica global del sistema.
- Se pueden obtener fácilmente informaciones analíticas sobre el modelo.
- Consigue manejar niveles moderados de ruido.
- Se puede utilizar algoritmos simples y robustos para la estimación de sus parámetros.

Se puede observar cómo los modelos NARMAX, a pesar de ser modelos no lineales en sus variables, son lineales en sus parámetros, pudiéndose representar la estructura mostrada en (1) de forma matricial de la siguiente manera:

$$\boldsymbol{y^i} = (\boldsymbol{\Psi^i})^T \hat{\boldsymbol{\theta}}^{\boldsymbol{i}} + \boldsymbol{\xi^i} \tag{3}$$

siendo

$$\begin{aligned} \boldsymbol{y^i} &= \left[\begin{array}{cccc} y(1) & y(2) & \dots & y(N) \end{array}\right]^T, \\ \boldsymbol{\xi^i} &= \left[\begin{array}{cccc} e(1) & e(2) & \dots & e(N) \end{array}\right]^T, \\ (\boldsymbol{\Psi^i})^T &= \left[\begin{array}{cccc} \psi_1(1) & \psi_2(1) & \dots & \psi_{n^i}(1) \\ \psi_1(2) & \psi_2(2) & \dots & \psi_{n^i}(2) \\ \vdots & \vdots & \vdots & \vdots \\ \psi_1(N) & \psi_2(N) & \dots & \psi_{n^i}(N) \end{array}\right] \end{aligned} \tag{4}$$

donde $\boldsymbol{\Psi^i}$ es una matriz que contiene las variables independientes para la i-ésima salida, también llamada matriz regresora, $\hat{\boldsymbol{\theta}^{\boldsymbol{i}}}$ es el vector de parámetros a ser estimados para la i-ésima salida, $\boldsymbol{\xi^i}$ es el vector de residuos de la i-ésima salida, y N el número de datos a utilizar en la identificación.

Los parámetros θ^i de la i-ésima salida del modelo pueden ser escogidos, dentro de un espacio de búsqueda, minimizando la siguiente función de coste:

$$J_{MQ} = (\boldsymbol{\xi^i})^T \boldsymbol{\xi^i} \tag{5}$$

3.2. MÍNIMOS CUADRADOS ORTOGONALES

Una identificación de un modelo no lineal conlleva normalmente identificar un elevado número de parámetros, llegando a un mal condicionamiento numérico. Un método para realizar una estimación precisa es utilizar el algoritmo de mínimos cuadrados ortogonales (MQO) [1].

El algoritmo MQO minimiza la función de coste (5), utilizando la representación matricial del modelo NARMAX (3), siendo $\boldsymbol{\Psi^i}$ una matriz de dimensiones $N \times n^i$, la cual puede ser descompuesta por:

$$\boldsymbol{\Psi^i} = \mathbf{Q}^i \mathbf{R}^i \tag{6}$$

donde $\mathbf{Q}^i$ es una matriz ortogonal ($(\mathbf{Q}^i)^T \mathbf{Q}^i = \mathbf{I}$) de dimensión $N \times n^i$ y $\mathbf{R}^i$ es una matriz triangular superior de dimensión $n^i \times n^i$.

Realizando una descomposición $\mathbf{QR}$ y expandiendo la matriz $\mathbf{Q}^i$ con $(N - n^i)$ columnas ortonormales, de forma que se genere un conjunto de N vectores ortonormales:

$$\tilde{\boldsymbol{Q}}^{\boldsymbol{i}} = \left[\begin{array}{cccc} \mathbf{Q}^i & \tilde{q}_{n^i+1} & \dots & \tilde{q}_N \end{array}\right] \tag{7}$$

se tiene que,

$$\boldsymbol{\Psi^i} = \tilde{\boldsymbol{Q}}^{\boldsymbol{i}} \tilde{\boldsymbol{R}}^{\boldsymbol{i}} = \left[\begin{array}{c} \mathbf{R}^i \\ \mathbf{0} \end{array}\right] \tag{8}$$

Aplicando $(\tilde{\boldsymbol{Q}}^{\boldsymbol{i}})^T$ a los vectores que contiene la salida del sistema se obtiene lo siguiente:

$$(\tilde{\boldsymbol{Q}}^{\boldsymbol{i}})^T \mathbf{y}^i = \left[\begin{array}{c} \mathbf{y}_1^i \\ \mathbf{y}_2^i \end{array}\right] \tag{9}$$

donde $\mathbf{y}_1^i$ es un vector con n^i elementos y $\mathbf{y}_2^i$ es un vector con $(N - n^i)$ elementos.

Reescribiendo la función de coste dada en la ecuación (5), se puede obtener:

$$\begin{aligned} J_{MQO} &= \left(\left[\begin{array}{c} \mathbf{y}_1^i \\ \mathbf{y}_2^i \end{array}\right] - \left[\begin{array}{c} \mathbf{R}^i \\ \mathbf{0} \end{array}\right] \hat{\boldsymbol{\theta}}^{\boldsymbol{i}}\right)^T \left(\left[\begin{array}{c} \mathbf{y}_1^i \\ \mathbf{y}_2^i \end{array}\right] \right. \\ &\quad \left. - \left[\begin{array}{c} \mathbf{R}^i \\ \mathbf{0} \end{array}\right] \hat{\boldsymbol{\theta}}^{\boldsymbol{i}}\right) \\ &= (\mathbf{y}_1^i - \mathbf{R}^i \hat{\boldsymbol{\theta}}^{\boldsymbol{i}})^T (\mathbf{y}_1^i - \mathbf{R}^i \hat{\boldsymbol{\theta}}^{\boldsymbol{i}}) + (\mathbf{y}_2^i)^T \mathbf{y}_2^i \end{aligned} \tag{10}$$

Nótese que J_{MQO} es mínimo cuando $\mathbf{y}_1^i = \mathbf{R}^i \hat{\boldsymbol{\theta}}^{\boldsymbol{i}} = \hat{\boldsymbol{y}}^{\boldsymbol{i}}$. Por tanto, se tiene que $\hat{\boldsymbol{\theta}}^{\boldsymbol{i}} = (\mathbf{R}^i)^{-1} \mathbf{y}_1$.

La ortogonalización de la matriz $\boldsymbol{\Psi^i}$ puede ser realizada por diferentes algoritmos, por ejemplo a través del método de Gram-Schmidt, método modificado de Gram-Schmidt y a través de las transformaciones de Householder. En este trabajo se utilizó una variación del método de transformación Householder denominado Golub-Householder, pudiendo ser utilizado con matrices con deficiencias de rango [10].

El proceso de identificación se mejoró utilizando el criterio de tasa de reducción del error (ERR). El ERR de cada término candidato es un número que indica la cantidad de información que se obtendría con la inclusión de dicho elemento en el modelo.

Para una salida en concreto i, la suma de los valores cuadráticos del vector de errores $\boldsymbol{\xi^i}$ es la parte de la suma cuadrática del vector de salida $\boldsymbol{y}^i$ que no es explicada por los regresores. Por ejemplo, en el caso que no existiera ningún regresor, $n^i = 0$, el error cometido es la propia señal observada, ya que el "modelo" no explica nada de las medidas observadas. La variación del error, $\boldsymbol{\xi^i}$, resultante al añadir un nuevo término al modelo es descrita por un factor $(1/N)[(\hat{y}_p^i)^2((\boldsymbol{q}_p^i)^T \boldsymbol{q}_p^i)]$ [1], donde $\boldsymbol{q}_p^i$ es el vector p-ésimo expresado dentro de la matriz Q^i obtenida en el proceso de ortogonalización de la salida i-ésima, y $\hat{\boldsymbol{y}}_p^i$ es la salida estimada por el modelo debida a la nueva inclusión del parámetro p-ésimo. Nótese que $\hat{\boldsymbol{y}}_p^i = \boldsymbol{r}_p^i \boldsymbol{\theta}_p^i$. De esta manera se

puede cuantificar la importancia de cada regresor individualmente. El ERR de cada término está definido como:

$$[ERR]_p^i = \frac{(\hat{y}_p^i)^2((\boldsymbol{q}_p^i)^T \boldsymbol{q}_p^i)}{(\boldsymbol{y}^i)^T \boldsymbol{y}^i} \ , \ 1 \leq p \leq n^i \qquad (11)$$

ERR indica la porción de variación de salida que es descrita por la inclusión del nuevo término en el modelo. De este modo, puede ser utilizada para la detección de la estructura del modelo no lineal. Los términos con mayor valor de ERR son aquellos términos más deseados para formar el modelo [5].

3.3. CRITERIO DE INFORMACIÓN

Uno de los aspectos más importantes en el proceso de identificación de modelos, es seleccionar el número de parámetros adecuado que represente con la mayor exactitud posible la complejidad estructural que refleje las dinámicas del sistema, sin llegar a producir un mal condicionamiento numérico debido a un número excesivo de parámetros a identificar. Se debe de llegar a un compromiso entre error de sesgo y de varianza.

El problema de determinar el número de parámetros del modelo en sistemas no lineales es mucho más complejo que en sistemas lineales, debido a que el número de estructuras posibles no se incrementa linealmente con el orden del sistema.

Para decidir el número de parámetros utilizados para representar el modelo, puede ser utilizado el criterio de información Akaike, definido por:

$$AIC(n^i) = N \ \ln(\sigma^2_{erro}(n^i)) + 2\, n^i \qquad (12)$$

donde N es el número de datos, $\sigma^2_{erro}(n^i)$ es la varianza del error de modelado y n^i es el número de parámetros del modelo para la salida i-ésima.

El valor AIC será mayor para modelos con errores de modelado o cuando la complejidad del modelo sea mayor que la del sistema real. Este índice atiende a un mínimo para un determinado número de parámetros n^i. El uso de AIC presupone la existencia de un orden predefinido de términos candidatos para ser incluidos en el modelo, cuando el criterio pasa por un mínimo, el proceso puede ser finalizado. Desde un punto de vista del criterio, este número de parámetros es óptimo.

3.4. TEST DE SIGNIFICACIÓN ESTADÍSTICA

Una vez identificado y seleccionado el modelo que mejor representa las dinámicas del proceso, se utiliza, en una etapa final, un test de significación estadística para eliminar posibles términos estadísticamente insignificantes, de forma que si un término cumplen que $-3\sigma_p^i \leq \theta_p^i \leq 3\sigma_p^i$, con σ_p^i desviación típica del parámetro estimado p de la i-ésima salida y θ_p^i el parámetro estimado p, este término puede ser eliminado, dado que se puede asegurar un nivel de 99 % de insignificancia de dicho término en el modelo. Se ejecuta un proceso iterativo en el que los términos no significativos son excluidos del conjunto de términos candidatos y se identifica un nuevo modelo con significancia testada.

En la figura 2 se muestra un diagrama de flujo global del algoritmo de identificación implementado. El algoritmo obtiene un número de modelos, max_nt, por medio del método de mínimos cuadrados ortogonales con tasa de reducción del error, incrementando gradualmente el número de términos usados en cada modelo. A continuación, se calculan los índices Akaikes de cada uno de ellos para compararlos y poder así escoger el modelo con menor índice Akaike posible. Una vez elegido el modelo es expuesto a una última fase, donde se realiza un test de significación estadística es realizado para eliminar los términos que sean insignificantes estadísticamente. Si el ensayo resultado positivo estos términos son eliminados del total de términos candidatos inicial y se vuelve a inicial el proceso de identificación hasta conseguir que todos sean significativos.



Figura 2: Diagrama de flujo del algoritmo de identificación.

4. RESULTADOS

A continuación se describen los resultados de identificación de un modelo no lineal de pH en un fotobiorreactor tubular. Como se explicó en la segunda sección, los cambios de pH del cultivo están influenciados por un descenso producido por el aporte de carbono mediante la inyección de CO_2, y la subida paulatina del mismo debido al proceso fotosintético en presencia de luz solar. Debido a este hecho, se ha identificado un modelo MISO con dos entradas y una única salida que represente dichos fenómenos.

Para la identificación del modelo se realizaron una serie de ensayos sobre el modelo formulado en Ecuaciones Diferenciales Parciales del fotobiorreactor, validado por los autores en trabajos anteriores [9]. Dicho ensayos utilizaron datos reales de radiación y se diseñaron entradas de inyección de CO_2 que permitiesen excitar las principales no linealidades de la planta. Para ello, se escogió un tipo de señal MPRBS (Multi-level Pseudo-Random Binary Sequence) con un periodo relacionado con la constantes de tiempo de pH del sistema, y considerando las diferentes dinámicas apreciadas en presencia de radiación solar o sin ella.

El proceso de identificación se realizó para una batería de 20 ensayos con diferentes entradas de radiación solar y caudal de inyección de CO_2, de forma que se pudiera identificar tanto las dinámicas rápidas producidas por la inyección de CO_2 como las lentas por la radiación solar. Estos ensayos se dividieron de forma que 10 de ellos se utilizarón para la calibración de los parámetros y otros 10 para la validación.

Se eligió un grado de no linealidad $\ell^1 = 2$, y un número máximo de retardos en las señales de entrada y salida de $n_y^1 = n_u^1 = n_u^2 = 15$, resultando un total de 1081 términos candidatos para formar la estructura del modelo. El número máximo de términos a utilizar fue de 30 términos.

En la figura 3 se representan los valores del criterio Akaike obtenidos para diferentes términos del modelo. El menor valor de AIC corresponde a un modelo con 18 términos. En la tabla 1 se puede observar la estructura del modelo identificado, mientras que en la figura 4 se muestran los resultados de calibración del modelo para los 10 ensayos realizados.

Para cuantificar la calidad del modelo, índices como el coeficiente de determinación (ver ecuación (13)), el error cuadrático medio, o la desviación típica del error pueden ser utilizados:

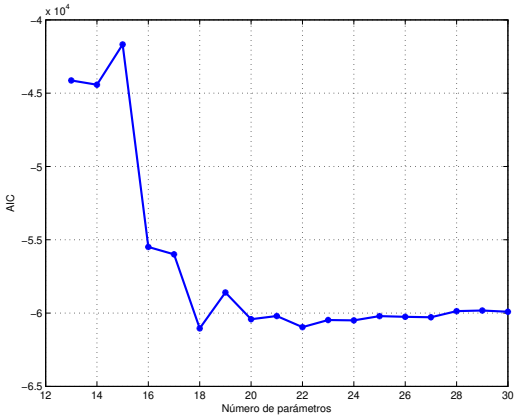


Figura 3: Criterio Akaike modelos identificados.

$$R^2 = 1 - \frac{\sum_{k=1}^{N}[y(k) - \hat{y}(k)]^2}{\sum_{k=1}^{N}[y(k) - \bar{y}]^2} \qquad (13)$$

siendo $y(k)$ la salida real del sistema, $\hat{y}(k)$ la salida estimada e $\bar{y}$ la media de las N muestras de la salida real del sistema. Cuando el valor de R^2 es igual a 1 indica una ajuste exacto del modelo para los datos medidos del proceso. El valor de R^2 entre 0.9 y 1 puede ser considerado suficiente para muchas aplicaciones prácticas en identificación.

Los resultados obtenidos en el proceso de calibración muestran un coeficiente de determinación de 0.931, por lo que se puede concluir que describe el sistema en un 93.1 %. Además el error cuadrático medio fue de 0.011 y la desviación típica de 0.4.

Tabla 1: Términos del modelo NARMAX identificado, y_1 (pH del cultivo), u_1 (radiación solar) y u_2 (inyección de CO_2)

Orden	Regresor	θ	σ
1	$y_1(k-1)$	1.5958	0.0055
2	$y_1(k-2)$	-0.81477	0.00997
3	$y_1(k-3)$	0.19266	0.006
4	$y_1(k-1)u_2(k-5)$	-0.001664	$5.25\ 10^{-5}$
5	$u_2(k-8)$	0.01283	0.00104
6	$u_2(k-6)y_1(k-10)$	0.023111	0.00121
7	$y_1(k-11)$	-0.030913	0.00301
8	$y_1(k-6)$	0.060876	0.00377
9	$u_2(k-5)u_2(k-8)$	0.00053689	0.0001
10	$u_2(k-8)^2$	-0.0054817	0.0001
11	$u_2(k-6)y_1(k-9)$	-0.026601	0.0012
12	$u_2(k-8)u_2(k-6)$	0.0037961	0.00012
13	$u_1(k-1)$	$1.5085\ 10^{-5}$	$6.18\ 10^{-7}$
14	$u_2(k-5)^2$	0.001589	0.0001
15	$u_2(k-8)y_1(k-15)$	0.0028153	0.00014
16	$y_1(k-10)^2$	-0.0042007	0.00025
17	$y_1(k-7)^2$	0.00376	0.00025
18	$u_2(k-5)u_1(k-7)$	$-5.1385\ 10^{-6}$	$3.48\ 10^{-7}$

En la figura 5 se muestran los resultados obtenidos en la fase de validación. Los resultados obtenidos muestran un coeficiente de determinación de 0.901, de forma que el 90 % de los datos son descritos mediante el modelo identificado. Por otro lado, el error cuadrático medio fue de 0.015 y la desviación típica de 0.39.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Se ha identificado un modelo polinomial NARMAX MISO que representa la variable de estado de pH de un fotobiorreactor tubular en un sistema de producción de microalgas, teniendo en cuenta tanto la inyección de CO_2 utilizada para su regulación, como la radiación solar como principal perturbación sobre el estado de pH del cultivo. El modelo se ha validado de forma adecuada capturando la dinámica del sistema de forma satisfactoria.

Como futuros trabajos, se identificará el modelo mediante datos reales obtenidos de la instalación y se desarrollará una estrategia de control avanzada NMPC (*Non-linear Model Predictive Control*) que utilice dicho modelo con el objetivo de mejorar la regulación de pH del sistema.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los siguientes proyectos: PHB2009-0008 (financiado por el Ministerio Español de Educación, CNPq; CAPES-DGU 220/2010); DPI2010-21589-C05-04 y DPI2011-27818-C02-01 (financiados por el Ministerio de Economía y Competitividad de España y los fondos FEDER).

Referencias

[1] Aguirre, L. A. and Rodrigues, G. G. and Jácome, C. R. F., (1989) Identificação de sistemas não lineares utilizando modelos NARMAX polinomiais - Uma revisão e novos resultados, Controle e Automação, volume 9, 2, 90-106.

[2] Becerra-Celis, G., Hafidi, G., Tebbani, S., Dumur, D., Isambert, A., (2008) Non-linear prediction control for continuous microalgae cultivation process in a photobioreactor, *Proceeding of the 10th International Conference on Control, Automation, Robotics and Vision*, 1373–1378.

[3] Beneman, J.R., Tillet, D.M., Weissman, J.C., (1987) Microalgae biotechnology, *Trends Biotechnol.*, 5, 47–53.

[4] Berenguel, M., Rodríguez, F., Acién, F., García, J., (2004) Model prediction control of ph in tubular photobioreactors., *Process Control*, 14 (4), 377–387.

[5] Billings, S. A. and Chen, S. and Korenberg, M. J., (1989) Identification of MIMO nonlinear systems using a forward regression orthogonal estimator, *International Journal of Control*, volumne 49, 6, 2157-2189.

[6] Dallagnol Filho, V. A., Pagano, D. J., Plucenio, A. (2006) Identification of polynomial NARMAX models for an oil well operating by continuous gas-lift, *International Symposium on Advanced Control of Chemical Processes Symposium - ADCHEM. Gramado*, Brasil.

[7] F. Camacho Rubio, F.G. Acién Fernández, J.A. Sánchez Pérez, F. García Camacho, E. Molina Grima, (1999) Prediction of dissolved oxygen and carbon dioxide concentration profiles in tubular photobioreactors for microalgal culture, *Biotechnology Bioengineering*, 62 (1) 71–86.

[8] Fernández, I., Acién, F.G., Fernández, J.M., Guzmán, J.L., Magán, J.J., Berenguel, M., (2012) Dynamic model of microalgal production in tubular photobioreactors, *Bioresource Technology*, volume 126, 172 - 181.

[9] Fernández, I., Peña, J., Guzmán, J.L., Berenguel, M., Acién, F., (2010) Modelling and Control Issues of Ph in Tubular Photobioreactors., *11th IFAC Symposium on Computer Applications in Biotechnology*, 11 (1), 186-191.

[10] Golub, G. H. and Van Loan, C. F., (1989) Matrix computations, *Johns Hopkins University Press*, Baltimore.

[11] Leontaritis, I. J. and Billings, S. A., (1985) Input-output parametric models for nonlinear systems. Part I: deterministic nonlinear systems; Part II: stochastic nonlinear systems, *International Journal of Control*, volume 41, 2, 303-344.

[12] Nijmeijer, H. and van der Schaft, A., (1990) Nonlinear Dynamical Control Systems, *Springer-Verlag*, New York.

[13] Pulz, O., (2001) Photobioreactors: production systems for phototrophic microorganisms, *Applies Microbiology and Biotechnology*, 57, 287–293.

[14] Pulz, O., Gross, W., (2004) Valuable products from biotechnolgy of microalgae, *Ed. Balckwell Science*, UK.

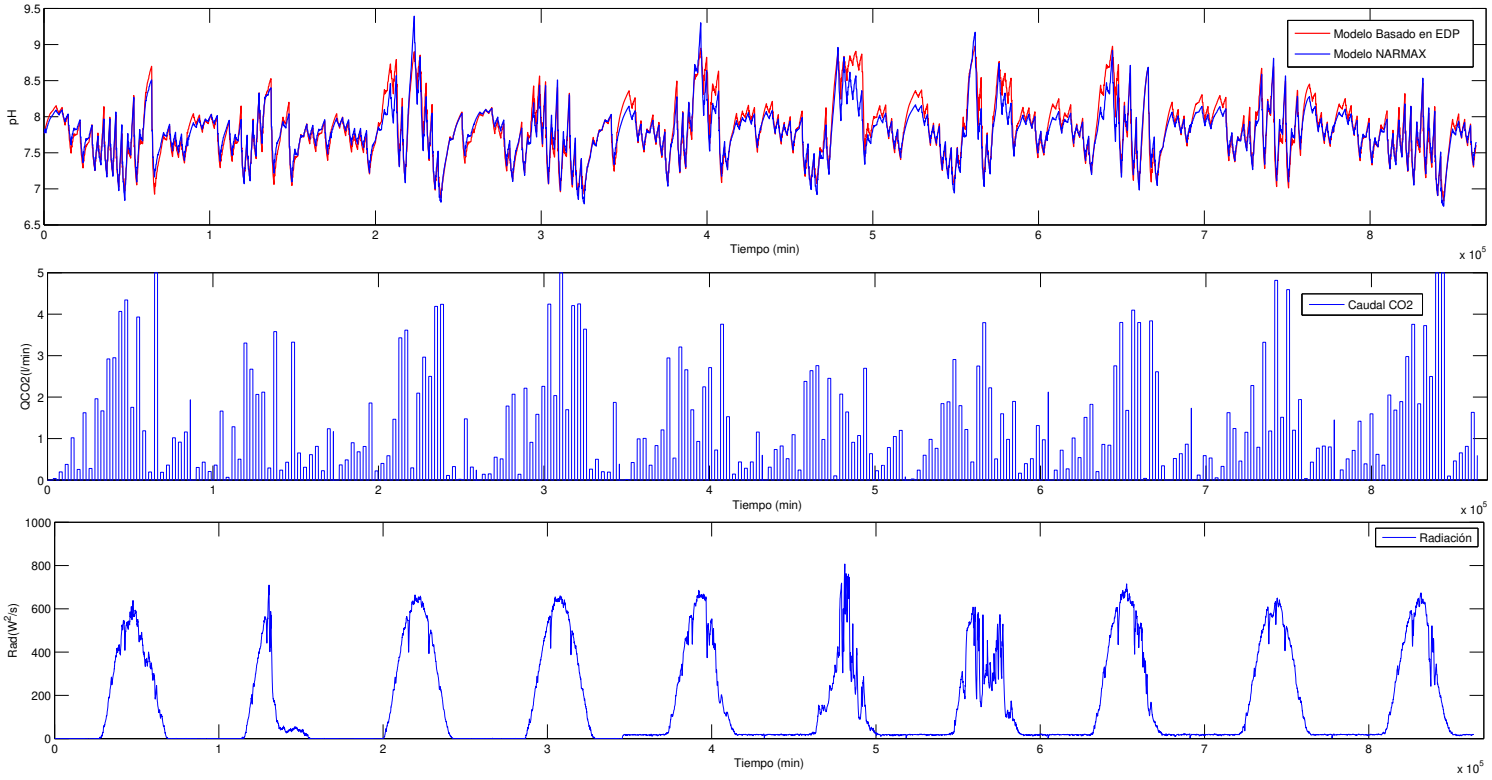


Figura 4: Ensayos realizados en el proceso de identificación del modelo NARMAX.

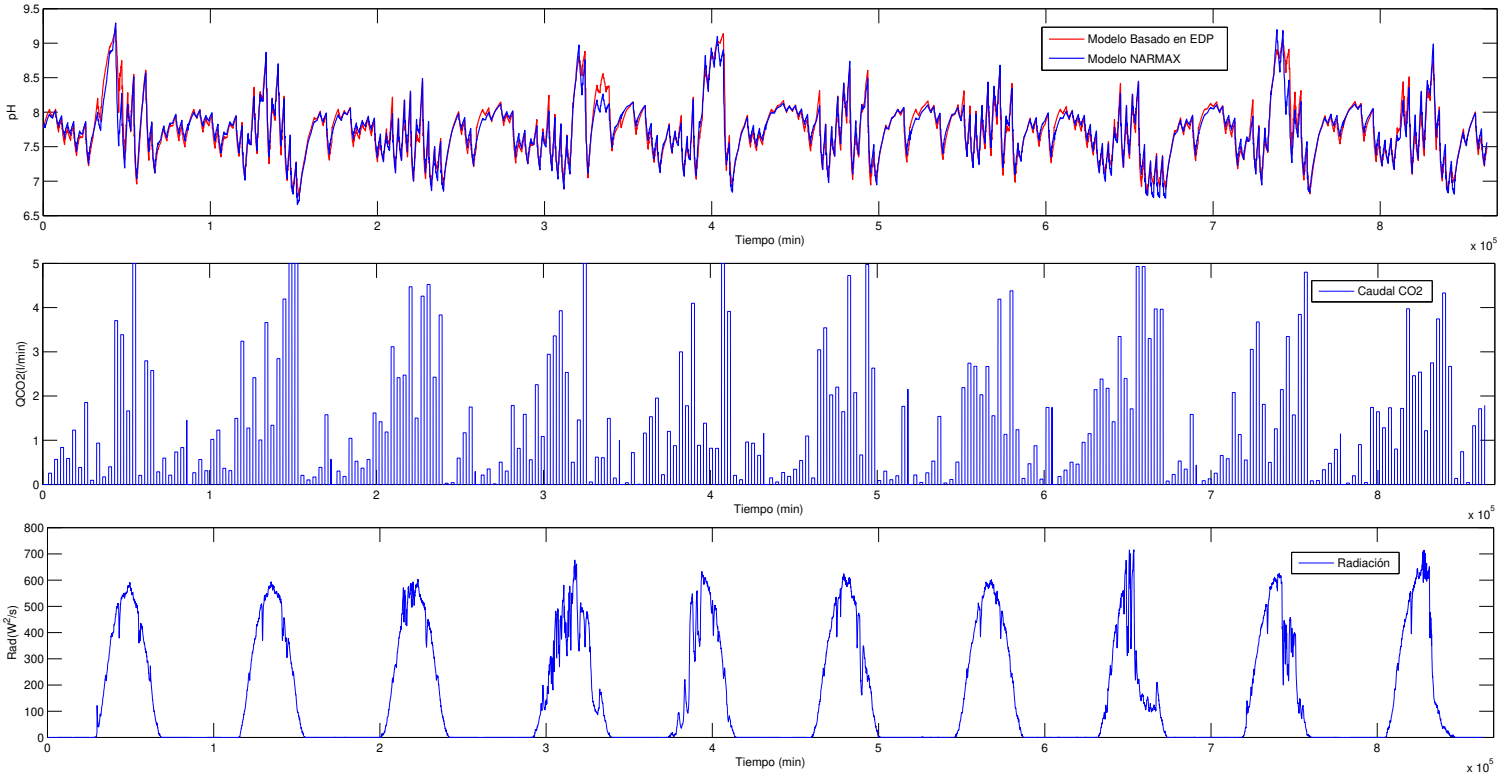


Figura 5: Validación del modelo NARMAX comparado con el modelo EDP bajo perturbaciones reales de irradiación solar.

Identificación de un modelo pasivo para la grúa Inteco 3D Crane e implementación en EJS

Pablo Falcón, Antonio Barreiro, Miguel D.Cacho, Emma Delgado

pfalcon@uvigo.es, abarreiro@uvigo.es, mcacho@uvigo.es, emmad@uvigo.es

Resumen

En este trabajo se ha estudiado y descrito el modelo de la grúa tipo pórtico Inteco 3DCrane por medio de la mecánica hamiltoniana y se ha realizado la identificacion de los parámetros del modelo mediante datos experimentales. Gracias a la utilización de las expresiones de energía cinética y potencial se han establecido las interacciones entre los distintos elementos de la grúa. El modelo propuesto en este trabajo se orienta hacia el enfoque de pasividad, por lo que se comprueba la pasividad del modelo obtenido. Para la simulación del modelo de la planta se ha utilizado EJS, que permite la fácil implementación de la dinámica del sistema y su análisis.

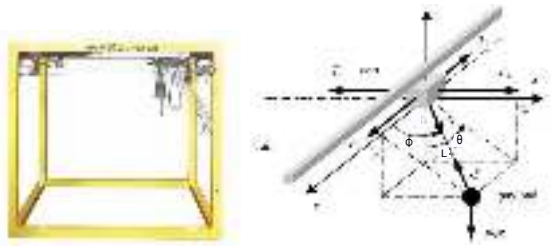
Palabras clave: Identificación, pasividad, grúa, EJS.

1 Introducción

Las grúas pórtico son ampliamente utilizadas para el transporte de mercancias y son un perfecto ejemplo de control complejo. Para la elaboración de este trabajo se ha utilizado la grúa 3DCrane de Inteco (Figura 1), que es perfecta para las pruebas y toma de datos experimentales en el laboratorio. La planta ha sido modelada previamente en otros trabajos como [4] o [2], pero ha sido propuesto un nuevo modelo que pruebe las características de pasividad de la grúa real. Este enfoque de pasividad es muy útil para demostrar la estabilidad del sistema, por lo que resulta muy interesante obtener los parámetros de este modelo para diseñar de manera óptima el control de la grúa.

En este capítulo se identifica las ecuaciones que definen el sistema mecánico teórico para luego ser comparado con el sistema real. El modelo mecánico utilizado es el que se muestra en la figura 2.

Este esquema se ha definido para el eje x, pero el eje y puede ser definido de la misma manera.



(a) Grúa 3DCrane (b) Parámetros de la grúa

Figura 1: Descripción Inteco 3DCrane.

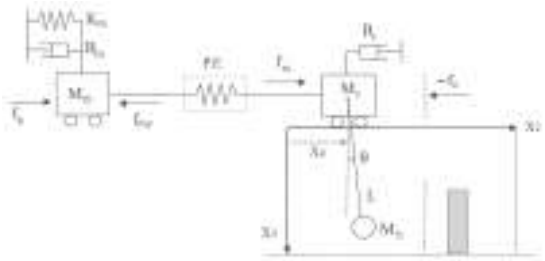


Figura 2: Modelo mecánico de la grúa Inteco 3DCrane

El eje z debe tratarse a parte ya que su comportamiento no tiene nada que ver con esto. El eje z simplemente representa la variación de la longitud del cable.

El modelo mecánico puede ser definido con la formulación hamiltoniana y es muy útil para la conexión con otros sistemas mediante puertos hamiltonianos de potencia, [5]. Esta definición hamiltoniana también permite obtener el balance de energía y determinar la pasividad del sistema. Entonces, se puede definir la dinámica del sistema como:

$$\sum : \begin{cases} \dot{x} = J(x)\frac{\partial H}{\partial x}(x) + g(x)u, \\ y = g^{\top}(x)\frac{\partial H}{\partial x}(x) \end{cases} \quad (1)$$

El hamiltoniano del sistema viene definido por la energía cinética y potencial del sistema mecánico

de la Figura 2 como:

$$H = K + P \tag{2}$$

Donde la energía cinética es K y la energía potencial es P y definidas de la siguiente manera:

$$K = \frac{1}{2} M_c \dot{x_c} + \frac{1}{2} M_b (\dot{x_c} + L\dot{\theta})^2 \tag{3}$$

$$P = M_b g L (1 - \cos\theta) \tag{4}$$

Es posible hacer una simplificación en la expresión de la energía potencial por medio del polinomio de Taylor, tal que (5). Haciendo la selección de los dos primeros elementos de la serie de Taylor para $\cos\theta$. La ecuación de la energía de potencial es (6).

$$\cos\theta = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} \theta^{2n} = 1 - \frac{\theta^2}{2} + \frac{\theta^4}{4} - ... \tag{5}$$

$$P = M_b g L (1 - \cos\theta) \simeq M_b g L \frac{\theta^2}{2} \tag{6}$$

Se deben hacer algunas definiciones antes de continuar. El vector de estados está formado por los venctores p y q. El vector q está compuesto por la posición (x_c) y el angulo entre la vertical y el cable del peso suspendido (θ):

$$q = \begin{pmatrix} x_c \\ \theta \end{pmatrix} \tag{7}$$

Matricialmente el sistema quedaría como:

$$K = \frac{1}{2} \begin{pmatrix} \dot{x_c} & \dot{\theta} \end{pmatrix} \begin{pmatrix} M_c + M_b & LM_b \\ LM_b & L^2 M_b \end{pmatrix} \begin{pmatrix} \dot{x_c} \\ \dot{\theta} \end{pmatrix} = \frac{1}{2} \dot{q}^T M_s \dot{q} \tag{8}$$

$$P = \frac{1}{2} \begin{pmatrix} x_c & \theta \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & M_b g L \end{pmatrix} \begin{pmatrix} x_c \\ \theta \end{pmatrix} = \frac{1}{2} q^T K_s q \tag{9}$$

Entonces, el sistema descrito en ecuaciones hamiltonianas sería:

$$H = K + P = \frac{1}{2} \dot{q}^T M_s \dot{q} + \frac{1}{2} q^T K_s q \tag{10}$$

Si se define el vector de momentum p como (11) y la matriz M_s se define como la matriz de inercias o matriz de masas genéricas, que es simétrica y positiva, la energía cinética puede escribirse como (12).

$$p = M_s \dot{q} \tag{11}$$

$$K = \frac{1}{2} \dot{q}^T M_s \dot{q} = \frac{1}{2} p^T M_s^{-1} p \tag{12}$$

Sustituyendo esto en la ecuación del hamiltoniano del sistema se obtiene finalmente:

$$H = K + P = \frac{1}{2} p^T M_s^{-1} p + \frac{1}{2} q^T K_s q \tag{13}$$

2 Descripción del modelo pasivo de la grúa

El sistema se define genéricamente como:

$$\begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \begin{pmatrix} \nabla_q H \\ \nabla_p H \end{pmatrix} + \begin{pmatrix} 0 \\ \lambda \end{pmatrix} u \tag{14}$$

Donde $\lambda = (1\ 0)^T$

$$\nabla_q H = K_s q = \begin{pmatrix} 0 & 0 \\ 0 & M_b g L \end{pmatrix} \begin{pmatrix} x_c \\ \theta \end{pmatrix} = \begin{pmatrix} 0 \\ M_b g L \theta \end{pmatrix} \tag{15}$$

$$\nabla_p H = M_s^{-1} p = M_s^{-1} M_s \dot{q} = \begin{pmatrix} \dot{x_c} \\ \dot{\theta} \end{pmatrix} \tag{16}$$

Entonces:

$$\begin{pmatrix} \dot{x_c} \\ \dot{\theta} \\ M_s \begin{pmatrix} \ddot{x_c} \\ \ddot{\theta} \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ M_b g L \theta \\ \dot{x_c} \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u \tag{17}$$

De la ecuación (17) se puede comprobar que el sistema mecánico cumple:

$$M_s \begin{pmatrix} \ddot{x_c} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} u \\ -M_b g L \theta \end{pmatrix} \Rightarrow \begin{pmatrix} \ddot{x_c} \\ \ddot{\theta} \end{pmatrix} = M_s^{-1} \begin{pmatrix} u \\ -M_b g L \theta \end{pmatrix} \tag{18}$$

$$\begin{pmatrix} \ddot{x_c} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{1}{M_c} & \frac{-1}{LM_c} \\ \frac{-1}{LM_c} & \frac{1}{L^2}(\frac{1}{M_c} + \frac{1}{M_b}) \end{pmatrix} \begin{pmatrix} u \\ -M_b g L\theta \end{pmatrix} \tag{19}$$

Estas ecuaciones pueden ser descritas en diagrama de bloques (Figura 3). En esta figura se puede ver la interacción del carro de la grúa sobre la bola suspendida y las fuerzas provenientes de la bola que afectan, a su vez, al carro.

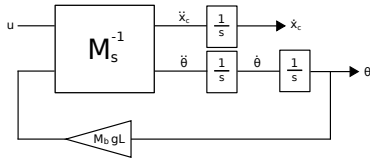


Figura 3: Diagrama de bloques de la grúa (I)

En este caso, lo interesante es realizar una disposición de bloques que permita demostrar la pasividad del sistema. Para ello, se pueden reubicar los bloques como muestran las figuras 3, 4, 5 y 6 y establecer unas relaciones más sencillas de analizar.

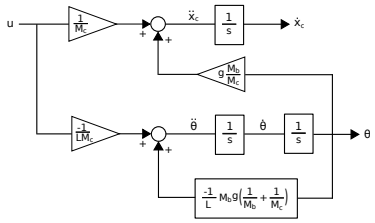


Figura 4: Diagrama de bloques de la grúa (II)

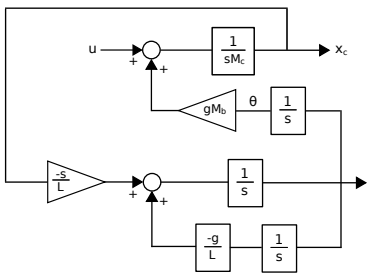


Figura 5: Diagrama de bloques de la grúa (III)

Entonces, el diagrama de bloques de la Figura 6 se puede resumir en $H_1(s)$ and $H_2(s)$, en la Figura 7. Se puede identificar la pasividad de los bloques si se descompone esta figura en bloques simples pasivos. Los sistemas realimentados negativamente son pasivos si los todos los bloques del lazo cerrado son pasivos, por lo que, los subsistemas descritos en las Figuras 8, 9 y 10 son pasivos y, por consiguiente, el sistema total (Figura 6) es

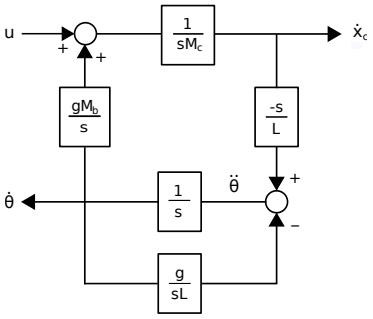


Figura 6: Diagrama de bloques de la grúa (IV)

pasivo por estar compuesto por elementos pasivos.

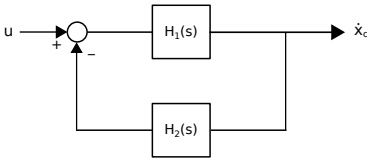


Figura 7: Diagrama de bloques de la grúa (V).

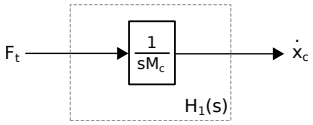


Figura 8: Descripción de $H_1(s)$.

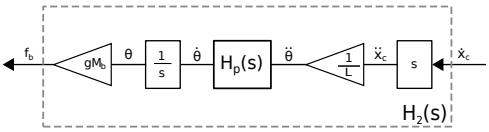


Figura 9: Descripción de $H_2(s)$.

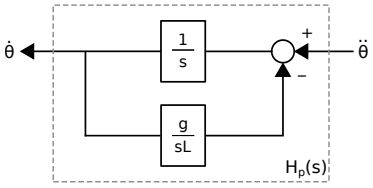


Figura 10: Descripción de $H_p(s)$.

Al modelo también se le debe añadir la resistencia viscosa dinámica que se descompone en varios elementos como se puede ver en la figura 11, la fricción viscosa del movimiento del carro y la fricción viscosa en el movimiento pendular del peso. Simplificando las no linealidades de estos elementos de fricción se puede comprobar la pasividad del bloque mediante su respuesta frecuencial. En todo caso, las fricciones viscosas

que muestran zonas muertas no lineales se pueden asumir como pasivas ya que no proporcionan energía al sistema en ningún caso. Mediante el diagrama de Nyquist (Figura 12) del sistema con fricciones viscosas lineales se puede comprobar la condición de la ecuación (20) como se establece en [6], por lo tanto, esta condición permite decir que el sistema es pasivo.

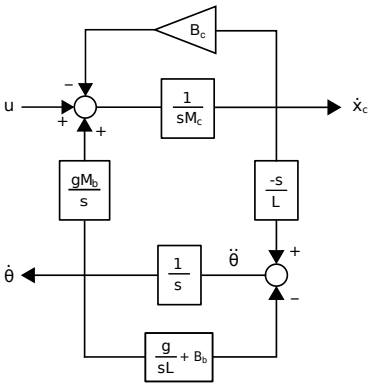


Figura 11: Descripción del modelo con amortiguamiento.

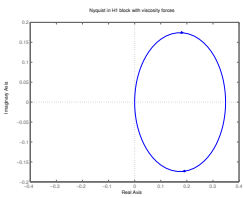
$$Re[H_n(j\omega) \geq 0] \tag{20}$$

Finalmente, el modelo para el eje z, correspondiente a la variación de la longitud del cable, se representa como una función de transferencia de primer orden con una constante de tiempo T_z y una ganancia estática K_z.

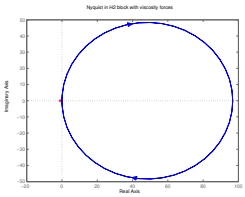
3 Identificación de los parámetros del modelo de la grúa

Para la identificación de la planta en los tres ejes espaciales se ha utilizado datos experimentales extraídos de la planta real. El modelo ha sido sometido a las mismas consignas de entrada que en los experimentos para obtener los valores de los coeficientes de amortiguamiento y de zona muerta. Esta consigna de entrada ha sido la misma para todos los ejes de la grúa.

En los ejes x e y, que se corresponden con el plano horizontal de movimiento de la grúa, se muestran las variaciones de los correspondientes ángulos que se producen con el movimiento. En la Figura 13 se puede observar el desplamiento en el eje x ante la consigna de entrada y el ángulo producido θ. Lo mismo ocurre con el eje y, en la Figura 14 y el ángulo ϕ. Finalmente, se



(a) Diagrama de Nyquist H_1



(b) Diagrama de Nyquist H_2

Figura 12: Diagrama de Nyquist H_1 y H_2.

muestra la respuesta para el eje z, en la Figura 15.

En estos experimentos se ha utilizado la medida de posición ya que es la variable que nos permite medir el sensor de la planta. Por lo tanto, el modelo de la planta, que tiene como salida velocidad, se ha integrado la señal para obtener las medidas de posición y así poder comparar los resultados obtenidos con los datos experimentales del modelo.

Los parámetros conocidos previamente son los relacionados con las dimensiones físicas de la grúa:

- Longitud del cable, $L = 645$ [mm]
- Masa del carro en el eje X, $M_c = 8.45$ [kg]
- Masa del carro en el eje Y, $M_c = 1.69$ [kg]
- Masa de la bola, $M_b = 0.845$ [kg]

Los datos de la velocidad del carro frente a la entrada de la planta, en este caso fuerza, permiten hallar el valor de los coeficientes de amotiguamiento o de fricción viscosa del carro en los ejes X e Y. Para obtener el valor de amortiguamiento del peso del cable se realiza una prueba durante un largo periodo, obteniéndose el coeficiente de amortiguamiento de la oscilación. Los coeficientes son los siguientes:

- Coeficiente de amortiguamiento del carro en el eje X, $B_c = 3.264$ [kg/s]
- Coeficiente de amortiguamiento del carro en el eje Y, $B_c = 2.850$ [kg/s]
- Coeficiente de amortiguamiento de la bola, $B_b = 0.0075$ [1/s]

En el modelo aparece una resistencia al movimiento debida a los elementos mecánicos del sistema. Este efecto es modelado como una zona muerta o *deadzone* a la entrada del modelo de la planta. El ajuste de las zonas muertas del

sistema mecánico determina en gran medida el comportamiento del modelo y los parámetros obtenidos de la identificación son:

- Zona muerta eje X
  - límite superior = 5.2 [N]
  - límite inferior = -5.6 [N]
- Zona muerta eje Y
  - límite superior = 3.4 [N]
  - límite inferior = -2.4 [N]
- Zona muerta eje Z
  - límite superior = 1.2 [N]
  - límite inferior = -4.8 [N]

Finalmente se realiza la identificación de los valores de la función de transferencia correspondiente al eje Z. Los resultados son los siguientes:

- $K_z = 0.15$
- $T_z = 0.039$ [s]

En las figuras se distinguen las señales de consigna de entrada, que corresponde con la línea negra discontinua; las señales de la respuesta de movimiento lineal del carro en los dos ejes, que se corresponden con la línea verde para los valores reales y en azul para los valores del modelo; y por último, las señales de los ángulos (sólo en las figuras 13 y 14), los valores de los ángulos reales de la planta se representan con la línea roja y la correspondiente señal de los ángulos del modelo en línea negra.

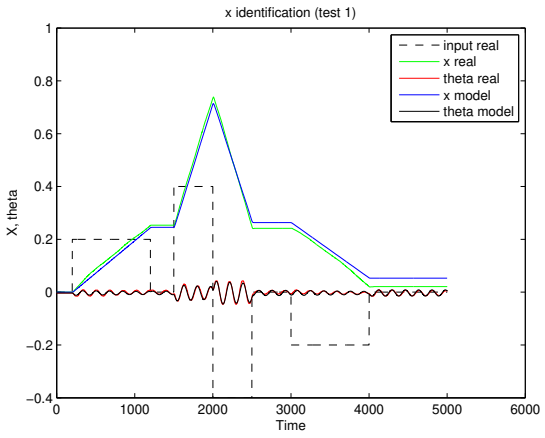


Figura 13: Identificación eje X y ángulo θ.

En las gráficas de identificación se puede observar la concordancia entre los datos recogidos del

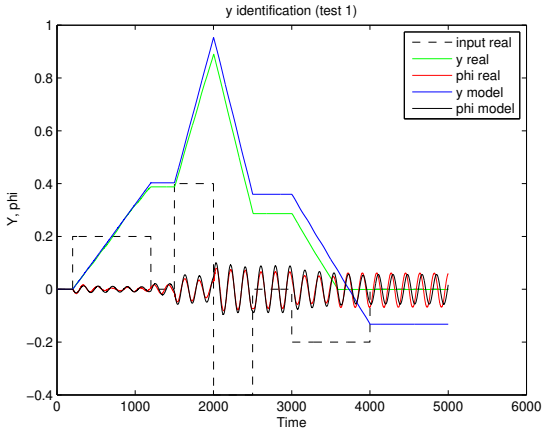


Figura 14: Identificación eje Y y ángulo ϕ.

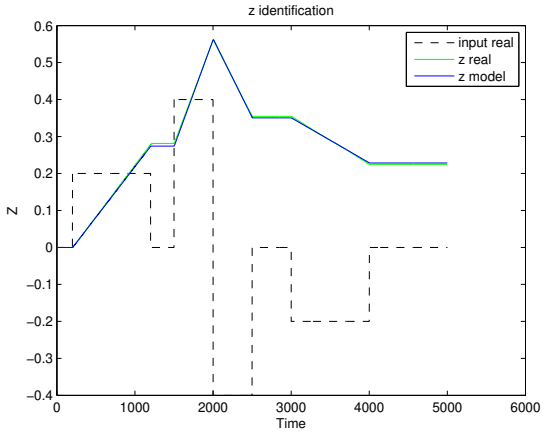


Figura 15: Identificación eje Z.

modelo teórico obtenido de las ecuaciones hamiltonianas y los datos recogidos de la planta real en el laboratorio. Existe cierta discrepancias de offset, esto está relacionado con la zona muerta y la fricción viscosa seca de arrancada y parada en el movimiento del carro de la grúa y que no puede ser ajustado perfectamente.

4 Control de las posiciones del carro y de los ángulos de balanceo de la grúa

Para el control de esta planta se ha optado por la aplicación de unos controladores PID sobre la posición del carro y sobre los ángulos θ y ϕ. El desplazamiento del carro es a través de los ejes x e y correspondientes al plano horizontal de la grúa. Y el eje z se corresponde con la longitud del cable que sujeta el peso suspendido de la grúa.

Para la correcta sintonización de los controladores bajo el marco de pasividad se debe tener en cuenta las características de pasividad del controlador para obtener un conjunto controlador-planta pasivo, [1]. En este caso, el estudio no se ha centrado en esta premisa y solamente se establece la posibilidad de sintonizar los parámetros del controlador K_p, K_d y K_i para todos los controladores.

En el caso del control de posición del carro y longitud de cable, el modelo de la grúa trabaja con consignas de fuerza y velocidad. Por lo tanto, es necesario incluir un bloque integrador a la salida de la planta para obtener la medida de posición y así, poder realizar el control en posición de los ejes x,y,z (Figura 16).

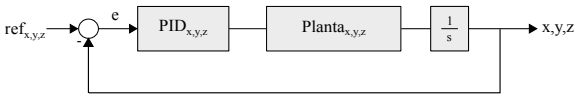


Figura 16: Lazo de control de posición para la planta en los ejes x,y,z.

Por otro lado, existe la posibilidad de controlar de igual modo el ángulo del cable del peso mediante otro lazo de control, ver la figura 17. Este lazo de control se puede utilizar para obtener una menor oscilación del objeto colgado del carro si se establece un valor de referencia ($ref_{\phi,\omega}$) nulo.

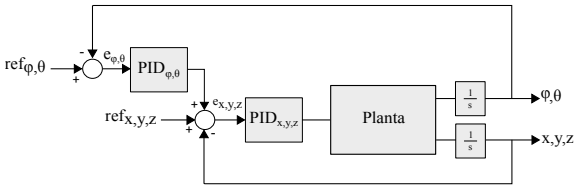


Figura 17: Lazo de control del ángulo con la vertical del cable θ y ϕ.

En la siguiente sección del artículo estos diseños de control se simulan sobre el modelo de la grúa.

5 Implementación en el entorno EJS

Para el modelado en tres dimensiones y simulación del control del modelo de la planta se ha utilizado el entorno Easy Java Simulations (EJS), [3]. Este software se ha escogido debido a los trabajos que se han venido desarrollando a partir de los cursos-taller "UNEDLabs: Una nueva red de Laboratorios Colaborativos, Virtuales y Remotos" [7], en los cuales se utiliza EJS para el diseño de prácticas para docencia. Existen otros entornos de simulación con otras características interesantes como puede ser: Matlab (simulink y virtual reality toolbox) o Ecosimpro, con capacidad de control en tiempo real. En este caso, el desarrollo del modelo 3D se ha realizado en EJS para la posible integración del proyecto en la red de laboratorios virtuales.

Como se comenta en apartados anteriores, en este software se ha implementado el modelo pasivo de la planta y se ha utilizado un control basado en controladores PID para controlar la posición del carro de la grúa y la longitud del cable del peso colgado del carro, así como el control opcional de la oscilación del péndulo.

El diseño de la interfaz diseñado en EJS para esta planta consta de dos elementos. El primero es un panel llamado "Control 3DCRANE" (Figura 18) que sirve para modificar las variables y parámetros del sistema y, también, para visualizar la respuesta del mismo. Las variables a modificar son las referencias de posición dadas al controlador y los parámetros correspondientes a los valores de las constantes proporcional, derivativa e integral de los controladores. El control de los ángulos θ y ϕ es opcional, por ello se diseñó un selector con el que poder conmutar su activación y desactivación.

Este panel también es utilizado para la visualización de la respuesta temporal de la planta, mediante las gráficas del panel inferior, para los desplazamientos en los tres ejes (x,y,z) y los valores de los ángulos θ y ϕ frente al tiempo. La última gráfica representa el movimiento de la carga colgada de la grúa en en plano x-y, que facilita la visualización del movimiento oscilatorio del péndulo.

El otro panel diseñado para esta aplicación, llamado "3DCRANE" (figura 19), se corresponde con una visualización en tres dimensiones de la planta. Los elementos que constituyen el cuerpo en 3D de la grúa han sido realizados completamente con los elementos de dibujo 3D de los que dispone EJS. Esta visualización permite la total movilidad de la posición de la cámara.

6 Conclusiones y líneas futuras

En este artículo se muestra la identificación del modelo de una grúa tipo pórtico utilizando la formulación hamiltoniana. Las ecuaciones de

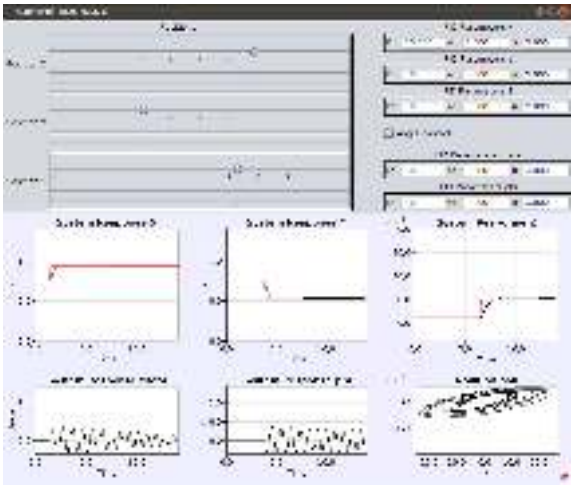


Figura 18: Panel de los parámetros de control y visualización de la respuesta de la planta.

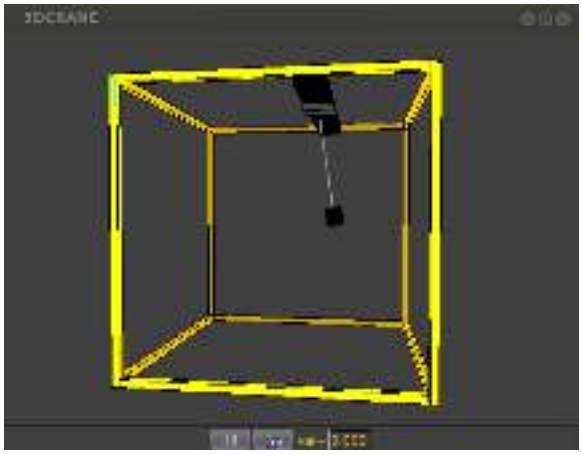


Figura 19: Visualización virtual de la grúa en 3D.

energía obtenidas son modeladas para obtener un modelo basado en pasividad. Se ha comprobado la pasividad por medio del análisis individual de los elementos que componen el sistema.

El modelo ha sido implementado el software de simulación EJS y se ha implementado una visualización en 3D a semejanza de la grúa real. También se han diseñado el control de posición del objeto colgado de la grúa.

Como trabajos futuros se propone la implementación del módulo de comunicaciones para el uso de la planta real del laboratorio remotamente. Esto implica el intercambio entre varios computadores conectados en red con transmisión/recepción de datos y video.

Dentro del marco de pasividad se pueden estudiar la pasividad de las comunicaciones entre los equipos remoto y de laboratorio y aplicar técnicas de pasivación de canales de comunicación, como puede ser el scattering.

Referencias

[1] J. Carrasco, A. Baños and A. van der Schaft, (2009) *A passivity-based approach to reset control systems stability.*

[2] E. Delgado, M. Diaz-Cacho, D. Bustelo, A. Barreiro, (2012) *Generic Approach to Stability Under Time-Varying Delay in Teleoperation: Application to the Position-Error Control of a Gantry Crane.*

[3] F. Esquembre et al. (2009) *Easy Java Simulations*

[4] A. Fernández-Villaverde, A. Barreiro, J. Carrasco, A. baños, (2011) *Reset Control for Passive Bilateral Teleoperation.* IEEE Transaction on industrial electronics, vol. 58.

[5] R. Ortega, A. van der Schaft , B. Maschkec , G. Escobar, (2002) *Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems*, Automatica 38, pp.585-596.

[6] J. E. Slotine and W. Li, (1991) *Applied nonlinear control*, Prentice-Hall, pp. 137-139.

[7] UNEDLabs: Una nueva red de Laboratorios Colaborativos, Virtuales y Remotos. Web: http://www.ceautomatica.es/og/educacion-en-automatica/curso-taller-cea-unedlabs

Intercambio de Modelos entre Herramientas de Simulación con Functional Mock-up Interface (FMI)

Javier Bonilla[1]
[1]CIEMAT - Plataforma Solar de Almería, ctra. de Senés km. 4,5 Tabernas (04200) Almería
javier.bonilla@psa.es

José Domingo Álvarez[2], Lidia Roca[3], Alberto de la Calle[3], Luis José Yebra[3], Francisco Rodríguez[4]
[2] Universidad de Sevilla, camino de los Descubrimientos s/n, (41092) Sevilla, jalvarez19@us.es
[3] CIEMAT - Plataforma Solar de Almería, {lidia.roca, alberto.calle, luis.yebra}@psa.es
[4] Universidad de Almería, ctra. Sacramento s/n, (04120) Almería, frrodrig@ual.es

Resumen

Debido a la heterogeneidad de las herramientas de modelado, simulación y control, un aspecto importante es el intercambio de modelos entre distintas herramientas de una forma transparente. Actualmente no existe ningún mecanismo, independiente de la herramienta, para el intercambio de modelos, aparte de Functional Mock-up Interface (FMI). En este artículo, modelos implementados en distintas herramientas de modelado y simulación (Dymola y Simulink) son intercambiados entre dichas herramientas usando FMI. Los resultados y tiempos de ejecución obtenidos son evaluados y comparados para determinar el rendimiento y precisión de las simulaciones.

Palabras clave: FMI, modelo, intercambio, simulación, herramientas de modelado.

1 Introducción

Cuando se consideran proyectos coordinados entre distintos grupos de trabajo, donde cada grupo utiliza sus propias herramientas especializadas, un eficiente intercambio de información y conocimiento, encapsulado en modelos, puede ser un reto.

Uno de los objetivos del uso de lenguajes de modelado no propietarios e independientes de herramientas (p. ej. *Modelica* [9]) es facilitar el intercambio de modelos [1]. Sin embargo, cuando las herramientas utilizan distintos lenguajes, algún mecanismo, preferiblemente automático, para el intercambio debe de ser considerado. Algunas herramientas proporcionan mecanismos propietarios para el intercambio de modelos (p. ej. *S-Function* en *Simulink* [7] o *DymolaBlock* en *Dymola* [5]). No obstante, estos mecanismos no pueden ser adoptados por otras herramientas que no sean las que considere el propietario de ellos.

Actualmente no existe ningún mecanismo para el intercambio de modelos que sea independiente de la herramienta, aparte de *Functional Mock-up Interface* (FMI).

2 Descripción de FMI

FMI es un estándar independiente de herramientas de modelado y simulación para el intercambio y co-simulación de modelos dinámicos.

El desarrollo de FMI fue iniciado en el proyecto *Information Technology for European Advancement* (ITEA-2) [8] *Modelisar* [12], cuyo principal objetivo era el de proporcionar mecanismos para el intercambio de modelos de simulación entre fabricantes y proveedores, independientemente de las herramientas que fuesen utilizadas. FMI ha sido desarrollado en estrecha colaboración con fabricantes de herramientas y centros de investigación.

FMI puede ser utilizado con los siguientes propósitos.

- *Intercambio de modelos* [11]. El objetivo es que una herramienta de modelado pueda generar código de un modelo dinámico que pueda ser utilizado por otra herramienta. Distintas instancias del modelo pueden ser utilizadas concurrentemente y estas, a su vez, pueden estar conectadas de forma jerárquica con otros modelos. El modelo es distribuido en un archivo comprimido en formato *zip*, llamado *Functional Mock-up Unit* (FMU).

- *Co-simulación* [10]. El objetivo es proporcionar una interfaz estándar parar acoplar dos o más herramientas de modelado y simulación. La información entre herramientas es intercambiada de forma síncrona. Los modelos son simulados independientemente entre intervalos de comunicación. Un proceso maestro controla el intercambio de datos y la sincronización entre herramientas que actúan como esclavos.

2.1 Distribución de Modelos

El componente FMU es un archivo en formato *zip* con extensión *fmu* que contiene todos los componentes necesarios para la utilización del modelo.

1. Un archivo en formato *eXtensible Markup Language* (XML) contiene la definición de las variables. Para co-simulación, la información referente a los procesos esclavos relevante para la sincronización es proporcionada en un archivo XML específico.

2. En el caso de intercambio de modelos, todas las ecuaciones son proporcionadas en un conjunto de funciones en código C. Estas funciones pueden estar en código fuente o binario. El mismo FMU puede incluir código binario para distintas plataformas. Adicionalmente, en el caso de co-simulación se incluye un conjunto de funciones en C (código fuente o binario) para la inicialización de la comunicación, el cálculo del intervalo de comunicación e intercambio de datos.

3. Otros datos que pueden ser añadidos al componente FMU son: icono del modelo, archivos de documentación, datos, tablas y librerías necesarias para el modelo.

2.2 Características

La principal característica de FMI es la de ser un mecanismo independiente de la herramienta para el intercambio de modelos y co-simulación. Haciendo uso del *Software Development Kit* (SDK), cualquier herramienta de modelado y simulación puede soportar FMI. Las principales características de FMI comparadas con respecto a una solución propietaria, *S-Function* en *Simulink*, son mostradas en la tabla 1. Para más información sobre las características de FMI, consulte [14].

Tabla 1: Comparativa entre *S-Function* (*Simulink*) y FMI [14] (Ecuación Diferencial Ordinaria (EDO)) .

Característica	***S-Function***	**FMI**
Ecuaciones	EDO	EDO
Eventos	Sí	Sí
Paso variable	Sí	Sí
Lenguaje	C, C++, Fortan	C
Multiplataforma	Sí	Sí
Recompilación	Sí	No
Archivo compacto	No	Sí
Propietario	Sí	No

2.3 Soporte en Herramientas de Modelado y Simulación

No todas las herramientas de modelado y simulación que existen en el mercado soportan todas y cada una de las características de FMI respecto al intercambio de modelos. Ni si quiera entre las que si lo hacen se permiten ambas opciones: la importación y exportación de modelos. La adopción de FMI está todavía llevándose a cabo, por lo que algunas herramientas tienen planificado incluir nuevas funcionalidades en futuras versiones de sus productos. La tabla 2 muestra algunas herramientas de modelado y simulación que soportan FMI y qué funcionalidades consideran para el caso de intercambio de modelos. Para más información sobre el soporte de FMI en herramientas de modelado y simulación, consulte [13].

Tabla 2: Soporte de FMI para intercambio de modelos en algunas herramientas de modelado y simulación (P = planificado incluir en próximas versiones) [13].

Herramienta	**Intercambio**	
	Importar	Exportar
Adams	No	P
AMESim	Sí	Sí
ANSYS	No	P
CATIA	Sí	Sí
Dymola	Sí	Sí
EcosimPro	No	No
Matlab y Simulink	Sí	Sí
JModelica.org	Sí	Sí
LMS Virtual.Lab	P	Sí
MapleSim	Sí	P
MWorks	Sí	P
NI LabVIEW	No	P
OpenModelica	Sí	Sí
OPTIMICA Studio	Sí	P
PyFMI	No	Sí
Silver	Sí	Sí
SIMPACK	P	Sí
SimulationX	Sí	Sí
SystemModeler	P	P
TLK FMI Suite	No	Sí
TLK TISC Suite	No	Sí

2.4 Limitaciones

Actualmente existen una serie de limitaciones en las herramientas de modelado y simulación con respecto a FMI. Estas se resumen a continuación.

- Las herramientas de modelado y simulación más relevantes incorporan o tienen planificado incorporar FMI en próximas versiones de sus productos. Por lo tanto, no toda la funcionalidad de FMI esta soportada por todas

Tabla 3: Herramientas consideradas en el intercambio de modelos mediante FMI.

Sistema Operativo		Microsoft Windows XP - 32 bits
Dymola [5]	**Versión**	2013
	Compilador	Visual C++ 2010 Express Edition (10.0)
	Importar FMU	Soportado
	Exportar FMU	Soportado
Matlab [17]	**Versión**	R2010a (7.10.0.499)
	Compilador	Visual C++ 2008 Express Edition (9.0)
	Importar FMU	FMI Modelon toolbox 1.4.1 [15]
	Exportar FMU	Dymola 2013 toolbox

las herramientas. Esta circunstancia limita la adopción de FMI por parte de los usuarios finales. Además, en algunos casos para utilizar FMI es necesario paquetes de software comerciales, ya que éste no es soportado directamente por la herramienta, lo que supone una inversión económica.

- FMI no se encuentra igualmente soportado en todas las plataformas por las herramientas de modelado y simulación, ya que algunas herramientas solo soportan FMI en plataformas Windows.
- Otra serie de limitaciones vienen dadas por las propias herramientas, por ejemplo en lo que respecta a modelos que pueden ser utilizados en componentes FMU, o al número de instancias de un mismo componentes FMU por modelo, que en el caso particular de la herramienta *Dymola* es de una instancia.

3 Intercambio de Modelos

En esta sección, dos modelos implementados en distintas herramientas de modelado y simulación son intercambiados, los resultados obtenidos junto con las estadísticas de simulación son comparados y analizados.

Las herramientas y versiones son especificadas en la sección 3.1. Los modelos considerados en esta comparativa son descritos en la sección 3.2. La configuración y parámetros de simulación son enumerados en la sección 3.3. Finalmente, los resultados son mostrados en la sección 3.4.

3.1 Herramientas Consideradas

Las herramientas de simulación consideras en esta comparativa son *Dymola* y *Matlab/Simulink*. La tabla 3 muestra el sistema operativo donde se han llevado a cabo las simulaciones, las versiones de las herramientas y los compiladores utilizados.

La exportación e importación de componentes FMU en *Dymola* está soportada directamente por la herramienta. Sin embargo en *Matlab/Simulink* es necesario software adicional. *Dymola* incluye un paquete para exportar modelos de *Simulink* a componentes FMU. Para la importación de componentes FMU existe la posibilidad de utilizar software de terceros, como por ejemplo el *FMI Modelon ToolBox* [15].

3.2 Modelos Considerados

Los dos modelos considerados en esta comparativa son descritos a continuación, cada uno de ellos ha sido implementado en una de las dos herramientas de modelado y simulación seleccionadas.

3.2.1 Modelo en Dymola

El modelo implementado en *Dymola* con el lenguaje de modelado *Modelica* se corresponde con la planta experimental *DIrect Solar Steam* (DISS) [18]. La planta DISS es una planta solar térmica de colectores cilindro-parabólicos de generación directa de vapor, donde el fluido de trabajo es agua en estado bifásico líquido-vapor. Esta planta pertenece al Centro de Investigaciones Medioambientales, Energéticas y Tecnológicas (CIEMAT) y está ubicada en su centro de la Plataforma Solar de Almería (PSA). El diagrama de componentes del modelo de la planta DISS se muestra en la figura 1 [2].

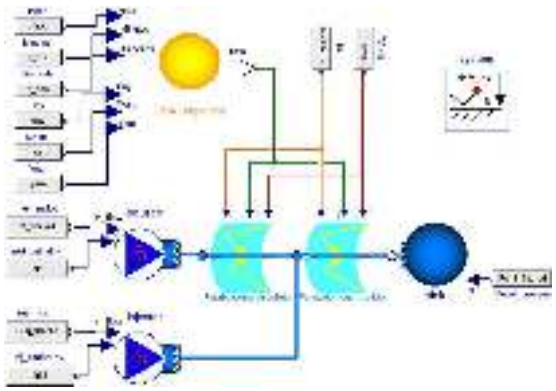


Figura 1: Diagrama de componentes del modelo de la planta DISS en Dymola

Dymola permite que el modelo de la planta DISS pueda ser exportado a un componente FMU. Dicho componente puede ser importado a *Simulink* usando el software *FMI Modelon ToolBox*. El componente FMU, de la planta DISS, importado en *Simulink* se muestra en la figura 2.

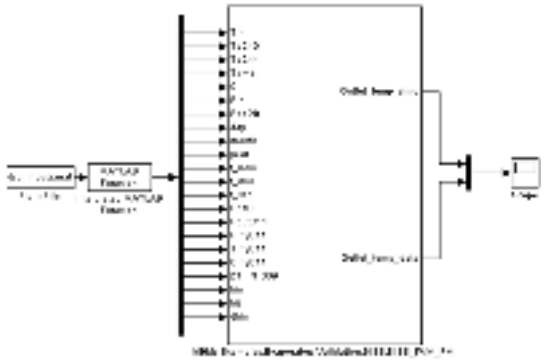


Figura 2: Componente FMU del modelo de la planta DISS en Simulink

3.2.2 Modelo en Simulink

La figura 3 muestra el diagrama de bloques en *Simulink* de un modelo formado por un controlador PID y un modelo lineal que relaciona un índice de confort térmico, el índice PMV (*Predicted Mean Vote*) [6], con la velocidad de una unidad *fan coil*. Para obtener más información sobre el modelo, consulte [3].

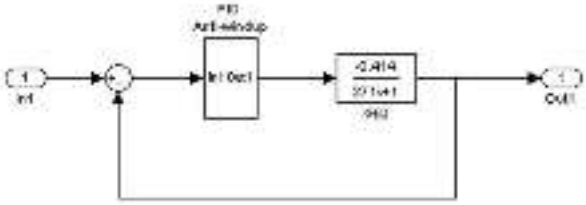


Figura 3: Diagrama de bloques del modelo del sistema de control en Simulink

Dymola incluye un *toolbox* para *Simulink* que permite la exportación de modelos a componentes FMU. Este *toolbox* para la exportación, junto con el soporte de *Dymola* para la importación de componentes FMU permite la simulación de modelos de *Simulink* en *Dymola*.

La figura 4 muestra el componente FMU del modelo *Simulink* del sistema de control importado en *Dymola*.

3.3 Configuración de las Simulaciones

En este apartado se indican los integradores numéricos junto con las tolerancias relativas y



Figura 4: Componente FMU del modelo del sistema de control en Dymola

absolutas para cada modelo y herramienta. La tabla 4 muestra el integrador numérico y tolerancia (la tolerancia absoluta y relativa tienen el mismo valor) para el modelo de la planta DISS (sección 3.2.1) en función de la herramienta de modelado y simulación.

Tabla 4: Integrador numérico y tolerancias para el modelo de la planta DISS en función de la herramienta de modelado

Herramienta	**Integrador**	**Tol.**
Dymola	DASSL [16]	10^{-4}
Simulink (FMU)	ODE23TB [17]	10^{-4}

La misma información se muestra en la tabla 5, en este caso para el modelo del sistema de control (sección 3.2.2).

Tabla 5: Integrador numérico y tolerancias para el modelo del sistema de control en función de la herramienta de modelado

Herramienta	**Integrador**	**Tol.**
Simulink	ODE23S [17]	10^{-4}
Dymola (FMU)	DASSL [16]	10^{-4}

3.4 Resultados y Estadísticas de Simulación

Ambos modelos, el modelo de la planta DISS y el modelo del sistema de control, se simularon en ambas herramientas, *Dymola* y *Simulink*. Los resultados y tiempos de simulación obtenidos han sido evaluados. Sin embargo, es asumible que pueda haber pequeñas discrepancias tanto en tiempos de simulación como en resultados, debido a que los integradores numéricos que incorporan ambas herramientas son diferentes. casos deben de ser las mismas.

3.4.1 Modelo de la planta DISS

Como entradas a este modelo se han considerado datos experimentales obtenidos durante los ensayos llevados a cabo en el proyecto DISS [19]. Los días de operación fueron seleccionados para

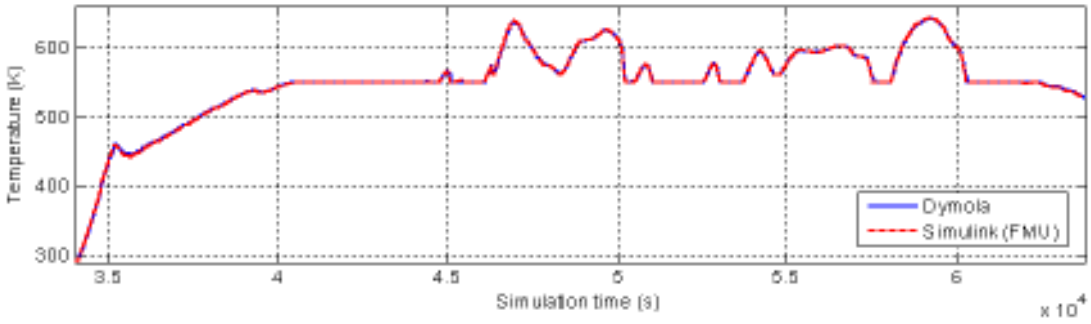


Figura 5: Temperatura de salida en el modelo de la planta DISS en Dymola y Simulink (FMU)

exponer al modelo a distintas condiciones de operación y perturbaciones (nubes, ya que la irradiancia solar directa es una entrada del modelo).

La figura 5 muestra los resultados obtenidos en ambas herramientas, en concreto la temperatura de salida del último colector cilindro-parabólico durante un día de operación. La máxima discrepancia entre ambos modelos en temperatura de salida es de 8.20 K (<1,5%). Mientras que el error medio es de 1.20 K (<0,25%).

La tabla 6 muestra los tiempos de simulación de ambos modelos. Las diferencias son significativas debido a la utilización de integradores numéricos diferentes, como se indica en la tabla 4.

El modelo de la planta DISS es un modelo complejo, se debe de resolver el problema de inicialización y debe de llevarse a cabo la gestión de: eventos de estado, control del tamaño del paso y el orden de integración. Estas tareas son realizadas por un parte del código de los integradores numéricos que se conoce como *código de producción* y que puede representar hasta el 95% del código total del integrador numérico [4].

DASSL incorpora un *código de producción* robusto y depurado [4], lo que parece estar relacionado con el hecho de que el tiempo de simulación sea menor en *DASSL* que en otros integradores numéricos, y no estando directamente relacionada esta diferencia de tiempo con el tipo de herramienta de simulación o con que el modelo sea un componente FMU.

Tabla 6: Tiempos de simulación de el modelo de la planta DISS en Dymola y Simulink

Herramienta	Tiempo de simulación
Dymola	1 minuto y 45 segundos
Simulink (FMU)	2 minutos y 58 segundos

3.4.2 Modelo del sistema de control

Este modelo recibe como entrada la señal que se muestra en la figura 6. La figura 7 muestra la salida de el modelo en *Simulink* y *Dymola*. El error máximo entre ellos es $2 \cdot 10^{-3}$ (<0,5%), mientras que el error medio es 10^{-3} (<0,25%).

La tabla 7 muestra los tiempos de simulación de ambos modelos. El modelo es muy rápido y las diferencias de tiempo no son significativas, aunque en los tiempos medios de simulación calculados (tabla 7) y en cada una de las simulaciones llevadas a cabo, el modelo en Dymola (FMU) con el integrador numérico DASSL ha sido el más rápido.

Tabla 7: Tiempos de simulación de el modelo del sistema de control en Simulink y Dymola

Herramienta	Tiempo de simulación
Simulink	$4,7 \cdot 10^{-2}$ segundos
Dymola (FMU)	$2,3 \cdot 10^{-2}$ segundos

4 Conclusiones

FMI es el único mecanismo estándar e independiente de plataforma, lenguaje y herramienta de modelado para el intercambio de modelos y co-simulación.

La especificación de FMI 1.0 se encuentra actualmente disponible. Dicha especificación ha sido elaborada en estrecha colaboración con la industria y la academia en el proyecto *Modelisar*. La especificación 2.0 se encuentra en fase beta. Actualmente, es posible el intercambio de modelos mediante herramientas que implementan el lenguaje de modelado Modelica y otras herramientas como *Matlab/Simulink*. La adopción de este estándar por parte de las herramientas de modelado sigue incrementándose día a día, añadiendo nuevas funcionalidades en las versiones más recientes, aunque todavía existan una serie de limitaciones.

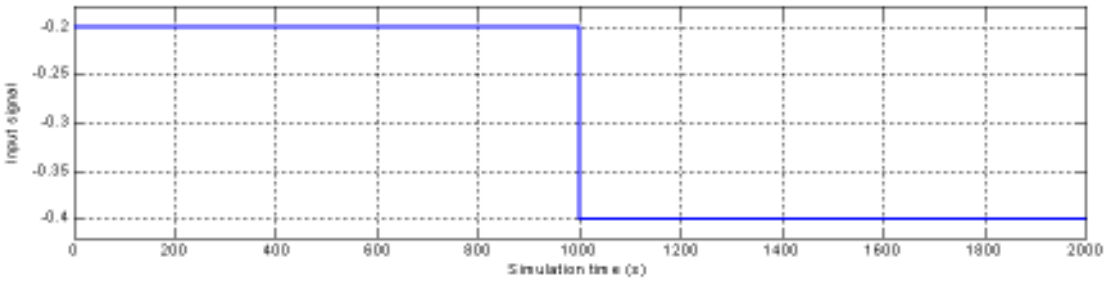


Figura 6: Entrada a el modelo del sistema de control

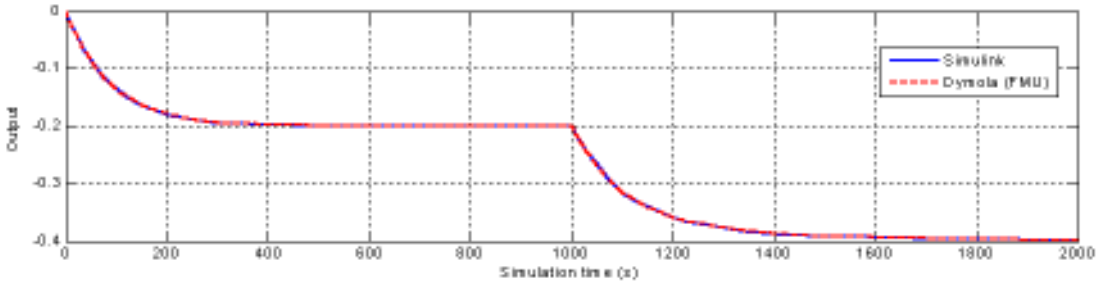


Figura 7: Comparativa de la salida en el modelo del sistema de control en Simulink y Dymola (FMU)

Los resultados presentados en este artículo muestran que FMI puede ser actualmente utilizado para el intercambio de modelos entre distintas herramientas. No se incluye el integrador numérico en el componente FMU en los casos analizados, lo que da lugar a discrepancias en resultados y tiempos de simulación entre herramientas. No obstante, el estándar especifica que el integrador numérico puede ser incluido o no en el componente FMU. Esta característica puede por lo tanto estar disponible en futuras implementaciones de FMI por parte de los desarrolladores de herramientas de modelado y simulación.

Agradecimientos

Este trabajo ha sido financiado con los proyectos: Proyecto POWER del Plan Nacional DPI2010-21589-C05-04/05 del Ministerio de Economía y Competitividad y Fondos FEDER. Proyecto HYSOL del Séptimo Programa Marco de la Unión Europea (FP7/2013-2016) según el acuerdo de subvención número 308912. Y el Proyecto de excelencia CONTROLCROP de la Consejería de Economía, Innovación, Ciencia y Empleo de la Junta de Andalucía PIO-TEP-6174.

Referencias

[1] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J-V. Peetz y S. Wolf. The functional mockup interface for tool independent exchange of simulation models. En *Proc. 8th International Modelica Conference*. Dresden, Alemania, 2010.

[2] J. Bonilla, L.J. Yebra, S. Dormido y E. Zarza. Parabolic-trough solar thermal power plant simulation scheme, multi-objective genetic algorithm calibration and validation. *Solar Energy*, 86(1):531–540, 2012. ISSN 0038092X. doi:10.1016/j.solener.2011.10.025.

[3] M. Castilla, J.D. Álvarez, M. Berenguel, F. Rodríguez, J.L. Guzmán y M. Pérez. A comparison of thermal comfort predictive control strategies. *Energy and Buildings*, 43(10):2737 – 2746, 2011. ISSN 0378-7788. doi:10.1016/j.enbuild.2011.06.030.

[4] F. E. Cellier y E. Kofman. *Continuous system simulation*. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006. ISBN 978-0-387-26102-7.

[5] Dassault Systemes. Dymola 2013. `http://www.3ds.com/products/catia/portfolio/dymola`, 2012.

[6] P. O. Fanger. Assessment of man's thermal comfort in practice. *British Journal of Industrial Medicine*, 30(4):313–324, 1973. doi: 10.1136/oem.30.4.313.

[7] A. C. W. Grace. SIMULAB, an integrated environment for simulation and control. En

Proc. 1991 American Control Conference, págs. 1015–1020. Boston, MA, USA, 1991.

[8] ITEA-2. ITEA-2, strategic pan-European programme for advanced pre-competitive R&D in Software-intensive Systems and Services (SiSS). http://www.itea2.org/, 2012.

[9] Modelica Association. Modelica standard library 3.2. https://www.modelica.org/libraries/Modelica, 2010.

[10] Modelisar. FMI for co-simulation specification. http://www.modelisar.com/specifications/FMI_for_CoSimulation_v1.0.pdf, 2010.

[11] Modelisar. FMI for model exchange specification. http://www.modelisar.com/specifications/FMI_for_ModelExchange_v1.0.pdf, 2010.

[12] Modelisar. AUTOSAR - Interoperability to support vehicle functional mock-up. http://www.modelisar.com/, 2011.

[13] Modelisar. FMI support in tools. http://www.modelisar.com/tools, 2012.

[14] Modelisar. Functional Mockup Interface (FMI). http://www.modelisar.com/fmi, 2012.

[15] Modelon AB. Modelica physical modeling toolbox for MATLAB. http://www.modelon.com/products/modelica-physical-modeling-toolbox-for-matlab, 2012.

[16] L.R. Petzold. A description of DASSL: a diferential/algebraic system solver. *Scientific Computing*, págs. 65–68, 1983. ISSN 0163-5964.

[17] The MathWorks Inc. MATLAB R2010a. http://www.mathworks.es/products/matlab/, 2010.

[18] E. Zarza. *Generación directa de vapor con colectores cilindro-parabólicos. El proyecto DISS*. Tesis Doctoral, Escuela Superior de Ingenieros Industriales de Sevilla, Sevilla, España, Noviembre 2000.

[19] E. Zarza, L. Valenzuela, J. León, D. Weyers, M. Eickhoff, M. Eck y K. Hennecke. The DISS project: direct steam generation in parabolic trough systems. Operation & maintenance experience and update on project status. *Journal of Solar Energy Engineering*, 124:126–133, Mayo 2002. ISSN 0199-6231.

Modelado del reactor químico del proceso BTO mediante redes neuronales artificiales

Gorka Sorrosal[a], Eloy Irigoyen[b], Cruz E. Borges[a], Ainhoa Alonso[a]
[a]DeustoTech Energy, Universidad de Deusto, Av. Universidades 24, Bilbao (Bizkaia), España, {gsorrosal, cruz.borges, ainhoa.alonso}@deusto.es
[b]Departamento de Ingeniería de Sistemas y Automática, UPV/EHU, Alda. Urquijo s/n E-48013 Bilbao (Bizkaia), España, eloy.irigoyen@ehu.es

Resumen

En el presente artículo se ha realizado el modelado global del reactor químico del proceso de transformación de Bioetanol a Olefina (BTO) mediante Redes Neuronales Artificiales (RNA). Para comprobar la bondad de las RNA, durante su estudio se ha realizado una comparativa con otras dos técnicas: Regresión Lineal (RL) y Support Vector Machines (SVM). Para el proceso de modelado se ha empleado un conjunto de datos obtenidos de un reactor químico a escala de laboratorio, y se ha llegado a la conclusión de que la mejor técnica de modelado para el presente caso es la basada en RNA. El objetivo principal ha sido la obtención de un modelo neuronal del reactor químico del proceso BTO con la precisión deseada que permita simplificar y reducir los tiempos de estimación con respecto a un modelo de conocimiento.

Palabras clave: Modelado, Proceso BTO, RNA.

1. INTRODUCCIÓN

Tanto en la industria química, como petroquímica, es muy habitual encontrarse en sus procesos con reactores catalíticos de lecho fijo. Según el proceso, estos reactores se cargan con un catalizador muy activo y selectivo. Sin embargo, con el tiempo el catalizador se desactiva por envenenamiento u otras causas, de forma lineal o exponencial en función del mecanismo de desactivación. El presente trabajo está focalizado en un proceso BTO y el estudio de su comportamiento para modelar su dinámica mediante RNA. El proceso BTO consiste en la transformación catalítica del bioetanol en olefinas, siendo éste un proceso clave en el concepto de la refinería sostenible, incorporando como alimentación la biomasa o derivados.

El conjunto de experimentos realizados para obtener los datos necesarios para el modelado de estas transformaciones fueron llevadas a cabo por A. Alonso en [2] mediante el equipo de reacción de la figura 1. Este consiste en un equipo automatizado de reacción provisto de un reactor isotermo de lecho fijo en línea con un cromatógrafo de gases.

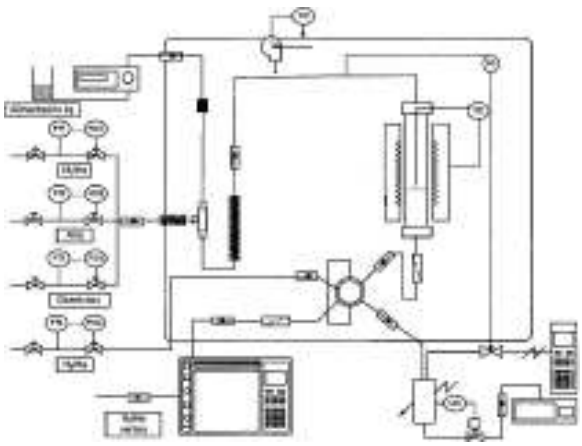


Figura 1: Esquema del equipo de reacción empleado para la obtención de los datos experimentales

El reactor es de acero inoxidable 316, con 9 mm de diámetro interno y 10 cm de longitud total efectiva; situado en el interior de una cámara cilíndrica de acero inoxidable calentado mediante una resistencia eléctrica cubierta de cerámica. Esta cámara se encuentra a su vez en el interior de otra, en la que la temperatura es suficiente para vaporizar la alimentación líquida y evitar la condensación de los productos de reacción entre la salida del reactor y el cromatógrafo de gases.

Para su modelado se ha trabajado con la principales variables de entrada y salida que afectan al reactor del proceso BTO [2]. Aunque a la salida del proceso se encontrarán presentes más compuestos, en nuestro caso, la única salida del proceso que se ha estudiado es la composición de olefinas. En este caso, las variables quedan definidas del siguiente modo:

- **Variables de control**:
  - X_w: fracción másica de agua referida a la masa de etileno equivalente en la alimentación del reactor ($g_{agua}g^{-1}$).
  - T: temperatura del reactor (K).
  - τ: tiempo espacial

$$g_{catalizador}h(g_{etanol})^{-1}.$$

- **Perturbaciones**:
  - a': actividad aparente del catalizador. La actividad (a) del reactor mide la desactivación del catalizador por acumulación de coque. Sin embargo, dado que nos es imposible obtener dicha información, se usará la actividad aparente (a') como indicador. Definiendola como la relación entre la concentración actual y la concentración inicial del producto a la salida del reactor.
- **Variable de salida o controlada**:
  - X_o: fracción másica del lump de olefinas referida a la masa de etileno equivalente en la alimentación ($g_{olefinas}g^{-1}$).

Uno de los puntos importantes y problemáticos para llevar a cabo el control de estos procesos es mantener la calidad del producto a la salida del reactor del proceso, bien aumentando la temperatura o ajustando otras variables de operación para contrarrestar la desactivación del catalizador. Sin embargo, aumentar la temperatura puede provocar la existencia de puntos calientes y la consecuente destrucción del catalizador llegando a situaciones peligrosas incluso para el propio equipo de reacción. Por ello, se ha trabajado siempre en los rangos de operación seguros, para evitar situaciones peligrosas y/o daños irreversibles.

Dado que los puntos de operación óptimos para contrarrestar la desactivación del catalizador en una planta comercial son limitados, debido a su dependencia de múltiples variables, ha sido necesario desarrollar una estrategia de optimización de las condiciones de operación que garantice unos objetivos de producción específicos para el cliente.

Las relaciones entre las diferentes variables de control y la desactivación del catalizador pueden dificultar el desarrollo de las estrategias de control. Por tanto, para la implementación de los controladores y para el desarrollo de nuevas estrategias, tanto de control clásico como de control inteligente, es muy importante disponer de modelos del proceso que nos permitan realizar simulaciones del mismo o su uso dentro del propio controlador. No obstante, los modelos matemáticos del proceso son muy complejos en su resolución por lo que generan tiempos de respuesta muy elevados para ser utilizados en planteamientos basados en técnicas de control inteligentes. Por ello, se hace necesario buscar una nueva estrategia para obtener un modelado del reactor químico de un proceso BTO, la cual nos permita reducir los tiempos de diseño de la estrategia de control y/o del propio control [3]. Una posibilidad de trabajo es el empleo de RNA, las cuales ofrecen buenos resultados a la hora de modelar procesos complejos [4].

Las RNA están siendo usadas en multitud de aplicaciones con muy buenos resultados, debido a su capacidad de apredizaje de comportamientos altamente no-lineales y dependientes del tiempo [1]. Además de su capacidad de aprendizaje, también hay que tener en cuenta su velocidad en la estimación de valores una vez entrenadas, con unos tiempos de ejecución generalmente mucho más reducidos que los modelos matemáticos tradicionales de los reactores químicos. Una de la ventajas que presentan las RNA es que simplifican el modelado de sistemas, reduciendo los tiempos de desarrollo y presentando también un mejor rendimiento que las técnicas de control convencionales, tal y como expone Norgaard en [11].

En este trabajo, con objeto de validar los modelos neuronales obtenidos en los correspondientes procesos de identificación, se ha utilizado como criterio de validación la comparación con un error experimental previamente establecido. Los modelos se han comprobado midiendo el error máximo cometido, solo considerándose válidos aquellos cuyo error máximo sea inferior al error experimental. Este error ha sido obtenido a partir del conocimiento experto del origen de los datos experimentales empleados en el presente trabajo [2]. Dicho error se compone de los errores acumulados de medida, experimentación, equipo de análisis y del propio catalizador.

El resto del artículo está estructurado de la siguiente manera. En la sección 2, hay un breve resumen sobre el modelado neuronal de reactores químicos. En la siguiente sección se expone el origen y la problemática del conjunto de datos experimentales empleado. En la cuarta sección se detalla la metodología seguida para el correcto modelado del proceso, mientras que en la quinta sección se muestran los resultados de dicho proceso. Y por último, se exponen las conclusiones y las líneas de trabajo futuras.

2. ALTERNATIVAS DE MODELADO DE REACTORES QUÍMICOS

Analizando la literatura se observa que los modelos más usados en el modelado de reactores químico mediante redes neuronales son de tipo híbrido «redes neuronales-modelos de conocimiento».Estos modelos permiten una posterior escalabilidad del reactor. No obstante, se han encontrado casos en la bibliografía en los que se ha abordado la problemática mediante redes neuronales globales con resultados satisfactorios [6], [10], [8], [12], [9].

En la mayoría de los casos analizados se han utilizado RNA con una estructura de 3 capas: la capa de entrada, la capa de salida y una capa oculta intermedia. Todas ellas de tipo *feed-forward* [6], [10], [8]. En cuanto a la configuración de los vectores de entrada-salida, en algunos casos se han implementado RNA con una topología simple, alimentándose únicamente con las señales de control del reactor. Estas RNA son denominadas *Finite Impulse Response* (NNFIR) y se caracterizan por tener unas predicciones muy estables [10], [9]. El modelo estructural de entradas-salidas se corresponde al mostrado en la ecuación 1.

$$\begin{aligned} Input &= (u(t-d), \ldots, u(t-d-n)) \\ Target &= y(t). \end{aligned} \tag{1}$$

No obstante, en la mayoría de los casos presentes en la literatura se han utilizado con estructuras *Nonlinear AutoRegressive with eXogeneous inputs* (NARX), en las que se alimentan no sólo con las entradas anteriores, sino también con los resultados de salida de las etapas anteriores [6], [10], [12], [9]. Esto permite a la RNA reproducir el comportamiento dinámico del sistema introduciendo una memoria de las salidas anteriores del reactor. El modelo estructural de entradas-salidas se corresponde al mostrado en la ecuación 2.

$$\begin{aligned} Input = (&u(t-d), \ldots, u(t-d-n), \\ &y(t-1), \ldots, y(t-n)) \\ Target &= y(t). \end{aligned} \tag{2}$$

Por último, se han propuesto las RNA con estructuras *AutoRegressive Moving Average with eXogeneous inputs* (ARMAX) [8], [9]. En éstas además de recibir al igual que las NARX la información sobre entradas y salidas anteriores, también obtienen información sobre el error que genera en sus estimaciones. Esto permite mejorar la estimación de la red corrigiendo el error generado de forma online. El modelo estructural de entradas-salidas se corresponde al mostrado en la ecuación 3.

$$\begin{aligned} Input = (&u(t-d), \ldots, u(t-d-n), \\ &y(t-1), \ldots, y(t-n), \\ &e(t-1), \ldots, e(t-k)) \\ Target &= y(t). \end{aligned} \tag{3}$$

Otra opción similar de modelado es modificar posteriormente dicha estimación en función del error detectado en la validación. En esta última opción se pretende corregir el error de forma *offline*. Esto es, una vez que ya se tiene disponible la estimación del modelo, a ésta se le agrega una función de error para corregir la desviación de dicha estimación, tal y como se puede ver en la ecuación 4.

$$\hat{y}(k) = \hat{y}_{RNA}(k) + F(error). \tag{4}$$

3. DATOS EXPERIMENTALES

Para el entrenamiento de la red neuronal se han empleado los datos experimentales obtenidos del reactor químico a escala de laboratorio del proceso BTO explicado en la introducción (ver figura 1). Estos datos, cuyo resumen se muestra en la tabla 1, tienen su origen en [2] y han sido obtenidos mediante un conjunto de pruebas realizadas en dicho reactor.

Tabla 1: Resumen de los datos experimentales.

	Experimentos	**Muestras**
T. constante	17	258
T. variable	9	141
Total	26	399

Los datos de estos experimentos no tenían un muestreo periódico, ni entre muestras de un mismo experimento, ni entre diferentes experimentos. Por ello, como paso previo, se ha realizado un preprocesamiento de dichos datos para que en todos los experimentos tuviesen el mismo periodo de muestreo. De esta forma además de conseguir una frecuencia de muestreo periódica, se soluciona otro problema muy habitual en estos sistemas, la escasez de ejemplos reales para realizar un estudio pormenorizado del comportamiento del proceso. En ese sentido, Molga [10] recomienda el uso de 100 muestras como mínimo por experimento. Al no disponer en dicho trabajo de tal cantidad de muestras experimentales, «multiplicaron» las muestras disponibles mediante interpolación polinomial logrando resultados satisfactorios. Sin embargo, este último punto es muy discutible, y no puede ser aplicado sistemáticamente. Esta recomendación podrá ser útil en el caso de que el sistema tenga un comportamiento sin cambios bruscos en sus señales de entrada y salida. Y tal es el comportamiento del caso de aplicación propuesto.

En nuestro caso, aunque en el trabajo citado se hace uso de una interpolación polinomial, se ha utilizado la interpolación lineal por su simplicidad en las primeras fases. Las pruebas iniciales han sido realizadas tanto con los datos interpolados como con los datos originales sin interpolar de varios experimentos, a fin de determinar la necesidad real de interpolación de todos los datos reales experimentales.



Figura 2: Topología de entrada/salida de la RNA 1



Figura 3: Topología de entrada/salida de la RNA 2

4. MODELADO GLOBAL DEL PROCESO BTO

En este capítulo se van a analizar los resultados del proceso de modelado que se ha llevado a cabo en el presente trabajo. Desde los modelos neuronales iniciales hasta el modelo neuronal final, explicando los pasos dados en cada fase.

Siguiendo las recomendaciones de los trabajos consultados en la sección 2 en cuanto a la estructura interna de las redes neuronales [6], [10], [8], el modelado del proceso BTO se llevó a cabo inicialmente con una configuración interna de tipo feed-forward. Mientras que para la configuración de los vectores de entrada-salida, se han propuesto dos tipos de RNA, tal y como se ven en las figuras 2 y 3.

A la hora de validar y comparar los modelos generados, el principal criterio de validación será el error máximo del modelo frente al error experimental ya explicado en la sección 1. No obstante, a lo largo del trabajo se han empleado los siguientes cálculos estadísticos para la validación:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - RNA(x_i)| \qquad (5)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - RNA(x_i))^2} \qquad (6)$$

$$MAX = \max_{i\epsilon[1,\ldots,n]}|f(x_i) - RNA(x_i)| \qquad (7)$$

$$HITS = \#\left\{i \in [1,\ldots,n] : RNA(x_i) \in [f(x_i) - \varepsilon, f(x_i) + \varepsilon]\right\}, \qquad (8)$$

siendo ε el error experimental.

Tal y como se ha comentado en la sección 3, las primeras pruebas han estado orientadas a determinar la necesidad de preprocesar el conjunto de datos original. Observando los resultados de la tabla 2, se llega a la conclusión de que el empleo de datos muestreados periódicamente mejoran ligeramente los resultados, pero tal vez no lo suficiente como para justificar el coste del preprocesamiento de todos los datos. Además, también se ha de tener en cuenta que el empleo de datos ficticios podría forzar a la RNA a seguir un modelo lineal irreal. Por todo ello, se descartó en este punto el preprocesamiento de los datos con este objetivo.

Tabla 2: Resultados de la interpolación lineal.

Errores	Interpolados	Originales
RMSE	0.0378	0.0396
MAE	0.0374	0.0390

Sin embargo, uno de los grandes inconvenientes del conjunto de datos empleados, ha sido tanto la pequeña cantidad de experimentos disponibles, como el número de muestras por experimento. De esta forma, el conjunto de datos útiles es limitado, habiéndose demostrado insuficiente para el correcto aprendizaje por parte de la red neuronal de todos los experimentos [7]. No obstante, en [7] se presentan varias soluciones ante la dificultad de aprendizaje con estos tipos de conjuntos de datos.

En los siguientes subapartados se presentan los pasos llevados a cabo hasta obtener un modelo con la precisión suficiente. En el subapartado 4.1 se presenta un modelo neuronal con estructura no recurrente (estándar), en el 4.2 un modelo neuronal con estructura recurrente y finalmente en el 4.3 el modelo neuronal por temperaturas que cumple con los criterios de validación establecidos.

4.1. MODELADO NEURONAL ESTÁNDAR DEL PROCESO BTO

Una vez finalizados los primeros estudios y decidida la estructura interna y la configuración de los vectores de entrada-salida, se han realizado varias pruebas con todos los datos experimentales disponibles para obtener el modelo neuronal del proceso BTO. Para ello se ha seguido un procedimiento iterativo hasta reducir al máximo posible los errores para la estimación de la salida.

El objetivo de esta fase ha sido el desarrollo de un modelo estático, comprobando la capacidad de la red neuronal de mapear los puntos de trabajo del proceso. Los mejores resultados han sido obtenidos con la red RNA2 mostrada en la figura 3. Para todos los experimentos se consiguió que en el 100 %

de las muestras se obtuviese un error de estimación inferior al error experimental.

Estas mismas pruebas han sido implementadas con otras dos metodologías diferentes, la RL y las SVM, para comparar sus resultados con los obtenidos por las redes neuronales.

Tal y como se puede observar en la tabla 3, los errores resultantes obtenidos con las tres metodologías propuestas son muy similares, siendo ligeramente mejores los obtenidos por las redes neuronales. No obstante, aunque los resultados finales sean similares, los tiempos de estimación requeridos en cada caso son muy diferentes. Siendo la RL la opción más rápida y las RNA casi 10 veces más lentas. Y dado que el objetivo de este trabajo es desarrollar un modelo del proceso BTO con un tiempo de estimación bajo, estos tiempos son determinantes a la hora de seleccionar el método.

Tabla 3: Comparativa del modelo estático entre varias metodología de modelado. Error medio de todos los experimentos.

		Errores medios
RL	MAE	0.0047
	RMSE	0.0058
SVM	MAE	0.0043
	RMSE	0.0056
RNA	MAE	0.0045
	RMSE	0.0050
	HITS	100 %

Las redes neuronales utilizadas han sido implementadas haciendo uso del Neural Networks Toolbox de MATLAB®. En cambio, para implementar las SVM y la RL se ha empleado tanto MATLAB® como Weka (*Waikato Environment for Knowledge Analysis*) [5].

4.2. MODELADO NEURONAL RECURRENTE DEL PROCESO BTO

A pesar de los buenos resultados obtenidos con el modelo estático, para modelar correctamente la dinámica del proceso se hace necesario un modelo de redes neuronales recurrentes. Para la validación de los nuevos modelos, ahora se emplean como entradas sus propias salidas estimadas y no las disponibles en el conjunto de datos reales. Además, dado que la actividad aparente del reactor depende de la concentración de olefinas a la salida, en esta fase se realiza una estimación lineal de dicha actividad con las estimaciones anteriores de concentraciones proporcionadas por la red neuronal (ecuación 9).

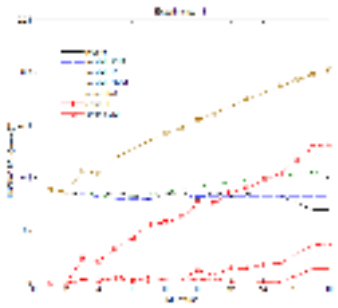


Figura 4: Validación de uno de los experimentos. Comparativa entre el modelo basado en RL, SVM y RNA

$$a'(k-1) = a'(k-2) + \frac{RNA(k-1) - RNA(k-2)}{Xo(0)\Delta t} \quad (9)$$

Con el nuevo tipo de validación para comprobar el correcto modelado de la dinámica del sistema, los resultados obtenidos varían con respecto a los obtenidos anteriormente. Los resultados empeoran independientemente de la estrategia de modelado empleada. Sin embargo, si se observa que las redes neuronales aun empeorando bastante, presentan una respuesta mejor que los modelos basados en RL o las SVM estándar.

Por ello, se decide continuar con las redes neuronales artificiales y se modifica la estructura interna de la red a una NARX, por ser ésta la estructura más idónea para aprender la dinámica de procesos(ecuación 2).

Con esta nueva modificación, tal y como se puede observar en la figura 4, los resultados mejoran lo suficiente como para descartar definitivamente la RL como técnica de modelado. Las estimaciones proporcionadas, tanto por la regresión lineal como por la SVM, divergen claramente de la curva original, mientras que las estimaciones del modelo neuronal la siguen más fielmente.

No obstante, a pesar de que los errores presentados en los modelos neuronales son claramente inferiores a los de la regresión lineal, los errores para la validación del modelado de la dinámica están en algunos casos por encima del error experimental. El modelo va acumulando errores y es útil únicamente para un número determinado de muestras a futuro, divergiendo en exceso a partir de ese punto y dejando de ser utilizable.

Con el objetivo de reducir estos errores se han realizado múltiples pruebas con diferentes métodos de entrenamiento, ya que la fase de aprendizaje es un punto crítico en el modelado debido a lo limitado del conjunto de datos disponible. Por un lado, se

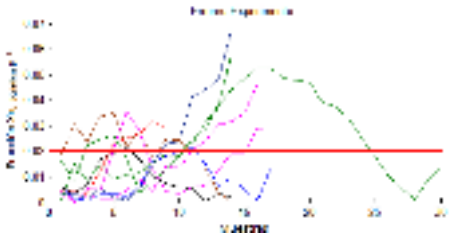


Figura 5: Experimentos cuyos errores de validación superan el máximo permitido

han probado todas las funciones de entrenamiento existentes en el toolbox para redes neuronales de MATLAB® y se ha seleccionado la que mejor resultados proporciona; en este caso *Bayesian Regularization* (trainbr). Por otro, se han diseñado dos formas de entrenamiento: global o por lotes.

4.3. MODELADO GLOBAL POR TEMPERATURAS DEL PROCESO BTO

Tras las pruebas descritas en el apartado anterior se ha llegado a la conclusión de que con los pocos datos disponibles no es posible implementar un único modelo general capaz de asimilar la dinámica de todos los comportamientos del proceso BTO y generalizar con un error inferior al error experimental. La figura 5 muestra cómo 9 de 26 experimentos presentan un error superior al permitido. Tras proceder a un análisis de los datos disponibles, se constató la existencia de puntos de trabajo muy dispares entre los diferentes experimentos. También se observaron dinámicas claramente diferenciadas entre los experimentos que trabajan a temperaturas constantes y aquellos en los que la temperatura varía a lo largo del tiempo. Por tanto, dada la diferencia de comportamiento entre experimentos en función de su temperatura de trabajo, se decidió implementar un modelo concreto para cada rango de temperaturas disponibles. Se ha elegido la temperatura entre el resto de variables de control por ser esta la variable más influyente en el comportamiento del proceso BTO [3].

Además, las diferencias en cuanto a la concentración de olefinas (Xo) entre algunos experimentos pueden llegar a ser de más de un orden de magnitud, estando algunos de sus valores muy cercanos al cero. Todo ello dificulta al modelo estimar correctamente. Por ello, también se ha considerado necesario trabajar con concentraciones relativas en lugar de hacerlo directamente con los valores obtenidos del reactor. Esta concentración relativa se ha definido como la variación de la concentración de olefinas (X_o) en la salida del reactor respecto de dicha concentración en el momento inicial del experimento ($X_o(0)$).



Figura 6: Topología de entrada/salida de la RNA

Por último, otro elemento clave para ser capaz de reproducir correctamente la dinámica del sistema ha sido la información temporal. Los datos experimentales disponibles no han sido obtenidos con un tiempo de muestreo periódico, ni entre experimento, ni entre muestras dentro de cada uno de ellos. Por ello, se ha introducido al modelo como nueva entrada los incrementos de tiempo (Δt) entre cada muestra de los datos. La tabla 4 muestra cómo esta información es muy necesaria para asimilar la dinámica del proceso, ya que representa la velocidad con la que se está llevando a cabo la reacción química dentro del reactor.

Tabla 4: Resultados del modelo con y sin la inclusión de la variable Δt con todas las curvas.

Errores	**Sin Δt**	**Con Δt**
MAE	0.0528	0.0077
RMSE	0.0609	0.0093
MAX	0.3219	0.0523

5. DISCUSIÓN DE RESULTADOS

Como resultado de las modificaciones llevadas a cabo sobre la propuesta inicial (RNA2) para mejorar los resultados proporcionados por el modelo, el nuevo modelo neuronal queda con la configuración mostrada en la figura 6.

Los errores obtenidos por el modelo neuronal son en la práctica totalidad de los casos inferiores al error experimental (ver tabla 5, Hits). Como se observa en la figura 7, las estimaciones del modelo siguen la dinámica del proceso con una precisión satisfactoria. A pesar de la existencia de una zona con un error superior al deseado, esta situación desaparece en puntos posteriores, acercándose de nuevo, el modelo, a la curva real del proceso. Son por tanto, picos puntuales de corta duración en los que no se abandona la dinámica del proceso y que se corresponden con el 4.589 % de las estimaciones que superan el error experimental.

La tabla 5 muestra que los resultados obtenidos hasta el momento han sido muy satisfactorios. En

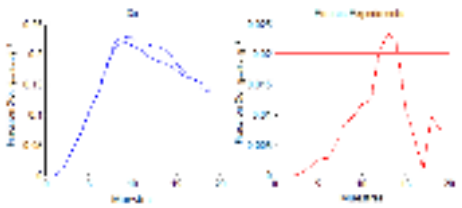


Figura 7: Estimación del modelo para una curva

Tabla 5: Errores de todos los experimentos de los modelos a T de operación constante y variable.

Errores	**T constante**	**T variable**
MAE $\pm$ σ	0.0074$\pm$0.0049	0.0074$\pm$0.0042
RMSE $\pm$ σ	0.0062$\pm$0.0041	0.0060$\pm$0.0033
HITS (%)	95.1637 %	95.6583 %

más del 95 % de las estimaciones del modelo el error cometido ha sido inferior al error experimental, en muchos casos con un orden de magnitud inferior. En la figura 8 se pueden observar las gráficas con los resultados de varios de los modelos a distintas temperaturas de operación, siendo la línea azul continua los valores relativos reales de X_o y la línea azul discontinua los valores relativos de Xo estimados por el modelo neuronal obtenido. En ellas se puede observar como las estimaciones del modelo siguen la dinámica del proceso con unos errores claramente por debajo del error experimental.

Los resultados obtenidos por el modelo neuronal también han sido comparados con los proporcionados por modelos basados en SVM estándar y regresión lineal. La tabla 6 muestra que los resultados obtenidos por el modelo neuronal implementado son claramente mejores que los obtenidos con la regresión lineal y con SVM.

En la figura 9 se vuelve a observar esa diferencia entre las técnicas de modelado para un experimento en concreto, siendo la estimación de la RNA la que más fielmente sigue la curva de referencia.

6. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

En el presente trabajo se han estudiado varias configuraciones basadas en redes neuronales artificiales

Tabla 6: Comparativa de resultados para el modelado a T variable.

Errores	**RNA**	**RL**	**SVM**
MAE	0.006000	0.065781	0.032470
RMSE	0.007400	0.091193	0.040116

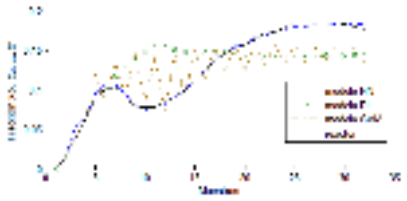


Figura 9: Estimación del modelo para una curva

con el objetivo de obtener un modelo neuronal que reproduzca el comportamiento dinámico de un reactor químico del proceso BTO. Estos modelos han sido iterativamente refinados hasta lograr un modelo modular en función de los rangos de la temperatura de operación capaz de aprender la dinámica del sistema con un error inferior al error experimental.

El modelo neuronal desarrollado mediante la *Neural Network Toolbox* de MATLAB presenta unos resultados muy satisfactorios, obteniendo en más de un 95 % de las estimaciones errores inferiores al error medio experimental.

Durante el estudio comparativo, se ha comprobado que los modelos basados en la regresión lineal son la mejor opción únicamente para el caso de un modelo estático, no recurrente, por su simplicidad y por presentar tiempos de estimación casi 10 veces inferiores a los modelos neuronales. En cambio, para reproducir adecuadamente la dinámica del proceso, manejando información recurrente, el modelo neuronal obtenido es claramente superior a los modelos basados en la regresión lineal o SVM.

Como líneas futuras, partiendo de las redes implementadas hasta el momento, se plantea reducir aún más el error y garantizar el cumplimiento del error límite establecido. Para ello y siguiendo las recomendaciones presentes en la literatura se desea llevar a cabo una inicialización inteligente de los pesos de las redes neuronales. Otro posibilidad para solucionar los problemas derivados del manejo de un conjunto de datos reducido es la implementación de un modelo basado en redes neuronales apiladas (*Stacked Neural Networks*). Y por último, se pretende continuar con las pruebas ya iniciadas para explorar más a fondo las posibilidades de las SVM.

Para determinar la capacidad de generalización del modelo, se pretende realizar una validación estadística empleando la técnica de validación *leave-one-out cross-validation*. En el caso de que varios modelos presentasen un error de validación menor que el error experimental usaremos técnicas de selección de modelos como AIC, VC-dimension o similares.

Una vez finalizada la fase de modelado, el siguiente

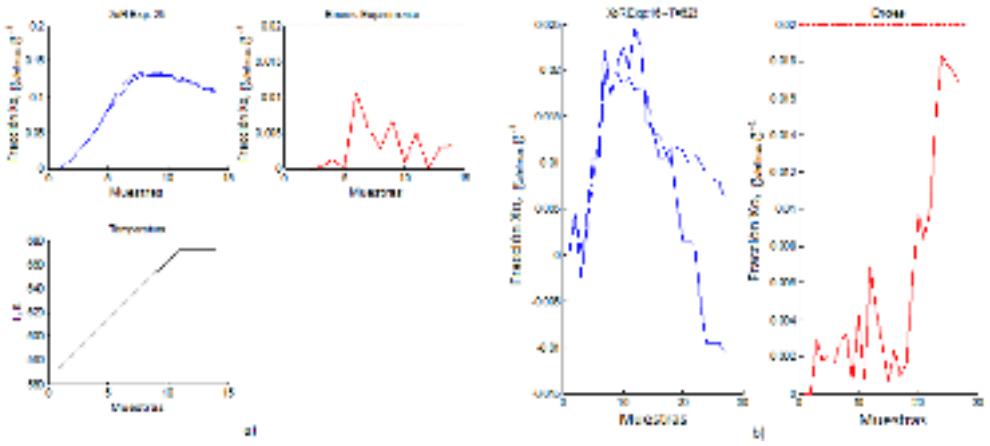


Figura 8: Resultados obtenidos por el modelo neuronal para un experimentos a T variable (a) y para otro a T constante (b)

objetivo será el diseño de un controlador inteligente con el objeto de maximizar la producción de olefinas y reduciendo las fases de regeneración del catalizador del proceso BTO.

Agradecimientos

Se desea agradecer a la Viceconsejería de Universidades e Investigación del G.V. por su financiación a través del proyecto de investigación básica y/o aplicada PI2013-40 y a la Subdirección General de Proyectos de Investigación del MEC por el proyecto DPI2012-32882.

Referencias

[1] S. Al-Asheh, F. S. Mjalli, and H. E. Alfadala. Forecasting influent-effluent wastewater treatment plant using time series analysis and artificial neural network techniques. *Chemical Product and Process Modeling*, 2(3), 2007.

[2] A. Alonso Vicario. *Catalizadores alternativos y modelado cinético del proceso bto (bioetanol a olefinas)*. PhD thesis, UPV/EHU, 2008.

[3] A. G. Gayubo, A. Alonso, B. Valle, A. T. Aguayo, and J. Bilbao. Selective production of olefins from bioethanol on HZSM-5 zeolite catalysts treated with naoh. *Applied Catalysis B: Environmental*, 97(1):299–306, 2010.

[4] A. Ghadirian and M. Zekri. Mimo nonlinear dynamic systems identification using fully recurrent wavelet neural network. In *Control, Instrumentation and Automation (ICCIA), 2011 2nd International Conference on*, pages 1113 –1118, dec. 2011.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[6] M. N. Kashani and S. Shahhosseini. A methodology for modeling batch reactors using generalized dynamic neural networks. *Chemical Engineering Journal*, 159(1):195–202, 2010.

[7] R. Lanouette, J. Thibault, and J. L. Valade. Process modeling with neural networks using small experimental datasets. *Computers & Chemical Engineering*, 23(9):1167 – 1176, 1999.

[8] G. Lightbody, G. Irwin, A. Taylor, K. Kelly, and J. McCormick. Neural network modelling of a polymerisation reactor. In *Control, 1994. Control '94. International Conference on*, volume 1, pages 237–242 vol.1, 1994.

[9] K. Meert and M. Rijckaert. Intelligent modelling in the chemical process industry with neural networks: a case study. *Computers & Chemical Engineering*, 22, Supplement 1(0):S587 – S593, 1998.

[10] E. Molga. Neural network approach to support modelling of chemical reactors: problems, resolutions, criteria of application. *Chemical Engineering and Processing: Process Intensification*, 42(8):675–695, 2003.

[11] P. M. Nørgård, O. Ravn, N. K. Poulsen, and L. K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-London, 2003.

[12] S. Yang, P. Chung, and B. Brooks. Multi-stage modelling of a semi-batch polymerization reactor using artificial neural networks. *Chemical Engineering Research and Design*, 77(8):779 – 783, 1999.

ANÁLISIS DEL PROBLEMA DE EXPLOSIÓN DE ESTADOS EN SISTEMS DICRETOS BASADO EN EL CUBO RUBIK

Emilio Jiménez Macías, F. Javier Leiva Lázaro, Juan Ignacio Latorre Biel, Mercedes Pérez de la Parte
Universidad de La Rioja. Escuela Técnica Superior de Ingeniería Industrial.
C/ Luis de Ulloa 20. 26004 Logroño (La Rioja).
Email: emilio.jimenez@unirioja.es

Resumen

En este artículo se presenta un análisis del cubo Rubik y de sus métodos de resolución empleado para exponer de manera sencilla y fácilmente entendible para alumnos el problema de explosión de estados que sufren los sistemas discretos y las posibilidades de solventarlo basadas en análisis, en simulación o en la combinación de ambos. El objetivo no es avanzar en el conocimiento del cubo, que es empleado simplemente a modo de benchmark, sino mostrar una analogía de cómo en sistemas de producción discretos se da que: a) es posible no disponer de una solución para hacer evolucionar al sistema hasta el estado deseado (la producción deseada), b) o a veces se dispone de una solución, aunque no sea la óptima, c) y la combinación de técnicas de análisis y de simulación muchas veces permite mejorar la solución aunque siga sin ser la óptima d) e incluso puede conocerse la forma de conseguir la solución óptima, pero sea imposible llevarlo a la práctica debido al coste computacional que conlleva calcularla.

Palabras Clave: Problema de explosión de estados, Modelado, Simulación, Análisis, Cubo Rubik.

1 INTRODUCCIÓN

Unos de los mayores problemas a los que se enfrentan quienes tienen que tratar con sistemas discretos es el conocido problema de explosión de estados. Este problema se da en muchos sistemas productivos y logísticos cuyo comportamiento puede representarse mediante un sistema discreto, y lleva a que sea muy complicado planificar la producción para conseguir el resultado esperado, es decir, para llevar el sistema a un estado concreto (por ejemplo, producir 20 coches de un modelo y cierto color, 30 de otro…). La utilización de técnicas de análisis permiten tratar esos sistemas y conseguir soluciones a los mismos, es decir, permiten conocer cómo llevar el sistema al estado deseado (por ejemplo, cómo conseguir producir exactamente los coches deseados para ese día). Pero muchas veces no es posible, o al menos no se conoce, cómo tratar analíticamente de manera eficiente esos sistemas, y en esos casos no queda más remedio que recurrir a la simulación. Pero tanto en unos casos (tratamiento analítico) como en otros (mediante simulación) resulta difícil llegar a conocer una solución óptima del sistema, es decir, cómo llegar al estado deseado de la manera más favorable de acuerdo a unos criterios establecidos (por ejemplo, cómo producir los coches deseados para ese día con el menor número de horas de trabajo, o con el coste económico más bajo posible). Además el problema de conocer la solución óptima podría resolverse mediante simulación, si se dispusiese de posibilidades infinitas de capacidad computacional, es decir, que se conoce cómo encontrar las soluciones óptimas, pero no es posible obtenerlas porque los requerimientos computacionales serían prohibitivos para los recursos (temporales, económicos, o tecnológicos) disponibles. Es muy frecuente que se den esos casos en los que se conozcan soluciones, pero no las óptimas. Además, la combinación de técnicas de simulación y analíticas pueden llevar a mejorar la solución, que sin llegar a ser la óptima (o a veces incluso lo es) sea mejor que la que se tenía.

Todos esos comportamientos, inherentes a los sistemas discretos que sufren el problema de explosión de estados, y por tanto los procesos productivos que pueden modelarse mediante esos sistemas, pueden apreciarse de manera muy intuitiva y gráfica a través del benchmark del famoso cubo Rubik, lo que permite emplear ese sistema como base para poder exponer a alumnos de ingenierías o matemáticas tales comportamientos mediante una analogía fácil de entender.

En este artículo se representa la analogía de la resolución del cubo Rubic como el problema a resolver, análogo a encontrar la solución de un sistema productivo, y se analizan los tipos de soluciones que pueden encontrarse empleando diferentes técnicas analíticas y de simulación, así como combinaciones entre ellas, para los diferentes tipos de estados iniciales que puedan darse.

MODELADO DE UN PROCESO DE EXTRUSIÓN EN ALIMENTACIÓN

Eduardo J. Moya de la Torre, F. Javier Antolín Jiménez, Alfonso Poncela Méndez
(ITAP) Instituto de las Tecnologías Avanzadas de la Producción, Ing. de Sistemas y Automática, EII, Paseo del Cauce 59, Valladolid, 47011, España, edumoy@eis.uva.es

Oscar F. Calvo Martínez
Fundación CARTIF, Parque Tecnológico. Parc 205, Boecillo (Valladolid), 47151, España, osccal@cartif.es

Resumen

Este artículo muestra los avances del trabajo de investigación que, sobre el proceso de extrusión de productos alimentarios, se está realizando en el Área de Instrumentación y Control de Procesos (ICP) de la Fundación CARTIF. En este trabajo se presenta la parte del modelado y construcción del simulador de un extrusor basándonos en las características de uno real. Las fases posteriores de análisis de datos y optimización del proceso tendrán como referencia la comparativa entre nuestro proceso simulado y el real a través de las correspondientes correlaciones.

Palabras Clave: Extrusión, Modelado, Simulación, Identificación.

1 INTRODUCCIÓN

El desarrollo de nuevos productos en la industria es un proceso laborioso que consume ingentes recursos tanto técnicos como humanos durante largos períodos de tiempo.

Este trabajo se justifica por la necesidad de realizar futuros ajustes en la máquina extrusora sin tener que recurrir a ensayos reales. El disponer de herramientas o modelos matemáticos que permitan predecir el comportamiento futuro del proceso de extrusión elimina la necesidad de realizar costosos ensayos con las máquinas reales.

Una ventaja añadida es evitar averías al modificar parámetros de funcionamiento como velocidad del tornillo, alimentación de materia prima, etc… sin saber muy bien cómo se va a comportar el sistema.

La extrusión de alimentos es un proceso en el que un material (grano, harina o subproducto) es forzado a fluir, bajo una o más de una variedad de condiciones de mezclado, calentamiento y cizallamiento, a través de una placa/boquilla diseñada para dar forma o expandir los ingredientes. [8].

Mediante este proceso de inducción de energía térmica y mecánica, se aplica al procesado alta presión y temperatura (en el intervalo de 100-180ºC), durante un breve espacio de tiempo [1]. Como resultado, se producen una serie de cambios en la forma, estructura y composición del producto.

Este tipo de técnicas, se emplea generalmente para el procesado de cereales y proteínas destinados a la alimentación humana y animal. Asimismo, se trata de un proceso que opera de forma continua, de gran versatilidad y alto rendimiento productivo [13].

La extrusión se ha empleado en la industria alimentaria durante los últimos 60 años. En la actualidad, un extrusor se considera un bio-reactor de alta temperatura y corto tiempo de residencia que transforma una amplia variedad de materias primas en productos intermedios modificados o productos finales.

Los requerimientos que, cada vez más, se imponen en los procesos de tratamiento de alimentos son los que han propiciado el auge en el desarrollo de la extrusión para el tratamiento de los mismos. Entre estos requisitos se incluyen fundamentalmente, la alta capacidad de procesado en continuo con alto rendimiento, la eficiencia energética, el procesamiento de materiales viscosos relativamente deshidratados, la mejora de las características de textura y sabor de los alimentos, el control de los cambios térmicos de los componentes de los alimentos y el uso de ingredientes poco convencionales.

La intensa competencia en la industria alimentaria durante los últimos años ha desembocado en una carrera por el desarrollo de nuevos productos. Esto ha fomentado el aumento de la flexibilidad y la versatilidad en los procesos de extrusión. Las industrias están cada vez más obligadas a hacer un mayor énfasis en la calidad de sus productos, lo que necesariamente implica el aumento en el conocimiento de los procesos y la mejora en los sistemas de modelado, simulación, control y optimización de los mismos.

Las condiciones generadas por el extrusor permiten el desempeño de muchas funciones para ser usadas en una amplia gama de alimentos, piensos y aplicaciones industriales [13]. Entre ellas se encuentran: *Aglomeración, Desgasificación,*

Deshidratación, Expansión, Gelatinización, Molido, Homogeneización, Mezcla, Pasteurización y esterilización, Desnaturalización de las proteínas, Conformado, Cizallamiento, Alteración de la textura.

La tecnología de extrusión abre nuevas posibilidades para el procesado de productos alimentarios con dos vías de actividad, en primer lugar el desarrollo de nuevos alimentos y en segundo lugar la mejora de las propiedades funcionales de productos vegetales [13].

Respecto de la primera actividad, la extrusión puede modificar distintos materiales alimenticios para producir una diversidad de nuevos productos en distintos sectores (alimentación animal, acuicultura, snack y aperitivos, cereales para desayuno, productos para confitería, alimentación infantil, análogos de carne, etc.).

Respecto de la segunda actividad, los recientes avances en tecnologías de procesado están permitiendo el desarrollo de nuevos ingredientes que favorecen las cualidades de distintos alimentos, al mejorar sus propiedades emulsificantes, la capacidad de retención de agua, modificación de textura y aroma, etc. En este sentido, el procesado mediante extrusión permite por ejemplo, la texturización de proteínas para el desarrollo de extensores y sustitutivos cárnicos

Esta variedad de funciones y de actividades, que se pueden realizar mediante la tecnología de extrusión, van a depender de los productos o materias primas utilizadas (recetas), del tipo de extrusor utilizado (equipo) y de las condiciones de funcionamiento (variables de proceso).

En este artículo se mostrará un breve resumen del trabajo que en la actualidad se está realizando para modelar el funcionamiento de un extrusor de doble eje, utilizado para desarrollar nuevos productos y modificar las propiedades funcionales de algunos ingredientes en el Centro Tecnológico CARTIF [2], [3], [9], [10] y [11], con el objetivo de conocer a priori el comportamiento del equipo, así como la relación existente entre las distintas variables de proceso.

El presente trabajo tiene como objetivos principales:

1. Comprensión del proceso a controlar determinando las variables de entrada, salida e intermedias del sistema.
2. Modelado del proceso a partir de datos experimentales con el consiguiente diseño de experimentos para obtenerlos.
3. Obtención de modelos, a partir de datos experimentales, con la herramienta Ident de Matlab.
4. Simulación de los modelos obtenidos y evaluación para representar el comportamiento dinámico del sistema.

Ahora vamos a definir alguna terminología para entender el proceso de extrusión:

- *Materia prima:* El material o la mezcla que van a ser procesados en el extrusor.
- *Tornillo sinfín:* Es el elemento que transporta el producto a través del extrusor.
- *Filete:* Es la superficie de transporte helicoidal.
- *Paso*: Es el ángulo de ataque, relativo al eje del cilindro.
- *Cuba:* Es la tubería (depósito) donde el tornillo extrusor gira.
- *Sobrecubierta enfriadora/calentadora:* Es una funda alrededor de la cuba para circulación de agua refrigerada, o vapor, u otro medio de calentamiento como resistencias eléctricas.
- *Molde:* Final del montaje para configuración del producto cuando sale del extrusor
- *Cortador:* Montaje que corta el producto final en pieza de longitud deseada.

El artículo está organizado de la manera siguiente. Aquí se hace una introducción sobre el concepto de extrusión y su utilidad. En la segunda sección se presenta una breve descripción del proceso que queremos modelar con sus principales características técnicas. En la tercera sección se hace un breve recordatorio de los conceptos relacionados con el modelado y la identificación. En la cuarta sección se presenta la identificación realizada sobre el extrusor de CARTIF para varios tipos de harinas. Finalmente se presentan las conclusiones finales y las líneas de trabajo planteadas en un futuro inmediato, así como las referencias bibliográficas utilizadas.

2 DESCRIPCIÓN DEL PROCESO A MODELAR

El proceso de extrusión debe tratarse esencialmente como un sistema de múltiples entradas y múltiples salidas (MIMO). La identificación y modelado [15] del equipo requiere que se elijan de forma adecuada las variables de control, y las variables manipuladas. El equipo es de doble eje en co-rotación de 15kW de potencia en el motor principal, modelo CLEXTRAL EV-25. [4], [5], [6].

La máquina de extrusión para alimentación está formada por varias partes. Una tolva por donde se introduce el producto, una bomba de agua para introducir humedad en el producto, un motor para hacer girar el tornillo extrusor y 6 zonas reguladas en temperatura mediante resistencias calefactoras y anillos de refrigeración por agua. Cada zona dispone de una sonda PT100 que indica su temperatura. A continuación se encuentra un cortador para obtener el

tamaño de producto final deseado.

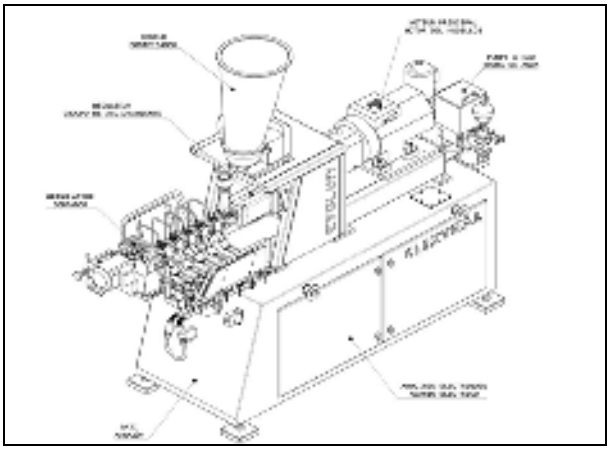


Figura (1): Sinóptico máquina extrusora

La máquina extrusora de la que dispone CARTIF modelo CLEXTRAL EV-25, tiene las siguientes características:

- *Tornillo:* El tornillo es del tipo doble del tipo imbricado o intermallado y en co-rotación (los 2 tornillos giran en el mismo sentido. Es del tipo segmentado y los ejes son del tipo estriado para poder cambiar la configuración de los segmentos.
- *Segmentos del Tornillo*: Podemos disponer de hasta 13 tipos [XX].

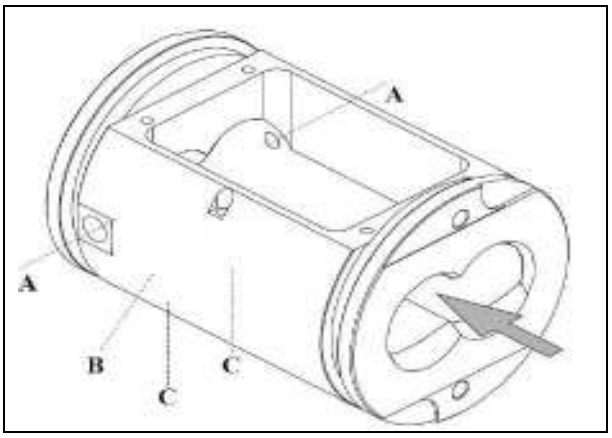


Figura (2): Imagen de un módulo AB1

- *Módulos de la máquina*: Los módulos de manguito son en los que van insertados los tornillos con sus respectivos segmentos. Les hay de varios tipos y pueden ir refrigerados/calefactados, etc... Vienen unidos mediante bridas especiales. Podemos distinguir entre:
  - ✓ *Módulo AB1*: De suministro de la primera zona con una ranura y agujeros debajo del módulo para su fijación sobre el soporte manguito de la máquina. Viene equipado con parches superiores.
  - ✓ *Módulo FER*: Módulo cerrado
  - ✓ *Módulo AB:* de suministro de zonas intermedias. Viene equipado con parches superiores.
  - ✓ *Módulo ABF*: De suministro para zonas intermedias, con suministro superior y lateral. Viene equipado con parches laterales
  - ✓ *Módulo VAP*: Para inyección de vapor

Donde: A son los orificios laterales de inyección de productos o colocación de sensores
B donde la va sonda de temperatura del manguito
C conexiones del circuito de refrigeración

Tabla 1: Características técnicas del extrusor EV025

Nombre	**Valores**
Marca	CLEXTRAL
Motor principal	13,5kW, 2000rpm
Velocidad de eje	1200 rpm
Consumo máximo	30 A
Longitud de eje extrusor	600mm
Potencia frigorífica	3,9kW
Modelo	EV025
Par de extrusión	1,6
Módulos de control de Tª	6
Potencia calorífica por módulo	1,1kW
Boquilla	2mm
Perfil de boquilla	02
Hueco libre	2,5 mm
Nº Cuchillas cortador	4

Dentro de la configuración mecánica del tornillo encontramos tres zonas: una zona de alimentación, una zona de compresión y una zona de cizallamiento.

Tabla 2: Calibración alimentación muestras

Nombre	**Valores**
Alimentación máxima muestra	67,0 Kg/h
Tiempo muestra	36 seg
Set Point Alimentación	20,0 Kg/h
Peso muestra	170 g
Alimentación máxima	56,95 Kg/h
Alimentación muestra	17,0 kg/h

Vamos a hacer una descripción de las entradas/salidas del sistema que nos van a servir para hacer el modelado posterior con MATLAB.

Como entradas del sistema vamos a tener:

- Velocidad del tornillo: Se puede configurar en el SCADA del sistema y se mantiene gracias un variador de velocidad que actúa sobre el motor principal de la máquina. Viene indicado en rpm.
- Alimentación: Es la cantidad de producto que se introduce en la máquina. Viene indicado en Kg/h.
- Agua añadida: Es la cantidad de agua que se introduce en la máquina. Viene indicado en Kg/h.

- Temperaturas zona 1-6: Son las temperaturas del encamisado en las zonas calefactoras/refrigeradas dentro de las cuales gira el tornillo extrusor. Estas zonas tienen una temperatura de consigna y la temperatura real marcada por una sonda PT100, el control se realiza para cada zona mediante un PID de acción todo/nada, bien sea a la resistencia calefactora o al encamisado refrigerante por medio de la activación de una electroválvula de agua fría.

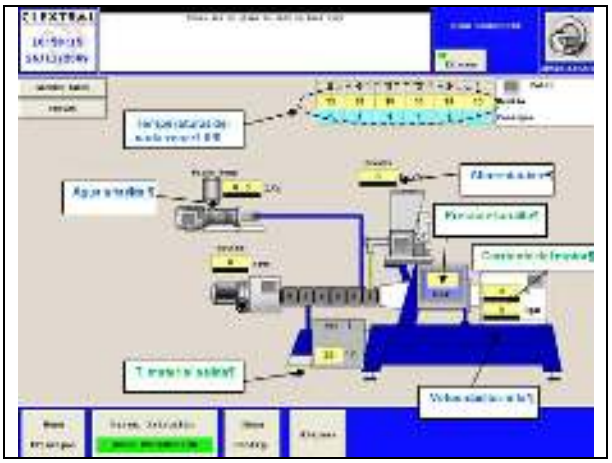


Figura (3): SCADA de la máquina extrusora

Por lo tanto la planta se encuentra en lazo abierto salvo las temperaturas en las 6 zonas del extrusor. Esto quiere decir que podemos modificar las entradas individualmente siempre dentro de sus límites de funcionamiento especificados en el manual de instrucción de la máquina.

Como salidas del sistema tenemos:

- Consumo del motor: O par en el eje del tornillo ya que el par es proporcional a la corriente consumida. Viene indicado en Amperios.
- Presión en el tornillo extrusor: La sonda de presión se encuentra ubicada en el tope del eje. Esta presión viene indicada en bares.
- Temperatura material de salida: Es la temperatura del producto a la salida de extrusor. Viene indicada en ºC.

3 MODELADO, IDENTIFICACIÓN DEL SISTEMA

Cuando se hace necesario conocer el comportamiento de un sistema en unas determinadas condiciones y ante unas determinadas entradas, como en este caso, se puede recurrir a la experimentación sobre dicho sistema y a la observación de sus salidas. Sin embargo, en muchos casos la experimentación puede resultar compleja o incluso imposible de llevar a cabo, lo que hace necesaria trabajar con algún tipo de representación que se aproxime a la realidad, y a la que se conoce como modelo.

Todo modelo matemático o paramétrico consta de una o varias ecuaciones que relacionan las entradas y salidas (además de la variable tiempo, importante en los modelos dinámicos). Por esto, los modelos matemáticos se les conoce comúnmente como modelos paramétricos, ya que pueden definirse mediante una ecuación más un número finito de parámetros.

En cuanto a los métodos de obtención de modelos, existen dos métodos principales para obtener el modelo de un sistema:

1) Modelado teórico: Se trata de un método analítico, en el que se recurre a leyes básicas de la física para describir el comportamiento dinámico de un fenómeno o proceso.
2) Identificación de un sistema: Se trata de un método experimental que permite obtener el modelo de un sistema a partir de datos reales recogidos de la planta.

El modelo teórico tiene un campo de aplicación limitado a procesos sencillos de modelar, o a aplicaciones en que no se requiera gran exactitud en el modelo obtenido. En muchos casos la estructura del modelo obtenido a partir del conocimiento físico de la planta posee un conjunto de parámetros desconocidos y que sólo se pueden determinar experimentando sobre el sistema real. De aquí la necesidad de recurrir a los método de identificación de sistemas.

Los modelos obtenidos mediante técnica de identificación tienen algunas desventajas:

- Su rango de validez suele ser limitado (sólo son aplicables a un determinado punto de trabajo, un determinado tipo de entrada y para un proceso concreto)
- En muchos casos es difícil dar significado físico al modelo obtenido, puesto que los parámetros identificados no tienen relación directa con ninguna magnitud física. Estos parámetros se utilizan sólo para dar una descripción aceptable del comportamiento del sistema.

En la práctica, lo ideal es recurrir a una mezcla de ambos métodos de modelado para obtener el modelo final. El uso de datos reales para identificar los parámetros del modelo provee a este de una gran exactitud, pero el proceso de identificación se ve tanto más facilitado cuanto mayor es el conocimiento sobre las leyes físicas que rigen el proceso.

En el presente artículo se ha optado por la identificación de sistema debido a la complejidad del sistema (9 entradas y 3 salidas). Se han utilizado métodos paramétricos ARX y de espacio de estado ya que son los únicos que permiten varias entradas/salidas.

El proceso de identificación comprende los siguientes pasos:

a) Obtención de datos del proceso (entradas-salidas): En este caso, hemos obtenido los datos realizando experimentos sobre la propia planta del sistema (la máquina de extrusión) para distintos materiales, monitorizando todos estos datos de entrada-salida.
b) Tratamiento previo de los datos: Los datos registrados están generalmente acompañados de ruidos y otro tipo de interferencias. De todas maneras, la herramienta *ident* también contiene varios filtros para facilitar y mejorar el proceso de identificación.
c) Elección de la estructura del modelo. Si el modelo que se desea obtener es un modelo paramétrico, el primer paso es determinar es determinar la estructura deseada para dicho modelo. En este caso al ser un sistema multivariable solo podemos utilizar el modelo ARX y el modelo de espacio de estado.
d) Obtención de los parámetros del modelo: A continuación se procede a la estimación de los parámetros de la estructura que mejor se ajusta a los datos de entrada-salida obtenidos experimentalmente.
e) Validación del modelo: El último paso consiste en determinar si el modelo obtenido satisface el grado de exactitud requerido para la aplicación en cuestión. Si se llega a la conclusión de que el modelo no es válido.se deben revisar los siguientes aspectos como posibles causas:
    1. El conjunto de datos entrada-salida no proporcionan suficiente información sobre sobre la dinámica del sistema.
    2. La estructura escogida no es capaz de proporcionar una buena descripción del modelo.
    3. El criterio de ajuste de parámetros seleccionado no es el más adecuado para el proceso.

Dependiendo de la causa estimada, deberá repetirse el proceso de identificación desde el punto correspondiente.

El registro de los datos se realiza aprovechando las pruebas de desarrollo de producto que se realizan con el equipo, y se tiene la limitación que no se pueden realizar cambios bruscos del punto de funcionamiento, ya que existe el peligro de deteriorar el producto. Se han realizado toma de datos con variantes de tipos de harinas en clases y propiedades. Previamente se realizaron pruebas de acondicionamiento de los productos hasta alcanzar las propiedades más idóneas del producto deseado.

Como ya se comentó en un trabajo anterior [YY], para la identificación se utiliza la herramienta "System Identificatión Toolbox" del software MATLAB. Después de importar los datos desde un fichero Excel, hay que aplicarles varios tratamientos. Primero se aplican modificaciones necesarias para para identificar las variables de entrada y salida de dicho fichero con el correspondiente script de MATLAB (iddata.m) Luego se eliminan valores constantes de los datos mediante "Remove means", a continuación se aplica un filtrado pasa-bajos para eliminar el ruido y las pequeñas oscilaciones con "Filter". Una vez realizado el procesamiento de los datos, se puede realizar la estimación de modelos. En cuanto al enfoque de la estimación, o "Focus" hemos utilizado la opción "Simulación".

Recordar que el valor de cada salida de los modelos cuando excitamos cada una de las entradas individualmente ante una entrada escalón, se puede visualizar con LTI Viewer.

El tipo de estructura utilizada como ya hemos indicado es ARX y de espacio de estado PSS. Para identificar cada uno de ellos añadimos una serie de números que indican el número de ceros, de polos y retraso en un caso, y el orden del sistema en el otro.

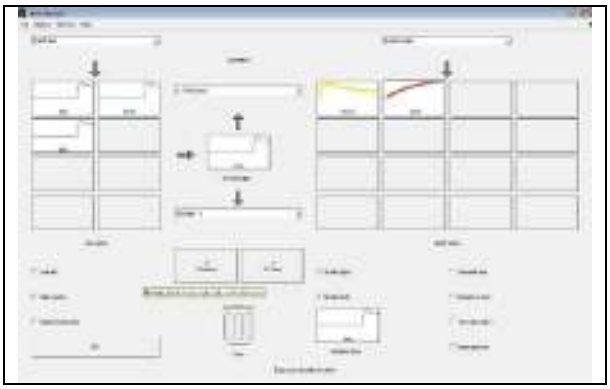


Figura (4): Herramienta Ident

4 RESULTADOS OBTENIDOS

A continuación se muestran los resultados obtenidos con los ensayos realizados: Vamos a realizar 3 ensayos con distintas materias primas:

1. Harina de garbanzos con NaCl (1,3 %).
2. Harina de trigo.
3. Harinas de pescados.

4.1- HARINA DE GARBANZOS NaCl (1.3%)

Los modelos que mejor se ajustan son el ARX221 y el PSS4

Los ajustes para las tres salidas del sistema están por encima del 90% por lo que los 2 modelos representan muy bien el comportamiento del sistema para este tipo de materia prima.

A continuación se realiza la simulación [7], [11], para ello es necesario enviar las funciones de transferencia del modelo desde Workspace a Simulink, así como los datos de entrada. En primer lugar , vamos a obtener las funciones de transferencia en tiempo continuo aunque luego al simular, utilizamos las funciones en tiempo discreto (transformada z) ya que los datos no son continuos (tiempo de muestreo 1 segundo). Una vez pasadas las funciones de transferencia de Matlab a Simulink así como las entradas del sistema, podemos comenzar con la simulación para obtener las salidas simuladas (I Motor Sim; P Tornillo Sim; T Material Sim).

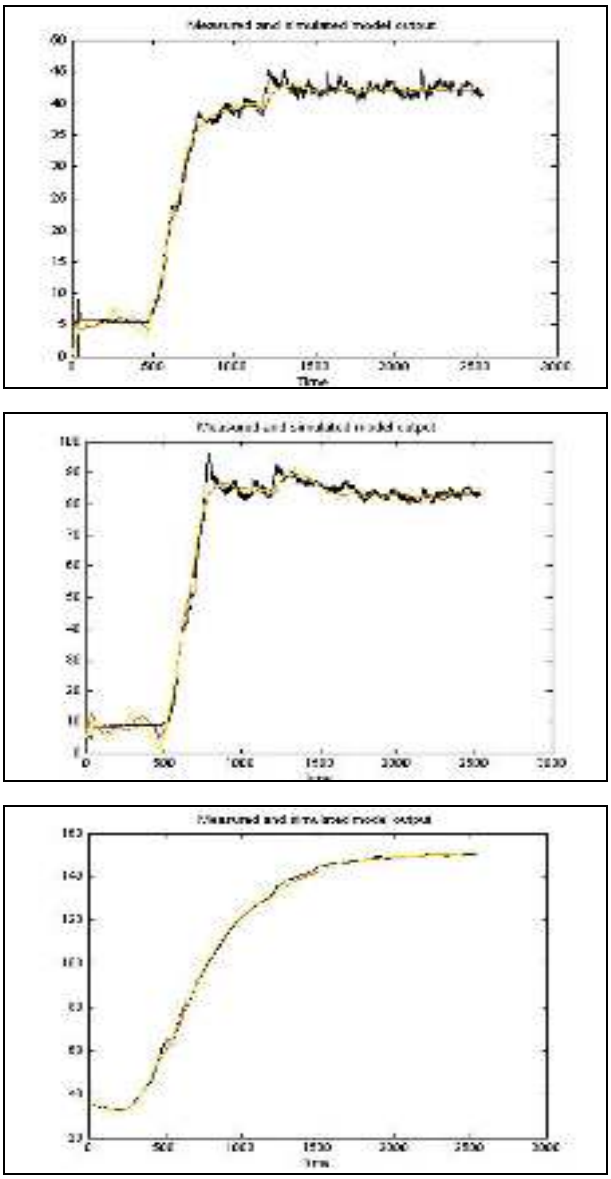


Figura (5): a) Salida Y1 (M_I_MOTOR), b) Salida Y2 (M_P_Tornillo) y c) Salida Y3 (M_T_Material)

Después pasamos las salidas simuladas a Matlab para poder dibujarlas con las salidas reales del sistema y ver los resultados de la simulación. La aplicación práctica es poder obtener las salidas simuladas aplicando variaciones en las entradas del sistema (para cada tipo de material) sin necesidad de hacerlo en la propia máquina, con el consiguiente ahorro de tiempo y dinero.

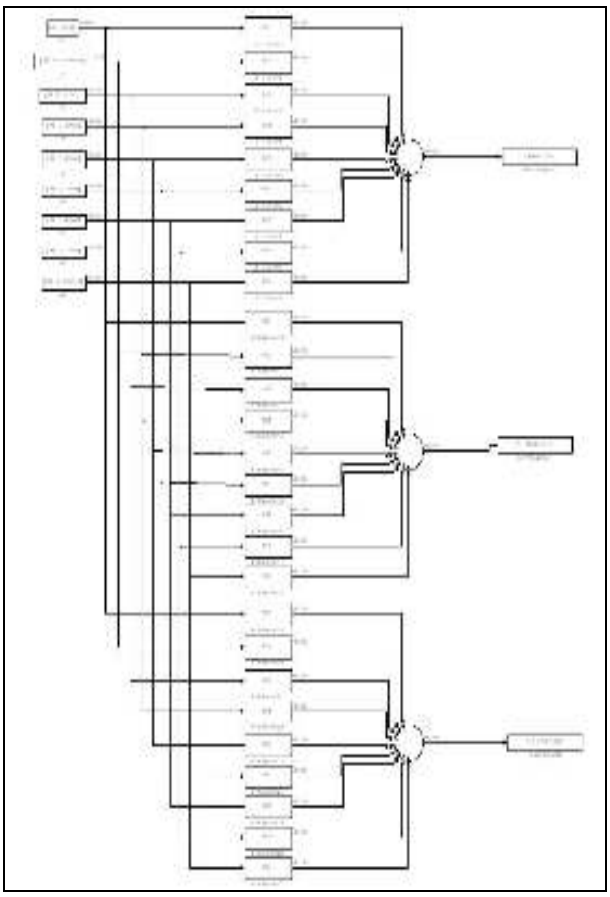


Figura (6): Modelo en SIMULINK

Si hacemos el análisis de residuos se puede ver que los 2 modelos tienen buen ajuste y correlación cruzada entre cada una de las 9 entradas y la salida Y1, que es la corriente en el motor del extrusor. En el caso del modelo ARX221, el ajuste es sensiblemente mejor ya que está más cerca de 0.



Figura (7): Ejemplo función transferencia

4.2- HARINA DE TRIGO

También los modelos que mejor se ajustan para la salida Y1 (M_I_MOTOR) son el ARX221 con 74,9% y el PSS4 con 75,79%., para la salida Y2 (M_P_Tornillo) son 77,16% y 78,41% y salida Y3 (M_T_Material) son 95,84% y 97,68%. Ambos

modelos se ajustan menos que el experimento anterior, principalmente porque el sistema es menos estable. Subiendo el orden de los modelos ARX333 y PSS5, no se obtienen mejoras significativas.

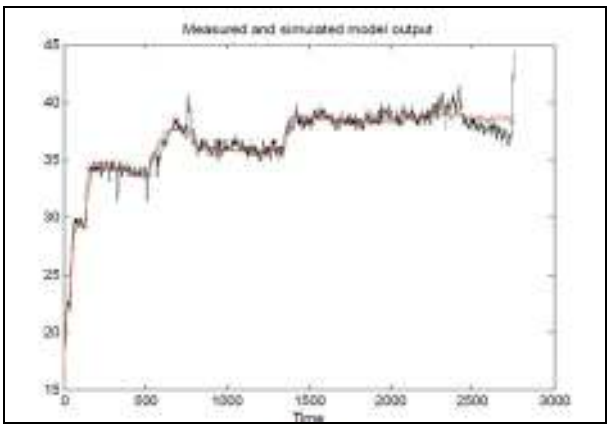


Figura (8): Salida Y1 (M_I_MOTOR) - Trigo

4.3- HARINA DE PESCADOS

También se han utilizado modelos de estructura tipo ARX y Espacio de estados como los anteriores y subiendo el orden de los modelos para ver si se produce mejor aproximación de las salidas al funcionamiento real, ya que este es mucho más inestable. Los modelos usados son del tipo ARX555 y PSS10. Esto influye a la hora de obtener las funciones de transferencia del sistema, ya que son de orden más elevado y por lo tanto requieren más tiempo para su simulación.

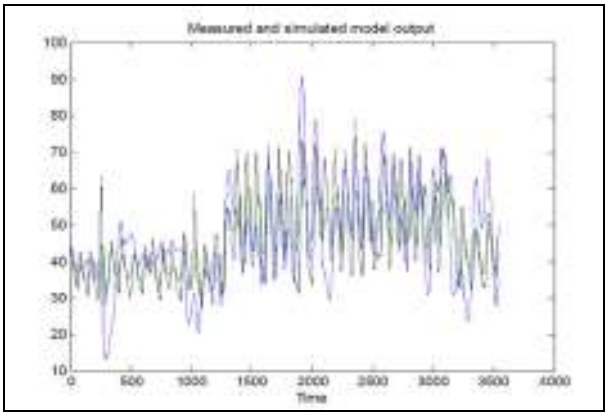


Figura (9): Salida Y1 (M_I_MOTOR) - Pescado

El mejor ajuste es de un 36,42 % por lo que los 2 modelos se ajustan poco principalmente porque el sistema es muy inestable por las propiedades del material de alimentación. Al subir el orden de los modelos a ARX555 y PSS10, el mejor ajuste es de un 17,96 % para el caso del ARX555 no se obtiene mejora en los resultados.

En todos los casos una vez obtenidos los modelos y planteadas las simulaciones se puede producir un gran ahorro de tiempo y energía en tiempo real visualizando las salidas del sistema: La corriente del motor del husillo del extrusor en amperios (I_MOTOR_TORNILLO), la presión dentro del extrusor (P_TORNILLO) en bares y por último, la temperatura del material dentro del extrusor (T_MATERIAL) en ºC.

Por último y para completar las simulaciones vamos a utilizar los datos del experimento 1 con las funciones de transferencia del experimento 2. El tipo de modelo es el ARX. Con esto vamos a ver que al simular, hay mucha diferencia entre las salidas reales y las salidas simuladas.

De esta manera justificamos la creación de un modelo en ident para cada tipo de material de alimentación en la extrusora. Por lo tanto, para cada experimento nuevo hay que modelar el sistema para generar las funciones de transferencia.

Esto es debido a que los materiales utilizados generalmente como material de alimentación de la extrusora son harinas de origen vegetal o animal. Estas harinas se comportan a altas temperaturas, como las que existen dentro del extrusor, como fluidos no-newtonianos de tipo pseudoplásticos. Es decir, la viscosidad de las harinas disminuye con el gradiente de la velocidad de deformación.

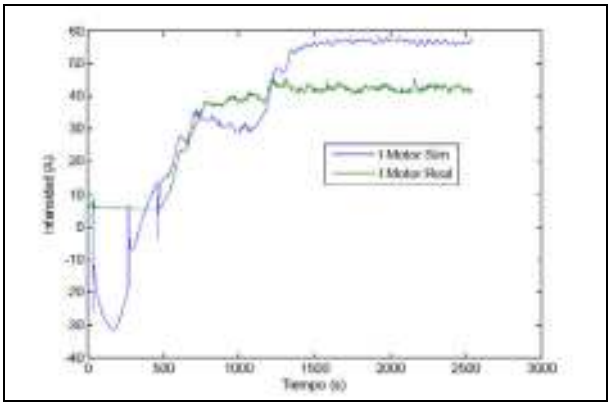


Figura (10): Respuesta Harina - Trigo

Se puede ver que la corriente en el motor simulada y real se asemeja poco hasta cuando están en régimen estacionario

5 CONCLUSIONES Y LINEAS FUTURAS.

En cuanto a las conclusiones que podemos extraer del presente trabajo tenemos:

1) Se ha comprobado que la simulación constituye una herramienta válida y eficaz para el estudio del comportamiento dinámico de los sistemas de extrusión en alimentación.
2) La identificación, mediante los datos

experimentales es una herramienta muy potente y se pueden obtener modelos muy aproximados al sistema real.

3) Los modelos pueden simularse para predecir el comportamiento del sistema. Este es el objetivo principal de este proyecto.
4) Las simulaciones corresponden al comportamiento dinámico del sistema en torno al punto de equilibrio, generalmente el del funcionamiento nominal para cada tipo de materia prima.
5) Para cada tipo de materia prima y configuración del husillo del extrusor, hay que realizar un nuevo modelado para obtener funciones de transferencia específicas.

En cuanto a las líneas futuras de trabajo, que ya están fijadas, y en algunos casos se están realizando avances interesantes son:

1) Desarrollar una aplicación que permita la identificación y posterior simulación del proceso que se ejecute en paralelo con el funcionamiento de la planta. Esta aplicación permitirá al operador de la planta ensayar las modificaciones de las consignas de funcionamiento en el simulador con anterioridad a realizarlo en la planta.
2) Desarrollar el control de la presión y temperatura del producto en la boquilla de salida del extrusor que regule de forma automática: la alimentación de sólido (kg/h), la adición de líquido (l/h), la velocidad del eje (rpm) y el perfil de temperaturas del extrusor (ºC)
3) Establecer los límites de funcionamiento de determinados parámetros del proceso: velocidad del eje en función de la alimentación de sólido y líquido, de adición de líquido en función de la alimentación de sólido y de temperatura en cada una de las zonas del extrusor.

Referencias

[1] Antila, J., Pipatti, R., and Linko, P., (1984) "Process control and automation in extrusion cooking". *Termal Processing and Quality of Foods. P Zuthen, eds. Elsevier Applied Science Publ. London.* pp. 44-48.

[2] Bécares, J., Cárdenas, C., Calvo, O., García, D., and E. Moya; "Modelado y simulación de un extrusor de simple tornillo". CAIP'09, 569-574 (2009).

[3] Calvo, O., Moya, E., and C. Cárdenas; "Extrusión de productos alimentarios: Análisis energético de los extrusores e identificación del modelo de un extrusor de doble eje". CAIP'11, ISBN: 978-84-8458-355-4 (2011).

[4] Eerikäinen, T, Zhu, Y.-H. and Linko, P., (1994) "Neural networks in extrusion process identification and control". *Food Contro,l 1994. Volume 5. Number 2,* pp. 111-119.

[5] Haley, Timothy A., (2000) "On-line system identification an control design o fan extrusion cooking process: Part I. System identification". *Food Control 11 (2000),* pp. 103-120.

[6] Janssen, L.P.B.M., (1986) "Models for cooking extrusion". *Food Engineering and Process Application Vol. 2, Unit Operations (Ed. Mavyer, M. and Jelen, P.) Elsevier Applied Science Publishers, Ltd. Essex,* pp. 115-130.

[7] Kulshereshtha, Manoj K., (1995) "Simulating the performance of control system for food extrusion using model-based set-point adjustment". *Food Control, 1995. Volume 6, Number 3,* pp. 135-141.

[8] Mercier, C., Linko, P., Harper, J.M., (1989). Extrusion Cooking. Ed. Amer Assn of Cereal Chemists.

[9] Moya, E., and J. Bécares; "Modelado y simulación de un proceso de extrusión". XXIX Jornadas de Automática, Tarragona, ISBN: 978-84-691-6883-7, F-264 (2008).

[10] Moya, E., Bécares, J., and O. Calvo; "Extruders in food applications: Simulation of extruder operation based of an energy model". IX Simposio de Automatización, La Habana (Cuba), ISBN: 978-959-286-010-0, AUT35 (2009).

[11] Moya, E., Poncela, A., and V. Castro; "Modelado de un proceso de extrusión de polímeros". XXXIII Jornadas de Automática, Vigo, ISBN: 978-84-8158-583-4, F-114 (2012).

[12] Newell, R.B., and Lee, P.L., (1989) Applied Process Control; A Case Study, Prentice Hall, Inc., NJ.

[13] Pipatti, R., (1983) Modeling o fan HTST extrusion cooker, M. Sc. Thesis, Helsink., Univ. of Technology, Finnish.

[14] Seborg, D.E., Edgar, T.F. and Mellichamp, D.A. (1989) Process Dynamics and Control, John Wiley & Sons Inc., New York.

[15] Singh, Bhajmohan and Mulvaney, Steven J., (1994). "Modeling and process control of twin-screw cooking food extruders". *Journal of Food engineering 23 (1994),* pp. 403-428.

Reducción de modelos en Biología Sintética: Modelado y simulación de ruido y variabilidad en expresión genética.

Yadira Boada, Alejandro Vignoni, Jesús Picó
{yaboa, alvig2, jpico}@upv.es
Institut d'Automàtica i Informàtica Industrial, Universitat Politècnica de València,
Camì de Vera s/n, 46022 Valencia, España

Resumen

El problema de ruido y variabilidad genética en poblaciones de microorganismos es conocido en biología. El ruido y la variabilidad en la expresión genética, en biología sintética juega un papel clave en la dinámica celular. Su análisis y simulación son temas de interés para la comunidad científica, que pretende entender, analizar y luego diseñar circuitos genéticos con comportamientos preestablecidos. En esta contribución, proponemos un modelo reducido que es capaz de captar las principales características que se desean modelar con suficiente exactitud. Este modelo permite simular ruido intrinseco y extrinseco junto con variabilidad en la población, de forma sencilla utilizando herramienta de teoría de control, como tamaño del estado. Los resultados de las simulaciones muestran que la aproximación que realiza el modelo reducido es acertada para los fines que se desea modelar al sistema. La simulación de ruido extrínseco o intrínseco, son mas tratables computacionalmente para simular la variabilidad poblacional, que simulaciones estocásticas del tipo Monte Carlo - Gillespie.

Palabras clave: expresión genética, ruido, variabilidad, simulación de sistemas biológicos, modelos reducidos.

1. Introducción

Un área que ha ganado especial relevancia en los últimos años dentro de la Biología Sintética es el diseño de la comportamientos de poblaciones de microorganismos, a partir de la interacción entre diferentes células individuales. Un enfoque común para inducir comportamientos colectivos es utilizar mecanismos de comunicación de célula a célula y circuitos genéticos para lograr una respuesta predeterminada. Por otro lado, la expresión genética es un proceso inherentemente estocástico, y juega un papel clave en la dinámica celular [2]. A nivel de la población, el efecto de el ruido se hace evidente por el hecho de que genéticamente idénticas células producen la misma proteína en diferentes concentraciones [7]. La variabilidad en las concentraciones de proteína se puede cuantificar con tecnologías de alto rendimiento, tales como la citometría de flujo, que permiten caracterizar esta variabilidad en términos de la histogramas de la población para la abundancia de la proteína [3].

En el presente trabajo se propone un modelo reducido, proveniente del análisis de un modelo completo que representa la expresión genética de circuitos sintéticos descritos en la sección 3 y cuyos principios básicos se detallan en la sección 2. El modelo reducido servirá para la simulación de ruido en la expresión genética y la variabilidad de la concentración de proteínas, los resultados se comparan tanto con simulacions determinísticas del modelo completo a partir de ODEs, como con simulaciones estocásticas SSA a partir de las reacciones mostradas en la sección 4. De los datos de simulación, se observa que el modelo reducido se comporta de manera adecuada y se puede utilizar para reemplazar al modelo completo en el caso de simulación de poblaciones de cientos de miles de células, donde los modelos extendidos y las simulaciones estocásticas requieren de mucho coste computacional. Finalmente la sección 5 presenta las conclusiones obtenidas.

2. Modelado de la expresión genética

2.1. Dogma central

El ADN (ácido desoxi-rribonucléico) puede subdividirse en segmentos de información llamados genes que contienen instrucciones codificadas para la producción de una proteína única con una función especializada dentro de cada célula. Para leer un gen y producir la cadena de aminoácidos que forman una

proteína, las células realizan un proceso en 2 etapas: *i)* trascripción y *ii)* traducción que se le conoce como el *Dogma Central* de la Biología.

En la transcripción, se copia la secuencia de ADN con sus bases adenina (A), guanina (G), citosina (C) y timina (T) en forma de ARN mensajero (ácido ribonucléico o ARNm) que también contiene 4 bases, con la diferencia de que en el ARNm el uracilo (U) reemplaza a la timina (T). La transcripción la realiza una enzima denominada ARN polimerasa, esta se une a regiones específicas del ADN conocidas como *promotores* que controlan el inicio de la transcripción de un gen. Al promotor se adhiere también otro compuesto denominado *factor de transcripción* que habilitan o impiden el inicio de la transcripción.

En la traducción, la información del ARNm es traducida por los ribosomas que generan la proteína.

2.2. Reacciones y ecuaciones de balance de masa

El Dogma Central de biología molecular puede condensarse en reacciones que incluyen tasas de cambio y de degradación entre sus elementos fundamentales: gen, ARNm y proteína.

$$\text{Gen} \xrightarrow{Transcr} \text{ARNm} \xrightarrow{Transl} \text{Proteína}$$

$$\text{ARNm} \xrightarrow{Degrad} \emptyset$$

$$\text{Proteína} \xrightarrow{Degrad} \emptyset$$

Si el proceso anterior no está regulado, se dice que su expresión es *constitutiva* y el gen siempre está encendido. Utilizando la ley de acción de masas el modelo se define según [11]:

$$\dot{m} = k_1 - d_1 m \tag{1}$$

$$\dot{p} = k_2 m - d_2 p \tag{2}$$

donde m es la concentración de ARNm, p es la concentración de proteína, k_1 es la tasa de transcripción constitutiva, k_2 es la tasa de traducción, d_1 es la tasa de degradación de ARNm y d_2 es la tasa de degradación de proteína.

Si los factores de transcripción controlan la expresión genética ya sea activando o inhibiendo su transcripción, se dice que su expresión es *regulada* mediante un activador o un represor.

$$\text{Gen} \xrightarrow{\text{Activador/Represor}} \underset{\downarrow}{\text{ARNm}} \longrightarrow \underset{\downarrow}{\text{Proteína}}$$
$$\qquad\qquad\qquad\qquad\quad \emptyset \qquad\qquad\quad \emptyset$$

La tasa de reacción anterior tiene la forma de una *función de Hill* que señala la probabilidad de que un factor de transcripción (activador o represor) se una a la ARNm polimerasa e inicie la etapa de transcripción. Por ejemplo, en un activador cuanto mayor sea su concentración, mayor será la producción de proteína. Ahora las ecuaciones 1 y 2 resultarían:

$$\dot{m} = k_1 \frac{A^n}{K^n + A^n} - d_1 m \tag{3}$$

$$\dot{p} = k_2 m - d_2 p \tag{4}$$

donde m, p y A son la concentraciones de ARNm, proteína y activador respectivamente, k_1 es la tasa máxima de transcripción, K es el coeficiente de activación, n es el coeficiente de Hill (número de activadores necesarios para unirse cooperativamente al promotor e iniciar activación de la expresión del gen).

Considerando ahora un represor el modelo resultaría:

$$\dot{m} = k_1 \frac{K^n}{K^n + R^n} - d_1 m \tag{5}$$

$$\dot{p} = k_2 m - d_2 p \tag{6}$$

donde R es la concentración del represor, K es el coeficiente de represión.

2.3. Simulación estocástica

Los modelos biológicos son de dimensión microscópica e inherentemente estocásticos, haciendo que las reacciones anteriores se describan ahora como una *probabilidad*. Las herramientas más utilizadas para el análisis de sistemas estocásticos son los algoritmos de Monte Carlo, concretamente el algoritmo de simulación estocástica (SSA) de Gillespie y sus variantes. Este es exacto, en el sentido de que su resultado es una variable aleatoria con una distribucción de probabilidad exactamente igual a la solución de su correspondiente *chemical master equation* (CME) [6].

Cada paso del SSA de Gillespie comienza al tiempo t en el estado $\mathbf{X}$(t)=x de la población celular, luego se tienen 3 etapas: *a)* generar el tiempo τ_i (variable aleatoria) hasta la siguiente reacción $R_i \epsilon [R_1, ..., R_M]$ donde i=índice de la reacción posterior, τ_i está distribuida exponencialmente con el parámetro w_i; *b)* determinar qué reacción ocurre en ese momento; y *c)* actualizar el tiempo y el estado para reflejar las 2 decisiones anteriores. En cada ejecución se calcula

una sola trayectoria y el proceso debe repetirse numerosas veces para conseguir varios datos estadísticos de interés. Podemos definir dos nuevas variables aleatorias:

$$\tau = \min_{\text{i}} \tau_i (\text{tiempo para la siguiente reacción})$$
$$R = \arg\min_{i} \tau_i (\text{índice de la siguiente reacción})$$

Se observa que τ tiene distribución exponencial con el parámetro $\sum_i w_i$ y que R tiene una distribución discreta $P(R = k) = \frac{w_k}{\sum_i w_i}$.

2.4. Aproximación de ruido lineal

La *aproximación de ruido lineal* (LNA) es un método que analiza el efecto del ruido considerando que las trayectorias del número de copias (número de moléculas) de cada especie del sistema, pueden aproximarse como la trayectoria determinista más una componente de fluctuación estocástica alrededor de la misma. Primero, la concentración total de la i-ésima especie $\mathbf{X}_i(t)$ está dividida en una concentración macroscópica $x_i(t)$ y una fluctuación microscópica $\alpha_i(t)$ que se escala inversamente a la raíz cuadrada del tamaño del sistema Ω. El número total de moléculas (copias) de la especie i-ésima $n_i(t)$ viene dado por [10]:

$$n_i(t) = \Omega X_i(t) = \Omega x_i(t) + \Omega^{1/2} \alpha_i(t) \qquad (7)$$

Segundo, se trata de aproximar el espacio de estados discreto de la ecuación 7 a uno continuo. El efecto del ruido en el espacio discreto se modela mediante el operador de paso E_i^k que incrementa en la i-ésima especie por un entero k y que actúa dentro de una función *f*,

$$E_i^k f(n_i, n_j) = f(n_i + k, n_j) \qquad (8)$$

Este cambio es imperceptible en escala macroscópica, por lo que para un sistema grande la acción del operador E_i^k se describe usando la serie de Taylor:

$$\begin{aligned} E_i^k f(n_i, n_j) &= f(n_i + k, n_j) = f(\Omega x_i + \Omega^{1/2}(\alpha_i + \Omega^{-1/2} k), n_j) \\ E_i^k f(n_i, n_j) &\approx \left[1 + \Omega^{-1/2} k \frac{\partial}{\partial \alpha_i} + \frac{k^2 \Omega^{-1}}{2} \frac{\partial^2}{\partial \alpha_i^2} + \dots \right] f(n_i, n_j) \end{aligned} \qquad (9)$$

La CME puede ser rescrita en términos del operador E_i^k y la ecuación 9 amplía la ecuación en potencias inversas. El orden más bajo devuelve las ecuaciones de velocidad macroscópicas, y el siguiente orden es una ecuación lineal que caracteriza las fluctuaciones.

2.5. Representación de modelos biológicos mediante redes de Petri

Las Redes de Petri son ideales para modelar gráficamente el funcionamiento de un sistema biológico y obtener la secuencia de sus reacciones químicas. Estas redes poseen tres elementos fundamentales como se muestra en la figura 1: un lugar P que representa una especie microcelular del modelo, la transición T que es la reacción individual y los arcos que llegan a una transición denotan reactivos, mientras que los arcos que salen de cada transición equivalen a productos [14]. Estos mismos arcos tienen pesos o números asociados (arco no numerado tiene peso de 1) que demuestran las estequiometrías de las reacciones. Dentro de cada P se tienen números enteros o *tokens* como la cantidad de moléculas de una especie que se tienen en un instante de tiempo. Una transición puede solo activarse o dispararse si hay los suficientes *tokens* asociados a cada entrada de un lugar, es decir, los reactivosnecesarios para cada especie. Las redes de Petri son de naturaleza discreta por lo que favorecen la comprensión de la cinética molecular estocástica de los modelos biológicos.

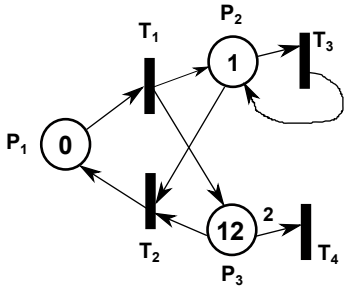


Figura 1: Esquema de una red Petri numerada y etiquetada con tokens de cada especie para empezar la activación de cada transición

3. Circuito propuesto

3.1. Descripción del sistema

El circuito analizado en este trabajo, combina dos redes de genes previamente implementadas en la bacteria *E.coli*: un sistema de comunicación célula a célula [4] y un promotor sintético represivo [1], como se observa en la figura 2. La comunicación entre células utiliza componentes tomadas del *quorum sensing* de V. fischeri [5], [9]. El circuito de realimentación comprende al gen luxI bajo el

control del promotor PluxR, a su vez, la proteína PluxI generada por el gen luxI sintetiza la proteína autoinductora AHL. AHL a su vez, puede unirse a la proteína LuxR y formar un complejo que al asociarse con el promotor PluxR reprime la expresión del gen luxI. El sistema tiene por lo tanto, un bucle de realimentación negativa entre la concentración de AHL intracelular y la expresión del gen luxI [12].

La molécula AHL puede secretarse al exterior de la célula, así como absorberse desde el exterior. Este proceso de difusión en ambas direcciones depende de la concentración intracelular de AHL, y de la extracelular. Esta última depende a su vez, del tamaño de la población celular. Este mecanismo constituye la base del proceso de comunicación entre células.

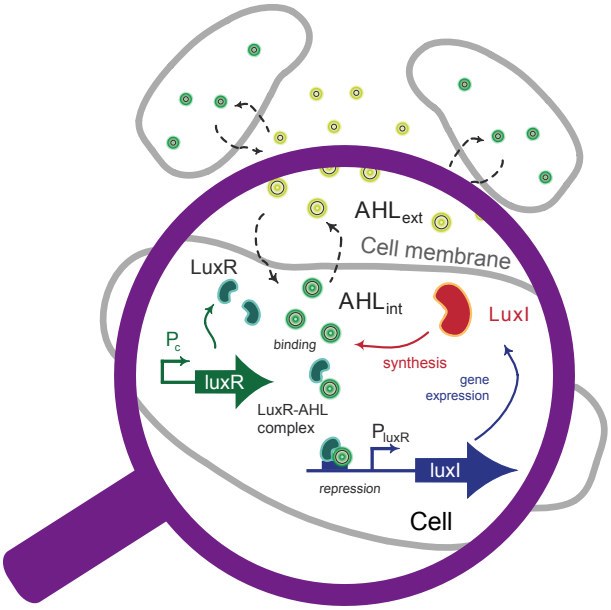


Figura 2: Esquema de la realimentación intracelular del circuito de control genético y mecanismo de comunicación entre células de la bacteria *E.coli*.

3.2. Modelo completo

El modelo completo de los procesos celulares intervinientes incluye aspectos como la generación del complejo LuxR-AHL, y la posterior dimerización del mismo, dando lugar al dímero $(LuxR \cdot AHL)_2$. Es este dímero el que realmente actúa como factor de transcripción, inhibiendo la activación del gen luxI. En esta sección, siguiendo pasos análogos a los de [13] se plantea el modelo completo. Este modelo será utilizado para realizar las simulaciones estocásticas mediante el algoritmo de Gillespie. En las secciones siguientes veremos que el modelo puede ser reducido, y que una aproximación basada en la LNA da resultados buenos.

Una molécula de AHL interna se une a otra de LuxR para formar el compuesto LuxR·AHL, este en contacto con otra molécula del mismo tipo forman el complejo $(LuxR \cdot AHL)_2$ dentro del proceso conocido como dimerización y que en nuestro caso representa el factor de transcripción C_2. Las moléculas de AHL se difunden dentro y fuera de la célula, contribuyendo al incremento de la concentración de AHL externa. Finalmente se incluyen la duplicación del ADN y las degradaciones:

$$\mathrm{ADN} \xrightarrow{K_I} \mathrm{ADN} + \mathrm{ARNm_{luxI}}$$

$$\xrightarrow{const_R} \mathrm{ARNm_{luxR}}$$

$$\mathrm{ADN \cdot (LuxR \cdot AHL)_2} \xrightarrow{\alpha_I K_I} \mathrm{ADN \cdot (LuxR \cdot AHL)_2 + ARNm_{luxI}}$$

$$\mathrm{ARNm_{luxI}} \xrightarrow{p_I} \mathrm{ARNm_{luxI} + LuxI}$$

$$\mathrm{ARNm_{luxR}} \xrightarrow{p_R} \mathrm{ARNm_{luxR} + LuxR}$$

$$\mathrm{LuxI} \xrightarrow{k_A} \mathrm{AHL + LuxI}$$

$$\mathrm{LuxR + AHL} \underset{k_1}{\overset{k_1/K_{d1}}{\longleftrightarrow}} \mathrm{LuxR \cdot AHL}$$

$$\mathrm{2(LuxR \cdot AHL)} \underset{k_2}{\overset{k_2/K_{d2}}{\longleftrightarrow}} \mathrm{(LuxR \cdot AHL)_2}$$

$$\mathrm{(LuxR \cdot AHL)_2 + ADN} \underset{k_1 ux}{\overset{k_{lux}/K_{dlux}}{\longleftrightarrow}} \mathrm{ADN \cdot (LuxR \cdot AHL)_2}$$

$$\mathrm{AHL} \underset{rD}{\overset{D}{\longleftrightarrow}} \mathrm{AHL_{ext}}$$

$$\mathrm{AHL} \xrightarrow{d_A} \emptyset$$

$$\mathrm{AHL_{ext}} \xrightarrow{d_A} \emptyset$$

$$\mathrm{ARNm_{luxI}} \xrightarrow{dm_I} \emptyset$$

$$\mathrm{ARNm_{luxR}} \xrightarrow{dm_R} \emptyset$$

$$\mathrm{LuxI} \xrightarrow{d_I} \emptyset$$

$$\mathrm{LuxR} \xrightarrow{d_R} \emptyset$$

$$\mathrm{LuxR \cdot AHL} \xrightarrow{d_C} \emptyset$$

$$\mathrm{(LuxR \cdot AHL)_2} \xrightarrow{d_{C_2}} \emptyset$$

$$\mathrm{ADN} \xrightarrow{\ln(2)/\tau} \mathrm{ADN + ADN}$$

$$\mathrm{ADN \cdot (LuxR \cdot AHL)_2} \xrightarrow{\ln(2)/\tau} \mathrm{ADN \cdot (LuxR \cdot AHL)_2 + ADN}$$

El modelo anterior puede escribirse conforme a las ecuaciones de masa:

$$\dot{x}_1^i = p_I x_3^i - \left(\frac{ln(2)}{\tau} + d_I\right) x_1^i \qquad (10)$$

$$\dot{x_2^i} = -\frac{k_1}{K_{d1}} x_9^i x_2^i + k_1 x_5^i + p_R x_4^i - \left(\frac{ln(2)}{\tau} + d_R\right) x_2^i \quad (11)$$

$$\dot{x_3^i} = k_I x_7^i + \alpha_I k_I x_8^i - \left(\frac{ln(2)}{\tau} + dm_I\right) x_3^i \quad (12)$$

$$\dot{x_4^i} = const_R - \left(\frac{ln(2)}{\tau} + dm_R\right) x_4^i \quad (13)$$

$$\dot{x_5^i} = \frac{k_1}{K_{d1}} x_9^i x_2^i + 2k_2 x_6^i + \left(-k_1 - 2\frac{k_2}{K_{d2}} x_5^i - \frac{ln(2)}{\tau} - dC\right) x_5^i \quad (14)$$

$$\dot{x_6^i} = \frac{k_2}{K_{d2}} {x_5^i}^2 + K_{lux} x_8^i + \left(-k_2 - \frac{k_{lux}}{K_{dlux}} x_7^i - \frac{ln(2)}{\tau} + dC_2\right) x_6^i \quad (15)$$

$$\dot{x_7^i} = -\frac{k_{lux}}{K_{dlux}} x_6^i x_7^i + \left(k_{lux} x_7^i + \frac{ln(2)}{\tau}\right) x_8^i \quad (16)$$

$$\dot{x_8^i} = \frac{k_{lux}}{K_{dlux}} x_6^i x_7^i - \left(k_{lux} x_7^i + \frac{ln(2)}{\tau}\right) x_8^i \quad (17)$$

$$\dot{x_9^i} = k_1 x_5^i + k_A x_1^i + D x_{10}^i - \left(\frac{k_1}{K_{d1}} x_2^i + \frac{ln(2)}{\tau} + d_A - D\right) x_9^i \quad (18)$$

$$\dot{x_{10}^i} = \frac{D}{N} \sum_{i=1}^{N} (x_9^i - x_{10}^i) - d_A x_{10}^i \quad (19)$$

donde: x_1^i es la concentración de LuxI, x_2^i es la concentración de LuxR, x_3^i es la concentración de ARNm_{LuxI}, x_4^i es la concentración de ARNm_{LuxR}, x_5^i es la concentración de $LuxR \cdot AHL$, x_6^i es la concentración de $(LuxR \cdot AHL)_2$, x_7^i es la concentración de ADN, x_8^i es la concentración de ADN$(LuxR \cdot AHL)_2$, x_9^i es la concentración de AHL y x_{10}^i es la concentración de AHL externa. En la tabla 1 se resumen los valores de los parámetros utilizados para la simulación del modelo completo.

3.3. Modelo reducido

Para la *i-ésima* célula y considerando únicamente las concentraciones de LuxI y AHL tanto interna como externa, puede obtenerse el siguiente conjunto de ecuaciones:

$$\dot{x_1^i} = K_0 + K_1 \frac{\theta_h^n}{\theta_h^n + {x_2^i}^n} - \gamma_1 x_1^i \quad (20)$$

$$\dot{x_2^i} = K_2 x_1^i - d(x_2^i - x_e) - \gamma_2 x_2^i \quad (21)$$

$$\dot{x_e} = \frac{d_e}{N} \sum_{i=1}^{N} (x_2^i - x_e) - \gamma_e x_e^i \quad (22)$$

donde x_1^i es la concentración de proteína LuxI en la i-ésima célula, x_2^i es la concentración de AHL interna en la i-ésima célula y x_e es la concentración de AHL externa.

Cuadro 1: Paramétros usados en simulaciones determinísticas y estocásticas

Parámetro	Descripción	Valor	Unidad
τ	Tiempo de duplicación	45	min
k_1	Tasa de separación de LuxR a AHL	10	min^{-1}
k_2	Tasa de disociación de $(LuxR \cdot AHL)_2$	1	min^{-1}
k_A	Tasa de síntesis de AHL por LuxI	0.04	min^{-1}
K_{Lux}	Tasa de disociación de $(LuxR \cdot AHL)_2$ a PluxR	10	min^{-1}
α_I	Tasa de relación entre genes no-activos y activos para expresión de luxI	0.01	min^{-1}
D	Tasa de difusión de AHL a través de membrana celular	0.06	min^{-1}
k_{d1}	Constante de disociación de LuxR a AHL de	100	nM
k_{d2}	Constante de disociación del dímero a $(LuxR{\cdot}AHL)$	20	nM
k_{dlux}	Constante de disociación de $(LuxR \cdot AHL)$ al promotor	200	nM
dm_I	Tasa de degradación de ARNm_{LuxI}	0.347	min^{-1}
dm_R	Tasa de degradación de ARNm_{LuxR}	0.347	min^{-1}
d_I	Tasa de degradación de LuxI	0.01	min^{-1}
d_R	Tasa de degradación de LuxR	0.002	min^{-1}
d_A	Tasa de degradación de AHL	0.001	min^{-1}
dA_e	Tasa de degradación de AHL externa	0.001	min^{-1}
d_C	Tasa de degradación de $(LuxR \cdot AHL)$	0.002	min^{-1}
d_{C2}	Tasa de degradación de $(LuxR \cdot AHL)_2$	0.002	min^{-1}
p_R	Tasa de traslación de ARNm_{LuxR}	6.94	min^{-1}
p_I	Tasa de traslación de ARNm_{LuxI}	6.94	min^{-1}
$const_R$	Tasa constituiva de transcripción de LuxR	10	min^{-1}
k_I	Tasa de transcripción de LuxI	2.5	min^{-1}

En la ecuación 20, el valor de K_0 es la expresión basal del promotor y K_1 es el rango dinámico del mismo. La regulación adopta la forma de una función de Hill represora, donde n es el coeficiente de Hill y θ es la mitad de la constante de concentración o umbral de represión. En la ecuación 21, la constante cinética K_2 modela la síntesis de primer orden de AHL, d es la constante de transporte interno y tanto LuxI como AHL están sujetas a un proceso de degradación siendo γ_1 y γ_2 sus respectivas tasas. Para la ecuación 22, la constante cinética d_e es la constante de transporte externo, γ_e es la tasa de degradación de AHL externa y finalmente se toma en cuenta la diferencia entre volúmenes de AHL interna y externa hallando su valor promedio cuando N es la población celular, como en [8].

En el modelo (20-22) se realizaron las siguientes aproximaciones: a)se asumió que la expresión del ARNm está en estado cuasi-estacionario (QSSA) dejando de lado el transitorio necesario para que la

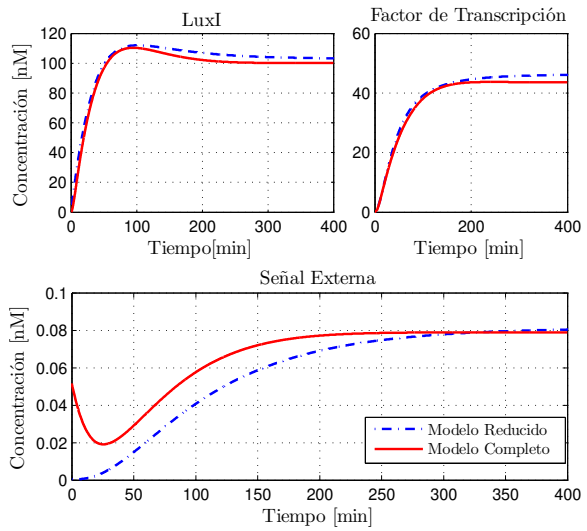


Figura 3: Comparativa del modelo completo y del modelo reducido. Arriba a la izquierda podemos ver las concentraciones de la proteína LuxI. Arriba a la derecha, el factor de transcripción ($LuxR \cdot AHL$ o AHL). Abajo la concentración de la señal de comunicación entre células (AHL externa).

concentración del ARNm alcance su estado estacionario, b) se consideró que el complejo ADN-Represor también alcanza rápidamente su valor de estado estacionario lo que permite modelar la represión como una función de Hill, c) no se modeló explícitamente la dimerización de la proteína LuxR al unirse con AHL interna, y d) se asumió que el gen luxR está expresado constitutivamente (siempre presente) y no es un factor limitante dentro del proceso [12].

4. Simulación

Utilizando ambos modelos presentados en la sección anterior se realizaron simulaciones a partir de las ODEs. En la Figura 3 de muestran los resultados comparativos de ambas simulaciones. Arriba a la izquierda podemos ver las concentraciones de la proteína LuxI, con una buena coincidencia. Por otro lado, arriba a la derecha en la misma figura, se observa la coincidencia entre ambos modelos en la concentración del factor de transcripción. Cabe destacar que en el caso del modelo completo el factor de transcripción es el complejo LuxR-AHL. Sin embargo, en el modelo reducido (ya que hemos asociado todas las dinámicas rápidas quedándonos solo con la lenta representando al factor de transcripción) consideramos que es directamente la concentración de AHL intracelular la que actúa como factor de transcripción. Finalmente en la parte inferior de la Figura 3 se observa la concentración de la señal de comunicación entre células AHL externa tanto para el modelo completo como para el modelo reducido. Ambas son similares y difieren únicamente debido a las condiciones iniciales de la simulación (ver [8] para más detalle).

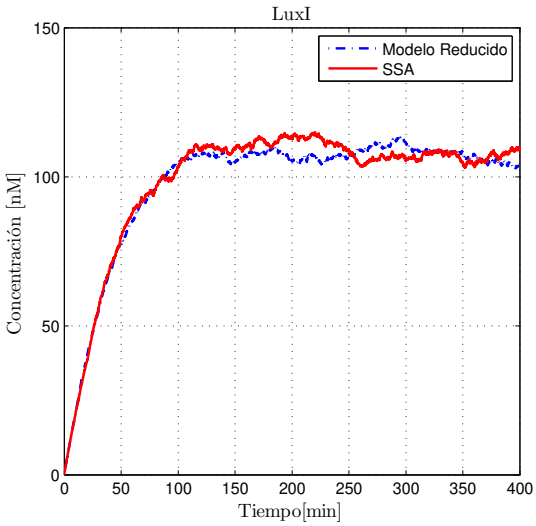


Figura 4: Simulación estocástica de concentración de Proteína LuxI: modelo reducido con ruido blanco y SSA.

Además de las simulaciones deterministas anteriores, se realizó la simulación estocástica a partir del modelo de reacciones bioquímicas de la sección 3.2 mediante el uso del software Copasi. Este software realiza una simulación mediante el algoritmo SSA de tipo Gillespie descrito en la sección 2.3. Estos resultados, luego de convertirlos de número de moléculas a concentraciones teniendo en cuenta el volumen standard de una célula de bacteria, los comparamos con los resultados de simular el modelo reducido con ruido blanco aditivo, cuya varianza hemos calculado a partir de la aproximación lineal de ruido (LNA) [10]. En la Figura 4 se puede ver el resultado para la proteína LuxI, mientras que en la Figura 5, el resultado para AHL, con la misma salvedad que en el modelo de ODEs anterior, a los efectos de realizar la comparación exclusivamente.

La concordancia, ya sea tanto en las simulaciones determinísticas (Fig. 3) como en las estocásticas (Figuras 4 y 5) es muy buena, teniendo en cuenta la gran reducción de complejidad y de coste computacional que se obtiene.

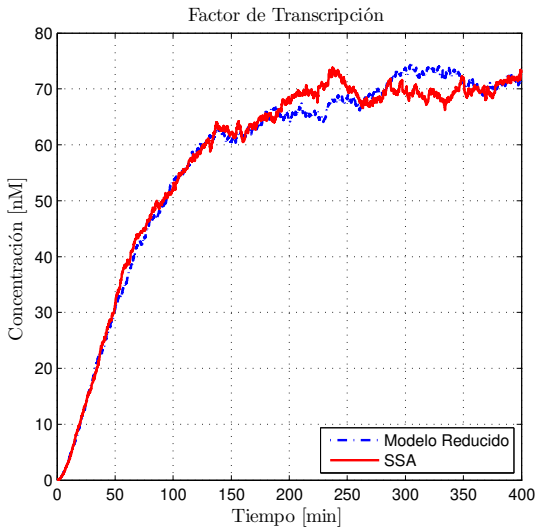


Figura 5: Simulación estocástica de concentración de Factor de Transcripción: modelo reducido con ruido blanco y SSA. Obsérvese que el factor de transcripción corresponde con el dímero $(LuxR \cdot AHL)_2$ en el caso de la simulación estocástica del modelo completo, y con la concentración de AHL intracelular, en el caso del modelo reducido.

5. Conclusiones

En este trabajo, hemos planteado un modelo reducido de los fenómenos de expresión genética y producción de proteínas en poblaciones de microorganismos genéticamente modificados con herramientas de biología sintética. Este planteamiento permite obtener simulaciones fieles del ruido en la expresión genética y de la variabilidad poblacional por este generada. La complejidad computacional del modelo reducido es pequeña comparada con otro tipo de simulaciones como SSA-Gilespie cuyo coste computacional las hace intratables a la hora de simular a una población de microorganismos. Por otro lado permite mas flexibilidad que otras herramientas como son la aproximación lineal de ruido (Linear Noise Approximation) ya que se puede analizar tanto ruido de origen intrínseco (LNA) como extrínseco.

Los resultados obtenidos avalan las aproximaciones hechas para generar el modelo reducido: quasi-estado estacionario del ARN, la velocidad de las reacciones de formación del complejo factor de transcripción con ADN y dimerización de LuxR y formación del complejo LuxR/AHL es mucho mas rápida que la de expresión genética, degradación de proteínas y síntesis de AHL por parte de LuxI. Asimismo se ha supuesto que el gen *luxR* se expresa de forma constitutiva y no es un factor limitante en el proceso de expresión genética.

Como trabajo posterior, se plantea generalizar la metodología y obtener modelos reducidos de distintos circuitos genéticos. Incorporar mas fuentes de ruido extrínseco y obtener la varianza de la fuente de ruido intrínseco a partir de la LNA, o estimarla a partir de algunas ejecuciones de SSA. Asimismo se esta trabajando en la utilización de modelos reducidos del tipo del descrito en este trabajo y en el diseño de circuitos de control genético en poblaciones celulares.

Agradecimientos

Este trabajo es apoyado parcialmente por el Ministerio de Ciencia e Innovación, Gobierno de España (FEDER-CICYT DPI2011-28112-C04-01) y fondos FEDER de la Union Europea. El primer autor agradece el apoyo para la realización de este trabajo a la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT-Ecuador) mediante el programa de Becas en el Exterior 2011, el segundo autor agradece el apoyo de la Universitat Politécnica de València mediante la beca FPI/2009-21.

Referencias

[1] K. A. Egland and E. P. Greenberg. Conversion of the vibrio fischeri transcriptional activator, LuxR, to a repressor. *Journal of Bacteriology*, 182(3):805–811, February 2000.

[2] Michael B. Elowitz, Arnold J. Levine, Eric D. Siggia, and Peter S. Swain. Stochastic Gene Expression in a Single Cell. *Science*, 297(5584):1183–1186, August 2002.

[3] R. Lencastre Fernandes, M. Nierychlo, L. Lundin, A.E. Pedersen, P.E. Puentes Tellez, A. Dutta, M. Carlquist, A. Bolic, D. Schaepper, A.C. Brunetti, S. Helmark, A.-L. Heins, A.D. Jensen, I. Nopens, K. Rottwitt, N. Szita, J.D. van Elsas, P.H. Nielsen, J. Martinussen, S.J. SÃ¸rensen, A.E. Lantz, and K.V. Gernaey. Experimental methods and modeling techniques for description of cell population heterogeneity. *Biotechnology Advances*, 29(6):575 – 599, 2011.

[4] C. Fuqua, M.R. Parsek, and E.P. Greenberg. Regulation of gene expression by cell-to-cell communication: acyl-homoserine lactone

quorum sensing. *Annual review of genetics*, 35(1):439–468, 2001.

[5] HB Kaplan and EP Greenberg. Diffusion of autoinducer is involved in regulation of the vibrio fischeri luminescence system. *Journal of Bacteriology*, 163(3):1210–1214, 1985.

[6] William S. Levine. *Control System Application*. University of Maryland. College Park, MD, USA, second edition edition, 2011.

[7] Mary E Lidstrom and Michael C Konopka. The role of physiological heterogeneity in microbial population behavior. *Nature chemical biology*, 6(10):705–712, 2010.

[8] Petros Mina, Mario di Bernardo, Nigel J Savery, and Krasimira Tsaneva-Atanasova. Modelling emergence of oscillations in communicating bacteria: a structured approach from one to many cells. *Journal of The Royal Society Interface*, 10(78), 2013.

[9] A L Schaefer, D L Val, B L Hanzelka, J E Cronan, and E P Greenberg. Generation of cell-to-cell signals in quorum sensing: acyl homoserine lactone synthase activity of a purified vibrio fischeri luxi protein. *Proceedings of the National Academy of Sciences*, 93(18):9505–9509, 1996.

[10] Matthew Scott and Brian Ingalls. Using the linear noise approximation to characterize molecular noise in reaction pathways. Department of Applied Mathematics, University of Waterloo,Waterloo-Ontario, Canada, 2013.

[11] Guy-Bart Stan. Modelling in biology. Lecture Notes in Modelling in Biology Course 2010-2012, Centre for Synthetic Biology and Innovation, Imperial College London, 2011.

[12] A. Vignoni, D.A. Oyarzún, J.Picó, and G.B. Stan. Control of protein concentrations in heterogeneous cell populations. *European Control Conference*, 2013.

[13] Marc Weber, Javier Buceta, et al. Dynamics of the quorum sensing switch: stochastic and non-stationary effects. *BMC systems biology*, 7(1):6, 2013.

[14] Darren J. Wilkinson. *Stochastic Modelling for Suystems Biology*. Champan and Hall/CRC. Mathematical and computational Biology Series, 2006.

Desarrollo de una clase en Matlab para la simulación realista de robots seriales

R. Morales, A. Sanchez, F. Badesa, N. Garcia-Aracil, C. Perez, J.M. Sabater
Neuroingenieria Biomedica
Universidad Miguel Hernández
email: nicolas.garcia@umh.es

Resumen

En este artículo se presenta una clase de carácter general capaz de modelizar el comportamiento de cualquier tipo de robot serial a partir de un archivo de configuración y de hacer simulaciones precisas y realistas a partir de los modelos stl del robot. Así pues, esta clase no sólo cuenta con potencial docente sino que pretende ser el primer paso hacia un driver general capaz de controlar y modelizar el comportamiento cualquier robot serial desde Matlab. Se presenta en este artículo dos ejemplos de aplicación: uno sobre un robot comercial y el otro sobre un robot diseñado y construido dentro del grupo de neuroingeniería biomédica de la Universidad Miguel Hernández de Elche

Palabras clave: Robótica, simulación, cinemática, dinámica

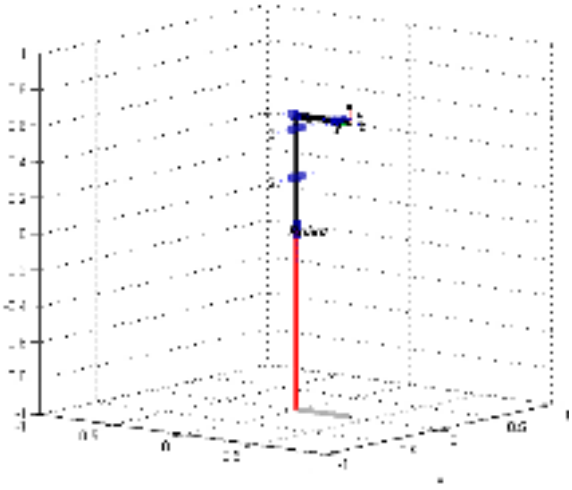


Figura 1: Representación alámbrica de la toolbox creada por Peter Corke.

1 INTRODUCCIÓN

Matlab es una herramienta de software que permite la programación en un lenguaje de alto nivel diseñado para computación numérica, análisis estadístico y visualización de datos. La programación en este tipo de lenguajes computacionales es muy útil cuando es necesario resolver operaciones complejas en un corto espacio de tiempo. Para ello, Matlab incluye gran cantidad de librerías matemáticas y toolboxes para poder ejecutar rápidamente y de forma sencilla gran cantidad de operaciones. Esta serie de ventajas, hacen de Matlab una plataforma ideal para diseñar y monitorizar comportamientos de sistemas en casi cualquier campo de ingeniería, como son la electrónica, la mecánica, las telecomunicaciones, el control o la industria aerospacial.

Para Matlab, existen algunos recursos que ayudan en el modelado y análisis en el campo de la robótica en forma de Toolbox. Tradicionalmente la toolbox más utilizada en robótica ha sido y es la creada por Peter Corke [1]. Otros autores han desarrollado distintas toolboxes para docencia e investigación en el diseño de dispositivos robóticos [2], [3], [4], [5]. En la mayoría de las toolboxes de robótica la visualización está basada en representaciones alámbricas de los robots (Figura **??**). Este tipo de representación es el soporte visual que tiene el ingeniero para apreciar en qué posición está el robot.

Para mejorar esta visualización se han desarrollado diferentes proyectos, tales como el 3D Puma 560 el cual visualiza un GUI con sencillos modelos pre-cargados similares a la forma de las piezas de un robot Puma [6].

Mas adelante se elaboró la IGES toolbox [7], capaz de cargar objetos CAD de tipo IGES en matlab además de poder hacer algunas operaciones como plots transformaciones o proyecciones. Otra herramienta para cargar modelos CAD en Matlab es SimMechanics [8], el cual permite simular entornos 3D para sistemas mecánicos y ejercer control sobre los sistemas usando Simulink. Simbad es un simulador 3D basado en java diseñado con fines educativos para ayudar a investigadores y programadores a estudiar algoritmos de inteligencia artificial y aprendizaje automático. Sin embargo no esta pensado para simulaciones realistas.

Se presenta una clase de carácter general capaz de modelizar el comportamiento de cualquier tipo

de robot serial a partir de un archivo de configuración y de hacer simulaciones precisas y realistas a partir de los modelos stl del robot. Es capaz de cargar los modelos 3D a partir de archivos STL (Stereo Litographic) ascii de fácil lectura por parte del usuario, o bien STL binarios de cortos tiempos de carga. Con distintos archivos de configuración, es posible generar distintos robots como objetos capaces de operar de forma independiente en una misma simulación. Así pues, esta clase no sólo cuenta con potencial docente sino que pretende ser el primer paso hacia un driver general capaz de controlar y modelizar el comportamiento cualquier robot serial desde Matlab. La posibilidad de crear múltiples objetos capaces de coexistir en una misma simulación también abre la puerta a su adaptación para el control de robots paralelos.

2 PROCEDIMIENTO DE USO

La clase es capaz de representar cualquier robot serial de hasta 7 grados de libertad. Con articulaciones rotacionales o prismáticas. El procedimiento para el correcto funcionamiento de la clase está dividido en cuatro pasos:

- La exportación de modelos CAD a archivos STL.
- La personalización del archivo de configuración.
- El cálculo de parámetros mediante las funciones de análisis.
- Funciones de visualización y ejemplo de guide.

2.1 Crear STLs desde Autodesk Inventor

El primer paso es exportar los modelos cad del robot a archivos STL. En este artículo se ha usado Autodesk Inventor para exportar las piezas, aunque el proceso es análogo para cualquier herramienta de modelado cad. Es necesario situar el robot completo en la posición de reposo sobre la que se habrá realizado el algoritmo de Denavit-Hartenberg asegurándose que la base se encuentra sobre el origen de coordenadas. A continuación se exporta cada pieza en un archivo STL individual. De esta manera las piezas ya se encuentran posicionadas respecto de sistema de coordenadas de la base. Si el robot no se encontrase sobre el sistema de coordenadas de la base, sería necesario posicionarlo posteriormente usando las funciones de desplazamiento que provee la clase.

Los archivos podrán ser exportados en modalidad STL ascii o STL binario. Para exportar cada archivo en autodesk inventor, es necesario entrar en File− >Export− >Cad model. En la ventana de exportar, se indica el nombre y tipo del archivo que se va a crear, en este caso STL. Antes de continuar, si se pulsa sobre el botón de opciones, se indica el tipo de STL que se va a crear y la resolución del nuevo modelo. Se recomienda usar una calidad del modelo baja, ya que el archivo STL sólo se usa para la visualización, no para la realización de los cálculos. Una calidad de modelo baja resulta en una imagen con menor número de triángulos, lo cual reducirá tanto el tiempo de carga de los modelos como la velocidad de refresco de la imagen en la simulación.

La clase cargará los archivos STL que se le indiquen independientemente de la modalidad en la que estén exportados, pero la carga de archivos binarios es notablemente más rápida. La razón de la inclusión de la posibilidad de carga de archivos ascii es que en este tipo de archivos la información de los polígonos está estructurada de manera muy clara. En ocasiones puede resultar interesante usar este tipo de archivos si el usuario desea tener la posibilidad de editar a mano alguno de los archivos.

La carpeta con los archivos STL se podrá ubicar donde se desee, el path se indicará en un campo del el archivo de configuración.

2.2 Archivo de configuración

El archivo de configuración es una función que crea una estructura de datos. Esta estructura se pasa como parámetro al robot en el momento de creación del objeto. De este modo el constructor asigna los valores a los parámetros del robot que se va a crear. En una sola ventana es posible simular múltiples robots. Si se desea simular robots de distintos modelos únicamente habrá tener en cuanta que en el momento de la creación del objeto robot, habrá que pasar como parámetro diferentes archivos de configuración.

Los parámetros contenidos en la estructura de datos del archivo de configuración están divididos en varios grupos.

2.2.1 Parámetros generales

Los parámetros generales englobarán datos como el nombre del robot, la ruta de la clase y de los archivos STL.

≫ conf.nameRobot = 'robotName';
≫ conf.path = pwd;
≫ conf.stlpath = <STL files path>;
≫ conf.dof = <Degrees of freedom>;

2.2.2 Cinemática

La cinemática estará definida por los grados de libertad del robot, la matriz con los valores theta, d, a y alpha extraídos tras la aplicación del algoritmo de Denavit-Hartenberg, y por el vector que indica el carácter rotacional o prismático de cada articulación.

≫ conf.parameters.origen.Tr = eye(4);
≫ conf.parameters.theta = <theta vector>;
≫ conf.parameters.d = <d vector>;
≫ conf.parameters.a = <a vector>;
≫ conf.parameters.alpha = <alpha vector>;
≫ conf.parameters.jointType = <joints vector>;

2.2.3 Dinámica

La dinámica del robot la definen los vectores de las dinámicas así como las matriz jacobiana de cada articulación.

≫ conf.dynamics.m = <dynamics vector>;
≫ conf.dynamics.cg{1} = <dynamic vector>;
≫ ... =
≫ conf.dynamics.cg{n} = <dynamic vector>;
≫ conf.dynamics.J{1} = <Jacobian matrix>
≫ ... =
≫ conf.dynamics.J{n} = <Jacobian matrix>
≫ conf.real.realdev = [];

2.2.4 STL

En el grupo de parámetros STL se encuentra la información que indica los modelos STL componen cada articulación, así como parámetros para su visualización tales como el color de representación.

≫ conf.tmp.stl.base.Link=<linkNumber> ;
≫ conf.tmp.stl.base.robot=[];
≫ conf.tmp.stl.base.element1.name='<nameStlFile>';
≫ conf.tmp.stl.base.element1.color=<color in [r g b] format>;

2.2.5 Vista

En la última parte se configuran los valores del espacio de visualización tales como los límites de los ejes que se comprenden en la vista y las coordenadas de posición de la cámara.

≫ conf.parameters.cameraAZ = ¡azimuth>;
≫ conf.parameters.cameraEL = ¡elevation>;
≫ conf.parameters.minX = <minimum value axis X>;
≫ conf.parameters.maxX = <maximum value axis X>;
≫ conf.parameters.minY = <minimum value axis Y>;
≫ conf.parameters.maxY = <maximum value axis Y>;
≫ conf.parameters.minZ = <minimum value axis Z>;
≫ conf.parameters.maxZ = <maximum value axis Z>;

2.3 Creación y funciones de uso

Para crear objetos de la clase robot se llama a la función pasando como parámetro el archivo de configuración:
≫ MyRobot = Robot(config()); Una vez esta creado el robot se pueden usar todas sus funciones llamando a los métodos.

Dentro de la clase se ha integrado un amplio conjunto de funciones de la toolbox "A Robotics Toolbox for Matlab" creada por Peter Corke [1].

2.4 Funciones de visualización

Una vez creado el robot, se puede visualizar con la función: - showRobot

Para cambiar la visualización se puede usar la función configView la cual puede ajustar los parámetros máximos y mínimos en la visualización de los ejes y el angulo de visualización mediante los parámetros de azimut o rotación horizontal y la elevación vertical (ambos en grados sexagesimales)
≫ configView(minX, maxX, minY, maxY, minZ, maxZ, AZ, EL);
≫ configView(minX, maxX, minY, maxY, minZ, maxZ);
≫ configView(AZ, EL);

plot(anglesVector)

Para realizar la visualización de un movimiento del robot se puede usar la función showMovementRobot (anglesVector, step) donde anglesVector es un vector que contiene los ángulos hacia los que va el robot y step es el la variación entre angulos para cada imagen.

transformEnvironment(name, Tr)

2.4.1 Guide

Puede ser interesante que el usuario se cree un GUI para modificar la posición del robot de manera visual. A continuación se muestran los pasos generales en el procedimiento de creación de un GUI para su uso con un robot de la clase. Los pasos aplicados se muestran en los ejemplos de aplicación de la sección 3 y 4. Al escribir en la consola el comando "guide" aparecerá la ventana de edición de guides. Si guardamos el archivo como

guideRobot.m ya tenemos el Sample del guide listo para ser editado.

En la ventana de edición del guide, añadir un slide por cada una de las articulaciones del robot. Pulsar sobre la imagen de cada slide con el segundo botón del ratón para abrir las properties. Renombrar los campos de los objetos de tipo slider para que sean identificables como las articulaciones de los grados de libertad de nuestro robot. Editar sus campos Max y Min con el giro máximo y mínimo de cada articulación.

En el archivo guideRobot.m, en el callback de cada articulación escribimos el siguiente código, por ejemplo, para la primera:
≫ handles.q(<NumArticulación>) = get(hObject,'Value');
≫ handles.MyRobot.plot(handles.q)
≫ guidata(hObject, handles);

Dentro del archivo guideRobot.m, En la creación del guide, (funcion robotGuide_OpeningFcn) añadir la asociación del guide con el objeto de nuestra clase.
≫ handles.MyRobot = Robot(configIRB());
≫ handles.q = zeros(1,handles.MyRobot.dof);

De este modo, en la ventana de edición de guides se van añadiendo todos los elementos que se desee y en el archivo guideRobot se escribe el código que se desea ejecutar.

3 EJEMPLO DE APLICAIÓN: ABB IRB120

El primero de los dos ejemplos con los que se ilustra la aplicación de la clase se refiere al robot comercial IRB120 de ABB de 6 grados de libertad (Figura 2) . En este ejemplo se exportan los modelos, se completa el archivo de configuración con los parámetros del ABB IRB 120 y se usan algunas funciones para ejemplificar el uso. Finalmente se indica como crear un guide para simular el moviemiento desde la misma ventana.

3.1 Exportación de los archivos STL

Desde Inventor se cargan los archivos step descargados de la pagina web de abb. Se colocan las piezas de manera que el robot quede en la posición inicial sobre al cual se habrá de realizar el algoritmo de Denavit-Hartenberg. Con la base sobre el eje del sistema de referencia, se exportan los links por separado para que cada pieza quede en un archivo STL individual. Para exportar los modelos, seleccionamos File− >Export− >Cad model y en la ventana de exportar elegimos el tipo STL. Debajo del campo de selección de tipo de archivo

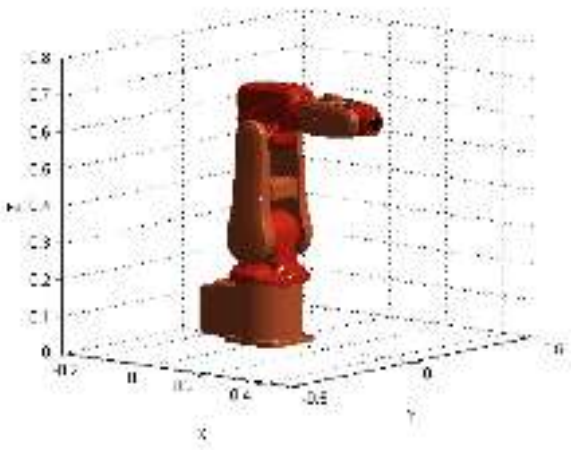


Figura 2: Aplicación de la clase al robot comercial IRB120 de ABB de 6 grados de libertad.

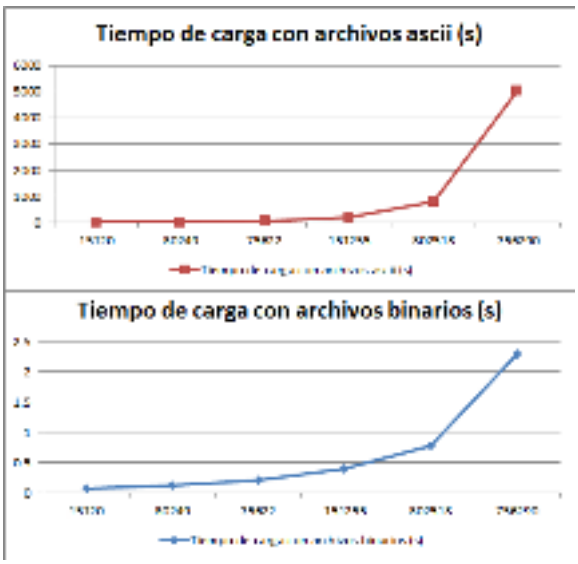


Figura 3: Tiempo de carga de archivos STL ascii y binarios

CAD, pulsamos sobre el botón opciones y elegimos formato binario, unidades metros y baja resolución. Pulsando sobre el botón "preview" se puede observar si la resolución nos resulta suficiente. Se han realizado simulaciones para comparar la diferencia de tiempos de cargas entre archivos STL ascii y STL binarios. En la Figura 3 se observa que los tiempos de carga para archivos binarios son extremadamente inferiores al de los archivos ascii. La diferencia aumenta conforme sube el numero de triángulos que conforman la representación de los modelos. En este caso resultan 7 archivos: la base, cinco eslabones y el efector. Todos los archivos se guardan en una carpeta llamada stl la cual se coloca dentro de la clase. Como se muestra en el apartado siguiente, la ubicación de los archivos se indicara en el archivo de

configuración, en el parámetro stlpath.

3.2 IRB120: Archivo de configuración

Se crea el archivo de configuración ConfigIRB.m con la estructura de datos que indican el comportamiento del robot y la información para acceder a los archivos con los modelos STL para su representación en 3 dimensiones.

3.2.1 Parámetros generales

conf.nameRobot = 'IRB120';
conf.path = pwd;
conf.stlpath = [pwd '
stl IRB];

3.2.2 Cinemática y Dinámica

conf.parameters.dof = 6;
conf.parameters.origen.Tr = eye(4);
conf.parameters.theta = [0 -pi/2 0 0 0 0];
conf.parameters.d = [0.29 0 0 0.30 0 0.072];
conf.parameters.a = [0 0.27 0.07 0 0 0];
conf.parameters.alpha = [-pi/2 0 -pi/2 -pi/2 pi/2 0];
conf.parameters.jointType = ['r' 'r' 'r' 'r' 'r' 'r'];

De una forma similar se configurarían los parámetros dinámicos del robot.

3.2.3 Información sobre los archivos STL

Informacion acerca de la base, la cual tiene un elemento cuyo archivos stl es RB120_base.stl:

conf.tmp.stl.base.Link=0;
conf.tmp.stl.base.robot=[];
conf.tmp.stl.base.element1.name='IRB120_base';
conf.tmp.stl.base.element1.color=[0.8922 0.3922 0.2118];

Informacion acerca del primer eslabon:
conf.tmp.stl.link1.Link=1;
conf.tmp.stl.link1.robot=[];
conf.tmp.stl.link1.element1.name='IRB120_1';
conf.tmp.stl.link1.element1.color=[0.8922 0.1622 0.1018];
...
Y así sucesivamente con el resto de eslabones

3.2.4 Configuración de la visualización

conf.parameters.cameraAZ = 40;
conf.parameters.cameraEL = 10;
conf.parameters.minX = -0.4;
conf.parameters.maxX = 1.0;
conf.parameters.minY = -0.8;
conf.parameters.maxY = 1.8;
conf.parameters.minZ = -0.4;
conf.parameters.maxZ = 1.0;

3.3 Creación del Robot y funciones de visualización

Para crear objetos de la clase robot se llama a la función pasando como parámetro el archivo de configuración: MyIRB = Robot(configIRB()); Una vez esta creado el robot se pueden usar todas sus funciones llamando a los métodos. Dentro de la clase se ha integrado un amplio conjunto de funciones de la toolbox "A Robotics Toolbox for Matlab" [1] .

Una vez creado el robot, se puede visualizar con la función: - showRobot

Para cambiar la visualización se puede usar la función configView la cual puede ajustar los parámetros máximos y mínimos en la visualización de los ejes y el ángulo de visualización mediante los parámetros de acimut o rotación horizontal y elevación vertical (ambos en grados sexagesimales)
- configView(minX, maxX, minY, maxY, minZ, maxZ, AZ, EL);
- configView(minX, maxX, minY, maxY, minZ, maxZ);
- configView(AZ, EL);

la función plot(anglesVector) colocará el robot en la posición deseada. Si no hay posición previamente definida, el robot será mostrado en la posición original de exportación.

Para realizar la visualización de un movimiento del robot se puede usar la función showMovementRobot (anglesVector, step) donde anglesVector es un vector que contiene los ángulos hacia los que va el robot y step es el la variación entre ángulos para cada imagen.

transformEnvironment(name, Tr) will translate and rotate the object depending the Tr.

Puede ser interesante que el usuario se cree un guide para modificar la posición del robot de manera visual. Los pasos en el procedimiento de creación de un guide para su uso con un robot de la clase será el siguiente (Figura 4):

3.3.1 Guide

A continuación se indican los pasos para crearse un guide por si el usuario desea controlar la posición del robot de manera gráfica.

Los pasos en el procedimiento de creación del guide serán los siguientes:

Con el comando $\gg$ guide abrimos la ventana de edición de guides y guardamos el archivo como

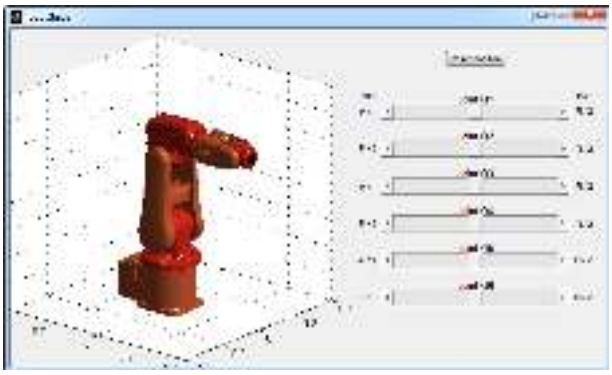


Figura 4: Detalle del Guide creado para el IRB120

guideIRB.m.

En la ventana de edición del guide y se añaden 6 slides, uno por cada una de las articulaciones del robot. Pulsando sobre la imagen de cada slide con el segundo botón del ratón se abren las properties. Se renombran los campos de los objetos de tipo slide, desde slide_Q1 para la primera articulación, hasta slide_Q6 para la última. También puede ser buena idea editar sus campos Max y Min con el giro máximo y mínimo de cada articulación.

Dentro de el archivo guideRobot.m, en el callback de cada una de las seis articulaciones es necesario escribir unas líneas de código. A continuación se muestran las correspondientes al slide_Q1:

```
handles.q(1) = get(hObject,'Value');
handles.MyRobot.plot(handles.q) guidata(hObject, handles);
```

Nótese que la única diferencia de código entre los distintos slides será la referencia al índice del vector q.

En la función de apertura del guide, (funcion robotGuide_OpeningFcn) se añade la asociación del guide con el objeto de nuestra clase:

```
handles.MyRobot = Robot(configIRB());
handles.q = zeros(1,handles.MyRobot.dof );
```

De este modo, en la ventana de edición de guides se van añadiendo todos los elementos que se desee y en el archivo guideRobot se escribe el código que se desea ejecutar.

4 EJEMPLO DE APLICAIÓN: Robot MAAT

En este segundo ejemplo se ilustrara la aplicación de la clase al robot Maat de 7 grados de libertad desarrollado por el grupos e investigación de Neuroingeniería Biomédica (Figura 5). En este ejemplo se exportaran los modelos, se completará el archivo de configuración con los parámetros del Maat, se usaran algunas funciones para ejemplificar el uso. Finalmente se indicara como crear un guide para simular el movimiento desde la misma ventana.

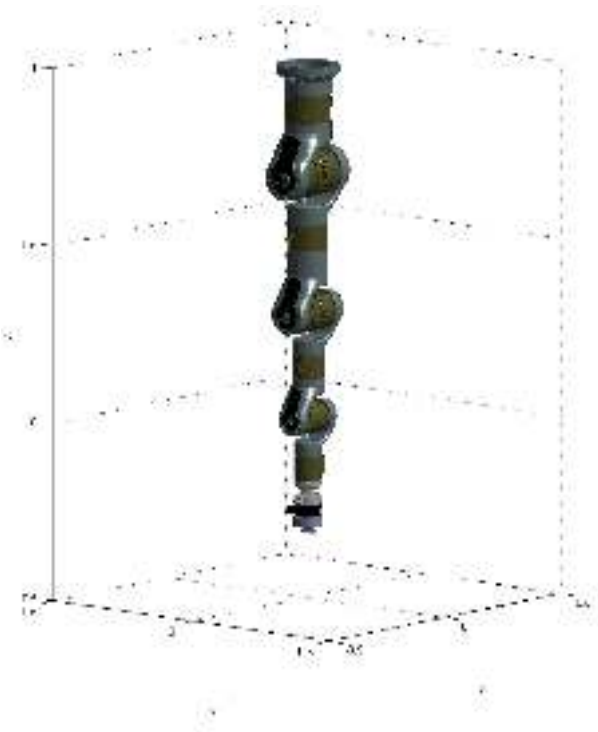


Figura 5: Aplicación de la clase al robot MAAT de 7 grados de libertad.

4.1 Exportación de los archivos STL

La exportación de los archivos STL se hace de forma análoga a la descrita para el ejemplo de aplicación anterior.Como en este caso la base está compuesta por dos piezas, se exportan en dos archivos distintos llamados
baseSchunk1.stl y baseSchunk2.stl.

Asimismo cada eslabón esta compuesto por cinco piezas, el motor, el cuerpo de la articulación, la tapa lateral, el modelo de las letras Nbio y la pieza de enlace con el motor del siguiente eslabón. El efector está a su vez compuesto por cuatro piezas, al final de las cuales está planeado colocar una mano robótica. De manera que en este caso se exportan 36 archivos stl.

4.2 MAAT: Archivo de configuración

Los parámetros usados para el maat se muestran a continuación.

4.2.1 General paramenters

```
conf.nameRobot = 'Maat';
conf.path = pwd;
conf.stlpath = [pwd '
stl schunk];
```

4.2.2 Cinemática y Dinámica

```
conf.parameters.dof = 7;
conf.parameters.origen.Tr = eye(4);
conf.parameters.theta = [ 0 0 0 0 0 0 0];
conf.parameters.d = [-0.306 0 -0.391 0 -0.2975 0
-0.3225];
conf.parameters.a = [ 0 0 0 0 0 0 0];
conf.parameters.alpha = [ pi/2 -pi/2 pi/2 -pi/2
pi/2 -pi/2 pi];
conf.parameters.jointType = ['r' 'r' 'r' 'r' 'r' 'r' 'r'];
```

De una forma similar se configurarían los parámetros dinámicos del robot.

4.2.3 Información sobre los archivos STL

Imformación acerca de la base, la cual tiene dos elementos cuyos archivos stl son BaseSchunk1.stl y BaseSchunk2.stl:

```
conf.tmp.stl.base.Link=0;
conf.tmp.stl.base.robot=[];
conf.tmp.stl.base.element1.name='BaseSchunk1';
conf.tmp.stl.base.element1.color=[0.471 0.545
0.564];
conf.tmp.stl.base.element2.name='BaseSchunk2';
conf.tmp.stl.base.element2.color=[0.392 0.392
0.212];
```

Y así sucesivamente con el resto de eslabones

4.2.4 Configuración de la visualización

```
conf.parameters.cameraAZ = 40;
conf.parameters.cameraEL = 10;
conf.parameters.minX = -0.5;
conf.parameters.maxX = 0.5;
conf.parameters.minY = -0.5;
conf.parameters.maxY = 0.5;
conf.parameters.minZ = -0.5;
conf.parameters.maxZ = 1.0;
```

5 Conclusiones

Como conclusión cabe destacar que se ha desarrollado una clase de carácter general capaz de modelizar el comportamiento de cualquier tipo de robot serial a partir de un archivo de configuración y de hacer simulaciones precisas y realistas a partir de los modelos stl del robot. Otra características muy interesante de esta clase es la posibilidad de generar distintos robots como objetos capaces de operar de forma independiente en una misma simulación (Ver Figura 6. Se están desarrollando funcionalidades asociadas a esta clase para poder cargar distintos tipos de entorno e interactuar con los mismos detectando colisiones.

6 Agradecimientos

Este trabajo has sido financiado por el Plan Nacional de I+D+i a través del proyecto "Interpretación de la intención y actuación humana mediante señales biomédicas y el análisis cinemático y dinámico del movimiento" (DPI2011-29660-C04-04).

Referencias

[1] Corke P., "A Robotics Toolbox for MATLAB". *IEEE Robotics and Automation Magazine,* March 1996, Vol.3-1, p.24-32.

[2] "Prácticas de Robótica utilizando Matlab", R.Saltarén, JM Azorín, M. Almonacid, JM Sabater. Ed. Miguel Hernández, 2000

[3] HEMERO. Una herramienta MATLAB-Simulink para la Robótica?, Maza, J.I. y Ollero, A., Marcombo, 2001 .

[4] "An Off-Line Robot Simulation Toolbox", S. Kucuk and Z. Bingul, Comput. App. Eng. Educ., 2009, online.

[5] "An Educational Tool for 6-DOF Industrial Robots with Quaternion Algebra", M. Cakir and E. Butun,, Comput. App. Eng. Educ., 15(2), 2007, pp. 143?154.

[6] "Puma article". *Puma Magazine,* March 2222, Vol.2-2, p.22-222.

[7] Per Bergström, "Computational methods for on-line shape inspection". *PhD. Thesis, Lulea university ISBN ,* November 2009, ISBN: 978-91-86233-09-9

[8] "Simmechanics" *www.mathworks.es*

[9] Huges L. and Bredeche N. "Simbad: an Autonomous Robot Simulation Package for Education and Research". *Proceedings of The International Conference on the Simulation of Adaptive Behavior and in Springer's Lecture Notes in Computer Sciences / Artificial Intelligence series,* March 2006, Vol.3-1, p.24-32.

[10] Arshad H., Jamal J. and Sahran S. "Teaching Robot Kinematic in a Virtual Environment". *Proceedings of the World Congress on Engineering and Computer Science ,* October 2010, Vol.1, p.307-310.

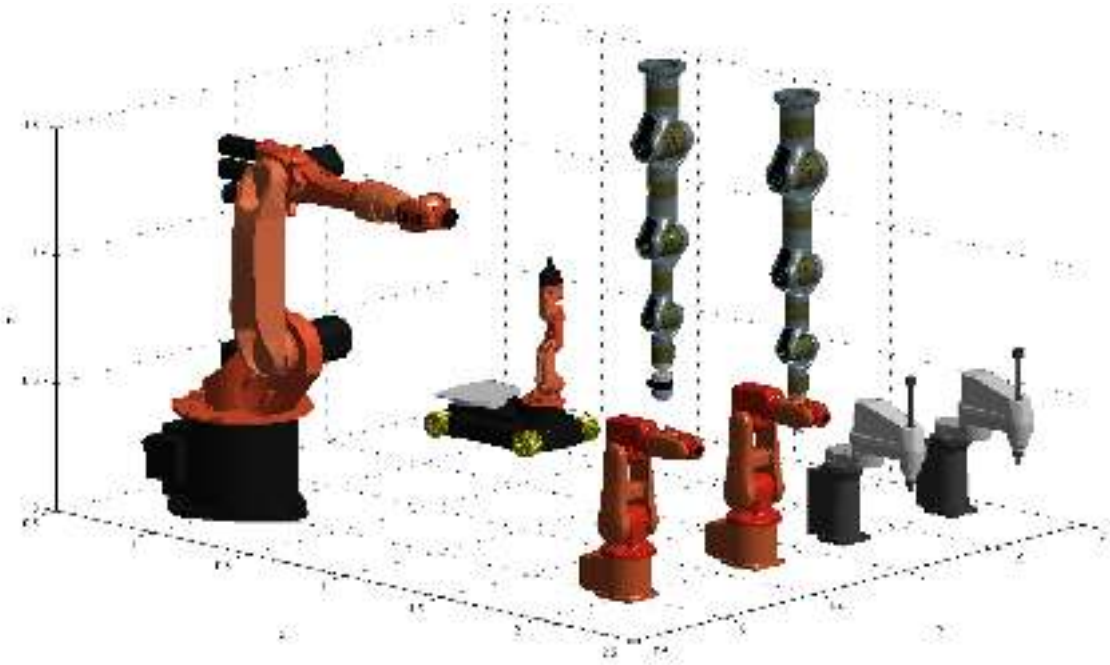


Figura 6: Varios robots cargados con la clase que pueden ser utilizados en una misma simulación

GEOMETRIC HYBRID PATH PLANNING FROM THE ARTIFICIAL POTENTIAL FIELD METHOD

Cristian C. Rincón
Control and Instrumentation Engineering, Distrital University FJC, Colombia, ing.camiloguzman21@gmail.com

Edwar Jacinto G.
Control and Instrumentation Engineering, Distrital University FJC, Colombia, ejacintog@udistrital.edu.co

Fredy H. Martínez S.
Electrical Engineering, Distrital University FJC, Colombia, fhmartinezs@udistrital.edu.co

Abstract

In this paper, a hybrid real-time path-planning algorithm that provide an optimal path for autonomous robots with static obstacles avoidance is studied. The proposed planning algorithm compute a path based on geometric strategies and visual identification of the environment. Is well known that traditional geometric strategies have convergence problems, local minima and definition of non-optimal paths. We propose a strategy that modify the original algorithm, resulting in a new strategy free of convergence problems and with optimal paths. The basic algorithm corresponds to the artificial potential field method modified by using dummy loads. We propose a radar strategy, in which we simulate dummy loads, loads similar to the robot load, at specific points of the environment. These points are determined iteratively by the robot's position and the target position, and are responsible for describing the navigation path. Some experiments with differential drive robots and several simulations are conducted to show the effectiveness and performance of the strategy.

Index Terms: Dummy load, global planning, hybrid system, mobile robot, path planning, potential field method.

1 INTRODUCTION

The traditional approach to design control schemes focuses on the search for global strategies for coordinating the plant dynamics. However, sometimes it is more convenient to define specific strategies for each of the different behaviors of the system. This switching between dynamics has not been immune to robotics.

Path planning is one of the most important topics in robotics. Nowadays, mobile robots work under dynamic environments with machinery parts as moving objects and are using many techniques for navigation, obstacle avoidance, and localization. The problem basically consists in that an agent, non-point object, moves from one point to another, avoiding obstacles by some margin [1]. Two key elements in the solution of this problem is the size of the robot and the speed with which the paths are determined, this latter parameter is often used to evaluate the performance of a given algorithm. Among the techniques used to investigate this problem are: visibility graphs [2, 3], free-space decompositions [4, 5], local path finding [6, 7, 8] and triangulations [9, 10, 11].

In the case of static and known environments, where the robot does not sense or process directly the environment calculating the route, but this task is performed by a central control, the strategies based on computational geometry are a powerful tool for generation and selection of a optimal path (safe path defined quickly).

The Artificial Potential Fields [12, 13, 14] is one of the most popular geometric strategies. In this method, a potential field is designed that combines attraction to the target, and repulsion from obstacles, which guides the robot navigation.

In the traditional scheme of artificial potential fields construction [15], the robot is a particle over which the obstacles apply a repulsive force, and the arrival point or target, applies an attraction force. The sum of forces at each point of the free space in the environment causes the potential field. This potential field guides the movement of the robot (figure 1). At points on which this sum is zero, the potential field does not provide navigation information. This is known as the local minima problem.

We propose a modification of this path planning method that guarantees the elimination of local minima problem, while maintaining the simplicity of the original method. Our proposal focuses on avoiding the obstacle which causes the local minimum, adding dummy loads in the free space,

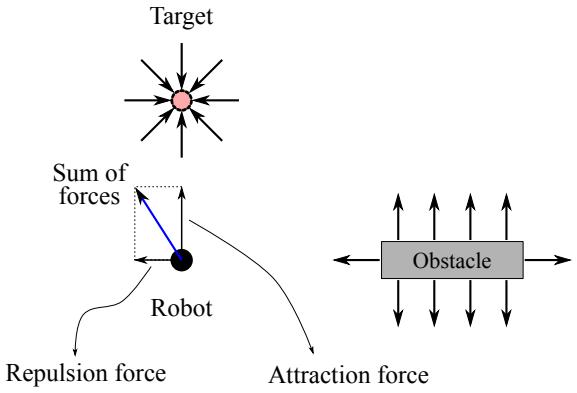


Figure 1: Artificial potential field force at a point p_i in the free space in the environment.

which modify the field distribution in areas where local minima are generated. We formulate a new dynamic for the system in these cases, and a new control strategy, which is why the new path planning scheme is hybrid.

In this modification we take some ideas from research in behavior-based control systems for autonomous robots [16, 17, 18, 19]. In this control scheme, the designer works the robots as a system, where each node corresponds to a robot, and to solve the task, each node has a different behavior. This structure is called a hybrid automaton, and allows use in the design the capacity for abstraction of hybrid systems [20, 21, 22, 23].

The paper is organized as follows. Section 2 presents preliminary concepts and problem formulation. Section 3 shows the sensor-based planning algorithms, definitions, formulations and structures. Section 4 we present the simulation results. And finally, in Section 5, we present our conclusions.

2 PROBLEM FORMULATION

Consider the problem of optimal path planning in a 2-dimensional Euclidean space. Let $W \subset \mathbb{R}^2$ be the closure of a contractible open set in the plane that has a connected open interior with obstacles that represent inaccessible regions. Let $\mathcal{O}$ be a set of obstacles, in which each $O \subset \mathcal{O}$ is closed with a connected piecewise-analytic boundary that is finite in length. Furthermore, the obstacles in $\mathcal{O}$ are pairwise-disjoint and countably finite in number. Let $E \subset W$ be the free space in the environment, which is the open subset of W with the obstacles removed, and the boundary, ∂E, of E is the image of a piecewise-analytic closed curve.

The robot is considered to be a moving polygon in $\mathbb{R}^2$, whose size is defined by the radius r of the circle that circumscribes it (figure 2), and whose special feature is that r is dimensionally small relative to the size of the environment and the obstacles on it. Furthermore, it assumed that the collision-free subset of the robot's configuration space $\mathcal{C}$ is connected. $\mathcal{C}$-space obstacle corresponds to that of a translating disc, that is, the extended boundary of E which is due to the robot radius r (radius of the circle which circumscribes to the robot).



Figure 2: Robot circumscribed in a circle of radius r

We chose to discard the use of on-board sensors (on the robot). Instead, we consider the use of images taken with a camera to the environment (figure 3). From these images, we extract the information of the robots and obstacles. Then, we use this information to implement our routing prediction algorithm. Therefore, for the central control unit, the unit responsible for calculate the navigation path and to control the movement of the robot, it is always possible sense and know the environment. This means that we always know the position of the robot and the obstacles in the environment.

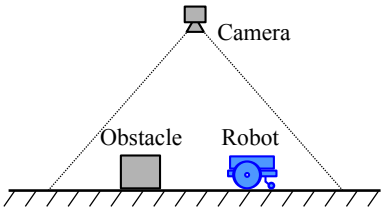


Figure 3: Camera as environment sensor.

The system has a single sensor, a camera capable of observing the entire environment, and clearly identifies the obstacles O, the free space E and its boundary ∂E, and the robot. These observations allow a central control to build an information space I. An information mapping is of the form:

$$q : E \longrightarrow S \tag{1}$$

where S denote an *observation space*, constructed

from sensor readings over time, i.e., through an observation history of the form:

$$\tilde{o} : [0, t] \longrightarrow S \tag{2}$$

The interpretation of this information space, i.e., $I \times S \longrightarrow I$, allows the central control to make decisions (information feedback) [24], that is to calculate the navigation path for the robot.

The problem is to find a plan that navigates the robot along the shortest path from some initial position $p_0(x_0, y_0)$, to the target position $p_t(x_t, y_t)$, while avoiding obstacles.

3 NAVIGATION ALGORITHM

In the calculated navigation path based on the traditional artificial potential field method, there are points for which the sum of forces is close to zero (local minimum, figure 4). The navigation path is modified on these points according to our proposed dummy loads.

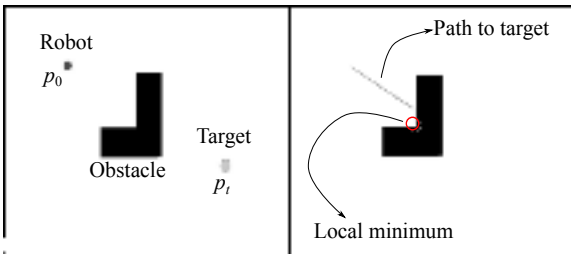


Figure 4: Local minimum. In the left frame the robot is in the upper left (p_0), and the target position in the lower right (p_t). The obstacle is in the middle. The right frame shows a local minimum in the path calculated by the traditional artificial potential field method.

The new path must be calculated to avoid the obstacle that generates the local minimum. Our strategy proposes to simulate two radars centered, one in the initial position of the robot and another in the target position. The range of these radars simultaneously and iteratively grows (figure 5) until they intersect at a point p_1 in the free space E (figure 6). This point (p_1) then becomes as an obliged transit point for the robot in the new path, i.e., a new dummy load attraction similar to the target point. This new dummy load attracts the robot and stops working when the robot reaches it. When the robot is at p_1, this position becomes the new starting point, and from there it is again attracted to the target position.

The model that is assumed for this system is hybrid, because the set of agents switches between

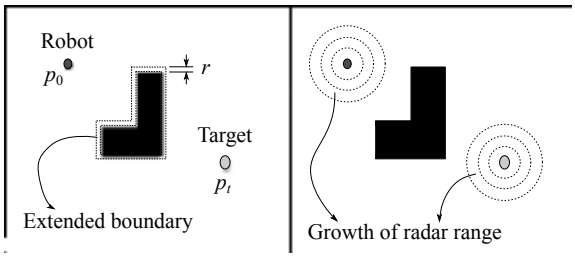


Figure 5: Strategy proposes. In the left frame the control system calculates the extended boundary of the obstacle. The right frame shows the simultaneous growth of the radars range.

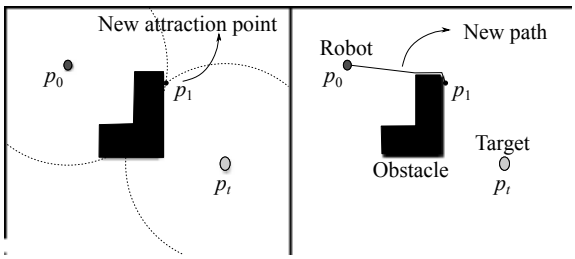


Figure 6: Strategy proposes. In the left frame, the two equal diameter radars intersect at point p_1. The right frame shows the new path calculated with the new attraction point p_1.

two different behaviors during the development of the task. This means that the state space is divided into a set of L_1 and L_2 regions. A first region L_1 when the obstacles do not induce local minima and the path is defined by the traditional artificial potential field method, and a region L_2 when appearing local minimum and the path is defined geometrically by the dummy loads. Each of these two fractions is characterized by a set of differential equations with state variables denoted by:

$$x_i \quad / \quad i = 1, 2, 3, \cdots, b \tag{3}$$

a set of dimension b that allows us to describe the dynamics of the two different behaviors. The system then can be described as:

$$\begin{array}{c} H_p = \{\boldsymbol{X}_p, \boldsymbol{L}_p\} \\ \boldsymbol{X}_p \subset \mathbb{R}^b \end{array} \tag{4}$$

where p is the number of behaviors identified in the system, in this case two, and $\boldsymbol{X}$ is the state space representation for each of the p regions of the state space of the system.

4 PROTOTYPE AND SIMULATIONS

We perform the initial evaluation of the navigation strategy by simulation. However, the configuration of our simulator fully respects the criteria of our current laboratory tests. the simulations consider not only the functionality of the algorithm, we also have taken great care of the physical characteristics of the robot, i.e. size (0.20 m x 0.22 m), weight (0.530 kg) and speed (0.2 m/s). For our experiments, we use the Oomlout Company's open-source design SERB Robot. This is a differential robot constructed of acrylic, and equipped with two continuous rotation servomotors.

The simulations were performed in MatLab. MatLab captures the image from a camera installed over the test environment. The frames captured from the camera are processed and analyzed with edge detection filters to obtain a set of monochrome images, where the free space E is shown in white and the obstacles and the boundary ∂E of E in black. The software is also able to identify the robot. Using this identification, the software determines the initial position of navigation. The target position is given by the user through the interface.

We ran a total of 50 tests in 25 different environments, 20 of these environments correspond to actual test environments in our laboratory, and five were artificially generated to evaluate particularly complex configurations. In each test, we randomly chose the starting point p_0 and the target point p_t. The robot is circumscribed in a circle of radius $r = 10.5$ cm, according to the actual size of our laboratory robots. The environment used is in rectangular shape with a size of 7 m $\times$ 5.5 m. Figure 7 shows four of these tests.

The settings of the obstacles in the environment (completely random) included up to a total of 10 different polygons simultaneously in the same environment. It also allowed the overlapping obstacles to generate even more complex shapes. The location of the start (p_0) and target (p_t) points, although it was random, we prefer to adjust it to the lateral ends of the environment.

In all evaluated cases the control was able to find a navigation path without the presence of local minimum. The robot always reached its target point without problems. The time required for each calculation is heavily dependent on the complexity of the task and processor used. For an AMD Phenom(tm) 9650 Quad-Core Processor with 8 GB of RAM running Linux 64-bit kernel 3.2.0 and MatLab R2011a, the average time for path calculation was 20 to 30 s. Some of these tests can be observed

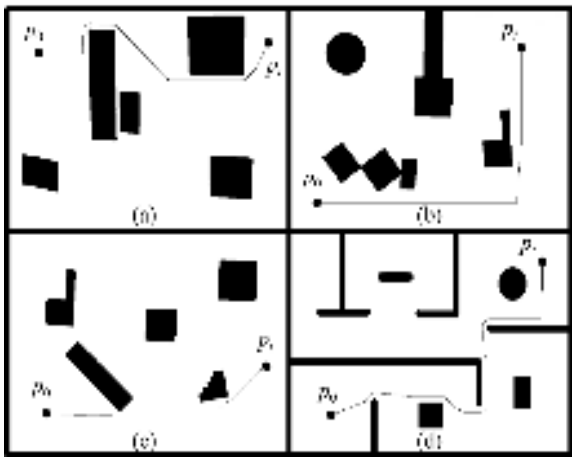


Figure 7: Simulation of the navigation strategy proposed in four different environments. The starting points of the robot p_0 are always in the left side of the environment, and the target points p_t on the right side. The environments (a), (b) and (c) were constructed with geometric obstacles placed randomly. The environment (d) was artificially constructed shaped maze.

at runtime in: http://youtu.be/boIYWRGokto

5 CONCLUSIONS

This paper presents a new strategy for path planning in observable, complex and static environments. The strategy is a modification of the artificial potential field method, thanks to which we achieve a geometric method simple to implement with low computational cost, and free of local minima.

The idea makes use of the theory of hybrid systems, from which we define a continuous dynamics for robots following the basic principles of potential field when obstacles do not induce local minima. When obstacles favor the formation of local minima, the model switches to a dynamic behavior defined by the geometrical characteristics of the environment, obstacles and the robot.

The navigation problem considered assumes a robot that executes commands from a central control unit, this central control unit is responsible for sensing the environment, navigating path design, and control the robot's movements. The proposed algorithm was fully implemented.

The proposed algorithm were tested in 25 different scenarios. In all test cases, polygonal obstacles were introduced. Always the same algorithms were applied, regardless of the environment's complexity or topology. In all test cases, the system was able to quickly determine the navigation path

without falling into local minima. The algorithm creates quite short navigation paths.

While we formulate this strategy for static environments, we are currently evaluating their performance in dynamic environments (moving obstacles and multi-robots environments).

Acknowledgement

This work was supported by the Distrital University Francisco José de Caldas, in part through CIDC, and partly by the Faculty of Technology. The views expressed in this paper are not necessarily endorsed by the Distrital University. The authors thank the research groups ARMOS and DIGITI for the evaluation carried out on prototypes of ideas and strategies.

References

[1] Guangshun, L. and Hongbo, S. (2008) In Chinese Control and Decision Conference CCDC 2008 : ISBN 978-1-4244-1733-9 pp. 3295–3300.

[2] Xu, D., Zhang, F., and Y., Y. (2009) In IEEE International Conference on Intelligent Computing and Intelligent Systems ICIS 2009 : ISBN 978-1-4244-4754-1 pp. 693–698.

[3] Afzulpurkar, N. and Thanh, N. (2008) In IEEE International Conference on Robotics and Biomimetics ROBIO 2008 : ISBN 978-1-4244-2678-2 pp. 2115–2120.

[4] Mannadiar, R. (2010) In 2010 IEEE International Conference on Robotics and Automation ICRA : ISBN 978-1-4244-5038-1 pp. 5525–5530.

[5] Zhang, L., Kim, Y., and Manocha, D. (2007) In IEEE RSJ International Conference on Intelligent Robots and Systems IROS 2007 : ISBN 978-1-4244-0912-9 pp. 7–14.

[6] Bobadilla, L., Martinez, F., Gobst, E., Gossman, K., and LaValle, S. (2012) In American Control Conference ACC2012 : ISBN 978-1-4577-1095-7 pp. 743–749.

[7] Kress-Gazit, H., Fainekos, G., and Pappas, G. (2009) *IEEE Transactions on Robotics* **25(6)**, 1370–1381 ISSN 1552-3098.

[8] Otte, M., Richardson, S., Mulligan, J., and Grudic, G. (2007) In IEEE RSJ International Conference on Intelligent Robots and Systems IROS 2007 : ISBN 978-1-4244-0912-9 pp. 2819–2826.

[9] Hongyang, Y., Huifang, W., Yangzhou, C., and Dai, G. (2008) In Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on : ISBN 978-1-4244-2113-8 pp. 5168–5173.

[10] Kallmann, M. (2005) In International Joint Conference on Artificial Intelligence IJCAI 2005 : ISBN 0938075934 pp. 49–54.

[11] Demyen, D. and Buro, M. (2006) In 21st National Conference on Artificial Intelligence : ISBN 978-1-57735-281-5 pp. 942–947.

[12] Li, G., Yamashita, A., Asama, H., and Tamura, Y. (2012) In 2012 International Conference on Mechatronics and Automation ICMA : ISBN 978-1-4244-4774-9 pp. 1227–1232.

[13] Bin-qiang, Y., Ming-fu, Z., and Yi, W. (2011) In 2011 International Conference on Multimedia Technology ICMT : ISBN 978-1-61284-771-9 pp. 3192–3195.

[14] Huang, Y., Hu, H., and XingQiang, L. (2010) In 2010 8th World Congress on Intelligent Control and Automation WCICA : ISBN 978-1-4244-6712-9 pp. 6414–6418.

[15] Khatib, O. (1985) In IEEE International Conference on Robotics and Automation : pp. 500–505.

[16] Egerstedt, M. and Hu, X. (2002) *Automatica* **38(1)**, 125–130.

[17] Gamage, G. W., Mann, G., and Gosine, R. G. (2009) In Proc. Canadian Conf. Electrical and Computer Engineering CCECE '09 : pp. 660–663.

[18] Lan, Y., Yan, G., and Lin, Z. (2009) In Proc. 48th IEEE Conf. held jointly with the 2009 28th Chinese Control Conf Decision and Control CDC/CCC 2009 : pp. 3032–3037.

[19] Hernandez-Martinez, E. G. and Albino, J. M. F. (2011) In Proc. IEEE Int Control Applications (CCA) Conf : pp. 941–946.

[20] Chaohong, C., Rafal, G., Sanfelice, R. G., and Teel, A. R. (2007) In Proc. Chinese Control Conf. CCC 2007 : pp. 29–36.

[21] Prieur, C., Goebel, R., and Teel, A. R. (2007) **52(11)**, 2103–2117.

[22] Zhang, H. and James, M. R. (2006) In Proc. 45th IEEE Conf. Decision and Control : pp. 4801–4806.

[23] Yoon, K.-H., Lee, J.-K., Kim, K.-H., Park, B.-S., and Yoon, J.-S. (2008) In Proc. Int. Conf. Control, Automation and Systems ICCAS 2008 : pp. 1367–1372.

[24] LaValle, S. M. (2006) Planning Algorithms, Cambridge University Press, Cambridge, U.K. Available at http://planning.cs.uiuc.edu/.

DISEÑO Y DESARROLLO DE ROBOT MÓVIL PARA TRANSPORTE Y DISPENSACIÓN DE MEDICAMENTOS

Roberto Pinillos, Salvador Domínguez
Fundación CARTIF, robpin@cartif.es

Eduardo Zalama, Jaime Gómez García-Bermejo
Instituto Tecnologías de la Producción. Universidad de Valladolid, ezalama@eii.uva.es, jaigom@eii.uva.es

Ángel de Miguel Navares
Proyectos Ingeniería y Gestión S.A. a.miguel@proingesa.es

Resumen

En el presente artículo se aborda el análisis, desarrollo e implantación de un sistema logístico de transporte basado en vehículos inteligentes autopropulsado de mediano tamaño. Se trata de un sistema modular que permite la variación del tipo de contenedor y por lo tanto capaz de transportar mercancías de diferentes clases. Estos robots pueden trabajar tanto como elementos independientes, como formando parte de una flota de carros-robots con interconexión a las aplicaciones informáticas, administrando materiales e integrándose en los sistemas de logística.

Palabras Clave: Robot móvil, sistema de navegación, dispensación medicamentos.

1 INTRODUCCIÓN

En la actualidad existe una tendencia a la automatización en los campos de producción industrial y logística, especialmente en actividades en las que se requiere una gran repetitividad, siendo cada vez más habitual la integración de vehículos autoguiados para transportar mercancías de gran tamaño [7]. En la inmensa mayoría de estos sistemas los recorridos que siguen los vehículos son fijos y predeterminados, de forma que la variación de los caminos supone una remodelación importante de la configuración de la planta [2]. Nuevos sistemas de localización y navegación independientes del entorno, y flexibles en cuanto a realización de trayectorias y tareas, hacen pensar en la posibilidad de utilizar vehículos autónomos en nuevos servicios.

Con el robot presentado en este artículo se pretende dar un paso más hacia el desarrollo de sistemas de transporte más flexibles, basados en robots de menor tamaño y de menor costo. La funcionalidad de este tipo de robots es múltiple. Desde su utilización en almacenes automatizados en los que es posible almacenar o solicitar un tipo de producto, o como sistemas de transporte logístico en recintos industriales destacándose aplicaciones como gestión de residuos, acopio de materiales en cadenas de montaje, sistema de transporte entre células de fabricación, etc. Además, el tamaño de las plataformas permite su extensión a la utilización en otros entornos no industriales tales como transporte de material médico, ofimático, transporte de mercancías en grandes superficies, expositores móviles, etc.

Cumpliendo todos estos requisitos se ha desarrollado una plataforma móvil de mediano tamaño suficientemente robusta dotada de los dispositivos necesarios que le confieran la suficiente movilidad, potencia, seguridad y conocimiento del entorno para realizar las tareas de manera rápida y eficiente sin necesidad de alterar el entorno de trabajo. Se trata de una base móvil modulable al que mediante un sencillo anclaje se le pueden acoplar diferentes tipos de estanterías o armarios móviles que permitan almacenar y dispensar materiales de diversas clases dentro de cualquier línea de logística tanto industrial como de otro tipo.



Figura 1: Robot móvil dispensador de medicamentos

Una de las primeras aplicaciones para las que se va a utilizar este robot es la trazabilidad y dispensación de medicamentos en hospitales (Figura 1). Para ello la plataforma móvil lleva acoplado un armario con cajones automatizados donde se van a transportar los medicamentos para cada paciente. Ya sea por tareas programadas en un determinado horario o por petición bajo demanda, vía web o mediante la interfaz a bordo del robot, la plataforma móvil acudirá a la farmacia para que allí el personal correspondiente prepare los medicamentos necesarios.

Una vez finalizada esta tarea y tras la validación del usuario, el carro-robot inicia el recorrido autónomamente parando en los destinos indicados donde personal autorizado se encarga de la distribución final de los medicamentos, pudiendo mover el carro-robot de forma manual. Cuando se han distribuido todos los medicamentos de esa zona el usuario da la orden al robot para que continúe el recorrido, dirigiéndose a la zona de recarga de batería al finalizar completamente el servicio.

El presente artículo se centra en la explicación del diseño y funcionamiento general de la plataforma móvil independientemente del tipo de mercancía que esté destinado a transportar.

2 ARQUITECTURA HARDWARE

2.1 DISEÑO ELECTROMECÁNICO

La estructura del robot está constituida por una base de Poliuretano Industrial y unos perfiles de aluminio que dan rigidez y soporte a la plataforma móvil.

Para facilitar las labores de montaje y mantenimiento, la plataforma se ha dividido en dos partes separadas por una bandeja extraíble (ver Figura 2). En la parte inferior se encuentran las baterías, el grupo motor, y el grupo de dirección, y en la parte superior las placas de control, la placa PC y los borneros de conexión.



Figura 2: Componentes electromecánicos

En cuanto al sistema de locomoción se ha optado por una plataforma móvil tipo triciclo con una rueda delantera directriz y dos ruedas traseras motrices, formando un grupo motor diferencial con electro-freno incorporado. Este tipo de cinemática permite una gran maniobrabilidad y precisión de movimientos, necesarios para la realización de trayectorias a través de los pasillos del entorno hospitalario y la realización de maniobras de aproximación a puntos concretos como los cargadores de batería o zonas de carga y descarga.

2.2 ELEMENTOS EMBARCADOS

Como se puede apreciar en la Figura 3, el carro robot en la parte delantera dispone de un escáner láser Sick LMS100 [11] que permite la localización del carro-robot y su navegación por el entorno mediante mapeado.

En la parte inferior lleva incorporados sensores de ultrasonidos encargados de la detección de obstáculos para evitar posibles colisiones, y un sensor de infrarrojos que permite la detección de marcas y la aproximación del robot de una forma precisa a puntos especiales (recarga de baterías, etc.).

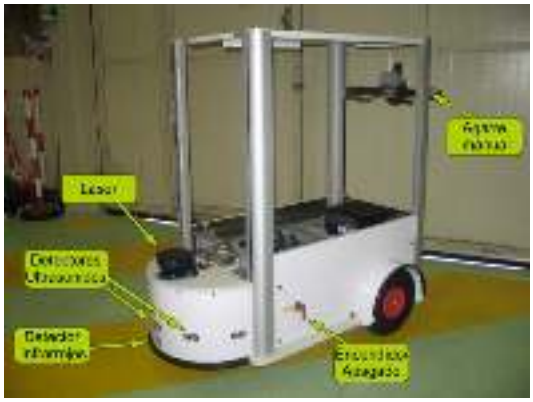


Figura 3: Vista delantera de la plataforma móvil

En la parte trasera (Figura 4) se encuentran más sensores de ultrasonidos detectores de obstáculos, un pulsador de enclavamiento que actúa como parada de emergencia manual deshabilitando los motores ante cualquier posible error de funcionamiento, y un conector adicional para la recarga manual de la batería (En el Apartado 5 se explicará el funcionamiento de la recarga). Además dispone de una serie de indicadores luminosos del estado de funcionamiento y nivel de baterías.

Por último, también en la parte posterior pero a una mayor altura, apropiada para una persona de altura media, el robot dispone de un agarrador con joystick

incorporado que permite al usuario mover la plataforma de forma manual.

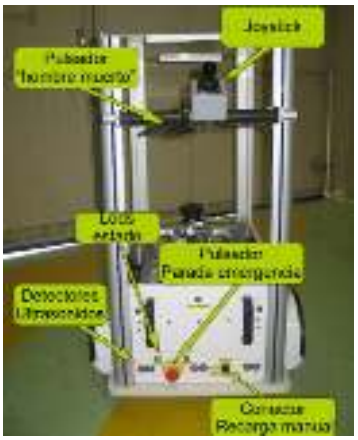


Figura 4: Vista trasera de la plataforma móvil

2.2 PLACA DE CONTROL

La placa base de control hardware es la GPMRC (General Purspose Robot Motion Controller), diseñada y desarrollada por la Fundación Cartif para el control de bajo nivel de los sensores y actuadores de plataformas robóticas. Se basa en un dsPIC de Microchip como procesador principal de cálculo. Esta placa realiza todos los controles de movimiento del robot (avance, retroceso y giro del robot, etc…), recibiendo las ordenes directamente del PC de alto nivel, y poniendo a disposición de éste toda la información sensorial del robot (lecturas de sonars, detección de marcas IR, odometría,….).

Se ha elegido este tipo de placa base para el control de todos estos procesos para poder realizar de forma rápida y eficiente las tareas de más bajo nivel que se precisa que sean atendidas en tiempo real. De esta forma se eliminan los efectos secundarios de utilizar un sistema operativo convencional. Esto proporciona al sistema en general una cierta robustez y velocidad en su funcionamiento y disminuye la posibilidad de cuelgues debidos al software.

3 ARQUITECTURA SOFTWARE

En la Figura 6 se muestra un diagrama de bloques de los módulos funcionales a bordo del carro-robot. Con el fin de aprovechar las ventajas de los sistemas modulares, se ha diseñado un sistema basado en módulos independientes que intercambian información mediante un simple mecanismo de publicación/suscripción de mensajes.

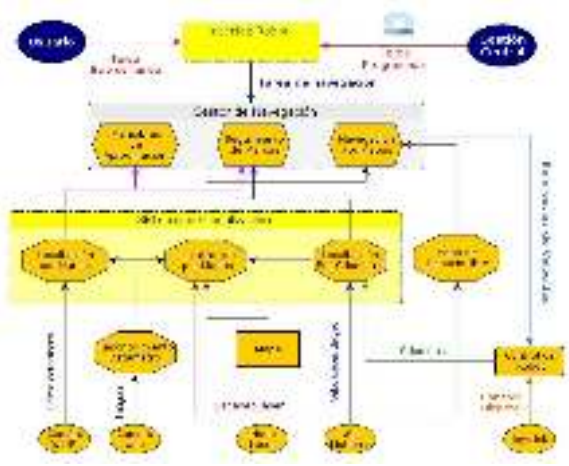


Figura 6: Módulos funcionales

Los diferentes módulos han sido desarrollados utilizando el lenguaje de programación C++ y la plataforma de Robótica ROS (Robot Operating System) [10]. ROS es una plataforma de desarrollo open source para sistemas robóticos que proporciona toda una serie de servicios y librerías que permiten una abstracción del hardware y la creación de aplicaciones complejas y distribuidas para robots. Es la plataforma para robótica más utilizada en la actualidad.

El objetivo principal de ROS, mas que convertirse en una plataforma con un gran número de funcionalidades, es el de permitir la reutilización de código entre distintos desarrollos. Para ello ROS permite la creación de procesos (Nodes) que intercambian información (Messages) a través de una serie de canales (Topics). Esto permite desacoplar los distintos nodos que, a su vez, pueden ser reutilizados en distintas aplicaciones.

A continuación se describen brevemente los distintos módulos que lo componen:

Control robot. Módulo que abstrae el resto del software de las características del robot, sus parámetros geométricos y tipo de cinemática. Transfiere las referencias de control a las características concretas del carro-robot y las comunica al driver de la controladora **GPMRC**.

Nodo Láser. Driver encargado de gestionar el láser sick LMS100 y obtener los datos de escaneo a través de la red TCP/IP y transformarlos a mensaje ROS.

Wii Motion +.- Realiza la función de giróscopo para mejorar la localización del robot leyendo la información del sensor WiiMotion+ [12]. Calibra los datos del giróscopo al iniciar el nodo y en momentos en los que el robot se encuentre completamente parado. Durante el movimiento comunica los valores

de los incrementos angulares recibidos del dispositivo para que el módulo de **Localización por Odometría** los integre con el resto de datos.

Localización mapas. Estima la posición del robot mediante distribución probabilística utilizando el algoritmo de localización adaptativo de Monte-Carlo (AMCL) [3], a partir de los datos de odometría (**Localización por Odometría**) y los escaneos láser (**Nodo Láser**), que son comparados con un mapa del entorno predefinido.

Localización por Odometría. Calcula la posición estimada en base al movimiento del robot con los datos recibidos desde la GPMRC (encoders) y del nodo **Wii Motion + (**giróscopo).

Mapa de Espacio Libre. Genera un mapa local de ocupación del robot para que pueda ser utilizado en la comparación con un mapa global del entorno y de esta manera determinar la posición y la detección de obstáculos.

Navegación mapas. Se encarga de llevar a cabo el seguimiento de trayectorias cuando el sistema de localización sea el de mapeado láser realimentado por la posición recibida del módulo **Localización mapas y** el **Mapa de Espacio Libre**. Escucha órdenes de inicio, pausa y reanudación del seguimiento de la trayectoria que puede recibir a modo de serie de puntos o leerla directamente desde un archivo.

Cámara Wii IR. Driver que obtiene los puntos infrarrojos detectados por la cámara de infrarrojos del mando wiimote de Nintendo [12]. Publica periódicamente las coordenadas de cada punto detectado dentro de la imágen y la intensidad.

Localización por marcas. Con las coordenadas de los puntos infrarrojos detectados por el nodo **Cámara Wii IR** determina si su disposición responde a alguna de las marcas existentes. Si es así, calcula la posición relativa a la que se encuentra de la marca y en el caso de que esa marca disponga de coordenadas globales, le permite establecer las coordenadas globales del robot dentro del entorno de funcionamiento.

Seguimiento de marcas. Obtiene la posición de las marcas previamente colocadas en el entorno desde el módulo de **Cámara Wii IR** y teniendo en cuenta la posición del dispositivo calcula las referencias necesarias para que el robot siga el camino. Detectando la posición de al menos 2 reflectores es capaz de seguir la trayectoria marcada. No proporciona una posición global del carro-robot.

Maniobras de Aproximación. Conoce la posición de la marca a partir de los datos obtenidos por **Localización por marcas** y a través de la realimentación de su posición relativa genera los comandos de velocidad angular y lineal para que el robot se aproxime a la marca con máxima precisión.

Gestor de Navegación. Recibe desde el sistema central o desde la interfaz a bordo del robot la tarea de ejecución de una trayectoria y será el encargado de decidir que sistema de navegación es el preferente en cada tramo de la trayectoria a realizar. Establecerá un orden de prioridades para que ante posibles deficiencias en el funcionamiento del sistema utilizado, si fuese posible, el robot intente continuar con su cometido con alguno de los restantes sistemas de navegación.

Sistema de identificación "DataMatrix" [1]. Permite identificar marcas bidimensionales mediante el uso de una cámara web. La identificación de estas marcas permite al sistema de navegación reafirmar la posición del robot. Una de las marcas utilizadas está situada en el propio cargador de baterías, de esta forma el robot puede comprobar que efectivamente está situado frente al cargador y que puede comenzar la maniobra de conexión. Otras marcas pueden ser utilizadas en puntos del recorrido o en destinos clave como puede ser la "farmacia", "ascensores", etc.

4 NAVEGACIÓN

4.1 FUNCIONAMIENTO MANUAL

El robot puede desplazarse manualmente mediante un joystick inductivo integrado en el asa (Figura 5), que permite moverlo libremente a cualquier posición. Este mando permite utilizar el robot como un carro de enfermería convencional, sin suponer ningún tipo de esfuerzo para el usuario debido a que va asistido tanto en dirección como en arrastre.

El joystick permite controlar el robot de forma manual directamente por conexión hardware sin necesidad de que el PC de alto nivel esté funcionando. En el caso de que el PC esté conectado, el joystick tiene prioridad sobre cualquier orden que reciba de éste.



Figura 5: Joystick funcionamiento manual

Para garantizar la seguridad y evitar posibles movimientos involuntarios el sistema está desarrollado para que el usuario deba tener las dos manos en contacto con la empuñadura del robot. Con la mano izquierda debe accionarse el freno de bicicleta que realiza las funciones de "hombre muerto", debe estar presionado para ejecutar cualquier orden desde el joystick.

Con la mano derecha se ha de manejar el joystick propiamente dicho, es de tipo palanca omnidireccional y establece el sentido de marcha, la dirección y velocidad del motor.

4.2 LOCALIZACIÓN Y NAVEGACIÓN AUTÓNOMOS

Uno de los puntos críticos para este tipo de robots autónomos es la localización, la cual permite situarlo en el entorno y llevar a cabo el recorrido hasta el destino previsto. Además es de vital importancia disponer de un sistema de localización que requiera una mínima adecuación del entorno, como es el caso del mapeado láser utilizado en el presente carro-robot. Este sistema utiliza el láser Sick LMS100 instalado en la parte frontal de la plataforma móvil para poder navegar sin necesidad de la instalación de ningún tipo de baliza.

El algoritmo de localización adaptativo de Monte-Carlo (AMCL) [3] estima la posición del robot mediante distribución probabilística, a partir de los datos de odometría y los escaneos láser, que son comparados con un mapa del entorno predefinido. Previamente a la localización mediante este algoritmo, es necesario crear el mapa del entorno a partir de los escaneos láser obtenidos en un recorrido realizado con el carro-robot de forma manual. Por medio de escaneos asociados a cada posición intermedia (calculada mediante odometría) se crea el mapa de ocupación del entorno. Estos datos son los que van a permitir calcular la posición del robot, al ser comparados con los escaneos láser actuales. La utilización de mapas preaprendidos frente a técnicas de SLAM (Simultaneous Location and Mapping [5]) tiene como ventaja el poder planificar trayectorias óptimas antes de comenzar la navegación y ejecutarlas de forma más rápida y segura.

Este sistema de localización puede variar su grado de exactitud dependiendo de lo dinámico que sea el entorno, por lo que en puntos críticos como las zonas de recarga de baterías donde se requiere una aproximación precisa es conveniente complementarlo con algún otro método.

En la Figura 6 se puede apreciar un mapa del entorno creado previamente y la posición actual del robot calculada a partir de los escaneos láser.

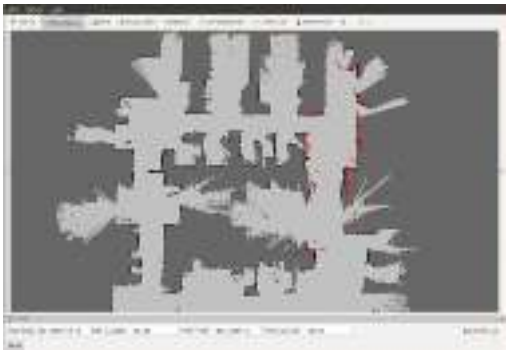


Figura 6: Localización mediante mapeado láser.

Conocida la localización actual, el robot ha de seguir la trayectoria correspondiente generando las referencias de movimiento necesarias que le dirijan hacia el siguiente punto de destino. Los robots han de operar en zonas del hospital formadas por corredores, pasillos y ascensores en las que es habitual la presencia de personas, por ello anteponiendo la seguridad, los carros-robots se detendrán ante la existencia de un obstáculo, sin intentar evitarlo, no continuando la marcha hasta que el camino vuelva a estar libre. En este caso el módulo de navegación reactivo incluye dos perfiles de distancia de seguridad alrededor del robot y configurables. Cuando un obstáculo invade el primer perfil de distancia (detectado por el láser Sick) el robot reduce la velocidad y emite señales acústicas, mientras que si se invade el segundo perfil de distancia más próximo al robot (detectado por los sensores de ultrasonidos) se detiene totalmente.

4.3 MANIOBRAS

Durante el seguimiento de una trayectoria el carro-robot va a encontrase con nodos especiales a los cuales no debe acceder de forma convencional, es decir, desplazamiento hacia adelante. Estos pueden ser puntos de carga o descarga de mercancía, elevadores, puntos de recarga de baterías, etc...

Las maniobras requieren un alto grado de precisión que en determinados entornos no se puede garantizar con la navegación por mapas. Para ello el robot está dotado de un sistema de aproximación basado en la detección de marcas mediante cámara infrarroja utilizando la tecnología del mando wiimote de Nintendo [4,9].

Cuando el módulo de **Gestión de navegación** lo requiere el módulo de **Maniobras de Aproximación** se activa y comienza a recibir datos del módulo **Cámara Wii IR**. Mediante los módulos de **Localización por Marcas** y el **Detector de Datamatrix**, puede distinguir la marca y comprobar que se encuentra ante el destino correcto, después

realizará la maniobra de aproximación a dicha marca de una forma precisa. Durante todo el proceso existe una realimentación con los datos de posición relativa a la marca para generar las referencias angulares y de velocidad necesarias, que serán comunicadas al módulo de **Control Robot**.

5 RECARGA DE BATERIAS

Uno de los puntos esenciales para lograr un robot móvil totalmente funcional es garantizar su disponibilidad durante el mayor tiempo posible y que no requiera ningún tipo de acción humana para su normal funcionamiento. Para ello se ha diseñado un sistema de recarga de baterías totalmente autónomo.

En la Figura 7 se puede ver el cargador diseñado para el sistema de recarga. Dispone de dos pines de contacto móviles activados por un contactor biestable evitando la presencia de tensión eléctrica expuesta. Cuando se encuentra en modo de reposo los pines están incrustados en el cargador y están libres de tensión. Cuando el robot detecta que se encuentra sobre el cargador emite mediante una señal infrarroja la orden de elevar los pines del cargador, al hacerlo éstos se conectan con dos placas metálicas situadas debajo del frontal del chasis del carro-robot, iniciándose el ciclo de carga de baterías.



Figura 7: Estación de carga. a) reposo b) activo

Cuando el sistema de gestión del robot detecta la carga completa o ha de ejecutar una tarea, desactiva la señal de presencia y la base de carga desconecta la tensión en los pines, retrayéndolos al mismo tiempo y ocultándolos en su interior.

La acción de alejamiento de la plataforma móvil respecto del cargador y la de aproximación, se consideran maniobras de precisión y son ejecutadas como se ha explicado en el Apartado 4.3.



Figura 8: Robot acoplado a la estación de carga.

6 CONCLUSIONES

Con el robot desarrollado presentado en este artículo se ha logrado un sistema de transporte modular, flexible, de mediano tamaño y bajo costo. Además su gran movilidad y su implantación sin necesidad de alterar el entorno de trabajo, hacen pensar en la posibilidad de integrarlo en diferentes líneas de logística de forma sencilla y no solamente en procesos industriales como hasta ahora venía siendo lo más habitual.

La inclusión del sistema de aproximación mediante marcas permite subsanar la posible desventaja de la navegación mediante mapas, la falta de precisión en determinados entornos. Este aspecto permite realizar entre otras, la maniobra de aproximación a la estación de carga de baterías lo que proporciona gran autonomía y aumenta el grado de disponibilidad del robot para realizar tareas cuando sea requerido.

Por último cabe destacar su versatilidad a la hora de recibir tareas. Puede funcionar bajo demanda aceptando tareas a través de la interfaz embarcada o mediante su aplicación web, o seguir una agenda de eventos preprogramados. Además admite la posibilidad de trabajar como elemento independiente o formar parte de un sistema más complejo conectado a un gestor central [8].

Agradecimientos

Este trabajo ha sido parcialmente financiado por Ministerio de Economía y Competitividad (Programa Innpacto IPT-2011-0854-900000), Ministerio de Ciencia e Innovación (Proyecto de Investigación Fundamental no Orientada Ref. DPI2011-25489) y Junta de Castilla y León (Agencia de Inversiones y Servicios, y Prog. de Apoyo a Proys. de Investigación VA013A12-2).

Referencias

[1] Benhaim, M., Houlette, C.,... (2011) "GS1 DataMatrix: An introduction and technical overview of the most advanced GS1 Application Identifiers complian symbology", GS1.

[2] CLEBAR (2013). *http://clebar.es/filoguiados.html*.

[3] Dellaerty, F., Foxy, D., Burgardz, W., Thrun, S. (1999) "MonteCarlo localization for mobile robots", *IEEE International Conference on Robotics and Automation(ICRA-99)*.

[4] Domínguez, S., Zalama, E., Gómez, J. (2012) "Corrección de la posición mediante marcas planas utilizando el sensor IR del wimote", *XXXIII Jornadas de Automática*, pp. 711-719.

[5] Durrant-Whyte, H.; Bailey, T. (2006). "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms". Robotics and Automation Magazine 13 (2): 99–110. doi:10.1109/MRA.2006.1638022.

[6] Fox, D., (2001) "KLD-Sampling: Adaptative Particle Filter", *Advances in Neural Information Processing Systems 14. MIT Press*.

[7] JBT Corporation (2013). *http://jbtcorporation.com*.

[8] López, J., Pérez, D., Pinillos, R., Domínguez, S., Zalama, E., Gómez, J. (2012) "Diseño y desarrollo de un sistema de transporte reconfigurable con navegación basada en mapas y balizas pasivas", *Revista Iberoamericana de Automática e Infomática Industrial, ISSN 1697-7912. Elseiver. Vol 09 num.01,pp. 57-68*.

[9] Po-Wei, C., Kuang-Shun, O., Kuo-Shen, C. (2010) "IR indoor localization and wireless transmission for motion control in smart building applications based on Wiimote technology", *SICE Annual Conference 2010, pp. 1781-1785, 18-21 Aug. 2010*.

[10] Quigley, M., Conley, K., Gerkey,B., (2009) "ROS: an open-source Robot Operating System", *International Conference on Robotics and Automation*.

[11] SICK, (2013) "LMS100 laser measurement sensor", http://*www.sick.com*.

[12] Wii Reverse engineering, (2012) http://*wiire.org*.

Manipulación de Objetos con dos Dedos Usando Información Táctil

Andrés Montaño
andres.felipe.montano@upc.edu

Raúl Suárez
raul.suarez@upc.edu

Resumen

La integración de sensores táctiles en manos robóticas contribuye de forma importante a la creación de robots autónomos capaces de interactuar con su entorno y manipular diferentes tipos de objetos, de los cuales pueden ser desconocidas sus características físicas, entre ellas su forma. En este trabajo se propone un método para realizar la manipulación de objetos desconocidos, sujetados con dos dedos de la mano robótica Schunk Dexterous Hand. Los dedos se usan para manipular el objeto cambiando su orientación, haciendo uso de la información táctil extraída de los sensores instalados en la yema de los dedos. Los puntos de contacto sobre la superficie de las puntas de los dedos son modelados introduciendo una articulación virtual que agrega un grado de libertad a cada dedo. El artículo incluye tanto la descripción de los desarrollos realizados como su aplicación en experimentos reales.

Palabras clave: Robótica, Sujeción, Manipulación, Sensores Táctiles.

1 Introducción

La manipulación diestra, que es la capacidad de cambiar la posición y orientación de un objeto con respecto a la mano mientras se mantiene una sujeción estable del objeto [5], ha impulsado el desarrollo de nuevos elementos terminales de sujeción, algunos con características antropomórficas [4], capaces de realizar tareas complejas aportando nuevas alternativas a las primeras pinzas robóticas que habían sido diseñadas para tareas más sencillas de tipo *pick-and-place*. Sin embrago, uno de los principales inconvenientes que se presentan al aumentar el número de grados de libertad (*gdl*) del elemento terminal de sujeción (mano robótica), es la complejidad del espacio de trabajo que se genera al intentar hacer manipulación diestra.

Entre las tareas que requieren manipulación diestra se encuentran, por ejemplo, abrir y cerrar la tapa de rosca de una botella. Generalmente, durante la manipulación de un objeto se espera que los contactos entre la mano y el objeto manipulado se produzcan en lugares específicos. Sin embargo, en las aplicaciones más complejas, la ubicación y las características de los puntos de contacto no se pueden predecir o no pueden ser modeladas con antelación [6]. Por lo tanto, una solución a este problema de incertidumbre es la incorporación de un sistema de sensores táctiles que sea capaz de detectar los contactos y de medir las fuerzas que se producen en ellos. La integración de sensores táctiles en diferentes partes del robot contribuye en gran medida a la creación de robots autónomos capaces de interactuar con su entorno.

Los seres humanos son capaces de manipular objetos desconocidos sin necesidad de observarlos o tener información previa sobre sus propiedades. Incluso, durante la manipulación son capaces de extraer información del objeto manipulado, como la forma y la postura. En cambio los robots tradicionales necesitan información precisa de las propiedades del objeto con el fin de manipularlo exitosamente [8]. Así, se han creado manos robóticas con sensores táctiles que tratan de imitar el comportamiento humano. La piel humana tiene diferentes receptores táctiles que permiten detectar el contacto con los objetos, las fuerzas en el contacto y saber cuando hay un deslizamiento del objeto que está manipulando. Además, muchas de las características del objeto pueden ser extraídas a partir de la interacción con el objeto mismo, por ejemplo, su forma, su temperatura o la textura. El control muscular preciso y la destreza de la mano humana se deben a la información de realimentación desde los receptores táctiles [3]. La información táctil es importante para que las manos robóticas diestras puedan reconocer las propiedades de los objetos y manejarlos con precisión. Los sensores táctiles robóticos, al igual que los receptores táctiles humanos, son capaces de detectar cuando se produce el contacto y medir las fuerzas aplicadas en el mismo, además los sensores se pueden utilizar para obtener información acerca de la forma del objeto, su posición en el espacio o la ubicación de los puntos de contacto. La detección de

deslizamiento y la estimación del coeficiente de fricción entre el dedo y el objeto son otras de las aplicaciones típicas de los sensores táctiles [10].

En este trabajo se presenta un modelo cinemático de la mano robótica Schunk Dextrous Hand (SDH2) teniendo en cuenta los sensores táctiles instalados en sus dedos. En el modelo se introduce una articulación virtual para representar el lugar en que se producen los puntos de contacto, esta articulación virtual agrega un grado de libertad a cada dedo de la mano. También se propone un método para manipular objetos desconocidos que puedan ser sujetados utilizando dos dedos de la mano robótica, la manipulación consiste en cambiar la orientación del objeto, evitando que este caiga. Durante la manipulación el objeto puede moverse sobre la superficie de los dedos dependiendo de la forma particular del mismo, así que se introduce un ajuste en línea de la distancia entre los dedos de forma que se mantengan las fuerzas de contacto, de esta forma los dedos seguirán el contorno de la forma del objeto mientras este se mueve.

El problema de manipulación diestra y sus aplicaciones ha sido estudiado por diversos autores. Xue et al. [11] presentaron un método para la rotación de objetos con superficie de revolución usando al menos dos dedos de una mano robótica con múltiples dedos. El método que proponen se basa en encontrar los puntos de contacto que sirven para resolver el problema de rotación del objeto y se optimiza la pose relativa entre la mano y el objeto para alcanzar el máximo ángulo de rotación. Odhner et al. [9] propone una solución para la sujeción de objetos pequeños usando dedos sub-actuados. En su trabajo se presenta un nuevo diseño de dedos robóticos y un método de sujeción que permite realizar tareas del tipo *flip-and-pinch*, en las cuales la mano recoge objetos pequeños planos que se encuentran sobre una mesa, girando los objetos hasta que se encuentran en una configuración estable. La información táctil también ha sido tenida en cuenta en el proceso de manipulación, Corrales et al. [1] presentaron un algoritmo para el reajuste de los dedos de la mano mientras se lleva a cabo la manipulación. Partiendo de un plan generado con un planificador geométrico, que tiene en cuenta las restricciones para mantener el contacto y la cinemática de los dedos, se utiliza la información táctil para garantizar que los dedos apliquen y mantengan una fuerza de contacto preestablecida durante la manipulación de un objeto. La reconstrucción del contorno del objeto manipulado en 2D propuesta por Moll et al. [7] o en 3D, propuesta por Meier et al. [6] son otros de los problemas que se pueden resolver combinando la manipulación diestra con la información táctil.



Figura 1: Imagen de la mano robótica Schunk Dexterous Hand (SDH2).

Este artículo está organizado de la siguiente forma. En la Sección 2 se hace un análisis de la mano robótica utilizada, presentando la cinemática directa e inversa que se utilizará para la manipulación. En la Sección 3 se encuentran los algoritmos propuestos para la manipulación moviendo uno o dos dedos. En la Sección 4 se presenta un algoritmo de manipulación que tiene en cuenta la realimentación de fuerza. En la Sección 5 se presenta un ejemplo de manipulación de un objeto real. Finalmente en la Sección 6 se da un sumario del trabajo realizado y propuestas para trabajo futuro.

2 Mano robótica utilizada

La mano robótica utilizada en este trabajo es la Schunk Dexterous Hand (SDH2) mostrada en la Figura 1. Ésta es una mano con tres dedos y siete *gdl*. Cada dedo tiene dos *gdl* activos para mover las falanges proximal y distal, dos de los dedos están acoplados mecánicamente por lo que comparten el *gdl* adicional que les permite rotar para ubicarse uno frente al otro. El tercer dedo se encuentra fijo en la base y no puede hacer rotaciones. Cada dedo posee dos sensores táctiles, uno en la falange proximal y el otro en la distal.

2.1 Cinemática directa

Para describir el modelo cinemático directo de la mano se utilizan los parámetros de Denavit-Hartenberg (DH) [2]. A cada dedo se le asigna un sistema coordenado al igual que a los puntos

Art.	α_i	a_i	d_i	θ_i
0	0	0	–	–
1	90	0	0	θ_1
2	0	86.5	0	$\theta_2 + \pi/2$
3	0	68	0	$\theta_3 + \pi/2 - \varphi_1$
4	0	r_1	0	0
0'	0	0	–	–
1'	90	0	0	θ_1
2'	0	86.5	0	$\theta_4 + \pi/2$
3'	0	68	0	$\theta_5 + \pi/2 - \varphi_2$
4'	0	r_2	0	0

Tabla 1: Parámetros de Denavit-Hartenberg para los dedos de la SDH2

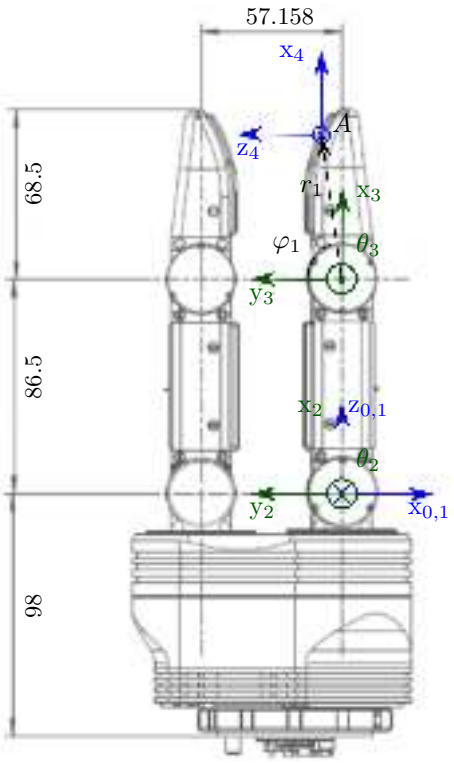
de contacto sobre ellos, A y B. En la Figura 2a se pueden observar los sistemas coordenados para el dedo F_1. Entre los sistemas coordenados de los dedos F_1 y F_3 que son los que se utilizan para la manipulación, existe un desplazamiento de 66 mm sobre el eje y_0 del sistema de coordenadas de la base del dedo F_1 que será tenido en cuenta en el análisis como el sistema de coordenadas global, como se puede observar en la Figura 2b. En la Tabla 1 se muestran los parámetros DH para los dos dedos usados de la SDH2.

Debido a que el punto de contacto puede ser cualquier punto sobre la superficie del sensor en la punta del dedo, se usa una articulación virtual para modelar el posible punto de contacto, adicionando un *gdl* al dedo. Para describir este punto, se introducen en los parámetros DH la longitud r_1 y el ángulo φ_1, para el punto de contacto A, y, r_2 y φ_2 para el punto de contacto B. Estas dos variables sólo dependen de la posición del punto de contacto sobre el sensor, cómo se muestra a continuación. Como los dedos de la mano tienen las mismas dimensiones físicas, el análisis que se haga para uno de ellos es válido para el otro, así que se presenta el análisis sólo para el dedo F_1. Las ecuaciones para el dedo F_2 se calculan utilizando el mismo procedimiento. Asumiendo que el punto de contacto A se produce en el plano x_3y_3, es decir con $z_3 = 0$, éste se puede expresar en el sistema coordenado 3 como,

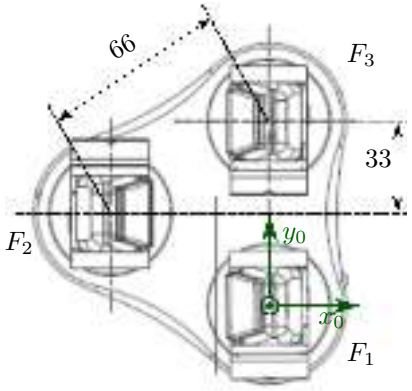
$$r_1(A_{x_3}, A_{y_3}) = \sqrt{{A_{x_3}}^2 + {A_{y_3}}^2} \qquad (1)$$

$$\varphi_1(A_{x_3}, A_{y_3}) = \arctan\left(\frac{A_{x_3}}{A_{y_3}}\right) \qquad (2)$$

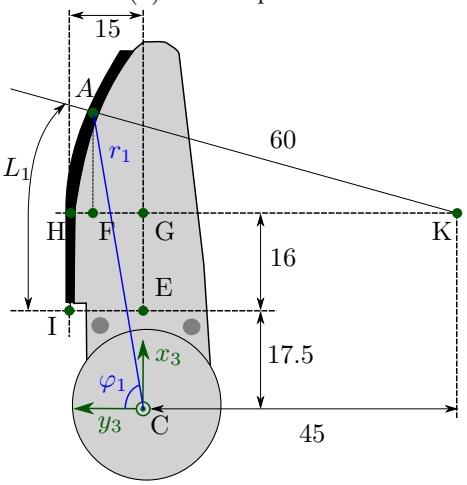
Debido a que la superficie del sensor táctil no es plana (ver Figura 2c), se necesita una función determinada para expresar un punto sobre esta superficie. En el plano x_3y_3, la parte curva del sensor se puede expresar como un arco con radio 60 mm y la parte plana del sensor como una línea



(a) Vista de perfil.



(b) Vista superior.



(c) Detalle de la punta del dedo.

Figura 2: Dimensiones y sistemas coordenados sobre la SDH2 utilizados para el análisis cinemático.

recta. La componente A_{y_3} del punto de contacto se puede expresar en función de la componente A_{x_3} distinguiendo dos casos, el primero cuando el punto de contacto está en la superficie plana del sensor y el segundo cuando está en la superficie curva, sobre el arco con centro en el punto $K = (33.5, -45)$ sobre el sistema coordenado 3. Las siguientes expresiones dan A_{y_3} en cada caso,

$$A_{y_3}(A_{x_3}) = 15 \text{ para } 17.5 \leq A_{x_3} \leq 33.5 \quad (3)$$

$$A_{y_3}(A_{x_3}) = K_{y_3} + \sqrt{60^2 - (A_{x_3} - K_{x_3})^2} \quad (4)$$
$$\text{para } 17.5 \leq A_{x_3} \leq 33.5$$

Usando las ecuaciones (1), (2), (3) y (4), r_1 y φ_1 pueden ser expresados en función de A_{x_3} como,

$$r_1(A_{x_3}) = \sqrt{{A_{x_3}}^2 + A_{y_3}(A_{x_3})^2} \quad (5)$$

$$\varphi_1(A_{x_3}) = \arctan\left(\frac{A_{x_3}}{A_{y_3}(A_{x_3})}\right) \quad (6)$$

Para calcular las coordenadas del punto de contacto A es necesario utilizar la información del sensor táctil. El sensor ofrece información sobre la altura L_1 en la cual se ha producido el contacto, con esta medida indirecta y haciendo un análisis geométrico basado en la Figura 2c se pueden deducir las siguientes expresiones,

$$A_{x_3}(L_1) = 17.5 + L_1 \text{ para } L_1 < 16 \quad (7)$$

$$A_{x_3}(L_1) = 33.5 + r_1 \sin\left(\frac{L_1 - 16}{r_1}\right) \quad (8)$$
$$\text{para } L_1 \geq 16$$

2.2 Cinemática inversa

El problema de la cinemática inversa para uno de los dedos de la mano SDH2 se define como, dada una posición y orientación del punto de contacto A, determinar los valores articulares, θ_2 y θ_3, que permitan al dedo alcanzar este punto. Existen diferentes métodos para calcular la cinemática inversa, en este trabajo se propone una solución geométrica ya que sólo se trabaja en el plano $y_0 z_0$ con dos *gdl*.

Para el análisis geométrico es necesario encontrar los valores de los ángulos ρ, σ y γ mostrados en la Figura 3, que se pueden calcular usando la ley de los cosenos como,

$$\rho = \arccos\left(\frac{-|OA|^2 + |OC|^2 + |CA|^2}{2|OC||CA|}\right) \quad (9)$$

$$\sigma = \arccos\left(\frac{-|CA|^2 + |OC|^2 + |OA|^2}{2|OC||OA|}\right) \quad (10)$$

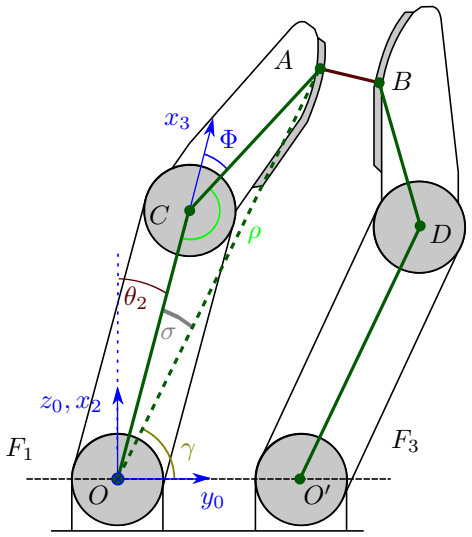


Figura 3: Modelo de dos dedos de la SDH2 usado para el análisis de la cinemática inversa.

$$\gamma = \arctan\left(\frac{A_{z_0}}{A_{y_0}}\right) \quad (11)$$

Por otro lado, la relación entre θ_3 y φ_1 está determinada por el punto de contacto y el valor del ángulo Φ como sigue,

$$\theta_3 = \Phi - \pi/2 + \varphi_1 \quad (12)$$

Al tratarse de un problema en el plano con dos *gdl* existen dos posibles soluciones para un punto dado, sin embargo, teniendo en cuenta las restricciones geométricas que impone la tarea de manipulación, en este caso sólo serán validas las configuraciones obtenidas para valores positivos del ángulo Φ. De este modo los valores articulares de θ_2 y θ_3 están dados respectivamente por las siguientes expresiones,

$$\theta_2 = -\sigma - \gamma + \text{sign}(A_{y_0})\pi/2 \quad (13)$$

$$\theta_3 = -\rho + \pi/2 + \varphi_1 \quad (14)$$

donde $\text{sign}(x)$ representa la función

$$\text{sign}(x) = \begin{cases} 1 \text{ si } x \geq 0 \\ -1 \text{ si } x < 0 \end{cases}$$

3 Manipulación de objetos

Cuando se sujeta un objeto usando dos dedos se puede manipular haciendo que un dedo siga una trayectoria independiente mientras que el otro dedo se mueve cambiando la orientación del objeto de tal forma que este no caiga. La trayectoria que sigue el primer dedo puede cumplir con algunas restricciones o simplemente ir hacia una posición final deseada. La información táctil obtenida

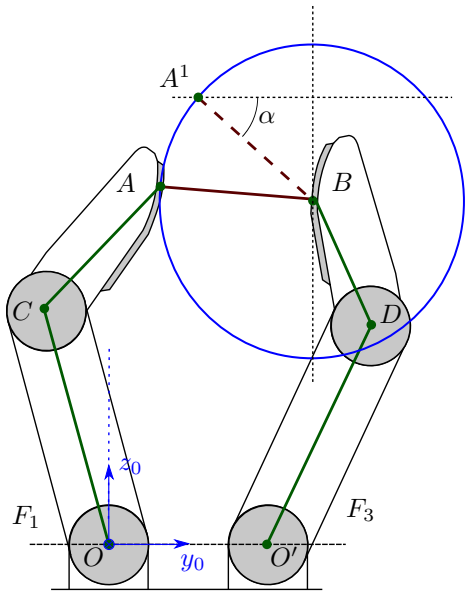


Figura 4: Modelo dos dedos de la SDH2 para realizar la manipulación moviendo un sólo dedo.

de los sensores se usa para ajustar la distancia entre los dos puntos de contacto y mantener una buena sujeción con una presión aproximadamente constante. Durante la manipulación esta distancia puede variar debido a la forma del objeto o debido al deslizamiento sobre la punta de los dedos que se produce al manipular objetos curvos. Además la información táctil permite detectar el punto de contacto sobre el dedo y las fuerzas que se están produciendo.

Para el proceso de manipulación se proponen dos alternativas, la primera, que un dedo se mueva y el otro se mantenga estático permitiendo cambiar la orientación del objeto, y, la segunda, que los dos dedos se muevan para realizar la manipulación. El análisis de estos dos posibles escenarios se presenta en las Secciones 3.1 y 3.2.

3.1 Manipulación con un dedo

Considérese un objeto muy fino, como un palillo, tomado por los extremos entre los dedos, como se muestra en la Figura 4, de modo que la distancia entre los puntos de contacto no cambia y no se produce deslizamiento durante la manipulación. La información táctil permite determinar los puntos de contacto sobre el sensor y la fuerza que se produce en ellos. La distancia Λ_{AB} entre los puntos de contacto A y B está dada por,

$$\Lambda_{AB}^2 = (A_{y_0} - B_{y_0})^2 + (A_{z_0} - B_{z_0})^2 \qquad (15)$$

La inclinación α del objeto con respecto al eje y_0 está dada por,

$$\alpha = \arctan\left(\frac{B_{z_0} - A_{z_0}}{B_{y_0} - A_{y_0}}\right) \qquad (16)$$

Asumiendo que el punto B se encuentra fijo en el espacio y la distancia Λ_{AB} depende de la forma del objeto, entonces dado una variación sobre el ángulo de inclinación, α, se debe encontrar el punto A^1 que se acomode correctamente a B sin dejar caer el objeto. Despejando A_{y_0} de la Ec. (15) y reemplazando en la Ec. (16) se obtienen las coordenadas para el punto A^1 como,

$$A^1_{y_0} = B_{y_0} - \sqrt{\Lambda_{AB}^2 - (A^1_{z_0} - B_{z_0})^2} \qquad (17)$$

$$\alpha = \arctan\left(\frac{B_{z_0} - A^1_{z_0}}{\sqrt{\Lambda_{AB}^2 - (A^1_{z_0} - B_{z_0})^2}}\right) \qquad (18)$$

$$A^1_{z_0} = \frac{B_{z_0} + B_{z_0}\tan(\alpha)^2 \pm \sqrt{\Lambda_{AB}^2\tan(\alpha)^2 + \Lambda_{AB}^2\tan(\alpha)^4}}{1 + \tan(\alpha)^2} \qquad (19)$$

La Ec. (19) tiene dos posibles soluciones, aquella solución que produzca una posición para A^1 tal que se cumpla con la inclinación deseada será la seleccionada. De este modo la manipulación se reduce a un problema de encontrar la posición adecuada para cumplir con una variación sobre la inclinación actual mientras se mantiene la posición de uno de los dedos fija, en este caso el dedo F_2, y mientras el otro dedo cambia su posición para mantener fija la distancia entre los puntos de contacto. La dirección de movimiento de los dedos determina la dirección de rotación del objeto, esta dirección debe variarse cuando se excedan los límites del cono de fricción o los límites del espacio de trabajo de los dedos.

El Algoritmo 1 resume los pasos que se siguen para realizar la manipulación con el movimiento de un único dedo. Cómo entradas el necesario conocer la variación deseada $\delta\alpha$ en la inclinación del objeto sujetado con respecto al sistema de referencia. Partiendo de un configuración inicial con los dedos abiertos, se cierran hasta sujetar el objeto; se reconocen las características de la sujeción, es decir, puntos de contacto, A, B, distancia entre los puntos de contacto, Λ_{AB}, y los ángulos, ω_A y ω_B, entre el objeto y la tangente a la superficie de contacto para después verificar que se encuentren dentro del cono de fricción. Con esto se da paso a la manipulación que consiste en calcular la posición a la que debe moverse el dedo F_1 para cumplir con la variación en la inclinación. Este proceso se repite en ambas direcciones de movimiento, cambiando entre ellas cuando se llega al límite del espacio de trabajo de los dedos o al límite del cono de fricción.

Algorithm 1 Manip. moviendo un único dedo

Require: $\delta\alpha$
Cerrar los dedos hasta sujetar el objeto
Leer L_1, L_2 de los sensores
Calcular los puntos A, B
Calcular la distancia Λ_{AB}
Calcular los ángulos ω_A y ω_B entre el objeto y la tangente a la superficie de contacto
repeat
  Calcular A^1 tal que la inclinación del objeto varíe en $\delta\alpha$
  Calcular θ_2 y θ_3 para A^1 usando CI
  Calcular la nueva posición de C usando CD
  Calcular los nuevos ω_A y ω_B
  if ω_A y $\omega_B \in$ cono de fricción **then**
    Mover F_1 a A^1
  else
    Cambiar la dirección de rotación
  end if
until $A^1 \notin$ espacio de trabajo del dedo

CI: Cinemática Inversa / CD: Cinemática Directa

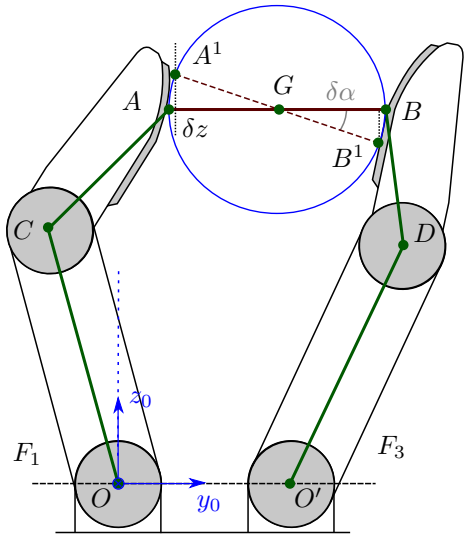


Figura 5: Modelo dos dedos de la SDH2 para realizar la manipulación moviendo los dos dedos.

3.2 Manipulación con dos dedos

Para el análisis del proceso de manipulación mediante el movimiento de dos dedos se considera nuevamente un objeto fino como se muestra en la Figura 5. Los dedos se moverán sobre una trayectoria circular virtual cuyo diámetro está dado por la distancia Λ_{AB} entre los puntos de contacto A y B, el centro de la circunferencia está dado por el punto G cuyas coordenadas están dados por,

$$G_{x_0} = \frac{B_{x_0} - A_{x_0}}{2} + A_{x_0} \tag{20}$$

$$G_{z_0} = \frac{B_{z_0} - A_{z_0}}{2} + A_{z_0} \tag{21}$$

Para una variación $\delta\alpha$ deseada es posible encontrar un δz que sumado a las coordenadas de los puntos de contacto permite calcular sus nuevas posiciones. La relación entre $\delta\alpha$ y δz está dada por,

$$\delta z = \sin(\delta\alpha)\frac{\Lambda_{AB}}{2} \tag{22}$$

$$\delta z = \sqrt{(B^1_{x_0} - B_{x_0})^2 + (B^1_{z_0} - B_{z_0})^2} \tag{23}$$

Además, teniendo en cuenta que los puntos de contacto se mueven sobre una trayectoria circular virtual dada por,

$$(B^1_{z_0} - G_{z_0})^2 + (B^1_{x_0} - G_{x_0})^2 = \left(\frac{\Lambda_{AB}}{2}\right)^2 \tag{24}$$

Es posible calcular las coordenadas del punto B^1, para después aplicar un procedimiento similar al Algoritmo 1. El Algoritmo 2 muestra el procedimiento que se sigue para realizar la manipulación moviendo dos dedos. Básicamente se calcula una posición para el dedo F_2 y el dedo F_1 debe acomodarse para mantener las condiciones de sujeción.

Algorithm 2 Manip. moviendo dos dedos

Require: $\delta\alpha$
Cerrar los dedos hasta sujetar el objeto
Leer L_1, L_2 de los sensores
Calcular los puntos A y B
Calcular la distancia Λ_{AB}
Calcular los ángulos ω_A y ω_B entre el objeto y la tangente a la superficie de contacto
repeat
  Calcular B^1 tal que la inclinación del objeto varíe en $\delta\alpha$
  Calcular θ_4 y θ_5 para A^1 usando CI
  Calcular la nueva posición de D usando CD
  Calcular A^1 ten función de B^1
  Calcular θ_2 y θ_3 para A^1 usando CI
  Calcular la nueva posición de C usando CD
  Calcular los nuevos ω_A y ω_B
  if ω_A y $\omega_B \in$ cono de fricción **then**
    Mover F_1 y F_2 a A^1 y B^1
  else
    Cambiar la dirección de rotación
  end if
until A^1o $B^1 \notin$ espacio de trabajo del dedo

CI: Cinemática Inversa / CD: Cinemática Directa

4 Realimentación de fuerza

Durante la manipulación de objetos reales, cuya forma difiere de los objetos finos usados en el análisis, los puntos de contacto sobre los dedos y las fuerzas aplicadas pueden variar debido a la forma del objeto o al deslizamiento que se produce al girarlo. Así que la distancia entre los puntos de contacto y las fuerzas aplicadas por los dedos tienen que ajustarse para evitar que el objeto caiga o sea sometido a presiones excesivas.

Usando las lecturas de la fuerza aplicada sobre el objeto se puede decidir sobre las variaciones que deben realizarse en la distancia entre los puntos de contacto, si la fuerza aumenta la distancia también debe incrementarse y si la fuerza disminuye los dedos deben cerrarse para buscar mantener la fuerza constante.

El Algoritmo 3 muestra el procedimiento que se sigue para la manipulación de objetos desconocidos usando la realimentación de fuerza. Como entradas se debe establecer una fuerza deseada F^d que se aplicará al objeto, una variación en la inclinación del objeto $\delta\alpha$ para la manipulación, un valor $\delta\theta$ usado como incremento al cerrar los dedos en la sujeción inicial y una variación en la distancia entre los puntos de contacto $\delta\Lambda$ para corregir la fuerza aplicada. El algoritmo consiste en cerrar los dedos variando $\theta_2, \theta_3, \theta_4$ y θ_5 en $\delta\theta$ hasta que se detecte una fuerza aplicada mayor o igual que la fuerza deseada F^d. Una vez que se tiene la sujeción inicial se leen los diferentes parámetros que caracterizan esta sujeción ($L_1, L_2, \theta_2, \theta_3, \theta_4, \theta_5$ y F) y se recalculan los puntos A y B y la distancia Λ_{AB}.

Algorithm 3 Manipulación con realimentación de fuerza

Require: $\delta\theta, \delta\alpha, F^d, \delta\Lambda$

repeat
  Cerrar los dedos cambiando $\theta_2, \theta_3, \theta_4$ y θ_5 en $\delta\theta$
  Leer F
until $(F \geq F^d)$
repeat
  Leer L_1, L_2 de los sensores
  Leer $\theta_2, \theta_3, \theta_4$ y θ_5
  Leer F de los sensores
  Calcular los puntos A y B usando CD
  Calcular la distancia Λ_{AB}
  Ajustar la distancia Λ_{AB} según $\delta\Lambda$
  Calcular δF
  Calcular B^1 tal que la inclinación del objeto varíe en $\delta\alpha$
  Calcular θ_4 y θ_5 para A^1 usando CI
  Calcular la nueva posición de D usando CD
  Calcular A^1 ten función de B^1
  Calcular θ_2 y θ_3 para A^1 usando CI
  Calcular la nueva posición de C usando CD
  Calcular los nuevos ω_A y ω_B
  if ω_A y $\omega_B \in$ cono de fricción **then**
    Mover F_1 y F_2 a A^1 y B^1
  else
    Cambiar la dirección de rotación
  end if
until A^1o $B^1 \notin$ Espacio de trabajo del dedo

CI: Cinemática Inversa / CD: Cinemática Directa

5 Experimentación

Para probar el algoritmo de manipulación usando la realimentación de fuerza se realizaron pruebas con diferentes objetos, a continuación se presenta



(a) Configuración inicial.



(b) Sujeción inicial.



(c) Límite de manipulación girando en sentido horario



(d) Límite de manipulación girando en sentido antihorario

Figura 6: Instantáneas de una ejecución real del algoritmo de manipulación usando realimentación de fuerza.

la configuración usada en uno de los experimentos y los resultados obtenidos. Partiendo de una configuración inicial de los dedos en la que están abiertos para poder sujetar un objeto (Figura 6a), se utilizó una valor de $\delta\theta = 0.25°$ para modificar los valores articulares de la mano de modo que ésta se cerrara e hiciera la sujeción inicial del objeto; la variación deseada de la inclinación del objeto $\delta\alpha$ se fijó en 2° para cada iteración del algoritmo; el valor de la fuerza deseada se obtuvo empíricamente mediante la repetición de sujeciones y la lectura de los sensores para el objeto sujetado; la variación $\delta\Lambda$ de la distancia entre los puntos de contacto se fijó en 0.2 mm para incrementar la fuerza de sujeción y en 1 mm para reducirla. Se asumió un coeficiente de fricción estático de 0.40.

En la Figura 6 se pueden observar algunas instantáneas tomadas durante el proceso de manipulación. La Figura 6a muestra la posición inicial de los dedos antes de sujetar el objeto; en la Figura 6b se observa el objeto sujetado antes de iniciar el proceso de manipulación; la Figura 6c muestra cuando los dedos han llevado al objeto hasta el límite de la manipulación haciendo

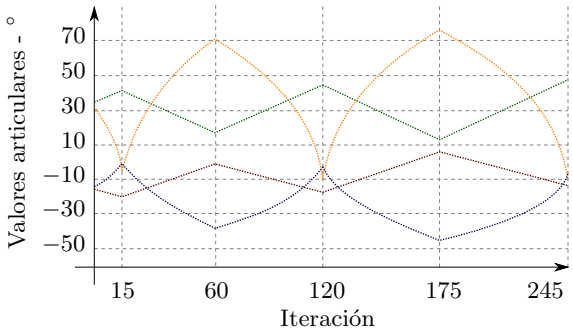


Figura 7: Valores articulares para θ_2 (azul), θ_3 (naranja), θ_4 (rojo) y θ_5 (verde) durante la manipulación.

movimientos en sentido horario y la Figura 6d muestra el límite del proceso de manipulación cuando se hacen los movimientos en sentido antihorario. La figura 7 muestra los valores articulares de los dedos durante la manipulación.

6 Sumario y trabajo futuro

En este trabajo se describió un método de manipulación de objetos desconocidos utilizando los dos dedos acoplados de la mano SDH2. Esta mano está equipada con sensores táctiles que permiten tener realimentación de las fuerzas ejercidas y de la posición de los puntos de contacto. Se planteó un modelo cinemático para los dedos que incluye una articulación virtual para modelar el punto de contacto, y se propuso un método geométrico para el cálculo de la cinemática inversa. La información táctil permite superar problemas como el desconocimiento de la forma del objeto manipulado y los deslizamientos que se pueden presentar durante la manipulación.

Como trabajo futuro se plantea optimizar los movimientos que se hacen con el dedo independiente durante la manipulación de tal forma que se pueda utilizar la mayor parte del espacio de trabajo de la mano, aumentando el rango de movimientos y con ello la destreza. Otro trabajo futuro es el reconocimiento de la forma de las partes del objeto que han contactado con los dedos a partir de la información obtenida durante la manipulación.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Plan Nacional de I+D+I a través de los proyectos DPI2010-15446 y DPI2011-22471.

Referencias

[1] Corrales Ramon, J. A., Torres Medina, F., and Perdereau, V. (2013). "Finger readjustment algorithm for object manipulation based on tactile information." Int. J. Advanced Robotic Systems, 10(9):1–9.

[2] Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd ed.

[3] Drimus, A., et al. (2011). "Classification of rigid and deformable objects using a novel tactile sensor." In *Proc. IEEE Int. Conf. Advanced Robotics*, pp. 427–434.

[4] Iwata, H. and Sugano, S. (2009). "Design of anthropomorphic dexterous hand with passive joints and sensitive soft skins." In *IEEE/SICE Int. Symp.System Integration*, pp. 129–134.

[5] Ma, R. and Dollar, A. (2011). "On dexterity and dexterous manipulation." In *Proc. IEEE Int. Conf. Advanced Robotics*, pp. 1–7.

[6] Meier, M., et al. (2011). "A probabilistic approach to tactile shape reconstruction." IEEE Trans. Robotics, 27(3):630–635.

[7] Moll, M. and Erdmann, M. (2001). "Reconstructing shape from motion using tactile sensors." In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 2, pp. 692–700.

[8] Moll, M. and Erdmann, M. (2002). "Dynamic shape reconstruction using tactile sensors." In *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, pp. 1636–1641.

[9] Odhner, L., Ma, R., and Dollar, A. (2012). "Precision grasping and manipulation of small objects from flat surfaces using underactuated fingers." In *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2830–2835.

[10] Sadigh, M. and Ahmadi, H. (2009). "Safe grasping with multi-link fingers based on force sensing." In *Proc. IEEE Int. Conf. Robotics and Biomimetics*, pp. 1796–1802.

[11] Xue, Z., Zollner, J., and Dillmann, R. (2008). "Dexterous manipulation planning of objects with surface of revolution." In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 2703–2708.

Búsqueda de prensiones óptimas con "force-closure" de objetos 2D articulados con 2 eslabones

Noé Alvarado Tovar
noe.alvarado@upc.edu

Raúl Suárez
raul.suarez@upc.edu

Resumen

Este trabajo propone un procedimiento por medio del cual se puede encontrar una prensión localmente óptima en objetos articulados 2D con 2 eslabones y considerando contactos sin fricción.La superficie de cada eslabón del objeto se representa con un conjunto finito de puntos. El enfoque propuesto encuentra en primer lugar una prensión inicial con force-closure. La medida de calidad considerada en este trabajo es el vector de fuerzas generalizadas mayor que una prensión puede resistir con independencia de la dirección de la perturbación. Este artículo describe cómo encontrar los elementos del vector de fuerzas generalizadas generada por una fuerza aplicada en cualquier eslabón del objeto articulado. El enfoque ha sido implementado y algunos ejemplos ilustrativos se incluyen en el artículo.

Palabras clave: Prensiones con force-closure, medidas de calidad, prensiones óptimas..

1. Introducción

La prensión y manipulación de objetos mediante el empleo de manos robóticas es una campo de gran interés. El objetivo de una prensión es restringir los grados de libertad de un objeto pese a la existencia de alguna fuerza de perturbación externa. En general, una prensión debe satisfacer una de las siguientes propiedades: "*force-closure*" (en lo sucesivo FC, las fuerzas aplicadas por los dedos aseguran la inmovilización del objeto), o "*form-closure*" (las posiciones de los dedos aseguran que el objeto se inmovilice) [1]. Ambas propiedades pueden ser caracterizadas en el espacio de configuraciones del objeto. El espacio de configuraciones de un objeto $2D$ rígido tiene dimensión $m = 3$. Markenscoff [2] y otros, al igual que Mishra y otros [3] muestran que un objeto $2D$ rígido puede ser inmovilizado con $m+1 = 4$ dedos sin fricción. Uno de los problemas más frecuentes es encontrar la configuración de la prensión óptima en base a una de las dos propiedades mencionadas, lo que implica tener que escoger un criterio de calidad para evaluar la prensión. De entre los criterios propuestos, cabe destacar el de Ferrari y Canny [4], quienes tienen en cuenta la máxima fuerza generalizada externa que la prensión puede equilibrar en cualquier dirección.

Varios trabajos abordan el problema de la prensión de objetos 2D, poligonales empleando 3 contactos con fricción [5] [6] y con 4 dedos sin friccón [7] [8], no poligonales con 4 contactos sin fricción [9], además de objetos de forma libre [10] y objetos 3D [11]. Por otra parte, varios trabajos aplican medidas de calidad para la evaluación de la prensión de objetos tanto en 2D como en 3D. Mangialardi y otros [12] proponen un criterio de calidad aplicable para objetos 2D y 3D que determina la prensión óptima que permita garantizar la estabilidad de prensión, mientras se minimizan las fuerzas de prensión necesarias para equilibrar cualquier fuerza externa. Zhu y otros [13] presentan un índice de calidad que mide la magnitud máxima de la fuerza generalizada que puede ser resistida por una prensión, para objetos 2D pero se puede extender para objetos 3D. En [14] se presentan 2 índices que permiten evaluar la fijación de objetos 2D y 3D, uno minimiza la suma de todas las fuerzas normales de contacto y el otro minimiza la máxima fuerza normal de contacto. Zhu y Wang [15] presentan un algoritmo basado en la distancia Q, que lleva cabo no solo la síntesis de prensiones con FC en objetos 3D, sino que además produce una prensión óptima. Zheng y Qian [16] describen un método para evaluar la calidad de una prensión en objetos 3D, la calidad se mide a partir de 2 criterios que son formulados como la distancia entre el origen del espacio de fuerzas generalizadas y dos conjuntos de fuerzas generalizadas. En [17] se propone una medida de calidad para prensiones de objetos 3D definida por la distancia entre el casco convexo (en lo sucesivo CH) de las fuerzas generalizadas aplicadas y el CH de las fuerzas generalizadas de perturbación esperadas sobre el objeto. Roa y otros [18] presentan evaluaciones experimentales y comparaciones de diferentes medidas de calidad, usando datos obtenidos de un objeto sensorizado. En [19] se presenta un trabajo muy reciente en el que se propone un algoritmo que puede encontrar prensiones óptimas en objetos 3D usando ciertas medidas de calidad.

Sin embargo, la mayoría de los trabajos desarrollados en el área de prensión de objetos con manos robóticas se enfocan en la prensión de un solo objeto (2D ó 3D) como los mencionados anteriormente, mientras

hay algunos trabajos que tratan de la prensión y la manipulación de objetos articulados compuestos por n eslabones. Katz y Brock [20] desarrollaron habilidades interactivas de percepción para la manipulación de objetos desconocidos que poseen grados de libertad inherentes (tales como tijeras, alicates, cajones). El robot es capaz de adquirir un modelo de la estructura cinemática del objeto para la manipulación. Zyada y otros [21] presentaron el control para la manipulación no prensil de un objeto con 2 eslabones empleando dos brazos robóticos. Muestran el modelo dinámico para la manipulación del objeto, el análisis para su sujeción estática, y un algoritmo de control para levantarlo en un plano. Aunque los trabajos mencionados anteriormente pueden sujetar y manipular objetos articulados, no muestran un procedimiento sistemático para encontrar un conjunto de puntos en la superficie del objeto que permitan una prensión con FC ni una prueba para comprobar si una prensión dada es FC.

Van der Stappent y otros [22] [23] [24] estudian la inmovilización usando contactos sin fricción de una cadena serie de n polígonos conectados por $n-1$ articulaciones rotacionales, es decir una cadena con un espacio de configuraciones con dimensión $m = n+2$. Determinan el número de contactos necesarios para inmovilizar una cadena y describen un método sistemático para colocar los puntos de contacto en cada objeto de la cadena. Una cadena de n polígonos sin lados paralelos puede ser inmovilizada con $n+2$ contactos, y cualquier cadena con n polígonos puede ser inmovilizada con $n+4$ contactos. Describen la inmovilización robusta(cualquier contacto pude ser perturbado ligeramente sin destruir la inmovilización) de cadenas sin bordes pararalelos y con bordes paralelos con $[\frac{6}{5}(n+2)]$ y $[\frac{5}{4}(n+2)]$ contactos respectivamente [22]. Cuando los polígonos tienen lados paralelos la cadena puede ser inmovilizados con $n+2$ contactos si n es par, y con $n+3$ si n es impar [23]. El trabajo más reciente describe 2 enfoques para inmovilizar cadenas tomando en cuenta los efectos geométricos de primer y segundo orden. Basándose en los efectos de segundo orden, cadenas de $n \neq 3$ polígonos sin lados paralelos pueden ser inmovilizadas con $n+2$ dedos. Por otra parte, al considerar los efectos de primer orden, cualquier cadena de n polígonos pueder ser inmovilizada con $n+3$ dedos [24]. Sin embargo los trabajos mencionados anteriormente relacionados con objetos articulados no presentan la evaluación y búsqueda de la prensión o fijación óptima.

En este trabajo se consideran prensiones generales de objetos articulados 2D con $n = 2$ eslabones, y $m = n+2$ grados de libertad (en adelante *GDL*), considerando $k = m+1 = n+3$ contactos sin fricción. En primer lugar encontrar prensiones con FC, y en segundo lugar encontrar la prensión óptima con FC. Para el primer problema, se comienza con una prensión inicial aleatoria, si ésta no es FC entonces los puntos de contactos se cambian iterativamente para buscar una prensión con FC. Para la optimización de la prensión se inicia con la prensión con FC obtenida del primer problema y luego se busca la prensión más óptima considerando una medida de calidad específica. Los algortimos desarrollados en este trabajo están basados en el trabajo de Roa y Suárez [25] que se extiende aquí para objetos articulados. El vector de fuerzas generalizadas juega un rol importante en la prensión de objetos, por lo que en este trabajo se presenta un procedimiento que permite encontrar una representación adecuada de este vector para objetos articulados con 2 eslabones. La principal contribución de este trabajo es el desarrollo de un procedimiento sistemático para la síntesis de prensiones con FC y la búsqueda de una prensión óptima con FC de un objeto articulado. En este trabajo no se tiene en cuenta el uso de una mano robótica específica como en [26], por lo que el enfoque descrito aquí puede producir prensiones que pueden no ser alcanzadas por una mano robótica en particular, pero, por otro lado, es de uso directo para la fijación de objetos en aplicaciones industriales.

El artículo se estructura de la siguiente manera. La Sección II presenta la descripción general del problema, incluyendo los principales supuestos. La Sección III presenta el procedimiento para encontrar los elementos del vector de fuerzas generalizadas para un objeto articulado con 2 eslabones. En la Sección IV se presenta el algoritmo para la síntesis de prensiones con FC. La Sección V presenta el algoritmo para el proceso de optimización de la prensón con FC. La sección VI muestra algunos ejemplos ilustrativos de la aplicación los enfoques propuestos. Finalmente, las conclusiones y trabajos futuros se presentan en la Sección VII.

2. Definición del problema y asunciones

Considérese un objeto 2D articulado con 2 eslabones, tal como se muestra en la figura 1. Los problemas a ser tratados son los siguientes:

- La búsqueda de un conjunto de puntos de contacto sobre la superficie de los eslabones que permitan una prensión con FC.
- La búsqueda de una prensión con FC óptima a partir de una prensión inicial que cumpla con la propiedad de FC. La optimización se lleva a cabo usando como medida de calidad el vector de fuerzas generalizadas mayor que la prensión puede resistir en cualquier dirección [4][27].

En este trabajo se hacen las siguientes asunciones:

- Los eslabones del objeto articulado están conectados por medio de articulaciones rotacionales.

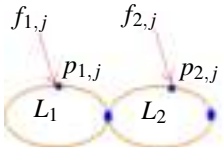


Figura 1: Objeto articulado con 2 eslabones.

- Los eslabones pueden solaparse entre sí.
- Los objetos pueden ser poligonales o no poligonales.
- Las superficie externa de cada eslabón se representa con un conjunto Ω (lo suficientemente grande) de puntos descritos por vectores de posición.
- Cada punto tiene asociada una dirección normal a la superficie del objeto conocida que apunta hacia el interior del eslabón.
- Los puntos de contacto entre los dedos y el objeto se consideran sin fricción

3. Fuerzas generalizadas para objetos articulados

3.1. Fuerzas generalizadas para un objeto rígido

Considérese un sistema de coordenadas situado en el centro de masa del objeto (CM) empleado para describir las posiciones de los puntos de contacto $\boldsymbol{p}_i$ y las fuerzas aplicadas en el objeto. Una fuerza unitaria $\boldsymbol{f}_i$ aplicada en el objeto genera un torque $\tau_i = \boldsymbol{p}_i \times \boldsymbol{f}_i$ con respecto al CM; $\boldsymbol{f}_i$ y τ_i agrupados en un vector de fuerzas generalizadas $\boldsymbol{w}_i = (\boldsymbol{f}_i, \tau_i)^T$. Para prensiones sin fricción, las fuerzas pueden ser aplicadas solo en dirección normal a la superficie del objeto, por lo tanto el vector $\boldsymbol{w}_i$ está dado por:

$$\boldsymbol{w}_i = \begin{bmatrix} \boldsymbol{f}_i \\ \tau_i \end{bmatrix} = f_i \begin{bmatrix} \hat{\boldsymbol{n}}_i \\ \boldsymbol{p}_i \times \hat{\boldsymbol{n}}_i \end{bmatrix} \qquad (1)$$

Una prensión definida por un conjunto decontactos sin fricción $G = \{\boldsymbol{p}_1, ..., \boldsymbol{p}_k\}$ es capaz de aplicar k fuerzas generalizados $\boldsymbol{w}_i$ sobre el objeto, las cuales pueden ser agrupadas en un conjunto $W = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_k\}$. La información en W es suficiente para analizar si G permite o no una prensión con FC.

Un objeto plano en un espacio 2D tiene 3 *GDL*, entonces el vector de fuerzas generalizas es de dimensión 3 y, en ausencia de simetrías rotacionales, 4 contactos son suficientes para asegurar la condición de FC, es decir $G = \{\boldsymbol{p}_1, ..., \boldsymbol{p}_4\}$ permite un conjunto $W = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_4\}$ que expande el espacio de fuerzas generalizadas.

3.2. Fuerzas generalizadas para un objeto articulado serie

Como se mencionó en la Sección 1 un objeto articulado con $n = 2$ eslabones tiene $m = n + 2$ *GDL* y pueder ser sujetado con $k = m+1 = n+3$ contactos sin fricción en los puntos de contacto $G = \{\boldsymbol{p}_{i,j}, i = 1, ...n, j = 1, ..., k_i\}$, donde k_i indica el número de puntos de contacto en un eslabón i (observese que $k = \sum_i k_i$). En esta sección se presentará un procedimiento para obtener las fuerzas generalizadas $\boldsymbol{W}_{i,j}$ m-dimensionales tal que el conjunto de k elementos $\mathscr{W} = \{\boldsymbol{W}_{i,j}, i = 1, ..., n, j = 1, ..., k_i\}$ permita determinar si el conjunto G puede producir o no una prensión con FC mediante el uso de un procedimiento análogo al utilizado para el caso de un objeto rígido utilizando W. El vector de fuerzas generalizadas generado por la aplicación de una fuerza en un eslabón del objeto articulado se deduce a partir del análisis general de una cadena cinemática abierta. La figura 2 muestra un robot virtual con $n+2$ eslabones, el cual contiene el objeto articulado con 2 eslabones y los elementos auxiliares empleados en el desarrollo del procedimiento. El procedimiento propuesto en esta sección es similar al empleado por [28].

La siguiente nomencaltura básica se emplea en este artículo:

L_i: Eslabón i del robot virtual, $i = 1, \ldots, n$.

q_i: Posición de la articulación i.

$\boldsymbol{p}_{i,j}$: Punto de contacto j en el eslabón L_i, $i = 1, \ldots, n, j = 0, \ldots, k_i$ donde k_i es el número de puntos de contacto en el eslabón L_i.

$\boldsymbol{f}_{i,j}$: Fuerza j que se aplica sobre el eslabón L_i en el punto de contacto $\boldsymbol{p}_{i,j}$.

El vector de fuerzas generalizadas se obtiene de la siguiente manera.

Considérese un robot virtual de $n+2$ eslabones, donde el primer y segundo eslabón son virtuales y el resto de ellos son los equivalentes al objeto articulado a ser sujetado. Las primeras 3 articulaciones no son reales, sin embargo son útiles para el desarrollo del modelo. La primera y segunda articulación son prismáticas y la tercera es rotacional, la articulación restante es la del objeto articulado. La figura 2 muestra el esquema general con el robot virtual.

Se calcula el Jacobiano J_i para cada eslabón L_i del robot virtual para relacionar las fuerzas aplicadas en el eslabón con las de las articulaciones del robot virtual bajo condiciones de equilibrio.

Tomando en cuenta que las 3 primeras articulaciones son virtuales, podemos considerarlas en la posición çero", es decir asumimos sin pérdida de generalidad que $q_1 = q_2 = q_3 = 0$, permitiendo la simplificación de los Jacobianos J_i.

El Jacobiano J_i permite relacionar las fuerzas externas

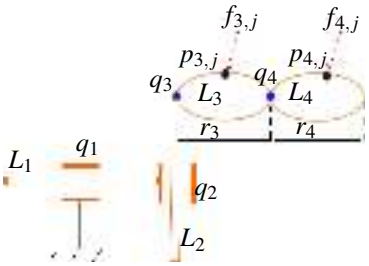


Figura 2: Esquema del robot virtual, donde $L_1 \dots L_4$ son los eslabones totales del robot, y $L_3 \dots L_4$ los eslabones del objeto articulado.

y momentos aplicados en cada eslabón L_i con los torques y fuerzas necesarios en las articulaciones para una condición de equilibrio. La figura 2 muestra una fuerza $\mathbf{f}_{i,j}$ aplicada en cada eslabón L_i del objeto articulado, que puede expresarse como:

$$\boldsymbol{f}_{i,j} = \begin{bmatrix} f_{x_{i,j}} & f_{y_{i,j}} \end{bmatrix}^T \tag{2}$$

La fuerza $\boldsymbol{f}_{i,j}$ y el momento $M_{f_{i,j}} = \boldsymbol{p}_{i,j} \times \boldsymbol{f}_{i,j}$ que ésta produce son agrupados en el vector $\boldsymbol{w}_{i,j}$:

$$\boldsymbol{w}_{i,j} = \begin{bmatrix} f_{x_{i,j}} & f_{y_{i,j}} & M_{f_{i,j}} \end{bmatrix}^T \tag{3}$$

El torque que se aplica en la articulación k para equilibrar el efecto de una fuerza $\boldsymbol{f}_{i,j}$ aplicada en el eslabón L_i es:

$$\tau_{k,w_{i,j}} = V_j[k] \tag{4}$$

donde $V_j[k]$ es el elemento k del vector columna $\mathbf{V}_j$ dado por:

$$\mathbf{V}_j = J_i^T \boldsymbol{w}_{i,j} \tag{5}$$

y el torque requerido en la articulación k para equilibrar todas las fuerzas $\boldsymbol{f}_{i,j}$ aplicadas en el eslabón L_i es

$$\tau_{k,w_i} = \sum_{j=1}^{k_i} V_j[k] \tag{6}$$

Finalmente, el torque total en la articulación k para balancear todas las fuerzas $\boldsymbol{f}_{i,j}$ que actúan en el robot virtual es

$$\tau_k = \sum_{i=1}^{n} \sum_{j=1}^{k_i} V_j[k] \tag{7}$$

Las fuerzas $\boldsymbol{f}_{i,j}$ que actúan en el robot virtual incluyen las fuerzas externas de perturbación y las fuerzas aplicadas por los dipositivos de prensión. Debido a que se desea inmovilizar el objeto articulado, el torque total en cada articulación debe ser nulo (considerando perturbaciones y fuerzas aplicacadas por los dedos), entonces $\tau_k = 0$.

El torque total τ_k se obtiene a partir de (7), igualando a cero para cada articulación, resulta:

$$\begin{array}{rcccl} \tau_1 & = & \sum_j f x_{3,j} + \sum_j f x_{4,j} & = & 0 \\ \tau_2 & = & \sum_j f y_{3,j} + \sum_j f y_{4,j} & = & 0 \\ \tau_3 & = & \sum_j M_{f_{3,j}} & = & 0 \\ \tau_4 & = & \sum_j M_{f_{4,j}} & = & 0 \end{array} \tag{8}$$

Ahora, es posible considerar un espacio de fuerzas generalizadas $\mathscr{W}$ definido por $\{\tau_1, \tau_2, \tau_3, \tau_4\}$ para el objeto articulado, donde las fuerzas generalizadas $\boldsymbol{W}_{1,j}\boldsymbol{W}_{2,j}\boldsymbol{W}_{3,j}, \boldsymbol{W}_{4,j}$ generadas por las fuerzas $\boldsymbol{f}_{1,j}, \boldsymbol{f}_{2,j}, \boldsymbol{f}_{3,j}, \boldsymbol{f}_{4,j}$ son definidas como:

$$\boldsymbol{W}_{1,j} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{W}_{2,j} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{W}_{3,j} = \begin{bmatrix} f x_{3,j} \\ f y_{3,j} \\ M_{f_{3,j}} \\ 0 \end{bmatrix} \quad \boldsymbol{W}_{4,j} = \begin{bmatrix} f x_{4,j} \\ f y_{4,j} \\ 0 \\ M_{f_{4,j}} \end{bmatrix} \tag{9}$$

Tomando en cuenta que las articulaciones virtuales realmente no existen y las fuerzas generalizadas $\boldsymbol{W}_{1,j}$ y $\boldsymbol{W}_{2,j}$ en el robot virtual son siempre nulas, para simplificar la nomenclatura, considerando solo los eslabones del objeto real, podemos referirmos a ellos directamente como L_1 y L_2, y por lo tanto los subíndices de las ecuaciones anteriores que indican el enlace, en la práctica son desplazados en un valor -2, es decir un punto de contacto $\boldsymbol{p}_{3,j}$ en el eslabón 3 del robot virtual, es ahora el punto $\boldsymbol{p}_{1,j}$ del objeto articulado real (figura. 1), y el vector de fuerza generalizado $\boldsymbol{W}_{3,j}$ aplicado en el eslabón 3 del robot virtual, es ahora el vector de fuerzas generalizado $\boldsymbol{W}_{1,j}$ aplicado en el objeto real.

Vale la pena observar que la dimensión de $\mathscr{W}$ es $n+2$ y que $\boldsymbol{W}_{i,j}$ tiene por lo tanto $n+2$ componentes; sin embargo, cada $\boldsymbol{W}_{i,j}$ tiene solo 3 componentes independientes.

3.3. Test de Force-closure

La condición necesaria y suficiente para la existencia de una prensión con FC es que el origen del espacio de fuerzas generalizadas esté dentro del CH de las fuerzas generalizadas primitivas de contacto [29][3]. Esta condición se verifica mediante técnicas de programación lineal [30][31]. La prueba empleada en este trabajo está basada en el siguiente Lema, derivado de [25].

Lema. Sea G una prensión con un conjunto $W = \{\boldsymbol{W}_{i,j}, i = 1,...,n, j = 1,...,k_i\}$ de fuerzas generalizadas primitivas de contacto para un objeto articulado con $m = k-1$ *GDL*, y sea H_l, $l = 1,...,k$, cada hiperplano definido en el espacio de fuerzas generalizadas por el conjunto de puntos $W - \{\boldsymbol{W}_{i,j}\}$, con $i = 1,...,n, j = 1,...,k_i$, y P el centroide de W. P y el origen O del espacio de fuerzas generalizadas deben estar en el mismo lado de cada hiperplano H_l a fin de que la prensión G sea FC. $\diamond$

Ejemplo: El Lema se ilustra con un hipotético espacio de fuerzas generalizadas 2D, tal como se muestra en la figura 3. Dada una prensión con un conjunto de fuerzas generalizadas $W = \{\boldsymbol{W}_{1,1}, \boldsymbol{W}_{2,1}, \boldsymbol{W}_{3,1}\}$, los hiperplanos H_l se definen por los siguientes conjuntos de puntos:
H_1 se define por $W - \{\boldsymbol{W}_{1,1}\} = \{\boldsymbol{W}_{2,1}, \boldsymbol{W}_{3,1}\}$
H_2 se define por $W - \{\boldsymbol{W}_{2,1}\} = \{\boldsymbol{W}_{1,1}, \boldsymbol{W}_{3,1}\}$
H_3 se define por $W - \{\boldsymbol{W}_{3,1}\} = \{\boldsymbol{W}_{1,1}, \boldsymbol{W}_{2,1}\}$
La prensión mostrada en la figura 3a satisface la condición de FC ya que O y P están en el mismo

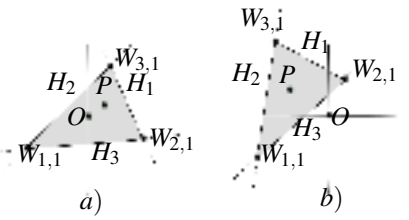


Figura 3: Prueba de FC en un hipotético espacio de fuerzas generalizadas 2D; a)prensión con FC, b)prención sin FC

Algoritmo 1 Generar prensión con FC

Salida: : Prensión G^k con FC
1: Generar una prensión inicial G^k, $k = 1$.
2: **repetir**
3: Calcular el conjunto de fuerzas generalizadas correspondientes $W^k = \{W_{i,j}, i = 1,...,n, j = 1,...,k_i\}$.
4: Determinar el subconjunto G_R^k con los puntos a reemplazar de G^k.
5: Generar el subconjunto Ω_C^k con los puntos - candidatos para reemplazar uno de los puntos en G_R^k.
6: Obtener una prensión auxiliar G_{aux} reemplazando un punto de G_R^k por un punto de Ω_C^k.
7: Actualizar el contador $k = k + 1$.
8: G^k=G_{aux}.
9: **mientras** G^k no sea una prensión con FC
10: **retornar** G^k

lado de cada hiperplano H_l. Sin embargo, la prensión mostrada en la figura 3b falla, ya que O y P están en diferentes lados del hiperplano H_3.

4. Grasp Synthesis

Considerando los desarrollos de la sección anterior, la idea principal del algoritmo descrito en esta sección es una generalización del presentado en [25] considerando que el espacio de fuerzas generalizadas puede tener cualquier dimensión, no solo 3 o 6 para objetos rígidos 2D o 3D respectivamente. El algoritmo genera una prensión G^1 seleccionando k puntos aleatorios del conjunto Ω que describe la superficie del objeto, luego se calcula el conjunto W^1 con las fuerzas generalizas y se verifica si los puntos permiten una prensión con FC. Si G^1 no permite una prensión con FC, entonces se lleva a cabo la busqueda de nuevos puntos de contacto, tomando en cuenta los hiperplanos de separación en el espacio de fuerzas generalizadas que definen los puntos candidatos que reemplacen uno de los puntos actuales en G^1. Este procedimiento se repite iterativamente hasta que se encuentre una prensión con FC. El algoritmo 1 detalla estos pasos.

Si la prensión G^k no supera el test de FC, el procedimiento de búsqueda (pasos 4 al 8) iterativamente trata de mejorar la prensión al ir cambiando uno de los puntos en G^k. En el paso 4 el

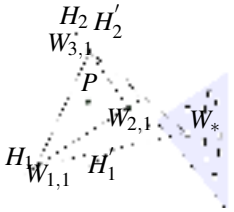


Figura 4: Proceso de busqueda para encontrar una prensión con FC en un hipotético espacio 2D de fuerzas generalizadas

subconjunto G_R^k está formado por todos las fuerzas generalizadas de W que petenezcan simultaneamente al mayor número de hiperplanos H_i que produzcan la falla en la prueba de FC, es decir aquellas que no satisfagan la condición del Lema 1. Si solo hay un hiperplano crítico entonces G_R^k incluye todos los puntos que definen dicho hiperplano. La figura 4 muestra un ejemplo hipotético en 2D (observese que el espacio real de fuerzas generalizadas es 4-dimensional), la prensión G produce las fuerzas generalizadas $W = \{W_{1,1}, W_{2,1}, W_{3,1}\}$ que no es FC, siendo H_1 y H_2 los hiperplanos que producen el fallo de la prueba de FC, entonces el conjunto de posibles puntos a ser reemplazados es G_R^k que incluye el punto $p_{2,1}$ que produce la fuerza generalizada $W_{2,1}$.

En el paso 5 el subconjunto Ω_C^k con los puntos candidatos para reemplazar uno de los de G_R^k se determinan usando hiperplanos H_i' que pasan por el origen y son paralelos a los hiperplanos críticos H_i. Los puntos candidatos a ser usados para la sustición son aquellos que simultaneamente se encuentren en el lado opuesto del punto P con respecto al hiperplano H_i'. En la figura 4 los puntos que producen las fuerzas generalizadas que se encuentran en el área gris determinada por los hiperplanos H_1' y H_2' pertenecen a Ω_C^k.

En el paso 6 uno de los puntos en G_R^k se reemplaza por un punto aleatorio de Ω_C^k produciendo una fuerza generalizada W_*. Se obtiene así una prensión auxiliar G_{aux}, tal como se muestra en la figura 4, produciendo el conjunto de fuerzas generalizadas $\{W_{1,1}, W_*, W_{3,1}\}$. Para la prensión candidata se calculan el centroide P^* y la distancia $\left|\overline{P^*O}\right|$. Si la relación $\left|\overline{P^*O}\right| < \left|\overline{P^kO}\right|$ se satisface para la prensión G^*, entonces se acepta esta prensión, y el correspondiente W_* se emplea para el reemplazo. Si todos los puntos en G_R^k han sido verificados y ninguno de ellos reduce la distancia $\left|\overline{P^kO}\right|$, la selección es la prensión candidata G^* que tenga la menor distancia $\left|\overline{P^*O}\right|$.

Finalmente, en los pasos 7 al 8, el contador k se actualiza y el punto seleccionado se incluye en la nueva prensión G^k=G_{aux}.

5. Prensión óptima

En esta sección se presenta el algoritmo que optimiza la prensión incial con FC obtenida con el Algoritmo 1 descrito en la Sección 4. La idea principal del algoritmo descrito aquí es una generalización derivada del presentado en [25], pero aplicado a un objeto articulado con 2 eslabones. La calidad de la prensión se mide mediante el criterio de la bola máxima [4][27]. Esta medida de calidad indica la mayor fuerza generalizada de perturbación que la prensión puede resistir en cualquier dirección. Geométricamente, la calidad es equivalente al radio de la mayor bola centrada en el origen del espacio de fuerzas generalizadas y totalemente contenida en el $CH(W)$. Esta medida de calidad es igual a la distancia desde el origen del espacio de fuerzas generalizadas a la cara más cercana del $CH(W)$ [4].

La búsqueda de la prensión óptima inicia con una prensión inicial G^1 obtenida con el Algoritmo 1. El algoritmo busca la cara de $CH(W^1)$ más cercana al origen, en lo sucesivo F_Q, y calcula su distancia al origen, es decir la calidad de la prensión actual Q^1. El paso siguiente del algoritmo es construir el subconjunto Ω_C^k con los puntos candidatos para reemplazar uno de los puntos de F_Q; para encontrar el subconjunto Ω_C^k se emplea un hiperplano de separación, de manera similar a como se hizo en el Algoritmo 1. Cada punto candidato en Ω_C^k genera varias prensiones candidatas G^*. Se elige un punto candidato a la vez y las prensiones candidatas correspondientes se verifican con la prueba de FC descrita en el Lema 1. Para cada prensión candidata con FC, se calcula la medida de calidad; si la calidad es mayor que Q^1, se selecciona la correspondiente prensión y se actualizan G, W y Q. El procedimiento continúa hasta que no se puede lograr ninguna mejora. El algoritmo 2 describe estos pasos.

La prensión inicial en el procedimiento de optimización es una prensión con FC, por lo tanto $O \in CH(W)$. La calidad de la prensión Q está fijada por F_Q, la cara de $CH(W)$ más cercana al origen. El subconjunto Ω_C^k con los puntos que pueden mejorar la calidad de la prensión se define usando H_Q, el hiperplano que contiene a F_Q. Los puntos candidatos son aquellos que se encuentran en el semi-espacio definido por H_Q que no contiene el origen, es decir H_Q^+ como se observa en la figura 5.

Para contactos sin fricción, las prensiones candidatas G_i^*, $i = 1,...,4$, se forman usando un punto candidato W^* para reemplazar cada uno de los vértices de la cara F_Q. La verificación de FC empleando el Lema 1 se lleva a cabo para cada una de las prensiones candidatas para cada punto candidato. Para las prensiones candidatas con FC, se calcula la calidad de la prensión Q^*; si una prensión candidata cumple

Algoritmo 2 Busqueda de una prensión óptima con FC

Salida: : Prensión óptima G^k con FC

1: Encontrar una prensión inicial con FC G^k, $k = 1$ usando el algortimo 1
2: **repetir**
3: Formar el conjunto de fuerzas generalizadas W^k.
4: Determinar la cara más cercana del $CH(W^k)$ al origen, en lo sucesivo FQ. La distancia desde el origen a FQ es la medidad de calidad actual Q^k.
5: Construir el subconjunto Ω_C^k (conjunto con m puntos) con los puntos candidatos que pueden provocar una mejora en la prensión.
6: $i = 1$
7: **repetir**
8: **repetir**
9: Generar prensiones candidatas G_i^*
10: $i = i + 1$
11: **mientras** (G_i^* no sea una prensión con FC)
12: Calcular la medidad de calidad Q^* para la prensión G_i^*
13: **mientras** ($Q^* < Q^k$)
14: Actualizar $k = k + 1$ y $G^k = G_i^*$
15: **mientras** ($i <= m$)
16: **retornar** G^k

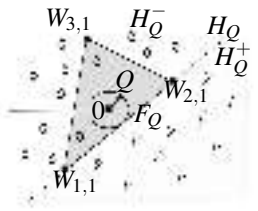


Figura 5: Selección del subconjunto Ω_C^k con los puntos candidatos (triángulos en el área sombreada) que pueden mejorar la calidad de la prensión; en este ejemplo $F_Q = \overline{W_{1,1}W_{2,1}}$

$Q^* > Q^k$, la prensión candidata se convierte en la nueva prensión G^K. La figura 6 ilustra 3 posibles casos relacionados con prensiones candidatas; el caso(a) es una prensión no fiable ya que pierde la condición de FC, el caso (b) se descarta porque la calidad de la prensión es menor que la calidad de la prensión anterior, y el caso (c) es una prensión que mejora la calidad de la prensión anterior, por lo tanto se convierte en la prensión para la siguiente iteración.

6. Ejemplos Numericos

Ejemplo 1: Objeto articulado con 2 elipses. La figura 7a muestra la prensión inicial con FC obtenida después de 3 iteraciones. La figura 7b muestra la prensión localmente óptima después de 16 iteraciones. La figura 7c grafica la variación de la calidad de la prensión hasta encontrar la prensión óptima.

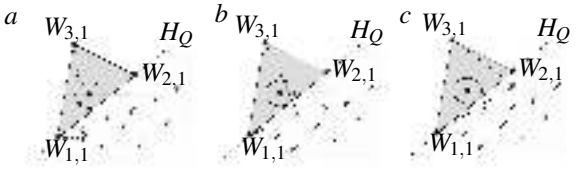
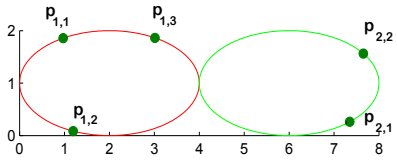
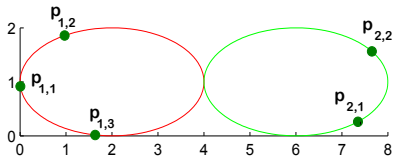


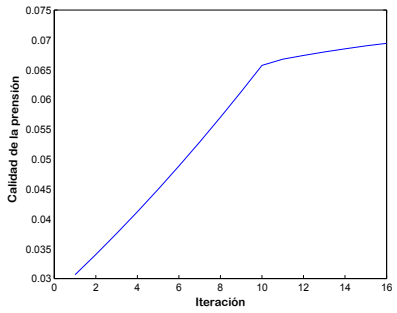
Figura 6: Posibles casos para una prensión en la optimización: (a) prensión candidata sin FC; (b) prensión candidata descartada; (c) prensión candidata factible.



(a) Prensión con FC inicial



(b) Prensión óptima con FC



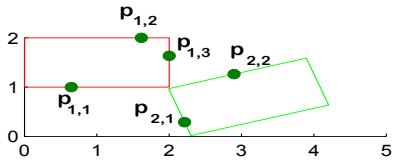
(c) Medida de calidad

Figura 7: Prensión óptima para el ejemplo1

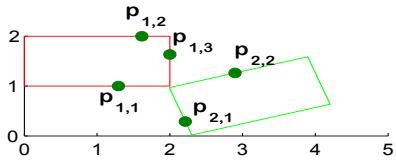
Ejemplo 2: Objeto articulado con 2 rectángulos. La figura 8a muestra la prensión inicial con FC obtenida después de 8 iteraciones. La figura 8b muestra la prensión localmente óptima después de 33 iteraciones. La figura 8c grafica la variación de la calidad de la prensión hasta encontrar la prensión óptima.

7. Sumario y Trabajo Futuro

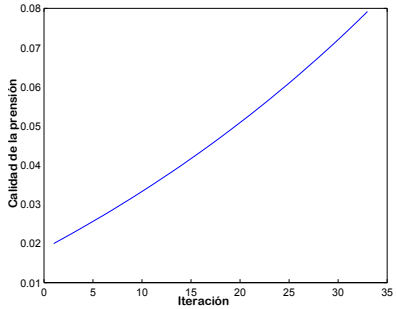
En este trabajo se propone un enfoque para obtener una prensión localmente óptima en objetos articulados $2D$ con 2 eslabones considerando contactos sin fricción. El enfoque tiene 2 etapas: la primera lleva a cabo la síntesis de prensiones que tengan FC. La segunda, encuentra una prensión óptima a partir de la prensión obtenida en la primera etapa. La medida de calidad empleada en el proceso de optimización es el vector de fuerzas generalizadas mayor que la prensión puede resistir en cualquier dirección. Como trabajo



(a) Prensión con FC inicial



(b) Prensión óptima con FC



(c) Medida de calidad

Figura 8: Prensión óptima para el ejemplo2

futuro se plantea la generalización del enfoque para objetos articulados $3D$ con contactos con fricción y sin fricción.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Plan Nacional de I+D+I a través de los proyectos DPI2010-15446 y DPI2011-22471.

Referencias

[1] A. Bicchi. On the closure properties of robotic grasping. *Int. J. Robotics Res.*, 14(4):319–44, 1995.

[2] X. Markenscoff and C. H. Papadimitriou. The Geometry of Grasping. *Int. J. Robotics Res.*, 9(1):61–74, 1990.

[3] B. Mishra, J.T Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):541–558, 1987.

[4] C Ferrari and J Cany. Planning optimal grasps. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2290–2295, 1992.

[5] Y.C. Park and G.P. Starr. Grasp synthesis of polygonal objects. In *Proc. IEEE Int. Conf. Robotics and Automation*, volume 3, pages 1574–1580, 1990.

[6] J. Ponce, D. Stam, and B. Faverjon. On computing three-finger force-closure grasps of polygonal objects. *IEEE Trans. Robotics and Automation*, 11(6):868–881, 1995.

[7] J. Cornellà and R. Suárez. On 2D 4-finger frictionless optimal grasps. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, volume 3, pages 3680–3685, 2003.

[8] J. Cornellà and R. Suárez. Efficient determination of four-point form-closure optimal constraints of polygonal objects. *IEEE Trans. Robotics*, 6(1):121–130, 2009.

[9] J. Cornellà and R. Suárez. On computing form-closure grasps/fixtures for non-polygonal objects. In *Proc. IEEE Int. Symp. Assembly and Task Planning*, pages 138–143, 2005.

[10] N. Niparnan and A. Sudsang. Computing All Force-Closure Grasps of 2D Objects from Contact Point Set. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 1599–1604, 2006.

[11] G. Dini and F. Failli. Planning grasps for industrial robotized applications using neural networks. *J. Robotics and Comp. Int. Manuf.*, 16:451–463, 2000.

[12] L. Mangialardi, G. Mantriota, and A. Trentadue. A three-dimensional criterion for the determination of optimal grip points. *J. Robotics and Comp. Int. Manuf.*, 12(2):157–167, 1996.

[13] X. Zhu, H. Ding, and H. Li. A Quantitative Measure for Multi-fingered Grasps. In *Proc. IEEE Int. Conf Adv. Intell. Mechat.*, volume 1, pages 213–219, 2001.

[14] C. Xiong, Y. Li, Y.K Rong, and Y. Xiong. Qualitative analysis and quantitative evaluation of fixturing. *J. Robotics and Autonomus Syst.*, 18:335–342, 2002.

[15] X. Zhu and J. Wang. Synthesis of Force-Closure Grasps on 3D Objects Based on the Q Distance. *IEEE Trans. Robotics and Automation*, 19(4):669–679, 2003.

[16] Y. Zheng and WH. Qian. Improving grasp quality evaluation. *J. Robotics and Autonomus Syst.*, 57:665–673, 2009.

[17] H. Jeong and J. Cheong. Evaluation of Grasps for 3D Objects with Physical Interpretations using Object Wrench Space. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 604–609, 2010.

[18] M. Roa, R. Koiva, and C. Castellini. Experimental Evaluation of Human Grasps Using a Sensorized Object. In *Proc. IEEE Int. Conf. Biom. Robotics and Biomech.*, pages 1662–16683, 2012.

[19] Y. Zheng. An Efficient Algorithm for a Grasp Quality Measure. *IEEE Trans. Robotics*, 29(2):579–585, 2013.

[20] D. Katz and O. Brock. Manipulating articulated objects with interactive perception. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 272–277, 2008.

[21] Z. Zyada, Y. Hayakawa, and S. Hosoe. Model-based control for nonprehensile manipulation of a two-rigid-link object by two cooperative arms. In *Proc. IEEE Int. Conf. Robotics and Biomimetics*, pages 472–477, 2010.

[22] J.S Cheong, Ken Goldberg, and A.F. van der Stappen. Fixturing hinged polygons. In *Proc. IEEE Int. Conf. Robotics and Automation*, volume 1, pages 876–881, 2002.

[23] J.S Cheong, A.F Stappen, K. Goldberg, M.H. Overmars, and E. Rimon. Immobilizing hinged polygons. *Int. J. Comp. Geometry & Applications*, 17(1):45–69, 2007.

[24] E. Rimon and A.F Stappen. Immobilizing 2-D Serial Chains in Form-Closure Grasps. *IEEE Trans. Robotics*, 28(1):32–43, 2012.

[25] M. Roa and R. Suárez. Finding locally optimum force-closure grasps. *J. Robotics and Comp. Int. Manuf.*, 25(3):536–544, 2009.

[26] G. Carbone and A. GonzÃ¡lez. A numerical simulation of the grasp operation by LARM Hand IV: A three finger robotic hand. *J. Robotics and Comp. Int. Manuf.*, 27(2):450–459, 2011.

[27] M. Roa and R. Suárez. Medidas de Calidad para la prensión de objetos. *Rev. Iberoam Automat Inform Ind*, 5(1):66–82, 2008.

[28] J. Schutter and T. Laet. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *Int. J. Robotics Res.*, 26(5):1–33, 2007.

[29] R.M Murray, Z. Li, and S.S Sastry. *A mathematical introduction to robotic manipulation*. CRC PressINC, 1994.

[30] YH Liu, ML Lam, and Dan Ding. A complete and efficient algorithm for searching 3-D form-closure grasps in the discrete domain. *IEEE Trans. Robotics*, 20(4):805–816, 2004.

[31] H. Asada and M. Kitagawa. Kinematic analysis and planning for form closure grasps by robotic hands. *J. Robotics and Comp. Int. Manuf.*, 5(4):293–299, 1989.

SOLUCIÓN AL PROBLEMA DE SLAM EMPLEANDO SGD CON IMÁGENES OMNIDIRECCIONALES

David Valiente, Arturo Gil, Miguel Juliá, Lorenzo Fernández, Óscar Reinoso.
Departamento de Ingeniería de Sistemas y Automática.
Universidad Miguel Hernández, Avda. de la Universidad s/n, Elche (Alicante).
dvaliente@umh.es, arturo.gil@umh.es, mjulia@umh.es, l.fernandez@umh.es, o.reinoso@umh.es

Resumen

Este trabajo aborda el problema de SLAM (Simultaneous Localization and Mapping)*, y presenta una solución basada en el algoritmo SGD* (Stochastic Gradient Descent)*. En concreto se plantea un modelo basado en imágenes omnidireccionales. Entre las aplicaciones de la robótica móvil nunca ha sido evaluado el funcionamiento de SGD junto con la información del entorno ofrecida por este tipo de imágenes. Este artículo sugiere el empleo de SGD en un sistema de SLAM, el cual explota los beneficios proporcionados por una cámara omnidireccional. Se presentan diversas mejoras sobre el modelo general de SGD para lograr su adaptación al caso de observación omnidireccional, cuya naturaleza es angular y sin escala. Esta nueva propuesta basada en SGD reduce los indeseados efectos producidos por los errores no lineales introducidos en el sistema, los cuales comprometen la convergencia de los modelos tradicionales de estimación, como los filtros. Confiamos en un modelo de construcción del mapa eficiente, constituido por un conjunto reducido de vistas omnidireccionales. En contraste con los esquemas tradicionales de SGD que únicamente procesan una medida de observación por iteración del proceso, proponemos una estrategia que considera varias medidas simultáneamente, con el objetivo de mejorar la velocidad de convergencia de la estimación. Presentamos diferentes conjuntos de experimentos con la intención de validar este nuevo modelo propuesto basado en SGD con medidas de observación omnidireccionales. Finalmente, comparamos dichos resultados con los obtenidos empleando un modelo tradicional de SGD, con la finalidad de demostrar los beneficios esperados en términos de eficiencia.*

Palabras clave: SLAM, SLAM visual, imagen omnidireccional, SGD.

1. INTRODUCCIÓN

El problema de SLAM es un aspecto fundamental en el ámbito de la robótica móvil ya que la representación del entorno es esencial para propósitos de navegación. Construir un mapa implica un proceso complejo puesto que debe realizarse de manera incremental, y además debe calcularse simultáneamente la localización del robot dentro del mismo. Obtener coherencia en el mapa generado es problemático cuando existe ruido añadido por los sensores, el cual afecta gravemente a la estimación de dicho mapa y del camino seguido por el robot. Tradicionalmente, los esquemas de SLAM se han clasificado en base al modelo de representación del mapa, al algoritmo de estimación y al tipo de sensor encargado de adquirir información del entorno. El uso de sensores láser [1] es muy extendido para la obtención de representaciones como los mapas de ocupación [13] y los mapas basados en marcas [10]. Recientemente, el uso de información visual ha establecido una tendencia hacia la utilización de cámaras digitales. Estos sensores ofrecen grandes mejoras con respecto a los tradicionales sensores láser, como la cantidad de información ofrecida del entorno, codificada en una única imagen. Pese a no ser tan precisos, son más ligeros, baratos y eficientes en cuanto a consumo se refiere. Respecto a la configuración posible, existen ejemplos basados en el par estéreo [5], donde se extraen medidas referidas a un conjunto de marcas 3D, o bien otros trabajos [2, 9] donde se emplea una única cámara para calcular la parametrización inversa que determina marcas 3D, debido a la falta de una medida de profundidad. Otras propuestas [8] han combinado dos cámaras omnidireccionales para aprovechar las ventajas del modelo estéreo junto con el amplio campo de visión que proveen las imágenes omnidireccionales para codificar gran cantidad de información por imagen. Además del sensor y del modelo de representación, el tipo de algoritmo es básico para la generación de una estimación válida para el problema de SLAM. Entre los más extendidos se encuentran el EKF (*Extended Kalman Filter*) [4], el filtro de partículas *Rao-Blackwellized* [10] y los algoritmos $offline$ como SGD [7].

Por tanto, el éxito de un modelo eficiente de SLAM depende directamente de la correcta combinación de los anteriores parámetros. Existe un gran número de trabajos que tratan con modelos de representación basados en la obtención de marcas 3D en un sistema de representación general [4, 5, 3, 2]. Se fundamentan en la capaci-

dad del filtro EKF para converger en una solución válida para el problema de SLAM. Más recientemente [14], propone un modelo de representación diferente, donde el mapa se constituye con la posición y orientación de un conjunto reducido de vistas del entorno, siendo mucho más compacta la representación final. El hecho más destacable es la capacidad de obtener un movimiento relativo entre dos vistas, que coincide con dos posiciones del robot, y por tanto permite su localización en el mapa. Sin embargo, todos estos esquemas basados en EKF presentan grandes problemas de convergencia ante la presencia de errores no gaussianos, normalmente introducidos por las medidas de observación y que involucran una gran problemática en el proceso de asociación de datos [11]. En este sentido, una observación visual como es la omnidireccional es muy susceptible de generar no linealidades causantes de estos errores.

La problemática asociada con la convergencia del EKF nos hace sugerir el uso de SGD para contrarrestar dichos efectos dañinos. El objetivo principal trata de beneficiarse del uso de imágenes omnidireccionales, y por ello el esquema tradicional de SGD ha de ser adaptado y mejorado para trabajar con dichas imágenes. Sin embargo, la convergencia de la solución sigue sin ser trivial para un algoritmo SGD cuando existe ambigüedad en la escala. Por tanto, también proponemos una nueva estrategia para procesar medidas de observación de manera simultánea, a diferencia del SGD tradicional. Se han tenido en cuenta ciertos detalles de la implementación que evitan que esta nueva estrategia pueda causar un mayor coste computacional o cuellos de botella. El artículo se estructura como sigue: La Sección 2 plantea el problema de SLAM dentro del contexto de aplicación. A continuación, en la Sección 3 se detallan las especificaciones del algoritmo SGD. La Sección 4 describe las contribuciones a destacar de esta propuesta. En la Sección 5 se muestran los resultados experimentales que evalúan la validez del modelo y sus beneficios esperados. Finalmente, la Sección 6 realiza un análisis de los resultados con el fin de extraer conclusiones.

2. SLAM

De un modelo de SLAM visual se espera fiabilidad en la estimación calculada de la posición del robot dentro de un cierto entorno, cuya localización también debe quedar determinada. En este trabajo se emplea un mapa compuesto por un conjunto de imágenes capturadas en distintas posiciones por las que el robot explora el entorno, nombradas como vistas. Dichas vistas no representan ninguna marca física ya que en realidad constan de una imagen omnidireccional, de su posición de captura $x_l = (\mathrm{x}_l, \mathrm{y}_l, \theta_l)$, y de un conjunto de puntos significativos extraídos de la propia vista. Tal configuración permite sacar el máximo partido a la información entregada por una única imagen, gracias a su amplio campo de visión. Como consecuencia se consigue una importante reducción en el número de variables a estimar en la solución.

La posición del robot queda definida como $x_v = (\mathrm{x}_v, \mathrm{y}_v, \theta_v)^T$. Cada vista $n \in [1, \ldots, N]$ se determina por su posición $x_{l_n} = (\mathrm{x}_n, \mathrm{y}_n, \theta_n)_i^T$, su incertidumbre P_{l_n} y un conjunto M de puntos de interés p_j expresado en coordenadas de imagen. A cada punto se le asocia un descriptor visual d_j, $j = 1, \ldots, M$.

Finalmente, el vector de estado para las variables del problema es el siguiente:

$$\bar{x} = \begin{bmatrix} x_v & x_{l_1} & x_{l_2} & \cdots & x_{l_n} \end{bmatrix}^T \qquad (1)$$

Un ejemplo ilustrativo de construcción del mapa se detalla en la Figura 1.

2.1. MODELO DE OBSERVACIÓN

Según el modelo de representación basado en vistas, es necesario describir un modelo de observación en concordancia con el anterior. La versatilidad de emplear imágenes omnidireccionales hace posible obtener una medida de observación que relaciona la transformación relativa entre dos imágenes con la transformación relativa entre las dos posiciones donde fueron adquiridas dichas imágenes [14], tal y como se puede observar en la Figura 1. Para la obtención de la transformación, sólo son necesarias dos imágenes y el conjunto de puntos correspondientes que se extrae entre ambas. El modelo de observación se describe como:

$$z_t = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{\mathrm{y}_{l_n} - \mathrm{y}_v}{\mathrm{x}_{l_n} - \mathrm{x}_v}\right) - \theta_v \\ \theta_{l_n} - \theta_v \end{pmatrix} \qquad (2)$$

donde ϕ es el ángulo que forma el robot al observar la posición de una vista N y β es la orientación relativa entre la vista observada y la imagen actual sobre la posición del robot. La vista N queda representada por $x_{l_n} = (\mathrm{x}_{l_n}, \mathrm{y}_{l_n}, \theta_{l_n})$, mientras que la posición del robot se representa como $x_v = (\mathrm{x}_v, \mathrm{y}_v, \theta_v)$. Ambas medidas (ϕ, β) son mostradas en la Figura 1.

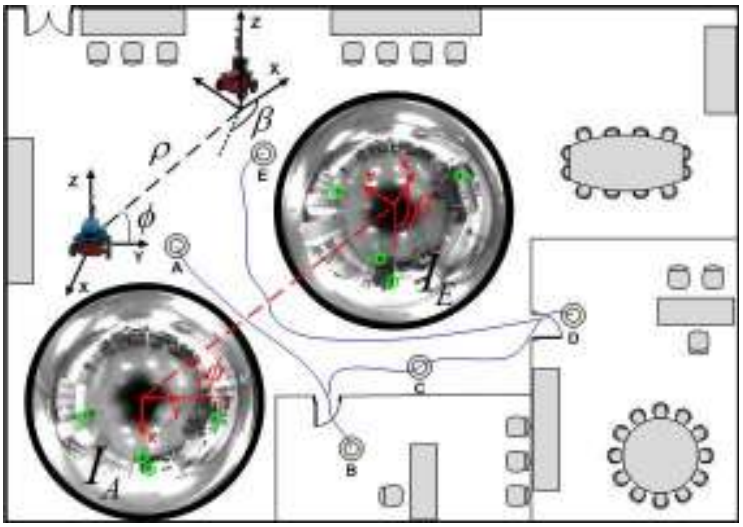


Figura 1: Proceso de construcción del mapa. El origen se establece en A, donde el robot almacena la vista I_A. El robot explora el entorno mientras compara la imagen actual con la vista I_A. Cuando ninguna correspondencia entre imágenes puede ser extraída, se inicializa una nueva vista, en este caso I_B en la posición B. El proceso continúa hasta que el mapa del entorno queda determinado con el resto de vistas, I_C, I_D. Además se muestra la transformación relativa entre las posiciones A y E y sus correspondientes vistas asociadas I_A e I_E.

3. MAPAS DE MÁXIMA VEROSIMILITUD

3.1. ESPECIFICACIONES DEL PROBLEMA

Los modelos de mapa basados en grafos se componen por un conjunto de nodos que definen las posiciones recorridas por el robot y las marcas inicializadas en el mapa. El vector de estado s_t codifica esta representación a través del conjunto de variables definidas como:

$$s_t = \left[(x_0, y_0, \theta_0), (x_1, y_1, \theta_1) \dots \quad (x_n, y_n, \theta_n)\right] \tag{3}$$

siendo (x_n, y_n, θ_n) las coordenadas 2D y la orientación en el sistema de referencia general. Por otro lado, existe un subconjunto de aristas que representan las relaciones entre nodos a partir de la distancia calculada por la odometría, o bien por las medidas de observación que entregan los sistemas sensoriales. Ambas medidas son comúnmente conocidas como restricciones y expresadas como δ_{ji}, donde j indica el nodo observado, visto desde el nodo i. El propósito general de estos modelos [12, 7] es minimizar la probabilidad de error total que presentan las restricciones:

$$P_{ji}(s) \propto \eta \exp\left(-\frac{1}{2}(f_{ji}(s) - \delta_{ji})^T \Omega_{ji} (f_{ji}(s) - \delta_{ji})\right) \tag{4}$$

donde $f_{ji}(s)$ es una función dependiente del estado s_t y de los nodos j y i. La diferencia entre $f_{ji}(s)$ y δ_{ji} es el error de desviación entre ambos nodos. Este término de error se pondera por la matriz de información:

$$\Omega_{ji} = \Sigma_{ji}^{-1} \tag{5}$$

donde Σ_{ji}^{-1} es la inversa de la covarianza asociada, la cual introduce la incertidumbre en las medidas de observación. Si se asume notación logarítmica se obtiene:

$$F_{ji}(s) \propto (f_{ji}(s) - \delta_{ji})^T \Omega_{ji} (f_{ji}(s) - \delta_{ji}) \tag{6}$$

$$= e_{ji}(s)^T \Omega_{ji} e_{ji}(s) = r_{ji}(s)^T \Omega_{ji} r_{ji}(s) \tag{7}$$

siendo $e_{ji}(s)$ el error calculado como $f_{ji}(s)$-$\delta_{ji}(s)$, también referido como $r_{ji}(s)$ con el fin de enfatizar su condición de residuo. En general, el problema global persigue la minimización de la función objetivo que representa el error de todas las observaciones o restricciones como:

$$F(s) = \sum_{\langle j,i \rangle \in G} F_{ji}(s) = \sum_{\langle j,i \rangle \in G} r_{ji}(s)^T \Omega_{ji} r_{ji}(s) \tag{8}$$

donde $G = \{\langle j_1, i_1 \rangle, \langle j_2, i_2 \rangle \dots\}$ determina el subconjunto específico de restricciones que constituyen el mapa, ya sean de odometría o relativas a las medidas de observación.

3.2. SOLUCIÓN AL PROBLEMA: SGD

Una vez presentada la formulación del problema, ahora se detalla el algoritmo encargado de resolverlo: SGD. Su misión es calcular una estimación

válida para el problema de SLAM de manera iterativa. En concreto, minimiza (8) mediante un proceso de optimización basado en mínimos cuadrados. El vector de estado se obtiene como:

$$s_{t+1} = s_t + \Delta s \tag{9}$$

donde Δs expresa una cierta actualización sobre s_t, término que se genera secuencialmente a través de un procedimiento de optimización donde las restricciones actúan como entradas. Hay que destacar que en el caso general de SGD, dicha actualización se calcula independientemente en cada paso con una única restricción, es decir $\Delta s_n = f(\delta_{ji})$. La expresión general que relaciona la transición de estados entre s_t y s_{t+1} tiene la forma:

$$s_{t+1} = s_t + \lambda \cdot H^{-1} J_{ji}^T \Omega_{ji} r_{ji} \tag{10}$$

- J_{ji} s la Jacobiana de f_{ji} con respecto de s_t, $J_{ji} = \frac{\partial f_{ji}}{\partial s}$. Transforma la desviación del error en una variación espacial.
- H es la matriz Hessiana calculada como $J^T \Omega J$, y que da forma al error a través de una matriz precondicionada para escalar las variaciones de J_{ji}. De acuerdo con [6], H puede obtenerse como:

$$H \approx \sum_{\langle i,j \rangle} J_{ji} \Omega_{ji} J_{ji}^T \tag{11}$$

- Ω_{ji} es la matriz de información asociada a una restricción, y queda determinada por la inversa de la matriz de covarianza de la observación Σ_{ji}^{-1}, para cada restricción de observación δ_{ji}.
- λ es un factor de aprendizaje el cual reescala el término $H^{-1} J_{ji}^T \Omega_{ji} r_{ji}$. Normalmente, λ toma valores decrecientes según el criterio $\lambda = 1/n$, donde n es el número de iteración. Esta estrategia pretende acercarse rápidamente a la solución final en las primeras iteraciones. En caso de estar cerca del óptimo, se entiende que el número de iteraciones será mayor, y por tanto el paso deberá ser menor, consiguiendo así que valores más bajos de λ eviten posibles oscilaciones alrededor del óptimo.

Este esquema actualiza la estimación gracias a la rectificación calculada con cada restricción por iteración. A pesar de que el factor de aprendizaje reduce el peso con el cual cada restricción actualiza la estimación, el proceso puede desembocar en un modelo ineficiente para alcanzar una solución estable. Esto se debe a las posibles oscilaciones indeseadas que pueden ocurrir por la naturaleza estocástica en la selección de restricciones. Por este motivo proponemos una optimización de dicho proceso, la cual considera la introducción en el sistema de SLAM de varias restricciones en la misma iteración. Pueden generarse dudas en cuanto al posible aumento de la carga computacional. Sin embargo, se han considerado ciertas adaptaciones y mejoras que evitan la aparición de sobrecargas en el sistema, llegando incluso a mejorar la ratio de velocidad de convergencia de la estimación. En los siguientes apartados se amplía con mayor detalle.

4. SGD MODIFICADO

En primer lugar, se debe tener en cuenta la nueva definición del vector de estado s_t, el cual será tratado como un conjunto de variables incrementales:

$$s_t^{inc} = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \\ dx_1 \\ dy_1 \\ d\theta_1 \\ \dots \\ dx_n \\ dy_n \\ d\theta_n \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \\ x_1 - x_0 \\ y_1 - y_0 \\ \theta_1 - \theta_0 \\ x_2 - x_1 \\ y_2 - y_1 \\ \theta_2 - \theta_1 \\ \dots \\ x_n - x_{n-1} \\ y_n - y_{n-1} \\ \theta_n - \theta_{n-1} \end{bmatrix} \tag{12}$$

donde $(dx_n, dy_n, d\theta_n)$ representa la variación de posiciones consecutivas en coordenadas del sistema de referencia global. Esta codificación presenta la ventaja de actualizar más de un nodo y sus adyacentes por cada restricción. En cambio una codificación global únicamente actualiza un nodo y su adyacente por restricción. Por tanto, para una codificación incremental, el factor Δs (9) realiza el efecto de actualización sobre todas las posiciones ya que el vector de estado se calcula de forma diferencial. Atendiendo a la formulación definida en (1) y (3), x_v, y cada x_{l_n} se corresponden con (x_0, y_0, θ_0), (x_1, y_1, θ_1) … (x_n, y_n, θ_n).

Cabe destacar que en esta propuesta donde se trabaja con observaciones omnidireccionales, las ecuaciones previamente definidas para un caso general de SGD, deben ser adaptadas. De acuerdo con (2), dados dos nodos, la medida de observación permite determinar una transformación para el movimiento entre dichos dos nodos, salvo un factor de escala. Como consecuencia, deben reformularse varios términos involucrados en la estimación de la solución. A continuación se detallan los cambios y modificaciones que han se llevado a cabo:

- La primera adaptación a considerar se realiza sobre $f_{ji}(s)$, diferenciando claramente entre odometría y observación visual:

$$f_{j,i}^{odo}(s) = \begin{pmatrix} dx_j \\ dy_j \\ d\theta_j \end{pmatrix} + \begin{pmatrix} dx_{j-1} \\ dy_{j-1} \\ d\theta_{j-1} \end{pmatrix} + \cdots + \begin{pmatrix} dx_i \\ dy_i \\ d\theta_i \end{pmatrix} \tag{13}$$

donde $(dx_j, dy_j, d\theta_j)$ se definió en (12), y para el caso de las restricciones asociadas a la observación visual:

$$f_{j,i}^{visual}(s) = \begin{pmatrix} \arctan\left(\frac{\mathrm{dy}_j - \mathrm{dy}_i}{\mathrm{dx}_j - \mathrm{dx}_i}\right) - d\theta_i \\ d\theta_j - d\theta_i \end{pmatrix} \tag{14}$$

donde ϕ y β se calculan directamente a partir de las medidas de observación [14] las cuales expresan la relación entre las vistas omnidireccionales y la codificación de la posición (12). Observando la Figura 1 puede definirse (2), y de modo similar (14).

- La segunda modificación tiene por objeto recalcular los términos $J_{ji} = \frac{\partial f_{ji}}{\partial s}$, puesto que $f_{j,i}(s)$ ha sido reformulada. Hay que señalar la importancia de los índices correspondientes a cada nodo ya que en caso de ser $j > i$, o bien $j < i$, las derivadas resultantes varían considerablemente. Además, como se ha podido comprobar, las dimensiones de $f_{j,i}(s)$ son distintas para el caso de la odometría y de la observación visual, hecho que hay que tener en cuenta a la hora de reajustar las dimensiones del resto de términos involucrados en la nueva implementación de SGD.

$$J_{j,i} = \frac{\partial f_{j,i}}{\partial s} = \frac{\partial f_{j,i}(s)}{\partial s} = \left[\frac{\partial f_{j,i}(\phi)}{\partial s}, \frac{\partial f_{j,i}(\beta)}{\partial s}\right] \tag{15}$$

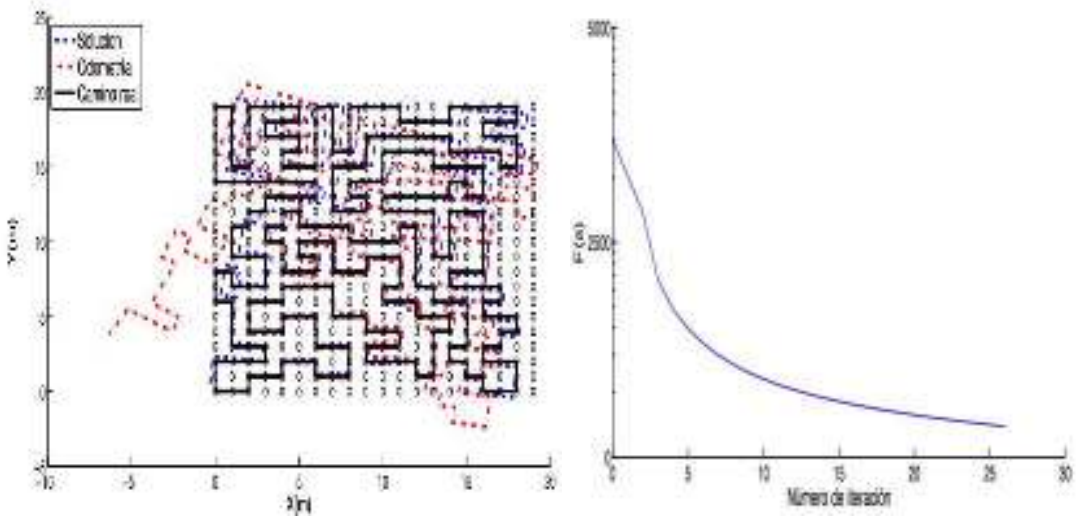
- La tercera modificación propuesta en este trabajo sugiere que la estimación del nuevo estado s_{t+1} sea calculada simultáneamente a partir de varias restricciones por cada iteración. Se pretende dar más peso a las medidas de observación con el objetivo de alcanzar una solución óptima más rápidamente. Obviamente, procesar más de una restricción conlleva una cierta sobrecarga del sistema. Sin embargo, hemos contrarrestado dicho efecto gracias a la reducción del coste computacional para calcular H. En un caso general, H se calcula independientemente para todas y cada una de las restricciones, es decir, se repite dicho cálculo un número de veces igual al número de restricciones existentes. Por el contrario, en nuestro caso calculamos H una única vez por cada subconjunto de restricciones que se introducen simultáneamente en el algoritmo. Consecuentemente, la obtención de H es más eficiente y se compensan las posibles sobrecargas computacionales mencionadas.

5. RESULTADOS

Hemos llevado a cabo tres conjuntos de experimentos bien diferenciados para los que se ha empleado un robot Pioneer P3-AT equipado con una cámara 1280x960 y un espejo hiperbólico. En primer lugar, en la Sección 5.1 se muestran resultados de SLAM obtenidos en un entorno de simulación con el propósito de confirmar la validez del nuevo modelo de SLAM basado en SGD con vistas omnidireccionales. A continuación, la Sección 5.2 presenta una comparación entre los resultados de SLAM obtenidos con nuestra propuesta y con un algoritmo general de SGD, el cual se ha desarrollado en base a [12, 7]. Por último, en la Sección 5.3 mostramos resultados de SLAM en un entorno más amplio y realista.

5.1. EXPERIMENTO 1

Asegurar la convergencia de un algoritmo de SLAM es de gran importancia cuando se introduce una nueva técnica de estimación, como en este caso es SGD. Además hay que considerar ciertas modificaciones para tratar con un modelo de observación visual, lo cual añade errores no lineales. La Figura 2(a) presenta una simulación en un entorno de 20x20m, donde el robot recorre aproximadamente $300m$. El camino real se muestra en línea continua, la odometría en línea a trazos, mientras que la solución estimada se representa con línea punteada. Un conjunto de vistas han sido situadas aleatoriamente a lo largo de la trayectoria. Su emplazamiento se ha controlado mediante una ratio de similitud de apariencia entre imágenes, de modo que se asegure la inicialización de una vista cuando la apariencia de la imagen actual y las vistas del mapa varíe significativamente. El número de iteraciones se ha fijado en 25. En la Figura 2(a) puede observarse la estimación final, la cual sigue a la trayectoria real, y que logra rectificar el valor inicial de la odometría mediante la introducción de las observaciones, con un error RMS aproximado de $0.5m$, frente a los $5m$ de la odometría. La Figura 2(b) muestra la tendencia decreciente en la evolución de la probabilidad de error $P_{ji}(s)$ en (4) asociada a las restricciones, expresada en términos logarítmicos como $F(s)$ en (8), frente al número de iteraciones. Por tanto, se confirma la viabilidad del nuevo modelo propuesto de SGD, ya que la solución generada es válida para el problema de SLAM con imágenes omnidireccionales.



(a) (b)

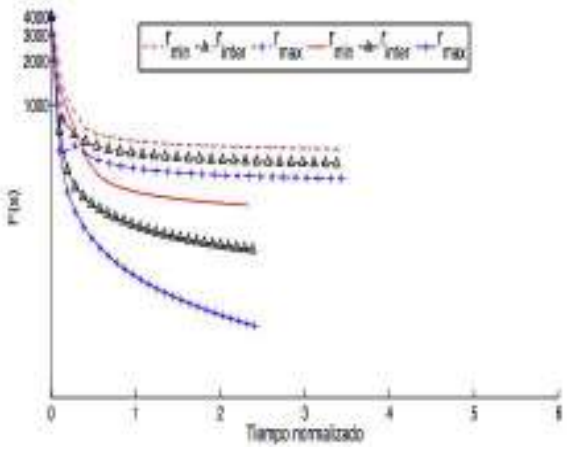
Figura 2: La Figura 2(a) muestra el mapa obtenido con el modelo propuesto en un entorno de 20x20m, con el camino real (línea continua), odometría (línea a trazos) y estimación (línea punteada). La Figura 2(b) muestra el error acumulado por las restriciones $F(s)$ frente al número de iteraciones.

5.2. COMPARACIÓN DE RESULTADOS

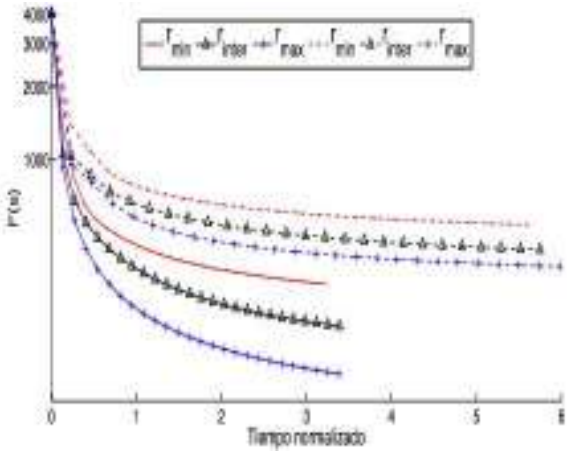
El siguiente experimento ha sido planteado con la intención de comparar nuestra propuesta con un algoritmo general de SGD en términos de eficiencia. Proponemos una estrategia que introduce varias restricciones de observación simultáneamente en el sistema. El objeto principal es mejorar la velocidad de convergencia a la hora de lograr la estimación final. En este sentido, hemos llevado a cabo un experimento de SLAM donde el robot recorre 50m en un entorno dado. De nuevo, el número de vistas en el mapa ha sido simulado siguiendo la misma política mencionada anteriormente. Este experimento ha sido repetido 200 veces a partir de la misma odometría inicial, lo cual da solidez y consistencia a los valores medios obtenidos. Ambos métodos han sido comparados, el SGD tradicional y el aquí propuesto. Se ha variado el número de vistas N que el robot es capaz de observar desde cada posición. El radio de observación del robot r también ha sido variado. La Figura 3 presenta resultados para el error acumulado por las restricciones $F(s)$ de todas las restricciones, siendo ésta la función objetivo que el algoritmo de SGD pretende minimizar. Comparamos la solución obtenida con nuestra propuesta, representada con línea continua, frente a la obtenida con un algoritmo general de SGD, representada con línea punteada. Las Figuras 3(a), 3(b) y 3(c) representan $F(s)$ cuando el robot observa N=2, N=4 y N=8 vistas, respectivamente. Puesto que buscamos obtener una comparación realista, el eje x, que originalmente representaba el número de iteración, se ha transformado en una variable de tiempo normalizada que permite una comparación fiable entre ambos métodos. Esta conversión tiene un mero efecto de simplicidad, ya que el tiempo empleado en cada iteración depende del número de vistas observadas N. En términos de eficiencia, puede comprobarse que la solución generada por el modelo propuesto mejora la solución entregada por el método básico de SGD. Esta conclusión se deduce al observar la pendiente de caída de $F(s)$, factor que demuestra mayor velocidad de convergencia. Este efecto surge como principal resultado de combinar varias restricciones de observación en una misma iteración, en lugar de únicamente una. También es destacable la relevancia del radio de observación del vehículo r. Como puede verse en las Figuras 3(a), 3(b) y 3(c), valores mayores de r generan mejores velocidades de convergencia en detrimento de valores más pequeños. Esto se debe a que la medida de observación omnidireccional es angular y sin escala, lo cual hace que las vistas más alejadas que observa el robot sean las que proporcionen una medida más consistente para localizar al mismo.

5.3. EXPERIMENTO 2

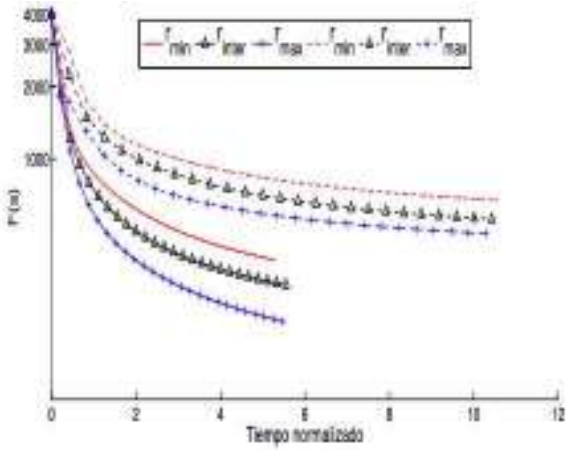
Este experimento tiene por objeto testear el funcionamiento de esta propuesta ante una situación más realista, la cual se define con un entorno de tipo oficina de dimensiones 20x40m por donde navega el robot. Se añaden elementos típicos obstructores como son puertas y paredes. La Figura 4 describe el camino real con línea continua, la odometría con línea a trazos y la solución estimada



(a)



(b)



(c)

Figura 3: Error acumulado por las restricciones $F(s)$ frente al tiempo en un experimento de SLAM. Se muestran los valores obtenidos con el modelo propuesto (línea continua) y con un modelo general de SGD (línea punteada). Las Figuras 3(a), 3(b) y 3(c) muestran $F(s)$ cuando el número de vistas observadas es N=2, N=4 y N=8 respectivamente. Se presentan varios valores del radio de observación: r_{min}, r_{inter} y r_{max}.

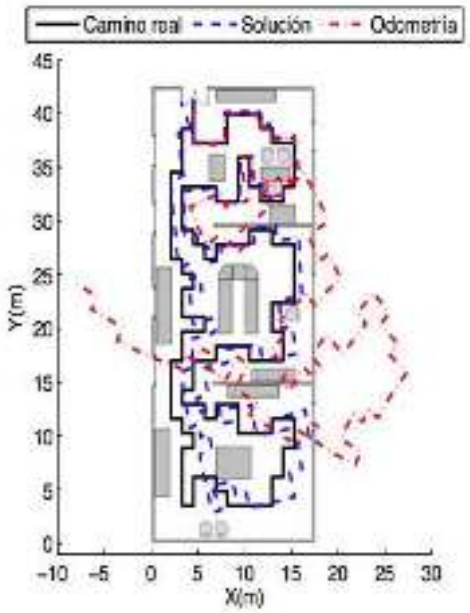


Figura 4: Resultados de SLAM en un entorno de oficinas de 20x40m. Se muestra el camino real (línea continua), la odometría (línea a trazos) y la solución estimada (línea punteada).

con línea punteada. El algoritmo consigue generar una solución aceptable con sólo 15 iteraciones, y cuya topología se asemeja al camino real. Por contra el error de la odometría crece sin límites. El error RMS se aproxima a $1m$, frente a los $10m$ de la odometría. La Figura 5 muestra una comparativa de la evolución del error acumulado por las restricciones $F(s)$ frente al tiempo para ambos modelos. Nuevamente, se demuestra la mayor capacidad de esta propuesta para alcanzar más rápidamente la solución, lo que implica mejor eficiencia. En este caso concreto, queda de manifiesto que el modelo propuesto presenta una velocidad de convergencia aproximadamente 6 veces mejor que la de un modelo tradicional de SGD.

6. CONCLUSIONES

En este trabajo se ha presentado un modelo de SGD para el problema de SLAM basado en imágenes omnidireccionales. La intención de esta propuesta ha sido la de contrarrestar las inestabilidades y efectos dañinos que provocan los errores no lineales y que comprometen gravemente la convergencia de los métodos tradicionales de SLAM, como por ejemplo el EKF. El mayor responsable de estos errores suele ser el tipo de observación visual, y en especial la omnidireccional. Hemos confiado en un modelo de SLAM visual que representa el entorno con un número muy reducido de vistas. La obtención de la posición del robot se

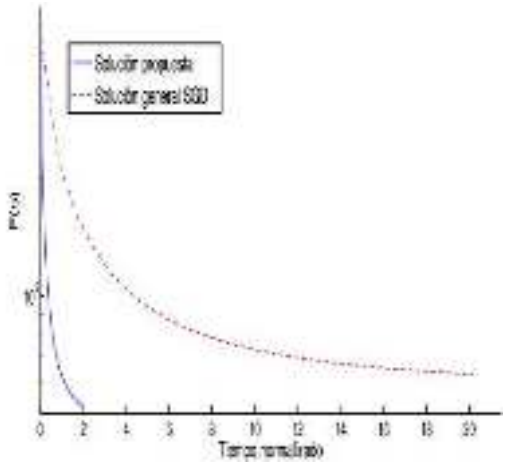


Figura 5: Comparativa del error acumulado por las restricciones $F(s)$ del modelo propuesto (línea continua) y del modelo general de SGD (línea punteada).

calcula con un esquema eficiente de comparación entre vistas, el cual proporciona directamente la transformación de movimiento entre dos posiciones. Se ha modificado el algoritmo general de SGD para trabajar con un modelo de observación omnidireccional. Además, se ha definido una nueva estrategia para procesar simultáneamente varias restricciones de observación en la misma iteración del SGD. Hemos mostrado resultados de SLAM que demuestran la validez de esta nueva propuesta para trabajar con imágenes omnidireccionales, así como también se ha comparado frente al modelo general de SGD. En consecuencia, se demuestra la confiabilidad de dicha propuesta, así como también las mejoras y beneficios obtenidos en términos de eficiencia.

Agradecimientos

Este trabajo ha sido posible en parte gracias al proyecto DPI2010-15308 del Ministerio de Ciencia e Innovación, titulado "Exploración integrada de entornos mediante robots cooperativos para la creación de mapas 3D visuales y topológicos que puedan ser usados en navegación con 6 grados de libertad".

Referencias

[1] Y. Chou and L. Jing-Sin. A robotic indoor 3D mapping system using a 2D laser range finder mounted on a rotating four-bar linkage of a mobile platform. *International Journal of Advanced Robotic Systems*, 10, 2013.

[2] J. Civera, A. J. Davison, and J. M. Martínez Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24:932–945, 2008.

[3] A. J. Davison, Y. Gonzalez Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, pages 117–124, Lisbon, Portugal, 2004.

[4] A. J. Davison and D. M. Murray. Simultaneous localisation and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24:865–880, 2002.

[5] A. Gil, O. Reinoso, M. Ballesta, M. Juliá, and L. Payá. Estimation of visual maps with a robot network equipped with vision sensors. *Sensors*, 10:5209–5232, 2010.

[6] G. Grisetti, C. Stachniss, and W. Burgard. Nonlinear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 10:428–439, 2009.

[7] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of the Robotics: Science and Systems (RSS)*, pages 1–8, Atlanta, USA, 2007.

[8] K. Jae-Hean and C. Myung Jin. Slam with omnidirectional stereo vision sensor. In *Proc. of the IROS*, Las Vegas, USA, 2003.

[9] C. Joly and P. Rives. Bearing-only SAM using a minimal inverse depth parametrization. In *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 2, pages 281–288, Funchal, Madeira, Portugal, 2010.

[10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In *Proceedings of the 18th national conference on Artificial Intelligence*, pages 593–598, Edmonton, Canada, 2002.

[11] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17:890–897, 2001.

[12] D. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 2262–2269, Orlando, USA, 2006.

[13] C. Stachniss, G. Grisetti, D. Haehnel, and W. Burgard. Improved Rao-Blackwellized mapping by adaptive sampling and active loop-closure. In *Proceedings of the Workshop on Self-Organization of Adaptive behavior (SOAVE)*, pages 1–15, Ilmenau, Germany, 2004.

[14] D. Valiente, A. Gil, L. Fernández, and O. Reinoso. View-based maps using omnidirectional images. In *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 2, pages 48–57, Rome, Italy, 2012.

CONTROL VISUAL EMBEBIDO PARA EL POSICIONAMIENTO AUTÓNOMO DE UN HEXACÓPTERO

Javier Pérez, Jorge Pomares, Gabriel J. García, F. Torres
Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal
Carretera San Vicente del Raspeig, s/n, San Vicente del Raspeig, CP: 03690
Universidad de Alicante
jpa29@alu.ua.es, {jpomares; gjgg; Fernando.Torres}@ua.es

Resumen

En este artículo se describen los principales componentes que conforman un sistema embebido capaz de realizar el guiado de un hexacóptero mediante visión artificial. En concreto, el sistema embebido implementa una arquitectura de control basado en imagen para poder guiar al vehículo aéreo empleando una cámara y de forma totalmente autónoma. Se detallan los distintos componentes de la arquitectura hardware que ejecuta a bordo tanto el procesamiento de las imágenes capturadas como los propios algoritmos de control. Finalmente se describen distintos experimentos de simulación para comprobar el correcto ajuste del sistema.

Palabras Clave: Control visual, vehículo aéreo no tripulado, sistema embebido.

1 INTRODUCCIÓN

Los sistemas de control visual actualmente se están aplicando a un amplio rango de ámbitos que van más allá de los industriales y poco a poco van integrándose también en la denominada robótica social. Dentro de este nuevo campo de aplicaciones cabe destacar la descrita en este artículo, es decir, la utilización de visión artificial para realizar el guiado de vehículos aéreos. Es bien conocido que los sistemas de control visual han venido clasificándose principalmente en sistemas de control visual basados en imagen y sistemas de control visual basados en posición [4]. El uso de sistemas de control basados en imagen permiten obtener un comportamiento en general satisfactorio de las características visuales en la imagen. Sin embargo, la trayectoria del robot en el espacio 3D es bastante impredecible y, en muchos casos, inadecuada para grandes desplazamientos rotacionales. Estas diferencias entre la evolución en la imagen y en el espacio 3D vienen determinadas por la no linealidad en la matriz de interacción. Para evitar este problema se implementan los denominados sistemas de control visual basados en posición. Estos últimos permiten obtener un guiado lineal más adecuado en el espacio 3D. Sin embargo, la necesidad de reconstruir posiciones 3D para conseguir este guiado hace que en muchas ocasiones la aproximación más utilizada sea la basada en imagen.

En este artículo se propone el uso de sistemas de control visual basados en imagen para conseguir realizar el guiado de un hexacóptero empleando directamente información visual obtenida de una cámara ubicada en el propio dispositivo. Se describirá una arquitectura software así como los resultados en simulación obtenidos para el ajuste del sistema de control implicado teniendo en cuenta la realimentación visual. En cuanto a la arquitectura de control, se propone separar el problema en un bucle interno de control para realizar el control y estabilización del vehículo mientras que un bucle externo será utilizado para obtener las referencias de posicionamiento directamente en el espacio imagen. El bucle interno de control es ejecutado a alta velocidad haciendo uso de información procedente de los sensores inerciales, acelerómetros, giróscopos y demás sensores del hexacóptero, mientras que el bucle externo implementará la tarea de control visual. La arquitectura hardware será capaz de realizar el procesamiento visual y el control directamente en el hexacóptero, por lo que el vehículo será totalmente autónomo en su tarea.

Los sistemas de control visual están siendo aplicados recientemente para realizar el guiado de vehículos aéreos como pueden ser helicópteros [10] o aviones [2][8]. Debido a su maniobrabilidad y fácil control, en los últimos años han ido creciendo las aplicaciones en las que se emplean los denominados drones. Así, la utilización de visión para el guiado de estos vehículos permite realizar su posicionamiento de forma precisa y autónoma. Es posible encontrar distintos grupos de investigación que trabajan en el uso de control visual para el guiado de este tipo de

vehículos aéreos (ver, por ejemplo [1][9][3]). Aunque la mayoría de los trabajos existentes hasta la actualidad emplean control visual basado en posición [1][6][9] ya existen distintas aproximaciones que tratan de realizar el guiado del vehículo directamente en el espacio imagen empleando los denominados sistemas de control visual basados en imagen (ver, por ejemplo [7][3]).

El artículo se ha estructurado de la siguiente manera. En el Apartado 2 se describe la arquitectura hardware propuesta para la implementación, tanto del procesamiento de las imágenes capturadas por la cámara, como del propio bucle de control. En el Apartado 3 se describe la estrategia de control visual basada en imagen implementadas. Finalmente, en los Apartados 4 y 5 se describen los resultados y conclusiones obtenidas.

2 ARQUITECTURA DEL SISTEMA

En este Apartado se describen los distintos componentes que conforman el hexacóptero así como los sistemas empleados para el procesamiento de la información visual y posterior ejecución "a bordo" de los bucles de control. En el Apartado 2.1 se describe la arquitectura hardware del hexacóptero y en el Apartado 2.2 se muestran aquéllos aspectos más relacionados con el sistema de visión utilizado.

2.1 ARQUITECTURA HARDWARE

En la Figura 1 se indican los componentes principales del hexacóptero. Está formado por seis brazos de aluminio de 240mm de largo que sostienen las seis hélices de 11 pulgadas con un paso de 4,7. Se ha empleado una estructura de fibra de carbono para soportar la estructura. Estos componentes forman parte del kit de 3D Robotics perteneciente al proyecto Arducopter. En cuanto a los motores, se han empleado seis motores brushless de 880Kv modelo AC2836-358 así como sus respectivos variadores de 20 amperios modelo m2mPower AC20-1.0.

Como se indicará en el Apartado 2.2 el hexacóptero dispone de una cámara "a bordo" que transfiere las imágenes capturadas a una plataforma Odroid-U2 de Hardkernel que consta de un procesador Exynos4412 Primer Quad-core y 2GByte DRAM. El Odroid-U2 dispone como sistema operativo Ubuntu Linaro para ARM corriendo en una tarjeta micro SD Sandisk Extreme Pro de clase 10 de 16GB. El sistema Odroid-U2 será el encargado de realizar el procesamiento de las imágenes capturadas así como la implementación del bucle de control necesario para conseguir el guiado autónomo del hexacóptero mediante control visual. La comunicación del Odroid-U2 con el controlador del hexacóptero se realiza por puerto serie.

La autonomía se consigue gracias a 2 baterías. El sistema Odroid-U2 se alimenta con una batería LiPo Rhino 2S de 750mAh 20C a través de un regulador de tensión a 5V. El hexacóptero y su electrónica se alimenta con una batería LiPo 3S Turginy nano-tech de 6000mAh 50C. La carga total del aparato con todo montado es de 2500g aproximadamente, y el aparato es capaz de levantar cargas de hasta 3500g.

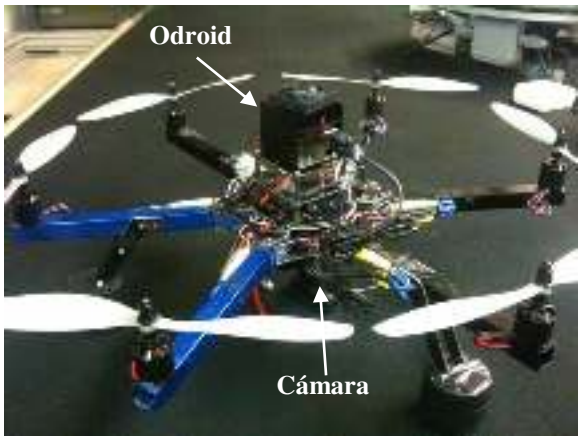


Figura 1: Arquitectura del sistema

2.2 SISTEMA DE VISIÓN

Como se ha indicado en la Figura 1, el hexacóptero dispone de un sistema de visión propio. La adquisición de imagen se ha realizado empleando una webcam USB Logitech HD C270 que trabaja a una resolución de 1280p x 720p. El procesamiento de imagen se ha realizado empleando el sistema Odroid-U2. Para conseguir el guiado del robot se ha empleado el patrón que se muestra en la Figura 2 del cual se han extraído cuatro características visuales. Para ello, tras realizar un primer procesamiento para la eliminación de ruido, se realiza una detección de objetos presentes en la imagen y un posterior cálculo de los centros de gravedad de dichos objetos detectados. Estos cuatro centros serán las características visuales empleados por el sistema de control visual basado en imagen que se describirá en el Apartado 3.



Figura 2: Procesamiento en la imagen

3 CONTROL VISUAL

En este apartado se describe la estrategia de control visual basada en imagen empleada para realizar el guiado del hexacóptero considerando $\mathbf{s} = \left[f_{1x}, f_{1y}, f_{2x}, f_{2y}, \ldots, f_{kx}, f_{ky}\right]^T \in \Re^{2k}$ el conjunto de k puntos característicos en la imagen extraídos. Por otro lado, se define $\mathbf{s}_d \in \Re^{2k}$ como las características visuales deseadas, es decir, aquéllas que deberían obtenerse una vez que la tarea haya finalizado. A partir de este conjunto de características extraídas y deseadas, la acción de control básica para un sistema de control basado en imagen es la siguiente:

$$\boldsymbol{v} = -\lambda \cdot \mathbf{J}_f^+ \cdot \left(\mathbf{s} - \mathbf{s}_d\right) \quad (1)$$

donde $\mathbf{J}_f^+$ se trata de la Moore-Penrose pseudoinversa del denominado Jacobiano de la imagen [4] o matriz de interacción y λ es una ganancia proporcional positiva y $\mathbf{v}$ es la velocidad de traslación que llevaría a la cámara ubicada en el hexacóptero hacia la posición deseada. Esa velocidad requiere procesarse posteriormente por dos motivos. Por un lado, el sistema de coordenadas del hexacóptero y de la cámara no coinciden (ver Figura 3), lo que requiere realizar un cambio en de sistema de coordenadas. Por otro lado, únicamente se emplearán las 3 componentes de traslación y la componente de rotación en z. Estas serán las referencias que se enviarán a los controladores internos del hexacóptero para permitir su guiado.

Para realizar el correcto ajuste de los distintos controladores implicados considerando la realimentación visual se ha implementado en Matlab un sistema de simulación cuyos principales componentes se han representado en la Figura 4. En esta figura, se ha indicado como CVI el controlador visual basado en imagen indicado en la Ecuación (1). Además, el controlador roll, pitch, yaw permite obtener los giros en estas direcciones a partir de las referencias obtenidas del controlador visual (velocidad en las tres componentes de traslación y rotación en z). Más detalles de este controlador pueden verse en [5]. Las referencias obtenidas de este controlador constituyen las entradas para la dinámica del hexacóptero.

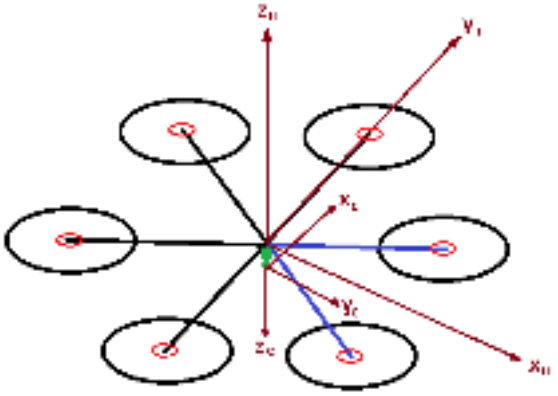


Figura 3: Sistemas de coordenadas del hexacóptero y de la cámara

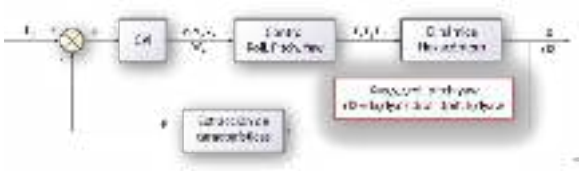


Figura 4: Diagrama de bloques del sistema de simulación

4 RESULTADOS

Con el objetivo de validar el controlador propuesto en la Sección 3, se ha simulado el hexacóptero en el entorno de Matlab. Los distintos experimentos realizados sobre este modelo permiten comprobar el correcto funcionamiento del controlador para, en una fase posterior, poder implementar el controlador en el hexacóptero con seguridad. Se muestran a continuación tres de los diversos experimentos llevados a cabo, donde se ha situado el hexacóptero en distintas posiciones respecto del patrón descrito en la Figura 2. La ley de control proporciona las velocidades lineales y la velocidad angular en el eje Z.

4.1 EXPERIMENTO 1

En el primer experimento, el hexacóptero se sitúa aproximadamente a 1 metro del patrón, momento en el que se activa el control autónomo mediante el controlador visual descrito en la Sección 3. En este experimento la cámara del hexacóptero observa el patrón como se muestra en la Figura 5 (marcas iniciales en forma de cruz verde). Las características deseadas se muestran también en esta imagen como círculos rojos. La figura permite observar el correcto funcionamiento del controlador visual en imagen, en una tarea de posicionamiento en la que el hexacóptero realiza un ascenso alejándose del patrón y un giro sobre el eje Z, pero donde no hay apenas desplazamiento ni en X, ni en Y. Esto último puede

verse en la Figura 6, donde se representa la evolución de la posición del punto central del hexacóptero durante la tarea de posicionamiento.

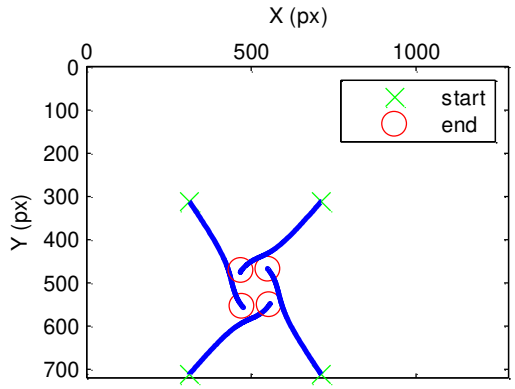


Figura 5: Experimento 1: evolución de las características en la imagen

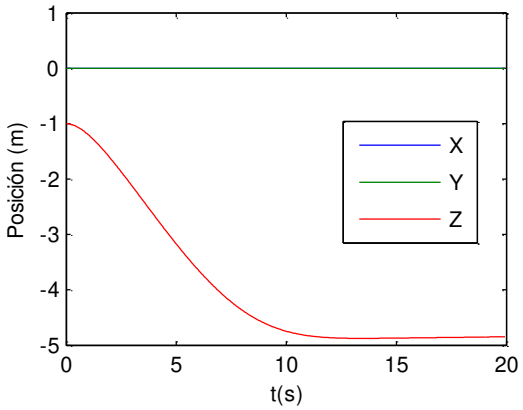


Figura 6: Experimento 1: evolución de la posición del hexacóptero

En la Figura 7 se puede observar cómo en este experimento el hexacóptero sólo debe realizar una orientación según el eje Z, además del posicionamiento que se observaba en la Figura 4 también sobre el eje Z.

La Figura 8 muestra la velocidad de traslación proporcionada por el controlador durante la tarea de posicionamiento. Las componentes en X e Y son prácticamente nulas.

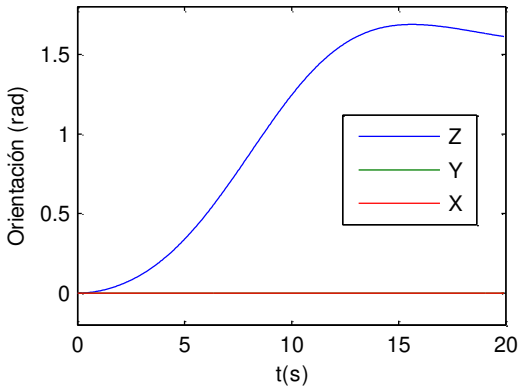


Figura 7: Experimento 1: evolución de la orientación del hexacóptero

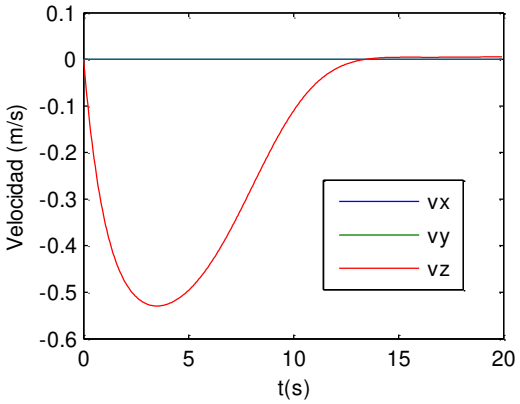


Figura 8: Experimento 1: evolución de la velocidad de traslación enviada al hexacóptero

Finalmente, la Figura 9 muestra el error cometido en imagen durante la tarea de control visual. Se ha representado el módulo del error de cada característica en la imagen. Se observa cómo el error va disminuyendo hasta valer prácticamente 0 a los 20 segundos.

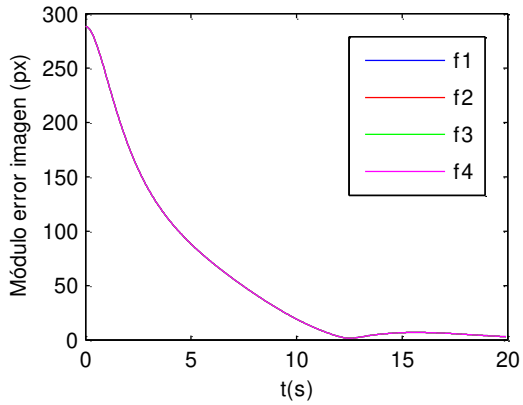


Figura 9: Experimento 1: evolución del error en imagen

4.2 EXPERIMENTO 2

En este segundo experimento el robot se sitúa respecto al patrón en una posición distinta a la del experimento anterior, en la que además de un desplazamiento en el eje Z, se produce también un desplazamiento en los ejes X e Y respecto a la posición final deseada. La Figura 10 representa la evolución de la posición del centro del hexacóptero en este experimento.

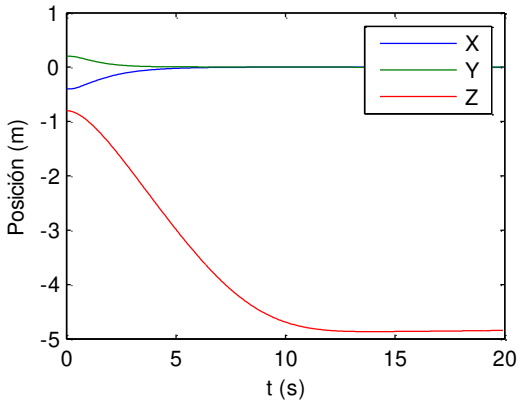


Figura 10: Experimento 2: evolución de la posición del hexacóptero

Se observa cómo el hexacóptero corrige la posición en los tres ejes. En la Figura 11 se observa cómo esta tarea añade movimientos de orientación también en los ejes X e Y para poder corregir esas posiciones de traslación.

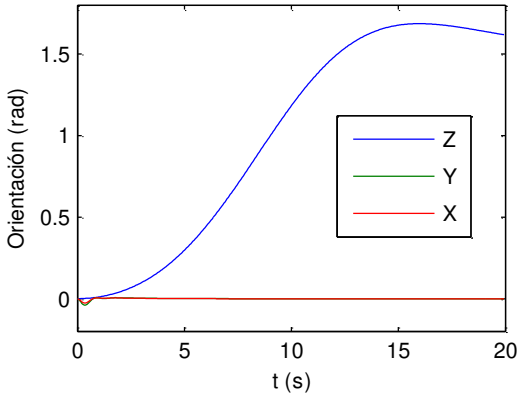


Figura 11: Experimento 2: evolución de la orientación del hexacóptero

Las velocidades de traslación y orientación enviadas por el controlador se muestran en la Figura 12 y 13, respectivamente. En ellas se observa con mayor precisión el ajuste sobre todo en orientación que realiza el controlador para permitir el posicionamiento en el plano paralelo al suelo (ejes X e Y).

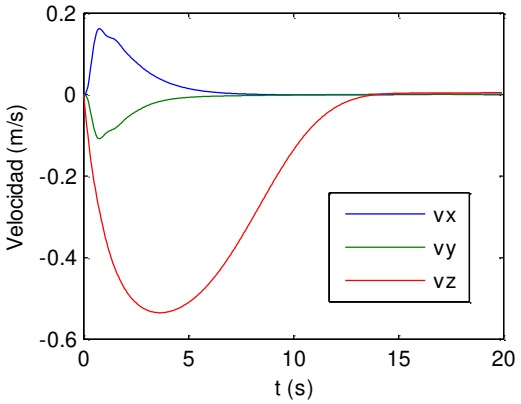


Figura 12: Experimento 2: evolución de la velocidad de traslación enviada al hexacóptero

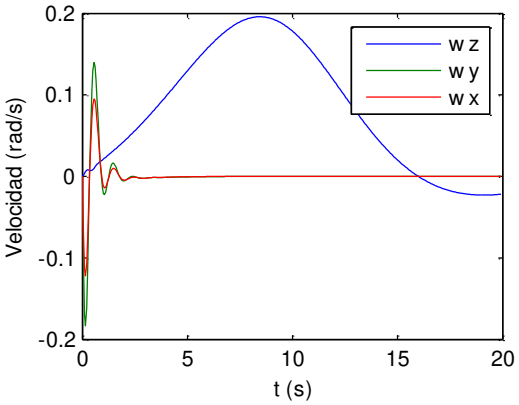


Figura 13: Experimento 2: evolución de la velocidad de rotación enviada al hexacóptero

La evolución de las características visuales en la imagen se observan en la Figura 14. En ella se puede observar la complejidad del posicionamiento requerido en este experimento. Las características visuales convergen desde la posición inicial (indicada con cruces verdes) hasta la posición final deseada (marcada con círculos rojos).

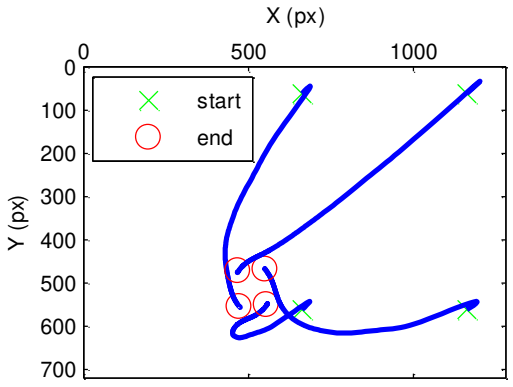


Figura 14: Experimento 2: evolución de las características en la imagen

Al observar la evolución del módulo del error en la imagen de cada característica (ver Figura 15), se puede comprobar el decrecimiento exponencial típico de los sitemas de control visual basado en imagen.

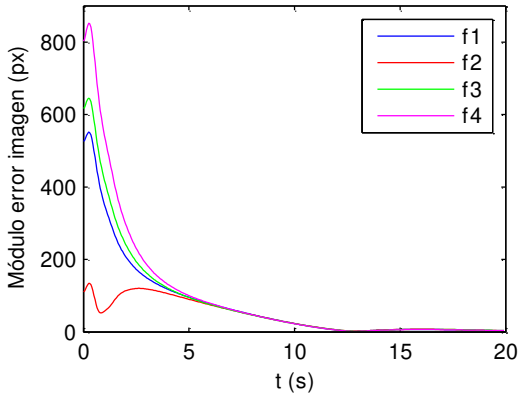


Figura 15: Experimento 2: evolución del error en imagen

4.3 EXPERIMENTO 3

Por último, se muestra a continuación las mismas gráficas mostradas en los experimentos anteriores, sobre otro experimento en el que, al igual que en el experimento 2, el hexacóptero inicia la tarea de posicionamiento mediante control visual observando el patrón de características desde una posición distinta a la de los experimentos anteriores. En la gráfica de la Figura 16 se puede apreciar que la posición inicial sitúa el hexacóptero aproximadamente a 2 metros del suelo, teniendo que posicionarse finalmente en unos 5 metros. También se puede observar un desplazamiento en los ejes X e Y de poco más de medio metro. El hexacóptero estabiliza su posición en aproximadamente 10 segundos.

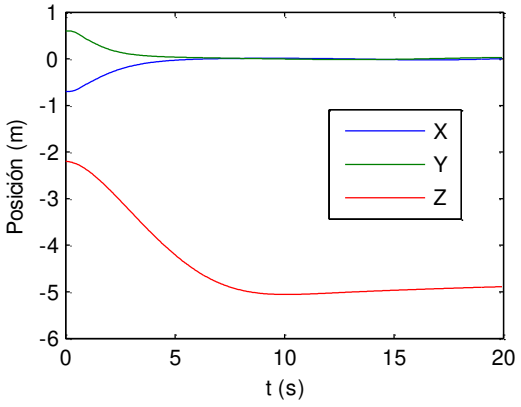


Figura 16: Experimento 3: evolución de la posición del hexacóptero

No ocurre lo mismo con la orientación. En la Figura 17 se muestra la evolución de la orientación durante el posicionamiento. La posición inicial y final están bastante alejadas en la orientación del eje Z.

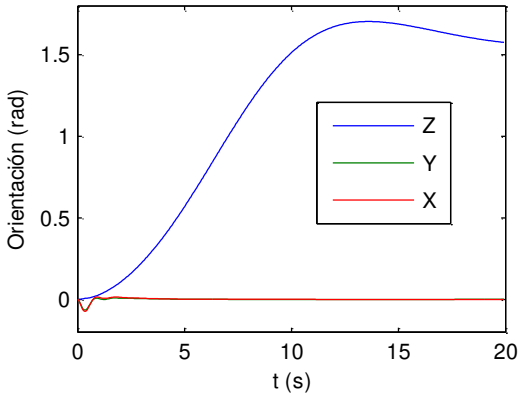


Figura 17: Experimento 3: evolución de la orientación del hexacóptero

La Figura 18 permite observar las velocidades de traslación enviadas al hexacóptero para obtener la posición descrita en la Figura 16. El sistema prácticamente está estabilizado a los 20 segundos. En la Figura 19 se observa las velocidades de equilibrado en la orientación de los ejes X e Y iniciales para poder posicionar en estos ejes al hexacóptero. En seguida se estabilizan y es la velocidad rotacional en el eje Z la que requiere más tiempo dado que es la orientación en este eje la que más alejada está entre posición inicial y deseada.

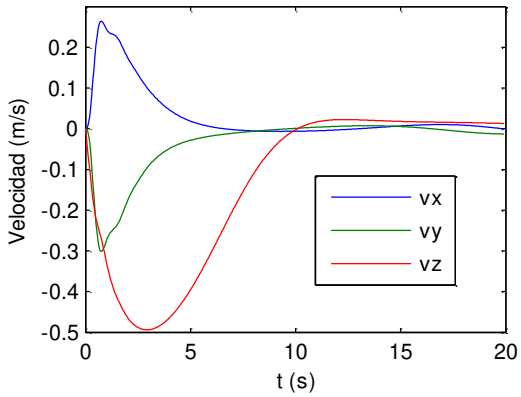


Figura 18: Experimento 3: evolución de la velocidad de traslación enviada al hexacóptero

Finalmente, la Figura 20 muestra la evolución de las características en la imagen y la Figura 21 la evolución del error en imagen para este último experimento. De nuevo, como en los experimentos anteriores, se observa que el error decrece de manera exponencial, observando que prácticamente se obtiene la convergencia a los 10 segundos. En el plano imagen las características se mantienen en todo momento dentro del campo de visión de la cámara embebida.

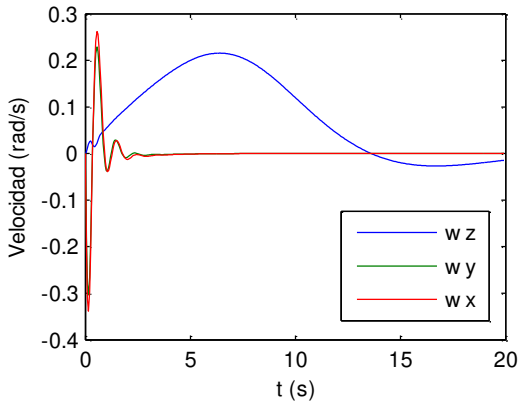


Figura 19: Experimento 3: evolución de la velocidad de rotación enviada al hexacóptero

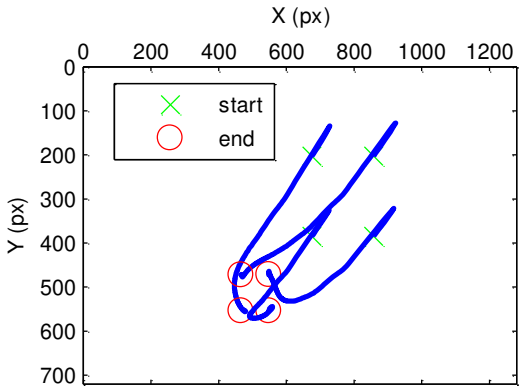


Figura 20: Experimento 3: evolución de las características en la imagen

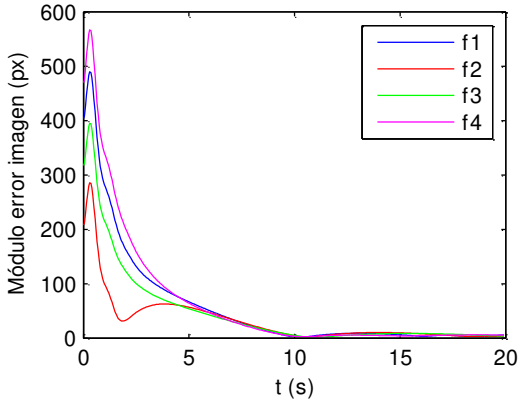


Figura 21: Experimento 3: evolución del error en imagen

4.4 RESULTADOS EXPERIMENTALES

En este apartado se muestran unos resultados experimentales preliminares que permiten evaluar la arquitectura así como comprobar el correcto comportamiento del hexacóptero y el sistema control visual empleado para su guiado.

Se han realizado tres experimentos de posicionamiento representados en la Figura 22. Haciendo uso de la arquitectura se ha obtenido una realimentación visual cada 80 mseg. En azul se ha indicado la evolución de las características en la imagen cuando el objeto del cual se extraen las características se encuentra fijo. En verde y rojo se muestran las trayectorias considerando que el objeto se desplaza durante la tarea de control visual. Se observa como en todos los casos la trayectoria en el espacio imagen converge hacia la configuración deseada. Además, el sistema de control visual compensa los movimientos del patrón seguido. Actualmente se está trabajando en introducir un módulo de telemetría para obtener más información acerca de la trayectoria seguida por el hexacóptero.

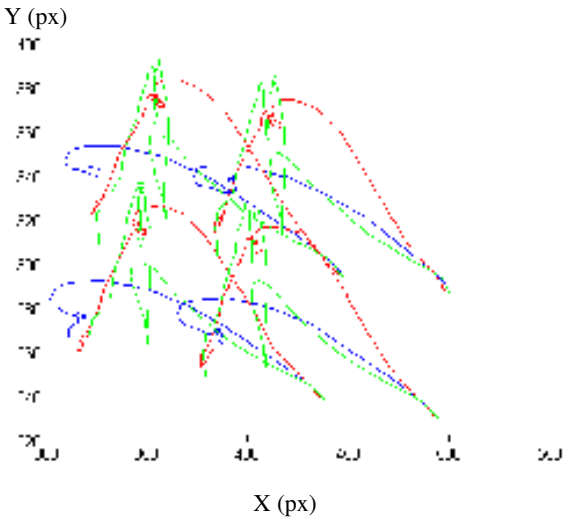


Figura 22: Resultados experimentales. Trayectoria en el espacio imagen.

5 CONCLUSIONES

En el presente artículo se muestra el trabajo realizado para controlar de manera autónoma la posición y orientación respecto al eje Z de un hexacóptero. Para ello se propone un sistema de control visual basado en imagen que utiliza una cámara para obtener referencias de posicionamiento relativo entre el hexacóptero y un patrón situado en el suelo. Se han descrito los distintos componentes hardware del sistema, así como los algoritmos empleados para el sistema de visión embebido. A partir del sistema propuesto se ha introducido en Matlab un modelo del hexacóptero de forma que pueda validarse en simulación el controlador visual. Como continuación de los trabajos expuestos en este artículo, se está trabajando en la implementación del algoritmo de

manera embebida sobre la plataforma Odroid-U2 que permita el guiado autónomo del hexacóptero.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía, los fondos FEDER y el Gobierno de la comunidad Valenciana mediante los proyectos DPI2012-32390 and PROMETEO/2013/085.

Referencias

[1] E. Altug, J. P. Ostrowski, and C. J. Taylor, "Control of a quadrotor helicopter using dual camera visual feedback," Int. J. Robot. Res., vol. 24,no. 5, pp. 329–341, 2005.

[2] O. Bourquardez and F. Chaumette, "Visual servoing of an airplane for auto-landing," in IEEE/RSJ Int. Conf. Intell. Robots Syst., IROS 2007, San Diego, CA, Oct., pp. 1314–1319.

[3] Bourquardez, O.; Mahony, R.; Guenard, N.; Chaumette, F.; Hamel, T.; Eck, L., "Image-Based Visual Servo Control of the Translation Kinematics of a Quadrotor Aerial Vehicle," Robotics, IEEE Transactions on , vol.25, no.3, pp.743,749, June 2009

[4] F. Chaumette, and S. Hutchinson. "Visual Servo Control, Part I: Basic Approaches". IEEE Robotics and Automation Magazine, vol. 13, no. 4, pp. 82-90, December 2006.

[5] R. Mahony, V. Kumar, P. Corke, "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," IEE Robotics & Automation Magazine, vol.19, no.3, pp.20-32, 2012.

[6] I. F. Mondragon, P. Campoy, C. Martinez, and M. Olivares-Mendez, "3D pose estimation based on planar object tracking for UAVs control". IEEE International Conference on Robotics and Automation, pp.35-41, May 2010

[7] R. Ozawa, F. Chaumette, "Dynamic visual servoing with image moments for an unmanned aerial vehicle using a virtual spring approach", Advanced Robotics, Special Issue on Aerial Robots, vol. 27, no. 9, pp. 683-696, 2013

[8] P. Rives and J. R. Azinheira, "Visual auto-landing of an autonomous aircraft," INRIA, Sophia Antipolis, France, Tech. Rep. 4606, Nov. 2002.

[9] H. Romero, S. Salazar, S. and R. Lozano, "Visual servoing applied to realtime stabilization of a multirotor UAV", Robotica, vo. 30, pp. 1203-1212, 2012.

[10] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually-guided landing of an unmanned aerial vehicle," IEEE Trans. Robot. Autom., vol. 19, no. 3, pp. 371–381, Jun. 2003.

LOCALIZACIÓN ASISTIDA POR GPS PARA ROBOTS MÓVILES EN CONFIGURACIÓN ACKERMANN DE RECURSOS LIMITADOS

Leonardo Marín, Ángel Soriano, Vicent Mayans, Marina Vallés, Ángel Valera, Pedro Albertos
email: {leomarpa,ansovi,vimaro,mvalles,giuprog,pedro}@ai2.upv.es
Departamento de Ingeniería de Sistemas y Automática
Instituto Universitario de Automática e Informática Industrial
Universidad Politécnica de Valencia, Valencia, España

Resumen

En el presente trabajo se expone la implementación de un algoritmo de fusión de datos provenientes de un sistema de posicionamiento global (GPS) con la estimación de la posición mediante sensores inerciales con el fin de mejorar la localización de robots móviles de recursos computacionales limitados. Para ello, se hace una revisión de las técnicas existentes para la fusión de datos para localización basada en GPS, con especial interés en las basadas en el filtro de Kalman. El algoritmo propuesto utiliza un filtro de Kalman y un modelo cinemático de un robot móvil de configuración Ackermann junto con una actualización de la posición basada en eventos. El evento se define con base en la covarianza del error de estimación, realizando la actualización del filtro cuando el evento supera un nivel preestablecido. Se incluye además la información de la cantidad de satélites disponibles en el GPS para mejorar la precisión de la estimación. La estrategia basada en eventos permite la implementación del método en un robot de recursos limitados ya que utiliza menor ancho de banda y menos tiempo de procesador que las soluciones existentes. El método propuesto se prueba en un robot móvil LEGO NXT en exteriores con resultados satisfactorios.

Palabras clave: Robots móviles, localización, fusión de datos, filtros de Kalman, sistema de posicionamiento global, sistemas basados en eventos, recursos limitados, *LEGO NXT*.

1. INTRODUCCIÓN

El proceso de localización, el determinar la posición y orientación en el espacio (postura), es un aspecto fundamental en robótica móvil. Sin esta información el robot no sería capaz de seguir una trayectoria, navegar en el entorno, ir a una posición meta o volver al punto de inicio del movimiento, lo cual reduce la mayoría de las funciones del robot móvil. Este aspecto ha sido ampliamente estudiado en la literatura ([19],[21]) y su solución no es trivial, debido a que la información del movimiento del robot se obtiene de sensores con ruido y no-linealidades. Este aspecto debe ser tomado en cuenta por el método de localización para evitar el crecimiento desmedido en la incertidumbre de la estimación de la postura.

Existen distintos métodos de localización según la fuente de información utilizada. Si la posición inicial del robot antes de iniciar el movimiento es conocida, la postura del robot se estima utilizando la información *local* de movimiento del robot para calcular la distancia recorrida desde el punto inicial mediante el procedimiento conocido como *odometría*, *navegación por estima* ([19] y [16]) o *estimación local* de la postura [11]. Este procedimiento tiene la ventaja de un tiempo de respuesta bajo pero el inconveniente de que el error crece indefinidamente. Por el contrario, se puede utilizar un sensor complejo para obtener directamente la postura absoluta del robot en el entorno [8], empleando un sensor global como el GPS sin tener que usar la información local. Este procedimiento es conocido como *estimación global* de la postura, cuenta con buena precisión dependiendo del sensor utilizado pero tiene la gran desventaja de tener un tiempo de respuesta elevado.

Ambas estrategias, local y global, se combinan con el fin de mejorar el tiempo de respuesta del método de localización, el cual debe ser lo suficientemente rápido para el buen funcionamiento del algoritmo de navegación del robot. Generalmente se utiliza la información local como principal fuente de información y se corrige el error en la estimación cuando está disponible la información global ([10], [21]). La integración de ambos métodos se realiza habitualmente mediante un esquema de fusión basado en alguna versión del filtro del Kalman (KF) tal como el KF lineal, el extendido (EKF) o el unscented (UKF) (ver algoritmos en [9], [22], [12] y [20]). Estos métodos toman en cuenta las distintas precisiones de los sensores disponibles (ruido aso-

ciado) y un modelo de movimiento del robot para realizar la fusión de forma óptima e incrementar la precisión de la localización ([13],[4] y [6]).

Existen múltiples ejemplos de métodos de fusión sensorial utilizando el KF y la información proveniente del GPS para el caso de un robot o vehículo en configuración Ackermann. Éstos se resumen a continuación junto con la teoría básica de los sensores GPS.

2. TECNOLOGÍA GPS PARA LOCALIZACIÓN ASISTIDA

El sistema de posicionamiento global (GPS) es un sistema basado en satélites que se emplea para localizar cualquier punto en la Tierra. Provee información sobre la posición 3D de forma continua y en tiempo real permitiendo la navegación y sincronización por todo el planeta. Este sistema está formado por tres segmentos [7]. El segmento espacial consta de 29 satélites orbitando alrededor de la Tierra, con cuatro relojes atómicos cada uno. Cada satélite envía una señal de radio de baja frecuencia a 300.000 km/s con su información de posición y tiempo, además de un código único que permite su identificación en el receptor. El segmento de control consta de una estación de control principal junto con cuatro adicionales no tripuladas, que realizan el rastreo de los satélites y redirigen su posición en caso de que difiera con la trayectoria preestablecida. Finalmente el segmento de usuario, formado por los receptores GPS, los cuales proveen el posicionamiento utilizando las señales recibidas por los satélites ya tratadas.

Para calcular su posición, los receptores miden la distancia hasta cada satélite rastreando, utilizando el tiempo que tarda la señal del satélite en llegar a la superficie terrestre (satellite ranging). Esto es posible gracias a la sincronización entre satélites y receptor. Al multiplicar este tiempo por la velocidad de transmisión se obtiene la distancia requerida. Con éstas se determina la posición absoluta del receptor mediante triangulación. El primer satélite forma una esfera, con centro en su posición y radio la distancia con el receptor, en la que se sabe que éste se encuentra. Un segundo satélite genera su esfera, estrechando el espacio donde se localiza el receptor. Por último, con la de un tercer satélite la incógnita se reduce a dos puntos, de los cuales uno se rechaza porque generalmente no estará en la Tierra. Un cuarto satélite ayuda a corregir errores de medición del tiempo del reloj.

Existen diversas fuentes de error que afectan la precisión de la posición calculada por el sensor GPS. Aparte de los factores ambientales que pueden afectar los tiempos de transmisión y disminuir la intensidad de las señales recibidas, existen problemas inherentes a la navegación en medios urbanos. Por ejemplo la interferencia electromagnética que puede afectar los receptores GPS. Además, una pequeña variación o error latente en el reloj del receptor puede suponer una diferencia considerable en el cálculo de la posición. Finalmente si la señal del satélite se refleja en la superficie de algún edificio, se pueden recibir en el receptor dos señales, la reflejada y la que viene directo del satélite. Esto genera el problema conocido como error multiruta o multicamino [21].

Como se mencionó con anterioridad, con el fin de mitigar los problemas inherentes al sistema GPS y de mejorar la precisión en la medición de la postura del robot, se utilizan los esquemas de fusión KF. Para el caso de robots con configuración Ackermann en [5] se equipa un vehículo con una unidad inercial (IMU) de alta precisión junto con un GPS de doble antena para estimar la postura del robot utilizando un KF para estimar los parámetros necesarios del modelo. En [24] se utiliza un EKF para realizar la fusión con la ayuda de una red neuronal con buenos resultados sobre la carretera. De forma similar, en [3] se utilizan dos GPS cinemáticos de tiempo real con tres antenas y una IMU para estimar la postura del vehículo utilizando un EKF adaptativo que toma en cuenta las condiciones reales de operación para reducir el error de estimación. En [18] se utiliza un algoritmo que adapta la estructura del modelo utilizado en línea mientras se encuentra disponible la señal GPS. Se utiliza en cascada con un KF, en las extensas pruebas en carretera se obtienen buenos resultados ante fallos en el GPS. Finalmente en [21] se puede encontrar una revisión bastante completa de los métodos de navegación y localización para robots y vehículos de configuración Ackermann.

Todos estos ejemplos comparten aspectos en cuanto a la implementación de la fusión basada en KF. La mayoría de los algoritmos propuestos son costosos en cuanto a consumo de recursos computacionales (necesitan inversión y manipulación de matrices grandes), ancho de banda y baterías. Este aspecto impide la implementación en plataformas de bajo coste y recursos limitados. Con el objetivo de solucionar esta limitación, se propone en el presente trabajo un algoritmo basado en fusión KF apto para plataformas de este tipo. Este algoritmo continúa el trabajo previo realizado para robots de configuración diferencial en interiores [15].

El resto del artículo se organiza de la siguiente manera. En la siguiente sección se presenta el modelo del robot en configuración Ackermann utilizado

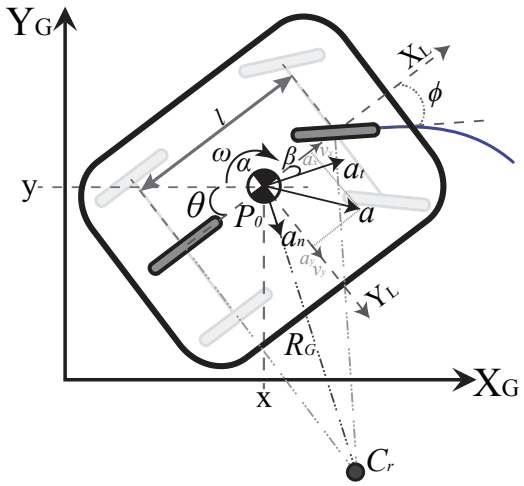


Figura 1: Cinemática del robot en configuración Ackermann

en la fusión sensorial. En la sección 4, se muestra el filtro EKF propuesto para mejorar la estimación de la postura utilizando la actualización global por eventos. En la sección 5 se presentan la implementación y pruebas realizadas con el robot en exteriores y finalmente, en la última sección, se plantean las conclusiones del presente trabajo.

3. MODELO DE UN ROBOT EN CONFIGURACIÓN ACKERMANN

Un robot en configuración Ackermann consiste en un cuerpo rígido con centro de masa y gravedad P_0, radio de rotación R_G, masa M_G y un momento de inercia I_G con dos ruedas traseras no deformables y no orientables (fijas) separadas una distancia b entre ellas y a una distancia l de dos ruedas delanteras orientables (también a una distancia b entre ellas) tal y como se muestra en la figura 1. La dirección Ackermann modifica la orientación de las ruedas delanteras de tal forma que, a bajas velocidades, todas las ruedas cumplen con la condición de rodamiento puro sin deslizamiento lateral [23]. Esto se produce debido a que, a baja velocidad, cada rueda sigue una trayectoria curva con radios distintos pero con un centro instantáneo de rotación común C_r (Fig. 1). Esta configuración se analiza utilizando el modelo de la bicicleta ([23], [17]) suponiendo que el movimiento se restringe a un plano. El modelo cinemático de esta configuración representa la evolución de las velocidades del robot en un marco inercial fijo. La *postura* del robot se define mediante las coordenadas del punto $P_0 = (x, y)$ y el ángulo de avance θ en el marco de referencia *global* (X_G,Y_G) en la fig. 1. Conociendo las velocidades lineales y angulares (v y ω) en el marco de referencia local (X_L,Y_L),la velocidad

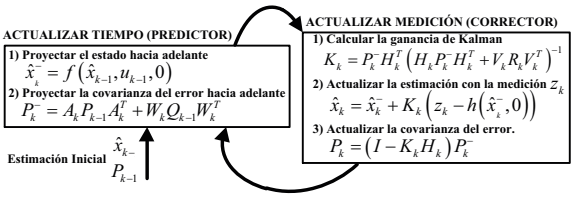


Figura 2: Algoritmo del filtro de Kalman Extendido [22]

global del robot se define como

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (1)$$

Discretizando e integrando recursivamente (1) con el tiempo de muestreo T_s, se obtiene la postura *global* del robot $\boldsymbol{L}$

$$\boldsymbol{L}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{k-1} + T_s \begin{bmatrix} \cos\theta_{k-1} & -\sin\theta_{k-1} & 0 \\ \sin\theta_{k-1} & \cos\theta_{k-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{k-1} \quad (2)$$

La aceleración lineal absoluta de P_0 en el marco de referencia global a, con componentes (a_x,a_y) se expresa en términos de las aceleraciones ($\dot{v_x}$,$\dot{v_y}$) y velocidades en el marco local utilizando sus componentes normal y tangencial (a_n,a_t) [23]

$$\begin{aligned} a_x &= \dot{v_x} - v_y\omega \\ a_y &= \dot{v_y} + v_x\omega \end{aligned} \quad (3)$$

La relación cinemática entre ϕ y ω se muestra en

$$\omega = (v_x \tan\phi)/l \quad (4)$$

El modelo formado por las ecuaciones (2) a (4) se utiliza en el algoritmo de fusión propuesto, el cual se expone a continuación.

4. ALGORITMO DE FUSION BASADO EN EVENTOS

El algoritmo propuesto parte del EKF [20] mostrado en la figura 2 para un sistema con entrada $\boldsymbol{u}_k \in \Re^u$, medición $\boldsymbol{z}_k \in \Re^m$ y estado $\boldsymbol{x}_k \in \Re^n$. El modelo se representa mediante las funciones no lineales $\boldsymbol{f}$ que relaciona el estado en el instante k con el instante anterior $k-1$ y $\boldsymbol{h}$ que relaciona x_k con z_k. Los términos $\boldsymbol{w}_k$, $\boldsymbol{v}_k$ representan los ruidos del proceso y medición en el instante k, los cuales se consideran con distribución de probabilidad independiente, blanca y normal con media cero. El filtro realiza la fusión sensorial en cada instante de tiempo k utilizando las matrices de covarianza del ruido del proceso $\boldsymbol{Q_k}$ y de la medición $\boldsymbol{R_k}$ para obtener el estado estimado $\hat{\boldsymbol{x}}_k$ (la postura del robot) y la covarianza del error de esta estimación $\boldsymbol{P}_k$. Se considera que el estado a utilizar en el filtro es la postura junto con las velocidades locales del

modelo propuesto. Además, las mediciones disponibles en los sensores son ϕ, $v_{x,enc}$, $v_{y,enc}$ y ω_{enc} de los encoders, ω_{gyr} de un giróscopo, ω_{comp} de una brújula, a_x, a_y obtenidos como el promedio de la medición de dos acelerómetros distintos y la postura global $L_{GPS} = (x, y, \theta)_{GPS}$ obtenida por el GPS.

El algoritmo propuesto reduce la carga computacional (procesador y memoria) al separar z_k en los componentes local y global. Con esto se determina la postura del robot cada instante k utilizando el modelo del robot con $\boldsymbol{z}_k = \begin{bmatrix} \phi & v_{x,enc} & v_{y,enc} & \omega_{enc} & \omega_{gyr} & \omega_{comp} & a_x & a_y \end{bmatrix}^T$ y únicamente se utiliza L_{GPS} para corregir la estimación local en instantes determinados (no en todo k). L_{GPS} se utiliza como una medición adicional de la postura cuando el error en la estimación local es suficientemente grande, indicado por un evento que utiliza la sección de la postura de P_k en el EKF. Esta $\boldsymbol{P}_{k,xy}$ se utiliza para obtener el área de los elipsoides de error con intervalo $3-\sigma$, utilizando la ecuación (5) siendo a_σ y b_σ los ejes de los elipsoides y $l = 3$ para el error $3-\sigma$. Con esto, el evento se genera utilizando la razón $\boldsymbol{R}_A$ del área de los elipsoides A_{ellip} con el área del robot Ackermann A_{ACK}. Este es un indicador normalizado del momento en el cual el error de la postura estimada es igual que el tamaño del robot. Por ejemplo, cuando $\boldsymbol{R}_A$ sobrepasa un nivel determinado, por ejemplo 1,5 (indicando que el área del error es 1,5 veces el área del robot) se utiliza L_{GPS} para corregir la postura del robot. Este límite se escoge como un compromiso entre el número de actualizaciones que se realizan (consumo de ancho de banda, procesador y energía) y la precisión deseada en la estimación de la postura.

$$\begin{array}{c} \boldsymbol{P}_{k,xy} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix} \\ T_\sigma = \sqrt{\sigma_x^4 + \sigma_y^4 - 2\sigma_x^2\sigma_y^2 + 4\sigma_{xy}^4}, \\ A_{ACK} = b \cdot c \end{array} \quad \begin{array}{c} a_\sigma = \sqrt{\frac{2l^2|P_{xy}|}{\sigma_x^2+\sigma_y^2+T_\sigma}} \\ b_\sigma = \sqrt{\frac{2l^2|P_{xy}|}{\sigma_x^2+\sigma_y^2-T_\sigma}} \\ A_{ellip} = \pi a_\sigma b_\sigma \end{array} \qquad (5)$$

Como condición adicional, se realiza la actualización utilizando la información del GPS si el número de satélites utilizados por el sensor es mayor a un valor mínimo preestablecido (por ejemplo 6 satélites). Con esto se busca garantizar una mayor precisión en la fusión sensorial. Esto se incorpora en la fusión modificando los valores correspondientes en la matriz R de forma dinámica, utilizando una función inversamente proporcional, es decir, si el número de satélites es alto entonces lo valores correspondientes a L_{GPS} en la matriz R deben ser bajos (de esta forma el EKF da mayor importancia a estos datos) y viceversa. Explicado el método de fusión propuesto se procede a exponer su implementación y pruebas realizadas

5. IMPLEMENTACIÓN Y PRUEBA DEL ESQUEMA DE FUSIÓN DE DATOS

El esquema de fusión propuesto se implementa en un robot móvil en configuración Ackermann LEGO como plataforma de recursos limitados para realizar la fusión sensorial y una tarjeta IGEP para almacenar los datos de la prueba y permitir el acceso al GPS mediante la conexión USB. Se utiliza un HUB USB para conectar el LEGO a la IGEP (GPS) y para proveer la alimentación a la IGEP utilizando la batería LiPo (con un convertidor DC-DC a 5V).

5.1. DESCRIPCIÓN DEL LEGO ®NXT

LEGO Mindstorms ®NXT es una plataforma de robot móvil de bajo coste. Incorpora la unidad de control NXT, basada en un microcontrolador ARM7 de 32-bits, con 256 Kbytes FLASH y 64 Kbytes de RAM. Cuenta además con un puerto USB 2.0 y un dispositivo inalámbrico Bluetooth clase II, V2.0. La nueva unidad de control tiene 4 entradas (una de ellas proporciona una expansión IEC 61158 Tipo 4/EN 50 170 para usos futuros) y 3 salidas analógicas. La versión básica del LEGO proporciona 4 tipos de sensores: contacto, luz, sonido y distancia. Pero, además de estos, actualmente se pueden comprar una gran variedad de distintos tipos de sensores como, por ejemplo, cámaras para aplicaciones de visión, brújula, acelerómetros, giróscopos, buscadores de infrarrojos, etc. [2]. Otros componentes LEGO muy importantes son los actuadores, los cuales consisten en motores de corriente continua que incorporan encoders integrados y de 360 grados de resolución por vuelta. Se puede encontrar una descripción más detallada de los motores de LEGO Mindstorms NXT en [2]. El robot LEGO NXT usado en las pruebas con configuración Ackermann se muestra en la figura 3. Los sensores usados para el esquema de fusión son dos acelerómetros, un giróscopo, una brújula magnética y los encoders de las ruedas. Se utiliza además un diferencial para regular las velocidades de las ruedas traseras durante el giro.

5.2. DESCRIPCIÓN DE LA TARJETA IGEP

La tarjeta IGEPv2 de ISEE [1] es un sistema embebido con un tamaño poco superior al de una tarjeta de crédito que, mediante una arquitectura TI OMAP3530, incorpora un microprocesador ARM CORTEX A8 core a 720MHz. y un DSP TMS320C64x+. La IGEPv2 incorpora además un acelerador gráfico hardware, adminis-

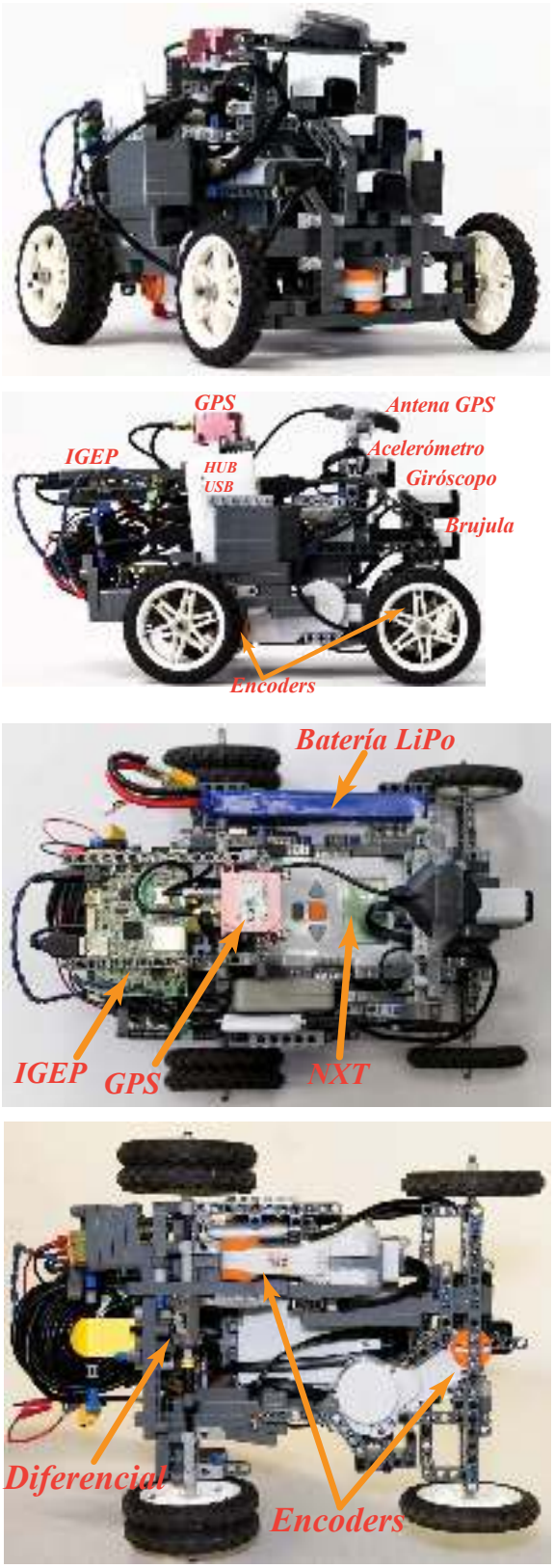


Figura 3: LEGO NXT en configuración Ackermann utilizado en las pruebas experimentales

trador de energía TPS65950, 512MB de memoria RAM y Flash, así como de diversos sistemas de comunicaciones, como Ethernet, Wi-Fi, Bluetooth, 2 puertos USB, DVI-D/HDMI, etc. Ésta se observa colocada sobre el LEGO en la figura 3. Para programarla se emplea el IDE Eclipse para escribir las aplicaciones en C++, las cuales son transferidas a la IGEP mediante la tarjeta microSD incorporada. Al tener instalada una distribución GNU/Linux como sistema operativo en la misma, se puede compilar mediante el compilador g++ y ejecutar sin problemas. Para realizar estas acciones se establece previamente conexión con el software PuTTY mediante ssh vía Wi-Fi, lo cual proporciona el control del entorno de consola de la IGEP desde un PC remoto. Esta tarjeta se programó para transmitir los mensajes con la posición del GPS IG-500N de SBG Systems al LEGO NXT y además para servir de supervisor y almacenar los datos provenientes del LEGO y del GPS. La IGEP no se utiliza para realizar los cálculos asociados a la fusión sensorial del filtro propuesto, ya que esta función la realiza el LEGO NXT.

5.3. PRUEBAS CON EL LEGO ®NXT

Para mostrar el funcionamiento del método propuesto, se establece que el robot siga una trayectoria de referencia cuadrada en interiores durante 30 minutos y con una velocidad promedio de 1 minuto por vuelta. Para esto se utiliza un control por punto descentralizado [14] para el control de trayectoria y un controlador P para el accionamiento de los motores. En lugar del GPS se utiliza una cámara cenital para medir la trayectoria real seguida por el robot. Esta prueba se muestra en la figura 4 en donde, para facilitar la visualización, se exponen únicamente la primera vuelta estimada por el EKF junto con las 30 vueltas medidas mediante la cámara. Se observa que el algoritmo propuesto tiene un comportamiento adecuado, estimando la postura de forma similar a la medida por la cámara lo que produce un seguimiento suave de la trayectoria cuadrada durante toda la prueba.

Para la prueba en exteriores se realiza el control mediante un mando a distancia ya que aún falta por implementar la evasión de obstáculos en el robot. Se realiza un recorrido en sentido horario sobre una pista deportiva de superficie plana, utilizando el GPS IG-500N sobre el LEGO como fuente de información global. Además se mide la trayectoria seguida por el LEGO con un GPS diferencial con el fin de obtener con mayor precisión la trayectoria seguida en el recorrido. Los resultados de esta prueba y el robot modificado con el receptor del GPS diferencial se muestran en la figura 5 y 6. Se observa que la estimación del algoritmo por eventos sigue aproximadamente la del

GPS, según aumenta el error en la estimación (covarianza del error de la postura del robot) se genera el evento que produce la actualización utilizando información global y por lo tanto se elimina el bias característico del método de estimación local. Esto manteniendo una precisión aceptable según las gráficas del error y error absoluto (respecto al GPS, figura 6) y sin necesidad de actualizar constantemente la estimación utilizando la información del GPS y por lo tanto usando más recursos de los necesarios. A pesar del buen comportamiento se observa que el GPS utilizado no es tan preciso como el GPS diferencial por lo que, aunque se utiliza información global, existe un error inherente al sensor utilizado y que se traslada a la estimación del algoritmo. Cabe destacar que la actualización se puede hacer de forma más frecuente para eliminar el bias de forma más regular, esto se realiza según los requerimientos de la aplicación.

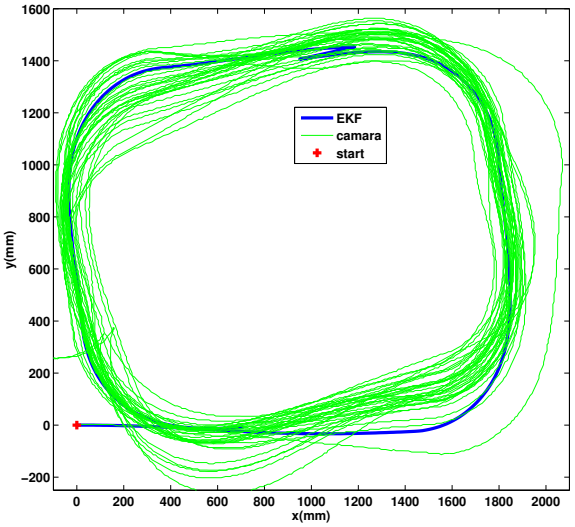


Figura 4: Prueba en interiores con camara cenital

6. CONCLUSIONES

Se ha presentado un esquema de fusión EKF basado en eventos y eficiente desde el punto de vista de utilización de recursos computacionales para incorporar la información global de un GPS a la estimación local basada en odometría utilizando un robot de configuración Ackermann. Los resultados muestran que la respuesta del método es adecuada tanto en interiores (utilizando una cámara en lugar del GPS) como en exteriores manteniendo acotado el error de estimación y utilizando adecuadamente la información del GPS. Existen varias formas de mejorar el presente aporte, mejorando las funciones de ajuste del EKF según en número de satélites para incorporar más información relevante al GPS para detectar problemas de multiruta y desvió de la señal por condiciones atmosféricas. Se pueden incorporar sensores más precisos e inclusive implementar un filtro para el GPS dentro de la IGEP que actualice en cascada al filtro dentro del LEGO con el fin de detectar la pérdida de la señal de los satélites. Como trabajo futuro se está trabajando en la utilización del robot con capacidad de posicionamiento global para localizar a un grupo de robots de recursos limitados sin acceso a información global pero con medios de comunicación y detección mutua.

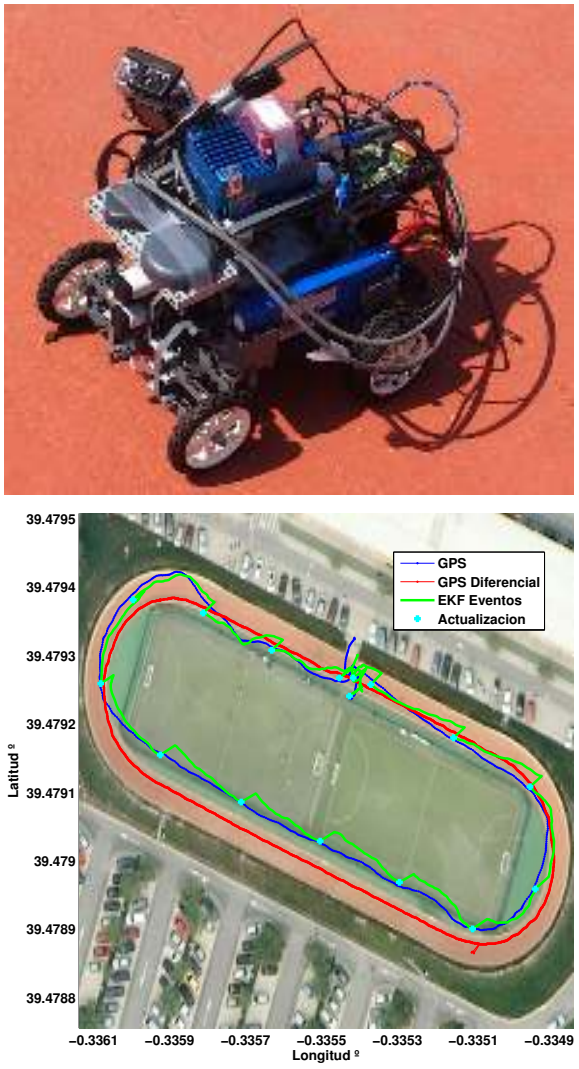


Figura 5: Prueba en exteriores con GPS diferencial

Agradecimientos

Los autores desean expresar su agradecimiento al Ministerio de Ciencia e Innovación de España por la financiación parcial de este trabajo bajo los proyectos de investigación DPI2011-28507-C02-01 y DPI2010-20814-C02-02. Además se agradece el soporte financiero por parte de la Universidad de Costa Rica y el apoyo recibido por Instituto U. de Automática e Informática Industrial (ai2) de la U.

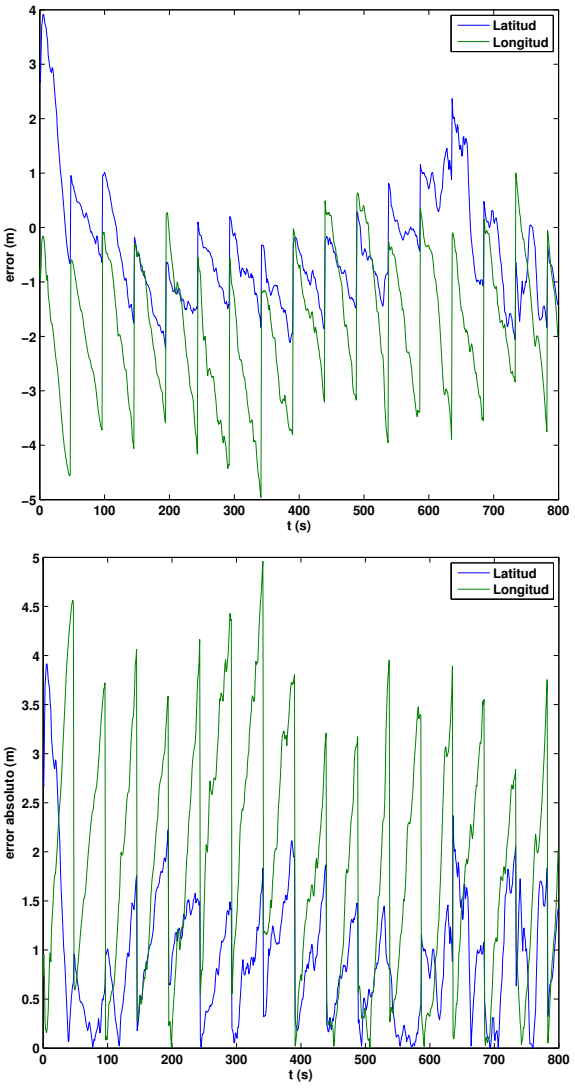


Figura 6: Evolución del error de estimación

Politècnica de València con el Plan de Ayudas a la I+D+i 2012.

Referencias

[1] Información en línea de la igep (isee) disponible en: http://www.igep-platform.com/.

[2] Información en línea del lego nxt (2011), disponible en: http://www.mindsensors.com/ y http://www.hitechnic.com/ motores del lego nxt: http://www.philohome.com/motors/motorcomp.htm.

[3] F. Aghili and A. Salerno. Driftless 3-d attitude determination and positioning of mobile robots by integration of imu with two rtk gpss. *Mechatronics, IEEE/ASME Transactions on*, 18(1):21 –31, feb. 2013.

[4] Ángel Valera, M. Weiss, M. Vallés, and J. L. Diez. Bluetooth-networked trajectory control of autonomous vehicles. *Eight IFAC Symposium on Cost Oriented Automation*, 8, 2007.

[5] D. Bevly, J. Ryu, and J. Gerdes. Integrating ins sensors with gps measurements for continuous estimation of vehicle sideslip, roll, and tire cornering stiffness. *Intelligent Transportation Systems, IEEE Transactions on*, 7(4):483 –493, dec. 2006.

[6] M. Boccadoro, F. Martinelli, and S. Pagnottelli. Constrained and quantized kalman filtering for an rfid robot localization problem. *Autonomous Robots*, 29:235–251, 2010.

[7] P. Enge and P. Misra. Special issue on global positioning system. *Proceedings of the IEEE*, 87(1):3–15, 1999.

[8] C. Fuchs, N. Aschenbruck, P. Martini, and M. Wieneke. Indoor tracking for mission critical scenarios: A survey. *Pervasive and Mobile Computing*, 7(1):1 – 15, 2011.

[9] M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice Using Matlab*. John Wiley & Sons, 2001.

[10] N. Houshangi and F. Azizi. Accurate mobile robot position determination using unscented kalman filter. *Canadian Conference on Electrical and Computer Engineering*, pages 846 – 851, 2005.

[11] D. Hyun, H. S. Yang, H.-S. Park, and H.-J. Kim. Dead-reckoning sensor system and tracking algorithm for 3-d pipeline mapping. *Mechatronics*, 20(2):213 – 223, 2010.

[12] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *Automatic Control, IEEE Transactions on*, 45:477 –482, mar 2000.

[13] J. Kim, Y. Kim, and S. Kim. An accurate localization for mobile robot using extended kalman filter and sensor fusion. In *Neural Networks, 2008. IJCNN 2008. IEEE International Joint Conference on*, pages 2928 –2933, june 2008.

[14] L. Marín, M. Vallés, Ángel Valera, and P. Albertos. Control de trayectorias en el robot móvil e-puck. *XXXI Jornadas Automatica JAEN*, 2010.

[15] L. Marín, M. Vallés, Ángel Valera, and P. Albertos. Mejora en la navegación de robots con recursos limitados mediante fusión de datos de distintos sensores. *XXXII Jornadas Automatica SEVILLA*, 2011.

[16] O. Michel, F. Rohrer, N. Heiniger, and wikibooks contributors. *Cyberbotics'Robot Curriculum*. Cyberbotics Ltd. and Wikibooks contributors, `http://en.wikibooks.org/wiki/Cyberbotics\'_Robot_Curriculum`, 2009.

[17] R. Rajamani. *Vehicle Dynamics And Control*. Mechanical Engineering Series. Springer US, 2012.

[18] Z. Shen, J. Georgy, M. J. Korenberg, and A. Noureldin. Low cost two dimension navigation using an augmented kalman filter/fast orthogonal search module for the integration of reduced inertial sensor system and global positioning system. *Transportation Research Part C: Emerging Technologies*, 19(6):1111 – 1132, 2011.

[19] R. Siegwart and I. Nourbakhsh. *Introduction to autonomous mobile robots*. The MIT Press Cambridge, Massachusetts London, England, 2004.

[20] D. Simon. *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*. John Wiley & Sons, 2006.

[21] I. Skog and P. Handel. In-car positioning and navigation technologies - a survey. *Intelligent Transportation Systems, IEEE Transactions on*, 10(1):4 –21, march 2009.

[22] G. Welch and G. Bishop. An introduction to the kalman filter, 2007.

[23] J. Wong. *Theory of Ground Vehicles*. John Wiley & Sons, 2008.

[24] T. Zhang and X. Xu. A new method of seamless land navigation for gps/ins integrated system. *Measurement*, 45(4):691 – 701, 2012.

IMPLEMENTACIÓN DE CONTROLADORES DE FUERZA Y POSICIÓN PARA UN MANIPULADOR PARALELO DE 3DOF

Jose I. Cazalilla[a], Miguel Corberán, M. Díaz-Rodríguez[b], Ángel Valera[a], Vicente Mata[c], Marina Vallés[a]
[a] Instituto de Automática e Informática Industrial
[c] Centro de Investigación de Tecnología de Vehículos
Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia (España)
[b] Dpto. de Tecnología y Diseño, Facultad de Ingeniería
Universidad de los Andes, 5101a, Núcleo La Hechicera, Mérida (Venezuela)
e-mail: {jcazalilla, giuprog, mvalles}@ai2.upv.es, migcorrui@gmail.com. dmiguel@ula.ve, vmata@mcm.upv.es

Resumen

El objetivo de este artículo es presentar el desarrollo de los controladores en tiempo real para un robot paralelo manipulador (PM) de 3 grados de libertad. El robot es capaz de generar y controlar un movimiento de traslación y dos movimientos de rotación (alabeo y cabeceo). Se pueden encontrar aplicaciones para este tipo de manipuladores en el campo de la simulación de conducción así como en el campo de la biomecánica. Para este robot, se ha desarrollado una arquitectura de control abierta, la cual permite implementar y validar diferentes esquemas de control dinámicos para PM con 3 grados de libertad. Así, el robot desarrollado se puede usar como banco de pruebas para validar distintos esquemas de control. En este artículo, se presenta cómo se han implementado distintos esquemas de control de posición y fuerza.

Palabras Clave: manipulador paralelo, control de robots, control de fuerza, mecatrónica, cinemática.

1 INTRODUCCIÓN

Un Manipulador Paralelo (PM) consiste en una plataforma móvil conectada a una base fija por medio de varias cadenas cinemáticas. Estos manipuladores tienen un elemento final conectado a la plataforma móvil. Los PMs tienen ventajas respecto a los robots serie que consisten, esencialmente, en que la carga se reparte entre varias articulaciones que conectan la plataforma móvil a la base. Así, los PMs presentan alta rigidez, capacidad de transportar carga, alta velocidad y exactitud alta. Sin embargo, los PMs tienen espacios de trabajo pequeños y problemas de singularidad. Además, la solución de la cinemática directa, la dinámica del sistema, y el control de los PMs son difíciles de desarrollar comparados con los robots serie.

Debido a sus ventajas, los PMs tienen varias aplicaciones. Estos manipuladores se han implementado para: simuladores de movimiento, máquinas de prueba de neumáticos, simuladores de vuelo, y aplicaciones médicas. Se pueden encontrar distintas arquitecturas mecánicas en trabajos como [10], [19], [15] o [21]. Las investigaciones sobre PMs se centraron inicialmente sobre plataformas con 6 grados de libertad (DOF). Sin embargo, para muchas aplicaciones no se requieren 6-DOF. Debido a esto, el número de trabajos de investigación sobre PMs con menos grados de libertad aumentó. El motivo es que los PMs con menos de 6-DOF mantienen las ventajas inherentes a los mecanismos paralelos pero presentan ventajas adicionales en cuanto a la reducción de costes de fabricación y de operación. Por ejemplo, el conocido Delta Robot tienen 3 DOF traslacionales (3T) [21]. Este PM es adecuado para tareas de pick-and-place [5]. Los PM 3T traslacionales no se usan únicamente en tareas de pick-and-place, sino que también se usan en aplicaciones médicas como equipos de resucitación cardiopulmonar [16], y como máquinas herramienta [14]. Cuando se requieren movimientos traslacionales y rotatorios, se proponen las arquitecturas 3-PRS [4] y 3-RPS [12] en donde la notación R, P y S simboliza articulación de revolución, prismática o esférica, respectivamente.

El objetivo de este artículo es presentar el robot desarrollado, y cómo se han implementado diferentes tipos de controladores en tiempo real de posición y de fuerza de forma que se generan y controlan un movimiento traslacional (1T) y dos rotativos (2R) (alabeo y cabeceo).

Con este tipo de movimiento (1T2R) se puede simular, por ejmplo, el movimiento de conducir: el ángulo de giro (alabeo, γ) reproduce el giro del vehículo, mientras que el ángulo de elevación (o cabeceo β) crea ilusiones de aceleración y freno. Además, los movimiento de subida y bajada se pueden reproducir controlando la elevación (z). Otra posible aplicación es en el tratamiento de lesiones de tobillo en las cuales la plataforma móvil simula la trayectoria del pie durante un ejercicio de fisioterapia [13].

Para implementar y probar esquemas de control dinámico para PM con 3-DOF se ha desarrollado una arquitectura abierta que se presenta en el artículo. Esta arquitectura abierta permite estudiar el control para mejorar la exactitud de los PMs [20]. Así, se usa el robot desarrollado como banco de pruebas para evaluar distintos esquemas de control. Los esquemas que se han probado son controladores de posición basados en en la dinámica inversa [17], utilizándose el modelo identificado en [9].

Cabe destacar que hay una gran variedad de aplicaciones que implican tareas de ensamblado donde una o más partes deben manipularse asegurando un contacto adecuado, o bien operaciones de mecanizado de superficies como fresado, rectificado, desbarbado o pulidos [1]. Por lo tanto, para este tipo de aplicaciones es necesario desarrollar un control preciso de la fuerza que ejerce el robot sobre el entorno. En este trabajo se ha abordado y desarrollado un control fuerza/posición para el robot paralelo.

2 EL ROBOT PARALELO

La elección de la arquitectura y movimiento del robot paralelo han venido determinadas por la necesidad de desarrollar un robot de bajo coste capaz de generar rotación angular en dos ejes y elevación como movimiento lineal. Se consideraron dos arquitecturas alternativas: 3-RPS y 3-PRS. Se seleccionó la arquitectura 3-PRS tras comparar las ventajas e inconvenientes de cada una de las alternativas. Por ejemplo, una de las ventajas de la arquitectura PRS es que los actuadores se localizan en la base fija mientras que en la arquitectura 3-RPS, los actuadores se mueven junto con las articulaciones de revolución.

La figura 1 muestra el robot diseñado. El manipulador se puede ver como tres piernas que conectan la plataforma móvil con la base. Cada pierna consiste en: un motor, que mueve un husillo a bola, un deslizador y un vástago de conexión.

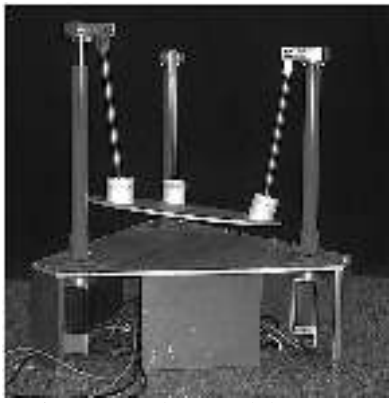


Figura 1: Robot paralelo 3-PRS

Los motores de cada pierna son servomotores de corriente continua brushless BMS465 AH (Aerotech) equipados con un amplificador de potencia Aerotech BA10. El sistema de control ha sido desarrollado en un computador personal industrial. Al estar basado en un PC este sistema tiene dos ventajas principales: la primera es que es un sistema completamente abierto, proporcionando una plataforma potente que permite la programación de tareas de alto nivel. Así se puede desarrollar aplicaciones de generación automática de trayectoria, estrategias de control basadas en sensorización externa (como visión artificial o sensores de fuerza), etc. La segunda gran ventaja que presenta la arquitectura de control es su coste puesto que el sistema operativo y de desarrollo está basado en software libre y gratuito.

2.1 MODELO CINEMÁTICO

Para poder establecer el control del robot se han tenido que desarrollar los modelos cinemáticos directo e inverso.

La localización de las 9 coordenadas generalizadas necesarias para definir este robot se muestran en la Figura 2.

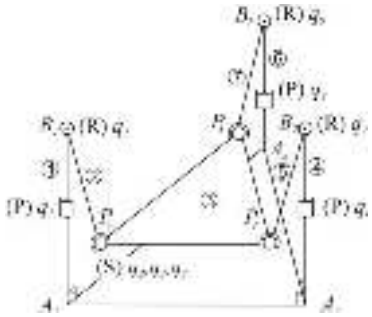


Figura 2: Localización del sistema de coordenadas

donde q_1, q_6 y q_8 son las articulaciones activas, que corresponden con las articulaciones prismáticas del robot que se controlan mediante los servomotores brushless.

En la figura se puede ver que la longitud entre p_i y p_j es constante e igual a l_m. Así, aplicando la aproximación geométrica:

$$f_1(q_1,q_2,q_6,q_7) = \left\| \left(\vec{r}_{A_1B_1} + \vec{r}_{B_1p_1}\right) - \left(\vec{r}_{A_1A_2} + \vec{r}_{A_2B_2} + \vec{r}_{B_2p_2}\right) \right\| - l_m = 0 \quad (1)$$

$$f_2(q_1,q_2,q_8,q_9) = \left\| \left(\vec{r}_{A_1B_1} + \vec{r}_{B_1p_1}\right) - \left(\vec{r}_{A_1A_3} + \vec{r}_{A_3B_3} + \vec{r}_{B_3p_3}\right) \right\| - l_m = 0 \quad (2)$$

$$f_3(q_6,q_7,q_8,q_9) = \left\| \left(\vec{r}_{A_1A_3} + \vec{r}_{A_3B_3} + \vec{r}_{B_3p_3}\right) - \left(\vec{r}_{A_1A_2} + \vec{r}_{A_2B_2} + \vec{r}_{B_2p_2}\right) \right\| - l_m = 0 \quad (3)$$

En el modelo cinemático directo se conoce la posición de los actuadores. El sistema expresado por las ecuaciones (1)-(3) es un sistema no lineal donde q_2, q_7 and q_9 son desconocidos. Para resolver este sistema no lineal se ha escogido el método numérico de Newton-Raphson (N-R). Este método converge de una forma rápida (convergencia cuadrática) cuando la estimación inicial está cerca de la solución deseada [11].

La localización de la plataforma móvil se define mediante un sistema de coordenadas locales asociado a ésta. Las coordenadas de las articulaciones esféricas de la plataforma se obtienen después de haber encontrado las coordenadas generalizadas de cada pierna del robot. Esas 3 articulaciones comparten el plano de la plataforma, por lo tanto se define el eje local X_p como un vector unitario $\vec{u}$ con la dirección dada por $p_1 p_2$. El eje Z_p se define por un vector unitario $\vec{v}$ perpendicular al plano definido el plano definido por los puntos p_1, p_2 y p_3. Finalmente, el eje Y_p (eje $\vec{w}$) se determina por el producto cruzado $\vec{u} \times \vec{v}$. La matriz de rotación de la plataforma móvil viene dado por:

$$ {}^{O}R_p = \begin{bmatrix} \vec{u}^T & \vec{v}^T & \vec{z}^T \end{bmatrix} \tag{4} $$

Las coordinadas generalizadas restantes (q_3, q_4 y q_5) se pueden obtener a partir de la matriz de rotación.

Por otra parte, el problema cinemático inverso consiste en encontrar los movimientos de los actuadores lineales a partir del alabeo (γ) y cabeceo (β) y la altura (z) de la plataforma. Utilizando un sistema de ángulo fijo X-Y-Z, la matriz de rotación se puede definir como:

$$ {}^{O}R_p = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \tag{5} $$

En la ecuación (5) *c** y *s** se corresponde con el *cos(*)* y *sin(*)* respectivamente. Así, dados los ángulos γ y β se puede calcular el ángulo α de guiñada como:

$$ \alpha = \operatorname{atan2}\left(s_\beta s_\gamma, \left(c_\gamma + c_\beta\right)\right) \tag{6} $$

Una vez calculado el ángulo α, se pueden calcular los términos restantes de la matriz de rotación, y las posiciones de los actuadores se pueden obtener mediante la expresión siguiente [21].

$$ \begin{aligned} q_1 &= p_x^2 + p_y^2 + p_z^2 + 2h\left(p_x u_x + p_y u_y + p_z u_z\right) - 2g\,p_x - 2gh u_x + g^2 + h^2 \\ q_6 &= p_x^2 + p_y^2 + p_z^2 - h\left(p_x u_x + p_y u_y + p_z u_z\right) + \sqrt{3}h\left(p_x v_x + p_y v_y + p_z v_z\right) \\ &\quad + g\left(p_x - \sqrt{3}p_y\right) + gh\left(u_x - \sqrt{3}u_y\right)/2 + gh\left(v_x - \sqrt{3}v_y\right)/2 + g^2 + h^2 \\ q_8 &= p_x^2 + p_y^2 + p_z^2 - h\left(p_x u_x + p_y u_y + p_z u_z\right) - \sqrt{3}h\left(p_x v_x + p_y v_y + p_z v_z\right) \\ &\quad + g\left(p_x - \sqrt{3}p_y\right) - gh\left(u_x - \sqrt{3}u_y\right)/2 + gh\left(v_x - \sqrt{3}v_y\right)/2 + g^2 + h^2 \end{aligned} \tag{7} $$

En la ecuación $h = l_m/\sqrt{3}$, $g = l_b/\sqrt{3}$, $p_x = -hu_y$, $p_y = -h\left(u_x - v_y\right)$, $p_z = z$ y l_b es la distancia $A_i A_j$.

2.2 JACOBIANO ROBOT PARALELO

Como se sabe, el modelo cinemático de un robot busca las relaciones entre las variables articulares y la posición del extremo del robot. En esta relación no se tienen en cuenta las fuerzas o pares que actúan sobre el robot. Sin embargo, sí que permite conocer la relación entre las derivadas de las coordenadas articulares y del extremo del robot.

La relación entre las velocidades de las coordenadas articulares y las de la posición y orientación del extremo del robot se obtiene a través de la denominada matriz Jacobiana.

Para obtener la Jacobiana en la Figura 2 se puede comprobar la siguiente expresión vectorial:

$$ \vec{p} + \vec{b}_i = \vec{a}_i + q_1 \cdot \vec{n}_{1i} + l_r \cdot \vec{r}_{2i} \tag{8} $$

donde $\vec{r}_{1i}$ y $\vec{r}_{2i}$ son vectores unitarios, l_r es la longitud del eslabón acoplador, $\vec{p}$ es el vector de posición del elemento terminal, $\vec{b}_i$ es el vector de posición entre el elemento terminal y el par esférico *i*. Por último $\vec{a}_i$ define la posición de la junta prismática y el eje coordenado global. En el caso del robot paralelo ubicado en la rama 1.

Derivando la ecuación (8):

$$ \vec{v}_p + \vec{\omega}_p \times \vec{b}_i = \dot{q}_i \cdot \vec{n}_{1i} + l_r \cdot \vec{\omega}_{2i} \times \vec{r}_{2i} \tag{9} $$

Para eliminar la coordenada pasiva contenida en $\vec{r}_{2i}$ se puede realizar el producto escalar en ambos lados de la ecuación por ese mismo vector:

$$ \vec{r}_{2i} \cdot \vec{v}_p + \vec{r}_{2i} \cdot \left(\vec{\omega}_p \times \vec{b}_i\right) = \vec{r}_{2i} \cdot \dot{q}_i \cdot \vec{n}_{1i} \tag{10} $$

Como $\vec{a} \cdot \vec{b} = \vec{b} \cdot \vec{a}$ y $\vec{a} \cdot \left(\vec{b} \times \vec{c}\right) = \vec{b} \cdot \left(\vec{c} \times \vec{a}\right)$:

$$ \vec{r}_{2i} \cdot \vec{v}_p + \left(\vec{b}_i \times \vec{r}_{2i}\right) \cdot \vec{\omega}_p = \vec{r}_{2i} \cdot \vec{n}_{1i} \cdot \dot{q}_i \cdot \tag{11} $$

Aplicando la ecuación anterior a cada una de las cadenas o piernas del robot:

$$ \begin{bmatrix} \vec{r}_{21} \cdot \vec{n}_{11} & 0 & 0 \\ 0 & \vec{r}_{22} \cdot \vec{n}_{12} & 0 \\ 0 & 0 & \vec{r}_{23} \cdot \vec{n}_{13} \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_6 \\ \dot{q}_8 \end{bmatrix} = J_q \cdot \dot{\vec{q}} \tag{12} $$

$$ \begin{bmatrix} \vec{r}_{21}^{\,t} & \left(\vec{b}_1 \times \vec{r}_{21}\right)^t \\ \vec{r}_{22}^{\,t} & \left(\vec{b}_2 \times \vec{r}_{22}\right)^t \\ \vec{r}_{23}^{\,t} & \left(\vec{b}3 \times \vec{r}_{23}\right)^t \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\gamma} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} = J_X \cdot \dot{\vec{X}} \tag{13} $$

De esta forma se verifica que:

$$J_q \cdot \dot{\vec{q}} = J_x \cdot \dot{\vec{X}} \tag{14}$$

Sin embargo hay que recordar que no todas las variables del vector $\dot{\vec{X}}$ son independientes. Para el robot paralelo las relaciones entre las coordenadas del elemento terminal restringidas vienen dadas por:

$$\begin{aligned} p_x &= -h\sin(\alpha)\cos(\beta) \\ p_y &= -1/2h(\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)\sin(\gamma) - \cos(\alpha)\cos(\gamma)) \\ \alpha &= \text{atan2}(\sin(\beta)\sin(\gamma), \cos(\gamma) + \cos(\beta)) \end{aligned} \tag{15}$$

Por ello es necesario encontrar el Jacobiano siguiente:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\gamma} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\delta x}{\delta z} & \frac{\delta x}{\delta \gamma} & \frac{\delta x}{\delta \beta} \\ \frac{\delta y}{\delta z} & \frac{\delta y}{\delta \gamma} & \frac{\delta y}{\delta \beta} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{\delta \alpha}{\delta z} & \frac{\delta \alpha}{\delta \gamma} & \frac{\delta \alpha}{\delta \beta} \end{bmatrix}}_{J_r} \begin{bmatrix} \dot{z} \\ \dot{\gamma} \\ \dot{\beta} \end{bmatrix} \tag{16}$$

De esta forma:

$$J_q \cdot \dot{\vec{q}} = J_x \cdot J_r \cdot \dot{\vec{X}}_c \tag{17}$$

Por consiguiente, la relación entre las coordenadas de las articulaciones y del elemento terminal viene dada por:

$$\dot{\vec{q}} = J_c \cdot \dot{\vec{X}}_c \tag{18}$$

donde:

$$J_c = J_q^{-1} \cdot J_x \cdot J_r \tag{19}$$

2.3 MODELO DINÁMICO

Si se necesita implementar controladores dinámicos es necesario que la ecuación del movimiento del robot se pueda expresar como:

$$M(\vec{q},\vec{\Phi}) \cdot \ddot{\vec{q}} + \vec{C}(\vec{q},\dot{\vec{q}},\vec{\Phi}) \cdot \dot{\vec{q}} + \vec{G}(\vec{q},\vec{\Phi}) = \vec{\tau} \tag{20}$$

La ecuación (20) muestra que la matriz de masas ***M***, el vector correspondientes a las fuerzas centrífugas y Coriolis ***C*** y el vector de fuerzas gravitacionales ***G*** dependen del vector de parámetros $\vec{\Phi}$ y de las fuerzas generalizadas $\vec{\tau}$.

Para poder identificar los parámetros dinámicos, el modelo del sistema se puede escribir en forma lineal respecto de los parámetros [7].

$$\text{K}(\vec{q},\dot{\vec{q}},\ddot{\vec{q}}) \cdot \vec{\Phi} = \vec{\tau} \tag{21}$$

En la ecuación (21), $\text{K}(\vec{q},\dot{\vec{q}},\ddot{\vec{q}})$ es la matriz de observación correspondiente al conjunto de coordenadas generalizadas, velocidades y aceleraciones. Para este robot paralelo se ha obtenido un modelo completo y reducido [8]. El modelo complete contiene todos los parámetros que afectan al comportamiento dinámico: los parámetros del cuerpo rígido, los parámetros de fricción viscosa y de Coulomb y los parámetros de la dinámica de los rotores de los actuadores.

Sin embargo, no todos estos parámetros se pueden identificar adecuadamente. Así se obtuvo un modelo dinámico reducido que contiene sólo los parámetros relevantes. Estos parámetros se determinaron considerando las simetrías de las piernas del robot, la significación estadística y la factibilidad física de los parámetros identificados.

Para analizar la robustez y validar los modelos dinámicos obtenidos, en [8] se desarrollaron un modelo virtual y un prototipo real del robot paralelo. Los resultados obtenidos indicaron que el modelo reducido tiene mejor rendimiento que el modelo completo, obteniéndose unos errores significativamente más pequeños. Además, el uso del modelo reducido no solo requiere un menor coste computacional, sino que además es capaz de proporcionar de seguimiento de trayectorias muy bueno.

3 DESARROLLO DEL CONTROL DEL ROBOT PARALELO

3.1 CONTROL DE POSICIÓN

Una vez resuelto el problema dinámico del robot paralelo e identificados y validados los parámetros dinámicos del robot, se puede abordar el control en tiempo real del manipulador paralelo. En este trabajo, se han implementado varias estrategias de control basadas en el método de dinámica inversa (o par calculado). Este tipo de control se analiza con mayor detalle en diversos textos como, por ejemplo, [6], [18], [23] o [3]. Las estrategias de control implementadas se basan en la de movimiento del robot (ecuación (20)) y vienen dadas por la expresión general siguiente:

$$\vec{\tau}_c = M(\vec{q})\vec{a} + \vec{C}(\vec{q},\dot{\vec{q}})\dot{\vec{q}} + \vec{G}(\vec{q}) \tag{22}$$

donde

τ_c : par generalizado de control

$M(\vec{q})$, $\vec{C}(\vec{q},\dot{\vec{q}})\dot{\vec{q}}$, $\vec{G}(\vec{q})$: matriz de inercias, vector de Coriolis y fuerzas gravitacionales del sistema

$\vec{a}$: acción de control lineal

En la ecuación (22) se puede observar como el controlador cancela los términos no lineales de la ecuación dinámica del robot (los términos de Coriolis y gravitacionales) sumando dichas fuerzas a la acción de control. Una vez canceladas el sistema se convierte en un sistema lineal simple de entrada-salida:

$$\ddot{\vec{q}} = \vec{a} \tag{23}$$

Dependiendo de la expresión que se utilice para el término $\vec{a}$ se pueden obtener diferentes estrategias de control, tal como se muestra en la Tabla 1:

Controlador	$\vec{a}$
Punto a punto	$-K_d\dot{\vec{q}} - K_p\vec{e}$
Control de trayectoria	$\ddot{\vec{q}}_d - K_d\dot{\vec{e}} - K_p\vec{e}$
C. Trayectoria con acción integral	$\ddot{\vec{q}}_d - K_d\dot{\vec{e}} - K_p\vec{e} - K_i\int_0^t \vec{e}(u)du$

Tabla 1. Acción de control lineal.

donde $\vec{e} = \vec{q} - \vec{q}_d$ y K_d, K_p y K_i son las matrices (diagonales y definidas positivas) de ganancias derivativas, proporcionales e integradoras.

El primero es un controlador punto a punto exponencialmente estable con una selección adecuada de las matrices K_d y K_p. El segundo controlador es similar al primero, pero en este caso se establece un control de trayectoria en el que la respuesta del robot debe seguir una trayectoria variable $\vec{q}_d(t)$ y sus sucesivas derivadas $\dot{\vec{q}}_d(t)$ y $\ddot{\vec{q}}_d(t)$ que describen la velocidad y la aceleración deseada. El tercer controlador es también un controlador de trayectoria en el que se ha añadido un término integral del error.

Para poder validar la metodología propuesta en este trabajo se han desarrollado varios esquemas de Matlab/Simulink. La Figura 3 muestra uno de estos esquemas. En ella se tiene el bloque *Cinematica_inversa*, que calcula el modelo cinemático inverso del robot descrito en el apartado 2.1, el bloque *C. Din Inv* que implementa el control por dinámica inversa punto a punto. El bloque *Robot Paralelo* implementa la ecuación dinámica del robot paralelo planteada en el apartado 2.3. Por último se ha implementado una función embebida que implementa la Jacobiana descrita en el apartado 2.2.

La Figura 4 muestra la referencia y la posición de la articulación q_1 del robot. Como se puede observar, se obtiene una respuesta muy buena en el seguimiento de la trayectoria. Para las articulaciones q_6 y q_8 la respuesta es completamente similar.

La Figura 5 permite verificar el correcto funcionamiento de la Jacobiana del robot paralelo. En ella se puede observar para la articulación q6 la velocidad y su estimación calculada a partir de la Jacobiana. Como se puede comprobar, la estimación que se ha obtenido es muy buena.

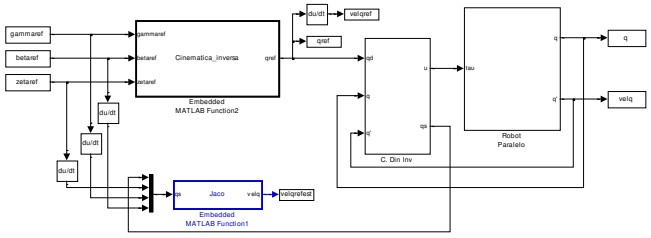


Figura 3: Esquema de control de posición

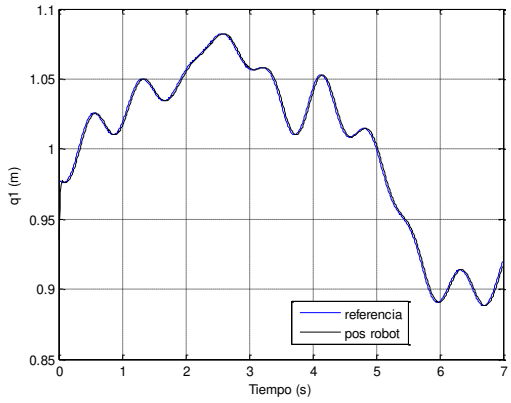


Figura 4: Referencia y posición articulación q_1

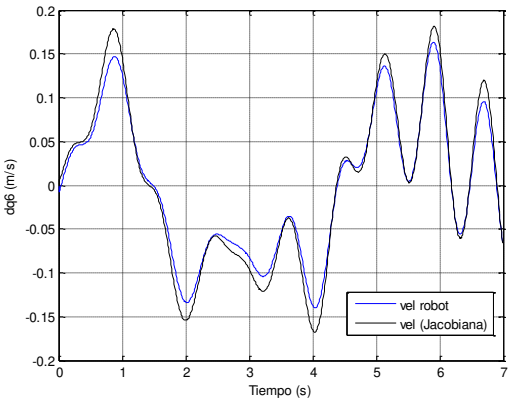


Figura 5: Velocidad articulación q_6

3.2 CONTROL DE FUERZA

Cuando se trabaja con el control de robots se suele tener únicamente un control de posición. Sin embargo el control de fuerza permite una adaptación flexible respecto a las imprecisiones en las piezas a fabricar, en el entorno o incluso en los robots. Lo que ocurre es que hace realmente poco tiempo los robots industriales no han proporcionado un control de fuerza.

Para establecer el control de fuerza se puede optar por diversos tipos de estrategias de control. El primer tipo está basado en el control explícito de control de fuerza, y cosiste en seguir el valor de referencia de fuerza mediante la realimentación de la fuerza que el robot está ejerciendo sobre el entorno.
Habitualmente el control explícito se restringe a un control lineal [22], como el clásico control PID de fuerza de la ecuación (24). El efecto de los 3 parámetros de un controlador PID es bien conocido y se ha descrito ampliamente en la literatura como por ejemplo en [2]. Así el término integral asegura un error de seguimiento cero, mientras que la función del término derivativo es amortiguar el sistema.

$$\tau = K_p(F_{ref} - f) + K_I \int (F_{ref} - f)dt + K_d \frac{d}{dt}(F_{ref} - f) \qquad (24)$$

donde τ es la acción de control, F_{ref} es la referencia de fuerza, f la fuerza de interacción entre el robot y el entorno y K_p, K_i, y K_d las constantes proporcional, integral derivativa respectivamente.

Sin embargo, la aplicación de control de fuerza tiene algunos problemas específicos que pueden requerir modificaciones del controlador PID clásico. Así, en el movimiento restringido la dinámica del sistema depende de las características del entorno. Básicamente esto significa que los parámetros del controlador se deben sintonizar para cada aplicación. Sin embargo, algunas veces las características del entorno no se conocen de antemano. Por otro lado, se puede demostrar que el término integral puede llevar a la inestabilidad al sistema. Una solución puede ser el uso de una pre-alimentación de la fuerza en vez del integrador, como en la ecuación (25).

$$\tau = F_{ref} + K_p(F_{ref} - f) + K_d \frac{d}{dt}(F_{ref} - f) \qquad (25)$$

Otro problema que habitualmente parece en el control de la fuerza es el alto nivel de ruido que suelen tener los sensores de fuerza, lo que puede provocar problemas el cálculo de la derivada, y por lo tanto imposibilitar la acción de control derivativa. Por esta razón, con el fin de amortiguar el sistema es común el uso de la velocidad ($\dot{x}$) puesto que ésta es equivalente a la derivada de fuerza. La ley de control resultante se da en la ecuación (26).

$$\tau = F_{ref} + K_p(F_{ref} - f) - K_v\dot{x} \qquad (26)$$

Al igual que en el caso del control de posición, se han desarrollado varios esquemas de Matlab/Simulink para simular y analizar el control de fuerza. La Figura 6 muestra uno de ellos.

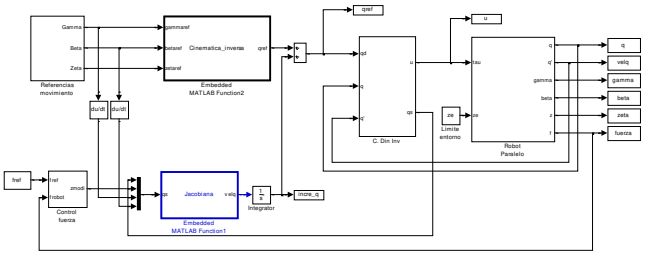


Figura 6: Esquema simulación control de fuerza

En la figura se pueden observar, además de los bloques descritos para el control de fuerza, dos bloques nuevos. En el primero (*Control fuerza*) se implementa el algoritmo de control de fuerza que se desee analizar. El bloque *Jacobina* es una función embebida que realiza la implementación de la matriz Jacobiana del robot paralelo.

Al bloque Jacobiana se le suministra a) las 9 coordenadas generalizadas del robot, b) la salida del controlador de fuerza y c) las velocidades de la orientación del robot. La salida de la Jacobiana se integra, sumándose posteriormente a la referencias de posición de las articulaciones activas del robot paralelo.

Para poder simular la fuerza que el robot sobre el entorno se ha modificado el bloque del robot. En este caso éste proporciona las posiciones y velocidades de las articulaciones activas, la orientación y la altura de la plataforma y la fuerza de interacción. Para el cálculo de dicha fuerza se ha implementado el siguiente modelo:

$$f = K_e \cdot (x_e - x) \qquad (27)$$

donde K_e y x_e son la matriz de rigidez y la posición del entorno. Por último x es la posición del robot en el espacio cartesiano.

Como se ha comentado anteriormente, debido a la configuración del robot paralelo, éste tiene 3 grados de libertad: la altura (z) de la plataforma y la orientación (γ y β). Por ello en el robot se ha establecido un control fuerza/posición: el control de fuerza se estable en el eje z, implementando un control de posición para controlar la orientación de la plataforma del robot.

La Figuras 7 y 8 muestran la respuesta simulada del control fuerza/posición desarrollado. En la Figura 7 se puede observar la fuerza de referencia y la fuerza ejercida por el robot. Se puede observar como la fuerza sigue a la referencia con un error nulo. La Figura 8 muestra las referencias de orientación y la orientación de la plataforma del robot. Se puede observar también como el seguimiento de dichas referencias es muy bueno presentando errores muy pequeños.

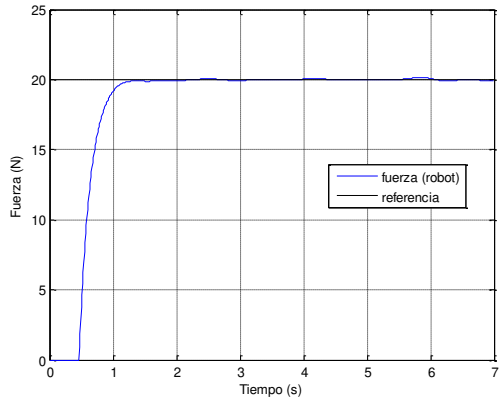


Figura 7: Fuerza ejercida por la plataforma

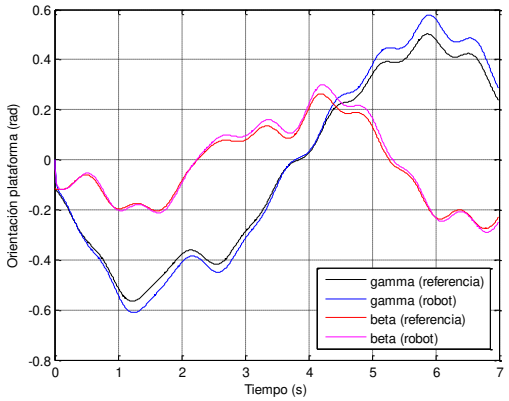


Figura 8: Orientación de la plataforma

4 DESARROLLO EN EL ROBOT REAL

Tal y como se ha comentado en el apartado 2, para implementar la arquitectura de control del robot paralelo se ha optado por un PC industrial. Éste se basa en un sistema industrial 4U Rackmount con 7 slots PCI y 7 ISA. Tiene un procesador IntelR© Core 2 Quad/Duo processor a 2,5GHz y una memoria de 4 Gb RAM. El sistema se complementa con las tarjetas de adquisición de datos necesarias para poder acceder a las señales de robot. En concreto se tiene una tarjeta Advantech PCI-1720 de convertidores Digital/Analógico para el suministro de las acciones de control a los 3 motores. Así mismo se utiliza otra tarjeta de Advantech: la PCL-833 que permite la lectura de las señales de encoders incrementales suministrados por los motores brushless.

Para poder establecer el control de fuerza el robot ha sido equipado con el sensor ATI Delta SI-330-30. Se trata de un sensor de 6 grados de libertad capaz de medir las fuerzas y los pares en los ejes X-Y-Z mediante un transductor monolítico instrumentado. El rango máximo de fuerzas son 330N para los ejes X e Y, y 990N para el eje Z. El rango máximo de pares es igual para los 3 ejes: 30N.m.

Para poder transmitir las señales desde sensor hasta el equipo de medida (el PC industrial en este caso) se tienen disponibles tres opciones: tarjeta de adquisición de datos, controlador F/T o sistema de comunicaciones Ethernet. En la arquitectura desarrollada se ha escogido esta última opción. El sistema NET F/T proporciona interfaces de comunicación Ethernet/IP y CAN bus. Al ser compatible con el estándar Ethernet este dispositivo se puede conectar fácilmente a cualquier red de área local, permitiendo una frecuencia de más de 7000Hz para medidas de las 6 componentes.

La Figura 9 muestra la arquitectura de control desarrollada para el robot paralelo.

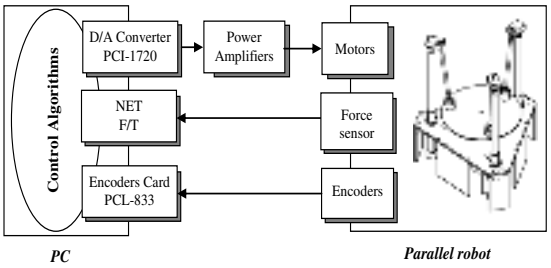


Figura 9: Arquitectura de control de la plataforma

Para establecer la programación del manipulador paralelo en este trabajo se ha utilizado el de alto nivel C++. El PC tiene instalado el sistema Linux Ubuntu, optándose por Xenomai para dotarlo de la funcionalidad de un sistema operativo de tiempo real.

Con este entorno se han desarrollado diversos algoritmos de control y aplicaciones del robot, utilizándose un periodo de muestreo de 0.01seg.

Las Figuras 10 y 11 muestran la referencia y la posición real y las acciones de control de la tercera articulación del robot paralelo. Se puede observar que el comportamiento suministrado por el controlador de dinámica inversa es muy bueno puesto que la posición sigue a la referencia con un error muy pequeño.

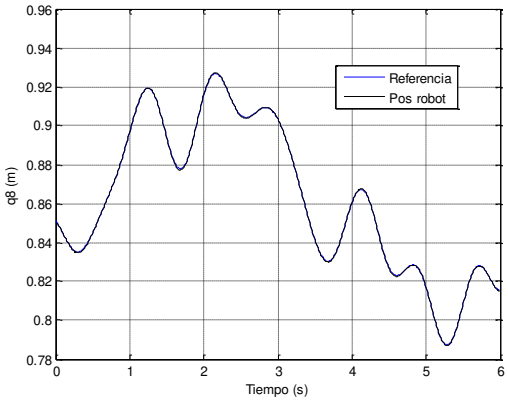


Figura 10: Posición 3 articulación del robot

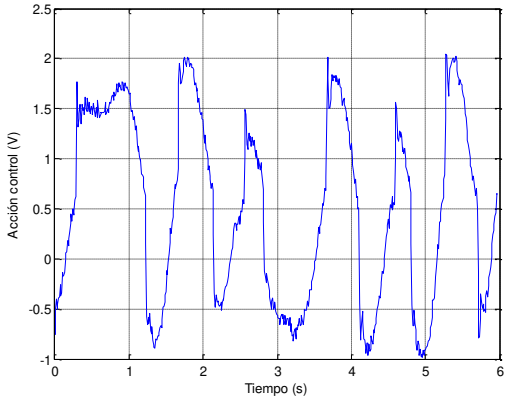


Figura 11: Acción control 3 articulación del robot

La Figura 12 muestra las referencias de velocidad de las articulaciones activas generadas por el control cinemático, y las salidas obtenidas con la Jacobiana del robot. Como se puede observar, su comportamiento es muy bueno y se corresponde completamente con los resultados obtenidos en la simulación del robot.

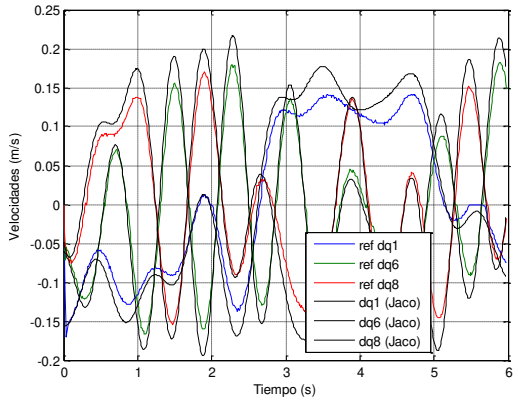


Figura 12: Velocidades articulaciones q_1, q_6 y q_8

Por último, la Figura 13 muestra la referencia y la fuerza ejercida en el eje Z por la robot paralelo. Durante los primeros 7 segundos (aproximadamente) la fuerza ejercida por el robot es 0 puesto que durante ese tiempo la plataforma está desplazándose hacia abajo y no tiene contacto con el entorno. A partir de ese instante de tiempo el elemento terminal choca con el entorno y se establece el control de fuerza, utilizándose para ello una referencia senoidal.

Como se puede apreciar en la ejecución, la fuerza real ejercida por el robot sigue de una forma muy precisa a la fuerza especificada como referencia.

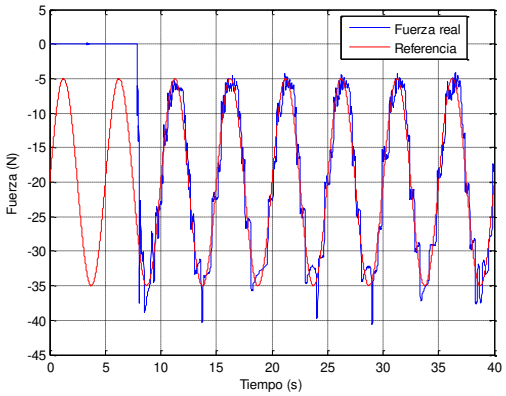


Figura 13: Fuerza Fz robot paralelo

CONCLUSIONES

En este trabajo se ha considerado el control de posición y fuerza de un robot paralelo de 3 grados de libertad. Para ello, además de desarrollar mecánicamente dicho robot, se le ha equipado con un sistema de control (basado en PC industrial) completamente abierto y flexible. El PC está equipado con tarjetas de convertidores D/A, lectura de encoders y un sensor de fuerza. Los programas y las aplicaciones de control se han desarrollado con el lenguaje de programación C++.

Para poder establecer el control del robot ha tenido que resolver e implementar el modelo cinemático directo e inverso del robot, el modelo dinámico y la Jacobiana.

Para el desarrollo del control de posición se ha optado con un controlador basado en el modelo. En concreto se ha establecido un control por par calculado (o dinámica inversa). Este controlador cancela los términos no lineales de la ecuación de movimiento del robot, en concreto los términos inerciales, centrífugos y de Coriolis y los términos gravitaciones.

La respuesta obtenida con el control de positición, tanto en las simulaciones como con el robot real han proporcionado un comportamiento muy bueno y unos índices de error de seguimiento muy pequeños.

Además, en el trabajo también se ha abordado el control de fuerza. Para ello se ha montado en la plataforma del robot un sensor industrial de 6 grados de libertad, de manera que la unidad de control puede medir las fuerzas y pares que el robot está ejerciendo en los ejes X-Y-Z.

Al igual que ocurre con el control de posición, el control de fuerza desarrollado tiene un comportamiento muy bueno, tanto en el modelo de simulación como en el robot real.

Agradecimientos

Los autores desean expresar su agradecimiento por la financiación parcial de este trabajo al Plan Nacional de I+D Comisión Interministerial de Ciencia y Tecnología (CICYT) bajo los proyectos de investigación DPI2010-20814-C02-(01, 02) and DPI2011-28507-C02-01, así como al proyecto CDCHT-ULA Grant I-1286-11-02-B.

Referencias

[1] Afonso, G., Pires, J. N. y Estrela, N., (2007). "Force control experiments for industrial applications: A test case using an industrial deburring example," Assem. Autom. J., vol. 26, no. 2, pp. 148–156.

[2] Astrom, K. J. y Murray, R., (2008). Feedback Systems. Princeton, NJ: Princeton Univ. Press.

[3] Canudas, C., Siciliano, B. y Basting, G., (1996). Theory of Robot Control. Ed. Springer.

[4] Chablat, D., y Wenger, P., (2003). Architecture optimization of a 3-DOF translational parallel mechanism for machining applications, the Orthoglide. IEEE Transactions on Robotics and Automation, vol. 19, nº 3, 403-410.

[5] Clavel, R., (1988). DELTA, a fast robot with parallel geometry. Proceedings of 18th International Symposium on Industrial Robot, Lausanne, April, 91-100.

[6] Craig, J., (1989). Introduction to robotics: Mechanics and Control. Ed. Addison-Wesley.

[7] Díaz-Rodriguez M., Mata, V., Valera A. y Page, A., (2010). A methodology for dynamic parameters identification of 3-DOF parallel robots in terms of relevant parameters. Mechanism and Machine Theory, Vol. 45, pp.1337-1356.

[8] Diaz-Rodriguez M., Valera, A., Mata, V. y Valles, M., (2012). Model-Based Control of a 3-DOF Parallel Robot Based on Identified Relevant Parameters. IEEE/ASME Transaction on Mechatronics. DOI: 10.1109/TMECH. 2012. 2212716.

[9] Farhat, N., (2006). Identificación de Parámetros Dinámicos en Sistemas de Cadena Cerrada. Aplicación a Robot Paralelos. Phd, Universidad Politécnica de Valencia.

[10] Gough, V.E. y Whitehall, S.G., (1962). Universal tire test machine. Proceedings of 9th International Technical Congress FISITA, 117-135.

[11] Jalón, J.G. y Bayo, E., (1994). Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time challenge. Springer-Verlag, New-York.

[12] Lee, K.M. y Arjunan, S., (1991). A three-degrees-of-freedom micromotion in-parallel actuated manipulator. IEEE Transactions on Robotics and Automation, vol. 7 nº 5, 634–641.

[13] Li, Y. y Xu, Q., (2004). Kinematics and stiffness analysis for a general 3-PRS spatial parallel mechanism. Proceeding of 15th CISM/IFToMM Symposium in Robot Design, Dynam-ics and Control, Montreal, Canada.

[14] Li, Y. y Xu, Q. (2007). Design and Development of a Medical Parallel Robot for Cardiopulmo-nary Resuscitation. IEEE/ASME Transaction on Mechatronics, vol. 12, nº 3, 265–27.

[15] Merlet, J.-P., (2000). Parallel Robots. Kluwer, London, U.K.

[16] Pierrot, F., Nabat, V., Company, O., Krut, S. y Poignet, P., (2009). Optimal design of a 4-dof paral-lel manipulator: From academia to industry, IEEE Transactions on Robotics, vol. 25, nº 2,213-224.

[17] Rosillo, N., Valera, A., Benimeli, F., Mata, V. y Valero, F., (2011). Real-time solving of Dynamic Problem in Industrial Robots, Industrial Robot, vol. 38, n. 2, pp. 119-129.

[18] Spong, M. y Vidyasagar, M., (1989). Robot dynamics and control. Ed. John Wiley & Sons.

[19] Steward, D. A. (1965). A platform with 6 degree of freedom. Proceedings of the Institution of mechanical engineers. Part 1 vol. 15, 371-386.

[20] Syrseloudis, C. E. y Emiris, I. Z., (2008). A parallel robot for ankle rehabilitation-evaluation and its design specifications. Proceeding of 8th IEEE International Conference onBio-Informatics and BioEngineering, 2008. Athens, 1-6, October.

[21] Tsai, L. W., (1999). Robot Analysis: The Mechanics of Serial and Parallel Manipulator. Wiley Interscience, Canada.

[22] Volpe, R. y Khosla, P., (1993). "A theoretical and experimental investigation of explicit force control strategies for robot manipulators" IEEE Trans. Automatic Control, vol. 38, no. 11, pp. 1634–1650, Nov.

[23] Yoshikawa, T., (1990). Foundations of Robotics: Analysis and control. Ed. The MIT Press.

Fusión sensorial de visión, fuerza y aceleración para el control de robots manipuladores con movimientos restringidos.

A. Sánchez García, J. Gámez García, J. Gómez Ortega
Grupo de Robótica, Automática y Visión por Computador
Universidad de Jaén
{asgarcia, jggarcia, juango}@ujaen.es

Resumen

Cuando un brazo robótico tiene que interactuar con entornos desconocidos, la eficacia y precisión al integrar todas las medidas sensoriales, así como la información proporcionada, contribuyen en el éxito de la tarea desempeñada. De hecho, en tales operaciones, los sensores juegan un papel crucial, ayudando a superar muchas de las dificultades propias de los entornos desconocidos. En este artículo, se presenta un nuevo enfoque de fusión sensorial para brazos robóticos con movimiento restringido. Está basado en la integración de tres tipos diferentes de sensores: fuerza, aceleración y visión artificial, lo cual tiene el objetivo de mejorar el desarrollo de la tarea en términos de precisión y velocidad. El enfoque propuesto de fusión sensorial ofrece una mejor estimación de la superficie de contacto, eliminando errores provocados por la calibración de la cámara o perturbaciones en las medidas tales como las fuerzas de inercia. El observador propuesto ha sido verificado experimentalmente en un brazo robótico Stäubli, mostrando cómo éste puede mejorar la respuesta del controlador de la tarea.

Palabras clave: Fusión sensorial, Fuerza, Aceleración, Visión por computador

1. Introducción

En los últimos años, la fusión multisensorial ha recibido una significativa atención en diferentes campos de la robótica. Uno es la interacción de brazos robóticos en entornos desconocidos. Obviamente, la eficacia y la precisión al integrar todas las medidas, así como la información proporcionada, contribuyen al éxito de las tareas desempeñadas.

En estas situaciones en las que un brazo robótico debe entrar en contacto con un entorno desconocido, se pueden presentar dificultades si éste controla la tarea solamente con visión. Principalmente esto es debido a que es muy difícil reconocer las relaciones entre objetos donde el entorno no está perfectamente definido [11]. Para solucionar parte de este problema, diversos autores han usado otro tipo de sensores, como por ejemplo sensores fuerza/par; sin embargo, el usar esta estrategia también presenta una desventaja importante, y es que las medidas de fuerza se ven influenciadas por perturbaciones en dichas medidas, tales como las fuerzas de inercia y de la gravedad [3], afectando también al estimador de posición de contacto que fusiona la información de ambos sensores.

Se han aplicado varios enfoques para estimar la superficie de contacto localmente usando mediciones de sensores de visión y fuerza. En [15], se propuso una estrategia de control basada en multisensores que permitía al elemento final de un robot manipulador realizar una clase de movimientos restringidos. Concretamente, la herramienta del robot describía una curva en una superficie desconocida usando la información de los sensores mencionados anteriormente junto a las mediciones de las articulaciones del robot. En [13], se desarrolló un método para combinar control de fuerza directo y control visual en presencia de superficies planas desconocidas. El enfoque propuesto usaba un lazo de control de retroalimentación de fuerza y una referencia de trayectoria basada en visión como señal de prealimentación.

Lippliello et al. [9], presentaron un enfoque complementario basado en la integración de visión y fuerza para mejorar la interacción de un manipulador con un entorno parcialmente conocido. Éstos usaron un Filtro de Kalman Extendido que integraba información visual, las medidas de fuerza y las medidas de posición del robot, con el objetivo de estimar la posición del objeto. Finalmente, en [2], fue propuesto un control visual para controlar del movimiento y la fuerza, con incertidumbre en la superficie de contacto, la cinemática, la dinámica y modelo de cámara. Siendo esta propuesta validada únicamente través de simulación.

Para el caso de integración de fuerza y aceleración, inicialmente se presentó en 2004 un primer trabajo de investigación [4] con el objetivo de desarrollar un observador de fuerza de contacto de 1 grado de libertad que usaba información recogida de un acelerómetro y de un sensor de fuerza/par situado en el extremo del robot. Posteriormente, en 2007, Kroger et al. [7] utilizaron un acelerómetro con seis grados de libertad combinándolo con un sensor fuerza y par de seis grados de libertad. Éste calculaba las fuerzas y pares provocadas por el contacto con el entorno. En 2008, Gamez et al. [5] propusieron un observador que fusionaba informa-

ción de tres sensores diferentes: la posición del robot robot, un sensor de fuerza/par y un sensor de aceleración. Éstos demostraron las ventajas de este enfoque de fusión sensorial puesto que las fuerzas inerciales fueron eliminadas correctamente de las mediciones.

En este artículo, se propone una estrategia de fusión sensorial unificando las ventajas de anteriores investigaciones. Esta propuesta consiste en la integración sensorial de fuerza, visión y aceleración para así obtener un estimador de la superficie de contacto en un entorno adecuado que podría mejorar la interacción al desempeñar la tarea. El observador propuesto tiene diferentes ventajas como, por ejemplo, la considerable reducción de los errores provocados por la calibración de la posición de la cámara, y que la estimación de posición de contacto no esté sujeta a perturbaciones como son las fuerzas inerciales.

El artículo se organiza de la siguiente manera. Primero, en la Sección II, se presenta la propuesta. La Sección III introduce la plataforma de experimentación utilizada para examinar la nueva metodología; mientras que la Sección IV describe las conclusiones y plantea una breve discusión.

2. Fusión de fuerza, aceleración y visión

El tema de interés es cómo mejorar, simultáneamente, el control de la fuerza de contacto y la posición y orientación de un robot manipulador en una superficie desconocida 3-D (Figura 1). Para tal propósito, se propone el uso de una nueva estrategia de fusión sensorial que integre la información de tres sensores diferentes: visión, fuerza y aceleración. El observador resultante, denominado observador de la superficie de contacto, presenta las siguientes ventajas:

- Se obtiene una estimación precisa de la posición y orientación de la superficie de contacto, eliminando errores de calibración debidos al proceso de calibración y posicionamiento de la posición de la cámara, que no pueden ser totalmente eliminados [14, 10].

- El observador no se ve afectado por fuerzas de gravedad e inercia, las cuales se miden, como las perturbaciones, por el sensor fuerza y par. Esto implica lo siguiente:

  - No existe ningún falso contacto que pueda afectar la estimación de la posición de la superficie de contacto.
  - La orientación de la superficie no esta influenciada por las perturbaciones producidas por las fuerzas de inercia.

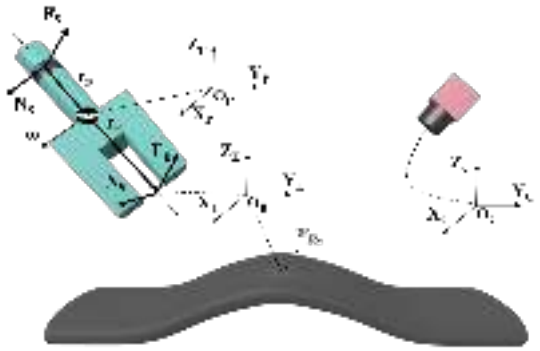


Figura 1: Sistemas de coordenadas. El formalismo de marcos de tarea [1] se usó para el modelado y los experimentos.

2.1. Descripción de los sistemas de coordenadas y movimiento

Tal y como se muestra en la figura 1, se asume la presencia de un sensor fuerza/par rígidamente unido al extremo del robot y un acelerómetro localizado sobre la herramienta de éste. También una cámara en configuración eye-to-hand.

Definamos entonces $\{O_S X_S Y_S Z_S\}$ y $\{O_I X_I Y_I Z_I\}$ como los sistemas de referencia de los sensores de fuerza e inercia respectivamente. $\{O_E X_E Y_E Z_E\}$ es el sistema de coordenadas del elemento final por el que los componentes de la fuerza externa y momentos están representados; $\{O_P X_P Y_P Z_P\}$ representa el sistema de coordenadas fijado en el elemento final de tal forma que $\{O_P\}$ coincide con el centro de gravedad de la herramienta, y los ejes $\{X_P\}$, $\{Y_P\}$ y $\{Z_P\}$ con los principales ejes del elemento final. $\{O_C X_C Y_C Z_C\}$ es el sistema de referencia de la cámara; y el sistema de referencia global es un sistema inercial de coordenadas que viene dado por $\{O_W X_W Y_W Z_W\}$, el cual lo hacemos coincidir con el sistema de referencia del robot para una notación más simplificada. Finalmente, $\{{}^A R_B\}$ es la matriz de rotación que relaciona el sistema de referencia B y A.

2.2. Modelado y definición del Observador de Superficie de Contacto

De una relación de equilibrio estático en $\{O_P\}$, usando una notación vectorial [12], las fuerzas y momentos vienen dados por:

$$\begin{aligned} {}^P U_P &= \begin{pmatrix} {}^P F_P \\ {}^P N_P \end{pmatrix} = \begin{pmatrix} {}^P R_S^T & 0_{3\times 3} \\ -{}^P R_S^T {}^S p^x & {}^P R_S^T \end{pmatrix} \begin{pmatrix} {}^S F_S \\ {}^S N_S \end{pmatrix} \\ &+ \begin{pmatrix} {}^P R_E^T & 0_{3\times 3} \\ -{}^P R_E^T {}^E p^x & {}^P R_E^T \end{pmatrix} \begin{pmatrix} {}^E F_E \\ {}^E N_E \end{pmatrix} \\ &= {}^P T_S \begin{pmatrix} {}^S F_S \\ {}^S N_S \end{pmatrix} + {}^P T_E \begin{pmatrix} {}^E F_E \\ {}^E N_E \end{pmatrix} \end{aligned} \quad (1)$$

con ${}^P R_S$, ${}^P R_E$ siendo las matrices de rotación que relacionan los sistemas de referencia $\{S\}$ y $\{E\}$ con el sis-

tema $\{P\}$; ${}^{S}p^{x}$ y ${}^{E}p^{x}$ son matrices de dimensión 3×3 donde p es el vector de posición desde $\{O_P\}$ hasta $\{O_S\}$ o desde $\{O_P\}$ hasta $\{O_E\}$. Finalmente, aplicando las ecuaciones Newton-Euler, la dinámica de la herramienta del robot se define como:

$$\begin{aligned} {}^{P}U_P &= \begin{pmatrix} m\,{}^{W}R_P^{-1}\,{}^{W}R_I\,a - m\cdot{}^{W}R_P^{-1}g \\ \mathscr{I}\cdot{}^{P}R_I\dot{\omega} + {}^{P}R_I\cdot\omega\times\mathscr{I}\,{}^{P}R_I\cdot\omega \end{pmatrix} \\ &= \begin{pmatrix} m\,{}^{W}R_P^{-1}\,{}^{W}R_I & 0_{3x3}0_{3\times 3} & \mathscr{I}\,{}^{P}R_I \end{pmatrix}\begin{pmatrix} a \\ \dot{\omega} \end{pmatrix} \\ &+ \begin{pmatrix} -m\,{}^{W}R_P^{-1} & 0_{3\times 3} \\ 0_{3\times 3} & ({}^{P}R_I\omega)^{\times}\mathscr{I}\,{}^{P}R_I \end{pmatrix}\begin{pmatrix} g \\ \omega \end{pmatrix} \quad (2) \end{aligned}$$

donde m es la masa de la herramienta del robot, $a=[\ddot{x},\ddot{y},\ddot{z}]^T$ es la aceleración de la herramienta medida en el sistema de referencia inercial $\{I\}$; $g=[g_x,g_y,g_z]^T$ es el vector de aceleración gravitacional; $\omega=[\omega_1,\omega_2,\omega_3]^T$ es la velocidad angular de la herramienta respecto al sistema de referencia $\{I\}$. La matriz denota el momento de inercia de la herramienta calculada respecto al sistema de referencia $\{P\}$.

Para definir la dinámica de la herramienta de las Eqs.(1-2) en formulación de espacio de estado, consideremos el siguiente vector de espacio de estado:

$$X=[x,y,z,\theta,\phi,\psi,\dot{x},\dot{y},\dot{z},\omega_1,\omega_2,\omega_3] \quad (3)$$

donde x,y,z son coordenadas de posición del centro de masa de la herramienta; y θ,ϕ y ψ son los ángulos de Euler que definen la orientación de dicha herramienta. Ambos se refieren al sistema de referencia $O_WX_WY_WZ_W$. $\dot{x},\dot{y},\dot{z}$ corresponden a la velocidad lineal del centro de masa de la herramienta.

Las variables que pueden ser medidas son consideradas como salidas del observador: la posición y orientación de la herramienta (p,o), las señales del sensor de fuerza/par (F_s,N_s), las velocidades angulares (ω) y la aceleración lineal de la herramienta $(\ddot{p})$. (ω) y $(\ddot{p})$ se miden por medio del sensor inercial. Además, las aceleraciones angulares $\dot{\omega}$ de la herramienta se estiman usando las velocidades angulares, incluyéndolas también como salidas. Así, el vector de salida Y será:

$$Y=\left(p^T,o^T,F_s^T,N_s^T,\omega^T,\ddot{p}^T,\dot{\omega}^T\right)^T \quad (4)$$

Por lo tanto, se propone el siguiente sistema de espacio de estados

$$\dot{X}=A(X)X+B_SU_S+B_EU_E+B_Gg+\nu_X \quad (5)$$
$$Y=C(X)X+D_SU_S+D_EU_E+D_Gg+\nu_y \quad (6)$$

Donde g es el vector de gravedad, $U_S=[{}^{S}F_S^T,{}^{S}N_S^T]^T$, y $U_E=[{}^{E}F_E^T,{}^{E}N_E^T]^T$. ν_X y ν_y son, respectivamente, el proceso y los ruidos de salida. Las matrices $A(X)$, B_S, B_E, B_G, $C(X)$, D_S, D_E y D_G obtienen sus valores de las Eqs. (1) y (2) (más información puede recabarse en [5]). Por tanto, proponemos un observador donde la entrada U_E no ha sido considerada

$$\dot{\widehat{X}} = A(X)\widehat{X}+B_SU_S+B_Gg+L(t)(Y-\hat{Y}) \quad (7)$$
$$\widehat{Y} = C(X)\widehat{X}+D_SU_S+D_Gg \quad (8)$$

donde $\widehat{X}$ corresponde a la estimación de estado de X, y $L(t)$ es la ganancia del observador para un instante dado t. Finalmente, el estimador de fuerza dinámica se define como [5]

$$\widehat{U}_E=D_E^{\dagger}(-C(X)\widetilde{X}+\widetilde{Y}) \quad (9)$$

En este contexto, considérese que parte del movimiento del robot (movimiento restringido) debe darse en el plano π_E en el espacio cartesiano definido por la ecuación implícita

$${}^{W}\hat{F}_E\pi_E+d_E=0 \quad (10)$$

donde

$$d_E=-{}^{W}\hat{F}_{E_x}\cdot{}^{W}p_{r_x}-{}^{W}\hat{F}_{E_y}\cdot{}^{W}p_{r_y}-{}^{W}\hat{F}_{E_z}\cdot{}^{W}p_{r_z} \quad (11)$$

siendo ${}^{W}p_r$ la posición del efector final medida en O_W. Nótese que en tal situación

$${}^{W}\hat{F}^{T\,W}\dot{p}_r=0;\ {}^{W}\hat{F}_{E_z}\neq 0 \quad (12)$$

Definamos entonces v como el vector que indica la trayectoria del movimiento determinado por

$$v={}^{W}p_r-{}^{W}p_c \quad (13)$$

donde ${}^{W}p_c=[{}^{W}x_c,{}^{W}y_c,{}^{W}z_c]^T$ es el punto deseado de contacto con el entorno, el cual es estimado por medio del sensor de visión a través de

$$\begin{pmatrix} {}^{W}x_c \\ {}^{W}y_c \\ {}^{W}z_c \end{pmatrix}={}^{c}R_w\begin{pmatrix} {}^{C}x_c \\ {}^{C}y_c \\ {}^{C}z_c \end{pmatrix}+{}^{W}t_c \quad (14)$$

siendo $({}^{W}x_c,{}^{W}y_c,{}^{W}z_c)$ y $({}^{C}x_c,{}^{C}y_c,{}^{C}z_c)$ las coordenadas del punto p_c en los sistema de referencia O_W y O_C respectivamente. ${}^{C}p_c$ está determinado aplicando un modelo pin-hole con una focal f, resultando

$$X=f\frac{{}^{C}x_c}{{}^{C}z_c} \qquad Y=f\frac{{}^{C}y_c}{{}^{C}z_c} \quad (15)$$

con X e Y como la posición del objeto en píxeles de cámara. Usando ahora el vector v, un nuevo plano π_1 se define como

$$({}^{W}\hat{F}_E\times v)\pi_1+d_1 \quad (16)$$

con

$$d_1=-\pi_{1_x}\cdot{}^{W}p_{r_x}-\pi_{1_y}\cdot{}^{W}p_{r_y}-\pi_{1_z}\cdot{}^{W}p_{r_z} \quad (17)$$

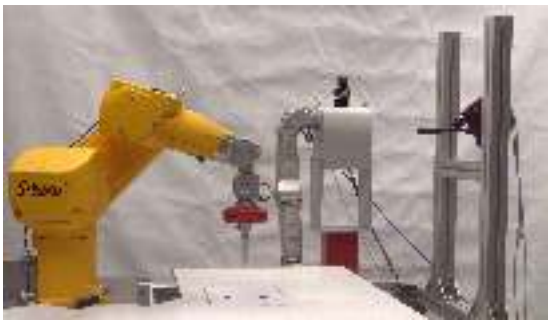


Figura 2: El sistema experimental. Un robot industrial Stäubli con un sistema de arquitectura de control abierto. El control de impedancia se realiza perpendicular a la pantalla. Mientras que el acelerómetro se sitúa sobre la herramienta, el sensor fuerza/par se localiza en la extremidad del robot. Obsérvese la cámara situada en una configuración eye-to-hand.

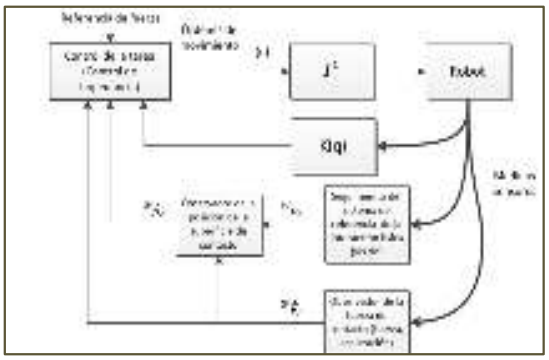


Figura 3: Lazo de control implementado para examinar el observador de la superficie de contacto. La referencia de posición se obtuvo por medio de este observador. El objetivo es aplicar una fuerza de referencia al entorno (cambiante) mientras se mantiene el contacto.

Definamos ahora un tercer plano π_2 como

$$n_2 \cdot \pi_2 = 0 \tag{18}$$

donde $n_2 = \pi_E \times \pi_1$. Observando las Eqs. (10), (16) y (18), se puede apreciar que ${}^W\hat{F}_E$, n_1 y n_2 son linealmente independientes, de tal forma que sólo pueden cruzarse en un punto, que se obtiene por

$${}^W\hat{p}_c = \frac{-d_E \cdot (n_1 \times n_2) - d_1 \cdot (n_2 \times {}^W\hat{F}_E)}{({}^W\hat{F}_E) \cdot (n_1 \times n_2)} \tag{19}$$

Así, ${}^W\hat{p}_c$ es el observador de la superficie de contacto que mejora la estimación de ésta usando solamente visión artificial o visión artificial y sensores de fuerza.

3. Plataforma de experimentación

En la figura 2 se muestra la plataforma robótica utilizada en la validación del observador. Esta se compone de los siguientes elementos: un robot industrial Stäubli RX60 de 6 grados de libertad, un sistema de visión compuesto por una cámara GX1050, un sensor de fuerza/par ATI de 6 grados de libertad y un acelerómetro lineal 3D de tipo capacitivo con propiedades DC. Se ha colocado una herramienta pesada, de 2 kg, para aumenter los efector producidos por las perturbaciones provocadas por las fuerzas inerciales. Cada articulación del robot manipulador es controlado por un PID con un ancho de banda de 250 Hz. Este sistema robótico ha sido desarrollado con una propuesta de arquitectura de software abierta basada en componentes software [6].

Se han llevado a cabo diferentes experimentos para probar el funcionamiento de la fusión sensorial propuesta. La idea es controlar el brazo robótico interactuando con un entorno desconocido y que el manipulador pueda moverse tanto en espacio libre como restringido. También se han llevado a cabo experimentos con dos entornos distintos: uno estático, fabricado con una pantalla metálica con un patrón visual; y el segundo con la misma pantalla pero sobre una superficie móvil en la que su posición y orientación varían.

En el caso del controlador, se escogió un enfoque de control de impedancia para verificar las propiedades del nuevo observador de superficie de contacto. En este sentido, se usó un controlador LQR (Regulador Lineal Cuadrático) para hacer que el valor de la relación de impedancia fuera cero [8]. La ley de control aplicada fue

$$u = -L \cdot X + c \cdot {}^W\hat{F}_E + l_r \cdot {}^W\hat{p}_c \tag{20}$$

con c siendo la ganancia de la fuerza en el control de impedancia, ${}^W\hat{F}_E$ la estimación de la fuerza del entorno, que en nuestro caso se estimó usando la ecuación 9, ${}^W\hat{p}_c$ la estimación de posición hallada a través del observador de superficie de contacto propuesto, y l_r la constante de ganancia de la posición, siendo L calculada como en [5].

3.1. Resultados

La figura 4 muestra un mismo experimento, pero repetido en cuatro ocasiones con diferentes condiciones. En todos los casos, inicialmente el robot se movió en espacio libre entrando en contacto con el entorno, intentando mantener una referencia de fuerza de 30N.

A continuación se presenta una explicación de cada uno de los casos. En todos ellos, la superficie era estática.

- **Línea azul (FS-VS)**: El controlador de impedancia usó directamente la medida del sensor de fuerza (FS) y sólo el sensor de visión (VS) para estimar la posición de contacto.

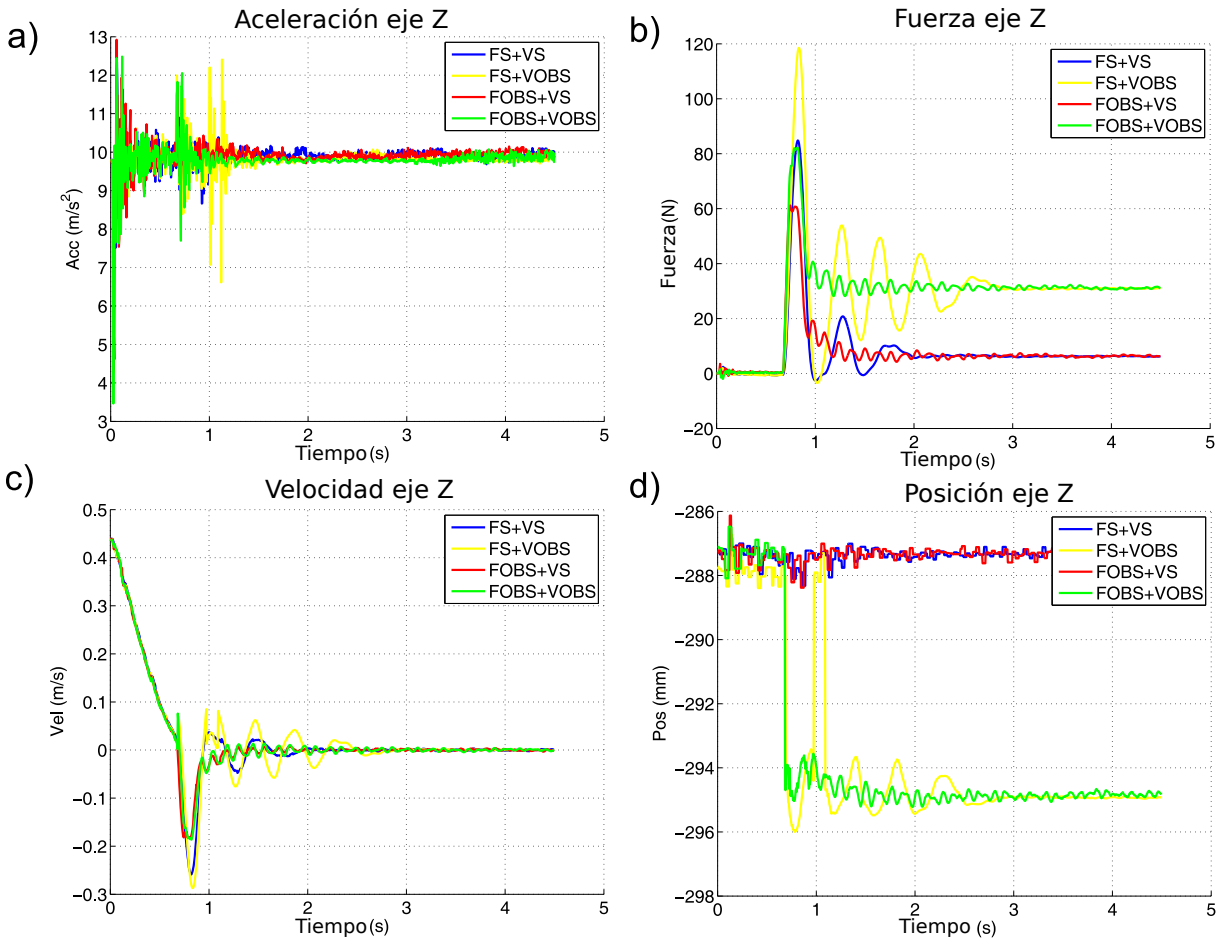


Figura 4: Resultados del mismo experimento pero con cuatro combinaciones de entradas a la ley de control. Las posibles entradas son: sensor de fuerza (FS), observador de fuerza (FOBS), estimación del sensor de visión (VS), y observador de la superficie de contacto (VOSB). a) describe la aceleración medida durante. b) la medición del sensor de fuerza y la salida del observador de fuerza. c) la velocidad ordenada en todos los casos. d) la estimación de la superficie de contacto.

- **Línea amarilla (FS-VOSB)**: El controlador de impedancia usó directamente la medida del sensor de fuerza (FS) y el observador de la superficie de contacto (VOBS) pero como entrada, este observador usó la medida de fuerza sin compensación inercial.
- **Línea roja (FOBS-VS)**: El controlador de impedancia usó la medida del observador de fuerza de contacto (FOBS) propuesto en Eq. (8) como entrada de fuerza, y solamente el sensor de visión (VS) para estimar la posición de contacto.
- **Línea verde (FOBS-VOBS)**: El controlador de impedancia usó el observador de la fuerza de contacto (FOBS) y el observador de la superficie de contacto (VOBS) el cual, como entrada de fuerza, usó el FOBS.

Por tanto, en la figura 4.a, la cual muestra las medidas de aceleración, se puede apreciar que estas medidas son básicamente las mismas en todo el experimento. Hay que tener en cuenta que la aceleración se mide a lo largo del eje z y que éste mide también la aceleración de la gravedad. La figura 4.b muestra las cuatro fuerzas, medidas y estimadas, para los cuatro experimentos. Comparándolos, se puede ver que: (1) la oscilación de la fuerza inicial se reduce cuando se utiliza el observador de fuerza. Esto es así ya que, debido a la deceleración, las fuerzas inerciales son eliminadas por el observador de fuerza. Nótese también cómo el observador reduce el ruido generado por los sensores; (2) la fuerza de contacto de referencia de 30N es alcanzada solamente cuando se usa el observador de la superficie de contacto (VOSB). La razón es que, cuando el controlador de impedancia usa la posición calculada por el sensor de visión, huy un error de 6 mm entre la posición real y la estimada por medio del sensor de visión.

La figura 4.c representa la velocidad ordenada, para la misma dirección, en los cuatro casos. Véase una oscilación mayor cuando el sensor de fuerza se usa directamente, y también que la velocidad ordenada es más baja cuando se usa el observador propuesto. Finalmente, la figura 4.d. muestra la estimación de posición que es obtenida para los cuatro experimentos. Hay que fijar la atención en cómo, estando en un estado estacionario, el observador de la superficie de contacto obtiene una mejor estimación cuando también se utiliza el observador de fuerza; de hecho, hay un error de 6 mm comparado con la opción del sensor de visión. En estado transitorio, aparecieron otras ventajas como, por ejemplo, que el observador de superficie no se ve afectado por los efectos inerciales al usar el observador de fuerza (la información mostrada en la línea amarilla refleja estas perturbaciones ocasionadas por errores en la estimación de la posición).

Por otro lado, consideremos un segundo experimento, el cual se representa en la figura 5. Como se muestra, la idea es desplazar el objeto donde el robot tomará contacto, y cambiará su orientación para analizar cómo estas modificaciones afectaban al observador de superficie propuesto. En cualquier caso, el robot se mueve de un espacio libre a un espacio restringido.

En la figura 5.a, se pueden apreciar las aceleraciones más importantes durante la etapa de transición. La figura 5.b. describe la medida del sensor de fuerza y el observador de fuerza obtenidos durante el experimento. La referencia de fuerza fue de 30N, y ligeramente variada debido a que el objeto estaba siendo sostenido con la mano. Téngase en cuenta cómo, aparte de eliminar los efectos inerciales durante los movimientos de aceleración y desaceleración, el observador de fuerza también reduce el ruido del sensor de fuerza. Este efecto se puede apreciar mejor en las figuras 5.c y 5.d, donde se muestra la densidad del espectral tanto para el sensor de fuerza como para el observador de fuerza. Analizando estos dos espectros, se puede apreciar cómo el observador de fuerza reduce, en su mayor parte, la ganancia a altas frecuencias, mientras que mantiene la respuesta a bajas frecuencias. La figura 5.e expone la estimación de posición obtenida usando solo el sensor de visión y el observador de la superficie de contacto. Dicho gráfico es bastante representativo puesto que muestra cómo el error es mayor cuando la superficie se mueve hacia abajo y, sin embargo, inicialmente el error en la estimación de la posición es cero. Esto es debido a que la posición de la cámara se calibró cerca de la posición inicial de este experimento.

4. Conclusiones

Este artículo ha presentado una nueva propuesta de fusión sensorial que estima la superficie de contacto usando tres sensores diferentes: visión, fuerza y aceleración. El observador resultante ofrece dos ventajas: la primera es la mejora de la etapa de transición en la cual las tareas del robot conllevan un contacto entre la herramienta del robot y un entorno desconocido. La segunda es una mejor actuación del movimiento restringido ya que la posición del entorno con una precisión mayor.

Para la validación, el comportamiento del observador de la superficie de contacto, y por tanto, del nuevo enfoque de fusión sensorial, fue implementado con éxito en un robot industrial Stäubli que interactuó, con un entorno estático, y también uno dinámico.

Agradecimientos

Parte de esta investigación fue apoyada a través de los proyectos DPI2011-27284 y TEP2009-5363.

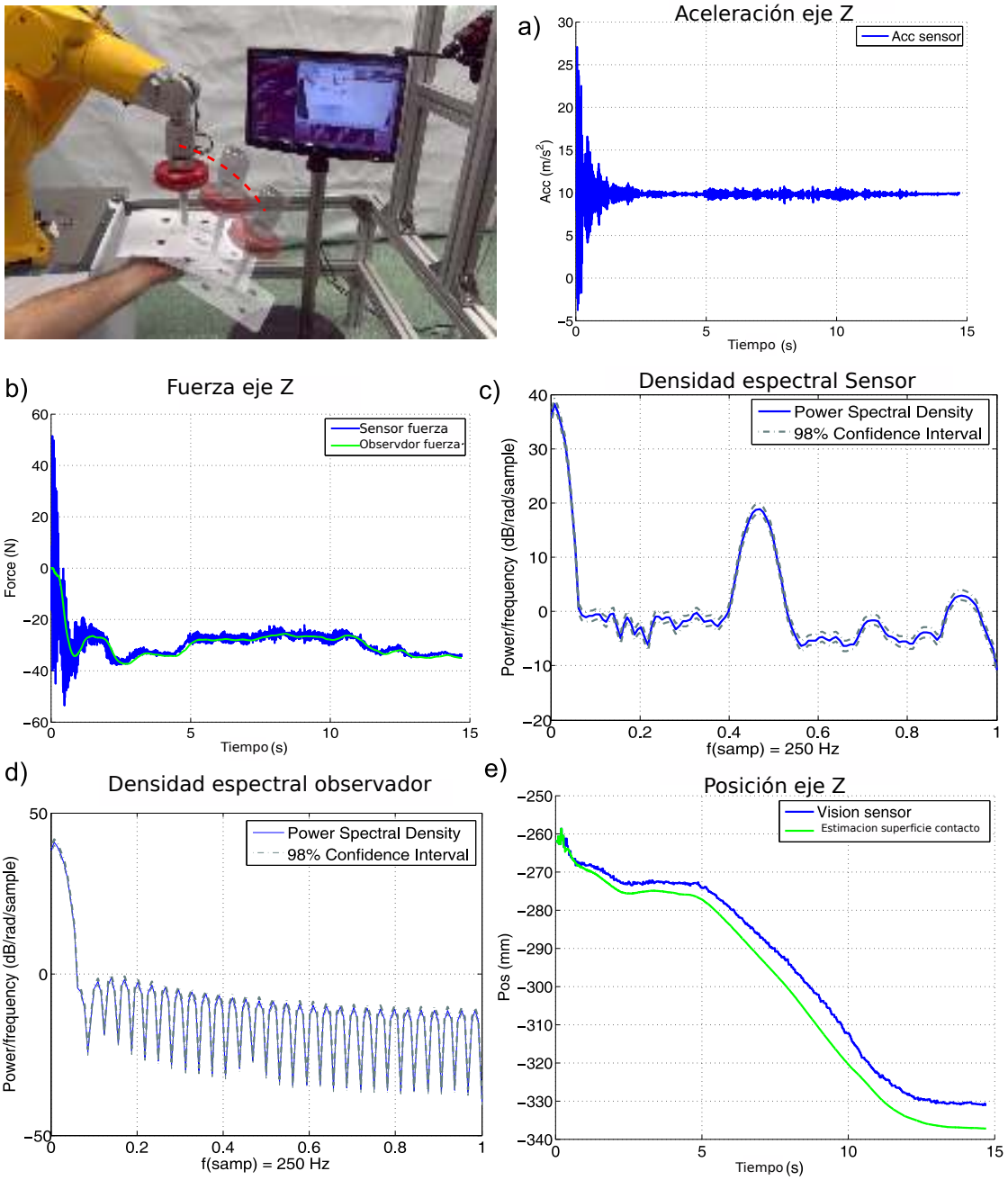


Figura 5: Control de movimiento restringido con una superficie en movimiento, a) describe las aceleraciones medidas durante el experimento. b) describe la medida del sensor de fuerza y el observador de fuerza obtenido durante el experimento. c) Espectro de frecuencias para la medida del sensor de fuerza. d) Espectro de frecuencias para el observador de la fuerza de contacto e) Posición estimada por el sistema de visión y por el observador de la superficie de contacto.

Referencias

[1] H. Bruyninckx and J. de Schutter, "Specification of force-controlled actions in the task frame formalism- a synthesis", IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 581 - 589, 1996.

[2] C. C. Cheah, S. Hou, Y. Zhao, and J.-J. Slotine, "Adaptive vision and force tracking control for robots with constraint uncertainty", Mechatronics, IEEE/ASME Transactions on, vol. 15, no. 3, pp. 389-399, june 2010.

[3] J. Gamez, A. Robertsson, J. G. Ortega, and R. Johansson, "Force and acceleration sensor fusion for compliant robot motion control", in IEEE Int. Conf. on Robotics and Automation (ICRA2005), Barcelona, Spain, 2005, pp. 2720-2725.

[4] J. Gamez, A. Robertsson, J. Gomez, and R. Johansson, "Sensor fusion of force and acceleration for robot force control", in Int. Conf. Intelligent Robots and Systems (IROS 2004), Sendai, Japan, 2004, pp. 3009-3014.

[5] J. Gamez, A. Robertsson, J. Gomez Ortega, and R. Johansson, "Sensor fusion for compliant robot motion control", IEEE Trans. on Robotics, vol. 24(2), pp. 430-441, 2008.

[6] J. Gamez, J. Gomez Ortega, A. Sanchez Garcia, and S. Satorres Martinez, "Robotic software architecture for multisensor fusion system", IEEE Trans. on Industrial Electronics, vol. 56(3), pp. 766-777, 2009.

[7] T. Kroger, D. Kubus, and F. Wahl, "Force and acceleration sensor fusion for compliant manipulation control in 6 degrees of freedom", Advanced Robotics, vol. 21, pp. 1603-1616(14), October 2007.

[8] R. Johansson and A. Robertsson, "Robotic force control using observer-based strict positive real impedance control", in IEEE Proc. Int. Conf. Robotics and Automation, Taipei, Taiwan, September 2003, pp. 3686-3691.

[9] V. Lippiello, B. Siciliano, and L. Villani, "A position-based visual impedance control for robot manipulators", in 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 2068-2073.

[10] S. J. Maybank and O. D. Faugeras, "A theory of self-calibration of a moving camera", International Journal of Computer Vision, vol. 8, no. 2, pp. 123-151, 1992.

[11] Y. Motai and A. Kosaka, "Eye calibration applied to viewpoint selection for robotic vision", Industrial Electronics, IEEE Transactions on, vol. 55, no. 10, pp. 3731 -3741, oct. 2008.

[12] L. Murray and Sastry, A Mathematical Introduction to Robotic Manipulation. CRC Press, 1994.

[13] T. Olsson, J. Bengtsson, R. Johansson, and H. Malm, "Force control and visual servoing using planar surface identification", in Proc. IEEE Int. Conf. Robotics and Automation. ICRA '02, vol. 4, 2002, pp. 4211-4216.

[14] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration", IEEE Trans. Robotics and Automation, vol. 5(3), pp. 345-358, June 1989.

[15] D. Xiao, M. Song, B. Ghosh, N. Xi, and T. J. Tarn, "Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment", IEEE Trans. Control Systems Technology, vol. 8(4), pp.635-645, Jul 2000.

Mejora de las Capacidades de un Robot Mini-Humanoide en la Tarea de Subir y Bajar Escaleras

Wilmer Cardenas Pilatuña (wcardeba@pa.uc3m.es)
Félix Rodríguez Cañadillas (ferodrig@ing.uc3m.es)
Alberto Jardón Huete (ajardon@ing.uc3m.es)

Resumen

En este artículo se presenta el diseño hardware y software de un sistema para el control de un robot mini-humanoide en la tarea de subir y bajar escaleras. Estos robots son utilizados en varios centros de investigación de todo el mundo, y también son ampliamente utilizados en competiciones de robots, como por ejemplo, el concurso español CEABOT, en el cual esta basado este artículo. Este campeonato abarca varias pruebas como la que se estudia en este proyecto, en concreto la prueba de subir y bajar escaleras. Por tanto, en este artículo se describirá de forma detallada la prueba de este campeonato. En el trabajo aquí presentado, se describe el diseño para un robot mini-humanoide de un sistema para abordar la tarea de subir y bajar escaleras. Este sistema abarca el desarrollo de elementos hardware necesarios para la mejora de las capacidades con las que el robot ya contaba. Además será necesaria la implementación de un nuevo software de control acorde con los nuevos elementos hardware introducidos en el sistema, de tal forma que se dote al robot mini-humanoide de un sistema autónomo y fiable para realizar la tarea de subir y bajar escaleras.

Palabras clave: Robot mini-humanoide, sensor infrarrojo, CEABOT, detección de escalones, centro de gravedad.

1. INTRODUCCIÓN

Uno de los principales desafíos en robótica humanoide es el abordar tareas de la vida cotidiana de una forma sencilla. Tareas como el desplazamiento por el entorno, subir y bajar escaleras o agarrar objetos de una forma correcta, son unas de las muchas tareas que se estudian en este campo.

Una manera de iniciarse en este campo es mediante la participación en concursos de robótica mini-humanoide, los cuales permiten a estudiantes de ingeniería realizar una primera toma de contacto con los dispositivos utilizados en estas pruebas, como se subrayó en [6] y [13]. Estos concursos son una manera fácil y activa de introducirse en el desarrollo de sistemas para abordar los principales desafíos que estas tareas codidianas nos plantean, además de una forma de comparar el desarrollo de nuestros sistemas con los del resto de participantes de estas competiciones. Un ejemplo es la competición española CEABOT [1], que se celebra anualmente, donde los equipos de estudiantes pueden probar sus robots mini-humanoides en tres pruebas diferentes. Esto se describe con más detalle en la sección II.

Se están realizando numerosos estudios sobre cómo debe abordar un robot humanoide la tarea de subir y bajar escaleras, los cuales abren un amplio abanico para la mejora de un robot mini-humanoide a la hora de realizar esta tarea. Uno de estos trabajos, es el que se muestra en [9], donde se propone abordar esta tarea a través del control de la fuerza aplicada a cada articulación y un acelerómetro. Aunque esta propuesta es una buena forma de controlar el robot en esta tarea, ésta depende de que la escalera sea conocida ya que no se realiza una detección de la posición de los escalones. Algo similar ocurre con el artículo mostrado en [8], en el cual se plantea el control de la cinemática del robot mediante la sensorización de la fuerza y el torque aplicado a cada articulación.

Existen otros estudios que utilizan detección del entorno para localizar la escalera, su tamaño, altura y demás características relevantes. Uno de estos estudios es el planteado en [11], donde se plantea la utilización de un sistema de visión apoyado por un láser para la detección de los escalones. Otro sistema que utiliza visión para determinar la posición de las escaleras, es el mostrado en [5], el cual a través de un sistema estereoscopico de visión es capaz de determinar la longitud, posición y tamaño de los escalones que desea subir. Estos sistemas basados en visión por computador son una buena opción a la hora de desarrollar un sistema de detección de escaleras, aunque con la capacidad de computación del controlador utilizado en este proyecto no es viable la creación de un sistema de este tipo.

A la vista de los estudios mostrados anteriormente y de las capacidades de computación de nues-

tro robot, se ha decidido utilizar una sensorización sencilla basada en sensores infrarrojos, debido a que el robot que se utiliza en este proyecto no tiene suficientes capacidades para desarrollar sistemas de visión. Además los motores de este robot no están capacitados para realizar un control fuerza-par.

En el resto de este artículo se describe el entorno de la competición CEABOT con una descripción detallada de la prueba de escaleras en la cual se basa este artículo, seguido del diseño tanto hardware como software del sistema propuesto para realizar la tarea de subir y bajar escaleras con un robot mini-humanoide. Este diseño está basado en la incorparación de sensores infrarrojos posicionados en los pies del robot, en el re-posicionamiento de las baterías y en la creación de un software de control apropiado para las nuevas incorporaciones hardware.

2. ENTORNO DE TRABAJO

Las mejoras que se plantean en este artículo tienen como objetivo dotar a un robot mini-humanoide con la capacidad para subir y bajar escaleras de una forma sencilla y fiable. Para comprobar estas capacidades, se participa en el concurso CEABOT de robótica mini-humanoide, el cual se realiza anualmente dentro de las jornadas de automática desarrolladas por el Comité Español de Automática (CEA). Podemos observar la normativa de este concurso en [1].

Para la elección de una plataforma de robot mini-humanoide en la que basar nuestro proyecto, debemos tener en cuenta las restricciones que se determinan en la normativa CEABOT, sobre todo la altura y las restricciones de peso. Las restricciones más relevantes son un peso máximo de 3 kg, una altura máxima de 50 cm con el robot completamente extendido y una longitud máxima de la base de 11 cm. Además no está permitido que los concursantes realicen instrucciones una vez que el robot está encendido y puesto en marcha, por lo que es necesario que los robots mini-humanoides sean totalmente autónomos.

Observando las plataformas de robótica mini-humanoide que se encuentran en el mercado, vemos que no todas las plataformas cumplen con estos requisitos. Por ejemplo, plataformas como DARwinOP[4] y HOAP-3[7], no pueden participar en este torneo, ya que superan el peso y/o las limitaciones de altura. Hay otras plataformas en el mercado que cumplen con estos requisitos y que se utilizan habitualmente en los torneos de este tipo. Plataformas como Robonova[12] y Bioloid[3] han sido las más utilizadas en ediciones anteriores de CEABOT. El trabajo aquí presentado está basado en el robot mini-humanoide Bioloid, el cual se observa en la figura 1.



Figura 1: Robot Bioloid.

El campeonato CEABOT está compuesto por tres pruebas con las que se determinan las capacidades de los robots mini-humanoides participantes. Estas pruebas son las siguientes:

- Carrera de obstáculos: Consiste en hacer que el robot cruce un escenario con una trayectoria de ida y vuelta desde un punto de partida previamente establecido. El campo estará provisto de varios obstáculos esparcidos por toda la superficie, por lo que el robot deberá ser capaz de detectarlos y esquivarlos.

- Escalera: Esta prueba se basa en superar varios escalones tanto de subida como de bajada. El robot deberá recorrer todos los escalones sin caerse y en el menor tiempo posible para superar la prueba de una forma óptima. En esta prueba es en la que se basa este proyecto y por tanto se describirá posteriormente de una forma más detallada.

- Prueba de lucha (Sumo): En esta prueba, dos robots se enfrentan en una lucha similar al Sumo, dentro de un área de combate. El escenario consiste en un círculo de color verde de 1.5 m de diámetro delimitado por una línea amarilla de 5 cm de ancho, denominada área exterior. El objetivo de la prueba es localizar y derribar al robot rival, sin que el robot propio sea derribado.

La puntuación final se realiza sobre 100 puntos siendo el 35 % de los puntos para la prueba de obstáculos, otro 35 % la prueba de la escaleras y el 30 % restante correspondiente a la prueba de sumo.

2.1. Descripción detallada de la prueba de escaleras

Como se ha mencionado anteriormente, este proyecto se basa en la consecución de la prueba de escaleras del campeonato CEABOT, de tal forma que, para diseñar un sistema que dote al robot de tales capacidades, es conveniente el estudio detallado de la prueba mencionada.

Esta prueba consiste en subir y bajar ciertos escalones con un robot mini-humanoide totalmente automatizado. La puntuación depende del número de escalones superados y del tiempo que se tarde en recorrerlos. Como se comprueba en la tabla 1, es conveniente que el robot supere el máximo de escalones posibles sin penalización aunque el tiempo que emplee en ello sea mayor, ya que el número de puntos es superior por completar los escalones que por hacerlo en poco tiempo.

Llegada parcial	Puntos	Número de escalones superados	Puntos
1	10	1	3
2	8	2	5
3	7	3	10
4	6	4	18
5	5	5	22
6	4	6	25
7	3		
8	2		
9	1		

Tabla 1: Puntuación de la prueba de escaleras

La escalera tiene una forma predeterminada, tiene un total de 1.4 m de longitud y 1 m de ancho y esta compuesta por seis escalones, tres de subida y tres de bajada. Estos escalones son de 3 cm de altura y su longitud puede variar entre 15 cm, 25 cm y 50 cm. El escalón más extenso, de 50 cm, está colocado en la parte superior de la escalera y el resto se distribuyen como se aprecia en la figura 2.

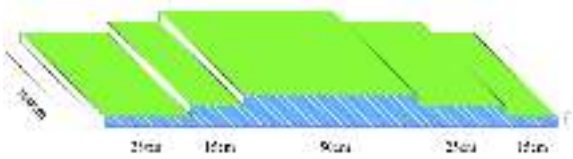


Figura 2: Esquema de las escaleras del campeonato CEABOT.

Como se comprueba en la figura 2, el recorrido es diferente en función del extremo por el que se comience la prueba. Esto es algo a tener muy en cuenta ya que no se podrá realizar un sistema basado en la distancia de los escalones.

El tiempo límite de la prueba es de cinco minutos, que se empieza a contabilizar a partir de que el robot se ponga en movimiento, previa pausa de 5 segundos. El cronómetro se parará una vez que el humanoide haya cruzado totalmente la línea de llegada.

Como se ha visto en esta sección, en el diseño de un sistema que ofrezca la capacidad de subir y bajar escaleras a un robot mini-humanoide, se deben tener en cuenta varios factores referentes al entorno de trabajo y forma de la escalera. La detección correcta de los escalones, un buen posicionamiento previo al abordaje del movimiento de bajada o subida, así como que dichos movimientos sean fiables y robustos, son los principales objetivos que se persiguen en este proyecto.

3. DESCRIPCIÓN DEL HARDWARE DEL SISTEMA PROPUESTO

La mejora de las capacidades de subida y bajada de escaleras de un robot mini-humanoide, conlleva realizar varios ajustes tanto en el software como en el hardware del robot. En esta sección se describen los elementos hardware utilizados para la mejora de las capacidades del robot mini-humanoide en dicha tarea.

Con el fin de mejorar la detección de las escaleras, se ha determinado utilizar sensores de menor tamaño que los utilizados anteriormente, con el fin de poder posicionar mayor número de sensores de tal modo que se aumente el número de puntos a localizar. De esta manera aumentará la precisión de la posición en el entorno, lo cual será de ayuda tanto a la hora localizar la escalera dentro de nuestro entorno, como de colocar el robot de manera correcta antes abordar el movimiento de subir o bajar escaleras.

Se ha determinado utilizar ocho sensores, cuatro para la detección de un escalón de subida y los otros cuatro para la detección de escalones de bajada. Como se muestra en la figura 3, los sensores de escalones de subida se posicionarán de tal forma que la línea de detección sea paralela al suelo. En los sensores de detección de escalón de bajada la línea de detección será perpendicular al suelo, lo cual se observa en la figura 4.

Los sensores frontales funcionan como detectores de escaleras a la vez que miden la distancia. Esta configuración de cuatro sensores permite determinar el ángulo con el que se aproxima el robot al escalón, lo cual nos es de gran ayuda a la hora de

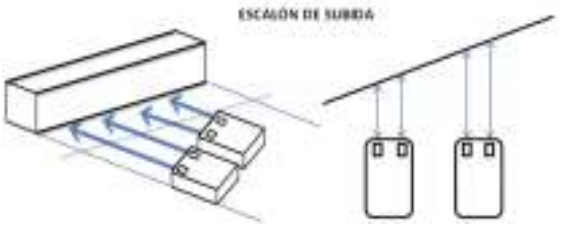


Figura 3: Esquema del posicionamiento de los sensores de detección de escalones de subida.

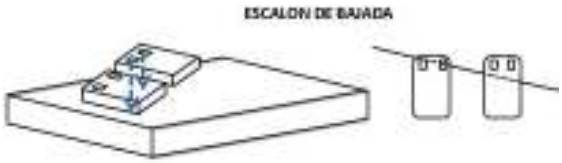


Figura 4: Esquema del posicionamiento de los sensores de detección de escalones de bajada.

posicionar el robot de forma correcta antes de comenzar el movimiento de subida de las escaleras. Mediante este sistema se evita que el robot pueda tomar un escalón de lado, lo cual puede producir caídas. Los sensores de detección de escalón de bajada, gracias a su posicionamiento similar a los anteriores, también permitirán realizar un posicionamiento correcto del robot antes de comenzar el movimiento de bajada.

Por otra parte, se ha decidido bajar el centro de gravedad del robot, de tal forma que a la hora de realizar un movimiento de subida o de bajada, éste sea más estable y fiable. Para ello se eliminará la batería que el robot Bioloid tiene incorporada en el cuerpo y se pondán nuevas baterías situadas en las bases del robot, con lo cual la posición del centro de gravedad se verá reducida en altura.

A la vista de los cambios en el posicionamiento de las baterías y de la gran cantidad de sensores que se van a colocar en los pies del robot, se ha determinado crear un soporte para el robot que contenga tanto las baterías como los sensores.

3.1. Descripción del sensor a utilizar

Para hacer una correcta elección del sensor es necesario tener en cuenta varios aspectos como la distancia que es capaz de medir, la precisión, la rapidez con la que detecta, el modo de lectura de los datos, el tamaño, etc.

Debido a que se colocarán cuatro sensores, con su respectivo circuito acondicionador, en cada pie (dos sensores de detección de escalón de subida y dos de escalón de bajada), es importante que el tamaño de dichos sensores sea lo más reducido posible.

Tras realizar un análisis de los posibles sensores a utilizar para esta tarea, se ha determinado la utilización de sensores infrarrojos ya que la mayoría de ellos se adaptan bien a las necesidades de este proyecto. Entre la gran variedad de sensores infrarrojos que existen, se ha decidido utilizar el sensor desarrollado por compañeros de la Universidad Carlos III, en concreto, el sensor desarrollado en [10], el cual se muestra en la figura 5.

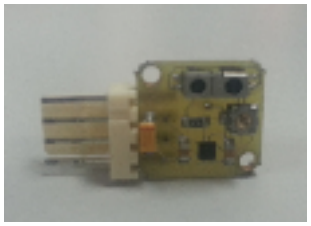


Figura 5: Imagen del sensor utilizado en este proyecto.

Este sensor nos permite recoger medidas del rango de 20mm a 100mm, ya que como se comprueba en la gráfica 6 existe una zona, entre 5mm y 20mm, en la cual las medidas podrían dar cabida a una lectura errónea de la distancia. Su funcionamiento está basado en un circuito que consta de un emisor que convierte una señal eléctrica en una señal óptica y que es emitida al receptor o fotodiodo mediante reflexión con los objetos. En función del tiempo que tarda en recibir esta señal óptica, el circuito acondicionador produce una variación de tensión, la cual es convertida a distancia.

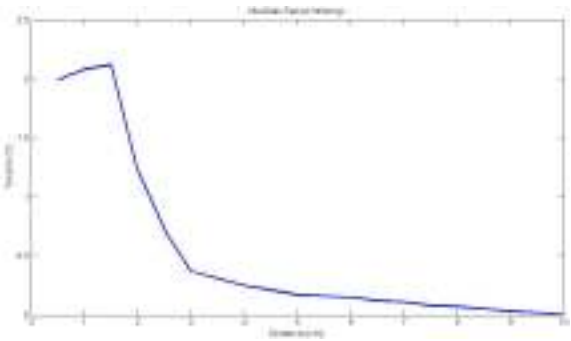


Figura 6: Gráfica tensión[V]/distancia[cm] del sensor infrarrojo.

Para que el fotodiodo sea capaz de emitir la señal es necesario introducirle a través del circuito acondicionador una señal PWM de 100us de periodo, una amplitud en la señal de 3V y un ciclo de trabajo del 20 %. Esto excita al emisor que envía la señal óptica que es captada por el receptor y procesada posteriormente por el circuito de acondicionamiento.

Dependiendo del puerto del chip de acondiciona-

miento que se utilice como salida, se obtenetrá un comportamiento distinto en el sensor. Si la salida es el pin 6 (PFLIT) obtendremos un sensor de distancia cuya tensión de salida varía dependiendo de la longitud a la que se encuentre el objeto. Este comportamiento se puede ver en la figura 6, donde se muestra la gráfica tensión/distancia de este sensor. Por otro lado, si la salida elegida es el pin 3 (DOUT), el sensor tiene un comportamiento todo o nada, es decir, proporciona un pulso a nivel alto si capta un objeto; por tanto se obtendrá una señal binaria.

3.2. Descripción del posicionamiento de las baterias

Las baterías son los componentes primordiales a la hora de dotar de autonomía a los robots, aunque también son uno de los elementos más pesados de un robot. Esto supone que un posicionamiento elevado de las baterías genera un centro de gravedad elevado, lo cual no es conveniente a la hora de realizar movimientos que pueden comprometer la estabilidad del robot.

Por defecto, el robot Bioloid tiene una batería instalada a una altura media, cerca del controlador. Para tener una mayor estabilidad y dotar al minihumanoide de mayor equilibrio es necesario cambiarlas a una posición más baja. El lugar escogido para tales efectos son los pies, con lo que se varia su centro de masas, situandose más bajo y obteniendo así un momento de inercia menor. Además para equilibrar el peso es necesario que se coloquen de manera simétrica con respecto a la estructura del robot, para lo cual, se utilizarán dos baterías en lugar de una, cada una de ellas colocadas en un pie. Además, se obtendrá una mayor autonomía al aumentar la capacidad de sistema de alimentación del robot.

Las baterías con las que cuenta el robot Bioloid, que se observan en la figura 7 son de tipo LIPO de 1000 mAh y proporcionan un voltaje de 11.1 V. Tienen un tamaño de 70mm de largo, 35mm de ancho, 15 mm de espesor y su peso es de 83 gramos. Estas medidas son de gran ayuda a la hora de diseñar un soporte para la colocación de las baterías en los pies, así como para saber la magnitud de peso que se va a bajar.

A continuación se describe el soporte donde se situará la batería y los sensores. Como ya se ha comentado con anterioridad, este soporte se realiza para ser colocado en los pies y albergará tanto las baterías como los sensores.



Figura 7: Imagen de la batería del robot Bioloid.

3.3. Descripción del soporte de sensores y bateria

Para poder alojar los nuevos sensores y las baterías, es imprescindible modificar la estructura de los pies del robot Bioloid. Para ello, se ha diseñado un soporte mediante el cual se fija la batería al pie del robot y que además permite anclar los nuevos sensores de una forma sólida. En la figura 8 se muestra el diseño de dicho soporte, en la cual se observa la nueva pieza de color verde. En esta figura se puede observar también la posición de los sensores sobre los pies del robot.

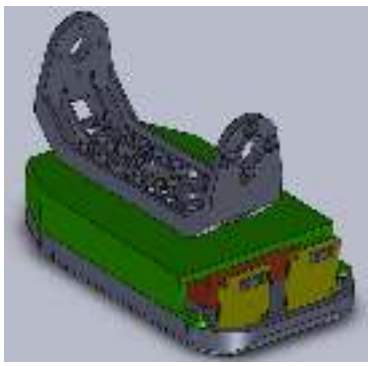


Figura 8: Imagen del diseño del soporte necesario para la nueva configuración.

Como se comprueba en la figura 8, este soporte satisface las necesidades para la configuración propuesta en este artículo, aunque también se comprueba que el robot tendrá algo más de altura y por tanto, se deberá realizar una nueva cinemática de movimientos de subida y bajada acorde con la nueva altura y configuración del robot.

4. DESCRIPCIÓN DEL SOFTWARE DE CONTROL DEL SISTEMA

En esta sección se describe el sistema software encargado de controlar el robot en la tarea de subir y bajar escaleras. Este sistema está basado en el flujograma que se muestra en la figura 9, el cual se describirá en esta sección.

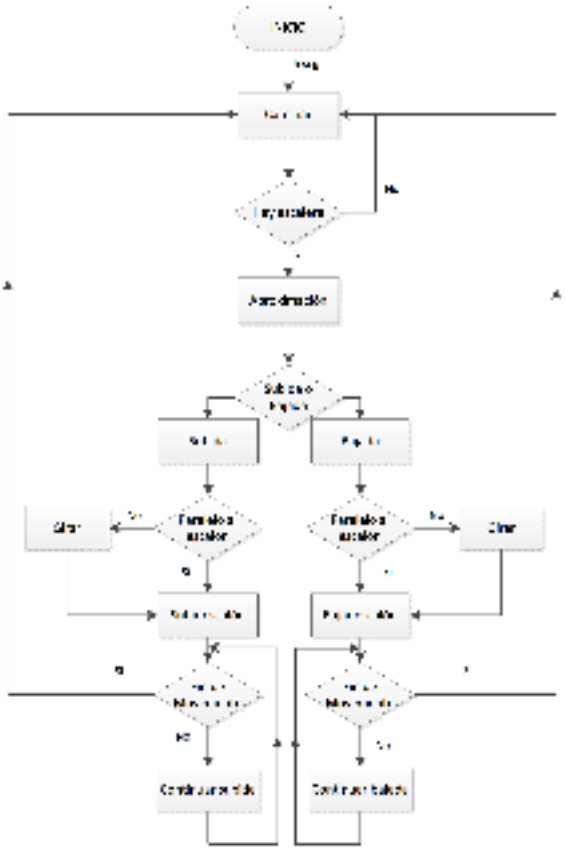


Figura 9: Flujograma del sistema de control del robot mini-humanoide para la prueba de escaleras.

Una vez que se pone en marcha el robot permanecerá parado durante cinco segundos, tiempo establecido por las reglas del concurso. Cuando transcurra ese tiempo, el robot comenzará a caminar hasta que consiga detectar un escalón. Cabe destacar que el movimiento de caminar debe realizarse de forma lenta y segura, ya que si se realizan los movimientos demasiado rápido, el robot podría no detectar a tiempo el escalón y podría producirse una caída. La condición para realizar la siguiente acción es la detección de un escalón, que aunque como se sabe por la normativa de CEABOT será de subida, el robot podría perfectamente detectar un escalón de bajada. Para ello, los sensores están siendo leídos en todo momento tratando de detectar el escalón.

Cuando se ha detectado el escalón, el robot debe de colocarse de manera que cuando vaya a subir o bajar las escaleras, esté posicionado de forma correcta, es decir, debe abordar la tarea de subir o bajar perpendicular al vértice del escalón. Esto se realiza para evitar que el robot tropiece o roce con el suelo, a causa de un mal posicionamiento en el comienzo del movimiento de subida o bajada de escaleras y así evitar caídas o penalizaciones de los jueces.

Para que el robot se posicione de manera correcta es necesario que los sensores utilizados tanto para la detección de los escalones de subida como para los de bajada tengan aproximadamente la misma medida en la lectura. Esto es posible gracias al posicionamiento de los sensores de los pies el cual se puede observar en la figura 10, donde también se aprecia la nomenclatura con la que se nombran los sensores.

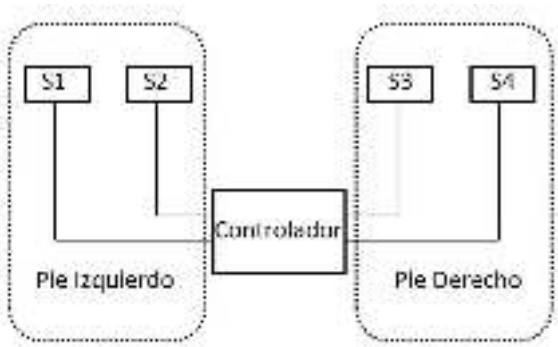


Figura 10: Esquema de conexiones de los sensores con el controlador.

Por tanto, cuando la medida del sensor S1 sea inferior que la medida del sensor S4, el robot se encontrará girado hacia la derecha respecto del escalón de tal forma que deberá girarse hacia la izquierda hasta que la medida de los cuatro sensores sea aproximadamente la misma.

Una vez que los sensores mantienen unas medidas de lectura similares, el robot está preparado para realizar el movimiento de subida o bajada del escalón, en función del tipo de escalón que haya detectado con anterioridad. Estos movimientos de subida o bajada son movimientos predeterminados, ya que, como hemos visto en la descripción de la prueba de escaleras, los escalones son siempre de la misma altura y por tanto se podrán realizar movimientos similares para los escalones de subida, al igual que ocurre para los escalones de bajada.

Cuando el robot ha terminado el movimiento de subida o bajada, éste vuelve al inicio, como se aprecia en la figura 9, de tal forma que volverá a caminar buscando un nuevo escalón para subir o bajar. El proceso que se ha descrito en esta sección se repite cíclicamente lo cual permitirá al robot finalizar la prueba completamente, ya que cuando haya subido y bajado todos los escalones de la prueba el robot continuará caminando hasta sobrepasar la línea que marca el final del campo de trabajo.

5. CONCLUSIONES

En este artículo se ha planteado el diseño de un sistema para que un robot mini-humanoide pueda subir y bajar escaleras de una manera sencilla y fiable. Gracias al estudio de las diferentes formas de abordar esta tarea se han comprendido mejor las dificultades y retos que plantea el desarrollo de un sistema de estas características.

Este proyecto en concreto se ha centrado en desarrollar un sistema para la prueba de escaleras del concurso español de robótica mini-humanoide CEABOT, el cual se ha descrito en este artículo, haciendo especial hincapié en la prueba ya comentada. Gracias a esto, se han podido determinar las principales características del entorno de trabajo, lo cual ha sido vital para poder desarrollar un sistema adecuado a las necesidades.

El diseño del sistema para subir y bajar escaleras ha consistido en la mejora de las capacidades, tanto hardware como software, que tenía el robot mini-humanoide Bioloid. Los principales cambios hardware que se han producido en este robot son la incorporación de varios sensores para una estimación del entorno más adecuada a las necesidades de la prueba. Además, se han desplazado las baterías a la parte baja del robot con el fin de proporcionarle una mayor estabilidad, lo cual ha supuesto el diseño de un nuevo soporte para los pies del robot que albergará tanto las baterías como los nuevos sensores. Por último, se ha desarrollado un nuevo sistema software para el control del robot acorde con los nuevos elementos hardware instalados en el robot.

Cabe destacar que, en estos momentos, el sistema se encuentra en fase de pruebas y afinamiento, aunque la prueba definitiva será cuando se compruebe como se comporta este sistema cuando sea probado y comparado, con otros concursantes, en el campeonato CEABOT 2013, en el cual el grupo de robótica mini-humanoide de la Universidad Carlos III participará este año.

Agradecimientos

Los autores de este artículo quieren agradecer el apoyo al proyecto ARCADIA DPI2010-21047-C02-01, proyecto financiado por el ministerio MINECO del Gobierno de España y a Robocity2030 S2009/DPI-1559 financiado por la Comunidad de Madrid y los fondos estructurales de la Unión Europea. También nos gustaría agradecer a la Asociación de Robótica (ASROB) [2] y al Espacio Abierto de Estudiantes de la Universidad Carlos III por darnos los medios para desarrollarlo.

Referencias

[1] A. Jardón, J. C. García, G. Alenya, F. Blanes and F. Gómez. CEABOT'13 - Normativa. VII Concurso de Robots Humanoides. Comité Español de Automática (CEA). Vigo, Spain, 2012.

[2] Asociación de Robótica de la Universidad Carlos III (ASROB). Disponible online[Jul, 2013]: http://asrob.uc3m.es

[3] Bioloid, ROBOTIS. Disponible online [Jul, 2013]: robotis.com/xe/bioloid-en

[4] DARwinOP, Open Plataform Humanoid Robot for Research and Education. Disponible online [Jul, 2013]: romela.org

[5] Gutmann, J. S., Fukuchi, M., Fujita, M. Stair climbing for humanoid robots using stereo vision. In Intelligent Robots and Systems. IROS Proceedings IEEE/RSJ International Conference on IEEE, Vol. 2, pp. 1407-1413, 2004.

[6] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda and E. Osawa. RoboCup: The Robot World Cup Initiative. IJCAI-95 Workshop on Entertaiment and AI/Alife, 1995.

[7] HOAP-3, Miniature Humanoid Robot. Fujitsu. Disponible online [Jul, 2013]: home.comcast.net/jtechsc/HOAP-3-SpecSheet.pdf

[8] Kim, J. Y., Park, I. W., Oh, J. H. Realization of dynamic stair climbing for biped humanoid robot using force/torque sensors. Journal of Intelligent and Robotic Systems, 56(4), pp. 389-423, 2009.

[9] Li, T. H., Su, Y. T., Kuo, C. H., Chen, C. Y., Hsu, C. L., Lu, M. F. Stair-climbing control of humanoid robot using force and accelerometer sensors. In SICE, Annual Conference IEEE, pp. 2115-2120, 2007.

[10] M. González, A. Jardón, S. Martínez, M.F. Stoelen, J.G. Victores and C. Balaguer. Educational initiatives related with the CEABOT contest. In Proc. of 2nd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010), Darmstadt, Germany, pp. 649-658.

[11] Obwald, S., Gorog, A., Hornung, A., Bennewitz, M. Autonomous climbing of spiral staircases with humanoids. In Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on, pp. 4844-4849, 2011.

[12] Robonova, Hitec Robotics. Disponible online [Jul, 2013]: robonova.de/store/home.php

[13] Tena G., Miguel Ángel. "Diseño y desarrollo de un sistema anticolisión con sensores de proximidad sin contacto". Proyecto Fin de Carrera, Departamento de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, Junio 2011.

Control de posición/fuerza de un robot antropomórfico basado en control predictivo

J. de la Casa Cárdenas, J. Gámez García, J. Gómez Ortega
Grupo de Robótica, Automática y Visión por computador. Universidad de Jaén
{jesus.casacardenas}@ujaen.es

Resumen

En este trabajo se propone un nuevo controlador para el control de posición/fuerza de un brazo robótico antropomórfico en entornos restringidos. El control de posición/fuerza se puede realizar mediante la conmutación de controladores de posición y de fuerza, una vez alcanzado el punto de contacto; esta aportación utiliza un único controlador para ambas magnitudes y el control de una magnitud u otra (posición o fuerza) se consigue mediante la modificación de los valores ponderativos presentes en la función de coste del controlador predictivo. Validado el trabajo a nivel de simulación, se observa la bondad del controlador, basado en el modelo identificado experimentalmente del brazo antropomórfico.

Palabras clave: Control de robots manipuladores, control de posición/fuerza, control predictivo.

1. Introducción

Dentro del campo de la robótica dedicado a la interacción de los robots manipuladores con personas o entornos no estructurados, se engloban aquellas investigaciones centradas en la robótica industrial o bien aquellas aplicadas a robótica de servicios. Particularizando en la robótica de servicios, ésta se enfoca en los robots que trabajan en entornos poco estructurados, precisando gran capacidad e integración de procesamiento sensorial.

A lo largo de las últimas décadas ha existido un interés creciente en el control de fuerza en robots manipuladores, también conocido como control de robots manipuladores con movimientos restringidos [2] [1]. Esto se debe a la necesidad de adaptar el robot a las incertidumbres del entorno y proveer al manipulador con una medida segura de la fuerza que le prevenga de colisiones o impactos [5]. Para resolver este problema, se han propuesto principalmente tres técnicas de control: control basado en la impedancia [4], control híbrido de posición/fuerza [6] y control paralelo de posición/fuerza [3].

La utilización del control predictivo [11] en el control de procesos es una línea de investigación ya clásica [9]. Este hecho viene motivado por las posibilidades que ofrece como técnica de control. Por un lado, permite la implementación de forma natural de restricciones al sistema; por otro, ofrece la posibilidad de avanzar la acción de control considerando futuras referencias, compensando posibles retrasos e incluso prediciendo y ayudando a superar posibles perturbaciones y comportamientos de fase mínima [10]. Aunque existen muchos trabajos que han mostrado que el control predictivo es una opción interesante para el manejo de diversos procesos industriales, existen pocas aplicaciones de este tipo de control en sistemas donde las dinámicas son relativamente rápidas, no lineales, o el lazo de control es de unos pocos milisegundos. En el caso de los robots manipuladores, estos tres factores confluyen, por lo que existen pocos trabajos al respecto.

Básicamente, existen tres tipos de aplicaciones en las que el control predictivo ha sido aplicado al control de robots manipuladores. La primera de ellas consiste en el desarrollo de aplicaciones de control en posición en robots manipuladores. Como en [10], donde se propone un controlador predictivo multivariable para un brazo robot de dos grados de libertad con accionamiento directo. Para llevar a cabo el controlador, se utilizan modelos lineales, obtenidos a través de linealizaciones basadas en las transformaciones bilineales.

La segunda línea de trabajo consiste en el desarrollo de algoritmos de realimentación visual basados en control predictivo y aplicados al control de la trayectoria de un robot manipulador. Por ejemplo, en [8], se presentó un esquema de optimización *online* de seguimiento de trayectoria para un robot manipulador en el que se utilizaba la información de un sensor de visión, la dinámica del manipulador y un generador de trayectorias.

Con respecto a la tercera línea de trabajo, consistente en la aplicación del control predictivo para controlar la fuerza ejercida por un robot manipulador a su entorno, en [7] se propone un lazo de control de fuerza predictivo para entornos no rígidos en el que se incorpora el modelo del entorno en el controlador para mejorar el rendimiento del controlador. También se llevan a cabo una serie de simplificaciones, como despreciar los términos de inercia y coeficientes de viscosidad del entorno o

bien considerar condiciones cuasi estáticas. Finalmente, comentar que el controlador se aplicó exclusivamente a la fase de movimientos restringidos.

También es interesante destacar que existen aplicaciones del control predictivo dirigidas a la coordinación entre robots y humanos. En [12] se plantea el problema del transporte de una carga entre un operario humano y un robot; un sensor de fuerza mide la fuerza y el par que realiza el operario al trasladar una carga y el controlador predictivo consigue que el robot anule esa fuerza, moviéndose de forma similar a él.

Englobado en el contexto de interacción hombre-robot (*HRI, Human-Robot Interaction*), en este trabajo se presenta una nueva propuesta de control de posición/fuerza basada en control predictivo y aplicada a un brazo robótico antropomórfico trabajando en entornos restringidos. A partir de la identificación experimental del modelo en posición del brazo robótico, se ha desarrollado una estrategia de control en la que, basándose en una única ley de control, se controla tanto la posición del brazo, como la fuerza ejercida por éste.

El trabajo se estructura como sigue: en la Sec. 2 se presentan los modelos del brazo antropomórfico, entorno y el modelo completo del sistema.La Sec. 3 expone el proceso de identificación experimental del modelo del brazo robótico antropomórfico. En la Sec. 4 se desarrolla la estructura del controlador propuesto en este trabajo, mostrando los resultados de simulación en la Sec. 5. Finalmente, la Sec. 6 muestra las conclusiones a las que se ha llegado tras experimentar con el nueva propuesta de control.

2. Modelado experimental

Modelo del brazo robótico

Se asume que la dinámica del robot, para cada eje cartesiano i, puede modelarse de forma discreta en el espacio de estados, Ec. (1). El sistema está formado por n estados, m entradas y p salidas.

$$\begin{cases} \xi_i(k+1) = A_{Ri}\xi_i(k) + B_{Ri}p_{ref_i}(k) \\ y_i(k) = C_{Ri}\xi_i(k) + D_{Ri}p_{ref_i}(k) \end{cases} \quad (1)$$

siendo $\xi_{i_{n\times 1}} = [\xi_{i_1} \quad \xi_{i_2} \quad ... \quad \xi_{i_n}]^T$ el vector de estado, $p_{ref_{i_{n\times m}}}$ la referencia de posición para el eje i e $y_{i_{p\times n}}$ la salida del modelo siendo A_{Ri}, B_{Ri}, C_{Ri} y D_{Ri} las matrices del modelo. k representa el instante de muestreo actual.

Modelo del entorno

Cuando un manipulador hace contacto con el entorno, la fuerza de reacción producida, para cada eje i, se define como (Fig. 1):

$$F_{env_i} = K_e(p_i - p_{e_i}) + B_e(\dot{p}_i - \dot{p}_{e_i}) \quad (2)$$

donde K_e y B_e representan la rigidez y el amortiguamiento del entorno, respectivamente, mientras que p_{e_i} corresponde a la posición del entorno (punto de contacto) y p_i es la posición del robot [14].

Puesto que la fuerza solo se ejerce a partir de que el robot haya llegado al punto de contacto, F_{e_i} tiene la forma:

$$F_{env_i} = \begin{cases} 0 & si \quad p_i \leq p_{e_i} \\ K_e(p_i - p_{e_i}) & si \quad p_i > p_{e_i} \end{cases} \quad (3)$$

donde, en este trabajo, se ha considerado que el entorno es rígido y se ha despreciado el coeficiente de amortiguamiento B_e.

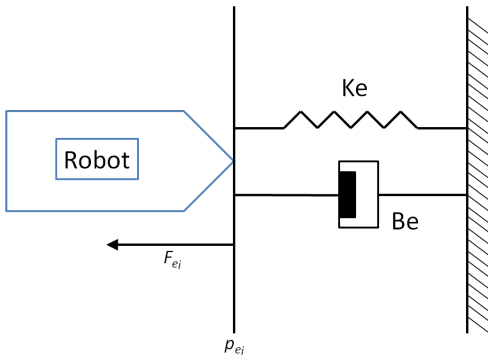


Figura 1: Esquema simplificado que define el comportamiento del entorno.

Modelo completo del sistema

Con el fin, como se ha comentado anteriormente, de tener una única ley de control que permita tanto el control de posición como el control de fuerza, esta última debe añadirse, para cada eje i, al modelo matemático del robot. Para ello, se define una nueva variable de entrada, $\Delta up_i = (p_i - p_{e_i})$, que modela la diferencia de posición entre la referencia y la posición del entorno. Así, la fuerza ejercida por el entorno puede reescribirse como $F_{env_i} = K_e\Delta up_i$, apareciendo ésta como un nuevo estado del modelo. Finalmente, el vector de salidas se incrementa con esta nueva variable.

El modelo completo, para cada eje i, expresado en espacio de estados tiene la siguiente forma:

$$\begin{cases} X_i[k+1] & = & A_iX_i[k] + B_iU_i[k] \\ Y_i[k] & = & C_iX_i[k] + D_iU_i[k] \end{cases} \quad (4)$$

siendo $X_i = [\xi_i \quad F_i]^T$ el nuevo estado del sistema, $U_i = [p_{ref_i} \quad \Delta up_i]^T$ la nueva señal de control (y variable a optimizar) e $Y_i = [y_i \quad F_{env_i}]^T$ el nuevo vector de salida.

Finalmente, las matrices del modelo, una vez añadida la variable de fuerza, quedarían según la Ec. (5).

$$A_i = \begin{bmatrix} A_{Ri} & 0_{n\times 1} \\ 0_{1\times n} & 0_{1\times 1} \end{bmatrix}; B_i = \begin{bmatrix} B_{Ri} & 0_{n\times 1} \\ 0_{1\times m} & K_e \end{bmatrix}$$
$$C_i = \begin{bmatrix} C_{Ri} & 0_{p\times 1} \\ 0_{1\times n} & 1 \end{bmatrix}; D_i = \begin{bmatrix} D_{Ri} & 0_{p\times 1} \\ 0_{1\times m} & 0_{1\times 1} \end{bmatrix} \quad (5)$$

3. Identificación del sistema

La identificación de sistemas hace referencia a los métodos utilizados para obtener modelos matemáticos de sistemas dinámicos, basados en la observación y posterior procesamiento de los datos de la planta. Los modelos obtenidos son útiles en la mayor parte de las aplicaciones de control y proporcionan la relación dinámica entre las entradas y salidas que intervienen en los experimentos [13].

Identificación experimental del brazo robótico antropomórfico

La identificación del modelo del robot se ha realizado en el espacio cartesiano para cada uno de los ejes, de manera que se ha obtenido un modelo matemático que describe la relación entre las posiciones de referencia y las reales. En la Fig. 2 se pueden observar las señales de referencia (sinusoidales), así como la respuesta del brazo robótico para los tres ejes cartesianos.

El modelo del robot se ha obtenido mediante técnicas de identificación de sistemas, utilizando el *System Identification Toolbox*^{TM} de Matlab. En la etapa de selección del modelo, se analizaron varios modelos matemáticos como el ARX, ARMAX, OE, BJ y espacio de estados. Finalmente, la mejor correlación de los datos se obtuvo con el modelo *Output Error* (OE); este modelo se define como se muestra en la Ec. (6).

$$\begin{aligned} &w(t) + f_1 w(t-1) + ... + f_{n_f} w(t-n_f) = \\ &= b_1 u(t-1) + ... + b_{n_b} u(t-n_k-n_b+1) \\ &y(t) = w(t) + e(t) \end{aligned} \quad (6)$$

donde,

- $w(t)$ es la salida sin perturbación.

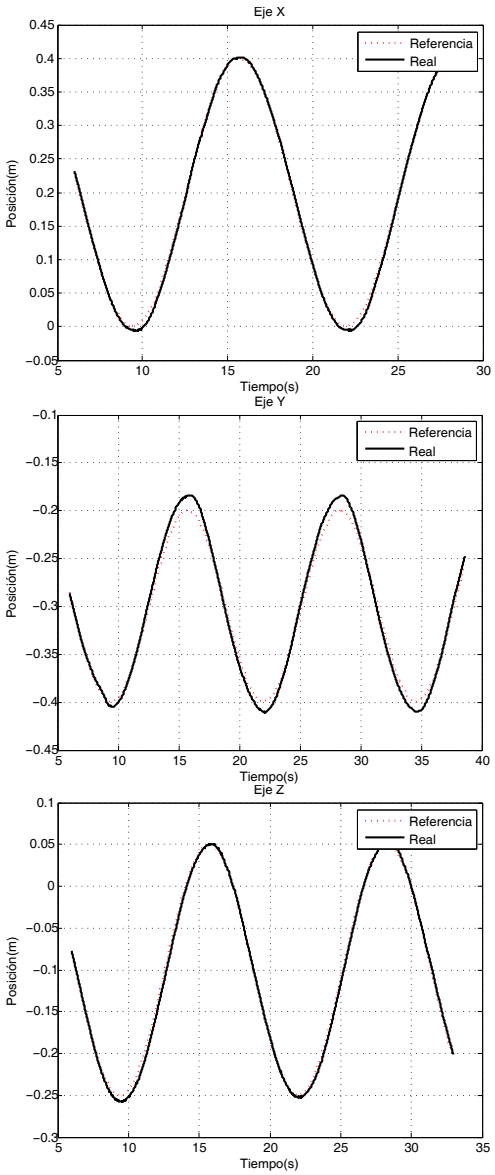


Figura 2: Trayectoria de referencia y real para cada uno de los ejes cartesianos.

- $y(t)$ es la salida en el instante t.
- $f_1...f_{n_f}$ y $b_1...b_{n_b}$ son los parámetros a estimar.
- n_f es el número de polos de la función de transferencia, desde la entrada a la salida sin perturbación.
- $n_b + 1$ es el número de ceros de la función de transferencia, desde la entrada a la salida sin perturbación.
- n_k es el número de muestras de entrada que tienen lugar antes de que las entradas afecten a la salida.
- $u(t-1)+...+u(t-n_k-n_b+1)$ son las entradas previas de las que depende la salida.

- $e(t)$ es una perturbación de tipo ruido blanco.

En forma compacta, el modelo OE puede escribirse como:

$$y(t) = \frac{B(q)}{F(q)}u(t - n_k) + e(t) \tag{7}$$

donde,

$$\begin{array}{l} B(q) = b_1 + b_2 q^{-1} + ... + b_{n_b} q^{-n_b+1} \\ F(q) = 1 + f_1 q^{-1} + ... + f_{n_f} q^{-n_f+1} \end{array} \tag{8}$$

y q^{-1} es el operador de retardo, definido como $q^{-1}u(t) = u(t-1)$.

Para el eje X, la identificación más aproximada se obtuvo tomando $n_b = 3$, $n_f = 3$ y $n_k = 1$, mientras que para los ejes Y y Z, $n_b = 4$, $n_f = 4$ y $n_k = 1$.

4. Control predictivo basado en modelo

En la Fig. 3 se muestra el esquema de control desarrollado en este trabajo. Está basado en el clásico control predictivo basado en modelo al que se le añaden dos elementos nuevos. Por un lado, un módulo que modifica los valores ponderativos de la función de coste; el objetivo es priorizar el control de la posición o de la fuerza, según se esté en espacio libre o restringido. Por otro lado, un módulo que modifica la señal de control en función de la fase de control en la que se encuentre el controlador.

Cada eje cartesiano del robot es controlado a través de un esquema de control como el que aparece en la Fig. 3, de forma que es posible configurar independientemente cada controlador. La función de coste utilizada en este trabajo, para cada eje, es la que se muestra en la Ec. (9). Se trata de una función de coste estándar en la que, dada la referencia de posición y fuerza $r[k]$, el estado actual $X[k]$, el horizonte de predicción N, el horizonte de control M y los valores ponderativos de los errores α y β, el optimizador (ver Fig. 3) proporciona la señal de control óptima $U^*(k)$.

$$\begin{array}{ll} min & J(U) = \sum_{i=1}^{N} \alpha(r[k+i] - \hat{Y}[k+i|k])^2 + \\ & \qquad \sum_{i=1}^{M} \beta(\Delta U[k-1+i])^2 \\ & \qquad\qquad s.a. \\ & \qquad U_{min} \leq U[k] \leq U_{max} \\ & \Delta U_{min} \leq \Delta U[k] \leq \Delta U_{max} \\ & \qquad Y_{min} \leq Y[k] \leq Y_{max} \end{array} \tag{9}$$

donde $\Delta U[k] = U[k] - U[k-1]$ y $\Delta U = [\Delta U[k], \Delta U[k+1], ..., \Delta U[k+M-1]]^T$ es el vector de las variables a optimizar.

Modificación de los valores ponderativos

Los pesos, o también llamados valores ponderativos, tienen una enorme importancia en el rendimiento del controlador de forma que, ajustando adecuadamente el valor de los mismos, se mejora el control de la posición y de la fuerza.

En este trabajo se pretende utilizar una única función de coste para controlar posición y fuerza. Estas magnitudes, como se ha indicado en la Sec. 2, actúan de forma independiente y no coinciden en el tiempo, salvo en la fase de contacto; es decir, en el espacio libre no se está ejerciendo ninguna fuerza, con lo que el único control que debe actuar es el de posición del brazo antropomórfico. Mientras que en espacio restringido, una vez producido contacto, se debe asegurar que el robot ejerce la fuerza, o perfil de fuerza, requerido (control de fuerza).

Con estas premisas, una forma de controlar ambas magnitudes es modificando el valor ponderativo que afecta a cada una de las salidas, de manera que en espacio libre se anula el peso que pondera el seguimiento de la referencia de fuerza y en el espacio restringido se anula el peso que pondera el seguimiento de la referencia de posición.

Según la Ec. 9, α es el valor que pondera el desempeño de las salidas. Puesto que en el modelo completo, mostrado en la Ec. (4), la salida $Y_i[k] = [y_i \quad F_{env_i}]^T$, α se puede definir como $\alpha = [\alpha_1 \quad \alpha_2]^T$, de manera que α_1 penaliza el error de seguimiento de la trayectoria (control de posición) y α_2 penaliza el error en el seguimiento de fuerza (control de fuerza). Suponiendo que los pesos se mantiene constantes durante cada fase de control (espacio libre y espacio restringido), se puede escribir:

$$\begin{array}{l} \alpha = [K_1 \quad 0]^T \quad si \quad p_i \leq p_{e_i} \\ \alpha = [0 \quad K_2]^T \quad si \quad p_i > p_{e_i} \end{array} \tag{10}$$

Modificación de la señal de control

El otro elemento añadido al esquema del controlador modifica la señal de control que le llega al brazo. Teniendo en cuenta la Ec. (4), la variable a optimizar es $U_i = [p_{ref_i} \quad \Delta up_i]^T$.

Sea u_{ref_i} la señal de control para el eje i que llega al robot. En función de la fase en la que se esté controlando el brazo, se tiene:

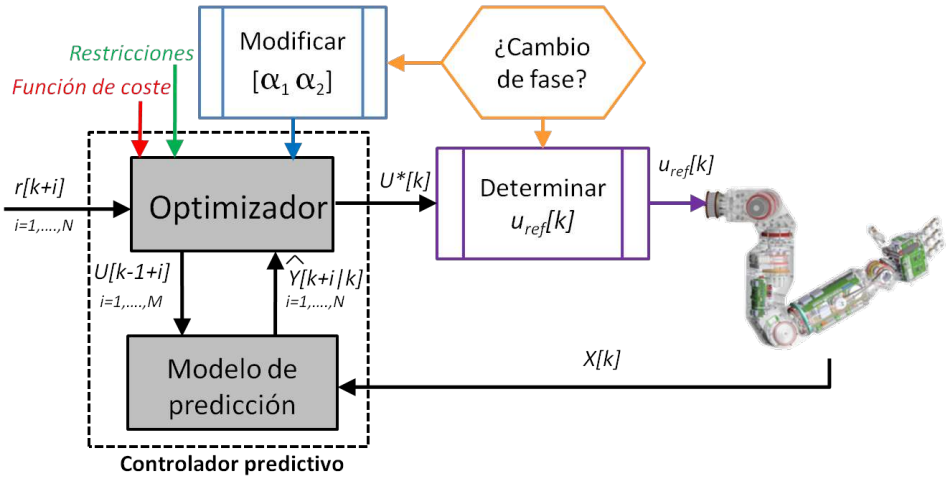


Figura 3: Esquema de control.

$$\begin{array}{llll} u_{ref_i} = p_{ref_i} & si & p_i \leq p_{e_i} \\ u_{ref_i} = \Delta up_i + p_{e_i} & si & p_i > p_{e_i} \end{array} \quad (11)$$

5. Resultados

Para demostrar la bondad del nuevo controlador, se ha realizado una simulación en la que el brazo antropomórfico debe seguir una trayectoria sinusoidal y, durante el movimiento, impacta contra un elemento rígido. Por simplificar el experimento, solo se simula el comportamiento del brazo a lo largo de la dirección de impacto (en este caso, eje X).

El entorno se simula con $K_e = 400N/m$ y esta situado en $p_{e_x} = 0{,}1m$. La referencia de salida es $r_x[k] = [0{,}2\sin(k)m \quad 5N]^T$. Los pesos toman los valores que aparecen en la Ec. 12.

$$\begin{array}{llll} \alpha = [10 \quad 0]^T; & \beta = [0{,}1 \quad 0{,}1]^T & si & p_x \leq 0{,}1m \\ \alpha = [0 \quad 10]^T; & \beta = [0{,}1 \quad 0{,}1]^T & si & p_x > 0{,}1m \end{array} \quad (12)$$

Y la señal de control u_{ref_x} se modifica tal y como aparece en la Ec. 13.

$$\begin{array}{llll} u_{ref_x} = p_{ref_x} & si & p_x \leq 0{,}1m \\ u_{ref_x} = \Delta up_x + 0{,}1m & si & p_x > 0{,}1m \end{array} \quad (13)$$

En la Fig. 4 se muestra el comportamiento del brazo en espacio libre. La Fig. 4(a) representa el desempeño del brazo al realizar el control en posición siguiendo la trayectoria $r_x[k]$ indicada anteriormente. En espacio restringido (Fig. 4(b)), el brazo mantiene la fuerza referenciada. Finalmente, la Fig. 4(c) muestra la señal de control, donde se observa cómo, al ejercer fuerza sobre el entorno, la señal de control se modifica, de acuerdo con el valor obtenido de Δup_x

Inclusión de restricciones

Se ha evaluado, también, el desempeño del controlador ante la presencia de restricciones. En este caso, se ha añadido una restricción a la salida de posición, del tipo límite inferior (Y_{min}); así, de acuerdo con la Ec. (9), la restricción quedaría

$$[-0{,}15m - \infty]^T \leq Y[k] \quad (14)$$

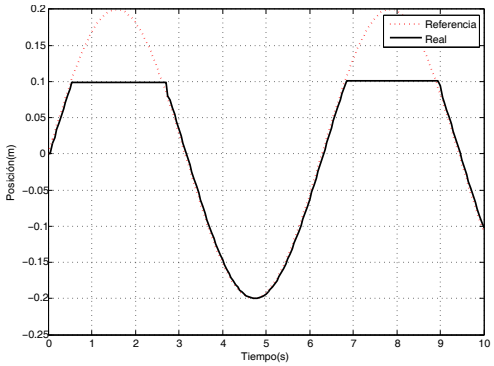
Además, se ha modificado la referencia de fuerza, de forma que $r_x[k] = [0{,}2\sin(k)m \quad (4 + \sin(k))N]^T$.

En la Fig. 5 se presentan los resultados del control en posición, fuerza y la señal de control.

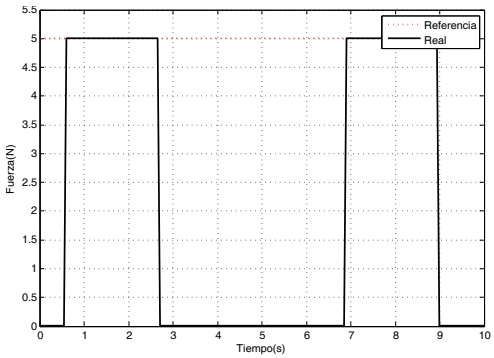
Al igual que en el apartado anterior, se destaca el seguimiento de la referencia de posición y el cumplimiento con la restricción en posición. En cuanto a la fuerza, también evoluciona según la referencia indicada.

6. Conclusiones

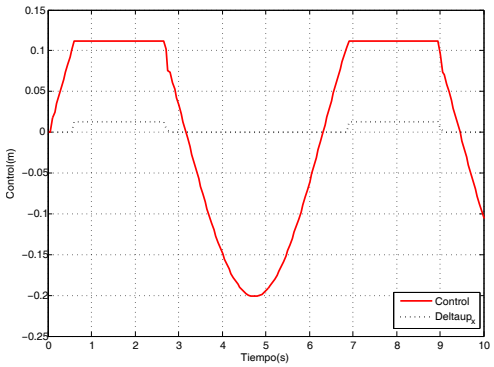
En este artículo se ha presentado una nueva propuesta de control de posición/fuerza basada en control predictivo aplicada a un brazo robótico antropomórfico. Tras identificar experimentalmente el modelo en el espacio cartesiano del brazo robotico, se realiza el diseño del controlador. La novedad reside en disponer de una única ley de control (función de coste) para realizar el control de posición y de fuerza. Esto se consigue mediante la variación *online* de los valores ponderativos que priorizan la consecución de los objetivos en la función de coste. Además, es necesario modificar, de



(a) Control de Posición.

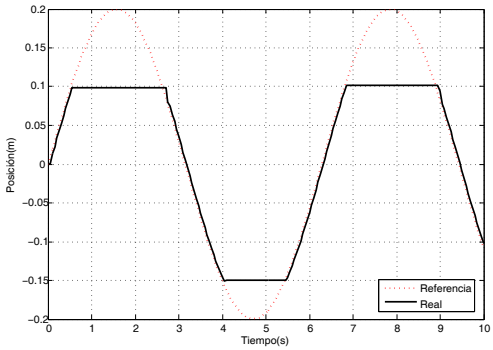


(b) Control de Fuerza.

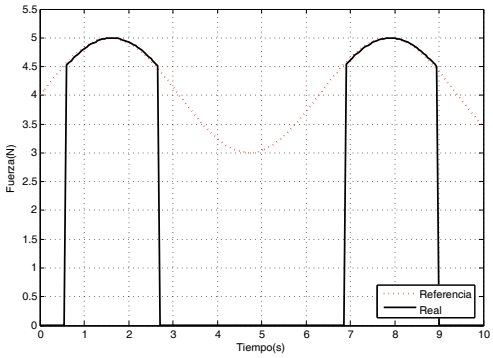


(c) Señal de Control.

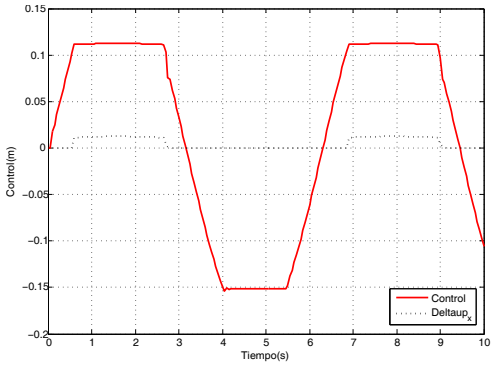
Figura 4: Resultados en simulación.



(a) Control de Posición.



(b) Control de Fuerza.



(c) Señal de Control.

Figura 5: Resultados en simulación con restricciones.

una forma sencilla, la señal de control que recibe el robot. Por último, en los resultados de simulación, se observa cómo el brazo sigue la referencia en posición indicada y, al entrar en contacto con el entorno, ejerce la fuerza que se indica, también, como referencia. Finalmente, al añadir una restricción a la posición, el brazo realiza la trayectoria respetando la imposición hecha. El siguiente paso que se dará será la experimentación del controlador con un brazo antropomórfico real, observando la evolución del mismo, sobre todo, en la fase de impacto.

Agradecimientos

Los autores quieren agradecer la subvención parcial de este trabajo a través de los proyectos DPI2011-27284, TEP2009-5363 y AGR-6429.

Referencias

[1] J. Gámez García. Sensor Fusion of Force, Acceleration and Position for Robot Motion Control. Phd Thesis. Jaén University, Spain. 2006.

[2] Siciliano B. Control in robotics: Open Problems and Future Directions. Proceedings of the International Conference on Control Applications. 1998.

[3] S. Chiaverini, B. Siciliano, L. Villani. Force and position tracking: Parallel control with

stiffness adaptation. IEEE Control Systems Magazine, 18(1), pp. 27-33, 1998.

[4] Hogan N. Stable Execution of Contact Tasks Using Impedance Control. IEEE International Conference on Robotics and Automation, 1987.

[5] Khatib O. and Burdick J. Motion and Force Control of Robot Manipulators. IEEE International Conference on Robotics and Automation, 1986.

[6] Raibert, M. and Craig, J. Hybrid Position/force control of manipulators. J. Dynamic Syst., Measurement Contr., vol. 103, no 2, pp 126-133, 1981.

[7] L. Baptista, J. Sousa, and J. S. da Costa. Force control of robotic manipulators using a fuzzy predictive approach. Journal of Intelligent & Robotic. pp. 359-376, 2001.

[8] Bauchspiess A., Absi Alfaro S.C., Dobrzanski L.A. Predictive sensor guided robotic manipulators in automated welding cells. Journal of Materials Processing Technology, (1-2), pp. 13-19. 2001.

[9] E. Camacho, C. Bordons. Model predictive control in the process industry. Springer-Verlag. 1995.

[10] Valle F., Tadeo F., Álvarez T. Predictive control of robotic manipulators. IEEE Conference on Control Applications. Proceedings, pp. 203-208. 2002

[11] J. Richalet, A. Rault, J. Testud, J. Papon. Model predictive heuristic control: Applications to industrial process. Automatica. Vol. 14. 1978.

[12] Iqbal Kamran, Zheng Yuan F. Arm-manipulator coordination for load sharing using predictive control. Proceedings - IEEE International Conference on Robotics and Automation, 4, pp. 2539-2544. 1999.

[13] Ljung, Lennart. System identification. John Wiley & Sons, Inc., 1999.

[14] D. Erickson, M. Weber, and I. Sharf. Contact Stiffness and Damping Estimation for Robotic Systems The International Journal of Robotics Research, pp. 41-57. 2003.

Hacia una arquitectura de exploración multirobot distribuida para entornos no controlados

Miguel Juliá, Oscar Reinoso, Luis Payá, Francisco Amorós, David Úbeda
Dpto. Ingeniería de Sistemas y Automática, Universidad Miguel Hernández,
Avda. Universidad s/n, edif. Quorum V, 03202, Elche (Alicante), mjulia@umh.es

Resumen

Este artículo aborda el problema de la exploración de entornos desconocidos mediante un grupo de robots móviles. Con este objetivo, se presenta un modelo de arquitectura distribuida en la que se separa la planificación de trayectorias y la creación de mapas entre alto y bajo nivel. A bajo nivel, se emplea un mapa local con ventana deslizante capaz de adaptarse rápidamente ante situaciones cambiantes del entorno, sobre este mapa un sistema de comportamientos reactivos deciden la trayectoria de cada robot. A alto nivel, se procede a utilizar un sistema de grid-SLAM para cada robot con una alineación y fusión de los mapas obtenidos. En este sentido, una de las novedades del artículo es un sistema para obtener características de los mapas de ocupación que permitan realizar el alineado de los mismos para su posterior fusión basado en la transformada distancia del mapa de ocupación y el detector SIFT. El mapa creado de forma conjunta fusionando los mapas independientes es empleado a bordo de cada robot por un planificador de alto nivel que realiza una planificación a largo plazo. Se mostrará dicha arquitectura funcionando en simulación en entornos que modelan escenarios reales.

Palabras clave: exploración autónoma, fusión de mapas de ocupación, SLAM, entornos dinámicos, coordinación multirobot, arquitecturas de navegación.

1. Introducción

Uno de los problemas comunes en robótica móvil es el proceso de recorrer de forma autónoma un entorno desconocido creando un modelo del mismo [8]. Dicho proceso, que se conoce con el nombre de exploración autónoma, puede ser empleado simplemente en la fase inicial de la puesta en marcha de un sistema. Para ello, es necesario adquirir un mapa del entorno de modo que este se pueda usar posteriormente por los robots para localizarse y planificar sus trayectorias de acuerdo a la tarea que se requiera realizar. Dicho caso no tiene generalmente demasiados requerimientos, el tiempo de exploración no es algo crítico y, por lo tanto, no es necesario emplear múltiples robots para crear el mapa, incluso se podría utilizar teleoperación en lugar de un control de trayectorias automático. Además, suelen ser situaciones en las que se puede controlar el entorno, de modo que se puede evitar la aparición de elementos dinámicos en el rango del sensor y de ese modo se pueden emplear los típicos algoritmos de SLAM que asumen un entorno estático [7].

Sin embargo, existen otras aplicaciones donde la tarea a realizar se realiza simultáneamente durante el proceso de exploración. Ese es el caso, por ejemplo, de tareas de búsqueda, búsqueda y rescate, despliegue de redes de sensores, etc. Para estas tareas, el tiempo de exploración puede ser crítico y por lo tanto se puede obtener una gran ventaja del uso de múltiples robots móviles. Además en dichas tareas el robot debe adaptarse a un entorno no controlado donde pueden aparecer elementos dinámicos. Otro factor a tener en cuenta es la robustez del sistema, requiriendo normalmente que se pueda completar la tarea aun ante fallos en alguno de los robots.

En este artículo proponemos una arquitectura de exploración autónoma multirobot que pueda ser usada en este segundo tipo de situaciones donde el tiempo es un factor crítico, el entorno es no controlado y donde se requiere robustez ante fallos en alguno de los robots. La arquitectura propuesta integra diferentes sistemas previamente diseñados junto con nuevas aportaciones. Como principal característica el sistema propuesto está completamente distribuido sin existir ningún nodo común para todos los robots. Además, en cada robot se separa una parte de planificación de trayectorias y creación de mapas de bajo nivel de otra parte de alto nivel. En este sentido, a bajo nivel, empleamos un mapa local con ventana deslizante diseñado para adaptarse rápidamente ante cambios en el entorno. Este mapa es usado por el sistema de comportamientos reactivos que decide la trayectoria del robot. A alto nivel, un sistema de grid-SLAM se emplea para crear un mapa de ocupación para cada robot. Una de las novedades del artículo es un sistema para obtener características de los mapas de ocupación que genera

cada robot que permite realizar de forma robusta el alineado de los mismos para su posterior fusión. La extracción de características está basada en la transformada distancia del mapa de ocupación y el detector SIFT [12]. El mapa creado de forma conjunta fusionando los mapas independientes es empleado a bordo de cada robot por un planificador de alto nivel que realiza una planificación a largo plazo.

El resto del artículo se organiza de la siguiente forma. En la sección 2 se revisan algunos de los trabajos relacionados, en la sección 3 se muestra la arquitectura desarrollada. La sección 4 muestra algunos experimentos que simulan la el uso del sistema desarrollado. Acabamos en la sección 5 mostrando las conclusiones y las futuras líneas de investigación.

2. Trabajos Relacionados

Recientemente hemos desarrollado arquitecturas de exploración multirobot [10, 9]. En general, dichos modelos estaban enfocados hacia la integración del SLAM con la planificación de trayectorias para la exploración en lo que se conoce como algoritmos de exploración integrada. En este sentido, veníamos empleando un sistema de SLAM global en la que un único algoritmo de SLAM centraliza la creación del mapa. En concreto se creaba un mapa de marcas visuales obtenidas por un par estéreo que se iba actualizando con las medidas aportadas por cada uno de los robots. De esta forma, existe un único mapa de marcas visuales global que es compartido por todos los robots. La integración del SLAM con la exploración se realiza teniendo en cuenta la incertidumbre en la localización de cada robot según el algoritmo de SLAM para, en caso de ser esta demasiado elevada, hacer regresar al robot a posiciones previas por donde el robot navegaba con baja incertidumbre.

A la hora de planificar las trayectorias se requiere, además del mapa de marcas visuales, de un mapa de ocupación. Por este motivo se empleaba la localización evaluada por el SLAM visual para determinar un mapa de ocupación con las medidas de un sensor de rango. Dicho mapa de ocupación se utiliza para evaluar una serie de comportamientos reactivos que permiten al equipo de robots explorar el entorno [10]. Este modelo funciona correctamente para explorar las proximidades del robot pero presenta deficiencias para explorar entornos más complejos. En [9] se muestra una arquitectura híbrida que incorpora una planificación de alto nivel para solventar dichas deficiencias.

Esta arquitectura que acabamos de exponer presenta ciertos inconvenientes. Por una parte, es un modelo con un SLAM altamente centralizado y que además es necesario procesar a alta frecuencia dado que es necesario tener un mapa de ocupación actualizado para ser utilizado en el control de movimientos del robot a bajo nivel. Por este motivo, el SLAM era el principal cuello de botella en nuestros trabajos previos [10, 9].

Una alternativa para reducir este problema es emplear un mapa local para el control reactivo con una actualización independiente del SLAM global. Un ejemplo de este tipo de modelo se puede ver en [13]. De esta forma, ampliamos el tiempo de actualización necesario para el SLAM global dado que la planificación de alto nivel puede funcionar con un ciclo de trabajo superior.

El segundo problema es el hecho de realizar un SLAM conjunto con todos robots, lo cual supone un nodo centralizado cuyo coste computacional aumenta además significativamente con el número de robots. En general, hemos venido empleando dos tipos de algoritmos de SLAM: EKF-SLAM [1] y FastSLAM [6] ambos basados en marcas. El tiempo de actualización del EKF es cuadrático con el número de marcas en el mapa y lineal con el número de observaciones y por lo tanto lineal con el número de robots. De esta forma el EKF es más escalable ante el número de robots pero poco escalable en cuanto al tamaño del mapa. Además, tiene el inconveniente de que es muy sensible a la asociación de datos y una mala correspondencia puede hacerle fallar. FastSLAM es más robusto a la asociación de datos y viene a ser el caso opuesto, es más escalable frente al tamaño del mapa, dado que cada marca se actualiza de forma independiente, pero es muy dependiente del número de robots debido a que, para mantener una distribución de partículas que represente de forma precisa la incertidumbre, el número de partículas necesario tiene una dependencia exponencial con el número de robots.

Dados los problemas de escalabilidad y considerando además que un modelo distribuido es más robusto ante fallos, la alternativa es emplear un SLAM independiente en cada robot y emplear un alineamiento y fusión de mapas [3]. El único problema de introducir este modelo es que el modelo de SLAM cambia sensiblemente respecto al SLAM conjunto. En primer lugar, un robot deja de poder emplear las medidas de los otros robots para localizarse. En segundo lugar, no hay un cierre de bucles entre robots, sólo un alineamiento y fusión. Por lo tanto, si hay varias zonas donde los mapas de los robots se solapen el mapa fusionado puede presentar un cierto error. A pesar de ello, consideramos que para muchas aplicaciones esto podría no suponer un factor crítico.

Por último, como se ha comentado en la introducción, también es importante que el sistema funcione bien ante la presencia de elementos dinámicos. El hecho de emplear un mapa local con un ciclo de actualización alto para el control de movimientos a bajo nivel supone que se pueden considerar dichos elementos dinámicos sin problema en cuanto al control de movimientos siempre que la dinámica del entorno no sea tan extrema que bloquee por completo las opciones de navegación del robot. Sin embargo, la presencia de elementos dinámicos puede introducir error en el SLAM global. Un ejemplo de esto es cuando varios robots navegan juntos y uno se localiza respecto del otro en lugar de respecto al entorno estático. Ante este problema se han desarrollado algoritmos de SLAM que separan la parte estática de la dinámica del entorno [17].

3. Arquitectura Desarrollada

3.1. Descripción

La figura 1 muestra la arquitectura que proponemos. Como se puede observar en ella dividimos no solo el control de movimientos en alto y bajo nivel sino también la creación de mapas. A bajo nivel, un mapa de ocupación de reducidas dimensiones y que se mueve con el robot como una ventana deslizante se actualiza con alta frecuencia y es empleado por los comportamientos reactivos. A alto nivel, el planificador trabaja con el mapa fusionado de los mapas creados por algoritmos de SLAM global independientes en cada robot que son recibidos a través de la red.

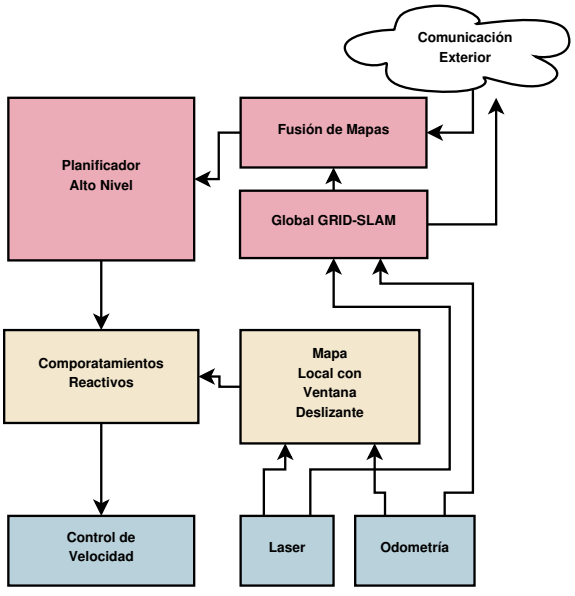


Figura 1: Arquitectura de un robot del equipo.

A continuación se analizan cada uno de estos elementos que componen la arquitectura.

3.2. Comportamientos Reactivos

La capa reactiva de nuestra arquitectura combina varios comportamientos básicos que controlan el proceso de exploración. Tomando como base modelos de comportamientos de otros autores [11], en [10] proponíamos el siguiente conjunto de comportamientos para realizar exploración integrada:

- Ir a fronteras
- Ir a zonas no exploradas
- Evitar obstáculos
- Evitar otros robots
- Ir a pose precisa
- Ir a destino

Mediante el comportamiento de *ir a fronteras* se persigue ir a las celdas de frontera que son aquellas celdas situadas en el límite entre las celdas no ocupadas del mapa de ocupación y las celdas desconocidas. Además, con el comportamiento de *ir a zonas no exploradas* damos preferencia a las celdas de frontera que tienen un mayor número de celdas no exploradas detrás introduciendo así un sentido de utilidad. El comportamiento de *evitar obstáculos* se emplea para evitar colisiones y el de *evitar otros robots* se utiliza para dispersar a los robots en el entorno introduciendo así una coordinación de bajo nivel. El comportamiento de *ir a pose precisa* hace al robot acudir a poses por donde ha pasado anteriormente en las que estaba localizado con baja incertidumbre. El último comportamiento, *ir a destino*, atrae al robot hacia una posición determinada por el planificador de alto nivel.

Cada uno de estos comportamientos se modela con un campo de potencial formado por una suma de Gaussianas. De este modo para el comportamiento i y para cada posición x se corresponde un campo de potencial:

$$P_i(\vec{x}) = A_i \sum_{j \in N_i} \exp\left(-\frac{||\vec{p_j} - \vec{x}||^2}{2\sigma_i^2}\right), \qquad (1)$$

donde A_i y σ_i son los parámetros de amplitud y ancho de la Gaussiana que se ajustan de modo experimental, N_i es el subconjunto de celdas a las que se refiere el comportamiento (por ejemplo, celdas de frontera para *Ir a fronteras* o celdas ocupadas para el comportamiento *Evitar obstáculos*) y donde p_j es la posición de la celda j. Para evitar que al combinar varios comportamientos se formen mínimos locales, solo se incluyen en N_i aquellas celdas que no están situadas detrás de obstáculos,

es decir aquellas que están visibles desde la posición del robot hasta una determinada distancia d_N que se fija como superior al rango del sensor láser $d_N > d_l$ para que de esta forma se incluyan celdas no exploradas como se explica en [9].

A continuación, el campo de potencial global se compone de la siguiente forma:

$$P(\vec{x}) = \sum \mu_i P_i(\vec{x}), \tag{2}$$

donde μ_i son términos binarios que son empleados por el planificador de alto nivel para habilitar o deshabilitar cada comportamiento de forma individual.

Finalmente, los comandos de velocidad para el robot se determinan mediante un controlador que hace al robot seguir la dirección del gradiente del campo de potencial combinado de todos los comportamientos $-\nabla P(\vec{x})$, acudiendo así a las zonas de mínimo potencial.

3.3. Mapa Local con Ventana Deslizante

Para evaluar los comportamientos reactivos se utiliza un mapa local que se mueve con el robot siguiendo un modelo de ventana deslizante. El tamaño de dicho mapa debe ser tal que incluya toda la circunferencia de radio d_N que se emplea en los comportamientos. Además dicho mapa se debe de actualizar a una frecuencia considerable dado que forma parte del bucle de control del robot a bajo nivel y por lo tanto es indispensable para evitar colisiones. Por este motivo la actualización de este mapa es independiente de la del mapa global. Dadas la reducidas dimensiones de este mapa no es necesario un algoritmo de SLAM complejo que permita cerrar bucles. Si la odometría es buena se puede trabajar directamente con esa localización para actualizar este mapa. Cuando la ventana deslizante del mapa se mueve a una zona que ya fue incluida en el mapa global es posible inicializar las celdas que se deben incluir en el mapa como consecuencia de desplazar la ventana a los valores que figuren en el mapa global creado por el SLAM.

Otro factor importante es que el mapa permita actualizar rápidamente a los valores que está recibiendo el sensor. En muchos algoritmos de creación de mapas de ocupación, una vez que la probabilidad de ocupación de una celda converge a ocupada o a libre es muy complicado que cambie su valor. En este sentido, queremos un sistema de actualización del mapa que permita a una celda cambiar su valor con relativa facilidad pero que al mismo tiempo permita filtrar el ruido del sensor. De esta forma el sistema se comportará bien incluso ante situaciones cambiantes en el entorno. Por este motivo, empleamos un modelo de actualización del mapa basado en el cociente entre el número de veces que una celda ha sido detectada como obstáculo por el láser y el total de veces que ha sido escaneada pero considerando un número limitado de medidas.

El algoritmo 1 resume como se actualiza el mapa de ocupación local. Para cada lectura del láser se determinan las celdas por las que pasan cada uno de los rayos del láser. En cada celda se realiza una operación de *limpiar* excepto en la última celda donde el láser alcanza el obstáculo para la cual se realiza la operación de *marcar* (salvo que se exceda el rango del láser, en cuyo caso no hay obstáculo). El valor de ocupación para cada celda es:

$$P_{oc}(\vec{x}) = \frac{\#hits}{\#total}. \tag{3}$$

Algorithm 1 Actualización del mapa local.

```
1: if limpiar then
2:    if total < umbral then
3:       total++
4:    else if hits > 0 then
5:       hits−−
6:    end if
7: else if marcar then
8:    if total < umbral then
9:       total++
10:      hits++
11:   else if hits < umbral then
12:      hits++
13:   end if
14: end if
```

El parámetro *umbral* en el algoritmo nos permite fijar cómo de rápido puede cambiar una celda de estar ocupada a estar libre o viceversa. Si el *umbral* no es excesivamente bajo además también estaremos filtrando el ruido del sensor. Otro dato interesante de utilizar este modelo es que nos permite trabajar con una cantidad de memoria reducida, ya que en general no se requieren más que 2 bytes para cada celda, uno para el contador *hits* y el otro para el *total*, aunque en este caso no es relevante dadas las reducidas dimensiones del mapa.

3.4. Planificador de alto nivel

Para el planificador de alto nivel utilizamos el modelo de subdivisión del entorno en árbol y su posterior análisis propuesto en [9]. Dicho árbol $T\langle nodos, aristas\rangle$ se crea situando el nodo raíz en la posición del robot y consiste en una serie de nodos $N_i(c_i, \rho_i, R_i, \gamma_i)$ y aristas $E_{l,m}(d_{l,m}, N_l, N_m)$. Cada nodo N_i representa una posición c_i en el entorno, el coste total para alcanzar dicho nodo ρ_i, una región asociada al nodo R_i, y el número

de otros robots γ_i presentes en dicha región. Cada arista $E_{l,m}$ representa el camino en línea recta que conecta los nodos N_l y N_m requiriendo viajar una distancia $D_{l,m}$.

Para evaluar este árbol, se da un beneficio $B(N_l)$ a cada nodo hoja del árbol N_l:

$$B(N_l) = \frac{\text{ICells}(R_l)}{\rho_l^2}, \tag{4}$$

siendo $\text{ICells}(R_l)$ una función que cuenta el número de celdas de interés dentro de la región R_l. Si el robot está bien localizado las celdas de interés serán las celdas no exploradas, mientras que si el robot presenta una pobre localización se considerarán como celdas de interés aquellas posiciones por donde el robot ha pasado previamente cuando estaba localizándose con un buen grado de certidumbre. A continuación, la siguiente función recursiva $V(N_b)$ da un valor para cada nodo N_b del árbol:

$$V(N_b) = \begin{cases} \frac{\text{máx}_j\, V(N_j)}{1+\gamma_i} & \text{si } N_b \text{ es rama} \\ B(N_b) & \text{si } N_b \text{ es hoja} \end{cases} \tag{5}$$

siendo j el subconjunto de nodos para los cuales existe una arista $E_{b,j}$ conectando el nodo N_b con el nodo N_j y siendo $b < j$, es decir, cada nodo N_j es un nodo hijo de N_b.

Como la ecuación (5) no considera la posible presencia de otros robots en la región asociada al nodo raíz, los nodos de primer nivel varían sus valores de la siguiente forma:

$$V'(N_f) = \frac{V(N_f)}{d^{\gamma_0}(c_0, c_f)} \prod_{r=0}^{\gamma_0} d(c_0^r, c_f), \tag{6}$$

donde c_0^r es la posición del robot r, y f denota el subconjunto de nodos de primer nivel, es decir los nodos hijos del nodo raíz. De esta forma los nodos que están más lejos de otros robots presentes en la región del nodo raíz aumentan su valor.

Finalmente, el mejor nodo de primer nivel N_g se obtiene como aquel que presenta un valor máximo:

$$N_g = \arg\max_{N_f} V'(N_f). \tag{7}$$

El mejor nodo de primer nivel N_g nos permite decidir en qué estado hacer trabajar a la capa de comportamientos reactivos de bajo nivel. Para ello, nos basamos en el estado de la localización (buena o pobre) y en el tipo de este nodo N_g (nodo hoja o nodo rama). Cuando la localización es buena y el mejor nodo es de tipo rama, se activará el comportamiento de *ir a destino* indicando como objetivo la celda c_g asociada a dicho nodo, además del comportamiento *evitar obstáculos* para evitar colisiones. Sin embargo, en caso de ser el mejor nodo de tipo rama, se utiliza una combinación de comportamientos para explorar el entorno local activando los comportamientos: *ir a fronteras*, *ir a zonas no exploradas*, *evitar otros robots* y *evitar obstáculos*. Para acabar, falta contemplar el caso de que el robot esté pobremente localizado. En este caso, si el mejor nodo es de tipo rama, se activará de nuevo la combinación de *evitar obstáculos* junto con *ir a destino* indicando como objetivo la celda c_g asociada a dicho nodo. Por otro lado, si el mejor nodo fuera de tipo rama, combinaríamos los comportamientos de *ir a poses precisas* con *evitar obstáculos*.

3.5. Grid-SLAM Global

A la hora de realizar el SLAM global que debe llevar a cabo cada robot de forma independiente hemos encontrado problemas dado que a nuestro saber no hay algoritmos de código abierto que estén diseñados para operar en entornos dinámicos. Hemos realizado pruebas empleando el software gmapping [7] consistente en un filtro de partículas Rao-Blackwellized con los datos de láser que emplea la observación a la hora de realizar cada nuevo muestreo para hacer un mejor uso de las partículas y de esta forma obtener buenos resultados con un número de partículas bajo. Dicha técnica funciona muy bien en entornos estáticos, sin embargo, presenta un error considerable en entornos dinámicos. Al situar a dos robots en posiciones próximas, el láser de uno de los robots puede incidir sobre el segundo robot. Esto produce que el primer robot se localice con respecto al segundo, y al moverse el segundo se introduce un error en la localización del primero y por lo tanto en el mapa que está creando, dado que gmapping asume un entorno estático.

De esta forma, esto queda como un punto a mejorar en el sistema. Algunas de las propuestas al respecto son [17, 14, 16].

3.6. Fusión de Mapas

El siguiente paso es poder alinear y fusionar los mapas de ocupación que crea cada uno de los robots de forma independiente. Para el alineamiento es interesante que los robots empiecen la exploración en posiciones cercanas. El motivo de esto es que es imprescindible que exista una parte en común entre los mapas creados por los diferentes robots para poder realizar el alineamiento. Si el alineamiento es correcto desde el principio del proceso de exploración, los robots podrán coordinarse de una forma más eficiente.

Para realizar dicho alineamiento, elegimos una serie de puntos característicos del mapa, a continua-

ción haremos la correspondencia entre los puntos de los distintos mapas y evaluamos la traslación y rotación entre dichos puntos que será empleada para poder fusionar los mapas.

A la hora extraer los puntos característicos del mapa, no funcionan bien los típicos detectores de puntos como SIFT o SURF [5] aplicados directamente sobre el mapa de ocupación. En su lugar, empleamos la transformada distancia del mapa de ocupación en la que cada píxel toma como valor la distancia mínima a la celda ocupada o desconocida más cercana. Aplicando el detector de características SIFT [12] sobre la transformada distancia obtenemos unos puntos muy robustos que permiten hacer fácilmente la correspondencia y posterior alineamiento.

La Figura 2 muestra este problema. Como se puede observar, en ella se muestran los mapas que están siendo creados de forma independiente por dos robots durante el proceso de exploración del entorno. En la parte superior se muestran los puntos extraídos por SIFT en los mapas correspondientes a cada robot. Como se puede ver, SIFT extrae muchos puntos que son bastante ruidosos y sobre los que es difícil encontrar correspondencias de forma fiable. En la parte inferior se ha calculado la transformada distancia y aplicado SIFT sobre la imagen resultante. Se puede observar que SIFT ahora extrae un número menor de puntos pero estos son mucho más robustos y es sencillo encontrar correspondencias entre ambas imágenes.

Una vez obtenidas las características y evaluados sus descriptores SIFT aplicamos una técnica de alineamiento como las presentadas en [2] y procedemos a fusionar los mapas empleando la transformación entre mapas obtenida en el alineamiento.

4. Simulación

Hemos realizado experimentos sobre el simulador Stage [4] implementando cada uno de los módulos de la arquitectura descrita sobre ROS [15]. El sistema funciona en tiempo real. El mapa local con ventana deslizante se actualiza a 10Hz y el control reactivo da comandos de velocidad a 7Hz. El grid SLAM actualiza un mapa de 60x60 metros con una resolución 0.05 cm cada 2 segundos mientras que el tiempo para fusión de mapas es de unos 14-15 segundos. Este tiempo se puede reducir a la mitad empleando SURF en lugar de SIFT pero hemos observado que SIFT ofrece mejores resultados. El planificador de alto nivel puede funcionar con un ciclo de hasta unos 20-25 segundos aproximadamente en función de la complejidad del mapa explorado hasta el momento.

A continuación mostramos un ejemplo de los mapas individuales creados por dos robots y el mapa fusionado empleando la arquitectura propuesta. Como se puede observar cada robot explora una parte distinta del entorno. A pesar de tener sólo una pequeña parte en común los dos mapas, estos se alinean y se crea el mapa global fusionando ambos.

Sin embargo, dado que pueden existir errores o incertidumbre en el SLAM, se puede apreciar como el mapa global fusionado presenta un cierto error en la esquina inferior izquierda donde como se puede observar no coinciden los mapas de ambos robots. La mayoría de los puntos encontrados en común entre ambos mapas pertenecen a la parte de la derecha del entorno donde los robots comenzaron la exploración. Dicha región se encuentra perfectamente alineada, sin embargo, en la parte opuesta se produce el error a la hora de cerrar el bucle que forma el pasillo principal.

En este sentido, podemos ver que el alineamiento entre los mapas aporta una cierta información que no está siendo considerada por el SLAM ya que este trabaja de manera independiente.

5. Conclusiones

Hemos mostrado una arquitectura en la un equipo de robots explora un entorno de forma totalmente distribuida. Este sistema integra trabajos realizados previamente e incluye como novedad una forma robusta de obtener el alineamiento de mapas de ocupación.

Como trabajos futuros hemos visto que habría que mejorar el SLAM de alto nivel en dos sentidos. En primer lugar, considerar que pueden existir elementos dinámicos en el entorno, y en segundo lugar incluir la información de alineamiento con el mapa del otro robot para reducir la incertidumbre del mapa.

Agradecimientos

Este trabajo ha sido apoyado por el Gobierno de España (Ministerio de Ciencia e Innovación) a través del proyecto DPI2010-15308, "Exploración Integrada de Entornos mediante Robots Cooperativos para la Creación de Mapas 3D Visuales y Topológicos que puedan ser usados en Navegación con 6 Grados de Libertad".

Referencias

[1] T Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, 2002.

[2] Monica Ballesta, Arturo Gil, Oscar Reinoso, Miguel Juliá, and Luis M. Jiménez.

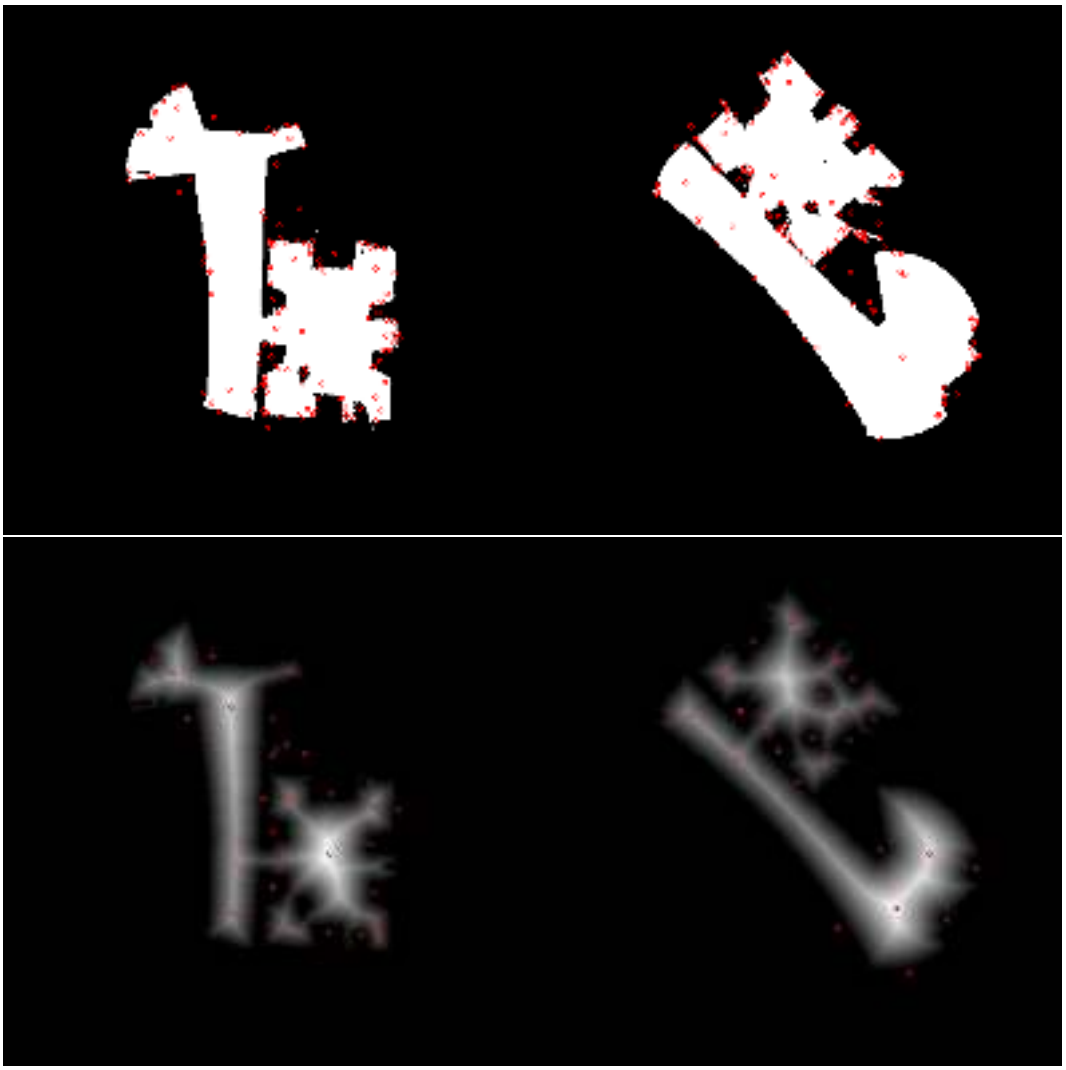


Figura 2: Detección de características. Sobre el mapa de ocupación en la parte superior y sobre la trasformada distancia en la parte inferior para los mapas creados por dos robots (derecha e izquierda).

Multi-robot map alignment in visual SLAM. *WSEAS Transactions on systems*, 9(2):213–222, 2010.

[3] D. Fox, J. Ko, B. Limketkai, D. Schulz, and B. Stewart. Distributed multi-robot exploration and mapping. *Proc. of the IEEE*, 94(7):1325–1339, 2006.

[4] B. Gerkey, R. Vaughan, and A. Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Int. Conf. on Advanced Robotics (ICAR 2003)*, 2003.

[5] A. Gil, O.M. Mozos, M. Ballesta, and O. Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. *Machine Vision and Applications*, 21(6), 2010.

[6] A. Gil, O. Reinoso, M. Ballesta, M. Juliá, and L. Payá. Multi-Robot Visual SLAM using a Rao-Blackwellized Particle Filter. *Robotics and Autonomous Systems*, 58(1):68–80, 2010.

[7] G. Grisetti, C. Stachniss, and W. Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *Robotics, IEEE Transactions on*, 23:34–46, 2007.

[8] M. Juliá, A. Gil, and O. Reinoso. A Comparison of Path Planning Strategies for Autonomous Exploration and Mapping of Unknown Environments. *Autonomous Robots*, 33(4):427–444, 2012.

[9] M. Juliá, O. Reinoso, A. Gil, M. Ballesta, and L. Payá. A hybrid solution to the multi-robot integrated exploration problem. *Engineering Applications of Artificial Intelligence*, 23(4):473–486, 2010.

[10] M. Juliá, O. Reinoso, A. Gil, M. Ballesta, and L. Payá. Behaviour Based Multi-Robot Integrated Exploration. *Int. Journal of Inno-*



Figura 3: Fusión de mapas. Los dos mapas superiores corresponden con los mapas creados por dos robots, en la parte inferior se muestran estos dos mapas fusionados.

vative Computing, Information and Control, 7(9), September 2011.

[11] H. Lau. Behavioural Approach for Multi-Robot Exploration. In *Proc. of the Australasian Conf. on Robotics and Automation (ACRA'03)*, Brisbane, Australia, 2003.

[12] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[13] Eitan Marder-Eppstein, Eric Berger, Tully Foote, Brian P. Gerkey, and Kurt Konolige. The Office Marathon: Robust navigation in an indoor office environment. In *ICRA*, pages 300–307. IEEE, 2010.

[14] N.C. Mitsou and C.S. Tzafestas. Temporal occupancy grid for mobile robot dynamic environment mapping. In *Control Automation, 2007. MED '07. Mediterranean Conference on*, pages 1–8, 2007.

[15] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In *International Conference on Robotics and Automation*, 2009.

[16] Jari Saarinen, Henrik Andreasson, and Achim J. Lilienthal. Independent Markov chain occupancy grid maps for representation of dynamic environment. In *IROS*, pages 3489–3495. IEEE, 2012.

[17] Denis F. Wolf and Gaurav S. Sukhatme. Online Simultaneous Localization and Mapping in Dynamic Environments. In *ICRA*, pages 1301–1307. IEEE, 2004.

DISEÑO DE UN DISPOSITIVO HÁPTICO DE BAJA RESOLUCIÓN BASADO EN SMA PARA APLICACIÓN EN TELEROBÓTICA

Francisco Javier Tormo
fratorg7@epsa.upv.es

Jaime Masiá
jmasia@mcm.upv.es

Juan Ramón Rufino
jrufino@eln.upv.es

José Ferri
joferse2@epsa.upv.es

Borja Mansilla
bormanlv@epsa.upv.es

Universidad Politécnica de Valencia, Campus De Alcoi. Plaza Ferrándiz y Carbonell s/n, 03801 Alcoy (Alicante)

Resumen

Los materiales inteligentes, activos, o también denominados multifuncionales son materiales capaces de responder de modo reversible y controlable ante diferentes estímulos físicos o químicos externos, modificando alguna de sus propiedades.

El objetivo del presente artículo es el diseño de un dispositivo háptico portátil para obtener una realimentación táctil mínima con la cual el operario pueda conocer de la posición de la pieza en el interior de una pinza conectada al extremo de un manipulador industrial.

El sistema de actuación está basado en una aleación níquel-titanio comercial llamada Flexinol.

Palabras Clave: SMA, Flexinol, actuador, dispositivos hápticos portatiles.

1 ANTECEDENTES

1.1 HISTORIA DE LAS SMA

Los primeros pasos en el descubrimiento del efecto memoria de forma fueron, según Miyazaki y Otsuka [MIY, 1989] en los años 1930, cuando A.Ölander descubrió el efecto superelástico en una aleación Au-Cd en 1932 y, más tarde, Greninger y Mooradian observaron la aparición y desaparición de fase martensita a medida que crecía y decrecía la temperatura en una aleación de Cu-Zn. Los fundamentos del efecto memoria gobernado por el comportamiento termoelástico de la martensita fue extensamente explicado una década más tarde por Kurdjumov y Khandros y también por Chang y Read.

Pese a estos descubrimientos, el más importante se realizó en 1962, cuando Buehler y sus colaboradores, en el Naval Ordnance Laboratory (ahora Naval SurfaceW arfare Center),desarrollaron una aleación de níquel y titanio (NiTi) con un efecto memoria de forma aún más acusado que el descubierto con el AuCd y denominaron el nuevo material como Nitinol (derivado de Nickel Titanium Naval Ordnance Laboratory).

El descubrimiento de Buehler abrió una gran puerta a la investigación en la búsqueda de nuevas aleaciones con capacidades de memoria de forma así como al estudio de su comportamiento mecánico y sus posibles aplicaciones. Pero, pese a todos los avances realizados en el descubrimiento de nuevas aleaciones metálicas con estas propiedades, el Nitinol sigue siendo la aleación más comercial y más extendida debido a su buena estabilidad al ciclado, biocompatibilidad, resistencia a la corrosión y capacidad de recuperar grandes deformaciones [2].

1.2 CARACTERÍSTICAS

Las aleaciones con memoria de forma (abreviado como SMA "Shape Memory Alloy") son aleaciones metálicas que, después de una deformación aparentemente plástica, vuelven a su forma original tras un calentamiento. Los mismos materiales, dentro de un determinado rango de temperaturas, pueden ser deformados hasta casi un 10% volviendo a recuperar su forma original al ser descargados.

Estos inusuales efectos son llamados memoria de forma térmica (o efecto memoria de forma) y memoria de forma elástica (o Superelasticidad) respectivamente. Ambos efectos son debidos a un cambio de fase llamada transformación martensítica termoelástica.

Dado que las SMA responden de una forma peculiar a los cambios de temperatura y tensión, han sido clasificados como "materiales inteligentes ("Smart materials"). Las aplicaciones potenciales de estos dos principales comportamientos son enormes, pudiendo ser usados para generar fuerza o movimiento (mediante el efecto memoria de forma) o almacenar energía (superelasticidad).

A continuación brevemente comentaremos el movimiento atómico que provoca el movimiento a escala macroscópica [1].

1.3 RAZÓN CRISTALOGRÁFICA

Al enfriarse, la fase de austenita de alta temperatura, con una red cúbica centrada en la cara, se convierte en la fase martensítica de baja temperatura. Debido a los esfuerzos experimentados durante el enfriamiento, la martensita producida a partir de la austenita experimenta un proceso de maclado de cristales: la formación de capas adyacentes relacionadas por simetría especular. La deformación remueve el maclado. La martensita demaclada posee una red cristalina tetragonal. El calentamiento de la martensita demaclada deformada la convierte nuevamente a la fase de austenita.

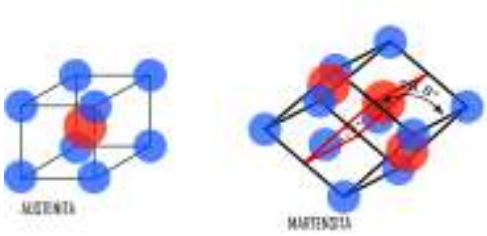


Figura 1: Mecanismo del efecto memoria de forma.

1.4 ALEACIONES NiTi

Las aleaciones NiTi tienen mayor capacidad de memoria (hasta un 8% mientras que sólo se alcanza un 4- 5% en las de base Cu), son mucho más estables térmicamente, tienen una excelente resistencia a la corrosión comparadas con las de cobre, tienen mayor resistividad eléctrica (siendo pues su activación eléctrica más simple), pueden ser aleadas y extruidas con facilidad y tienen un mayor rango de posibles temperaturas de transformación.

2 INTRODUCCIÓN

En el camino hacia la interacción hombre maquina nos encontramos con multitud de dispositivos que hacen que esta sea lo más natural posible, entre ellos podemos encontrar desde pantallas táctiles[5, 8, 11], dispositivos hápticos portátiles [4, 9, 12] o interfaces para teleoperación [6, 10], todos ellos con un denominador en común, la utilización de materiales inteligentes como mecanismo actuador, ya sean poliméricos o aleaciones, con lo que conseguimos una reducción considerable del dispositivo gracias a la facilidad de transformación e implementación de estos materiales. Lo cual apunta a una mayor inmersión del operario en el sistema.



Figura 2: Prototipo del dispositivo montado.

Se ha estudiado el caso para un dispositivo, ya que se puede considerar simétrico el sistema completo.

El presente trabajo se enmarca dentro de un proyecto mayor de teleoperación mediante visión artificial, donde el principal objetivo es depender del menor número de periféricos posible.

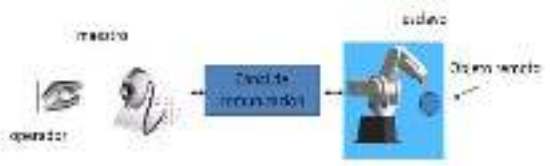


Figura 3: Plataforma de teleoperación

3 METODOLOGÍA

Partimos de un trabajo anterior donde se realizaron pruebas para distintos espesores de la aleación, escogiendo el cable con menor consumo y respuesta aceptable, siendo este el de 100 µm. Para el dispositivo diseñado se tendrá una longitud de 3cm.

Se utilizará un cable de la marca comercial Flexinol, una plataforma de harware libre Arduino [3], sensores de presión comerciales y una fuente de alimentación de PC para alimentar a una placa de potencia que proporciona la corriente necesaria.

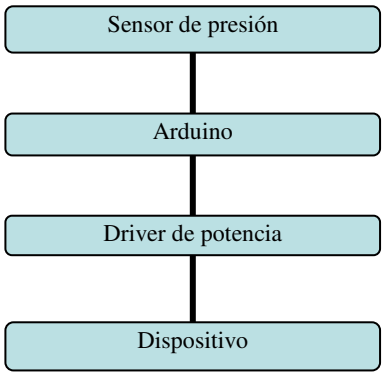


Figura 4: Diagrama de bloques

En la plataforma de teleoperación real con un manipulador y pinza con sensores resistivos, se tendría que la función del arduino, la realizaría el autómata o la unidad lógica encargada de comunicar el manipulador con el PC (esclavo) con la configuración típica maestro-esclavo, a su vez el dispositivo háptico estaría conectado al PC (maestro) con una cámara para realizar la teleoperación mediante visión artificial.

3.1 ACTIVACIÓN

Se detecta la presión en los sensores conectados al Arduino, estos activaran la interrupción externa del microcontrolador Atmega168, que genera la señal desde el timer interno. Esta señal es de tipo PWM, activará la etapa de potencia permitiendo que aumente la ganancia para proporcionar una respuesta en contracción y consumo optimas [1].

En el caso del artículo se requiere activar el material para que desplace la parte de del dispositivo que está en contacto con el dedo del usuario pudiendo sentir la diferencia de presión entre una parte del dedo y otra, emulando la sensación que tendría la pinza del manipulador al coger un objeto de manera parcial, con lo que el material realizará una contracción y se mantendrán en ese estado todos los cables Ni-Ti que coincidan con la posición de los sensores en la pinza.

Contamos con 6 sensores resistivos en con una disposición de 3 X 2 quedando dos columnas (izquierda-derecha) y tres filas (punta-intermedio-base) pudiendo codificar esto en binario con lo que tenemos el bit de menor peso en la base del dedo y el de mayor peso en la punta.

Estos bits se pasan en paralelo a la placa de potencia para permitir el paso de corriente de la señal PWM.

3.1.1 Señal PWM

Para activar los SMA se requiere cambiar su temperatura interna hasta llegar al cambio de fase durante este cambio muchas de las propiedades físicas del material varían. Se intenta llevar a la fase de austenita lo antes posible para así evitar no linealidades y retardos por la histéresis.

Para este caso concreto se tiene que la mejor manera de activar el flexinol será haciendo pasar una corriente por su interior.

Para mejorar la distribución del calentamiento a causa de la resistencia del flexinol se aplicarán los pulsos generados desde el Atmel16 como muestra en la siguiente figura. Ya que aplicando pulsos cortos y no corriente continua se mejora la respuesta y vida del material [7].

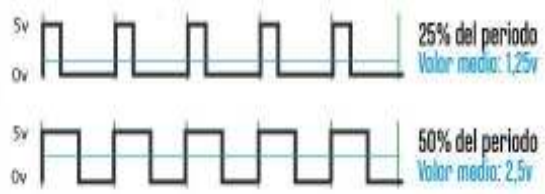


Figura 5: Distintos pulsos para un mismo periodo [ms].

3.1.2 Driver de potencia

Se tiene la señal PWM de amplitud 5V como control para la alimentación del cable flex100 como la alimentación deseada para el experimento son 12V y la corriente aportada por el arduino no son suficientes se requiere de un driver o etapa de potencia que amplifica la señal PWM de 5 a 12 voltios y regula la corriente aportada a la carga. Trasladando así el consumo de potencia a este circuito y salvaguardando el arduino incapaz de suministrar la corriente deseada por sí mismo.

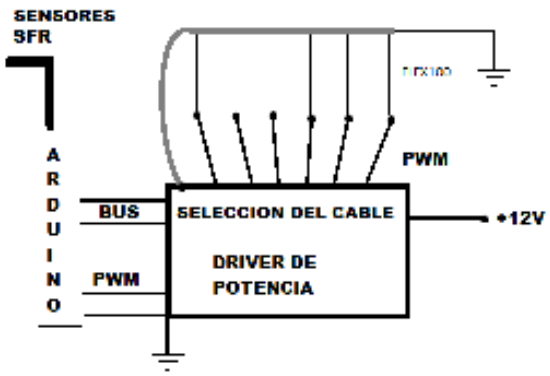


Figura 6: Driver de potencia.

La tensión de +5V se obtiene de un regulador de tensión integrado en la placa y los +12V se obtienen de una fuente de PC. De realizarse una versión en base a estos experimentos se adecuaría el transformador de potencia y la electrónica de control en una PCB que permitiera la portabilidad del dispositivo.

4 DISEÑO MECÁNICO

Se propone el diseño de un actuador mecánico en el que una aleación Ni-Ti (Flexinol) genera el movimiento de las plataformas que representaran la posición del extremo del manipulador sobre la pieza, sabiendo si esta siendo bien sujetada o si por lo contrario puede tener el riesgo de perderse durante la trayectoria, consiguiendo así una respuesta táctil directamente proporcional al extremo del manipulador.

El dispositivo fue implementado mediante una impresora 3D por su simpleza en el diseño, lo que nos concedía una alta velocidad a la hora de rehacer alguna pieza del prototipo y la ligereza obtenida, alrededor 16g.

La compresión del hilo se consigue por el calentamiento generado mediante el paso de pulsos de corriente y la recuperación mediante un muelle a compresión a medida que se enfría el material.

Hemos considerado este dispositivo de baja resolución debido a que realizamos la representación táctil con una matriz de pivotes de 3x2, mediante la unión de las filas como se muestra en la figura.

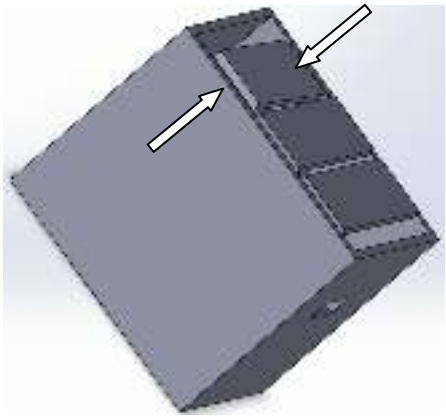


Figura 7: Direcciones de las plataformas.

Para este dispositivo hemos hecho uso de un cable de 100 µm de sección y 3cm de longitud. Obteniendo una deformación del 5% siendo esta la mínima para mover la plataforma y obtener así una representación táctil suficiente para que el usuario perciba la posición del objeto en el extremo del manipulador.

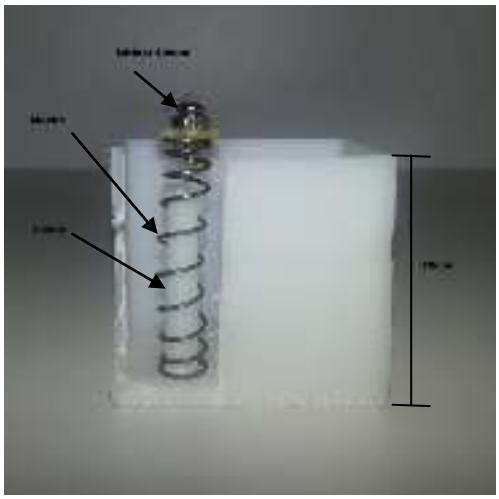


Figura 8: Medidas.

5 CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

La respuesta del dispositivo para un sistema de teleoperación considerado de baja resolución como es el caso que nos ocupa, una pinza simple, es satisfactoria.

Para sistemas de mayor resolución este mismo dispositivo puede cumplir su cometido ya que el aumento de resolución de este no entraña ninguna dificultad, simplemente con replicar el mecanismo de actuación podemos obtener hasta matrices de pivotes de 5 × 5, todo esto sin variar el tamaño del mecanismo actuador.

Por otro lado el aumento de temperatura al mantener la contracción nos hace plantearnos mejorar control del ciclo de trabajo variándolo según varíe la temperatura manteniendo esta, en punto de trabajo.

Los nuevos planteamientos están orientados al trabajo con polímeros conductores y con polímeros con memoria de forma, debido a su facilidad de manipulación e implementación, que junto a las tecnologías existentes de prototipado rápido como las impresoras 3D, el desarrollo de este tipo de dispositivos con diferentes morfologías y la posibilidad de ser adaptadas a diferentes casos pude verse favorecidas y tener un desarrollo mas elevado.

Agradecimientos

Agradecer al departamento de Ingeniería Electrónica y al departamento de Ingeniería Mecánica y Materiales de la Universitat Politècnica de València campus d'Alcoi por el acceso a los laboratorios y el apoyo recibido.

Referencias

[1] Cano Sánchez A., (2010) "Estudio e implementación de actuadores basados en aleaciones SMA", Universidad Carlos III de Madrid.

[2] De La Flor López, Sílvia "Simulación numérica y correlación experimental de las propiedades mecánicas en las aleaciones con memoria de forma.".UniversitatPolitècnica de Catalunya. Departament de Resistència de Materials i Estructures a l'Enginyeria.

[3] http://arduino.cc/es/Main/Software.

[4] Ig Mo K., Kwangmok J., Ja Choon K., Jae-Do N., Young Kwan L., and Hyouk Ryeol C., (2008) "Development of Soft-Actuator-Based Wearable Tactile Display" IEEE Transactions on Robotics, Vol. 24, No. 3.

[5] Nakatani M., Kajimoto H., Sekiguchi D., Kawakami N., and Tachi S., "3D Form Display with Shape Memory Alloy" The University of Tokyo, Japan

[6] Kobayashi F., Ikai G., Fukui W., and Kojima F., Research Article Two-Fingered Haptic Device for Robot Hand Teleoperation. Hindawi Publishing Corporation Journal of Robotics Volume 2011, Article ID 419465, 8 pages.

[7] Manual de referencia Atmega168

[8] Matsumgal[1] T., Makishi[1] W., Totsu[2] K., Esashi[2] M. and Haga[1] Y.. "2-D and 3-D Tactile Pin Display Using SMA Micro-coil Actuator and Magnetic Latch"; Tohoku University Biomedical Engineering Research Organization (TUBERO), Japan. Graduate School of Engineering, Tohoku University, Japan.

[9] Minamizawa K., Fukamachi S., Kajimoto H., Kawakami N., Tachi S.,"Gravity Grabber: Wearable Haptic Display to present Virtual Mass Sensation", The University of Tokyo and The University of Electro-Communications.

[10] Shimoga K.B. y Khosla P.K., "Touch and Force Reflection for Telepresence Surgery", Center for Medical Robotics and Computer-

Aided Surgery The Robotics Institute, Carniegie Mellon University.

[11] Taylor P.M., Moser A., Creed A., (1998) “A sixty-four element tactile display using shape memory alloy wires”

[12] Wang H., Kaleas D., Ruuspakka R., Tartz R., (2012) “Haptics using a smart material for eyes free interaction in mobile devices”, IEEE Haptics Symposium 2012 4-7 March, Vancouver, BC, Canada

CÁLCULO AUTOMÁTICO DE SECUENCIAS DE ENSAMBLADO BASADO EN UNA TÉCNICA DE AGRUPACIÓN PARA LA CONSTRUCCIÓN DE ESTRUCTURAS MEDIANTE EQUIPOS DE ROBOTS

Álvaro Sempere, Ivan Maza y Aníbal Ollero
Grupo de Robótica, Visión y Control, Universidad de Sevilla
sempere1987@gmail.com,imaza@us.es,aollero@us.es

Resumen

Este trabajo aborda el cálculo automático de la secuencia de ensamblado que permite construir una determinada estructura tipo celosia a partir del análisis geométrico 3D de la misma. Dicho cálculo constituye el primer bloque del módulo de planificación de un equipo de robots aéreos equipados con brazos manipuladores al cargo de la construcción de la estructura en lugares de difícil acceso. En el cálculo de la secuencia de ensamblado se ha aplicado la técnica conocida en la literatura como "assembly-by-disassembly" a partir de los grafos NDBG (Non-Directional Blocking Graphs) obtenidos en el análisis geométrico de la estructura. En el artículo se presenta una heurística novedosa basada en agrupar piezas de cara a obtener un plan de construcción lo más paralelizable posible.

Palabras clave: ensamblado automático, manipulación aérea, planificación automática, grafos NDBG

1. INTRODUCCIÓN

El trabajo que se presenta en este artículo se enmarca dentro del proyecto europeo ARCAS financiado por la Comisión Europea. Uno de los objetivos de dicho proyecto es la construcción de estructuras tipo celosia (ver figura 1) usando un conjunto de robots aéreos equipados con brazos manipuladores. La motivación de usar estos robots viene del interés práctico de realizar estos montajes en lugares de muy difícil acceso por medios convencionales.



Figura 1: Estructura tipo celosia correspondiente a una torre

En diversos artículos se han abordado distintas técnicas para la obtención de la secuencia de ensamblado de piezas complejas. Este artículo se centra en un criterio importante: el cálculo de la secuencia que permite aumentar el grado de paralelización en su construcción. Por ello nuestro objetivo es calcular una secuencia de construcción factible, que necesite el mínimo número de pasos para su construcción completa.

2. TRABAJOS RELACIONADOS

El problema concreto abordado en este trabajo es el de obtener la secuencia de operaciones de ensamblado, en adelante OE, para una estructura dada. En [9] podemos encontrar una clasificación de los tipos de estructuras enumerando una serie de características intrínsecas de cada una de ellas: **secuencialidad**; **monotonicidad**, la necesidad de realizar desplazamientos de las piezas a unas posiciones intermedias; y **coherencia**, cada pieza conectada hace contacto con las ya colocadas previamente. Las estructuras abordadas en este trabajo son **secuenciales para dos robots**, **monótonas** y **coherentes**. A continuación se revisan los trabajos de la literatura que abordan el análisis geométrico de la estructura, el modo de representar las secuencias de ensamblado y los algoritmos para el cálculo de dichas secuencias.

2.1. Análisis geométrico

Varios autores han introducido ideas como el *grafo de enlaces* para determinar los contactos existentes entre piezas. En la referencia [3] se especifican las uniones existentes entre piezas y el tipo concreto, mientras que las referencias [14, 5] se representan dichos grafos como matrices de contactos en determinados vectores. También hay especificaciones más concretas [16] que indican las posiciones de las piezas mediante predicados (pieza 1 encima de pieza 2, pieza 2 en $+x$ con pieza 5, etc.). Se han aplicado *grafos relacionales* para analizar las distancias entre piezas desde sus vértices y/o aristas [10] y se empleado el *espacio de configuración* o *C-space* en [6, 7] para definir un área geométrica que indica para cada dos piezas la región de

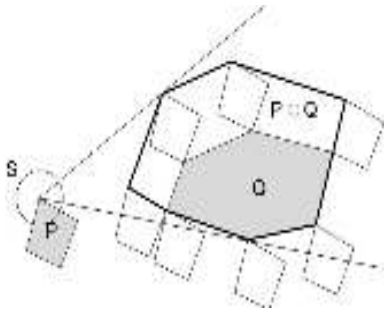


Figura 2: Construcción de la región $P \ominus Q$ desde un vértice de P. Como puede observarse esta región crea dos líneas desde el vértice origen indicando el ángulo de sección de S que **no** puede cruzarse, esto es, siempre que se traslade P hacia un $\vec{v}$ fuera de la sección, no existirá colisión

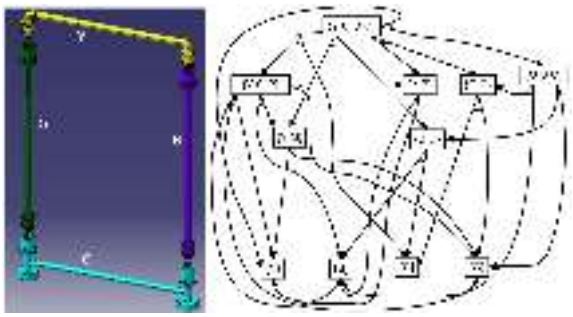


Figura 3: Representación de un grafo AND/OR con todas las secuencias posibles para el ensamblado de un pórtico

libre contacto entre ellas. En particular en [6] se hace alusión a la operación $P \ominus Q$, la *diferencia de Minkowski* (ver figura 2), determinando así la zona libre de obstáculo entre dos polígonos P y Q.

2.2. Representación de secuencias de ensamblado

En éste ámbito existen un cierto número de trabajos que van desde la creación de técnicas base claramente diferenciables hasta las constantes adaptaciones realizadas a partir de estas. Una primera categoría podría ser la técnica que hace uso del conocido grafo AND/OR, presentada en [8] y que ha sido citado en numerosas ocasiones en el ámbito de la *secuenciación*. Se calcula la combinatoria total y se representa en un grafo AND/OR, en el cual las ramas AND (bifurcaciones con el símbolo del ángulo en la figura 3) indican descomposición en una única pieza o en un subconjunto, mientras que las OR (distintos caminos dobles desde un mismo nodo) permite diversificar el camino de desmontaje presentando varias posibilidades.

Por otro lado en [6] se analiza un algoritmo basado en el *motion space* estableciendo las traslaciones que hay que seguir para ensamblar la estructura usando trayectorias comprendidas en las áreas del $C - space$. También es posible estudiar la viabilidad de las secuencias a partir de bases de conocimiento [3], por geometría [14], con algoritmos adhoc [18] diferenciando entre conectores y piezas, o desde la construcción de grafos de intersección NDBG (*Non-Directional Blocking Graph*) [17].

2.3. Métodos para el cálculo de las secuencias de ensamblado

Un método ampliamente usado es el denominado **assembly-by-disassembly**. Se parte de la estructura inicial, se van obteniendo las distintas piezas o subconjuntos que pueden ser desconectados (actualizándose la estructura en cada paso) y se vuelve a repetir el proceso hasta que queda completamente desconectada. Esta técnica puede aplicarse sobre la representación basada en los grafos de intersección NDBG mencionados anteriormente, donde a partir de los bloqueos entre piezas se pueden obtener las distintas desconexiones (también conocidas como *descomposiciones*) a lo largo del desmontaje completo.

Hay otros algoritmos basados en colonias de hormigas [16, 15, 5], uso de redes neuronales [13], técnicas evolutivas [4, 11] y sistemas inmunitarios [2]. Otra técnica interesante es la de [12], que a partir del *grafo de enlaces* transformado en una matriz de uniones, calcula una secuencia factible eliminando sucesivamente las columnas de las partes ya colocadas en el ensamblado y examinando las filas para otras uniones candidatas.

Todos estos algoritmos pueden ser ajustados para obtener la secuencia factible desde el punto de vista geométrico o para la optimización en base a una serie de características como el número de reorientaciones [16], herramientas usadas, número de piezas movidas, etc. También hay algoritmos basados en redes de Petri que sirven para evaluar el grado de dificultad de una secuencia, pudiendo obtener la óptima [1].

3. SOLUCIÓN ADOPTADA

El esquema de diseño que se ha seguido para resolver el problema se presenta en la figura 4 y se basa en los siguientes módulos: cálculo del NDBG, aplicación de la técnica *assembly-by-disassembly*, heurísticas de selección, selección de la secuencia óptima y construcción final. Dentro de esta sección se describen los distintos módulos.

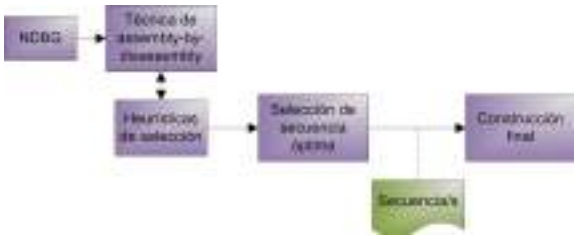


Figura 4: Esquema propuesto para la obtención de la secuencia de ensamblado

3.1. Cálculo del grafo NDBG

Sea S una estructura, $s_i \subseteq S$ un subconjunto de piezas y g_S el grafo DBG, *Directional Blocking Graph* de S en una dirección $\vec{t}$, definimos $g_S(\vec{t})$ como un grafo dirigido $G = (V, E)$ con:

$$\begin{aligned} V &\neq 0 \\ E &\subseteq \{(a,b) \in V \times V : a \neq b\} \ , \end{aligned} \tag{1}$$

donde (a, b) es una arista con origen en a y destino en b, y el conjunto V son los vértices, cumpliendo que el nodo $n_i \in V$ se identifica con la pieza P_i de la estructura S.

Si una pieza $P \in S$ colisiona con otra $Q \in S$ a lo largo de $\vec{t}$ entonces $(P, Q) \in g_S(\vec{t})$, siguiendo la definición de la expresión (1). El conjunto de todos los $g_S(\vec{v})$ para cualquier dirección $\vec{v}$ formará el grafo $NDBG$, en adelante G_S.

Sea $w = (n_1, n_2, ..., n_k)$ el camino de nodos comenzando en n_1 y terminando en n_k. Se define un *conjunto fuertemente conectado* CFC de $g_S(\vec{v})$ como aquel subconjunto maximal de nodos tal que

$$\begin{aligned} \forall n_j \in g_S(\vec{v}) \wedge \forall n_k \in g_S(\vec{v}) \quad &\rightarrow \\ \exists w = (n_j, n_{j+1}, ..., n_{k-1}, n_k) \wedge & \\ (n_{j+1}, ..., n_{k-1}) \in g_S(\vec{v}) \ . & \end{aligned} \tag{2}$$

Es decir para cualquier par de nodos $(n_1, n_2) \in g_S(\vec{v})$ existe un camino de n_1 a n_2 y además todos los nodos del camino pertenecen a $g_S(\vec{v})$. Sabiendo esto, los *conjuntos fuertemente conectados* de un grafo $g_S(\vec{v})$ se traducen a subconjuntos de piezas o simplemente piezas únicas que pueden ser desconectados de S.

Sea $d_{\vec{t}}$ una descomposición o un CFC representado como una lista de piezas que pueden ser desconectadas hacia una dirección $\vec{t}$, y sea D el conjunto desconectable que incluye todas las descomposiciones posibles, se dice que un conjunto de n piezas pertenece a D si

$$\left\{n|(n,p_i)/\forall p_i \in S \wedge (n,p_i) \notin g_S(\vec{t_j})\right\} \in d_{\vec{t_j}} \ . \tag{3}$$

En la figura 5 se observa un ejemplo que ilustra los distintos $g_S(\vec{v})$ que componen el conjunto G_S. Cada uno está asociado a una dirección concreta, de tal manera que el espacio posible es una circunferencia S^2. Por el contrario, para una estructura en tres dimensiones el espacio de direcciones sería una esfera S^3 tal que

$$\begin{aligned} G_S &= \bigcup_{k=1}^{k=n} g_S(\vec{t_k}) \\ \forall \vec{t_k} \in S^j, n &= \left|\vec{d}\right| \ . \end{aligned} \tag{4}$$

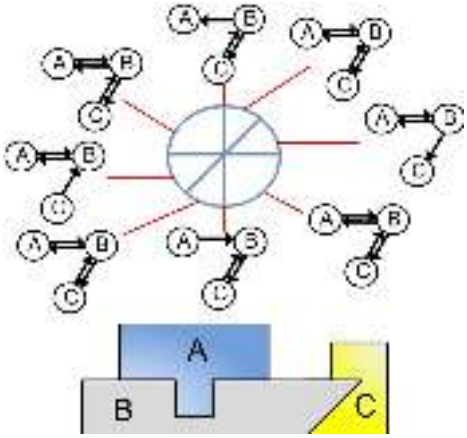


Figura 5: Representación del conjunto G_S completo para una estructura. Se pueden ver los distintos grafos $g_S(\vec{v})$ agrupados por secciones donde $\vec{v} \in S^2$

Cabe notar que un nodo P_1 que no interseccione con otro P_2 en $g_S(\vec{t_1})$ asegura que P_1 no chocará con P_2 desde su posición actual $\vec{x}$ hasta $\vec{x} + \vec{t_1} \cdot \infty$. Esto es así porque una de las características de las estructuras abordadas en este trabajo es la monotonía: las piezas nunca necesitan trasladarse a posiciones intermedias para su desconexión.

Como se verá más adelante el proceso de desensamblado es iterativo hasta que la estructura queda totalmente desconectada, esto es, $\forall n_i$ y $\forall g_S(\vec{t_j})$ con $n_i \in g_S(\vec{t_j})$ se cumple que n_i no tiene ningún arco, ni de entrada ni de salida. Finalmente se calcula la secuencia de montaje como la inversa de la secuencia de desmontaje obtenida y con sus traslaciones invertidas.

Por último añadir que aunque en [17] se hable de rotaciones para desconectar las piezas o incluso de

traslaciones múltiples en [7], en este trabajo solamente se consideran **traslaciones de un paso**. Esto permite cumplir las definiciones de los grafos de intersección vistas más arriba y la expresión (3). Asimismo como direcciones de traslación para construir el NDBG se han empleado las correspondientes a cada uno de los ejes cartesianos en ambas direcciones: $\pm x, \pm y, \pm z$.

3.2. Técnica "assembly-by-disassembly" y heurísticas de selección

Tomando como partida el conjunto G_S, se ha empleado el algoritmo 1 para obtener la secuencia de ensamblado. En dicho algoritmo la función *calcDesconectable* calcula el conjunto desconectable D para cualquier dirección $\vec{v}$ del G_S actual y la elección correspondiente de una $d \in D$ o varias de ellas. Dicha función se describe en el algoritmo 2.

$G_S \longleftarrow$ Construye grafo de intersecciones ;
Secuencia $\longleftarrow$ inicializar lista;
while *existan piezas conectadas en* G_S **do**
  $D^* \longleftarrow calcDesconectable(G_S)$;
  Actualizar información en G_S;
  Añadir D^* en *Secuencia*;
end

Algorithm 1: Algoritmo base de secuenciación

Sea D^* un subconjunto desconectable que contiene varias descomposiciones y que cumple que $D^* \subseteq D$. La variable *Secuencia* es una lista de descomposiciones $d_i \in D^*$ y tras pasar por el módulo denominado *construcción final* (ver figura 4), dicha variable contendrá las operaciones de ensamblado OE en su orden correcto (ver sección 3.4).

ListaDescomposiciones $\longleftarrow$ lista de *conjuntos fuertemente conectados*;
for $\Delta \longleftarrow$ *ListaDescomposiciones* **do**
  if $H(\Delta) = true$ **then**
    $D^* \longleftarrow$ Añade Δ;
  else
  end
end

Algorithm 2: Función que selecciona descomposiciones Δ en base a la función heurística $H(\Delta)$ que se describe en la sección 4, y que sirve para construir un subconjunto desconectable D^*

En base a estos algoritmos y siguiendo el esquema de la figura 4, el módulo *assembly-by-disassembly* se encarga de calcular la lista de descomposiciones, y este a su vez se comunica con el módulo que implementa las heurísticas de selección. Éste último módulo selecciona una o varias descomposiciones en función de un criterio concreto en base a la función $H(\Delta)$. Por tanto se tiene un ciclo de consultas continuo entre módulos para decidir las descomposiciones a seleccionar.

3.3. Selección de secuencia óptima

Una vez el bucle del algoritmo 1 finaliza, la estructura queda completamente desconectada y la secuencia calculada pasa al siguiente módulo encargado de seleccionar la secuencia óptima. Tanto en este módulo como en la función $H(\Delta)$ se medirá la calidad de una descomposicion o de la secuencia completa usando la aproximación que se explicará en la sección 4.1.

3.4. Construcción final

Finalmente las secuencias seleccionadas pasan al último módulo encargado de construir la lista de OE a partir de la *secuencia de desmontaje* obtenida cambiando el orden de las operaciones, invirtiendo las traslaciones y realizando procesamientos opcionales, como por ejemplo especificar las posiciones de las piezas a lo largo de la secuencia calculada.

El formato seguido para la representación de la secuencia toma como base el lenguaje PDDL (Planning Domain Definition Language). Cada acción de montaje será una OE caracterizada por el tipo (pieza fija o móvil), vector de traslación, partes involucradas, efectos que indican la posición de las piezas colocadas y precondiciones que determinan las posiciones de las piezas objetivo a las que se conectan las involucradas en esta operación.

En la siguiente sección se describen con más detalle los módulos encargados de la heurística de selección y de la selección de secuencia óptima, enumerando distintas técnicas implementadas para lograr la secuencia óptima paralela con menor número de pasos para su construcción.

4. TÉCNICAS DE BÚSQUEDAS DE LA SOLUCIÓN

4.1. Definición de índice de una secuencia o descomposición

Se han definido dos índices para medir la calidad de una secuencia: índice basado en el número de movimientos requeridos (I_m) e índice de *paralelización* (I_p). Ambos usan filosofías parecidas, basándose en la adición de un contador a cada pieza. Un contador c_P asociado a una pieza P indicará a partir de qué instante concreto puede conectarse una pieza cualquiera a P. Dicho contador permite incorporar de manera sencilla el dominio temporal a la planificación. El cálculo de I_p y I_m

explicado a continuación se realiza en el módulo que selecciona la secuencia óptima siguiendo un enfoque de *búsqueda global*.

Sea un subconjunto T_{OE} de piezas trasladas de una operación de ensamblado concreta OE y C_{OE} el conjunto de piezas a las que se conectan T_{OE}, el algoritmo 3 se ha empleado para calcular los contadores de una secuencia. Cada vez que se mueve una pieza p, su contador c_p es incrementado.

Inicializar contadores a 0 de cada pieza;
for $OE \longleftarrow$ *secuencia* **do**
  for $PiezaM \longleftarrow T_{OE}$ **do**
    $c_{PiezaM}++$
  end
  for $PiezaC \longleftarrow C^*_{OE}$ **do**
    $c_{PiezaC}++$
  end
end

Algorithm 3: Función que calcula los contadores de una secuencia dada. Se define $C*_{OE}$ como el conjunto de piezas $\in C_{OE}$ tal que intersecionen con T_{OE} en la dirección de traslación de OE

Para el índice I_m se calcula la suma de todos los contadores $\sum_{i=1}^{n} c_{p_i}$, mientras que para I_p se empleará el máximo de los contadores $max(c_{p_i})$, dado que este último índice indica el número de pasos necesarios para el montaje. Como métrica de referencia se usará normalmente la del I_p, dejando la primera para comparar con secuencias en las que se minimice el número de traslaciones para cada pieza. No obstante con I_m se tiene una definición de la calidad de secuencia más precisa, ya que el intervalo de valores va desde $|P_{OE}|$ a ∞, incrementando por cada OE el valor conjunto en r, siendo $r = +(|T_{OE}| + |C^*_{OE}|)$. Es decir, se incrementa en el número de piezas de los conjuntos T y C para la OE actual, que en el peor de los casos será 2. Por el contrario I_p suele incrementarse en menor medida.

A continuación se describen las dos técnicas de optimización (*heurísticas de selección*) que se han desarrollado. Estas técnicas se centran en el algoritmo 2.

4.2. Ramificación y poda (BF)

Inicialmente se desarrolló un algoritmo de fuerza bruta (*brute force, BF)* que sirviera de referencia para evaluar las nuevas técnicas. Para el algoritmo 1 se trata de que el bucle recorra los subconjuntos de piezas en profundidad, esto es, si una desconexión genera un nuevo subconjunto, el algoritmo inspecciona este último hasta desconectarlo completamente, y así hasta terminar con todos. Introduciendo una cota en la ejecución para evitar seguir calculando secuencias que superen una métrica umbral y realizando un cómputo parcial de la métrica para las secuencias parcialmente construidas, se consigue reducir el tiempo de ejecución T_c, pero no tanto como para que esta técnica sea útil de cara a estructuras complejas como se verá en la sección 5.

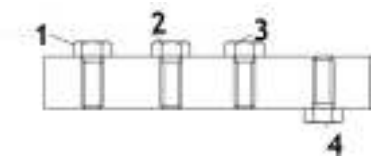


Figura 6: El $\mathbf{G}_S$ obtiene que los tornillos del 1 al 4 son desconectables, pero se ve claramente que en vez de realizar 4 operaciones separadas podría realizarse solamente una con los 4 tornillos

4.3. Técnica de agrupación (TA)

En este caso, a lo largo de las operaciones se detecta que existen varios subconjuntos que podrían desconectarse **a la vez**, esto es, mediante una única operación de desmontaje. Con esta simple idea el despliegue de un árbol de ejecución se reduce drásticamente. Para la estructura de la figura 6 con BF se obtienen 4! secuencias en el peor de los casos y solamente una con la técnica de agrupación.

Para realizar esta agrupación de desconexiones se deben seguir las siguientes reglas:

- Los subconjuntos no pueden tener piezas en común. Esto daría lugar a incongruencias, apareciendo en la secuencia operaciones como *(conectar A y B a C)* y *(conectar B a C)* a la vez, algo carente de sentido físico.
- Siguiendo las definiciones vistas en la sección 4.1, para dos descomposiciones a agrupar Δ_1 y Δ_2 y dos piezas p_1 y p_2, no puede cumplirse que

$$(p_1 \in T_{\Delta_1} \wedge p_1 \in C^*_{\Delta_2}) \vee \quad (5)$$
$$(p_2 \in T_{\Delta_2} \wedge p_2 \in C^*_{\Delta_1}) \, . \quad (6)$$

  Esto evita situaciones en las que se desconectan dos subconjuntos de piezas en traslaciones tales que existe colisión geométrica.

Con todo esto se evita un proceso muy determinista en el cálculo de la secuencia, dando lugar a un pequeño abanico de posibilidades aun realizando la agrupación, y se consigue además decrementar el tiempo de cómputo T_c considerablemente.

4.4. Características de las técnicas BF y TA

Las distintas características son enumeradas a continuación:

- **Ejecución independiente**. Sea $\mathbf{N}_i$ el número de secuencias de un subconjunto i, $|S_i|$. Sabiendo que el algoritmo 1 sigue un comportamiento en profundidad por cada uno de los subconjuntos generados a partir de las descomposiciones del *assembly-by-disassembly*, el número de secuencias con BF sería de $\prod_{i=1}^{i=n} \mathbf{N}_i$. Para cada subconjunto de manera independiente la mejora es considerable: $\sum_{i=1}^{i=n} \mathbf{N}_i$.
- **Cacheado de subconjuntos**. A partir de lo anterior se determina que es posible incluso almacenar las secuencias calculadas de los subconjuntos, de tal manera que se eviten cálculos redundantes. Experimentalmente se ha visto que para estructuras de 12 piezas existen más de 5000 coincidencias.
- **Combinaciones de subconjuntos**. Esto es una característica para evitar ejecuciones masivas. No es más que diferenciar el caso de obtener desconexiones sólo de subconjuntos presentes en el perímetro de la estructura o bien calcular todas las posibles combinaciones de subconjuntos.

5. ANÁLISIS DE LAS TÉCNICAS PROPUESTAS

Se define la *complejidad de una estructura* C como la suma del número de arcos de todos los grafos del conjunto G_S. Dicha complejidad se emplea siempre en el eje horizontal de las gráficas de esta sección.

Las figuras 7 y 8 comparan los tiempos de ejecución T_c de las técnicas BF y TA. En la figura 7 se puede comprobar la diferencia en T_c para la técnica BF, analizando el uso del **cacheado de subconjuntos**. Con complejidades superiores a 120 el T_c se hace demasiado alto. Se puede observar la existencia de un máximo en la complejidad 108 y una bajada en la siguiente estructura más compleja. Eso se debe a los subconjuntos obtenidos: una estructura a la que en un momento dado se le puedan desconectar 10 piezas individuales tendrá mucho más T_c que otra con 2 subconjuntos de 3 piezas cada una, ya que la casuística es mucho mayor en el primer caso.

En la figura 8 se analiza la técnica TA cuando se emplean descomposiciones perimetrales o todas las combinaciones posibles. Como es obvio la casuística para todas las combinaciones es inmensa y el T_c aumenta. No obstante los tiempos son menores que en la figura 7 para las mismas estructuras.

En la figura 9 se muestra la comparativa a efectos de calidad de la secuencia en base a los índices I_m e I_p. Se aprecia que los I_m del método BF son peores que los obtenidos mediante la técnica TA a partir de cierto punto (cuando teóricamente deberían ser iguales o mejores). La explicación es que a partir de una complejidad 120, el método BF obtiene resultados incompletos debido al excesivo T_c que obliga a detener la ejecución. Se puede observar también que tomando todas las combinaciones posibles, el TA obtiene mejores secuencias que TA con sólo perimetrales. También es importante explicar que desde 100 hacia atrás los resultados son idénticos, con lo cual las secuencias obtenidas con la técnica TA son tan óptimas como las del método BF de referencia.

Por último en la figura 10 se observa el mismo fenómeno explicada anteriormente para el método BF. A partir de una complejidad 54, el número de pasos necesarios para construir los montajes es el óptimo (3) tanto con descomposiciones superficiales o perimetrales como para todas las combinaciones.

6. CONCLUSIONES Y DESARROLLOS FUTUROS

Se ha comprobado que la técnica TA desarrollada ofrece buenos resultados, tanto en tiempos de ejecución como en las métricas empleadas para valorar las secuencias. Asimismo, se ha visto que la métrica del número de movimientos I_m en la secuencia ofrece peores resultados que la I_p. En la práctica, I_m intenta dejar el máximo número de piezas sin mover, como piezas base, a costa de mover subconjuntos muy grandes, incrementando su contador. Por el contrario la I_p intentará desplazar todos los subconjuntos que pueda para un mismo paso. Por ejemplo, en el primer paso de desmontaje desconectará todos los subconjuntos que modifiquen los I_m lo mínimo posible, mejorando así la paralelización de la secuencia.

Como se ha visto en el presente artículo, hay una *búsqueda local* en la parte de la selección de una o varias descomposiciones y una *búsqueda global* cuando se elige una o varias secuencias de entre todas las generadas. Siguiendo esta línea se podría definir otra métrica como $M_{sec} = E + P$, en la que E sería el esfuerzo generado hasta una fase (medición de las operaciones anteriores a la actual en la secuencia que se está generando) y P una medida probabilística que indicaría un valor aproxi-

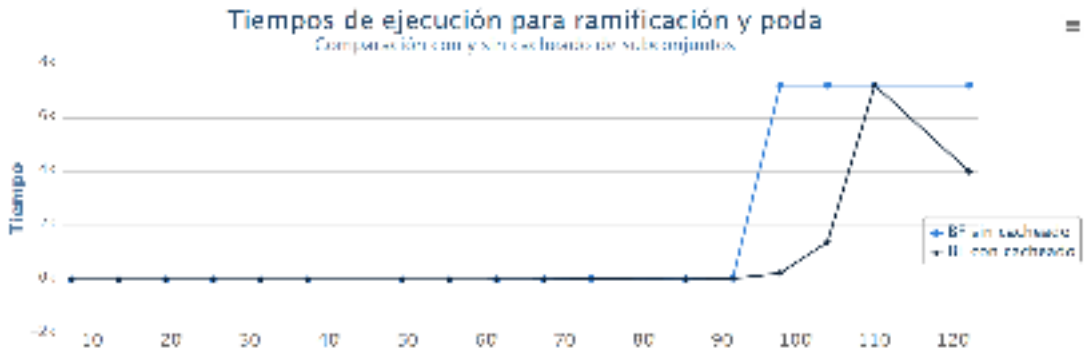


Figura 7: T_c de ramificación y poda

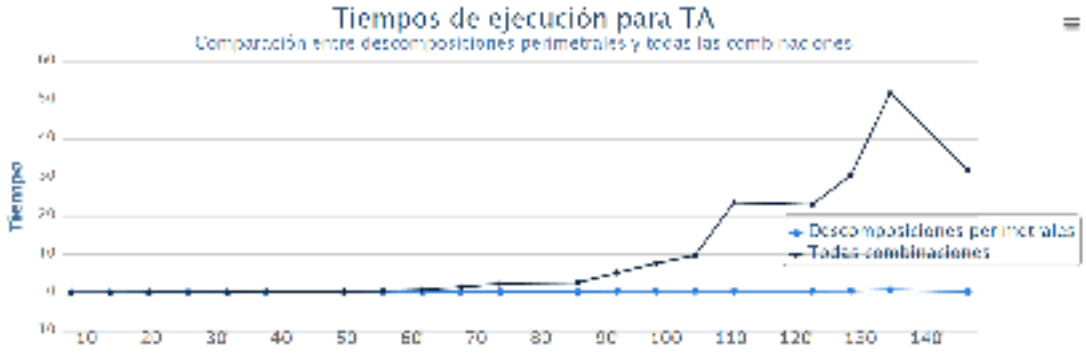


Figura 8: T_c de la técnica TA

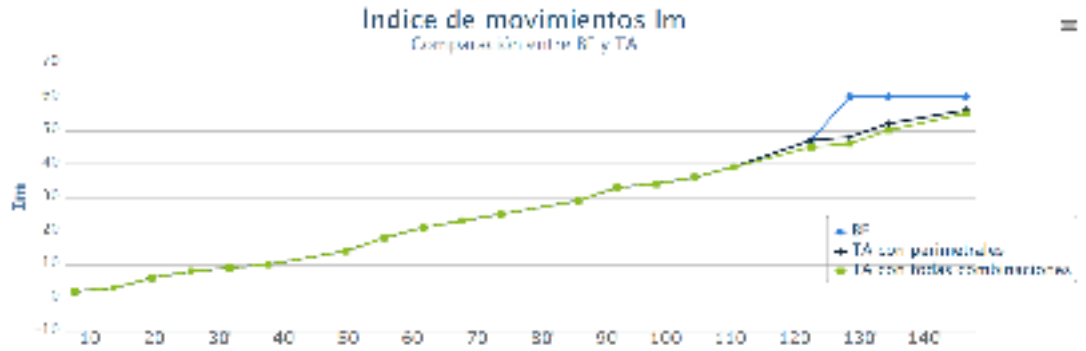


Figura 9: I_m de ramificación y poda frente a TA

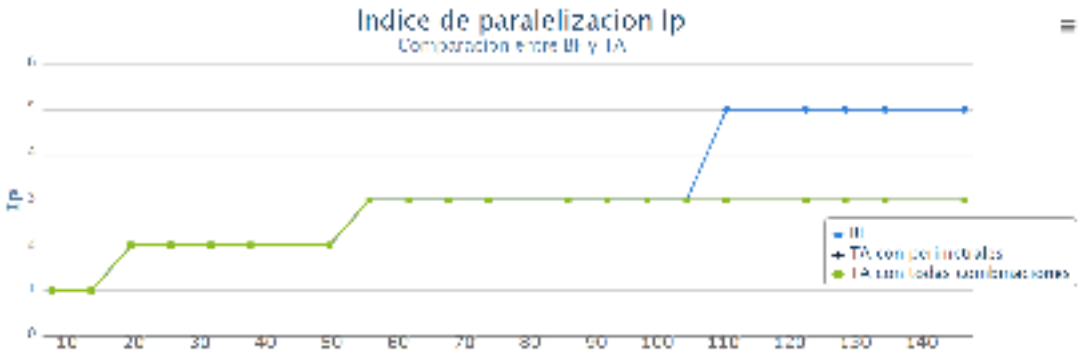


Figura 10: I_p de BF frente a TA

mado del esfuerzo que falta para completar la secuencia en caso de elegir una descomposición concreta. Asimismo, como desarrollo futuro también se podría diseñar otro algoritmo para seleccionar aquellas descomposiciones que más componentes conexas del grafo genere tras su descomposición, además de que dichas componentes dispongan de un número de piezas parecido.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el proyecto ARCAS (ICT-2011-287617) del séptimo Programa Marco de la Comisión Europea y los proyectos nacionales RANCOM (P11-TIC-7066) y CLEAR (DPI2011-28937-C02-01).

Referencias

[1] D. Ben-Arieh, R.R. Kumar, and M.K. Tiwari. Analysis of assembly operations'difficulty using enhanced expert high-level colored fuzzy petri net model. *Robotics and Computer-Integrated Manufacturing*, 20(5):385–403, 2004.

[2] P.-B. Cao and R.-B. Xiao. Assembly planning using a novel immune approach. *International Journal of Advanced Manufacturing Technology*, 31(7-8):770–782, 2007.

[3] C.K. Choi, X.F. Zha, T.L. Ng, and W.S. Lau. On the automatic generation of product assembly sequences. *International Journal of Production Research*, 36(3):617–633, 1998.

[4] S.D. Dao and R.M. Marian. Genetic algorithms for integrated optimisation of precedence-constrained production sequencing and scheduling. *Lecture Notes in Electrical Engineering*, 130 LNEE:65–80, 2013.

[5] Jifeng Guo, Ping Wang, and Naigang Cui. Adaptive ant colony algorithm for on-orbit assembly planning. In *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, pages 1590–1593, 2007.

[6] D. Halperin, J.-C. Latombe, and R.H. Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica (New York)*, 26(3-4):577–601, 2000.

[7] D. Halperin and R.H. Wilson. Assembly partitioning along simple paths: The case of multiple translations. *Advanced Robotics*, 11(2):127–145, 1997.

[8] Luiz S. Homem de Mello and Arthur C. Sanderson. And/or graph representation of assembly plans. *IEEE Transactions on Robotics and Automation*, 6(2):188–199, 1990.

[9] P. Jimenez. Survey on assembly sequencing: a combinatorial and geometrical perspective. *Journal of Intelligent Manufacturing*, pages 1–16, 2011.

[10] J.-C. Latombe, R.H. Wilson, and F. Cazals. Assembly sequencing with toleranced parts. *CAD Computer Aided Design*, 29(2):159–174, 1997.

[11] R.M. Marian, L.H.S. Luong, and K. Abhary. Assembly sequence planning and optimisation using genetic algorithms part i. automatic generation of feasible assembly sequences. *Applied Soft Computing Journal*, 2(3):223–253, 2003.

[12] R.M. Marian, L.H.S. Luong, and K. Abhary. A genetic algorithm for the optimisation of assembly sequences. *Computers and Industrial Engineering*, 50(4):503–527, 2006.

[13] C. Sinanoglu and H.R. Borklu. An assembly sequence-planning system for mechanical parts using neural network. *Assembly Automation*, 25(1):38–52, 2005.

[14] C. Sinanoglu and Borklu H.R. An approach to determine geometric feasibility to assembly states by intersection matrices in assembly sequence planning. *Journal of Intelligent Manufacturing*, 15(4):543–559, 2004.

[15] H.-E. Tseng. An improved ant colony system for assembly sequence planning based on connector concept. *Lecture Notes in Electrical Engineering*, 97 LNEE(VOL. 1):881–888, 2011.

[16] J.F. Wang, J.H. Liu, and Y.F. Zhong. A novel ant colony algorithm for assembly sequence planning. *International Journal of Advanced Manufacturing Technology*, 25(11-12):1137–1143, 2005.

[17] R.H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71(2):371–396, 1994.

[18] Z. Yin, H. Ding, H. Li, and Y. Xiong. A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. *CAD Computer Aided Design*, 35(1):37–56, 2003.

ESTUDIO DE FIABILIDAD DE UNA PLATAFORMA DE COMUNICACIÓN BASADA EN EL ESTÁNDAR 802.11

Francisco Cárdenas Capitán[1], José M. Fornés Rumbao[1], Francisco Rodríguez Rubio[2], Rafael Estepa Alonso[1].
[1]Departamento de Ingeniería Telámatica, [2]Departamento de Ingeniería de Sistemas y Automática
Escuela Superior de Ingenieros – Universidad de Sevilla
Camino de los Descubrimientos s/n, Isla de la Cartuja, 41092.
fcardenas2@us.es, fornes@trajano.us.es, rubio.us.es, rafaestepa@us.es

Resumen

Hemos realizado una plataforma (red) de comunicación basada en el estándar 802.11 formada por un punto de acceso y una estación con el objetivo de desarrollar futuros algoritmos que optimicen parámetros fundamentales como el rendimiento, el consumo de energía, el retraso, etc. En este artículo se estudia la fiabilidad de dicha plataforma para considerar la veracidad de los resultados que se obtendrán en las pruebas experimentales realizadas en esta plataforma, así como determinar y acotar su error.

Palabras Clave: 802.11, tiempo de transmisión, Madwifi, régimen binario.

1 INTRODUCCIÓN

Este artículo tiene como objetivo analizar y estudiar la fiabilidad de un sistema de comunicación basado en el estándar 802.11 con el objetivo de utilizarla en el estudio de la eficiencia energética a través de algoritmos y métodos alternativos a los ya existentes.
Analizar la fiabilidad de un sistema es fundamental para garantizar los resultados que deriven de ella. Es por este motivo por lo que dedicaremos la totalidad de este artículo a establecer unas cotas de error y garantizar la fiabilidad de los resultados de nuestro sistema.

El artículo se dividirá de la siguiente manera; la Sección II especificará el escenario en el cual han tenido lugar las pruebas. También se detallará los parámetros seleccionados para el estudio y el método de su recopilación. La Sección III mostrará las estadísticas de los parámetros, anteriormente recogidos, de distintas formas (histogramas, porcentajes, filtrados, etc). En la Sección IV; se expondrán la interpretación de las estadísticas realizadas y especificaremos los motivos de sus valores. La Sección V realizará una comparativa entre los resultados obtenidos experimentalmente y un modelo teórico previamente desarrollado. Por último, la Sección VI se dedicará a la conclusión.

2 PLATAFORMA Y CONDICIONES DEL ESCENARIO

El sistema de comunicación; objeto de estudio; se basará en el estándar 802.11g. Además estará formado por dos elementos; un punto de acceso y una estación asociada al anterior mediante el modo infraestructura que contempla la norma ya mencionada. Dichos elementos se corresponden físicamente con dos PC´s equipados con una tarjeta pci D-Link DWL-G520 con chipset Atheros AR5001X y espaciados tres metros entre sí.
Como driver para la tarjeta ha sido elegido Madwifi por varias razones:

- Desarrollado específicamente para chipset Atheros.
- Software de libre licencia (aunque existe una capa HAL propietaria).
- Escrito en lenguaje C++; facilitando su comprensión y modificación.
- Está sujeto a continuas actualizaciones.
- Gran soporte de información.

Dicho driver se comunicará con un software (escrito en C) mediante llamadas IOCTL para obtener información (estadísticas, valores, parámetros, registros, etc) del driver relacionada con la comunicación. Para ello se adaptará tanto el código del driver como el software de más alto nivel.

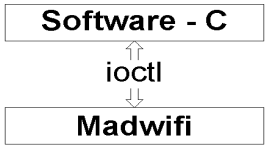


Figura 1: Capas software de la plataforma.

Se utilizará el canal 6 (de los 14 posibles) para la comunicación entre ambos elementos. Por dicho canal el punto de acceso enviará una trama Beacon

cada 1000 ms. Por otro lado; la estación hará uso de la norma 802.11e para el envío de sus tramas de datos; asignando la prioridad Best Effort a cada uno de los paquetes que serán enviados desde la estación hasta su punto de acceso. Estos paquetes serán enviados cada 100 milisegundos (estas tramas contendrán varios (configurable) números aleatorios que pueden representar sondeos de una variable como por ejemplo la temperatura).

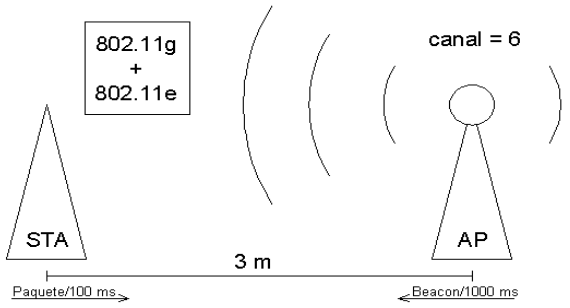


Figura 2: Escenario físico de la plataforma.

La elección del parámetro que demuestre la fiabilidad de nuestro sistema es fundamental y debe ser apto para la comparación con otros resultados de ese mismo parámetro por otros métodos teóricos o de simulación.

Un parámetro que puede ser fácilmente observable; que cumple con el requisito anterior y además es muy importante para el objetivo posterior de esta plataforma es el **tiempo**. Mediremos el tiempo de transferencia completa experimentalmente. Dicho tiempo lo definiremos como el tiempo transcurrido desde que el driver introduce una trama en cola hasta que se recibe el asentimiento de dicho paquete por parte del punto de acceso. Tomaremos 300 muestras de este valor (tiempo de transferencia de 300 paquetes) proporcionadas por el driver, tras modificar su código fuente para ello y recogidas por un software de más alto nivel a través de comandos IOCTL. Junto a estas 300 muestras tomaremos otros parámetros importantes que nos ayudarán en la realización de las estadísticas. Estos parámetros son el número de intentos, el tamaño del paquete y el numero de secuencia del paquete; todos ellos asociados a cada muestra.

El procedimiento de recogida de estas muestras es muy sencillo: Se inician los dos equipos; estación y punto de acceso, se espera un pequeño tiempo para que pueda llevarse a cabo las tareas de asociación entre las estaciones. Después iniciamos el envío de las tramas cada 100 ms a través de un software en C muy simple. Transcurridos los primeros 30 segundos (tiempo en el que obtenemos 300 muestras) obtenemos el primer informe de estas 300 muestras junto con sus tres parámetros asociados. Con el objetivo de obviar tramas iniciales de gestión en nuestros resultados; este primer informe será omitido y esperaremos el segundo; tras otros 30 segundos de espera.

Mediante otra tarjeta pci (realtek rtl8185) instalada en el punto de acceso monitorizaremos el tráfico que está teniendo lugar en el canal 6 mediante la herramienta airodump-ng.

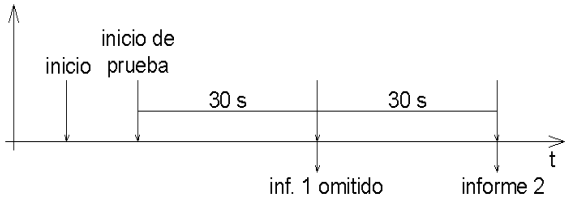


Figura 3: Procedimiento de recogida de muestras.

Debido al entorno urbano, y especialmente universitario en el que nos encontramos; el canal 6 no está libre de tramas ajenas a nuestro sistema que podemos considerar como ruido. Estas tramas son generalmente aleatorias. Las tramas ajenas más frecuentes son Probe Request y Probe Response. Intentando caracterizar dicho tráfico hemos observado mediante la monitorización con airodump-ng; en el periodo de una prueba (30 segundos) una media de 12 Probe Request; enviadas en pares y con una PSDU de 49 bytes. Además; en ocasiones aleatorias se responde a estas peticiones con sucesivas tramas Probe Response desde el punto de acceso con una PSDU de 114 bytes.

3 EXPOSICIÓN DE LAS ESTADÍSTICAS

Conocido el método de obtención de muestras; realizaremos en cuatro situaciones este procedimiento modificando el régimen binario y el tamaño del paquete para obtener una visión más global del comportamiento y fiabilidad del sistema. Estos cuatro escenarios son los siguientes:

- Régimen Binario a 54 Mbps y paquete con PSDU de 80 bytes.
- Régimen Binario a 54 Mbps y paquete con PSDU de 800 bytes.
- Régimen Binario a 6 Mbps y paquete con PSDU de 80 bytes.
- Régimen Binario a 6 Mbps y paquete con PSDU de 800 bytes.

Mostraremos los resultados obtenidos para cada uno de los casos mediante gráficas, porcentajes, etc. Es necesario decir que las estadísticas mostrarán los resultados de las muestras cuyos paquetes asociados no han necesitado retransmisiones durante la

transferencia; es decir; sobre las 300 muestras obtenidas se efectuará un filtrado de aquellas muestras que tengan un número de intentos asociado superior a 1. Por lo tanto; tendremos un número de muestras siempre menor o igual a 300. De todos modos se realizará posteriormente un análisis de estas tramas retransmitidas.

3.1 Rb = 54 Mb Y LONGITUD DE PAQUETE DE 80 Bytes

Los resultados para este escenario se muestran a continuación. Los primeros datos a tener en cuenta son los siguientes:

- Mediana: 140 µs
- Reintentos: 20
- Muestras analizadas: 280

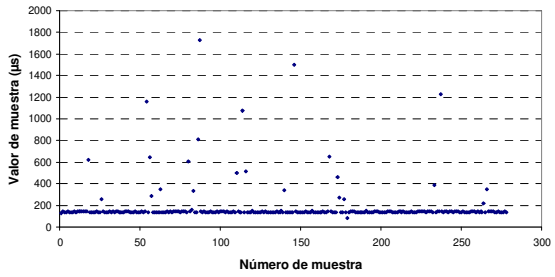


Figura 4: Valores de muestras en orden secuencial.

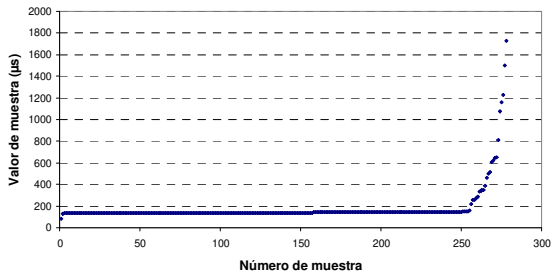


Figura 5: Valores de muestras en orden ascendente.

En estas dos gráficas podemos observar como 256 muestras de las 280 totales están fijadas en un valor aproximado de 140 µs. Las 24 muestras restantes superan ese valor.

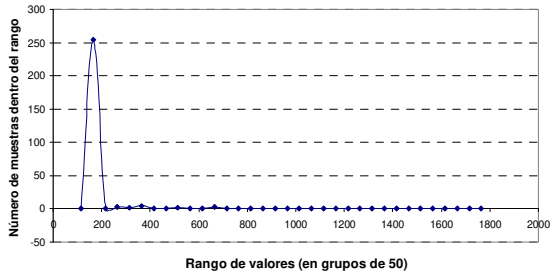


Figura 6: Distribución de muestras – Histograma.

Podemos observar como más del 90% de las muestras se encuentran en torno a la mediana.

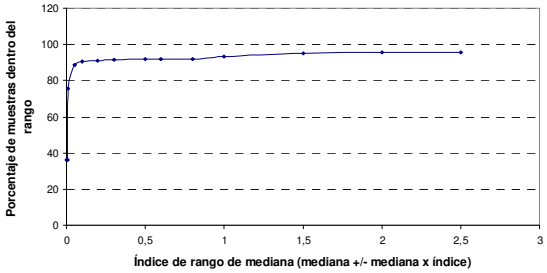


Figura 7: Distribución de muestras.

La anterior gráfica refleja como más del 90% de las muestras se encuentran dentro del rango mediana +/- mediana x 0.1; es decir; se encuentran dentro del rango limitado por el 10% de su valor [126,154].

Además; podemos establecer una cota centrada en la mediana y que albergue el 95% de las muestras (después de la filtración de reintentos y valores anómalos): 140 +/- 4 µs; es decir, el rango [136, 144] comprende el 95% de las muestras centradas en la mediana.

3.2 Rb = 54 Mb Y LONGITUD DE PAQUETE DE 800 Bytes

Los resultados para este escenario se muestran a continuación. Los primeros datos a tener en cuenta son los siguientes:

- Mediana: 244 µs
- Reintentos: 12
- Muestras analizadas: 288

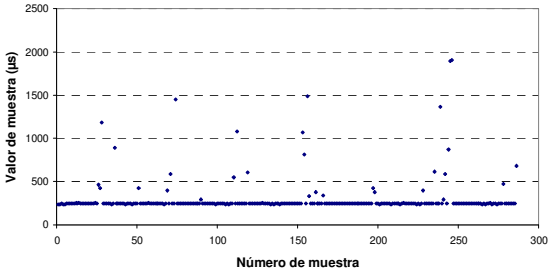


Figura 8: Valores de muestras en orden secuencial.

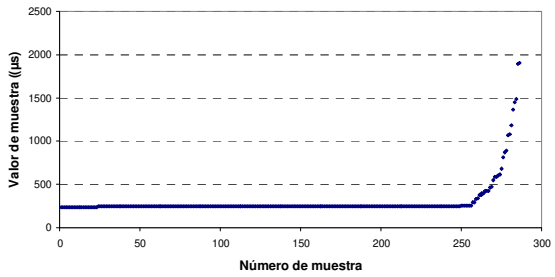


Figura 9: Valores de muestras en orden ascendente.

Estas dos gráficas muestran como 258 muestras de las 288 totales están fijadas en un valor aproximado de 244. Las 30 muestras restantes superan ese valor.

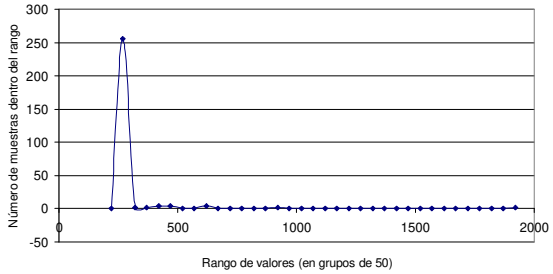


Figura 10: Distribución de muestras – Histograma.

En el histograma para este escenario podemos observar como más del 88% de las muestras se encuentran en torno a la mediana.

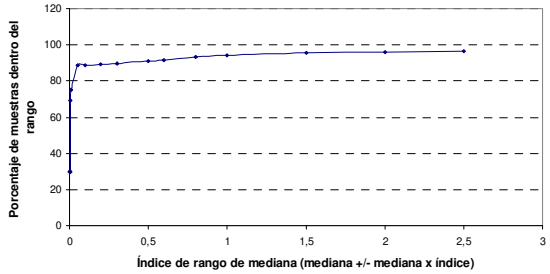


Figura 11: Distribución de muestras.

Esta distribución de muestras refleja como más del 88% de las muestras se encuentran dentro del rango mediana +/- mediana x 0.1; es decir; se encuentran dentro del rango limitado por el 10% de su valor [219,269].

Además; podemos establecer una cota centrada en la mediana y que albergue el 95% de las muestras (después de la filtración de reintentos y valores anómalos): 244 +/- 6 µs; es decir, el rango [238, 250] comprende el 95% de las muestras centradas en la mediana.

3.3 Rb = 6 Mb Y LONGITUD DE PAQUETE DE 80 Bytes

Los resultados para este escenario se muestran a continuación. Los primeros datos a tener en cuenta son los siguientes:

- Mediana: 258 µs
- Reintentos: 5
- Muestras analizadas: 295

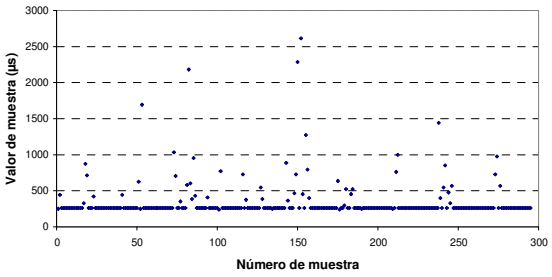


Figura 12: Valores de muestras en orden secuencial.

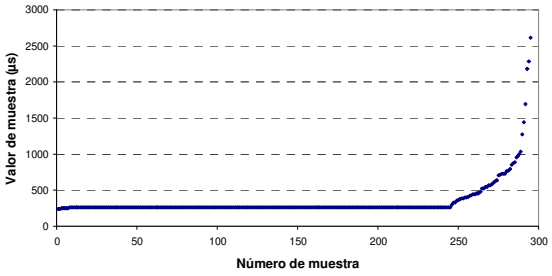


Figura 13: Valores de muestras en orden ascendente.

Observando dichas funciones observamos como 245 muestras de las 295 totales están fijadas en un valor aproximado de 258. Las 50 muestras restantes superan ese valor. En este caso el número de muestras con un valor significativamente mayor que la mediana es más alto que en casos anteriores debido a que se ha observado un alto tráfico de tramas Probe Response. En los otros escenarios esta situación era menos acentuada o simplemente no existió.

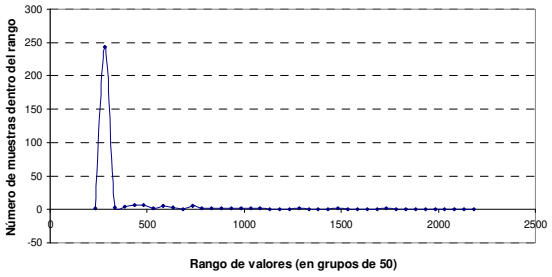


Figura 14: Distribución de muestras – Histograma.

Podemos observar como prácticamente 83% de las muestras se encuentran en torno a la mediana.

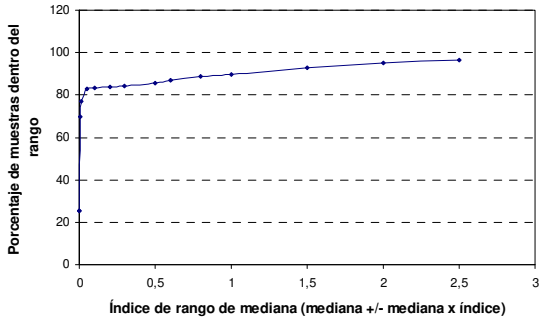


Figura 15: Distribución de muestras.

Esta distribución refleja como más del 83% de las muestras se encuentran dentro del rango mediana +/- mediana x 0.1; es decir; se encuentran dentro del rango limitado por el 10% de su valor [233,283].

Además; podemos establecer una cota centrada en la mediana y que albergue el 95% de las muestras (después de la filtración de reintentos y valores anómalos): 258 +/- 3 µs; es decir, el rango [255, 261] comprende el 95% de las muestras centradas en la mediana.

3.4 Rb = 6 Mb Y LONGITUD DE PAQUETE DE 800 Bytes

Los resultados para este escenario se muestran a continuación. Los primeros datos a tener en cuenta son los siguientes:

- Mediana: 1217 µs
- Reintentos: 14
- Muestras analizadas: 286

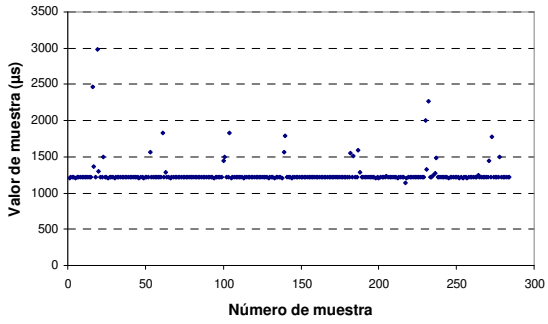


Figura 16: Valores de muestras en orden secuencial.

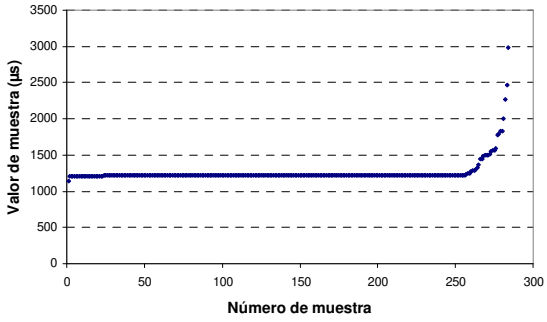


Figura 17: Valores de muestras en orden ascendente.

En sendas gráficas podemos observar como 259 muestras de las 286 totales están fijadas en un valor aproximado de 1217. Las 27 muestras restantes superan ese valor.

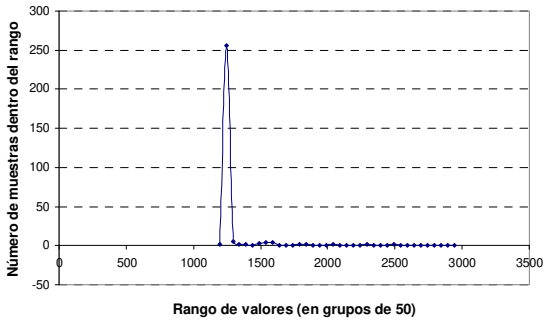


Figura 18: Distribución de muestras – Histograma.

Podemos observar como casi el 90% de las muestras se encuentran en torno a la mediana.

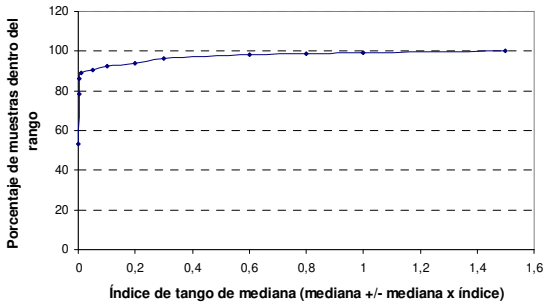


Figura 19: Distribución de muestras.

La anterior gráfica refleja como más del 92% de las muestras se encuentran dentro del rango mediana +/- mediana x 0.1; es decir; se encuentran dentro del rango limitado por el 10% de su valor [1192,1242].

Además; podemos establecer una cota centrada en la mediana y que albergue el 95% de las muestras (después de la filtración de reintentos y valores anómalos): 1217 +/- 5 µs; es decir, el rango [1212, 1222] comprende el 95% de las muestras centradas en la mediana.

Mostradas las estadísticas para cada escenario; procedemos a analizar y justificar dichos resultados.

4 ANÁLISIS DE RESULTADOS

En esta sección pretendemos proporcionar motivos y argumentos para justificar el resultado parcial o global de las muestras obtenidas en cada uno de los escenarios.

Es necesario destacar que para este análisis es fundamental la información obtenida mediante la herramienta airodump-ng ya mencionada. De ella obtendremos los instantes de tiempos exactos y la secuencia de los paquetes que dicho software de monitorización sea capaz de recopilar. Por tanto; visualizaremos tramas Beacon, de datos, Probe Request, Probe Response u otra trama cualquiera que esté emitiéndose en el canal 6.

Analizando el fichero de registro de airodump-ng correspondiente a cada escenario; podemos observar las secuencias de las tramas captadas e incluso acceder al número de secuencia de cada trama. Esto nos permite contraponer los resultados obtenidos en nuestras pruebas (menor o igual a 300 paquetes) con los números de secuencia de los paquetes registrados con airodump-ng.

Con este procedimiento observamos que hay **coincidencias repetitivas** entre muestras con un valor alejado de la mediana con la aparición de tramas de gestión (beacon, Probe Request o Probe Response).

Llevando a cabo un análisis de cada una de las muestras podemos justificar el valor anómalo de algunas de ellas.

4.1 Rb = 54 Mb Y LONGITUD DE PAQUETE DE 80 Bytes

Definiendo muestras anómalas como una muestra alejada sustancialmente de la mediana (fuera del rango +/- 10% de su valor); mostramos a continuación, la tabla que recoge el análisis llevado a cabo para este escenario:

Tabla 1: Análisis Rb = 54 mb y psdu = 80 bytes

	Número	Porcent. (%)
Muestras totales:	280	----
Mediana:	140	----
Muestras anómalas (mayores de 160)	24	8,6
Muestras anómalas debidas a Beacon	9	3,2
Muestras anómalas debidas a Probe Request	2	0,7
TOTAL		
Muestras anómalas justificadas	11	3,9
Muestras anómalas injustificadas	13	4,6
Muestras jutificadas	267	95,4

De esta forma; justificamos el 45% de las muestras anómalas debido a la aparición de tramas Beacon y Probe Request. Además podemos proporcionar para las tramas anómalas justificadas el retraso máximo producido por la trama que lo originó en este escenario:

- Retraso máximo producido por la trama beacon: 1590 µs.
- Retraso máximo producido por trama Probe Request: 205 µs.

En la siguiente figura aparecen con distinto color; por un lado las muestras con un valor cercano a la mediana o más alto pero justificado y por otro lado las muestras con un valor anómalo injustificado.

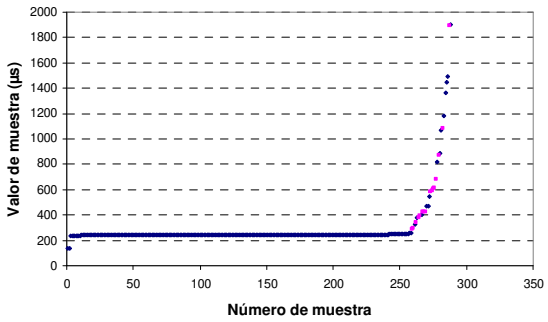


Figura 20: Distribución de muestras.

4.2 Rb = 54 Mb Y LONGITUD DE PAQUETE DE 800 Bytes

A continuación mostramos la tabla que recoge el análisis llevado a cabo para este escenario:

Tabla 2: Análisis Rb = 54 mb y psdu = 800 bytes

	Número	Porcent (%)
Muestras totales:	288	----
Mediana:	244	----
Muestras anómalas (mayores de 260)	30	10.4
Muestras anómalas debidas a Beacon	8	2.8
Muestras anómalas debidas a Probe Request	6	2.1
TOTAL		
Muestras anómalas justificadas	14	4.9
Muestras anómalas injustificadas	16	5.6
Muestras jutificadas	267	94.4

De esta forma; justificamos el 47% de las muestras anómalas debido a la aparición de tramas Beacon y Probe Request.

- Retraso máximo producido por la trama beacon: 1660 µs.
- Retraso máximo producido por trama Probe Request: 1245 µs.

En la siguiente figura aparecen con distinto color; por un lado las muestras con un valor cercano a la mediana o más alto pero justificado y por otro lado las muestras con un valor anómalo injustificado para este escenario.

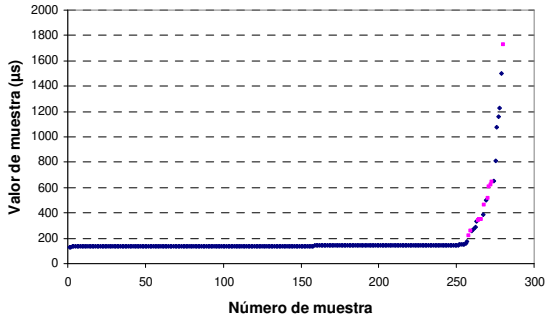


Figura 21: Distribución de muestras.

4.3 Rb = 6 Mb Y LONGITUD DE PAQUETE DE 80 Bytes

A continuación mostramos la tabla que recoge el análisis llevado a cabo para este escenario:

Tabla 3: Análisis Rb = 6 mb y psdu = 80 bytes

	Número	Porcent. (%)
Muestras totales:	295	----
Mediana:	258	----
Muestras anómalas (mayores de 260)	50	16.9
Muestras anómalas debidas a Beacon	14	4.7
Muestras anómalas debidas a Probe Request	5	1.7
Muestras anómalas debidas a Probe Response	3	1
Muestras anómalas debidas a otras tramas en el canal	2	0.7
TOTAL		
Muestras anómalas justificadas	24	8.1
Muestras anómalas injustificadas	26	8.8
Muestras jutificadas	269	91.2

De esta forma; justificamos el 48% de las muestras anómalas debido a la aparición de tramas Beacon y Probe Request, Probe Response y otras tramas de datos.

- Retraso máximo producido por la trama beacon: 1432 µs.
- Retraso máximo producido por trama Probe Request: 2027 µs.
- Retraso máximo producido por trama Probe Response: 1185 µs.
- Retraso máximo producido por otras tramas aleatorias 512 µs.

En la siguiente figura aparecen con distinto color; por un lado las muestras con un valor cercano a la mediana o más alto pero justificado y por otro lado las muestras con un valor anómalo injustificado.

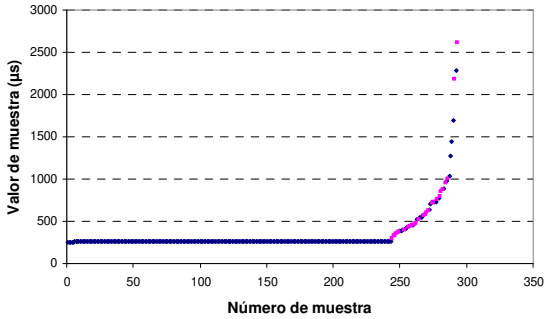


Figura 22: Distribución de muestras.

4.4 Rb = 6 Mb Y LONGITUD DE PAQUETE DE 800 Bytes

A continuación mostramos la tabla que recoge el análisis llevado a cabo para este escenario:

Tabla 4: Análisis Rb = 6 mb y psdu = 800 bytes

	Número	Porcentaje (%)
Muestras totales:	286	----
Mediana:	1217	----
Muestras anómalas (mayores de 260)	27	9.4
Muestras anómalas debidas a Beacon	6	2.1
Muestras anómalas debidas a Probe Request	1	0.3
TOTAL		
Muestras anómalas justificadas	7	2.4
Muestras anómalas injustificadas	20	7
Muestras jutificadas	266	93

De esta forma; justificamos el 26% de las muestras anómalas debido a la aparición de tramas Beacon y Probe Request.

- Retraso máximo producido por la trama beacon: 1055 µs.
- Retraso máximo producido por trama Probe Request: 27 µs.

En la siguiente figura aparecen con distinto color; por un lado las muestras con un valor cercano a la mediana o más alto pero justificado y por otro lado las muestras con un valor anómalo injustificado.

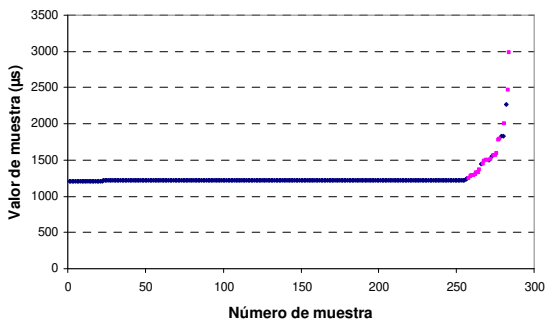


Figura 23: Distribución de muestras.

5 COMPARATIVA CON EL MODELO TEÓRICO

Esta sección plasmará una comparativa entre los resultados de las muestras recogidas para cada escenario en la sección anterior y los cálculos obtenidos de un análisis teórico que a continuación desarrollaremos.

El siguiente análisis teórico tendrá como objetivo calcular el tiempo de transferencia completo de un paquete para cada escenario siguiendo fielmente la norma 802.11g. Según esta norma el procedimiento durante la transmisión de un paquete, sin la utilización del mecanismo RTS-CTS, es el siguiente:

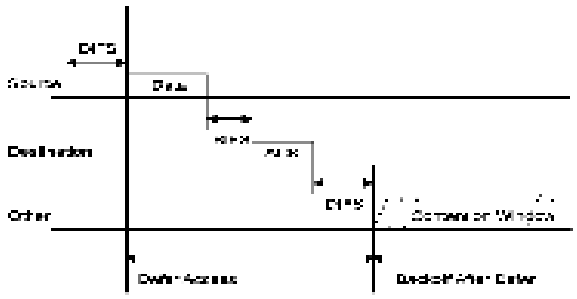


Figura 24: Proceso de transmisión de un paquete sin mecanismo RTS-CTS.

Además debemos tener en cuenta que empleamos la norma 802.11e y debemos regirnos por los tiempos inter-frame (AIFS) que se especifican para cada uno de las categorías de acceso (Background, Best Effort, Video y Voice):

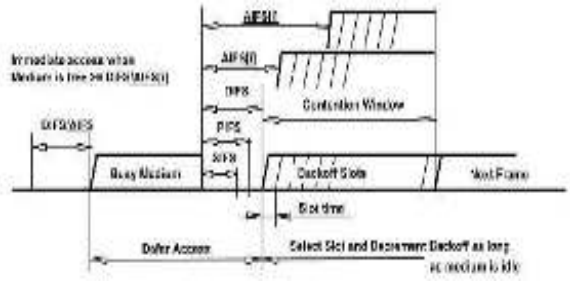


Figura 25: Tiempos AIFS aplicados a las categorías de acceso.

Teniendo en cuenta estas dos gráficas podemos desarrollar; el tiempo de transferencia completo (hasta asentimiento) de la siguiente manera:

$$T_Transf. = T_backoff + AIFS[AC] + Tx_time + SIFS + T_ack \quad (1)$$

Para computar cada uno de los elementos; acudiremos a la norma [1]. En ella se dice; en primer lugar que existen dos tipos de implementaciones físicas de esta norma: DSSS-OFDM y ERP-OFDM. La primera se usa para compatibilizar esta norma con las estaciones que utilizan 802.11b para su comunicación. ERP-OFDM; es utilizada por las estaciones cuando no existen en su red ningún nodo que implemente el anterior modo 802.11b.

En nuestro caso contamos únicamente con un nodo y un punto de acceso con idénticas tarjetas pci habilitadas con las normas 802.11b/g. Al no existir ningún nodo externo en nuestra red que use exclusivamente 802.11b; nuestras estaciones utilizarán la implementación física ERP-OFDM.

Siguiendo estas directrices en la norma; podemos fijar el valor de SIFS = 10 µs y el tiempo de slot = 9 µs. Por otro lado, la norma 802.11e define el tiempo AIFS de la siguiente manera:

$$AIFS[AC] = AIFSN[AC] \times aSlotTime + aSIFSTime \quad (2)$$

Donde AC es la categoría de acceso y AIFSN[AC] es el número de espacio intertrama para un AC en concreto. En nuestro caso AIFSN[BE] = 2.

Aplicando la fórmula para nuestra plataforma;

$$AIFS = 2 \times 9 + 10 = 28\ \mu s.$$

El tiempo de Backoff depende del tiempo de slot y de un número aleatorio comprendido entre 0 y CW; siendo CW un número comprendido entre CWmin y CWmax. En concreto; la norma define este tiempo de la siguiente manera:

$$BackoffTime = Random(0,CW) \times aSlotTime \quad (3)$$

Donde;

Random(0,CW) = entero aleatorio obtenido de una distribución uniforme en el intervalo [0,CW], donde CW es un entero dentro del rango de valores característicos CWmin ≤ CW ≤ CWmax.

En nuestro escenario; el punto de acceso fijará para el tráfico de la estación; CWmin = 7 y CWmax = 1023 en la categoría de acceso Best Effort. Teniendo en cuenta la ecuación anterior y que en un principio CW = CWmin; tomamos un valor promedio de este tiempo con el siguiente cálculo:

$$T_backoff = 7/2 \times 9 = 31.5\ \mu s.$$

El cálculo de tiempo de transmisión de un paquete es simple atendiendo a la fórmula que especifica la norma para este caso:

$$Tx_time = T_preamble + Tsignal + Tsym \times Ceiling((16 + 8 \times LENGTH + 6)/Ndbps) + Signal\ Extension \quad (4)$$

Donde: T_preamble = 16 µs; Tsignal = 4 µs; T_sym = 4 µs. y Signal Extension = 6 µs. Además; Ndbps depende de la tasa con la que estamos transmitiendo. Ndbps(54 Mbps) = 216 y N(6 Mbps) = 24.

Aplicando esta fórmula a un escenario concreto de los cuatro que hemos propuesto podemos obtener fácilmente el tiempo de transmisión de un paquete. Por ejemplo; para un paquete de 80 bytes de PSDU enviado a una tasa de 54 Mbps; obtenemos:

$$Tx_time = 16\ \mu s + 4\ \mu s + 4\ \mu s \times Ceiling((16 + 8 \times long + 6)/ Ndbps) + 6\ \mu s$$

Donde Ndbps = 216 y long = 80 bytes;

$$Tx_time = 20 + 4 \times 4 + 6 = 42\ \mu s$$

Por último; nos queda computar el tiempo de recepción del asentimiento. El estándar 802.11 especifica que una trama ACK debería ser transmitida a una tasa igual o menor que la tasa de transmisión del paquete que es asentido y además esté dentro del conjunto básico de tasas de esa comunicación. Dicho conjunto "basic rate set" es definido en la norma como el conjunto de tasas que todas las estaciones en una BSS son capaces de usar para recibir tramas del medio. Como nuestra red está formada únicamente por dos estaciones sin limitaciones en cuanto a tasas y sin presencia de estaciones "NonERP"; el conjunto de tasas, en nuestro caso son 1, 2, 5.5, 11, 6, 12, 18, 24, 36, 48 y 54 Mbps; de forma que si un paquete de datos es transmitido a 24 Mbps; la tasa por defecto correspondiente a su ACK es 24 Mbps. Aplicando esto para el caso del supuesto anterior (54 Mbps) y conociendo que la trama ACK tiene una PSDU de 14 octetos; podemos desarrollar lo siguiente:

$$T_ack = 20 + 4 \times Ceiling((16 + 8 \times long + 6)/Ndbps) + 6$$

Donde Ndbps = 216 (correspondiente a la tasa de 54 Mbps) y long = 14 bytes;

$$T_ack = 20 + 4 \times 1 + 6 = 30\ \mu s$$

Ya disponemos de todos los términos necesarios para calcular el tiempo completo de transferencia:

T_Transf.(54 Mbps y 80 bytes) = T_backoff (31,5) + DIFS (28) + Tx_time (42) + SIFS (10) + T_ack (30) = 141.5 µs

Haciendo lo propio para cada escenario; obtenemos los siguientes tiempos de transferencia:

Tabla 5: Tiempos de transferencias teóricos

	80 bytes - 6 Mb	80 bytes - 54 Mb	800 bytes - 6 Mb	800 bytes - 54 Mb
Teórico	257.5	141.5	1217.5	245.5

Estos resultados no son definitivos ya que debemos tomar en consideración el carácter especial del tiempo backoff en redes no saturadas, como es nuestro caso. Según la propia norma 802.11 [1] y las referencias [2], [3] y [4]; el proceso de backoff debe efectuarse siempre después del final de cada transmisión; incluso si no existen paquetes en la cola. En nuestro escenario se introduce un paquete en cola cada 100 ms y es transmitido en el peor de los casos en una centésima parte de dicho tiempo. Por tanto; la cola permanecerá vacía la mayor parte del tiempo dando lugar a efectuarse el proceso completo de "post-backoff". Debido a ello; este elemento no debe ser tenido en cuenta para el cómputo global del tiempo de transferencia pues tenemos en cuenta que el proceso de backoff se efectúa seguidamente de la recepción del asentimiento quedando a la espera de una llegada del paquete a la capa mac para ser transmitido directamente después de un tiempo DIFS. Por tanto; descontando el tiempo de backoff (31,5) del cálculo teórico para cada uno de los escenarios, recogiendo además los resultados experimentales y su diferencia obtenemos la siguiente tabla:

Tabla 6: Comparación teórico – experimental

	80 bytes - 6 Mb	80 bytes - 54 Mb	800 bytes - 6 Mb	800 bytes - 54 Mb
Teórico	226	110	1186	214
Experimental (95%)	258 +/- 3	140 +/- 4	1217 +/- 5	244 +/- 6
Diferencia	32	30	31	30

Podemos observar como la diferencia entre el modelo teórico y las pruebas prácticas oscilan entre 30 y 32

µs; lo que muestra una precisión alta. Esta diferencia es asumible teniendo en cuenta el retraso que provoca el procesamiento del kernel de linux bajo el que se están llevando a cabo las pruebas. Sin embargo; esta diferencia oscilante entre 30 y 32 µs sugiere que en el caso real de nuestra plataforma; el tiempo de backoff sí se está llevando a cabo dentro del proceso de transferencia del paquete (antes del periodo DIFS) pues coincide prácticamente con dicho tiempo (31,5 µs) ya calculado anteriormente. Teniendo en cuenta este componente en el cálculo teórico; la precisión sería muy alta con una diferencia de 1,5 µs en el peor de los casos.

También es útil analizar los datos teóricos y los experimentales de las muestras que han sufrido retransmisiones. Teniendo en cuenta que el timeout de ack es de 19 µs y que la ventana de contención aumenta prácticamente el doble (2 x (CW + 1) - 1); podemos calcular el tiempo de transferencia computando los siguientes elementos de la misma forma ya planteada anteriormente:

$$T_Transf\ (retry = 1). = DIFS + Tx_time + Tout_ack + T_backoff + AIFS[AC] + Tx_time + SIFS + T_ack \quad (5)$$

Realizando este cálculo para cada uno de los escenarios y comparándolos con las tramas que fueron retransmitidas en una ocasión; obtenemos los siguientes datos:

Tabla 7: Comparación teórico – experimental con reintentos

	80 bytes - 6 Mb	80 bytes - 54 Mb	800 bytes - 6 Mb	800 bytes - 54 Mb
Tramas reenviadas una vez	3	9	13	12
Teórico	458.5	266.5	2378.5	470.5
Mediana Experimental	478	248	2387	458.5
Diferencia	19.5	18.5	8.5	12

Otra comparativa interesante es la relativa a los retrasos producidos por las tramas que originan las tramas anómalas; es decir; podemos calcular el retraso teórico que produciría las tramas de gestión que provocan las muestras anómalas y compararla con aquellas que se han visto afectadas por dichas tramas. De esta forma procedemos al cálculo teórico de cada una de las tramas de gestión monitorizadas con airodump-ng; de la misma forma que hemos aplicado anteriormente:

Tr_Beacon = 192 µs (cabecera cck) + 1032 µs (129 bytes a 1Mb) + 10 µs (SIFS) = 1234 µs.

Tr_ProbeRequest = 192 µs (cabecera cck) + 392 µs (49 bytes a 1Mb) + 10 µs (SIFS) = 584 µs.

Tr_ProbeResponse = 192 µs (cabecera cck) + 912 µs (114 bytes a 1Mb) + 10 µs (SIFS) = 1114 µs.

La siguiente tabla muestra la comparación entre el retraso teórico provocado por una Beacon y los máximos experimentados en cada escenario:

Tabla 8: Comparación teórico – experimental beacon

	80 bytes - 6 Mb	80 bytes - 54 Mb	800 bytes - 6 Mb	800 bytes - 54 Mb
Teórico Beacon	1234			
Max. Experimental	1432	1590	1055	1660

Veamos lo análogo para la trama Probe Request. En este caso debemos tener en cuenta que esta trama puede aparecer individualmente, en pares o múltiples veces.

Tabla 9: Comparación teórico – experimental probe request

	80 bytes - 6 Mb	80 bytes - 54 Mb	800 bytes - 6 Mb	800 bytes - 54 Mb
Teórico Probe Request	584			
Max. Experimental	2027	205	27	1245

Por último; nos queda la comparativa de la trama de gestión Probe Response. Esta trama sólo ha sido registrada en el escenario caracterizado por Rb = 6 Mbps y PSDU = 80 bytes.

Tabla 10: Comparación teórico – experimental probe response

	80 bytes - 6 Mb
Teórico Probe Response	1114
Max. Experimental	1185

6 CONCLUSIONES

Mediante este artículo queda demostrada una fiabilidad aceptable de la plataforma creada bajo la norma 802.11. Se ha decidido estudiar la variable tiempo por su importancia en el objetivo final de esta plataforma y por su sencilla monitorización entre otros motivos. Hemos llevado a cabo pruebas en cuatro escenarios con distintos parámetros, recopilados sus resultados y estudiado sus estadísticas. Por otro lado hemos desarrollado unos cálculos teóricos siguiendo fielmente la normativa para poder realizar una comparativa con los anteriores resultados. En dicha comparación; los resultados se asemejan mucho; existiendo una diferencia máxima de 32 µs (sin contar con el tiempo de backoff) lo que supone un error entorno al 12 % para el caso más desfavorable. En otros escenarios el error llega a oscilar entre el 2 % del valor de tiempo.

Además la precisión es muchísimo mayor si tenemos en cuenta el componente backoff en nuestros cálculos teóricos; llegando incluso al 99% de precisión.

Agradecimientos

Los autores agradecen al Proyecto de Excelencia de la Junta de Andalucía. (P09-AGR-4785) y al Proyecto de Investigación y Desarrollo Tecnológico del Ministerio de Ciencia e Innovación (DPI2010-19154) por subvencionar este trabajo.

Referencias

[1] IEEE Computer Society, "IEEE Standar for Information technology – Telecommunications and information exchange between systems," IEEE Std 802.11 – 2007 Revision, http://standards.ieee.org/getieee802/download/802.11-2007.pdf

[2] Jahanzeb Farooq y Bilal Rauf, "Implementation and Evaluation od IEEE 802.11e Wireless LAN," Febrero 2006, http://folk.uio.no/paalee/referencing_publications/ref-wlanmodel-farooq-thesis-2006.pdf.

[3] Wei Zhang, Jun Sun, Jing Liu, Hai-bin Zhang, "Performance analysis of IEEE 802.11e EDCA in wireless LAN," Journal of Zhejiang University SCIENCE A ISSN 1009-3095 (Print); ISSN 1862-1775 (Online), October 2006, http://www.springerlink.com/content/f3u73103346710n7/fulltext.pdf

[4] Paal E. Engelstad and Olav N. Ostervo, "Queueing Delay Analysys of IEEE 802.11e EDCA", http://hal.inria.fr/docs/00/05/46/41/PDF/15-Engelstad.pdf.

LA RECTA DE CHEBYSHEV EN LA APROXIMACIÓN AUTOMÁTICA A CRÍTICO DE LOS REACTORES NUCLEARES.

J. L. Fernández Marrón
E.T.S.I. de Informática, UNED, C/Juan del Rosal 16, 28040 Madrid,
jlmarron@dia.uned.es

Juan Blázquez Martínez, José Miguel Barcala Riveira
CIEMAT, Avda. Complutense, 22, Madrid-28040,
juan.blazquez@ciemat.es, miguel.barcala@ciemat.es

Resumen

El proceso de estimar la criticidad inicial de un reactor nuclear se automatiza con facilidad, pues es costumbre utilizar curvas de mínimos cuadrados para la extrapolación. El procedimiento permite estimar la masa crítica y los errores en la estima. En rigor, el cálculo del error debe tener en cuenta la correlación entre los errores de los parámetros de la regresión, lo cual complica su estima. El estudio teórico del proceso confirma que en los tramos finales del mismo, la recta es la mejor opción, la parábola ya sobre-parametriza la predicción, pero para la estima del error, la recta de Chebyshev tiene grandes ventajas, por lo que se propone su utilización. Con los errores que proporciona se puede determinar cuando estamos en la fase final de la aproximación a crítico, y así garantizar la Seguridad del procedimiento.

En el trabajo se ha puesto a punto un programa para el cálculo de la recta de Chebyshev para datos experimentales que se adquieren y procesan en línea. Esta recta permite además evaluar los errores en la estima, siendo una distribución de probabilidad uniforme.

También se ha aplicado con éxito a datos reales, para determinar la criticidad del Reactor CORAL-I según se añadía masa de Uranio. La constancia del error en la estima es la garantía de que el proceso es seguro.

Finalmente se ha diseñado un equipo portátil para la medida de la criticidad de reactores, aplicando la técnica de la recta de Chebyshev para la determinación del punto de criticidad.

Palabras Clave: automatización, sistema de adquisición de datos, criticidad reactores nucleares, Recta de Chebyshev.

1. INTRODUCCIÓN

En 1988 alcanzó la primera criticidad el reactor nuclear de Vandellós II. Desde entonces no se ha llevado a crítico ningún nuevo reactor, de manera que es muy probable que si se construyera una nueva Planta, no habría técnicos nacionales que pudieran acreditar experiencia en el proceso empírico de la primera criticidad [1].

Cada vez que hay una recarga de combustible, el reactor ha de ponerse crítico para generar electricidad; sin embargo, los códigos de cálculos son tan potentes hoy en día, que no es necesario proceder empíricamente para buscar la carga crítica. La confianza en los cálculos es absoluta ya que los códigos están plenamente validados para este tipo de reactores.

Sin embargo, para los nuevos tipos de reactores, hoy bajo diseño, denominados prototipos de la Generación IV, los códigos no están validados y será necesario usar el viejo procedimiento empírico de la primera criticidad [2]. En este procedimiento se representa la inversa de la población neutrónica frente a un parámetro de criticidad, que tomaremos como la masa cargada de combustible. La masa crítica se obtiene por extrapolación cuando la inversa de la población neutrónica tiende a cero. En los tramos finales de la aproximación a crítico, la representación de la inversa de la población neutrónica frente a la masa de combustible es una línea recta.

Con los medios informáticos actuales el proceso se automatiza con facilidad, pues es costumbre utilizar curvas de mínimos cuadrados para la extrapolación. El procedimiento permite estimar la masa crítica y los errores en la estima. En rigor, el cálculo del error debe tener en cuenta la correlación entre los errores de los parámetros de la regresión, lo cual complica su estima. El estudio teórico del proceso confirma que en los tramos finales del mismo, la recta es la mejor opción [3]; la parábola ya sobre-parametriza la

predicción, pero para la estima del error, la recta de Chebyshev tiene grandes ventajas [4,5], por lo que se propone su utilización. Con los errores se puede determinar cuando estamos en la fase final de la aproximación a crítico, y así garantizar la Seguridad del procedimiento.

En este trabajo se han empleado las medidas reales de aproximación a crítico del reactor experimental CORAL-I, ya decomisionado [6]. La automatización se hace usando acoplando, a los detectores de neutrones, un equipo de uso general que pueda ser utilizado en cualquier otro menester, ya que la aplicación de criticidad es inusual. La programación se realiza en MATLAB y Lab-View, a fin de que pueda ser usado en cualquier PC.

2 FUNDAMENTO TEÓRICO

La población neutrónica de un reactor en estado subcrítico necesita de una fuente externa de neutrones de intensidad constante S para mantenerse estable. Una buena candidata es la fuente de Americio-Berilio de 3 Ci de intensidad. Esta fuente suministra 6.6 x 10^6 n/s y se mantiene relativamente estable en el tiempo, con una semivida de 458 años. Para detectar los neutrones se usan contadores proporcionales de BF_3 con una eficiencia, e_D, del orden de 10^{-6} cuentas/n.

La relación entre la población neutrónica N y la fuente S es [7]:

$$N = (S\ell)\frac{1}{1-k} \qquad (1)$$

en donde ℓ es la vida media de los neutrones en el reactor, del orden de µs. La medida de la subcriticidad viene dada por la constante de multiplicación k, que es inferior a la unidad cuando el reactor está subcrítico. Esta magnitud es la que calculan los códigos nucleares, pero la precisión del cálculo depende de la precisión en los datos nucleares. Como mínimo hay que calcular k con una precisión del 0.1%, lo cual no puede siempre asegurarse y hay que recurrir a procedimientos empíricos para los prototipos de reactores.

Un detector no mide realmente N sino la cantidad:

$$N^* = \frac{e_D}{?\,t}\int_0^{?t}\frac{N}{\ell}dt \qquad (2)$$

siendo ?t el tiempo de recuento de los neutrones. Haciendo $S^* = e_D S$, resulta de (1) que:

$$\frac{1}{N^*} = \frac{1}{S^*}(1-k) \qquad (3)$$

Si se representa 1/N* frente a k se obtiene una línea recta de pendiente 1/S*. Además, cuando k tiende a 1, reactor crítico, entonces 1/N* tiende a cero. En la aproximación empírica a la criticidad se representa 1/N* frente a otra magnitud medible, que sea una función monótona de k, tal como la masa de combustible, la concentración de ácido bórico, el buckling, etc. Elegimos la masa cargada de combustible para el proceso, pero éste sería semejante con cualquier otra elección.

Dado que la k es una función monótona de la masa, se hace un desarrollo de Taylor en torno a k = 1, resultando:

$$k = 1 + (\frac{dk}{dM})(M - M_c) + \frac{1}{2!}(\frac{d^2k}{dM^2})(M - M_c)^2 + ... \qquad (4)$$

La magnitud M_c es la masa crítica. Cuando estemos en la cercanía de la criticidad, se puede despreciar el término parabólico, y entonces la Eq. 3 se escribe como:

$$\frac{1}{N^*} = \frac{1}{S^*}(\frac{dk}{dM})(M_c - M) \qquad (5)$$

si se representa 1/N* frente a M, se obtiene una línea recta en los tramos finales que corta al eje de abscisas en la masa crítica. En los tramos iniciales de la aproximación a crítico, la representación se alejará de una recta por causa del término cuadrático en la Eq. 4. La concavidad de la curva dependerá del signo del término cuadrático.

3 LA RECTA DE CHEBYSHEV

Cuando se ajusta un polinomio a un conjunto de puntos experimentales, la distancia de los puntos medidos a los ajustados suele presentar una distribución gaussiana de probabilidad. Si en vez de proceder por los mínimos cuadrados se utilizan los polinomios ortogonales de Chebyshev, dicha distancia tiende a seguir una distribución de probabilidad uniforme, por ello se dice que los polinomios de Chebyshev "reparten" el error.

Si solamente se utilizan tres puntos, la recta de Chebyshev se construye de manera que la distancia de cada punto a la recta es constante. Si en vez de tres puntos se dispone de cuatro, entonces se construye la parábola de Chebyshev con la misma propiedad de error constante; y así sucesivamente.

Sean (x_i,y_i) , i = 1,2,3 los tres puntos por entre los cuales discurre la recta de Chebyshev; sea e la distancia vertical, constante, de cada punto a la recta; si la ecuación de la recta es: y = ax+b, los coeficientes a y b se obtienen de la resolución del sistema lineal de tres ecuaciones con tres incógnitas a, b y e:

$$\begin{aligned} y_1 - ax_1 - b + e = 0 \\ y_2 - ax_2 - b - e = 0 \\ y_3 - ax_3 - b + e = 0 \end{aligned} \qquad (6)$$

Obsérvese la alternancia de signos de e. La solución es sencilla; la pendiente corresponde a la de los puntos extremos:

$$a = \frac{y_3 - y_1}{x_3 - x_1} \qquad (7)$$

Las restantes soluciones se obtienen de inmediato:

$$\begin{aligned} b &= \frac{y_1 + y_2}{2} - a\frac{x_1 + x_2}{2}; \\ \varepsilon &= a\frac{x_1 - x_2}{2} - \frac{y_1 - y_2}{2} \end{aligned} \qquad (8)$$

Estamos interesados en el punto de corte x_c con el eje de abscisas:

$$x_c = -\frac{b}{a} \qquad (9)$$

La incertidumbre de este punto se obtiene suponiendo una distribución uniforme de probabilidad:

$$s_c = \frac{|e|}{a\sqrt{3}} \qquad (10)$$

No hay duda que este procedimiento es mucho mas rápido que la aproximación por mínimos cuadrados, y en este caso, que la aproximación es a una recta nos proporciona el resultado final y su error de una manera muy rápida y segura.

El proceso de aproximación a crítico consiste en construir rectas de Chebyshev con los tres últimos puntos disponibles y calcular el corte con el eje de las masas. Cuando las sucesivas extrapolaciones caigan dentro del último intervalo de confianza, estaremos razonablemente seguros de que nos aproximamos a la predicción por una línea recta.

4. RESULTADOS DE LA CRITICIDAD DEL REACTOR CORAL-I

CORAL-I es un reactor rápido experimental de potencia cero [6]. Consta de un núcleo cilíndrico de uranio metálico enriquecido al 90%. El núcleo está dividido en dos mitades aproximadamente iguales. En la mitad fija hay 11 Kg aproximadamente, mientras que la carga de la mitad móvil es variable. La parte móvil se carga con discos pequeños de 114 mm de diámetro y altura variable.

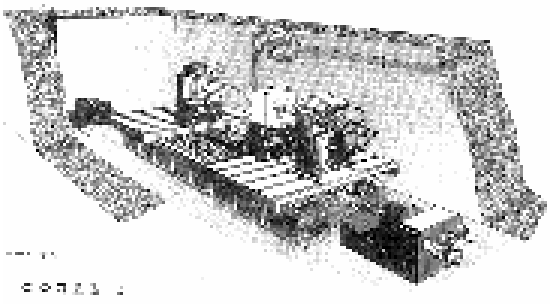


Figura1: Vista general del Reactor CORAL-I

El núcleo está rodeado de un reflector paralelepipédico de uranio natural metálico 55 cm de lado.

El control es por fugas neutrónicas, de manera que al introducir completamente los elementos de control, aumenta el valor de k.

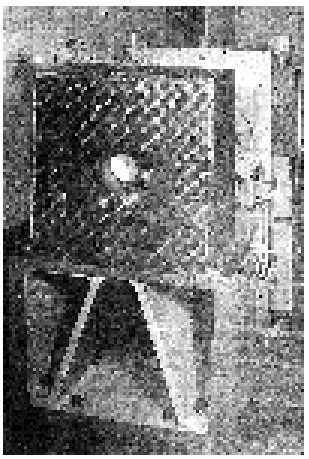


Figura 2: Interior del Reactor CORAL-I

En la Figura 2 se observan la parte móvil del núcleo y los diversos elementos de control en forma de barras. El proceso de aproximación a crítico consiste en ir agregando pequeños discos de uranio enriquecido, tal como se muestra en la figura, hasta que se alcanza la criticidad. El proceso se hace con el

sistema de control introducido y luego se repite con el mismo extraído.



Figura 3: Elemento del Reactor CORAL-I

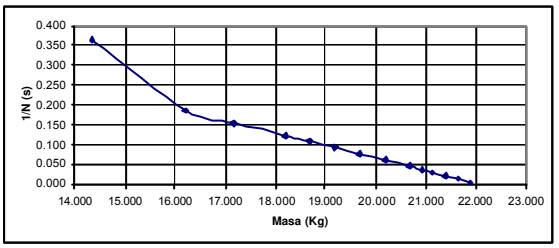


Figura 4: Aproximación a crítico del Reactor CORAL-I

En la Figura 4 se han representado las medidas de aproximación a crítico obtenidas con un contador de F_3B. Se parte de reactor muy subcrítico, cargando 14.3 Kg , luego 16.2 Kg y finalmente 17.1 Kg.

La recta de Chebyshev con estos tres primeros puntos predice una masa crítica muy inferior a la real; pero si añadimos una cuarta carga, 18.2 Kg,, la recta de Chebyshev predice una carga crítica mayor que la verdadera. Esto puede ser peligroso, por lo que es procedimiento consiste en cargar sucesivamente pequeños incrementos de masa hasta asegurarse que la predicción de la masa crítica está dentro del intervalo de confianza.

5 SISTEMA DE MEDIDA:

5.1 METODO DE CALCULO

El método para obtener la recta de Chebyshev y la previsión de criticidad parte de la adquisición de la cantidad de neutrones medida por algún detector de neutrones que posea un número suficiente de cuentas. Los datos con los que se trabaja es la variable que incrementa la criticidad (en nuestro caso masa de Uranio) frente a la inversa del número de cuentas de los detectores de neutrones.

Se adquieren y representan los dos primeros valores (variable de criticidad en abscisas e inversa del número de cuentas en los detectores en ordenadas) y se calcula la recta que pasa por ellas. Su cruce con el eje de abscisas nos da el valor de la variable (en este caso masa de Uranio) que hará crítico el reactor. A continuación se adquiere los siguientes valores y como ya existen tres podemos calcular la recta de Chebyshev, su abcisa en el origen y el error con respecto a ella de los valores medidos. Repitiendo esto para cada valor nuevo adquirido añadiendolo al cálculo, despreciando el más antiguo, continuamos calculando los valores de la abscisa en el origen (valor que hace critico el reactor) y errores de los valores medidos.

5.2 APLICACIÖN A LOS DATOS DEL REACTOR CORAL-I

Al introducir los datos que poseemos del reactor CORAL-I, se trata de comprobar el funcionamiento del programa. Los datos se van leyendo de forma consecutiva como si fueran siendo muestreados a intervalos de tiempo.

El programa calcula la recta de manera rápida presentándola y estimando el valor de la masa crítica (valor del corte de la recta con el eje de abscisas). Además calcula el error de las medidas (el mismo para todas) con la recta de Chebyshev.

Las desviaciones de los datos con respecto a la Recta de Chebyshev son, a partir de la quinta medida, inferiores al 3 por mil, por lo que la determinación de la masa crítica a partir de ese instante es lo suficientemente precisa.

En la Figura 5 se representan las dos rectas de Chebyshev una con el sistema de control introducido y otra con el mismo extraído. También se representan los valores extrapolados que hacen crítico el reactor en ambos casos.

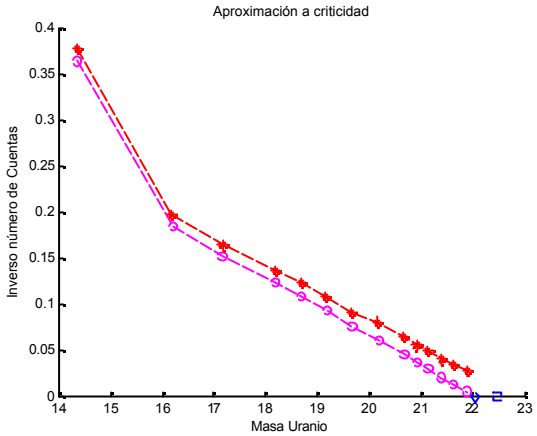


Figura 5: Representación de de la inversa de la población neutrónica frente a la masa de combustible. Se representan también los valores extrapolados

5.3 DISEÑO DEL EQUIPO DE MEDIDA

Dado que el equipo se usará solo durante el arranque se ha diseñado sobre un equipo que pueda utilizarse en otras tareas y por ello que sea lo más general posible.

El equipo se ha diseñado sobre el modulo de adquisición de datos NI USB-6210 [9] que está alimentado por bus USB y ofrece una alta precisión a unas buenas velocidades de muestreo. Ofrece 16 entradas analógicas, una velocidad de muestreo de un solo canal de 250 muestras por segundo, cuatro líneas de entrada digital, cuatro líneas de salida digital, cuatro rangos de entrada programable (±0.2 V a ±10 V) por canal, disparo digital y dos contadores-temporizadores de 32 bits a 80 MHz.

El USB-6210 está diseñado específicamente para aplicaciones móviles ya que es plug-and-play, con ello se minimiza el tiempo de configuración y montaje. Además no requiere conexión externa ya que se alimenta directamente del USB.



Figura 6: USB-6210

El software de control de este dispositivo es compatible con LabVIEW 2011, que es sobre el que se ha desarrollado la aplicación. La pantalla de presentación de resultados es muy sencilla ya que solo debe mostrar la evolución de la recta de Chebyshev y los valores actual y previsto de la variable que determina la criticidad.

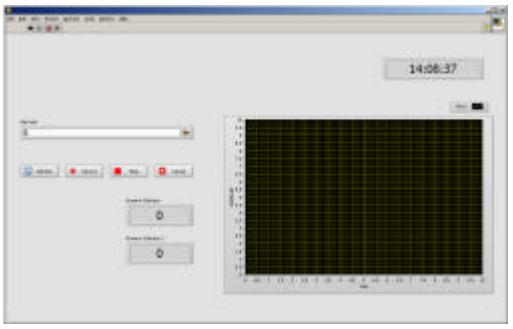


Figura 7:Pantalla de LabView del equipo de medida

6. CONCLUSIONES

Se ha puesto a punto un programa para el cálculo de la recta de Chebyshev para datos experimentales que se adquieren y procesan en línea. Esta recta permite además evaluar los errores en la estima, siendo una distribución de probabilidad uniforme.

Se ha aplicado con éxito a datos reales, para determinar la criticidad del Reactor CORAL-I según se añadía masa de Uranio. La constancia del error en la estima es la garantía de que el proceso es seguro.

Se ha diseñado un equipo portátil para la medida de la criticidad de reactores, aplicando la técnica de la recta de Chebyshev para la determinación del punto de criticidad.

Referencias

[1] IAEA, (2004) The nuclear power industry´s aging workforce: transfer of Knowledge to the next generation, IAEA-TENDOC-1359, Vienna.

[2] Schultz, (1959) Control of nuclear reactor and power plants, Mc Graw Hill, Chapter 2, pp 15-16.

[3] Progio A.E., (1976) Experimental reactor physics, John Willey and Sons Inc, Chapter 10, pp 607-611.

[4] Nikiforov A.F. et al, (1991) Classical orthogonal polynomials of a discrete variable, Chapter 1.

[5] Scheid F., (1968) Numerical analysis, Schaum outline series, Chapter 22, pp 268-269.

[6] Blazquez J., (1985) Medidas de coeficientes de acoplo en el reactor Coral-I,Ph.D.Thesis, Facultad Ciencias Físicas. Universidad Complutense Madrid. Capítulo 3.1.

[7] Hetrick D.L., (1971) Dinamics of nuclear reactors, The university of Chicago Press, Chapter 2, pp 18-19.

[8] Walter A.C, et al, (1981) Fast breeders reactors, Pergamon Press, Chapter 5.

[9] National Instruments (2009) NI-USB-621X User Manual

MONITORIZACIÓN Y CONTROL REMOTO DE CONSUMO ENERGÉTICO MEDIANTE COMUNICACIÓN WIFI

Isidre Ortega Gómez
Dpt. Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, Edifici D4 Campus Nord, 08034 Barcelona, isidre.orgo@gmail.com

Sergio Zarza Sánchez, José Antonio Lázaro Villa, Rosa M. Fernández-Cantí
Dpt. Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, Edifici D4 Campus Nord, 08034 Barcelona, {szarza,jose.lazaro,rfernandez}@tsc.upc.edu

Resumen

En esta comunicación se presenta un sistema modular para la monitorización y control remoto del consumo de energía. En este caso en concreto nos hemos decantado por la energía eléctrica. También, por simplicidad, nos centraremos en el ámbito doméstico. Los diferentes módulos de los que se compone este sistema nos permitirán realizar una monitorización exhaustiva del consumo y un control óptimo del gasto energético.

Palabras Clave: Monitorización, control remoto, comunicación WIFI, control de consumo, eficiencia energética.

1 INTRODUCCIÓN

El control de consumo ya sea de corriente eléctrica como de agua o gas es un aspecto imprescindible en cualquier instalación tanto industrial como doméstica. Una correcta monitorización nos permite detectar posibles puntos de fuga, averías o simplemente partes de la instalación que necesitan de un mantenimiento o sustitución. La combinación entre supervisión y control permite mantener un gasto energético adecuado consiguiendo a la vez un ahorro económico sustancial [4]. En este trabajo presentamos un sistema modular para la monitorización y control de consumo centrándonos en el caso concreto de la corriente eléctrica. Además de la monitorización, el sistema proporciona la posibilidad de ejercer un control remoto consistente en la desconexión de la red eléctrica de los equipos que hayan sobrepasado un determinado umbral de consumo, debido a algún tipo de problema técnico o por alcanzar un cierto nivel definido por el usuario. Tanto la supervisión como el control remoto se pueden realizar desde cualquier tipo de terminal con conexión a internet (ordenador, móvil, tableta,…) [1, 9]. Aunque en el presente trabajo nos centraremos en el ámbito doméstico, la solución aquí presentada también se puede implementar en otras aplicaciones donde los requerimientos de seguridad y ahorro energético sean relevantes. Es por ejemplo el caso de instalaciones industriales o equipamientos públicos (como universidades, polideportivos, hospitales,…).

Actualmente, existen diversas soluciones comerciales para la monitorización y control de consumo de corriente eléctrica como, por ejemplo, Fibaro Wall Plug, Efergy, Onzo, o sistemas OpenHardware como Energino o OpenEnergyMonitor. La mayoría de las soluciones inalámbricas emplean un medio de comunicación vía radio 433MHz, Bluetooh o Z-Wave. En otras aplicaciones, como el control de fluidos, es posible encontrar en el mercado soluciones que utilizan comunicaciones por GSM.

Con respecto a los protocolos de comunicación en domótica cabe destacar, entre otros, el KNX (de la *Konnex Association*) y Z-Wave, que llevan tiempo en el mercado y son sistemas muy probados. Sin embargo, en el presente proyecto se ha optado por realizar la comunicación vía WIFI (*Wireless Fidelity*). La mayor ventaja del estándar de comunicación WIFI es que nos permite abaratar económicamente el conjunto del sistema con respecto a los anteriores. Además se trata de un estándar de comunicación muy desarrollado y con gran aceptación entre el personal técnico. Esto facilita la expansión del sistema a otras aplicaciones, con la posibilidad de adaptación a otro tipo de sensores y poder así monitorizar y controlar cualquier tipo de consumo energético como, por ejemplo, el caudal en tuberías. El protocolo WIFI también nos permite mayor margen a la hora de encriptar los datos transmitidos entre los diferentes módulos de nuestro sistema.

Finalmente, en el sistema diseñado se ha buscado tener un consumo eléctrico razonable de todo el conjunto de componentes, por lo que no repercutirá en la factura del usuario. Este punto es de vital importancia ya que el objetivo de la implantación de este sistema en cualquier tipo de instalación es el ahorro y, si los componentes tuvieran un consumo

elevado, el ahorro conseguido con la monitorización y control de los diferentes equipos lo veríamos mermado por el consumo de nuestro sistema.

La presente comunicación está organizada de la siguiente manera. En la Sección 2 se describen los principales módulos que forman el sistema y se detallan sus características técnicas más relevantes. La Sección 3 se centra en el protocolo de comunicación WIFI y en la configuración de la VLAN. En la Sección 4 se describe la interfaz con el usuario, la gestión de alarmas y se detalla la comunicación con la base de datos. La Sección 5 ilustra la viabilidad del proyecto mediante un estudio económico. Y, finalmente, en la Sección 6 se presentan las conclusiones de este trabajo.

2 DESCRIPCIÓN DEL SISTEMA

La Figura 1 muestra la representación del conjunto implementado para monitorizar el consumo de la energía en una vivienda piloto.



Figura 1: Módulos que forman el sistema de monitorización y control.

El sistema consta de un conjunto de enchufes inteligentes (equipos de monitorización) a los cuales se les puede conectar cualquier tipo de equipo que funcione con corriente eléctrica y disponga de una toma europea tipo E. Dichos enchufes están interconectados mediante WIFI en una red privada y doméstica a un pequeño ordenador que realiza la función de servidor web y en el que se almacenan y tratan los datos adquiridos de las bases de enchufes del sistema. Este servidor nos permite monitorizar y controlar las bases de enchufes de nuestro sistema vía web. Para ello está conectado a un enrutador o módem que es el que le proporciona la conexión a internet. El propósito es que el usuario pueda monitorizar el consumo *online* y además pueda controlar remotamente las bases para desconectarlas desde cualquier dispositivo con conexión a internet.

Los siguientes apartados describen con más detalle cada uno de los elementos que componen el sistema.

2.1 ENCHUFES INTELIGENTES

Los enchufes inteligentes están diseñados para ser conectados en las tomas europeas tipo E. Típicamente, en aplicaciones domésticas, a estas bases se les puede conectar cualquier tipo de electrodoméstico o bien una base con más enchufes a las cuales se les conectarán los electrodomésticos de mayor consumo tales como radiadores eléctricos, aire acondicionado, etc.

Los diferentes elementos que integran los enchufes inteligentes se muestran en la Figura 2. Estos son un convertidor AC/DC, dos tarjetas de monitorización de tensión y corriente respectivamente, una tarjeta de control y una tarjeta de comunicación.

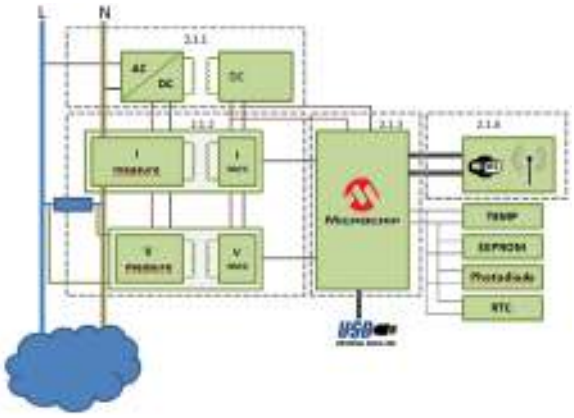


Figura 2: Componentes del enchufe inteligente

2.1.1 Convertidor AC/DC

Para alimentar la electrónica con los niveles de tensión correctos, el enchufe inteligente incluye de un convertidor de doble etapa (ver Fig. 3). En la primera etapa, se dispone de un convertidor Buck AC/DC no aislado para convertir la tensión alterna de la red eléctrica en un nivel de tensión de 5V en continua. La segunda etapa se basa en un convertidor DC/DC con una relación de transformación 1:1, a fin de garantizar un aislamiento galvánico entre las magnitudes a medir y el sistema de control.

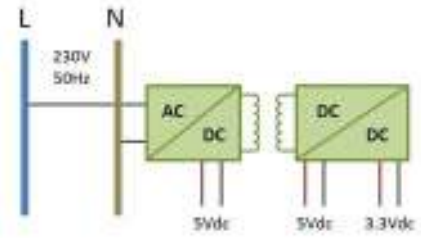


Figura 3: Convertidor AC/DC de doble etapa

2.1.2 Tarjetas de monitorización

La monitorización del consumo eléctrico se realiza mediante dos circuitos integrados diseñados para tal propósito por los fabricantes Allegro y Texas Instruments, uno para la medición de corriente y otro para la medición de voltaje respectivamente [8]. La particularidad de estos circuitos es que ambos disponen de un aislamiento eléctrico entre la magnitud medida y el transductor de señal, evitando de este modo que cualquier perturbación en la línea eléctrica dañe el equipo.

La medida de tensión la realiza el circuito integrado AMC1100DUB mediante un divisor resistivo conectado entre la línea y la fase de la tensión de red. La salida del dispositivo es una tensión diferencial proporcional a la magnitud medida (ver Fig. 4).

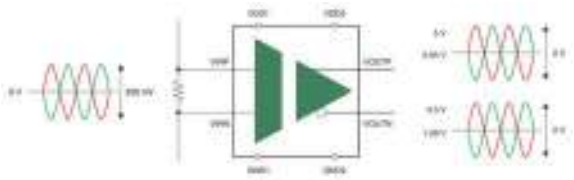


Figura 4: Medida de la tensión de red mediante Shunt y salida diferencial

La medida de corriente se basa en un sensor de efecto Hall acondicionado en un circuito integrado, ACS712, con una sensibilidad de 100mV/A.

El enchufe inteligente dispone de un sensor de temperatura para compensar el error y la no linealidad en la medida de la corriente que será corregido mediante *software* por el sistema de control.

Adicionalmente, se le ha puesto al enchufe una célula fotovoltaica, que nos permitirá controlar la luminosidad de la sala en la que se ubique la base. Esto hace que el enchufe inteligente adquiera una funcionalidad adicional a un bajo coste de implementación.

2.1.3 Tarjeta de control

Formada por un microcontrolador de bajo consumo del fabricante Microchip, incorpora los periféricos necesarios para efectuar la conversión ADC de las señales de tensión V y corriente I facilitadas por los circuitos presentados en el apartado anterior. El sistema de control envía la información al módulo WIFI con los datos obtenidos y procesados. Puesto que el objetivo es realizar un seguimiento del consumo eléctrico, la tarjeta de control muestrea las medidas de tensión $V(n)$ y corriente $I(n)$ proporcionadas por las tarjetas de monitorización y realiza los siguientes cálculos, donde N es el número de muestras [6]:

- Potencia activa

$$P_{activa} = \frac{1}{N}\sum_{n=1}^{N} V(n)\cdot I(n) \quad (1)$$

- Potencia aparente

$$P_{aparente} = V_{RMS}\cdot I_{RMS} \quad (2)$$

donde se definen los valores de tensión eficaz y corriente eficaz como $V_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} V(n)^2}$ y $I_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} I(n)^2}$, respectivamente.

- Potencia reactiva

$$P_{reactiva} = \sqrt{P_{aparente}^2 - P_{activa}^2} \quad (3)$$

- Factor de potencia

$$FP = \frac{P_{activa}}{P_{aparente}} \quad (4)$$

En instalaciones domésticas la compañía eléctrica nos aplica la tarifa vigente sobre la potencia activa consumida mensualmente. Por este motivo la información transmitida al servidor será la potencia activa.

Utilizando tres equipos podemos realizar la medida del consumo eléctrico en instalaciones trifásicas, donde las compañías eléctricas facturan sobre la potencia activa y aplican una bonificación o penalización de la potencia reactiva consumida. Por este motivo se ha incluido también la posibilidad de calcular y transmitir también la potencia reactiva.

2.1.4 Tarjeta de comunicación

Los datos de consumo calculados en el apartado anterior se envían en serie a una tarjeta de comunicación WIFI del fabricante Roving Networks, modelo RN-171. Esta tarjeta es la encargada de transmitir los datos al servidor web residente en el Raspberry Pi.

Los bajos consumos que aporta el enchufe inteligente se deben en parte a la posibilidad de disponer de esta tarjeta de comunicación integrada con canal de comunicación en 2.4-GHz IEEE 802.11b/g Tx/Rx y

consumos: en modo “sleep” de 4 uA, como Rx de 30 mA y variable para Tx dependiendo de la potencia de salida y como máximo 200 mA.



Figura 5: Tarjeta de comunicación

Para enlazar los datos con el servidor web el microcontrolador debe enviar la siguiente trama de texto por el puerto de comunicación:

```
http://www.webserver.com/ob.php?obvar=value
```

donde:

- *webserver*: nombre del servidor web implementado en la Raspberry Pi.
- *obvar*: alias del equipo.
- *value*: consumo de energía.

Por ejemplo, en el caso de un televisor en el comedor que estaría consumiendo 300W, la trama sería:

```
http://www.smartmeterupc.com/ob.php?TV=300
```

3 SISTEMA DE COMUNICACIÓN WIFI

Los diferentes enchufes inteligentes que componen el sistema envían y reciben señales mediante WIFI a un pequeño ordenador que hace la función de servidor, el cual está dentro de la misma VLAN (*Virtual Local Area Network*) que las bases de enchufes.

3.1 SERVIDOR DE COMUNICACIONES

En el presente trabajo se ha escogido como servidor una Raspberry Pi, que es un pequeño ordenador con las suficientes prestaciones para realizar las funciones de servidor requeridas [3]. Las funciones que realiza este servidor son las siguientes: servidor web, base de datos, interconectar entre sí la red interna y la red exterior.

En el servidor se ha montado como sistema operativo una distribución derivada de Debian llamada Raspbian [3]. A fin de que el servidor realice las funciones requeridas, se le han instalado los siguientes aplicativos (ver Fig. 6): Servidor sSMTP como servidor de correo, isc-dhcp-server como servidor de DHCP (*Dynamic Host Configuration Protocol*) , Apache como servidor web, MySQL (*Structured Query Language*) como base de datos. Para interconectar estas dos últimas aplicaciones, servidor web y base de datos, se ha optado por utilizar el lenguaje de programación PHP (*Hypertext Pre-processor*).

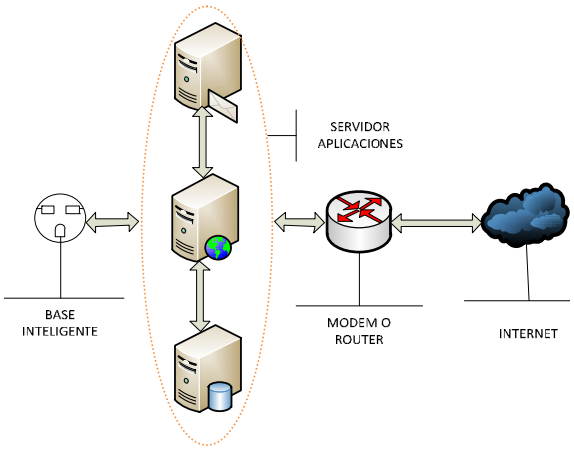


Figura 6: Aplicativos implementados

La base de enchufes tiene asignada una dirección MAC (*Media Access control*) mediante la cual el usuario le podrá asignar un nombre que nos permita identificar inicialmente el sistema de medida con un alias (ver Fig. 7). Por ejemplo, si tenemos dos tomas que monitorizan el consumo de los equipos del comedor, podríamos identificarlas con los siguientes alias, TV y Hi-Fi para la televisión y el reproductor de música, así el usuario puede identificar más fácilmente el consumo de cada uno de los equipos que tiene conectados a la red (ver Fig. 8).



Figura 7: Aplicación para inicializar e identificar cada una de las tomas.

Con el conjunto de todos estos aplicativos lo que conseguimos es realizar un entorno HTML amigable para que el usuario final pueda trabajar con todo el sistema de monitorización y control de energía de forma clara y fácil.

3.2 CONFIGURACIÓN DE LA VLAN

Las diferentes bases de enchufes inteligentes y el servidor se comunican mediante WIFI y forman una red de área local virtual (VLAN) privada. Esta solución ofrece ventajas en relación a la seguridad y

encriptación de los datos ya que podemos configurarla dentro de sus restricciones. Cada enchufe tiene un código o nombre, el cual podremos configurar dependiendo de la zona donde lo ubiquemos, de esta forma podremos seguir en línea el consumo de cada zona específica.

4 INTERFAZ CON EL USUARIO

En este apartado se describe la interfaz con el usuario y se comentan sus funcionalidades más relevantes.

4.1 PÁGINA WEB

Los objetivos a la hora de la programación han sido conseguir un entorno lo más amigable posible y que el usuario final no tenga ningún tipo de dificultad en el momento de realizar cualquier actividad con el sistema.

Desde este entorno el usuario es capaz de monitorizar tanto el consumo en tiempo real como el historial de éste. Adicionalmente, facilitando la tarifa que tiene contratada el usuario con su respectiva compañía, la aplicación nos muestra el gasto económico en cada momento.

Por otra parte el usuario será capaz de activar y desactivar la base de enchufes desde cualquier terminal con conexión a internet, con tan solo conectarse a la URL (*Uniform Resource Locator*) predefinida en el servidor de aplicaciones.

La Fig. 8 muestra el consumo de dos equipos en la página principal de la aplicación, mediante la interfaz de usuario definida como Smart Meter (Medidor Inteligente).

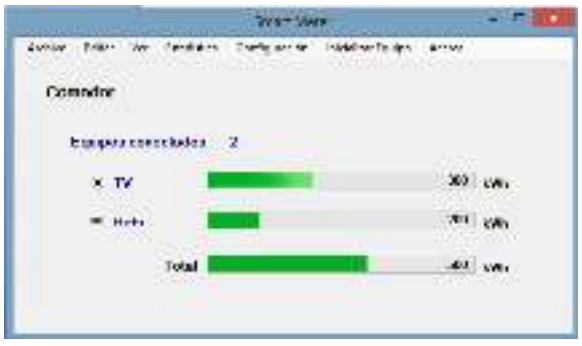


Figura 8: Interfaz con el usuario

4.2 GESTIÓN DE ALARMAS

El sistema es capaz de gestionar todo tipo de alarmas definidas por el usuario.

Gracias a que el servidor de gestión del sistema tiene instalado un servidor de correo, el usuario podrá configurar sus alarmas y avisos para que el sistema le envíe mediante correo electrónico cualquier tipo de incidencia.

Una vez avisado mediante un correo electrónico, el usuario podrá activar o desactivar la base de enchufes, ya sea por un consumo máximo de dicha base o bien por algún otro tipo de incidencia observada desde el sistema.

La implementación de sensores en el entorno de la base, tanto de temperatura como de iluminación, por ejemplo, permitirá al usuario detectar la temperatura ambiente del habitáculo donde el sistema está instalado y gestionar tanto un aire acondicionado como un sistema de radiadores eléctricos.

En el ámbito concreto de la seguridad, con el sistema de células fotovoltaicas seremos capaces de detectar la iluminación del habitáculo sirviéndonos esto tanto como medida de precaución de robos (encendiendo luces por la noche en caso de ausencia), como en el caso de que entren a robar por la noche y, al detectar más luz de lo habitual, el sistema nos dé un aviso de la incidencia.

4.3 BASES DE DATOS

Para generar los gráficos de consumo y las estadísticas tanto en línea como históricamente, la aplicación debe acceder a una base de datos, en nuestro caso hemos optado por MySQL, donde hemos almacenado toda la información proporcionada por cada uno de los enchufes inteligentes.

La comunicación entre la base de datos y nuestra aplicación web se lleva a cabo mediante el lenguaje de programación PHP.

Finalmente, cabe destacar que para este proyecto se ha hecho servir *software Open Source*, lo que permite al usuario final (si éste es un usuario avanzado) que pueda retocar a su gusto tanto el entorno como la gestión de la base de datos.

5 RESULTADOS

La Tabla 1 muestra el consumo eléctrico de cada uno de los componentes que forman el sistema.

Tabla 1: Consumo del enchufe inteligente

Componente	Descripción	Consumo
Diodo	DIODE, STANDARD, 1A, 1000V, SMD	12mW
Convertidor DC/DC	IC, OFF LINE SWITCHER, 360MA	200mW
IC MAX253	TRANSFORMER DRIVER	25mW
LDO 3.3V	V REG LDO +3.3V, 1117	200mW
ACS712	SENSOR, CURRENT, 20A	65mW
MC1100	AMP, ISOLATION, 60KHZ	70mW
18F14k50	Microcontroler 8 bits	100mW
Wifi RN171	Module WiFi	200mW
		870mW

En condiciones nominales de operación resulta un consumo de 870mW, al que aplicamos un factor de guarda y consideraremos que es de 1W. Al consumo de cada uno de los enchufes inteligentes hay que añadir el de la Raspberry Pi que es de 3.5W.

Con estos datos podemos estimar el consumo eléctrico anual de nuestro sistema en una vivienda tipo con 5 equipos. El consumo anual en kWh debido a los enchufes es de:

- 1W x 5 Equipos x 365 días x 24 horas = 43,8kWh

mientras que el de la Raspberry Pi es de:

- 3.5W × 1 Rasp.Pi ×365días × 24horas = 30.6kWh

En concepto de coste económico, asumiendo un precio de 0,138658€ el kWh, según tarifa básica de Iberdrola, el coste total del sistema es de 8,5€ anuales, con un desglose mensual de 0,70€.

Según un estudio de la OCU (*Organización de consumidores y usuarios*), los electrodomésticos o aparatos en suspensión o inactivos pero conectados a la línea eléctrica suponen 231kWh al año por cada hogar. Ello nos da una idea de cuánto es posible ahorrar con solo programar la desconexión de la red de los equipos en *stand-by*.

6 CONCLUSIONES

Se ha presentado un sistema modular para la monitorización y control remoto de consumo eléctrico.

Desde un principio, la idea de este proyecto se ha basado en realizar un sistema que nos permita:

1) un control de la energía consumida tanto en hogares como en entornos laborales y corporativos

2) aportando, de una forma sencilla y modular, una visión detallada del consumo energético y su evolución temporal; y

3) permitiendo una gestión inteligente del consumo y un ahorro de energía.

Este objetivo último, el ahorro de la energía consumida en los hogares o corporaciones supone, por un lado, un ahorro económico relevante en estos tiempos de austeridad actuales y, por otro, una gestión sostenible de los recursos energéticos beneficiosos para nuestro medio ambiente.

Además el sistema propuesto aporta una serie de funcionalidades como son:

- programación del consumo
- control distribuido de la temperatura y luminosidad en el hogar o lugar de trabajo
- sistema de alarmas ante fallos técnicos y/o intrusiones

Todas estas funcionalidades pueden ser adecuadamente gestionadas de forma remota.

Agradecimientos

Los autores de este artículo agradecemos a los miembros de los Servicios Informáticos del Dept. de la Teoría de la Señal y Comunicaciones de la UPC, en especial a Joaquín Fernández y a Josep Pujal, sus buenas ideas y enfoque en esta línea de trabajo.

Referencias

[1] Flick T., Morehouse J., (2010) *Securing the Smart Grid: Next Generation Power Grid Security*, Syngress Media.

[2] Grulich, J., (2005) "On Semiconductor, AND8226/D, Positive output buck AC/DC converter", *Semiconductor Components Industries, LLC.*

[3] http://www.raspberrypi.org

[4] IDAE (2011), *Plan de Ahorro y Eficiencia Energética 2011-2020. 2º Plan Nacional de Eficiencia Energética de España*, Ministerio de Industria.

[5] Kurose J. F., Keith W. R., *Computer Networking: A Top-Down Approach,* Pearson Publish.

[6] Power Integrations, (2004) *LNK304-306, Lowest component count, energy efficient off-line switcher*

[7] Sant-Pierre, R., (2002) "On Semiconductor, AND8078/D, Low-cost non isolated universal input off-line converter", *Semiconductor Components Industries*.

[8] Sirohiwala A., (2013), "New appoaches to high efficiency current sensing: integrated hall sensor IC save power and space", *Design News*, Allegro MicroSystems, LLC.

[9] Yang Xiao, (2012), *Communication and Networking in Smart Grids,* CRC Press.

Plataforma de gestión de aplicaciones distribuidas de seguridad funcional con tolerancia a fallos

Aitor Agirre†, Rafael Priego*, Dario Orive*, Elisabet Estévez**, Jon Pérez†, Marga Marcos*

† IK4-Ikerlan Research Alliance, España
{aagirre, jmperez}@ikerlan.es

* ETSI Bilbao, UPV/EHU, España
{rpriego001, dario.orive, marga.marcos}@ehu.es

** Escuela Politécnica Superior de Jaén, UJA, Spain
eestevez@ujaen.es

Resumen

Los sistemas distribuidos embebidos son cada vez más complejos en términos de flexibilidad, dinamismo, heterogeneidad y calidad de servicio (QoS). En el caso de los sistemas de seguridad críticos, se añade el obligado cumplimiento de ciertos requisitos de seguridad funcional especificados en los estándares de certificación. Esto ha motivado que la ingeniería de software basada en componentes (CBSE) desempeñe un papel cada vez más importante en el desarrollo de dichos sistemas.
En este artículo se presenta una plataforma de gestión de aplicaciones distribuidas reconfigurables basadas en componentes y con requisitos de seguridad funcional y tolerancia a fallos. En concreto se proponen una serie de mecanismos que permiten implementar una capa de software de comunicación certificable y reutilizable, que provee una abstracción de los aspectos relativos a la distribución segura de datos y por tanto reduce considerablemente los costes de desarrollo y certificación de este tipo de aplicaciones. La plataforma y los mecanismos propuestos se han aplicado en el campo de la automatización para asegurar la disponibilidad del sistema de control.

Palabras Clave: CBSE, SCA, seguridad funcional, sistemas distribuidos, aplicaciones críticas, automatización, aplicaciones reconfigurables, QoS, IEC 61131.

1 INTRODUCCIÓN

Los sistemas embebidos de seguridad funcional son sistemas confiables que podrían causar una pérdida económica significativa, lesiones o pérdida de vidas humanas en caso de fallo. La norma internacional de seguridad funcional IEC-61508 [1] es una norma de seguridad genérica para la industria que se ha particularizado para diversos dominios como por ejemplo los procesos industriales, el ferrocarril o la automoción. El desarrollo y certificación de sistemas integrados de seguridad funcional crítica es caro [2, 3], y el comportamiento o la funcionalidad dinámica deben ser cuidadosamente analizados y justificados.

Un aspecto esencial de los sistemas embebidos distribuidos es el software relacionado con las comunicaciones [4]. En ocasiones se implementa desde cero utilizando mecanismos de bajo nivel como sockets o incluso protocolos de nivel OSI 2 como EtherCAT o CAN. Otras veces se utilizan tecnologías de middleware de distribución estandarizadas, ya sean genéricas (Java RMI, Servicios Web SOAP, CORBA…) o adaptadas al dominio de los sistemas embebidos (CORBA/e [5], REST, ICE-e [6] o DDS [7]). Asimismo, se ha invertido un considerable esfuerzo en proporcionar calidad de servicio a este software de distribución, a menudo enfocada a restricciones temporales y a gestión de recursos [8, 9] pero no específicamente a aspectos relacionados con la confiabilidad o seguridad funcional.

La disponibilidad de un middleware de distribución certificable, genérico, reutilizable, y con funciones y propiedades de seguridad funcional adecuadas, puede facilitar el desarrollo y reducir el costo de los sistemas distribuidos de seguridad funcional crítica. Parece por tanto razonable la aplicación de los principios de CBSE [10] a este tipo de sistemas distribuidos con requisitos de seguridad. Además del objetivo principal de disponer de este middleware de seguridad acorde a la norma IEC 61784-3-3 [11], se han considerado otros requisitos adicionales a la hora de diseñarlo: Abstracción de los detalles de bajo nivel relativos a la comunicación, flexibilidad, modularidad, independencia de la plataforma y gestión de aplicaciones reconfigurables.

Para cumplir estos requisitos no funcionales y abstraer los mecanismos de seguridad en las comunicaciones descritos en la norma, se han diseñado e implementado una serie de artefactos que se basan en el modelo de componentes SCA [12]. Este modelo de componente es el utilizado por la plataforma de gestión de aplicaciones distribuidas que se ha desarrollado en trabajos previos [13], y que se ha ampliado en este trabajo para proporcionar la segunda de las aportaciones: Un servicio de monitorización de salud (*Health Monitor*) que comprueba el estado de los componentes de aplicación distribuidos y proporciona mecanismos de tolerancia a fallo de los mismos, con recuperación íntegra del estado de ejecución de los mismos.

El resto del artículo se divide así: En la Sección 2 se presenta la plataforma de gestión de aplicaciones distribuidas, junto con el nuevo servicio *Health Monitor*. En la Sección 3 se detalla la extensión SCA propuesta para aplicaciones con restricciones de seguridad funcional, haciendo un breve repaso de los mecanismos de seguridad descritos en la norma IEC 61784-3-3. En la Sección 4 se describe un caso de estudio en el ámbito de la automatización industrial, que valida las propuestas anteriores en el contexto de la tolerancia a fallos de sistemas basados en PLC-s.

2 PLATAFORMA DE GESTIÓN DE APLICACIONES DISTRIBUIDAS

En trabajos previos [14, 15] se ha desarrollado un núcleo de plataforma de gestión de aplicaciones distribuidas basadas en componentes, denominado DCMP (*Distributed Component Management Platform*), que en su última versión [13] adopta el modelo de componente de SCA.

En la Sección 2.1 se describe el modelo de componente SCA y los mecanismos de extensión que ofrece; en la Sección 2.2 los elementos principales de la plataforma DCMP, y en la Sección 2.3 sus servicios principales y el nuevo servicio *Health Monitor* propuesto para soportar tolerancia a fallos con recuperación de estado.

2.1 MODELO DE COMPONENTE SCA

Uno de los puntos clave de la ingeniería de software basada en componentes es la adopción de un modelo de componente adecuado a unas necesidades concretas. En trabajos anteriores [13] se seleccionó SCA debido a varias razones: Aparte de proporcionar las típicas ventajas de modularidad, flexibilidad, y reusabilidad que están asociadas a la tecnología de componentes en general [10], SCA desacopla perfectamente los aspectos funcionales de los no funcionales, el despliegue de las aplicaciones puede estandarizarse de forma relativamente sencilla y soporta reconfiguración declarativa de las aplicaciones. Adicionalmente, proporciona mecanismos de extensión que hacen que el motor de ejecución sea extensible y altamente personalizable.

La arquitectura SCA se define en un conjunto de estándares que definen un modelo de programación para crear aplicaciones a partir de estructuras (componentes) que encapsulan la lógica de negocio y la ofrecen en forma de servicios.

En la Figura 1 puede verse la representación gráfica de un componente SCA. Un componente SCA está compuesto por una implementación y un conjunto de propiedades, servicios y referencias. La implementación del componente puede realizarse en distintos lenguajes. Dependiendo de la distribución de SCA, pueden estar disponibles más o menos lenguajes. El estándar SCA incluye C, C + +, BPEL, Java y Spring, pero las implementaciones concretas de SCA, como por ejemplo FraSCAti [16] o Apache TUSCANY [17], pueden añadir tipos de implementación (lenguajes) adicionales. En este sentido, las anteriormente citadas soportan lenguajes de script basados en Java, Scala u OSGi.

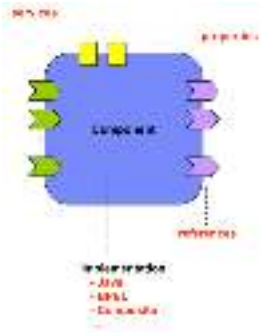


Figura 1: Representación de un componente SCA

Los servicios y las referencias se pueden especificar a través de un conjunto ampliable de tipos de interfaz (Figura 2): interfaces Java, WSDL (Web Services Description Language), UPnP (Universal Plug and Play), encabezados C, etc. Además del tipo de interfaz, se pueden asociar uno o varios conectores ("*bindings*") a un servicio o referencia. El concepto de conector se refiere al middleware de distribución que se utiliza para interconectar un servicio con una referencia. Si no se especifica nada, se utiliza un middleware predeterminado dependiente de la implementación SCA elegida [18].

Actualmente, la norma incluye los conectores para Servicios Web, JMS (Java Messaging Service) y JCA (Java EE Connection Architecture), pero pueden añadirse nuevos conectores de forma relativamente sencilla. En este sentido, las distribuciones

TUSCANY y FraSCAti proporcionan conectores para CORBA, RMI REST o ATOM.

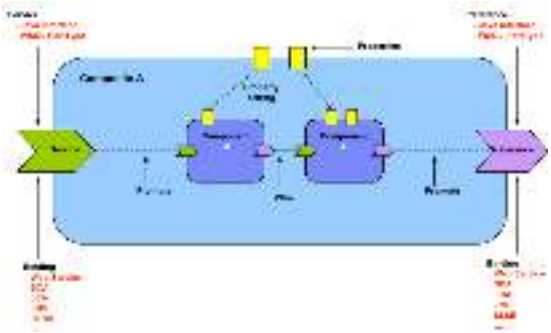


Figura 2: Diagrama de *composite* SCA

Junto con el concepto de conector, SCA proporciona soporte para la especificación de restricciones, capacidades y expectativas de calidad de servicio, tanto en la fase de diseño como en la de implementación. Siguiendo el paradigma AOP (*Aspect Oriented Programming*), SCA permite asociar políticas (aspectos no funcionales) a los componentes SCA [19]. De esta manera, los componentes pueden declarar los servicios no funcionales de los que dependen, y dejar en manos del motor SCA la aplicación de dichas políticas [20]. Hasta ahora, es el estándar de políticas SCA quien incluye la seguridad (*security*) y las transacciones, pero el conjunto de políticas se puede extender para satisfacer las necesidades específicas de la aplicación, como por ejemplo la persistencia, la traza de información, o en el caso que nos ocupa, la seguridad funcional.

Todos los conceptos anteriores asociados a un componente SCA, junto con las interconexiones entre los componentes de una aplicación, se especifican declarativamente en un archivo XML con extensión *.composite*. La Figura 2 representa un composite que se compone de dos componentes interconectados entre sí.

En resumen, SCA ofrece varios mecanismos de extensión que hacen que un motor de tiempo de ejecución SCA sea altamente configurable y extensible: (1) los tipos de implementación o lenguajes, (2) los tipos de interfaz, (3) los tipos de conector y (4) las políticas.

2.2 ELEMENTOS PRINCIPALES

La plataforma DCMP se compone de dos componentes principales, el *MW_Manager* y el *MW_Daemon*. Adicionalmente, la plataforma provee una clase base de la que los componentes de aplicación heredan. Esta clase base implementa un puerto de control que permite realizar operaciones de gestión sobre los componentes individuales (lanzar, parar, etc.). En la Figura 3 se representan los componentes principales de la plataforma junto con dos aplicaciones distribuidas formadas por un número variable de *composites* SCA (que a su vez pueden estar formados por uno o varios componentes SCA). En amarillo se representan los componentes específicos de la plataforma DCMP. No está representado el puerto de control de los componentes de aplicación, que es accedido por el *MW_Daemon* y el *MW_Manager* para lanzar y parar los composites respectivamente.

En un sistema distribuido formado por *n* nodos físicos, existe *un* único *MW_Manager, n* instancias de *MW_Daemon* (una por nodo) y *m* composites de aplicación, cada uno alojado en un proceso.

El *MW_Manager* gestiona el ciclo de vida de las aplicaciones, desde el registro e instalación hasta el control de ejecución (iniciar/parar) y la desinstalación. El *MW_Daemon*, presente en todos los nodos, tiene dos funciones: Por una parte ejecuta las órdenes de lanzamiento de los composites enviadas desde el *MW_Manager*, y por otra empaqueta la información de estado relativa a los composites alojados en su nodo y la reenvía periódicamente al *MW_Manager*, que de esta forma conoce el estado real de todo el sistema y puede tomar decisiones ante determinados eventos como por ejemplo la caída de un nodo o de un componente de aplicación.

La comunicación de infraestructura entre los componentes de la plataforma y el puerto de control de los componentes de aplicación se realiza sobre DDS, ya que este middleware permite un control fino de algunos parámetros de QoS muy interesantes desde el punto de vista de la tolerancia a fallos. Por ejemplo, sería posible tener en funcionamiento más de una instancia del *MW_Manager* simultáneamente, de forma que en caso de caída de uno de ellos, el siguiente en orden de prioridad tomara las riendas.

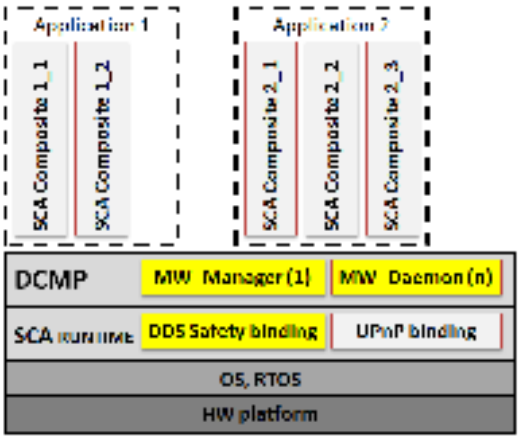


Figura 3: Plataforma DCMP

2.3 SERVICIOS DE LA PLATAFORMA

La plataforma DCMP se ha concebido de forma que la inclusión de nuevos servicios sea sencilla y modular. En la primera versión de la plataforma [21] se contemplaban los servicios de la Tabla 1, aunque ya se apuntaban (servicios en color naranja en la Figura 4) otras posibles funcionalidades como el control de admisión o la reconfiguración dinámica a partir de un *Health Monitor* y un algoritmo de composición de aplicaciones basadas en servicios.

Tabla 1: Servicios básicos de la plataforma.

Servicio	Notas
Registro	Registro de aplicaciones y componentes
Control de ejecución	Inicio / parada de aplicaciones
Configuración de la QoS	Soporte a la configuración de parámetros de QoS de los componentes de aplicación
Tolerancia a fallos	Basado en la redundancia de componentes, tanto de aplicación como de plataforma

Los servicios básicos comprenden el registro de los componentes y aplicaciones, el control de ejecución de las mismas, la reconfiguración dinámica de parámetros de calidad de servicio de los componentes de aplicación, así como una tolerancia a fallos básica tanto de los componentes de la plataforma como de los de aplicación.

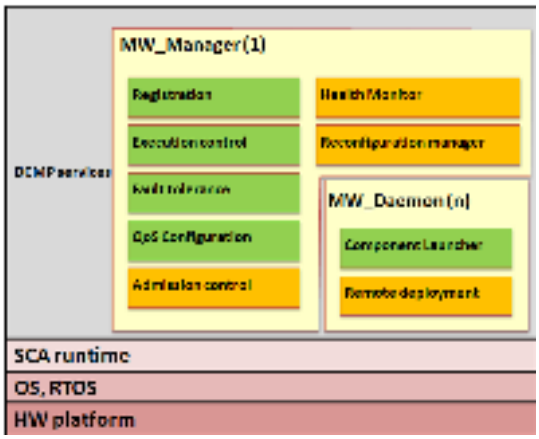


Figura 4: Servicios de la plataforma DCMP

Este servicio de tolerancia a fallos actúa en dos sentidos: Por una parte proporciona una alta disponibilidad de los componentes de plataforma (*MW_Manager* y *MW_Daemon*) gracias al soporte que DDS brinda a través del parámetro *strength*, que permite la ejecución simultánea de múltiples componentes idénticos configurados con valores diferentes para dicho parámetro. El componente con mayor *strength* es el que comanda el sistema, y en caso de caída del mismo, el siguiente componente en orden de *strength* asume el mando, estando por tanto garantizada la disponibilidad de dicho componente mediante este mecanismo de redundancia. Esta característica es especialmente útil en el caso del *MW_Manager*, que actúa como gestor centralizado de todo el sistema, y por tanto como punto único de fallo (*single point of failure*) susceptible de protección.

Por otra parte, los componentes de aplicación son monitorizados por el *MW_Daemon*, que envía periódicamente al *MW_Manager* un informe sobre el estado de ejecución de los componentes alojados en su nodo. En este informe se indica si el componente está vivo o no, así como el porcentaje de CPU y memoria que está consumiendo. Teniendo en cuenta esta información de la situación real, y contrastándola con el estado deseado del sistema que el *MW_Manager* también conoce, el *MW_Manager* puede decidir el lanzamiento o la parada de determinados componentes, según una política predefinida. Por ejemplo, si un componente de aplicación cae, el *MW_Manager* comprueba si dispone de un componente gemelo (*backup*) desplegado en algún otro nodo, y si es así ordena su lanzamiento. El algoritmo de búsqueda de alternativas puede ser implementado en función de diversas políticas, como por ejemplo la búsqueda del nodo más descargado (balance de carga), el primero disponible, etc.

En este trabajo se presenta un nuevo servicio, denominado *Health Monitor*, que no sólo monitoriza el estado de ejecución de los componentes de aplicación, sino que recibe información de estado de aquellos componentes configurados como con estado (*statefull*), y por tanto puede efectuar un restablecimiento completo del estado del componente caído. Para ello, se ha ampliado el puerto de control de los componentes de aplicación, que ahora proporcionan información de su estado interno directamente al *MW_Manager*, que la almacena en aras de un eventual restablecimiento de un componente de aplicación o incluso de un nodo completo.

Con el objetivo de ahorrar ancho de banda en la transmisión de la información de estado de los componentes de aplicación, los componentes sólo comunican sus cambios de estado. Esto hace necesario que el mecanismo de transmisión de estos mensajes de estado sea confiable, evitando la pérdida de los mismos. Esta confiabilidad está basada en el conector DDS de seguridad funcional descrito en la Sección 3, que junto con el servicio *Health Monitor* ya descrito, conforma la principal contribución de este trabajo.

3 EXTENSIÓN PARA APLICACIONES DE SEGURIDAD FUNCIONAL

La norma IEC 61784-3-3 [11] describe una serie de técnicas y medidas para conseguir una comunicación segura sobre un canal de transmisión intrínsecamente inseguro ("*black channel*"). En este trabajo preliminar se han considerado, como prueba de concepto, tres técnicas de diagnóstico apuntadas en la norma. Estas técnicas se han incluido en la Tabla 2, junto a los errores que permiten detectar.

Tabla 2: Técnicas de diagnóstico.

	Técnica de diagnóstico		
Error	Nº sec	Watch-dog	Firma
Corrupción			X
Repetición	X		
Error sec	X		
Pérdida	X		
Retraso		X	

Las medidas de seguridad proporcionadas por la capa de comunicación segura deben garantizar que los datos enviados por el emisor llegan al receptor de manera fiable, sin repeticiones ni retrasos, de forma ordenada e íntegra, evitando la corrupción o modificación de los mismos en el trayecto. Para lograr esto, se contemplan las siguientes técnicas:

(1) *Firma*: Previene la corrupción de los mensajes en el medio de transmisión, y por tanto asegura su integridad. Esta eventual corrupción de datos puede ser motivada por diversas causas, desde ruido en el cableado a hardware defectuoso. El mecanismo de firma se basa en el cómputo de un CRC (*cyclic redundancy check*) para cada mensaje transmitido, que se adjunta en el propio mensaje y que puede ser posteriormente validado en la recepción.

(2) *Watch-dog*: Las aplicaciones de control normalmente tienen que garantizar un intervalo de muestreo determinado. En este sentido, puede definirse un plazo de recepción para distintos tipos de mensajes. La eventual pérdida de plazo de dichos mensajes puede ser detectada mediante el mecanismo de watch-dog, con el consiguiente disparo de las oportunas medidas de seguridad, que normalmente llevan al sistema a un estado seguro prefijado.

(3) *Número de secuencia*: Para detectar la pérdida y la repetición (intencionada o no) de mensajes, el emisor adjunta un número de secuencia en el mensaje a transmitir. De esta forma el emisor puede chequear si el mensaje ha llegado en el orden correcto, sin repetición o pérdida de paquetes entre medias, y concluir si es válido o no.

En la parte superior de la Figura 5 se representa una aproximación *ad-hoc* simplificada para una aplicación distribuida de seguridad funcional, que debe garantizar la transmisión segura de datos sobre un canal no seguro. Los componentes en gris están relacionados con la seguridad.

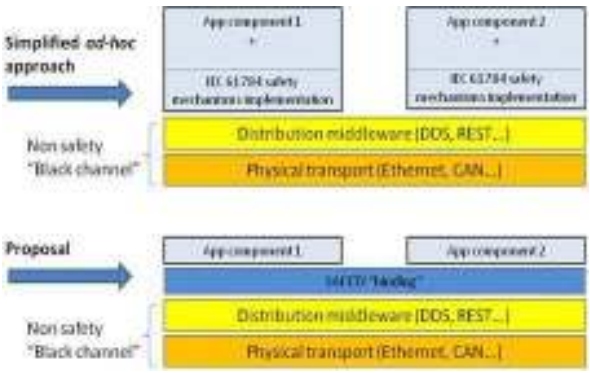


Figura 5: Ejemplo de leyenda de una figura

En dicha aproximación *ad-hoc* simplificada, los componentes de aplicación son los responsables de implementar los mecanismos de seguridad descritos anteriormente. Generalmente, las soluciones adoptadas con este esquema son muy particulares y poco reutilizables, con una API (*Application Programming Interface*) semánticamente enfocada a las comunicaciones y que por tanto proporciona un nivel de abstracción muy bajo de los aspectos relativos a la distribución.

Por el contrario, se propone una aproximación basada en una capa de software genérica (ver "*safety binding*" en la Figura 5) que proporciona todos los mecanismos de seguridad descritos anteriormente, de forma transparente para el desarrollador. Esta capa de comunicaciones seguras está basada en dos extensiones SCA:

(1) *Conector ("binding") DDS*: DDS es un middleware de distribución altamente configurable en términos de calidad de servicio. Aspectos tales como la confiabilidad de la transmisión, el ancho de banda, o la latencia y plazo de los mensajes son fácilmente configurables de forma declarativa mediante un fichero XML. Además, puede ser implementado sobre cualquier transporte de nivel OSI 2, como por ejemplo CAN [22], lo cual es muy interesante para sistemas distribuidos embebidos.

(2) *Política de seguridad funcional*: El aspecto de seguridad funcional se implementa a través del concepto de política de SCA, que puede ser declarativamente asociado al conector DDS

descrito anteriormente. Esta política incluye los mecanismos de seguridad anteriormente descritos (firma, watch-dog, número de secuencia). Para cada función de nivel aplicativo (lógica de negocio) que es invocada a través de una referencia de componente, la política de seguridad funcional computa la firma CRC del tópico DDS original, que incluye el nombre de la función, sus argumentos y su número de secuencia. A continuación, esta firma se adjunta al tópico original como un campo adicional, y es éste tópico "firmado" el que finalmente se transmite al medio físico. En la parte receptora, el proceso se invierte: Se recibe el tópico firmado, que contiene el tópico original y la firma. Se computa el CRC sobre el contenido útil y se compara con la firma recibida, y si el resultado es positivo se garantiza la integridad del mensaje. Por otra parte, el número de secuencia permite detectar la repetición o pérdida de mensajes, y el watch-dog eventuales retrasos en mensajes que pudieran llevar a pérdidas de plazo críticas.

Esta aproximación abstrae completamente al desarrollador de toda la fontanería relativa a la distribución. No hay necesidad de manejar API-s orientadas a comunicaciones, en tanto en cuanto esta capa de comunicación genérica se encarga de ligar el acceso a los interfaces de negocio con el código de comunicaciones de bajo nivel, que permanece invisible al usuario. De esta forma, el desarrollador puede centrarse en el código funcional, como si estuviera trabajando en una aplicación local no distribuida. Lo único que es necesario especificar el fichero *.composite* de SCA es el conector ("*binding*") y la política requerida en los extremos "referencia" y "servicio" de dicho conector.

Es importante reseñar que el conector DDS podría ser utilizado sin política de seguridad en aplicaciones sin necesidad de seguridad funcional, lo que redundaría en el ahorro del tiempo de computación asociado a los mecanismos de seguridad (p.ej. CRC).

4 APLICACIÓN A SISTEMAS DE AUTOMATIZACIÓN CONFIABLES

El siguiente caso de estudio presenta la utilización de DCMP para implementar un nivel de supervisión centrado en mantener la disponibilidad del sistema de control frente a fallos en el equipo de control. El proceso de reconfiguración no sólo recupera la funcionalidad asociada al equipo de control en fallo, sino que también reinicia la funcionalidad desde el último estado conocido. Para ello, utiliza el canal de comunicación segura comentado en el apartado anterior.

La aplicación de supervisión mantiene el estado de la funcionalidad asociada al PLC. El estado representa el conjunto de valores que caracterizan la historia del sistema de control en un instante determinado y depende de la aplicación; así que debe ser definido a la vez que se diseña la aplicación.

Para proporcionar la disponibilidad del control, el sistema se basa en un conjunto de Soft PLCs distribuidos, siendo cada uno de ellos responsable de parte del control de la planta. Un Soft PLC cuenta con un sistema operativo y una máquina virtual de PLC que se ejecutan en paralelo compartiendo el recurso CPU. Cada Soft PLC ejecuta en su máquina virtual su código de control de su parte de la planta. Este código se comunica con su correspondiente componente de aplicación para enviar o restaurar su estado. Además el PLC puede contener el código de control y componente de aplicación de otro PLC, como *backup* de este. Es importante destacar que el código de control y componente de aplicación de *backup* sólo se ejecutará como consecuencia de una reconfiguración.

Como se ha mencionado anteriormente, una de las funciones del *MW_Daemon* es la de enviar los *heart-bits* de su nodo al *MW_Manager.* La falta de de recepción de los *heart-bits* alguno de los nodo (PLC) provoca el lanzamiento de una reconfiguración. La re configuración consiste en la recuperación de la funcionalidad desde el último estado conocido en otro PLC. La selección de los PLC de *backup* se realiza en la fase de diseño.

La Figura 6 ilustra un sistema compuesto por dos Soft PLc y un Supervisor PC que ejecuta el *MW_Manager*.

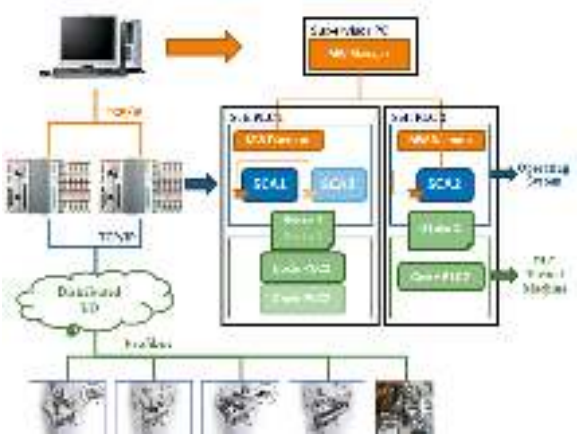


Figura 6: Escenario general para sistemas de automatización

Cada uno de los PLC tiene su código de control y componente de aplicación, pero el PLC1 puede funcionar además como *backup* del PLC2 por lo cual

el PLC1 tiene el código de control y componente de aplicación asociados a PLC2.

En funcionamiento normal los dos PLCs envían su estado al *Health Monitor*. En caso de fallo del PLC2 el *MW_Manager* detecta la pérdida de los *heart-bits* del nodo y lanza la reconfiguración.

Para la reconfiguración se envía el último estado conocido al correspondiente *backup* (en este caso el PLC1) quien recupera la funcionalidad. Al terminar la reconfiguración se tienes las dos funcionalidades de los PLC ejecutándose en el PLC1, enviando sus correspondientes estados al *Health Monitor.* Es importante destacar que durante la reconfiguración la funcionalidad del PLC1 no se ve afectada por la reconfiguración.

Durante la fase de codificación, el código de control de cada PLC debe ser extendido con el código correspondiente a aquellos PLC de los que se considera *backup*. También se debe de definir los estado de los PLCs (determinar las variables que definen el estado). La información de los estados se utiliza para generar el código de los componentes de aplicación.

Para probar el funcionamiento de la aplicación se han hecho pruebas en una estación de una célula de fabricación. El control de la estación se realiza con dos PLC que funcionan como *backup* entre ellos. Cada vez que uno de ellos falla el otro recupera el funcionamiento.

5 CONCLUSIONES Y LINEAS FUTURAS

Los costes de certificación de aplicaciones distribuidas con requisitos de seguridad funcional son muy elevados. Cualquier modificación en un código ya certificado supone en general repetir total o parcialmente el proceso de certificación. El presente trabajo se centra en dos contribuciones:

1) Por una parte considera la distribución segura de datos en dichos sistemas, proponiendo una serie de mecanismos genéricos y reutilizables que abstraen los aspectos relativos a dicha distribución. Estos mecanismos se basan en dos puntos de extensión que proporciona el estándar SCA. Concretamente, se ha desarrollado un nuevo conector sobre DDS y una política de seguridad funcional asociada al mismo, que implementa tres técnicas de diagnóstico definidas en la norma IEC 61784-3-3.
2) Sobre el núcleo de una plataforma de gestión de aplicaciones SCA distribuidas (DCMP) desarrollado en trabajos previos, propone una extensión de los servicios proporcionados por la misma (*Health Monitor*) de cara a soportar mecanismos de tolerancia a fallos con recuperación de estado.

Ambas contribuciones se han validado en un caso de estudio en el campo de la automatización industrial, que demuestra las capacidades de tolerancia a fallos con recuperación de estado de la plataforma DCMP.

Este enfoque persigue dos objetivos principales: (1) Seguridad de sistemas críticos de desarrollo de reducción de costes a través de la reutilización de una capa de comunicación de seguridad certificable, y (2) hacer cumplir la eficiencia desarrollador a través de la abstracción del código relacionado distribución.

Como trabajo futuro se contempla ampliar el conector DDS para soportar el paradigma de interacción cliente-servidor (en la actualidad soporta únicamente el paradigma publicador-suscriptor), así como la política de seguridad funcional, ampliando el número de técnicas de seguridad soportadas actualmente.

Además se esta trabajando en automatizar la generación de los componentes de aplicación y el código del PLC con sus *backup*, a partir del modelo hardware y software del PLC, la definición de los estados y una vista de QoS para la definición de los *backups*.

Agradecimientos

Parte de este trabajo fue apoyado por el MCYT&FEDER en virtud de la concesión DPI-2012-37806, el Gobierno Vasco (GV/EJ) con la subvención BFI-2011-251 y el proyecto IT719-13, además de la UPV/EHU bajo la subvención UFI11/28.

Referencias

[1] "IEC 61508-1: Functional safety of electrical/electronic programmable electronic safety-related systems ", ed: IEC, 2010.
[2] J. C. Knight, "Safety critical systems: challenges and directions," in *Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on*, 2002, pp. 547-550.
[3] F. Carvalho, S. R. L. Meira, B. Freitas, and J. Eulino, "Embedded Software Component Quality and Certification," in *Software Engineering and Advanced Applications, 2009. SEAA '09. 35th Euromicro Conference on*, 2009, pp. 420-427.

[4] Atomineerutils. (2013). *Atomineer Pro Documentation*. Available: http://www.atomineerutils.com/download.php
[5] OMG, "Common Object Request Broker Architecture (CORBA) for embedded Specification v1.0," ed, 2008.
[6] ZeroC. *Ice-E ("Embedded Internet Communication Engine")*. Available: http://www.zeroc.com/icee/index.html
[7] OMG, "Data Distribution Service for Real-time Systems v1.2," ed, 2007.
[8] Microsoft. *Visual Studio Warning Levels*. Available: http://msdn.microsoft.com/en-us/library/thxezb7y.aspx
[9] C. D. Gill, J. M. Gossett, D. Corman, J. P. Loyall, R. E. Schantz, M. Atighetchi, and D. C. Schmidt, "Integrated adaptive QoS management in middleware: A case study," *Real-Time Systems,* vol. 29, pp. 101-130, 2005.
[10] C. Szyperski, *Component Software: Beyond Object-Oriented Programming*: Addison-Wesley, 1998.
[11] "IEC 61784-3-3: Industrial communication networks - Profiles - Part 3-3: Functional safety fieldbuses - Additional specifications for CFP 3," ed: IEC, 2007.
[12] OASIS, "Service Component Architecture," ed, 2007.
[13] A. Agirre, M. Marcos, and E. Estevez, "Distributed Applications Management Platform Based on Service Component Architecture," in *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, 2012, pp. 1-4.
[14] A. Agirre, E. Estévez, and M. Marcos, "QoS enabled application management platform over DDS," presented at the Proceedings of the Middleware Workshop on Posters and Demos Track, Lisbon, Portugal, 2011.
[15] A. Agirre, M. Marcos, and E. Estevez, "Distributed component management platform for QoS enabled applications," in *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on*, 2011, pp. 1-4.
[16] L. Seinturier, P. Merle, R. Rouvoy, D. Romero, V. Schiavoni, and J.-B. Stefani, "A component-based middleware platform for reconfigurable service-oriented architectures," *Software: Practice and Experience,* pp. n/a-n/a, 2011.
[17] S. Laws, M. Combellack, R. Feng, H. Mahbod, and S. Nash, *Tuscany SCA in Action*, 2011.
[18] OASIS, "Service Component Architecture Assembly Model Specification Version 1.1," ed, 2011.
[19] OASIS, "SCA Policy Framework Version 1.1," ed, 2011.
[20] L. Seinturier, P. Merle, D. Fournier, N. Dolet, V. Schiavoni, and J. B. Stefani, "Reconfigurable SCA Applications with the FraSCAti Platform," in *Services Computing, 2009. SCC '09. IEEE International Conference on*, 2009, pp. 268-275.
[21] A. Agirre, E. Estévez, and M. Marcos, "Fault tolerant component management platform over Data Distribution Service," in *1st IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control (CESCIT)*, Wurzburg, 2012, pp. 218-223.
[22] R. Rekik and S. Hasnaoui, "Application of a CAN BUS transport for DDS middleware," in *2nd International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2009*, London, 2009, pp. 766-771.

Diseño Basado en Componentes y generación de aplicaciones robóticas manipuladoras bajo el middleware de OROCOS

A. Sánchez García, E. Estévez, J Gómez Ortega, J. Gámez García
Grupo de Robótica, Automática y Visión por Computador.
Universidad de Jaén.
Email: {asgarcia, eestevez, juango, jggarcia }@ujaen.es

Resumen

Uno de los objetivos fundamentales de la robótica manipuladora consiste en la integración de los robots en las tareas cotidianas de las personas. Para ello, existe un amplio rango de dispositivos físicos –sensores, actuadores, elementos auxiliares como herramientas, etc- que pueden incorporarse en un robot, así como una gran variedad de algoritmos y componentes lógicos que permiten extender las funcionalidades propias del robot. De la misma forma, numerosas herramientas software para aplicaciones robóticas, pero la gran mayoría de éstas son de código cerrado o dependientes del hardware utilizado. Este trabajo explora las ventajas que proporcionan el uso de diseño basado en componentes, así como la ingeniería conducida por modelos, para dar soporte al ciclo de desarrollo de robots manipuladores.

Palabras Clave: Arquitectura Software Robóticas, Diseño Basado en Componentes, Ingeniería Conducida por Modelos, Middleware, OROCOS.

1 INTRODUCCIÓN

Uno de los objetivos fundamentales de la robótica manipuladora consiste en la integración de los robots en las tareas cotidianas de las personas. Para ello, existe un amplio rango de dispositivos físicos-sensores, actuadores, elementos auxiliares como herramientas, etc- que pueden incorporarse en un robot, así como una gran variedad de algoritmos y componentes lógicos que permiten extender las funcionalidades propias del mismo. Hoy en día existen una gran variedad de herramientas software que integran dichos dispositivos o funcionalidades, pero la gran mayoría de éstas son de código cerrado o dependientes del hardware utilizado.

Desde hace más de una década, existe un creciente interés por parte de la comunidad investigadora en mejorar el proceso de desarrollo de aplicaciones robóticas avanzadas. Así, en los últimos años, los avances en hardware, sensores, actuadores, algoritmos de control, elementos informáticos y, en menor medida, en tecnologías software, han posibilitado el avance de la robótica. Un hardware más complejo y avanzado requiere una infraestructura software más sofisticada, debido principalmente a las siguientes causas [22]:

- ***Heterogeneidad del hardware***. Un robot integra multitud de sensores, actuadores, sistemas de comunicación y unidades de procesamiento que abarca una variedad mucho más amplia que la mayoría de los sistemas mecatrónicos.
- ***Heterogeneidad del software***. Un robot completamente desarrollado debe integrar una gran variedad de funcionalidades, que incluyen distintas áreas del conocimiento como la inteligencia artificial, planificación y programación, aprendizaje y adaptación, computación evolutiva y genética, visión artificial, procesamiento y producción del lenguaje, comprensión y diálogo del lenguaje natural, interpretación sensorial y fusión sensorial, control inteligente ... Muchas de estas áreas han desarrollado su propio conjunto de métodos para solventar sus respectivos problemas, usando a menudo lenguajes de programación y enfoques de control muy diferentes. Por lo tanto, es necesario integrar estas características en la arquitectura de control de los robots de forma coherente y bien coordinada.
- ***Rigidez de los fabricantes***. El software que desarrollan las empresas es propietario con lo que no existe ninguna divulgación.
- ***Informática distribuida en tiempo real***. Los robots, generalmente, usan varios dispositivos conectados entre sí, formando pequeñas redes de comunicación que tienen que funcionar de forma conjunta, coordinada y, en la mayoría de los casos con restricciones temporales.

Para reducir esta complejidad, se ha ido evolucionando en el uso de tecnologías de la *ingenería del software* [12]. En este contexto, este trabajo explora las ventajas que proporcionan el uso de diseño basado en componentes (*Component Based Software Engineering - CBSE*) [11], así como la

ingeniería conducida por modelos (*Model Driven Engineering - MDE*) [2], para dar soporte al ciclo de desarrollo de aplicaciones robóticas manipuladoras.

El concepto de componente ha sido ampliamente utilizado en la comunidad robótica para el desarrollo de módulos independientes software [2], [7] y [13]. Así, una aplicación robótica se define mediante el uso de dichos componentes a los que se les añade la lógica de la misma. CBSE permite incrementar el grado de reusabilidad del código en una determinada plataforma. Sin embargo, esta disciplina software es poco flexible, ya que los componentes desarrollados en una plataforma, generalmente, no son exportables a otra, siendo necesaria, por tanto, nuevamente su re-implementación.

Para resolver este problema, se utiliza MDE, donde el concepto clave es el *modelo* - representación de un sistema-. En todo modelo se representan los aspectos relevantes del sistema, filtrando los no esenciales, logrando así un problema más comprensible [15]. En los últimos años, se ha comenzado a introducir esta disciplina en el campo de la robótica [8]. En [21] se presenta una herramienta llamada *EasyLab* basada en dos lenguajes gráficos propietarios de modelado: *Synchronous Data Model* muy similar a *Sequential Function Chart* de IEC 61131-3 para definir la funcionalidad del sistema. El segundo lenguaje de modelo está centrado en la descripción del hardware, como una colección de sensores y actuadores, pero sin ninguna representación ni nomenclatura estándar. Dicha herramienta, inicialmente, estaba enfocada a sistemas mecatrónicos; aunque posteriormente los autores la adaptaron para plataformas *Robotino Mobile Robot* © [16]. En [5] se presenta un lenguaje de modelado que contempla la descripción de aplicaciones robóticas desde tres puntos de vista: (1) *estructural* para definir la estructura estática de la aplicación basada en componentes; (2) *coordinación* donde se define el comportamiento de cada componente y (3) *algoritmo* para la definición del código ejecutado en cada componente en función de su estado. Uno de los resultados del proyecto *SmartSoft* es la herramienta MDSD Toolchain [4] que está inspirada en el estándar Model Driven Architecture [14], propuesto por Object Management Group (OMG). El proyecto BRICS (Best Practices in Robotics) [20] también proporciona una herramienta llamada BRIDE para facilitar el diseño de las aplicaciones robóticas. Ambas herramientas propietarias se han desarrollado a través del plug-in *EMF-Eclipse Modeling Framework* de eclipse [6].

Pese a que todos los trabajos comentados previamente hacen uso del diseño basado en modelos, el presente trabajo va un paso más allá ya que: (1) propone un interfaz común para todos los elementos que participan en las tareas de robots manipuladores (e.g. sensores, robot, algoritmos de control…); (2) especifica cómo es la interacción de unos con otros, definiendo así la lógica de la propia tarea; y finalmente, (3) se indican los pasos para la generación de código que se ejecute bajo un middleware de comunicaciones basado en componentes.

Por tanto, las principales aportaciones de dicho trabajo son:

(1) Definición de *interfaces "estándares"* independientes a los fabricantes de los dispositivos que se manejen (e.g. sensores, robots,…). Estas interfaces, permiten, por un lado, abstraer al diseñador del conocimiento de las funciones del *API-Application Program Interface*, propias del fabricante. Además, por otro lado, son un punto de partida para los programadores, encargados de añadir un nuevo componente a la plataforma donde se ejecutan las tareas robóticas manipuladoras.
(2) Modelado con notación estándar UML-Unified Modeling Language [10] de la tarea robótica manipuladora. En concreto se hacen uso de un sub-conjunto de diagramas UML como son: el diagrama de clases, diagrama de secuencia diagrama de componentes así como el diagrama de despliegue.
(3) Generación de código que se ejecute en un middleware de comunicaciones basado en componentes, por ejemplo, OROCOS-Open Robot Software [17].

La estructura del presente trabajo es la siguiente: en la sección 2 se detallan las propiedades y métodos comunes que caracterizan todo tipo componente presente en la definición de una tarea robótica manipuladora. La sección 3 presenta el modelado de toda tarea robótica manipuladora con notación estándar UML. Esta sección concluye con la generación de los componentes para el middleware de OROCOS. La sección 4 ilustra un caso de estudio diseñado siguiendo los componentes y pautas de los apartados anteriores. Finalmente, en la sección 5 se presentan algunas conclusiones.

2 MODELADO DE COMPONENTES PARA ROBOTS MANIPULADORES

En esta sección se identifican y caracterizan los componentes software que intervienen en la ejecución de tareas de robots manipuladores.

En situaciones en las que un robot manipulador ha de interaccionar con un entorno no conocido, la información obtenida de éste, a través de los sensores

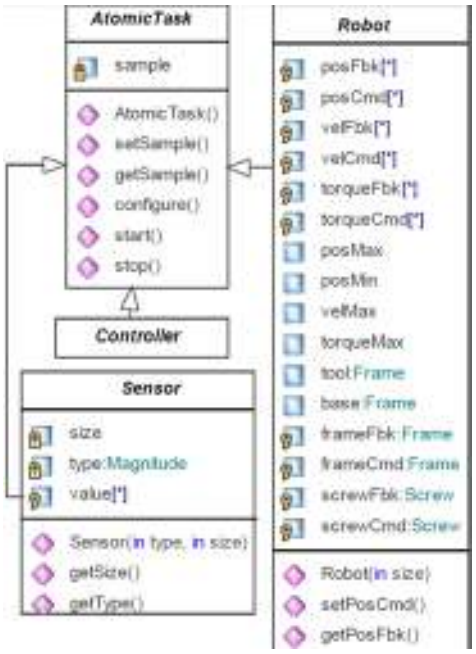


Figura 1: Caracterización de los componentes básicos de una tarea robot manipulador.

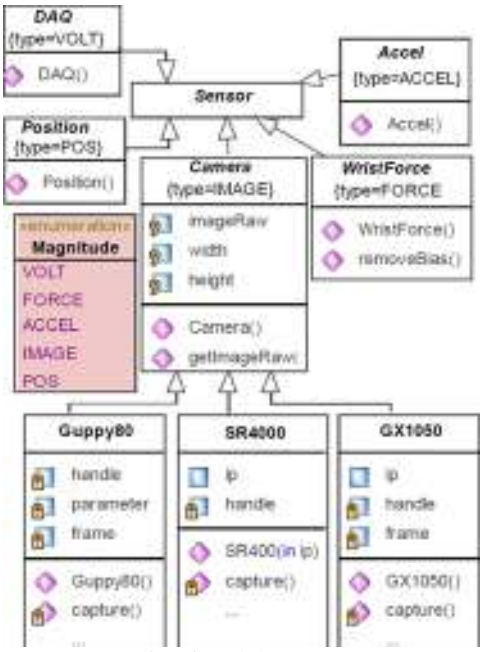


Figura 2: Caracterización del componente Sensor.

instalados, juega un papel determinante. Esto hace que el elemento *sensor* sea una unidad indispensable del modelo que describe la plataforma.

Además del propio *robot* (considerado como indivisible), otros elementos que se han de tener en cuenta son los *algoritmos de control*, los cuales, a partir de la información captada por los sensores, calculan las señales de control que se ordenarán al controlador de bajo nivel, encargado de gobernar el robot.

La figura 1 ilustra a través de un diagrama de clases UML las características comunes a todos los tipos de componentes identificados. Todas ellas, se encuentran agrupadas en la clase abstracta denominada *AtomicTask*. Como propiedad común está el periodo de muestreo (*sample*) para indicar a los sensores cada cuánto se ha de efectuar la medida, y a los robots, y a los algoritmos de control, cada cuanto se ejecutan.

Por otro lado, los métodos comunes están también ligados al estado en el que se encuentre el componente final desde el punto de vista del *middleware de comunicaciones*. En este caso, podrá estar en ejecución (*start*), o parado (*stop*). Además, siguiendo el paradigma de Orientación a Objetos, se dispone de un método *get* y otro *set* para manejar la propiedad *sample* (protegida).

En los siguientes apartados se detalla la caracterización de cada uno de los componentes.

2.1 SENSOR

Los sensores, además de las características heredadas de la clase abstracta *AtomicTask*, necesitan propiedades y métodos específicos (ver *Sensor* de la figura 1). La propiedad *type* se utiliza para indicar la naturaleza de la magnitud a medir. Actualmente se han considerado sensores para medir magnitudes como voltaje, fuerza, aceleración, imagen o posición. Por medio de la propiedad *size* se especifica el número de muestras a medir. Por último, el atributo de *value* se utiliza para proporcionar el valor de la medición. Este es un vector cuya longitud se relaciona con el tamaño del sensor. Al tratarse de propiedades protegidas, a todos ellos se les dota de un *get*/*set* para facilitar su consulta y actualización respectivamente.

Por otra parte, dependiendo del tipo de magnitud medida, los sensores requieren métodos y propiedades específicos adicionales (Fig. 2). Por ejemplo, tomando como referencia las cámaras, se puede apreciar que toda cámara hereda las propiedades y métodos comunes a todos los sensores y añade otros específicos, que a su vez, serán comunes a todo tipo de cámaras. Por ejemplo, la clase UML *Guppy80* de la figura 2 añade las propiedades y métodos privados proporcionados de los drivers del fabricante para manejar el dispositivo cámara Guppy F-080C.

Lo mismo ocurre con las clases *SR4000* y *GX1050*, definidas para el manejo de los dispositivos cámara tiempo de vuelo SR4000 y Prosilica GX-1050 respectivamente.



Figura 3: fichero de cabecera para la clase Guppy80.

La figura 3 ilustra el fichero cabecera de la clase Guppy80 para el código de manejo de su respectivo dispositivo (la cámara Guppy F-080C). En verde están resaltados los métodos comunes que ha de conocer el diseñador para utilizarla en una determinada tarea. En violeta, sin embargo, está resaltada la información propia de cada fabricante. Esto último es lo que ha de añadir un programador cuando necesite incorporar a la plataforma de trabajo la posibilidad de manejar un nuevo dispositivo cámara.

Esta misma filosofía se ha seguido para la caracterización del resto de tipo de sensores (*DAQ*, *Position*, *Acell*, *WristForce* de la figura 2).

2.2 ROBOT

El hecho de fijar el interfaz para el manejo de los robots permite a los diseñadores utilizar el mismo componente en diferentes aplicaciones. La figura 1, a través de la clase UML *Robot,* fija las características comunes a todos los robots manipuladores que se añaden a las que ya posee la tarea *AtomicTask*.

Un robot puede trabajar en coordenadas Cartesianas o Polares y su control de movimiento puede ser en posición, velocidad o fuerza.

Si el robot trabaja en espacio articular y el control es en:

- posición, las propiedades involucradas son *posFbk*, para la posición actual del robot y *posCmd*, para la posición demandada. Los métodos son *getPosFbk()*, para la obtención de la posición actual de cada articulación, y *setPosCmd()*, para establecer la posición a la que se desea ir. Además, se disponen de las propiedades *posMax* y *posMin* que definen los límites de posición alcanzables para cada una las articulaciones;
- velocidad, las propiedades involucradas son *velFbk*, para la velocidad actual de las articulaciones y *velCmd*, para la velocidad demandada de las articulaciones. Los métodos asociados a estas propiedades son *getVelFbk()* y *setVelCmd()*, respectivamente. El valor máximo que puede tomar la velocidad de las articulaciones está almacenado en la propiedad *velMax;*
- fuerza/par, las propiedades involucradas son *torqueFbk*, para el valor de fuerza/par aplicado a cada articulación y *torqueCmd*, para el comando de fuerza/par aplicado. Al igual que los modos anteriores se dispone de los métodos *getTorqueFbk()*, para obtener el valor de la fuerza/par aplicado y *setTorqueCmd()*, para establecer la consigna de fuerza/par a aplicar. La propiedad *torqueMax* establece la fuerza/par máximo aplicable por articulación.

Si, por el contrario, el robot trabaja en espacio cartesiano, es posible definir posiciones del extremo del robot X, Y y Z, junto con su orientación. La propiedad *tool* establece la relación entre el sistema de referencia del extremo del robot y la herramienta montada en el mismo. Existirán las siguientes posibilidades para el control del robot:

- Posición. Se dispone de la propiedad *frameFbk*, para obtener la posición en el espacio cartesiano y la propiedad *frameCmd*, para establecer la posición deseada en coordenadas cartesianas. Los métodos relacionados son *getFrameFbk()*, para obtener la posición y *setFrameCmd()*, para establecer la posición.
- Velocidad. Involucran a la propiedad *screwFbk*, para la velocidad actual del robot en el espacio cartesiano y *screwCmd*, para el valor de la velocidad deseada en coordenadas cartesianas. Los métodos relacionados con esta propiedad son *getScrewFbk()*, para obtener el valor de la velocidad actual y *setScrewCmd()*, para asignar una nueva referencia de velocidad al robot.

La plataforma desarrollada en la tesis [1] cuenta actualmente con dos manipuladores. El primero es un robot industrial Stäubli RX60 de 6 grados de libertad con un controlador CS8. En este caso, el fabricante ofrece dos posibles modos de funcionamiento para este robot: (1) a través del lenguaje propio VAL3. En este modo de funcionamiento la comunicación es posible llevarla a cabo mediante una conexión RS-232 o bien Ethernet TCP/IP. (2) acceso directo al control de las articulaciones con la librería LLI (Low Library Interface) [19]. En el presente trabajo se ha elegido la segunda opción. El ancho de banda de este modo de funcionamiento es 250Hz, por lo que el tiempo de muestreo es de 4ms, siendo cada articulación controlada únicamente por un PID.

El segundo robot es un robot antropomórfico Mekabot A2R3 de 7 grados de libertad, cuyo controlador reside en un PC con sistema operativo Linux modificado para soportar tiempo real (RETAI). Este robot dispone de un control por esfuerzo articular, posee un control de par para cada articulación y su forma y tamaño es similar a las de un adulto de estatura reducida. Sigue una configuración cliente-servidor. El cliente se encarga de definir las consignas de cada una de las articulaciones y el envío de cada una de las mismas al controlador específico de cada articulación. El servidor trabaja en tiempo real y permite la definición de nuevos controladores, así como conexiones para interactuar con primitivas del robot, pudiendo definir de esta manera nuevos modos de su funcionamiento.

La figura 4 ilustra las características adicionales utilizadas para los dos robots comentados anteriormente. Por ejemplo, la clase *MekA1*, añade la propiedad *socket* como un enlace de comunicación con su controlador de bajo nivel. *dataSend* y *dataRecv* se utilizan para el intercambio de datos.

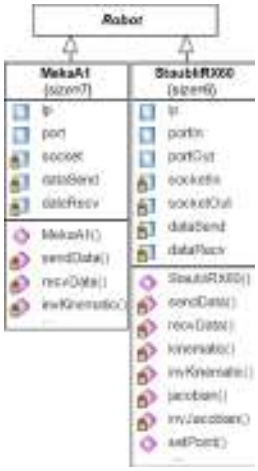


Figura 4: Caracterización del componente Robot.

Como el controlador de bajo nivel no proporciona cinemática inversa se ha añadido el método correspondiente.

2.3 ALGORITMOS DE CONTROL

Los algoritmos de control tienen como objetivo generar los valores de consigna del Robot. Los controladores se han agrupado en función de la señal de la variable manipulada de control que proporcionan. Ésta puede ser de tipos distintos, en función del espacio de trabajo: angular o cartesiano, y de distinta naturaleza, en función del control a utilizar: posición, velocidad o fuerza.
Todos estos algoritmos tienen únicamente común el hecho de calcular el siguiente estado del robot. Por esta razón, no tiene sentido la definición de una interfaz estándar que tenga las propiedades y métodos comunes a todos ellos.

3 MODELADO DE LAS TAREAS EN ROBOTS MANIPULADORES

El modelado de una tarea para robots manipulares consiste en la selección de los componentes (es decir, sensores, robots y algoritmos de control) de la plataforma que participen en la tarea. En este sentido es necesaria la figura de un planificador donde se detalle la lógica de la tarea y la secuencia de intercambio de información entre los componentes que la forman (ver diagrama de secuencia UML ilustrado en la figura 5).

En naranja se destacan las instancias de las clases que son necesarias para gestionar de forma aislada los *sensores* de la plataforma, los *robots* y *algoritmos de control*, que participan en una tarea robótica manipuladora.

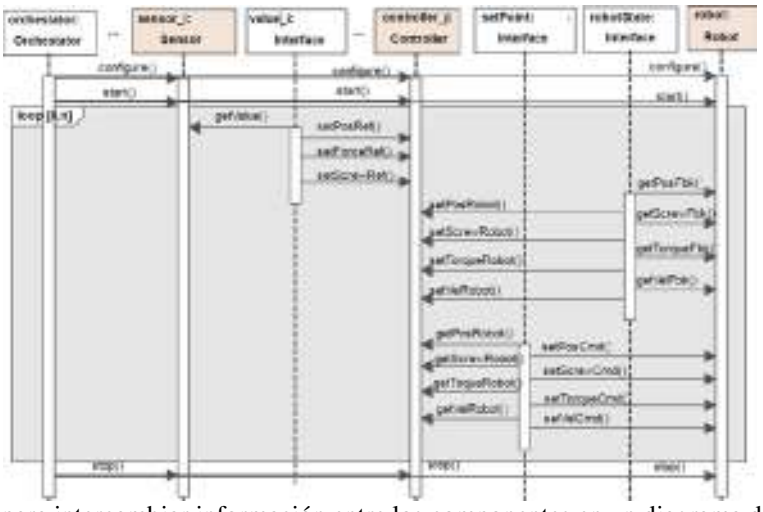


Figura 5: Plantilla para intercambiar información entre los componentes en un diagrama de secuencia UML.

Con un recuadro blanco se utilizan instancias de interfaces UML donde se almacenan datos intercambiables entre los componentes. Por ejemplo, *value_i* almacena la medida del *sensor_i* y *robotState* dispone el estado del robot necesario para el controlador (*controller_j*) en la generación de la orden de control (*setPoint*).
El control de ejecución de las tareas depende de los planificadores de los middleware de comunicaciones. Por tanto, es necesario desplegar los componentes que participen en la tarea en la plataforma con un determinado middleware de comunicaciones. La sección 2.1 de [3] presenta las características fundamentales de los middleware de comunicaciones más utilizadas en el campo de la robótica.

En este trabajo, se ha utilizado OROCOS, siendo el middleware basado en componentes más expandido. Este middleware proporciona un conjunto de librerías de las que cabe destacar *RTT-Real Time Toolkit,* ya que proporciona la infraestructura y funcionalidad para la generación de las aplicaciones basadas en componentes. Así, todo componente que aparezca en el diagrama de secuencia ha de ser transformado a un componente OROCOS. La clase *TaskContext,* ofrecida por RTT, fija el interfaz de todo componente OROCOS, que está compuesto por:

- Métodos: para definir el estado del componente desde el punto de vista del middleware. *constructor*, *destructor*, *configureHook*, *startHook* y *stopHook*.
- Puertos de datos: tienen como finalidad añadir accesibilidad exterior al componente.
- Propiedades: representan información de configuración del componente y pueden ser modificables en tiempo de ejecución.

La figura 6 ilustra el componente Guppy como ejemplo de componente OROCOS.

Por otro lado, cada interfaz presente en el diagrama de secuencia (Fig. 5) se corresponderá a un punto de conexión (*connectionPoint*) del fichero *deploymentComponent.xml* [18]. Este fichero es necesario para el motor del middleware por lo que, además, recoge información sobre las características de ejecución de los componentes OROCOS. Dicha información se obtiene procesando la propiedad *sample* de la clase *AtomicTask* (Fig. 1). Si dicho valor es diferente a cero implica que se ha de definir un hilo de ejecución periódico. Por lo contrario, si su valor es nulo, implica un hilo de ejecución no periódico. En relación al planificador, OROCOS, a través de dos primitivas, permite seleccionar entre un planificador en tiempo real (ORO_SCHED_RT) o no (ORO_SCHED_OTHERS).



Figura 6: Componente OROCOS Guppy.

4 CASO DE ESTUDIO: PLATAFORMA ROBÓTICA MULTI-SENSORIAL

Un escenario general de la plataforma se ilustra en la figura 7. Dicha plataforma consta de los siguientes componentes: tres cámaras- Prosilica GX1050C, SR4000 y Guppy F080C, sensores de fuerza (ATI); acelerómetros; giroscopios; un robot antropomórfico Staübli RX60 y un robot humanoide MEKA.

Para el caso de estudio, se ha seleccionada el robot industrial Staübli que tiene como finalidad interactuar con una superficie de la cual se desconoce su posición. Para el cálculo de la posición de dicha superficie se requiere de la cámara Guppy F080C y del algoritmo de procesamiento *fourPoint* para procesar la imagen capturada y calcular la posición objetivo. Se ha seleccionado un control de impedancia que a partir de la posición objetivo, la medida de la fuerza que está ejerciendo el robot y posición actual del mismo genera la siguiente orden. En este caso, el control de robot se hace en velocidad.



Figura 7: Escenario general de la plataforma.

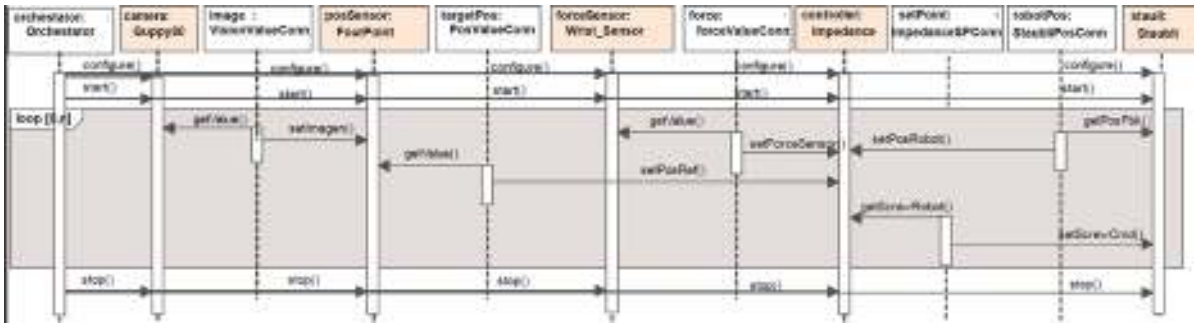


Figura 8: Intercambio de información entre los componentes de la tarea.

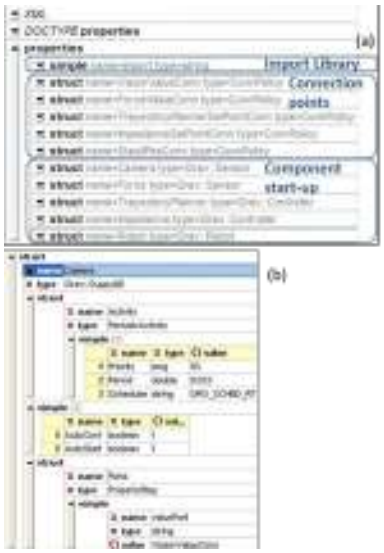


Figura 9: Fichero DeploymentComponent.xml.

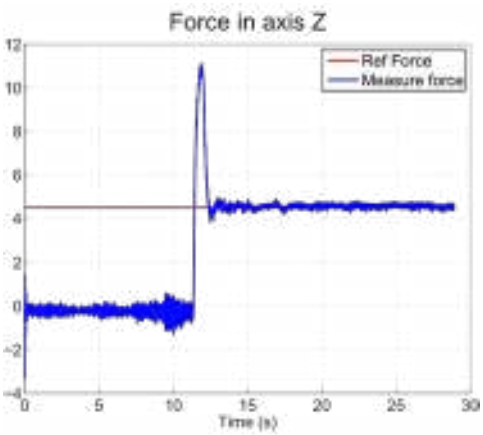


Figura 10: Evolución de la fuerza ejercida sobre la superficie a pulir.

La figura 8 ilustra a través de un diagrama de secuencia el intercambio de información entre los componentes involucrados (resaltados en naranja).

El planificador, en primer lugar, configura (*configure*) cada componente y posteriormente lo arranca (*start*). La cámara captura imágenes, con una resolución 640x480 píxeles, cada 33ms que se almacena en la instancia de interfaz *image* (con el método *getValue()*). El algoritmo de procesamiento de imágenes (*posSensor*) requiere de la imagen capturada y genera la posición del objeto procesado también cada 33ms. Dicha posición se almacena en *targetPos.* La fuerza ejercida por el robot se mide cada 1ms por *forceSensor*. El controlador CS8 del robot Staübli actualiza su posición cada 4ms. Dicha posición se almacena en *robotPos* (instancia de una interfaz UML llamada *StaubliPosConn*). Finalmente, el control de impedancia se ejecuta cada vez que haya alguna variaciones en la información que requiere, i.e. se ejecutan de forma no periódica.

En la parte (a) de la figura 9 se ilustra la estructura general que tiene el fichero de despliegue. En primer lugar, se indica la localización de los componentes OROCOS generados (Fig. 6). Posteriormente, por cada instancia de interfaz, se define un *ConnectionPoint.* Finalmente, se añade la información particular de la puesta en marcha de cada componente generado. En la parte (b) de la figura 9 se detalla la información para el componente OROCOS que maneja la cámara Guppy F-080C.

Por último en la figura 10 se puede apreciar cómo el robot ha tardado un poco más de 12 segundos en alcanzar la superficie de contacto, momento en el cual se comienza a contactar con la superficie con una fuerza igual a la consigna.

5 CONCLUSIONES

En este trabajo se ha hecho uso de las disciplinas de Ingeniería Basada en Componentes e Ingeniería Conducida por Modelos para dar soporte al ciclo de desarrollo de aplicaciones robóticas manipuladores.

Por un lado, se han definido y caracterizado las interfaces de los elementos comúnmente utilizados en este tipo de aplicaciones. El criterio seguido permite abstraer al diseñador del conocer los drivers para manejar cada uno de los dispositivos. Por otro lado, sirven como de guía y plantilla para los programadores que deseen añadir un nuevo componente en la plataforma.

El hecho de utilizar notación UML estándar para la definición de la lógica de las aplicaciones hace que la metodología propuesta sea genérica y válida en cualquier herramienta de modelado UML que soporte el interfaz XML Metadata Interchange.

Agradecimientos

Los autores quieren agradecer la subvención parcial de este trabajo a través de los proyectos DPI2011-27284, TEP2009-5363 y AGR-6429.

Referencias

[1] A. Sánchez, (2013) "Integración sensorial de fuerza, aceleración y visión en robots manipuladores con movimientos restringidos y entornos parcialmente estructurados". Tesis Doctoral.

[2] A. Brooks, T. Kaupp, A. Makarenko, S. Williams, y A. Oreback (2005), "Towards component-based robotics," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 163 – 168.

[3] A. Shakhimardanov, J. Paulus, N. Hochgeschwender, M. Reckhaus y G. K. Kraetzschmar, (2011) "D2-1 best practice assessment of software technologies for robotics", [Online]. Disponible en: http://www.best-of-robotics.org/pages/publications/BRICS_Deliverable_D2.1.pdf

[4] C. Schlegel, A. Steck, D. Brugali, y A. Knoll, (2010) "Design abstraction and processes in robotics: From code-driven to model-driven engineering," in Simulation, Modeling, and Programming for Autonomous Robots, LNCS, Eds. Springer Berlin/Heidelberg, vol. 6472, pp. 324–335.

[5] D. Alonso, C. Vicente-Chicote, F. Ortiz, J. Pastor y B. Álvarez (2010), "V3CMM: a 3-view component metamodel for model-driven robotic software development", *Journal of Software Engineering for Robotics*, pp: 3-17.

[6] D. Steinberg, F. Budinsky, M. Paternostro y E. Merks, (2008), EMF: Eclipse Modeling Framework, 2nd ed. Addison-Wesley Professional.

[7] D. Brugali y P. Scandurra (2009), "Component-based robotic engineering (Part I) [Tutorial]," *Robotics Automation Magazine, IEEE*, vol. 16, no. 4,pp. 84 –96.

[8] D. Brugali y A. Shakhimardanov (2010), "Component-based robotic engineering (Part II) [Tutorial]," *Robotics Automation Magazine, IEEE*, vol. 17, no. 1, pp.100 –112.

[9] D. Schmidt (2006), "Guest editor's introduction: Model-driven engineering," *Computer*, vol. 39, no. 2, pp. 25 – 31.

[10] G. Booch, J. Rumbaugh, y I. Jacobson, (2005), *The Unified Modeling Language User Guide, (2nd Edition)*. Addison-Wesley Professional.

[11] G. T. Heineman y W. T. Councill, (2001) Component-Based Software Engineering: Putting the Pieces Together, 1st ed. Addison-Wesley Professional.

[12] I. Sommerville (2010), *Software Engineering*, 9th ed. Addison-Wesley.

[13] J. Gamez, J. Gomez Ortega, A. Sanchez Garcia, y S. Satorres Martinez (2009), "Robotic software architecture for multisensor fusion system," *IEEE Trans. on Industrial Electronics*, vol. 56(3), pp. 766–777.

[14] J. Miller y J. Mukerji, "Mda guide version 1.0.1," 2003. [Online]. Disponible en: http://www.enterprise-architecture.info/Images/MDA/

[15] K. Balasubramanian, A. Gokhale, G. Karsai, J. Sztipanovits, y S. Neema, (2006) "Developing applications using model-driven design environments,"Computer, vol. 39, no. 2, pp. 33 – 40.

[16] Michael Geisinger, Simon Barner, Martin Wojtczyk, y Alois Knoll (2009), "A software architecture for model-based programming of robot systems," *LNCS, Advances in Robotics Reseachr*, Springer, pp. 135–146.

[17] "Orocos toolchain," (2002). [Online]. Available: http://www.orocos.org/wiki/orocos/toolchain/

[18] "Orocos - the deployment component," 2012. [Online].Disponible en: http://www.orocos.org/stable/documentation/ocl/v2.x/docxml/orocos-deployment.html

[19] Pertin, F. y des Tuves, J.-M. B. (2004). "Real time robot controller abstraction layer". In Proc. Int. Symposium on Robots (ISR), Paris, France.

[20] R. Bischoff, T. Guhl, E. Prassler, W. Nowak, G. Kraetzschmar, H. Bruyninckx, P. Soetens, M. Haegele, A. Pott, P. Breedveld, J. Broenink, D. Brugali, y N. Tomatis, (2010), "BRICS - best practice in robotics," 41st International Symposium on and 6th German Conference on Robotics (ROBOTIK), pp. 1 –8.

[21] S. Barner, M. Geisinger, C. Buckl, y A. Knoll, (2008) "EasyLab: model-based development of software for mechatronic systems," in IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, pp. 540 –545.

[22] Utz, H., Sablatnog, S., Enderle, S., y Kraetzschmar, G. (2002). Miro - middleware for mobile robot applications. *IEEE Transactions on Robotics and Automation*, 18(4):493 – 497.

EVENT DRIVEN MIDDLEWARE FOR DISTRIBUTED SYSTEM CONTROL

Manuel Muñoz, Eduardo Munera, J. Francisco Blanes, José E. Simó, Ginés Benet
Institute of Control Systems and Industrial Computing
Polytechnic City of Innovation
Polytechnic University of Valencia, Spain
{mmunoz; emunera; pblanes; jsimo; gbenet}@ai2.upv.es

Abstract

This paper presents a control kernel based middleware description. Capabilities and developed functionality are enhanced. The establishment of a event-based middleware control system offers a more reliable and efficient way to perform control tasks by ensuring a proper distribution of the requirements, actions and services between the system devices according to their availability of resources. Middleware is able to perform its capabilities over a distributed or partitioned system allowing to the programmer the control application design from a topology-independent point of view of the whole system. Quality of service (QoS) concept are introduced in data exchange providing a real-time communications between nodes. data distribution system (DDS) is used as base to achieve this reliable communications. A user interface has been designed to help the programmer in the middleware configuration task.

Key words: Real Time Systems, Embedded Systems, Distributed Control Systems, Control Kernel.

1 INTRODUCTION

distributed control systems (DCSs) are used extensively in industry to monitor and control distributed equipment with remote human intervention. Furthermore, civil engineering applications, or even usual consumer products, require or adopt a distributed platform as solution. In general, DCSs are real time embedded control systems and they are usually constrained by limited resources as computational power, memory, or network-bandwidth.

DCSs achieve a correct solution for most of the control scenarios. But technology changes drive its evolution by modifying its traditional morphology and design criteria. While computing power of embedded systems is increasing over time, moving the ad-hoc node implementation the to powerful embedded computers, networking technology trend is to move from control-specific networks to Ethernet based infrastructures and wireless communication. Even the spectacular computing power increment allow to partition a embedded system, obtaining different criticity subsystems with added benefits such as isolation and safety.

This paper is organized as follow: In Section 2 is introduced an approach to the control kernel (CK) and middleware (MW) architecture. Section 3 decribe the proposed topology for a distributed embedded control kernel while the middleware main services and features are described along Section 4, which manage to get an overview of the proposed system. In Section 5 is presented the human middleware interface and its functionalities for seting the services and manage middleware configurations. Finally, in section 6 is presented the conclusions which summarize the presented system and the benefits of proposed topology.

2 CONTROL KERNEL CONCEPT

The availability of a robust kernel or middleware can provide functionalities, services and interfaces to the hardware resources of the system. This paradigm leads to the specification of the CK concept as is presented on [1] which achieves an user transparent design that offers an execution management interface for the control application.

We consider a scenario where the control system components (sensors, controllers, operator terminals and actuators) are often spatially deallocated offering a distributed perspective, in which all of them are interconnected through a network. Even in the case of a mixed criticality system, composed by a set of partitions running on a virtualized environment, some kind of communication between components is a must.

CK interface is composed by a set of services that helps the designer to configure the control application and define the basic and prioritized control activities involved. Also can guarantee autonomous operation at the same time as guarantee fault tolerance and safety. The design of a flexi-

ble system can offer reconfigurability and hybrid control solutions.

The main objective of the CK development is to allow control application adapt to the growing complexity and functionality which impact control and real time embedded systems having to extends its services to the demand of new applications. That paradigm moves away from classical control sytems like PLC arrangements which are usually less evolved and offer poor programing capabilities. Instead of this it takes the advantage of the characteristics provided by microcomputers like DSPs which are optimized for high speed and real time data processing and also creates a more profitable way for software reuse.

The design of a CK improves the reliability of control activities execution by implementing a task model and execution containers. This will help to divide control operations into subtasks, which also can be isolated locally or distributed in several nodes. Task partitioning is performed by the use of code delegation techniques and adaptive configuration.

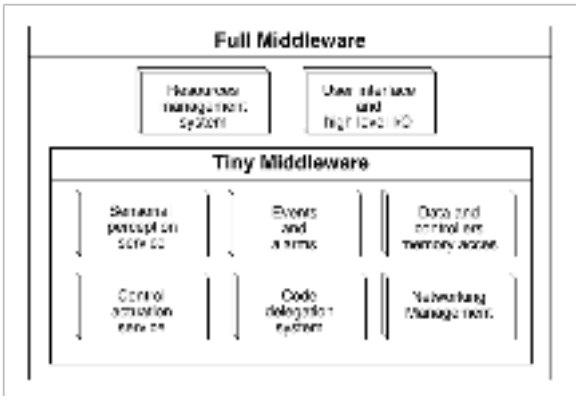


Figure 1: Control Kernel Blocks

An application running on top of control kernel middleware (CKM) is isolated from the operating system (OS) and the middleware is responsible of offering funcionalities and abstractions related to the software control components. It also have to provide real-time services (strict or not), management of the control tasks, access to its physical surrounding (sensors, actuators and plants), or communication management in both local and distributed way.

3 DISTRIBUTED EMBEDDED CONTROL KERNEL

Distributed Systems usually applies middleware technology in order to provide communication services and data exchange. There are available many middleware solutions: Java Message System (JMS) [5], Internet Communication Engine (ICE) [6], Common Object Request Broker Architecture (CORBA) [12], or data distribution system (DDS) [13]. Although there are different perspectives like a Message-Centric (JMS) or Data-Centric (DDS) system all of them aims to provide reliable and flexible services for asynchronous data exchange. JMS is based on the Java Platform and enables distributed communication by adding a common API for the development of message based applications in Java. ICE, CORBA, and DDS all of them make use of an Object Request Broker (ORB) which defines the interface for exchanged objects.The use of an ORB provide end-to-end Quality of service (QoS). Most ORB implementations are based on The ACE ORB (TAO) [17] which, as can be noticed in its name, is built on the Adaptive Communication Enviroment (ACE) framework [16].However the most evolved of them is DDS which is created with the prpose of standardize data distribution frameworks. DDS provides a high performance QoS-based service which offers a very adaptable communication.

A proper QoS setting is a crucial factor for any network connecting nodes with distinct available resources, so services must have the capability to adapt different system restrictions. For that reason there must be taken into account the differences among the management of QoS in each system. In this way [14] introduced a survey for QoS support on some communication systems, including the previously mentioned and some more, explaining its own characteristics. Also [7] presents a study of the adaptability and QoS needs of middleware according to the offered typology of event-dissemination in the system.

More extended and specific middleware implementations can be found as Open RObot COntrol Software (OROCOS) [3] which offers a hard real-time framework for control systems providing tools for data exchange and event-driven services. On the other hand some robotic-specific frameworks like Carmen [11] or Robot Operating System (ROS) [15] performs successfully but lack of the desired generality. In every case these frameworks do not provide real-time communication with fault tolerance system.

Proposed solutions aims to offer similar extended services but preventing the unavoidable restrictions found in other frameworks. Adaptability to physical embedded control systems is the key of the system, this goal is reached by exploding the capabilities of event-dissemination and QoS aware real-time communications.

3.1 Underlying Topology

From this background, the shape of a distributed control network could be depicted without loss of generality as shown in Figure 2.

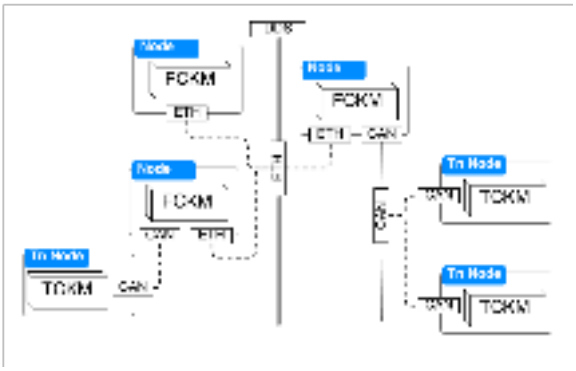


Figure 2: Real Time Control System Topology

Every component in the heterogeneous network could be characterized by its own performance and capabilities. From that, two type of nodes could be distinguished: the less powerful and cheaper (or light) nodes and the powerful service nodes.The difference between these nodes are the capabilities to offer services, according their resources limitation. In the same way, light nodes include a CAN bus connection (playing the role of a low-cost specific bus), while powerful nodes additionally have a ethernet network.

3.2 Light nodes and Tiny Middleware

The Tiny-Middleware Control Kernel (TCKM) is designed for working on less performance nodes and offers a cost-effective solution with low power consumption. Usually these nodes are physically closer to the controlled process and are responsible of low level operations as acting/sensing task. Relevant information is exchanged with the service node which, is the responsible of the high level tasks and code delegation. Light nodes also provide safety mechanism in order to mitigate and alert about defective operations through the rise events and alarms and its handlers. In order to detect this situations, execute handlers and raise events, the TCKM includes monitoring tasks supervising the state of the system. TCKM main services can be organized in the following categories:

- Sensorial perception service: This service interacts with sensors in order to provide data at the required sample period.
- Control actuation service: Is in charge of applying the calculated control action and interacts with the plant trough actuators.
- Code delegation: Offers a mechanism of adaptability by performing the delegation of functions from nodes with a high load to idle ones in order to obtain better use of the resources.
- Monitoring and events/alarms system: Provides fault tolerance by the monitoring the available resources and the controllers performance, raising alarms in case of a anomalous situation..
- Data and controllers: Deal to obtain transparency between data in local memory and the distributed objects. It also manage the access and switch between 'controllers pages' reserved in memory.
- Network Management: Manage all the network resources allowing the communication between the system nodes through both ethernet or can.

3.3 Service nodes and Full Middleware

The basic TCKM services are extended in the Full-Middleware Control kernel (FCKM) with the aim to expose a richer an interface to the control application. Powerful service nodes which hosts FCKM are the responsible of the execution of most complex control loops in order to provide a high quality control. At the same time the FCKM take profit of the high capabilities of Input/Output (IO) and networking of the Service Node. Resource management employs all the process information in order to adapt the system to the control requirements. Code delegation is the main mechanism to perform this adaptation as will be introduced. The desired control requirements are defined by the declaration of a QoS record which defines the necessary conditions to guaranty the performance and reliability of the designed control. Exclusive FCKM services can be synthesized as:

- Manage and guarantee specified QoS:Relying on DDS QoS mechanisms service nodes must guarantee the fulfillment of the requested policies, informing the system otherwise.
- Resources Management: This service performs an optimal use of the availables resources in the system by the application of code delegation mechanisms.
- Interface control application and high level I/O: As most control systems provides an HMI and SCADA interfaces, proposed system must also provide an interface for the control application and high level I/O for end-user HMI devices.

3.4 Virtualized and Partitioned Nodes

Virtualization [4, 9] is a well know technology in the field of powerful computers and servers. The increasing of processing power of embedded processors has opened new possibilities of virtualization application in the embedded systems market. One example of virtualization software, focused on embedded systems development, is the open source project XtratuM [10] used in our prototype.

Talking about embedded control systems it is mandatory to address the problem of putting together different subsystems having different criticality levels. Some application-critical elements are essential for the mission performance because safety reasons (p.e. airplane flaps servocontrollers) while others (p.e. graphical interfaces or comfort control) can be shutdown and restarted with minimal impact on the mission performance. The design trend in Distributed Control Systems is to spread the critical tasks on dedicated computing nodes and then, replicate them. Critical activities run on dedicated nodes in order to avoid side effects caused by faults in other (normally less critical) activities. On the other hand, preventing hardware failures these nodes are 3x replicated sending computing results to some kind of voting unit witch decides the correct result and detects possible failures. Following this design pattern "n" critical activities will need at least "3n" computing nodes in the distributed infrastructure.

On the other hand, virtualization technology provides temporal and spatial isolation properties. Spatial isolation is understood as the inviolable assignment of hardware memory areas and peripherals access to "partitions" running in the same computing node. Temporal isolation is the predictable and inviolable computing time assignment to "partitions" running in the same computing node. The use of virtualization technology lead us to a more simple and efficient engineering pattern running "n" redundant critical tasks on 3 computing nodes holding "n" partitions each. See Figure 3. Different partitions hosted in the same node, can run different Operating Systems and can be loaded and restarted independently. In this way, a partition can be seen as a separated distributed node, but in practice, the communications abstractions and discontinuous temporal profile available at partition level difficult the application of the partitioned design pattern.

Control Kernel Middleware has been extended to support partitioned nodes in a transparent way. Control applications run using Middleware services witch implementation and mechanisms differ with the underlying infrastructure. This independence is one of the key requirements from the earliest implementations of the Middleware.

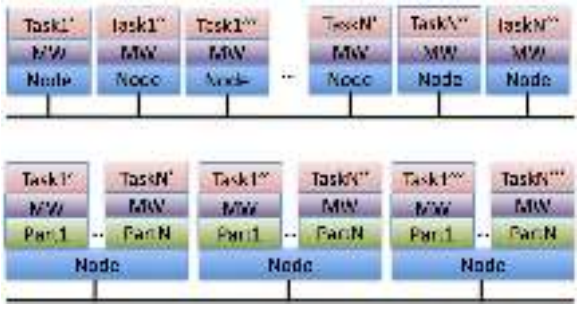


Figure 3: Virtualization

4 MIDDLEWARE GENERAL FEATURES

Below will be described some of the major features performed by the proposed CKM. These functions have been grouped around services:

4.1 Sensory Perception Service

Sensor measurements are acquired according to the time periods required by the application and suitable for each sensor. Data acquisition delays should not be reflected as control loop delays, so, the instant when data is captured must not affect to the controller computation start neither the instant when control actions are delivered. Its supposed that controllers are calculated for a specified period, the variations on it, known as jitter[2, 8], entails losses in control quality. CKM provide a extrapolated delay-free measure calculated for the instant when control task starts as well as actuation service.

Acquired measures must be adapted from raw sensor values to processed understandable information. This adaptation affects to the capacity of integration between nodes to the point that every element in the system can read and understand any data that has been provided by the data service. That way, CKM provide some primitive functions that allow this adaptation (e.g. raw analog digital converter (ADC) data form 10bits values to -5 to 5 range).

4.2 Control Actuation Service

The main provided functionality is to supply a valid control action at each moment to the corresponding actuator. To avoid possible system failures, has been implemented a safety mechanisms that execute a secure routine in case that any control action has arrived from the controller.

In that situations the programmer can choose between several options, ranging from acting with the last available value, to applying a safe-stop maneuver, allowing to force a concrete dynamic performance in the actuation signal.

In this line the service allow to use the algorithm described in [18] where the proposed control model allows to perform a set of basic activities to ensure the safe operation of the controlled system. In order to provide this functionality, a set of future control actions are extrapolated using a generalized predictive control (GPC) controller, and stored by the actuation service. If is not possible to close the loop, then this service will use these stored actions until the data losing problem is solved.

Other functionality offered by this service is the adequacy of the information provided from the data service to a type understandable by the actuator, just as the inverse of the adjustment made by the sensing service.

4.3 Monitoring and Alarms System

Monitor is in charge to obtain the information that represents current or previous system states as well as the resource utilization. Mentioned state information can be requested by the application to take appropriate decisions about the actions to perform, like code delegation, mode exchange, disconnection, or any other that has been required.

Additionally an alarm system has been developed offering the application to subscribe a set of states or situations where pretends to be warned by the middleware. This module takes information from the monitor and checks if the system can provide the quality previously agreed, otherwise an specific alarm will be raised.

4.4 Resource Management

Resources must be managed and adapted in order to satisfy specified QoS. For that reason different performance QoS levels are established for control application execution. These criteria should be agreed before the execution starts. Also must be defined the conditions for identify high computational load situations in order to perform appropriate adaptation mechanisms. Early actuation on anomalous situations can avoid to reach malfunction situations which can be hard to solve. Moreover, it guarantees a high performance of the system because a node never come under overload, whenever another node in the system presents enough available resources.

4.4.1 Code Delegation

In order to provide the system has been integrated in the middleware a code delegation mechanism. This service allow to exchange sections of code between system nodes. However, only some specific functions or modules are suitable to be exchanged. Delegation of code segments can be performed at runtime allowing its execution at anytime. Main delegated codes are related to a certain action just as a new controller execution or the addition/removal of a sensing process. That functionality allows the middleware to move code from one busy-node to another load-free. Among the criteria for perform these movements of code, not only computational load levels are taken into account, but also all those criteria which improve the use of resources.

4.4.2 Controller Switching

Controllers can be delegated, added, or deleted, but this actuations could entail a unstable state. Middleware guarantees the proper switching between controllers. The suitable switch strategy should be chosen by the control designer. If the previously agreed conditions are not properly respected, actuation service would be responsible for delivering safe control action to the system.

4.4.3 Scheduling and Flexible Tasking

Tasks will be optimally planned based on current control objectives, the quality levels and the amount of resources available. Likewise, we consider in some cases the use of optional tasks that will be performed only when is available an excess of resources. Following this concept control algorithms can be divided into a required and an optional part, being this last an optional task. That implies that high quality control will be performed when system have enough resources and otherwise executing a basic control. Optional tasks can also be delegated to any other node with available resources thanks to the code delegation paradigm.

One last adaptation mechanisms is the design of flexible tasks which can modify its parameter of execution at run time. It is well known that, in general, quality control decreases if the sampling period is increased, but it is also known that the lower the sampling period greater resources will be required. Therefore, if the sensory data availability changes, the task period also should change control for optimization, which imply changes in the parameters of the controller and the information stored. This assures the availability of a control action at anytime, and consequently a safe

operation, although it is forced to discard a high quality control.

4.5 Network Management

Being a distributed system communications between nodes are really important. The topology shown in the Figure 2 is taken as reference for this development. In it you can see two different buses that must be handled by the communications service.

The communication between service nodes will rely on a DDS-based implementation which in turn will employ TCP as transport protocol.

Tinny nodes use an ad-hoc protocol over CAN bus, maintaining some QoS properties. A service node is connected as gateway to DDS network, in that way the information in this bus is accessible for the rest of service nodes.

In a partitioned node, communications between partitions are implemented using queuing mechanisms offered by the hypervisor, so that increase the performance compared to the other buses.

In all cases, the middleware is responsible for managing the access to all the information with a common interface.

4.6 Data

The middleware provides transparent access to distributed (or partitioned) data in control applications, and between internal and external devices (sensors and actuators), so that distributed components can be accessed regardless of their location. Although this, data must also meet with several QoS. This is a significant step in order to reach the desired reliability of the system. This can be specially remarkable in the dissemination of sensor data in which QoS policies like "Deadline" or "Lifespan" can prevent to compute a mistaken control action using delayed values.

5 DEVELOPMENT MODEL

The CK proposed in this work has been implemented as a Middleware, that configures a software layer between the user applications and the OS or hardware abstraction layer (HAL). Middleware offers a set of services to the control application. Every node in the system runs a Middleware instance in the proper version (TCKM or FCKM) depending on their capabilities.

5.1 DEVELOPMENT INTERFACE

Control engineer develops the application from the whole system point of view.The aim is to ease as much as possible the application design process. For that reason an user interface has been developed, which allows the user to design the control application avoiding required real time (RT) programming, communications, etc. In Figure 4 can be observed a capture of the configuration and edition application.

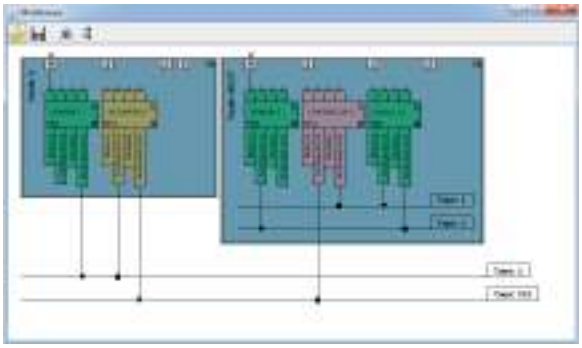


Figure 4: Configuration editor interface

Once the design is fulfilled using the interface, application generates xml files that will be used by the MW to load the application. This files description is shown in section 5.4. Using this process, manual data introduction is avoided, so a big amount of mistakes is prevented. Interface also allows to connect with the middleware at run time for handling the configuration dynamically.

5.2 MIDDLEWARE INITIALIZATION

When the Middleware starts up all the services are initialized, but no application is loaded. To load the desired application, the manager module is called to launch the proper services. TCKM or FCKM follow different strategies to achieve this goal:

- In the FCKM nodes, which are able to manage xml files, a initialization procedure is launch, where the configuration is read from the file and applied directly. If none configuration file were provided, the services remain in waiting mode, till some of them are called through the network.

- In every TCKM nodes, who can not manage any file, configuration process is not launch. Instead, a FCKM node reads the configuration and calls the TCKM node using the network interface to request its configuration.

5.3 CONFIGURATION MODEL

It has been defined a configuration model based in xsd files which specify all the needed parameters for starting the system offered services and load the desired application. Hierarchical model allow to extract partial configuration from distinct modules and/or components. This model contains precise information about the definition of the required parameters to perform a service request and its configuration.

5.4 CONFIGURATION FILES

The configuration is obtained through the use of xml files, who are generated from the model. In the same way as the model, xml files can describe the configuration both totally or partially. Every xml contains valid configurations in any case that is able to be validated itself against the model through standard xml/xsd validation mechanisms.

Once the services has been initialized, at the system start up, the middleware is able to load this xml configuration files. After loading and validating these files are performed the needed middleware calls for setting the described configurations.

In addition, in those cases that the user perform configuration changes during the execution, these changes can be stored within the rest of configuration files for future executions.

Configuration files can be edited in manual mode using a text editor, and following the described criteria in the model. However this edition mode is very unfriendly and can easily lead to errors. For that reason it has been developed an user interface which allows the user to generate and modify configurations in a practical way, without the need of a deep known about the model

6 Conclusions

A middleware based on the Control Kernel concept has been developed. This software allows the development of complex control applications. The establishment of a event-based middleware control system offers a more reliable and efficient way to perform control tasks by ensuring a proper distribution of the requirements, actions and services between the system devices according to their availability of resources. That way, the enhancement of the interaction between service and light nodes helps to make use of its own characteristics in a more effective way. Also, DDS-based communication system allows to exploit QoS benefits improving the interaction between nodes by providing a reliable real-time communication, as well as promotes the generation of events. The design of a new user interface facilitates the system design and helps to specify and manage configurations through the use of xml files, as a result, workflow is speed up considerably. An early version of proposed system has been implemented in order to test the viability and performance. Future works will be focused on performance improvements related with the management, generation, and handling of events, and extend the QoS unit parameters.

Acknowledgment

This work has been supported by the Spanish Science and Innovation Ministry MICINN under the CICYT project COBAMI: DPI2011-28507-C02-01/02. The responsibility for the content remains with the authors.

References

[1] P Albertos, A Crespo, and J Simó. Control kernel: A key concept in embedded control systems. In *4th IFAC Symposium on Mechatronic Systems*, 2006.

[2] KE Arzén, A Cervin, J Eker, and L Sha. An introduction to control and scheduling co-design. In *39th IEEE Conference in Decision and Control*, Sydney, 2000.

[3] H Bruyninckx. The real-time motion control core of the Orocos project. *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference*, 2:2766–2771, 2003.

[4] A Crespo, I Ripoll, and M Masmano. Partitioned Embedded Architecture Based on Hypervisor: The XtratuM Approach. In *2010 European Dependable Computing Conference*, pages 67–72. IEEE, 2010.

[5] M Hapner and R Burridge. Java message service. Technical report, Sun Microsystems Inc, Santa Clara, CA., 2002.

[6] M Henning and M Spruiell. Distributed programming with ice. Technical report, ZeroC Inc., 2003.

[7] SP Mahambre, M Kumar, and U Bellur. A taxonomy of qos-aware, adaptive event-dissemination middleware. *Internet Computing, IEEE*, 11(4):35–44, 2007.

[8] Pau; Marti, Josep M; Fuertes, Gerhard; Fohler, and Krithi; Ramamritham. Jitter compensation for real-time control systems. In *Real-Time Systems Symposium, 2001.(RTSS 2001). Proceedings. 22nd IEEE*, pages 39—-48, 2001.

[9] M Masmano, I Ripoll, and A Crespo. XtratuM : a Hypervisor for Safety Critical Embedded Systems Virtualising technologies overview. In Tecom Project, editor, *11th RealTime Linux Workshop*, 2009.

[10] M Masmano, I Ripoll, A Crespo, and JJ Metge. Xtratum: a hypervisor for safety critical embedded systems. In *Proc. of Real-Time Linux Workshop.*, 2009.

[11] M Montemerlo, N Roy, and S Thrun. Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference*, 3:2436–2441, 2003.

[12] Object Management Group (OMG). The Common Object Request Broker (CORBA): Architecture and Specification. Technical report, Object Management Group, 1995.

[13] Object Management Group (OMG). *Data Distribution Service for Real-time Systems.* Version 1 edition, 2007.

[14] JL Poza-Luján. A Survey on Quality of Service Support on Middleware-Based Distributed Messaging Systems Used in Multi Agent Systems. *International Symposium on Distributed Computing and Artificial Intelligence*, pages 77–84, 2011.

[15] M Quigley and K Conley. ROS: an open-source Robot Operating System. *ICRA workshop on open source software*, 3(3.2), 2009.

[16] DC Schmidt. The adaptive communication environment. Technical report, DOC group, Washington University, 1994.

[17] DC Schmidt. TAO, The Ace Orb. Technical report, DOC group, Washington University, 2007.

[18] R Simarro, J Coronel, J Simó, and JF Blanes. Hierarchical and distributed embedded control kernel. In *17th IFAC World Congress*, 2008.

Herramienta basada en MDE para sistemas orientados a servicio dinámicamente reconfigurables

Aintzane Armentia*, Rafael Priego*, Marga Marcos*, Elisabet Estévez**, Francisco Gómez Molinero†

* ETSI Bilbao, UPV/EHU
{aintzane.armentia, rpriego001, marga.marcos}@ehu.es

** Escuela Politécnica Superior de Jaén, UJA
eestevez@ujaen.es

† Visual Tools SA
fgomez@visual-tools.com

Resumen

Las arquitecturas orientadas a servicio son cada vez más usadas para hacer frente a los exigentes requisitos que demandan las complejas aplicaciones actuales. Por otro lado, con el objetivo de gestionar el dinamismo inherente a las aplicaciones distribuidas, se utilizan plataformas de middleware que controlan su ejecución. Estos middlewares están dotados de mecanismos de reconfiguración que permiten cumplir con los requisitos de la aplicación, optimizando así los recursos del sistema. En este contexto, el uso de técnicas de meta-modelo, como es el caso de la Ingeniería Basada en Modelos (Model Driven Engineering, MDE), permite facilitar el diseño y desarrollo de estos complejos sistemas al poder representarlos desde diferentes niveles de abstracción. El presente trabajo propone una herramienta de modelado basada en MDE que permite un desarrollo guiado y validado de aplicaciones dinámicamente reconfigurables orientadas a servicio, de cuya gestión se encarga una plataforma de middleware que asegura el cumplimiento de sus requisitos temporales, reconfigurando si fuera necesario. Esta herramienta también encapsula la interacción con dicho middleware, abstrayendo al usuario final de un conocimiento exhaustivo de la plataforma.

Palabras Clave: Ingeniería basada en modelos, aplicaciones basadas en servicios, reconfiguración, Graphical Modeling Framework.

1 INTRODUCCIÓN

Las complejas aplicaciones actuales demandan arquitecturas software más completas, entornos distribuidos y heterogéneos, así como una asignación de recursos óptima que garantice que sus requisitos no funcionales se cumplen. Introducir dinamismo durante la ejecución de la aplicación puede ayudar a cumplir con sus requisitos y optimizar los recursos. Este dinamismo puede ser implementado como una reconfiguración llevada a cabo en tiempo de ejecución que cambie la arquitectura de la aplicación.

Para hacer frente a estos exigentes requisitos, se está extendiendo el enfoque de las arquitecturas orientadas a servicio, principalmente en aplicaciones distribuidas que comprenden diferentes tecnologías y plataformas [23][5]. Este paradigma permite el desarrollo de aplicaciones software como colecciones de servicios débilmente acoplados que interactúan [3]. Es más flexible que otros enfoques más tradicionales, ya que facilita la reutilización y la interoperabilidad, evitando dependencias de plataformas y reduciendo costes y tiempo de desarrollo [11]. Es más, también se mejoran la escalabilidad y flexibilidad del sistema haciendo posible la evolución de la aplicación con respecto a requisitos cambiantes. Como consecuencia, se espera que estas aplicaciones sean menos costosas de mantener y más fáciles de escalar que las soluciones personalizadas tradicionales [17].

En sistemas de estas características, resulta casi imprescindible la utilización de una plataforma que se encargue de la gestión global del sistema en aspectos tales como el registro, despliegue inicial de las aplicaciones, el control de ejecución de las mismas, la reconfiguración dinámica, la gestión de recursos y el control de admisión. Esta plataforma puede ser implementada por un middleware como en [2] y [9].

Debido a que estos complejos sistemas son difíciles de diseñar y desarrollar, la consolidación de metodologías de meta-modelado resulta beneficiosa. De hecho, en el desarrollo de aplicaciones software, en los últimos años se ha experimentado una clara

tendencia a pasar de aproximaciones centradas en el código a aproximaciones centradas en modelos [4]. Estas metodologías se basan en la utilización de lenguajes de modelado que permitan una completa descripción de dichos sistemas antes de su construcción, posibilitando la detección temprana de errores en fase de diseño. El Desarrollo Basado en Modelos (Model-Driven Development, MDD) es un paradigma emergente que permite resolver muchos de los problemas asociados con la composición e integración de grandes sistemas. Se basa en el uso de modelos para representar los elementos de un sistema y sus relaciones [19]. La Ingeniería Basada en Modelos (Model-Driven Engineering, MDE) es una metodología que promueve el uso intensivo de modelos para hacer frente a la complejidad inherente al diseño de aplicaciones. MDE va más allá que las simples actividades de desarrollo, de hecho combina otras tecnologías como por ejemplo el uso de lenguajes de modelado específicos de dominio y generadores o transformadores. Por un lado, los lenguajes de modelado se basan en meta-modelos, permitiendo la descripción de la estructura, comportamiento y requisitos de un sistema. Por otro lado, las transformaciones de modelos permiten el análisis de los modelos para la obtención bien de código (transformaciones modelo-texto, M2T) bien de otros modelos (transformaciones modelo-modelo, M2M) [18].

En la literatura se pueden encontrar diferentes trabajos que han desarrollado herramientas basadas en modelos para arquitecturas orientadas a servicio como [20] para servicios web, [21] centrada en el control de la tolerancia a fallos y [14] que propone una clasificación de los servicios incluyendo un tipo para aquellos que se encargan de la gestión del proceso de negocio (orquestación). Todos ellos presentan propuestas muy interesantes, pero que implican un conocimiento exhaustivo de la plataforma subyacente.

Este trabajo presenta una herramienta basada en modelos, de ahora en adelante MTS (Modeling Tool Suite), que guía y valida el desarrollo de aplicaciones dinámicamente reconfigurables orientadas a servicio, cuya ejecución es gestionada por una plataforma de middleware. La MTS guía al usuario desde la especificación de la aplicación hasta su ejecución, a través de la generación de código. La herramienta desarrollada proporciona mecanismos para la definición de requisitos funcionales y no funcionales (centrados en requisitos temporales) y para la generación automática del esqueleto del código de los servicios. Toda esta información es pasada al middleware encargado de gestionar tanto los servicios como los recursos compartidos, asegurando que los requisitos no funcionales de la aplicación se cumplen.

La estructura del artículo es la siguiente: la Sección 2 presenta una breve descripción de los conceptos básicos de la propuesta. La Sección 3 describe un prototipo de la MTS, que se basa en la aproximación de modelado propuesta, y que es usado en la Sección 4 para el desarrollo de una aplicación de video-vigilancia. Finalmente en la Sección 5 se presentan las conclusiones.

2 CONCEPTOS BÁSICOS

El trabajo presentado en este artículo fue desarrollado dentro del proyecto iLAND (mIddLewAre for deterministic dynamically reconfigurable NetworkeD embedded systems) [13]. Uno de los principales objetivos del proyecto consiste en el desarrollo de un middleware que proporciona gestión de la ejecución y reconfiguración acotadas en el tiempo de aplicaciones distribuidas basadas en servicios [9]. Otro de los objetivos consiste en el desarrollo de una aproximación de modelado, que es descrita en esta sección.

Una aplicación iLAND, a partir de ahora una *SOA* (Service Oriented Application), es un caso particular de aplicaciones basadas en servicios, formada por un conjunto de servicios que cooperan para lograr la funcionalidad de la aplicación. En el contexto de dicho proyecto, un *servicio* es una entidad software, con una interfaz bien definido, que proporciona una determinada funcionalidad, la cual es representada por el elemento *función de servicio* (código reutilizable).

Un servicio consta de un *puerto de entrada* y/o un *puerto de salida* que por un lado proporcionan acceso a la *función de servicio*, y que por otro lado encapsulan la recepción y el envío de datos. Por lo tanto, un servicio recibe datos a través de su *puerto de entrada*, encargado de ejecutar la lógica de conexión de entrada obteniendo los parámetros necesarios para la ejecución de la *función de servicio*. Del mismo modo, su *puerto de salida* ejecuta la lógica de conexión de salida con el objetivo de obtener los datos que se enviarán a otros servicios, a partir de los parámetros generados tras la ejecución de la *función de servicio*. Resulta importante destacar que los parámetros de entrada necesarios para la ejecución de la *función de servicio* pueden provenir de más de un origen, y que los parámetros de salida obtenidos pueden ser enviados a más de un servicio. Es más, este envío de datos puede ser realizado bajo determinadas condiciones. Para recoger todas estas diferentes situaciones se han definido cuatro tipos de lógica:

- ***Lógica de entrada WAIT***: tipo de lógica por defecto según la cual para la ejecución de la *función de servicio* se debe esperar a que

todos sus parámetros de entrada (tengan uno o más orígenes) estén disponibles.

- ***Lógica de entrada NO_WAIT***: en este caso también se permite la recepción de los parámetros de entrada desde varios orígenes. A diferencia de la anterior, cada uno de los orígenes debe enviar todo el grupo de parámetros, de manera que en cuanto estén disponibles, por parte de cualquiera de los posibles anteriores, se ejecuta la *función de servicio*.
- ***Lógica de salida AND***: tipo de lógica por defecto según la cual siempre se envían datos a uno o más destinos, de la misma manera.
- ***Lógica de salida GENERIC***: este tipo permite el envío de datos bajo determinadas condiciones que se pueden expresar a través de un diagrama de actividad [22], basado en los parámetros de salida de la *función de servicio* encapsulada.

Para la comunicación entre servicios se define el elemento *conector* encargado de recoger los datos intercambiados. Los conectores tienen como origen un puerto de salida y como destino un puerto de entrada. Un servicio no tiene por qué enviar todos sus parámetros de salida a todos sus servicios siguientes, y del mismo modo, puede recibir sus parámetros de entrada desde más de un servicio. Por lo tanto, resulta necesario el establecer una *conexión* entre cada uno de los parámetros de salida de un servicio origen y cada uno de los parámetros de entrada del servicio destino, unidos a través de un *conector*.

La reconfiguración a nivel de aplicación implica establecer relaciones entre aplicaciones, ya que las decisiones tomadas en tiempo de ejecución por una, pueden afectar a la creación o incluso destrucción de otras. Para expresar, en fase de diseño, estas relaciones entre aplicaciones, se introduce el elemento *evento*, que representa la necesidad de creación/destrucción de aplicaciones y de reserva/liberación de recursos. De nuevo, la lógica para el lanzamiento del *evento* se describe a través de un diagrama de actividades asociado al puerto de salida. Se distinguen cuatro tipos de eventos: 1) *Create*, se solicita la creación de una nueva aplicación sin que ello afecte a la ejecución de la actual; 2) *Destroy*, se solicita la destrucción de una aplicación que se encuentra en ejecución, manteniéndose la de la actual; 3) *Replace*, se solicita la creación de una nueva aplicación destruyéndose la actual; 4) *Switch*, relaciona dos aplicaciones que se ejecutan de forma alternativa.

Por lo tanto, la funcionalidad de una aplicación se puede definir como un grafo formado por un conjunto de servicios interconectados, en el que cada servicio encapsula una funcionalidad concreta, la lógica de conexión con otros servicios y la lógica de reconfiguración. Además, las aplicaciones también pueden llevar asociados requisitos no funcionales que demandan al middleware, como por ejemplo el periodo si la SOA es periódica, el nombre del evento que la activa si fuera así, un plazo de extremo a extremo, etc.

Por último, se debe tener en cuenta que cada servicio puede ser implementado por varias *implementaciones de servicio* que pueden ser desplegadas en diferentes nodos. Proporcionan la misma funcionalidad pero con diferentes atributos no-funcionales como el tiempo de procesamiento.

3 PROTOTIPO DE LA HERRAMIENTA DE MODELADO

Tal y como se ha comentado anteriormente, cualquier propuesta basada en MDE debería estar provista de su correspondiente herramienta de soporte cuyo punto de partida debe ser un meta-modelo previamente definido, y que proporcione una interfaz gráfica de usuario (GUI, Graphical User Interface) amigable. En esta sección se presenta un prototipo de la MTS desarrollada para el middleware de iLAND, que guía y valida el desarrollo de aplicaciones. Para ello, se desarrolla un lenguaje de dominio específico basado en la aproximación de modelado descrita en el apartado anterior, junto con un conjunto de transformaciones M2T para la generación de código.

La Figura 1 muestra el escenario general de la MTS, en el que ésta actúa como un intermediario entre el usuario final y el middleware de iLAND. La MTS permite la definición de aplicaciones, capturando sus requisitos funcionales en forma de grafos, y sus requisitos no-funcionales en forma de parámetros temporales. Presenta una GUI amigable, que no sólo permite un diseño fácil y gráfico de la aplicación sino que también asegura que la especificación de la aplicación es conforme a la aproximación de modelado propuesta.

Por lo tanto, para poder crear una nueva aplicación, en primer lugar se deben definir sus requisitos funcionales y no funcionales. En base a esta especificación de aplicación, se genera automáticamente el esqueleto del código de los servicios que deben ser desplegados en los nodos correspondientes. A continuación, la aplicación se registra en el middleware, es decir, se registra la aplicación, sus servicios y sus implementaciones de servicio. Por último, la aplicación puede ser lanzada, momento en el que el middleware asigna todos los recursos necesarios. Los algoritmos de composición del middleware son los encargados de seleccionar el.

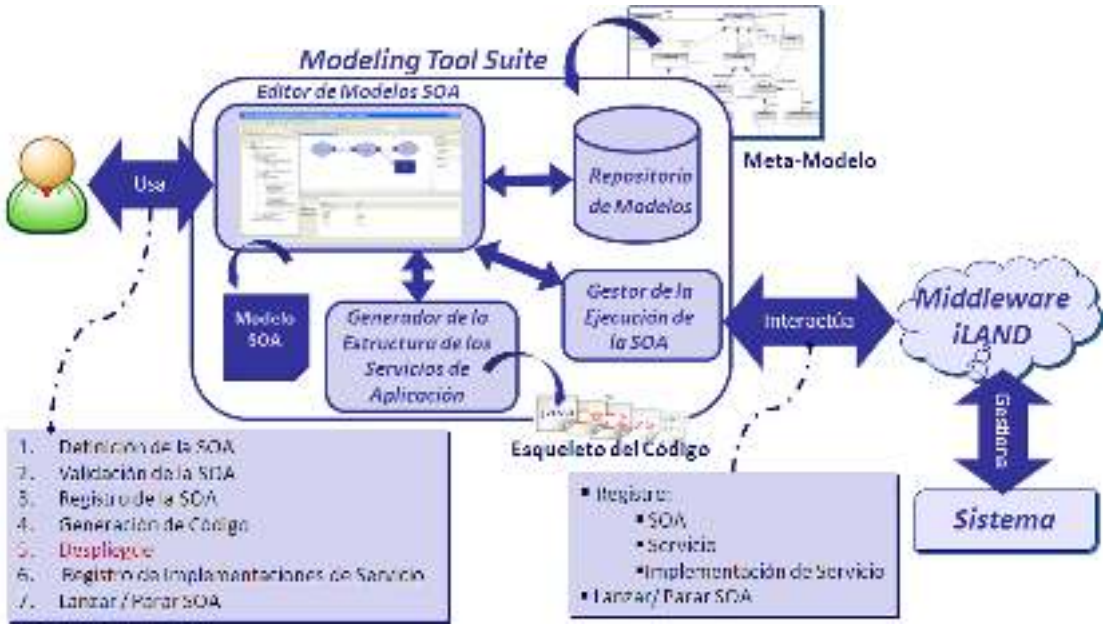


Figura 1: Escenario general de la MTS

conjunto de implementaciones de servicio que aseguran que los requisitos de la nueva aplicación, y del resto de aplicaciones en ejecución, se cumplen.

El middleware de iLAND es el encargado de gestionar las aplicaciones en tiempo de ejecución. Por lo tanto, proporciona mecanismos para el registro de aplicaciones, la forma en que se pasa información de la aplicación de la herramienta al middleware. También es capaz de seleccionar las implementaciones de servicio más adecuadas para asegurar que los requisitos no funcionales de la aplicación se cumplen. Y por último, es el encargado de controlar que estos requisitos se cumplen no sólo cuando la aplicación se arranca, sino también durante toda su ejecución, reconfigurando en caso de que se detecte que no se van a poder cumplir.

Tal y como se muestra en la Figura 1, la MTS se divide en cuatro componentes que colaboran, y que se detallan en las siguientes sub-secciones.

3.1 Editor de Modelos SOA

Este componente permite la especificación de aplicaciones. Dos son sus principales objetivos. Por un lado proporciona una GUI amigable. Por otro lado, se encarga de la generación del modelo de la SOA, que es conforme al meta-modelo de iLAND. Este componente guía al usuario durante la especificación de la SOA asegurando que únicamente se editan modelos correctos. Además, es el encargado de la interacción entre el resto de componentes, eso sí, de forma transparente al usuario.

Existen diferentes *frameworks* y *toolkits* que permiten el desarrollo de GUIs para modelado de sistemas o aplicaciones, pero muchas no están basadas en modelos en el sentido de que no se construyen a partir de un meta-modelo. Entre las que sí están basadas en modelos, se han tenido en cuenta dos aproximaciones: Generic Model Environment (GME) [10] y Graphical Modeling Framework (GMF) [12]. Aunque ambas se basan en los principios de MDE y tienen capacidades gráficas similares, se ha optado por GMF para la implementación del prototipo. A pesar de que el proceso de desarrollo es más complejo que en GME, la herramienta generada puede ser más fácilmente extendida y particularizada, ya que consiste en un conjunto de clases Java gestionadas en forma de plug-ins. Además, GMF pertenece al proyecto Eclipse Modeling Project [6] que proporciona buenas herramientas de soporte para transformaciones M2M, transformaciones M2T, etc.

Por lo tanto, el núcleo de este componente es el meta-modelo detallado en [15], pero implementado mediante el lenguaje Ecore. A determinados elementos de modelo se les asigna una representación gráfica, un icono, y una entrada en la paleta de herramientas. Tal y como se ha comentado, la herramienta básica proporcionada por GMF es extendida y particularizada para añadirle nuevas funcionalidades y validaciones. Más concretamente, con el objetivo de guiar y validar la especificación de aplicaciones se añaden nuevos menús emergentes, formularios, validaciones y acciones. La Figura 2 muestra un ejemplo de un formulario usado para introducir los parámetros no funcionales (las imágenes por segundo a procesar) correspondientes a

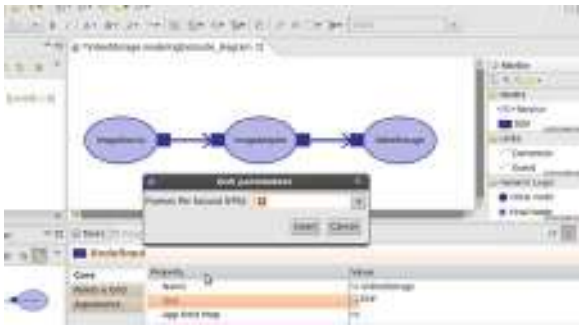


Figura 2: Formulario de la MTS que muestra las características de una función de servicio

una aplicación de video vigilancia. Como resultado de la especificación de una aplicación se generan dos ficheros, uno para el modelo de la especificación y otro para su representación gráfica.

3.2 Repositorio de Modelos

Este componente contiene los diferentes modelos utilizados en la MTS. En primer lugar almacena las caracterizaciones de las funciones de servicio necesarias. Los desarrolladores las definen almacenándolas en el repositorio para que los diseñadores de la aplicación puedan buscarlas y seleccionar la más adecuada. Los modelos de las aplicaciones generados por el "Editor de Modelos SOA" y sus correspondientes representaciones gráficas también son almacenados. Por último, también se guardan las implementaciones de servicio registradas.

Teniendo en cuenta que todos los elementos almacenados son documentos XML (eXtensible Markup Language) [8], las bases de datos XML nativas parecen ser la opción más adecuada, ya que pueden hacer uso de las capacidades específicas de XML y pueden gestionar documentos XML directamente. Además a la base de datos elegida se le exige también que proporcione mecanismos de búsqueda, trabajar con partes de un documento y que sea abierta y multiplataforma. Por todo ello, la base de datos eXist [7] ha sido seleccionada.

3.3 Generador de la Estructura de los Servicios de Aplicación

Este componente se encarga de la generación del esqueleto del código de los servicios de una aplicación, teniendo en cuenta la lógica de conexión definida así como las invocaciones a reconfiguración a nivel de aplicación. Por lo tanto, el esqueleto generado incluye la interacción entre la aplicación y el middleware, la cual se implementa mediante llamadas a su interfaz de programación (API, Application Programming Interface). El código generado junto con el código de la función de servicio se despliega en los nodos necesarios dando lugar a las implementaciones de servicio.

Los generadores de código son componentes importantes dentro del MDD ya que su objetivo es transformar en código la descripción en forma de modelos abstractos (ej. Ecore) de un sistema software. A pesar de que existe la posibilidad de crear manualmente modelos abstractos y transformarlos en código, la potencialidad que proporciona MDD viene de la automatización de este proceso. Como resultado, se mejora la calidad del código generado y se acelera su generación.

Debido al uso de GMF para el desarrollo de la GUI, para la implementación de este componente se han tenido en cuenta los *frameworks* incluidos en el Eclipse Modeling Project, resultando Acceleo [1] el más adecuado. Acceleo transforma modelos en código siguiendo la aproximación Model Driven Approach (MDA) [16]. Se basa en la definición de un conjunto de plantillas que determinan el texto a generar para cada elemento del modelo, por lo que debe seguir un determinado meta-modelo, el anteriormente propuesto para este prototipo.

3.4 Gestor de la Ejecución de la SOA

Este componente es el encargado de la interacción con el middleware de iLAND. Por un lado, se facilita el registro de las aplicaciones después de su definición y validación (lo cual implica también el registro de sus servicios), así como de las implementaciones de servicio después de su despliegue. Por otro lado, este componente proporciona al usuario final mecanismos para lanzar una aplicación registrada o parar una aplicación en ejecución. Por lo tanto, se puede decir que este componente encapsula todas las invocaciones necesarias al API del middleware de iLAND para poder registrar, arrancar y parar aplicaciones, pudiendo realizar todos estos procesos de forma automática, evitando así muchos errores.

4 CASO DE ESTUDIO: ALMACENAMIENTO DE VIDEO

Uno de los objetivos del proyecto ILAND consiste en obtener resultados lo suficientemente generales como para poder ser aplicados a distintos dominios. Por ello, se desarrollan demostradores en tres distintas áreas de aplicación: aplicaciones de video vigilancia, aplicaciones de atención de la salud y aplicaciones de detección temprana usando infraestructura pública oportunista. Este apartado verifica la efectividad de la herramienta propuesta aplicándola a un caso de uso de una simple aplicación de video-vigilancia llamada "Almacenamiento de Video". El objetivo de

esta aplicación consiste en el almacenamiento en disco de las imágenes obtenidas por una cámara, permitiendo a un operador el poder acceder a ellas en cualquier momento.

En primer lugar, la MTS permite definir los requisitos funcionales de dicha aplicación como un grafo formado por tres servicios conectados, al y como se aprecia en la Figura 2, cada uno con su correspondiente función de servicio asociada:

- *Image Source*: proporciona las imágenes capturadas por una cámara IP que son enviadas al servico *Image Adapter*.
- *Image Adapter*: realiza el pre-procesamiento de las imágenes capturadas (descompresión, ajuste del color y tamaño, etc.), que posteriormente serán enviadas al servicio *Video Storage*.
- *Video Storage*: se encarga del almacenamiento de las imágenes recibidas en un grabador de video en red (NVR, Network Video Recorder)

En esta aplicación todos los puertos de entrada y todos los puertos de salida llevan asociada su correspondiente lógica por defecto. Además, todos los parámetros de salida son siempre enviados al siguiente servicio.

En base a esta funcionalidad, se genera el esqueleto del código de los servicios. Tal y como se ha comentado anteriormente, cada servicio de este ejemplo recibe todos los parámetros de entrada necesarios para su función de servicio de un único origen; y después de la ejecución de la función de servicio, todos los parámetros de salida obtenidos son enviados al siguiente servicio. Como es el middleware el encargado de gestionar el envío y recepción de datos, el código generado encapsula la interacción con el middleware haciendo uso de su API. La Figura 3 muestra un parte del código generado para el servicio *Image Adapter*.

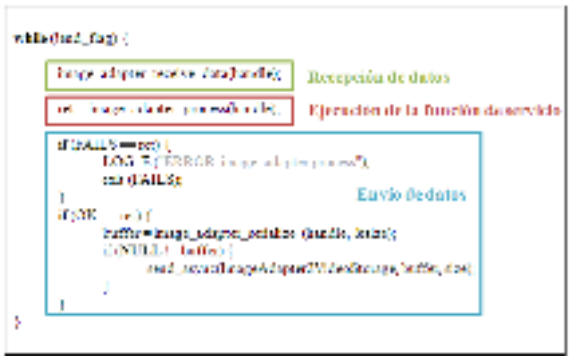


Figura 3: Parte del esqueleto del código generado para el servicio *Image Adapter*

En lo que se refiere a los requisitos no funcionales, por un lado la aplicación presenta limitaciones temporales ya que las imágenes deben almacenarse con un frecuencia mínima, los llamados frames por segundo (FPS, Frames Per Second). Para poder tener sensación de movimiento continuo al recuperar las imágenes, es necesario que se almacenen como mínimo a 12FPS. Por otro lado, debido a que la capacidad de almacenamiento de los NVR es limitada, se debe asegurar que no se pierden imágenes, cambiando un NVR por otro cuando la capacidad de almacenamiento del primero esté a punto de agotarse.

El middleware de iLAND es el encargado de asegurar que los requisitos no-funcionales deseados se cumplen. Por lo tanto, debe asignar los recursos necesarios para ello. No obstante, si la aplicación "Almacenamiento de Video" fuera la única ejecutándose en el sistema, el middleware podría decidir asignarle los máximos recursos posibles, alcanzando por ejemplo los 25FPS. Cuando el middleware detecte: (1) que los recursos del sistema están sobrecargados o (2) incluso cuando detecte que un nodo ha caído, puede decidir reconfigurar la aplicación. Como resultado, en el primer caso los recursos asignados son liberados, reservando nuevos que aseguren los 12FPS establecidos, al mismo tiempo que nuevas aplicaciones pueden ser admitidas para ejecución. En el segundo caso, el middleware puede reemplazar el nodo fallido por otro nuevo. Lo mismo ocurre cuando se detecta que la capacidad de almacenamiento de un NVR está a punto de acabarse.

Para ello, el diseñador de la aplicación, a través de la MTS, debe establecer los FPS, 12FPS, asociados a la aplicación, tal y como se muestra en la Figura 2.

Además, se deben desplegar dos implementaciones de servicio por cada servicio de la aplicación: una que permita procesamientos a 12FPS y la otra a 25FPS. El trabajar a 25FPS implica que las implementaciones de servicio deben procesar imágenes cada 40ms. Del mismo modo, el trabajar a 12FPS implica un tiempo de procesamiento de 80ms.

El registro es el mecanismo usado para pasar al middleware información acerca de las aplicaciones, sus servicios y las implementaciones de servicio. La MTS permite este registro tal y como se muestra en la Figura 4 donde se puede ver que el registro de la aplicación implica el registro de sus tres servicios.

Finalmente, una vez que la aplicación ha sido registrada, haciendo uso de la MTS se puede lanzar dicha aplicación, o incluso parar una aplicación en ejecución.

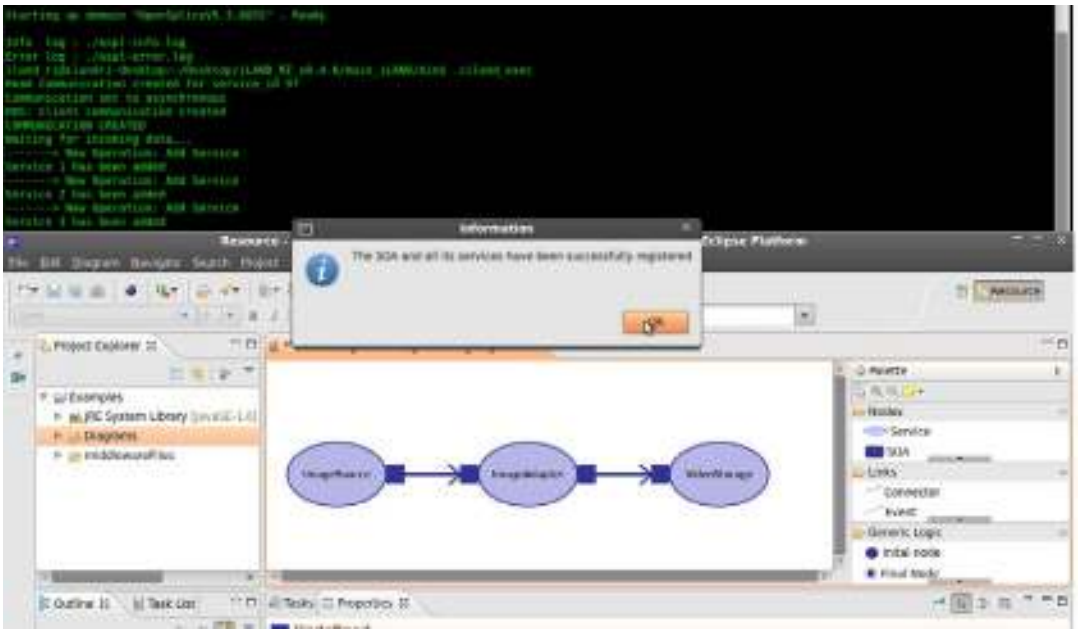


Figura 4: Registro de la aplicación "Almacenamiento de Video"

5 CONCLUSIONES

Este trabajo propone el uso de una herramienta de modelado basada en MDE para dar soporte al ciclo de desarrollo de aplicaciones dinámicamente reconfigurables orientadas a servicio. La gestión en tiempo de ejecución de dichas aplicaciones la lleva a cabo una plataforma de middleware que proporciona funcionamiento y reconfiguración acotados en el tiempo. De hecho, la herramienta de modelado propuesta abstrae al usuario final de los detalles de dicho middleware, encapsulando no sólo la interacción con él sino también la generación automática del esqueleto del código de los servicios de la aplicación. Es más, la herramienta propuesta ha sido pensada como un conjunto de herramientas que interactúan para guiar y validar el desarrollo de las aplicaciones desde su especificación hasta su ejecución.

Se han empleado técnicas de meta-modelado para la caracterización de las aplicaciones y para la generación automática de código. Por lo tanto, se ha presentado un meta-modelo que identifica los elementos del sistema y las relaciones entre ellos, y que constituye el punto de partida de la herramienta. La generación automática de código se ha llevado a cabo a través de transformaciones M2T basadas en plantillas.

También se ha presentado un prototipo de la herramienta desarrollado mediante GMF. El uso de este framework de Eclipse ha permitido el desarrollo de una herramienta, fácilmente ampliable y personalizable, formada por cuatro componentes diferentes pero complementarios.

Agradecimientos

Este trabajo ha sido financiado por el proyecto ARTEMIS iLAND (ARTEMIS-JU 100026), por UFI11/28, por GV-IT19-13, y por MCYT&FEDER bajo la subvención DPI- 2012-37806.

Referencias

[1] Acceleo Project, Eclipse (2012). Available: http://eclipse.org/acceleo

[2] Agirre, A., Marcos, M., Estévez, E., (2012) "Distributed Applications Management Platform Based on Service Component Architecture", *17th IEEE International Conference on Emerging Technologies and Factory Automation*, Cracow.

[3] Bih J. (2006) "Service Oriented Architecture (SOA). A New Paradigm to Implement Dynamic E-business Solutions". *Ubiquity*, vol. 7 (30).

[4] Brambilla, M., Cabot, J., Wimmer, M., (2012) Model-Driven Software Engineering in Practice, Morgan & Claypool, USA.

[5] Cucinotta, T., Mancina, A., Anastasi, G.F., Lipari, G., Mangeruca, L., Checcozzo, R., Rusina, F., (2009) "A Real-Time Service-Oriented Architecture for Industrial Automation", *IEEE Transaction on Industrial Informatics*, vol. 5, pp. 267-277.

[6] Eclipse Modeling Project, Eclipse (2013). Available: http://www.eclipse.org/modeling/

[7] eXist-db (2013). Available: http://exist-db.org/
[8] eXtensible Markup Language, W3C (2012). Available: http://www.w3.org/XML/
[9] García-Valls, M., Rodríguez López, I., Fernández Villar, L., (2013) "iLAND: An Enhanced Middleware for Real-Time Reconfiguration of Service Oriented Distributed Real-Time Systems", *IEEE Transaction on Industrial Informatics*, vol. 9, pp. 228-236.
[10] Generic Modeling Environment, GME (2008). Available: http://www.isis.vanderbilt.edu/Projects/gme
[11] Gottschalk K., Graham S., Kreger H., Snell J., (2002) "Introduction to Web Services architecture". *IBM Systems Journal*, vol. 41, no. 2, pp. 170–177.
[12] Graphical Modeling Framework, Eclipse (2011). Available: http://www.eclipse.org/modeling/gmp/
[13] iLAND-Artemis project (2012). Available: http://www.iland-artemis.org/
[14] López-Sanz, M., Marcos, E., (2012) "ArchiMeDeS: A model-driven framework for the specification of service-oriented architectures", *Information Systems*, vol. 37, pp. 257-268.
[15] Marcos, M., Estévez, E., Jouvray, C., Kung, A., (2011) "An Approach to use MDE in Dynamically Reconfigurable Networked Embedded SOAs". Proc. of the *18th IFAC World Congress*.
[16] Model Driven Architecture, OMG (2013). Available: http://www.omg.org/mda/
[17] Ort, E. (2005) "Service-Oriented Architecture and Web Services: Concepts, Technologies and Tools", *White Paper. Sun Microsystems*. Available: http://www.oracle.com/technetwork/articles/javase/soa2-137288.html
[18] Schmidt, D. C., (2006) "Guest Editor's Introduction: Model-Driven Engineering", *Computer*, vol. 39, pp. 25 – 31.
[19] Selic, B., (2003) "The pragmatics of model-driven development", *IEEE Software*, vol. 20 (5), pp. 19 – 25.
[20] Sun, C., Rossing, R., Sinnema, M., Bulanov, P., Aiello, M., (2010) "Modeling and managing the variability of Web service-based systems", *Journal of Systems and Software*, vol. 83, pp. 502-516.
[21] Tsai, W. T., Fan, C., Chen, Y., Paul, R., (2006) "A service-oriented modeling and simulation framework for rapid development of distributed applications", *Simulation Modelling Practice and Theory*, vol. 14, pp. 725-739.
[22] Unified Modeling Language, UML, v2.4.1, OMG (2011). Available: http://www.omg.org/spec/UML/2.4.1/
[23] Wang, G., Tao, L., Di, H., Ye, X., Shi, Y., (2012) "A Scalable Distributed Architecture for Intelligent Vision System", *IEEE Transaction on Industrial Informatics*, vol. 8, pp. 91-99.

Reconfigurando aplicaciones de automatización industrial con FTT-MA

Mikel González[1], Adrián Noguero[2], Federico Pérez[1], Isidro Calvo[3]

[1]DISA (UPV/EHU) E.T.S.I. de Bilbao, Alda Urquijo s/n 48013, Bilbao
mgonzalez187@ikasle.ehu.es; federico.perez@ehu.es
[2]TECNALIA División ICT/ESI, Parque Tecnológico de Zamudio, #202 48170, Zamudio, Bizkaia
adrian.noguero@tecnalia.com
[3]DISA (UPV/EHU) E.U.I. de Vitoria-Gasteiz, C/Nieves Cano, 12, 01006, Vitoria-Gasteiz
isidro.calvo@ehu.es

Resumen

Las aplicaciones de automatización modernas deben ser capaces de adaptarse a los cambios en los requerimientos de producción. Dado que la mayoría de las aplicaciones de fabricación son distribuidas, puede ser necesario reconfigurar las operaciones que ejecutan varios dispositivos del sistema de forma sincronizada. Desafortunadamente, reconfigurar este tipo de aplicaciones es una tarea compleja para los desarrolladores de aplicaciones, especialmente cuando las operaciones se deben ejecutar en determinados instantes de tiempo o los recursos son escasos. En este escenario, las arquitecturas middleware de alto nivel pueden proporcionar patrones de diseño que proporcionen soluciones a esta problemática. Este trabajo ilustra, a través de una aplicación de laboratorio, cómo utilizar la arquitectura FTT-MA (Flexible Time-Triggered Middleware Architecture) para reconfigurar aplicaciones de automatización industrial.

Palabras Clave: Reconfiguración de sistemas, Automatización, Middleware, FTT-MA.

1 INTRODUCCIÓN

Los sistemas de fabricación modernos deben ser capaces de responder adecuadamente a los cambios repentinos en la demanda, tanto con respecto al número de productos involucrados como en sus características y tipo [3]. Tal y como se recoge en [1] los sistemas de fabricación reconfigurables deben reaccionar rápidamente y eficientemente a las demandas de un mercado dinámico, a través de un diseño modular y escalable del sistema de fabricación a distintos niveles: a nivel de sistema y a nivel de máquina.

La mayoría de las aplicaciones de fabricación actuales son distribuidas es decir, existe un elevado número de dispositivos heterogéneos involucrados. Desafortunadamente, su construcción no resulta una tarea sencilla para los desarrolladores, dado que deben proveer soluciones a problemas comunes como: (1) priorizar diferentes flujos de tráfico de comunicación de red (relacionados con el procesado de datos periódicos, alarmas, datos no-críticos, etc.); (2) sincronizar acciones entre distintos dispositivos distribuidos; (3) hacer frente a la heterogeneidad, es decir, integrando diferentes plataformas hardware, SO, etc.; y, (4) facilitar la gestión de recursos, por ejemplo, detección de fallos, manejo de las sobrecargas, etc. Afortunadamente, están emergiendo nuevas arquitecturas middleware de alto nivel que van más allá de la mera distribución de datos a través de los dispositivos involucrados. Estas arquitecturas incluyen abstracciones de alto nivel que representan los dispositivos, recursos e información y proporcionan servicios comunes que facilitan la construcción de las aplicaciones distribuidas.

Este trabajo ilustra el uso de la arquitectura *Flexible Time-Triggered Middleware Architecture* (FTT-MA) [7] en aplicaciones de automatización industriales a través de una aplicación de laboratorio. FTT-MA es una arquitectura middleware dirigida a sistemas que requieren: (1) la ejecución de sus actividades en unos instantes de tiempo concretos; y (2) la reconfiguración de la aplicación en tiempo de ejecución de acuerdo a cambios en la composición del sistema (por ejemplo, conectando/desconectando dispositivos o servicios). Es importante observar que FTT-MA es una arquitectura middleware *time-triggered* que permite a los desarrolladores centrarse en la funcionalidad de las aplicaciones con independencia de las cuestiones de gestión de los recursos y la sincronización de las actividades, ya que la arquitectura de middleware es capaz de hacer frente a estas cuestiones de forma integrada.

El diseño de este artículo es el siguiente: la Sección 2 presenta una breve descripción de la arquitectura

FTT-MA, la Sección 3 propone un componente que encapsula los programas que se ejecutan en dispositivos de automatización como PLCs; la Sección 4 presenta un caso de estudio orientado a ilustrar cómo aplicar FTT-MA en aplicaciones de fabricación; por último, en la sección 5 se extraen algunas conclusiones.

2 BREVE DESCRIPCIÓN DE FTT-MA

Las arquitecturas *event-triggered* abordan problemas de tiempo relativos y son intrínsecamente flexibles. Por el contrario, las arquitecturas *time-triggered* se basan en un reloj global sincronizado, lo cual reduce la flexibilidad de las aplicaciones. A pesar de décadas de discusión entre estos dos paradigmas, los autores creen, como en [5], que la sincronización de tiempos en un sistema proporciona alto rendimiento, seguridad y facilidad de desarrollo de los sistemas distribuidos. Sin embargo, la flexibilidad no es una característica menos relevante, especialmente en aplicaciones de automatización industrial. FTT-MA es una arquitectura time-triggered basada en el paradigma FTT [2], que es capaz de mantener los beneficios de las arquitecturas time-triggered y, al mismo tiempo, proporcionar cierto grado de flexibilidad [6, 7].

FTT-MA está diseñado para aplicarse aplicaciones que son fundamentalmente periódicas, a pesar de que puntualmente también se pueden incluir actividades no periódicas. FTT-MA: (1) asegura la sincronización de tiempo de las operaciones de las aplicaciones distribuidas; (2) permite la coexistencia de diferentes aplicaciones distribuidas sobre la misma infraestructura (equipos hardware y red); (3) gestiona las comunicaciones mediante el uso de diferentes niveles de prioridad y (4) proporciona flexibilidad para llevar a cabo los cambios de las aplicaciones en tiempo de ejecución.

FTT-MA ha sido diseñado como una arquitectura en capas en la que un conjunto de servicios colaboran entre ellos. Algunos de estos servicios están centralizados, en el llamado *Orchestrator*, mientras que otros están distribuidos en cada nodo participante, que normalmente son ordenadores embebidos que ejecutan la funcionalidad del sistema distribuido encapsulada como tareas. Un componente de la arquitectura, el llamado *Clerk*, centraliza los servicios comunes de la arquitectura en cada nodo distribuido.

En FTT-MA el tiempo se representa en términos de un intervalo de tiempo de duración fija llamado Ciclo Elemental (*Elementary Cycle*, EC), que determina la granularidad del sistema distribuido. Las aplicaciones en FTT-MA se componen de una secuencia ordenada de tareas, descritas por gráficos dirigidos. Cada tarea se representa con un conjunto de parámetros relacionados con propiedades temporales como son el *deadline* y el *offset*. Estas tareas se encapsulan como métodos que son invocados por la arquitectura.

La arquitectura FTT-MA es una arquitectura abstracta que puede ser implementada sobre diferentes tecnologías de comunicación. Actualmente, existe una implementación de FTT-MA sobre CORBA, llamada FTT-CORBA, que está disponible como código abierto en [7]. En este caso, las tareas son encapsuladas como métodos CORBA. Sin embargo, los mismos principios se pueden adaptar fácilmente sin pérdida de generalidad a otras tecnologías de distribución, tales como ICE o DDS.

3 ENCAPSULANDO PROGRAMAS DE PLC COMO TAREAS FTT-MA

Las aplicaciones en FTT-MA se construyen a partir de una secuencia de tareas. En el caso de las aplicaciones de automatización industrial, es frecuente que estas tareas se implementen como programas ejecutados en dispositivos de control industrial, normalmente PLCs, de acuerdo al estándar IEC 61131. En tal caso es necesario encapsular estos programas para que puedan ser usados por FTT-MA. Esta sección presenta brevemente cómo encapsular componentes de automatización como tareas FTT-MA (Fig. 1).

Para ello se propone utilizar el llamado *FTT-MA PLC Task*. Este componente está compuesto por un conjunto de elementos que permiten intercambiar información con un dispositivo/recurso de automatización, p.e. un programa de PLC. También permite cubrir tanto el control interno del dispositivo como su integración en FTT-MA. Por lo tanto, el encapsulado accede al programa de control por medio de los Bloques de Función (FB) específicos, los llamados *Middleware Interface Blocks* (MIB), que ocultan mecanismos de intercambio de información propietarios. Estos MIBs también proporcionan mecanismos de comunicación entre el control del programa y el *wrapper* FTT-MA, proporcionando una capa de encapsulación.

El presente trabajo utiliza FTT-CORBA [7], que es la implementación actualmente disponible de FTT-MA. Por tanto, se utiliza la tecnología CORBA para encapsular las tareas que se ejecutan en la aplicación propuesta en sección 4. Las tareas que se ejecutarán como parte de las aplicaciones FTT-MA se encapsularán como métodos CORBA que serán invocadas en cada dispositivo por el *Clerk* local de acuerdo con el plan de ejecución diseñado en el *Orchestrator*.

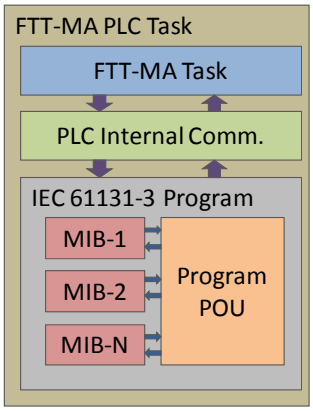


Figura 1: Arquitectura de FTT-MA PLC Task.

4 CASO DE ESTUDIO

En esta sección se describe una aplicación de laboratorio destinada a ilustrar la aplicación del FTT-MA en aplicaciones de automatización industrial. El caso de estudio propuesto consiste en la automatización de una célula de ensamblado (ver fig. 2) en la cual participan cinco estaciones. Cada estación realiza varias operaciones y como resultado final se obtiene la pieza ensamblada, compuesta por una base, un rodamiento, un eje y una tapa, y se almacena en un almacén. Para transportar las piezas de una estación a otra en el proceso de ensamblado se utilizan unos pallets que se recuperan después de almacenar las piezas.

La célula es capaz de ensamblar diferentes tipos de piezas, que varían en el tamaño y material de los componentes básicos (base, rodamiento, eje y tapa). Las estaciones individuales son capaces de distinguir el tipo de pieza a montar por medio de un código de la pieza, situado en la parte inferior del pallet, y ejecutan diferentes operaciones sobre la pieza a ensamblar de acuerdo al código del pallet. Cada estación tiene una cinta transportadora para mover el pallet a través de la célula. Después de realizar la operación de ensamblado de una pieza, la cinta transportadora mueve el pallet con la pieza a la siguiente estación, con el fin de continuar con el proceso de montaje. Las cinco estaciones implicadas son:

1. **Cargador:** Responsable de poner la base sobre el pallet situado en la cinta transportadora.

2. **Robot:** Esta estación obtiene el rodamiento y el eje de los puntos de alimentación correspondientes y los ajusta en la base colocada sobre el pallet.

3. **Sellador:** Esta estación coloca la tapa sobre la parte montada, previamente fijada por el robot.

4. **Almacén:** Esta estación recoge la pieza montada de la plataforma y la deja en el área de almacenamiento.

5. **Estación de selector:** Esta estación recupera los pallets vacíos, después de dejar la pieza montada en el almacén, y los reorganiza para procesar nuevas piezas según el código de pieza situada en la parte inferior del pallet, atendiendo los requisitos del supervisor con respecto a las piezas que se deben ensamblar.

Las estaciones y cintas transportadoras se rigen por dos controladores que automatizan todo el proceso:

1. **Controlador de proceso:** Responsable de controlar simultáneamente todos los procesos locales, es decir, los movimientos del equipo en cada estación, con la excepción de la estación de selector.

2. **Controlador de transporte:** Este controlador gestiona todas las operaciones de transporte a lo largo de la célula de fabricación automatizada. Gobierna simultáneamente el movimiento de los pallets en cada estación, así como la selección del tipo de pieza a ensamblar por la elección del pallet de acuerdo al código de pieza situado en su parte inferior.

Además, hay cuatro cámaras (una por estación, excepto en el caso del sistema selector) que permiten monitorizar los procesos de automatización. En este caso de estudio, se pretende que cada cámara sea capaz de monitorizar y recoger una señal de vídeo de 20 fotogramas por segundo. A modo de ejemplo, considerando que cada fotograma tiene un tamaño aproximado de 10 kBytes, y que la célula de fabricación estuviese comunicada con una red de 100 Mbps de ancho de banda (p.e. Ethernet), se dispondrá de tiempo suficiente para el envío de los fotogramas, ya que el envío del vídeo por cada cámara requerirá 16 milisegundos/segundo.

La aplicación propuesta requiere el uso de otro nodo, el Supervisor, que gestiona la producción de la célula, monitoriza el estado de las estaciones, se encarga de las alarmas y permite la visualización de los fotogramas de vídeo generados por las cámaras en tiempo de ejecución.

Por último, FTT-MA requiere el uso de un nodo de coordinación donde residen todos los servicios centralizados de la arquitectura, es el llamado Orchestrator. Este componente activa la secuencia de tareas de la aplicación, *FTT-Tasks*, que el sistema automatizado debe ejecutar en cada dispositivo de acuerdo con un plan de ejecución específico (el grafo dirigido que representa la funcionalidad), evitando que los diferentes tipos de tráfico interfieran. El Orchestrator también se encarga de la ejecutar la reconfiguración del sistema cuando el Supervisor lo solicite, por ejemplo, al cambiar el tipo o el número de partes a ensamblar, la gestión de la calidad de servicio de vídeo, o para atender las alarmas generadas en el proceso de una manera segura y determinista.

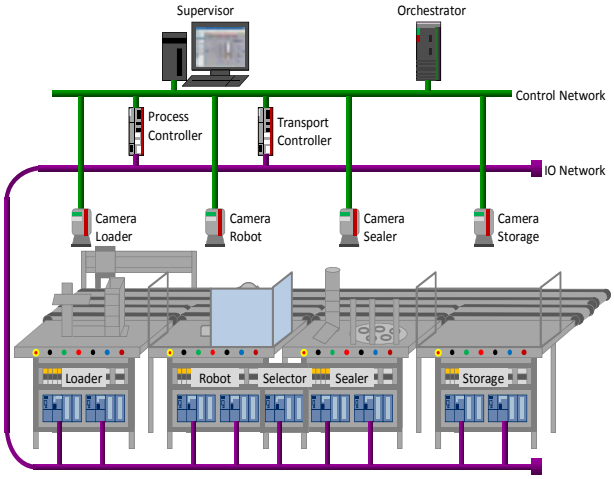


Figura 2: Célula de fabricación automática.

4.1 TOPOLOGÍA DE RED

La Figura 3 muestra la topología de red. Dos redes de comunicación están implicadas, una para acceder a las entradas y salidas de red (*IO Network*) y otra para la conexión de los dispositivos con capacidades de procesamiento (*Control Network*).

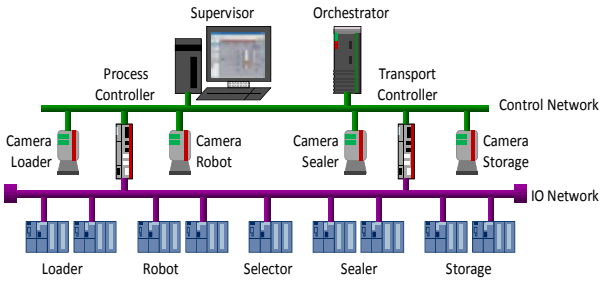


Figura 3: Distribución de los dispositivos de red.

La primera red es una red Profibus-DP, mientras que la segunda utiliza Industrial Ethernet. Tanto el controlador de procesos y el controlador de transporte se comportan como Profibus-DP clase 1 Masters. Hay dos esclavos Profibus-DP por estación, con la excepción de la estación de selector. Uno de estos esclavos se utiliza para acceder a las variables de proceso de la estación y el otro se utiliza para el control de la cinta transportadora de la estación. En la estación de selector sólo hay un esclavo Profibus-DP, el cual es el responsable de controlar tanto la estación como su cinta transportadora.

El Controlador de Proceso gobierna los esclavos que gestionan señales de control de las estaciones, mientras que el Controlador de Transporte es responsable de los esclavos que gestionan las cintas transportadoras, así como de la estación del Selector.

4.2 FTT-MA APLICACIONES Y TAREAS

El sistema de automatización propuesto se divide en varias aplicaciones FTT que se ejecutan concurrentemente sobre la misma infraestructura. A pesar de que estas tareas comparten recursos (red y CPUs de los dispositivos), tienen diferentes parámetros de configuración, tales como plazos, períodos y offsets. La Figura 4 representa estas aplicaciones, incluyendo las tareas que las componen, los mensajes intercambiados y sus parámetros principales.

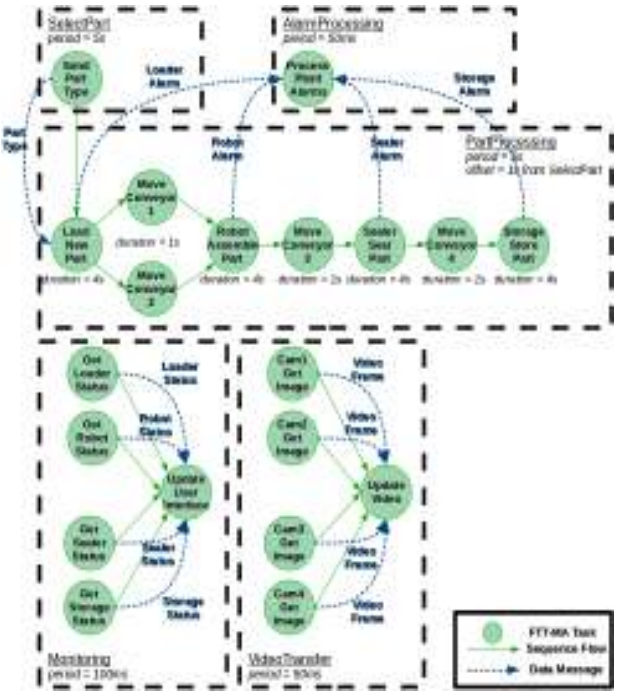


Figura 4: Relación de Tareas.

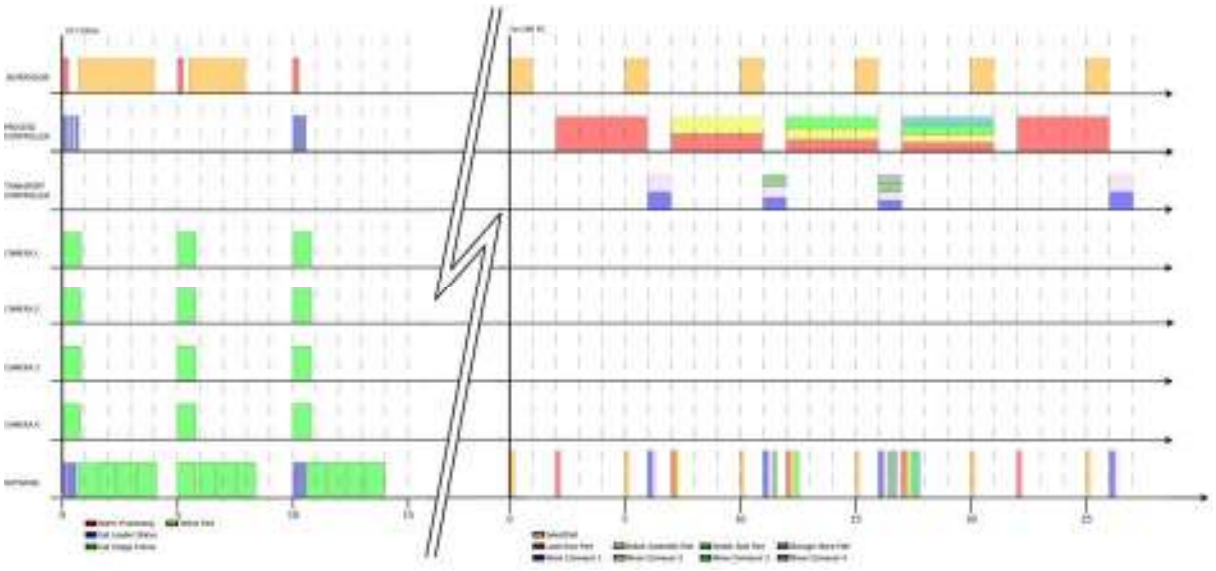


Figura 5: Cronograma de tiempos de las tareas.

Como se aprecia en la Figura 4, todo el sistema se compone de cinco aplicaciones FTT-MA: *SelectPart, AlarmProcessing, Monitoring, VideoTransfer and PartProcessing*. Cada aplicación requiere la invocación de diferentes tareas FTT-MA con diferentes períodos, prioridades y offsets. Por ejemplo, las aplicaciones *SelectPart* y *PartProcessing* se activan periódicamente, cada 5s. Su objetivo es activar síncronamente la ejecución de los procesos adecuados en las diferentes estaciones, así como el manejo del movimiento de las cintas transportadoras. Es importante tener en cuenta que, para que un pallet pueda pasar de una estación a la siguiente, ambos transportadores necesitan estar activos. Por otra parte, la aplicación periódica *PartProcessing* se ha establecido como el proceso de fabricación de duración más larga en cualquier estación (es decir, el Robot), más la duración del proceso de movimiento de parte en los transportadores. Este período es mucho menor que la duración de todo el proceso de ensamblado. Sin embargo, la repetición de la aplicación permite el establecimiento de una activación secuencial de todas las estaciones en la célula de fabricación. Esto significa que durante los primeros 5s, sólo la primera estación estará activa. Tras otros 5s, cuando el robot comience a procesar la parte # 1, el cargador comenzará a procesar la parte # 2, y así sucesivamente.

La aplicación *SelectPart* se activa justo antes de la aplicación de fabricación para seleccionar el tipo de pallet que va a sostener la pieza, y por lo tanto configurar el procesamiento de la pieza en la célula. Esta información se envía mediante un mensaje específico.

La Figura 5 presenta un cronograma en el cual se observa la gestión de los tiempos para las diversas tareas mostradas en la Figura 4, así como la utilización de la red. Los dispositivos involucrados son el *Supervisor, el Controlador de Proceso, el Controlador de Transporte* y las cámaras de las distintas estaciones, que en este caso en concreto, son cuatro. Además, en dicha figura, se muestran dos gráficas diferentes, ya que la escala de tiempos para algunas tareas es de milisegundos (tareas correspondientes a las aplicaciones de *Monitoring, VideoTransfer, AlarmProcessing y SelectPart*), y en cambio para otras tareas la escala de tiempo es de segundos (tareas correspondientes a *PartProcessing*).

Las tareas que ejecuten el proceso de fabricación pueden activar alarmas en caso de un mal funcionamiento. Si esto ocurre, los mensajes de alarma se enviarán al canal de eventos de FTT-MA para ser consumidos por el Supervisor, cuando la aplicación *AlarmProcessing* esté activa. Con las aplicaciones *Monitoring y VideoTransfer* se sigue un enfoque similar. En este caso se obtiene de forma periódica información de las estaciones y cámaras, respectivamente, orientada a obtener el estado de las estaciones e imágenes de video. Esta información será también consumida por el supervisor y se mostrará en la interfaz de usuario.

FTT-MA gestiona las comunicaciones de datos por medio del llamado Canal de Eventos FTT. A cada tipo de tráfico se han asignado diferentes niveles de prioridad para garantizar que los mensajes críticos alcanzan la tarea consumidora a tiempo. De esta manera, los mensajes de alarma se han configurado con prioridad *muy alta*, los mensajes periódicos con

prioridad alta. Por el contrario, los mensajes informativos se han configurado con niveles de prioridad más baja. Concretamente, los mensajes de estado se han configurado con prioridad *baja* y los mensajes de vídeo con prioridad *muy baja*.

Es importante señalar que las reconfiguraciones serán dirigidas al Orchestrator mediante la interfaz de Administración que proporciona la arquitectura FTT-MA. Normalmente estas solicitudes se realizarán desde el Supervisor. El Orchestrator aplica test de admisión para comprobar que la nueva configuración es válida, y hará cumplir las modificaciones coherentemente.

5 CONCLUSIONES

Este trabajo describe cómo se puede usar FTT-MA en aplicaciones de automatización industrial con necesidades de reconfiguración por medio de un proceso de ensamblado que se ejecuta en una célula automatizada de laboratorio. Dado que se usa implementación del FTT-MA para CORBA denominada FTT-CORBA, también se proporciona el esqueleto de un *wrapper* que permite encapsular programas de PLC como objetos CORBA. En el documento se describe una aplicación de laboratorio capaz de ensamblar diferentes tipos de piezas. Están implicados diferentes tipos de tráfico con diferentes niveles de prioridad tales como procesamiento de datos, alarmas y datos no críticos (por ejemplo fotogramas de vídeo). FTT-MA permite la gestión de los recursos del sistema distribuido mientras que proporciona la capacidad de realizar las reconfiguraciones de las aplicaciones en tiempo de ejecución de una forma determinista. Más específicamente, FTT-MA (1) asegura la sincronización de tiempo de las operaciones; (2) permite la coexistencia de diferentes aplicaciones distribuidas sobre la misma infraestructura (equipos y de la red); (3) gestiona las comunicaciones mediante el uso de diferentes niveles de prioridad y (4) proporciona flexibilidad para llevar a cabo los cambios de las aplicaciones en tiempo de ejecución.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Economía y Competitividad español (MINECO) bajo el proyecto DPI2012-37806-C02-01 y por la Universidad del País Vasco (UPV/EHU) a través de la subvención EHU11/35.

Referencias

[1] Ahmed, M.D., ElMaraghy, W. “Investigating optimal capacity scalability scheduling in reconfigurable manufacturing systems”. *Int. J. Adv. Manuf. Technol,* 52(1), pp. 174-193, 2005.

[2] Almeida, L., Pedreiras, P., Fonseca, J.A.G. “The FTT-CAN Protocol: Why and how”, *IEEE Trans. Industrial Electron.*, Vol. 49, 1189-1201, 2002.

[3] Ateekh-Ur-Rehman, Subash Babu, A. “Reconfigurations of manufacturing systems – an empirical study on concepts, research, and applications”, *Int. J. Adv. Manuf. Technol,* 66, pp. 107-124, 2013.

[4] FTT-CORBA project webpage. Available online: http://sourceforge.net/projects/fttcorba/ (accessed on 30

[5] Lee, E.A., “Time for High-Confidence Distributed Embedded Systems”, Invited keynote talk, IEEE Real-Time Systems Symposium (RTSS), San Juan, Puerto Rico, 4 December 2012. Available online: http://chess.eecs.berkeley.edu/pubs/954.html (accessed on 30 May 2013).

[6] Noguero, A., Calvo, I. “A Time-Triggered Data Distribution Service for FTT-CORBA”, In Proc. of the IEEE ETFA2012 Conf, Krakow, Poland, 17-19 September, 2012.

[7] Noguero, A., Calvo, I., Perez, F., Almeida, L. “FTT-MA: A Flexible Time-Triggered Middleware Architecture for Time Sensitive, Resource-Aware AmI Systems”, *Sensors,* Vol.13, pp. 6229-6253, 2013.

Implementación de un sistema Hardware-In-the-Loop para la simulación en tiempo real de pilotos automáticos para UAVs

J. Velasco, S. García-Nieto, G. Reynoso-Meza, J. Sanchis.

[sergarro, jsanchis]@isa.upv.es, [jvelascocar, gilreyme]@upv.es

Instituto Universitario de Automática e Informática Industrial
Universitat Politècnica de València

Resumen

Los vehículos aéreos no tripulados (UAVs) son sistemas generalmente diseñados para la consecución de misiones de vuelo de alto riesgo. Precisamente este hecho ha impulsado un fuerte desarrollo de dichos sistemas en el campo de la aviación militar, contexto en el que los objetivos de coste quedan desdibujados a razón de otros a los que se confiere mayor importancia. Existen sin embargo ciertas aplicaciones para las que una aeronave sería de gran utilidad pero, actualmente, su empleo en ellas no es rentable. Es precisamente aquí donde los vehículos aéreos no tripulados, a pequeña escala y de bajo coste, pueden encontrar su sitio, constituyendo una mejora significativa en un amplio abanico de proyectos y empresas. En este ámbito, la técnica de simulación aumentada Hardware In the Loop (HIL), se presenta como una herramienta de gran utilidad para abaratar costes en las fases de desarrollo, limitando el número de ensayos experimentales a realizar, y disminuyendo al mismo tiempo los riesgos de pérdida de material que estos suponen. Un sistema HIL permite validar no solo el diseño de algoritmos de control, sino también las unidades en las que estos serán implementados, junto con las interfaces hardware entre dispositivos.

Palabras Clave: UAV, HIL, Simulink, Real-Time Windows target.

1 Introducción

Los vehículos aéreos no tripulados (UAVs) son sistemas principalmente desarrollados para la aviación militar. En este campo, los pilotos de aeronaves se enfrentan a misiones de confrontación directa, reconocimiento geográfico en zona enemiga, y búsqueda de objetivos estratégicos entre otras. Todas esas misiones tienen como común denominador el riesgo de bajas humanas que suponen. Es por ello que el reemplazo de la tripulación en la medida de lo posible para este tipo de tareas, justifica sobradamente una fuerte inversión en el desarrollo de los UAVs [1].

Existe sin embargo una vertiente cada vez más popular de aplicaciones que, si bien no representan una motivación tan importante como lo es la preservación de la vida humana, también justifican su desarrollo desde un ángulo totalmente diferente. Tareas como fotografiado para control de litoral costero y erosión de playas, detección y control de incendios [2], inspección de infraestructuras o realización de mediciones para la agricultura [3] son sólo algunas de las aplicaciones posibles. En este nuevo campo para la aeronáutica, un coste lo suficientemente bajo que haga rentable su utilización a las compañías es el objetivo principal.

Encontramos varios frentes que deben ser atendidos para la consecución de este objetivo. Por un lado, es necesario reducir el coste de la aeronave en sí misma. Surge así una nueva generación de aviones a pequeña escala, cuyo tamaño es el mínimo necesario para albergar los equipos de propulsión, sensorización y control de la aeronave.

En segundo lugar, los equipos integrados (sensores, actuadores y unidades de control), deben ser suficientemente potentes para controlar la rápida dinámica de estos aviones y llevar a cabo, de forma paralela, la misión para la que han sido adquiridos. El coste de dichos equipos es cada día menor gracias a la rápida evolución de la tecnología informática que se viene observando desde hace ya varios años.

Finalmente, debido a las características del producto, el coste de la fase de desarrollo se transforma en un alto porcentaje del precio final. Se requiere en este punto, minimizar el número de horas de vuelo real haciéndose uso de herramientas de simulación. Además es en esta fase en la que la integridad de los equipos corre mayor peligro. Asegurar el correcto funcionamiento de los controladores diseñados, no solo en las máquinas en las que se realizan las simulaciones, sino en el los equipos que más tarde incorporarán esos algoritmos de control, puede ahorrar muchas horas de desarrollo y recursos en el proyecto.

Precisamente las estrategias Hardware in the Loop (HIL) permiten reducir el salto de pasar de una implementación en simulación a una implementación real. Constituyen así un avance sobre los sistemas de simulación clásicos, evitando o disminuyendo los riesgos para los quipos en la fase de desarrollo.

Se trata de insertar el control diseñado, no ya en el entorno de simulación, sino directamente en la unidad hardware de control. Esta se conectará a un ordenador en el que, un software capaz de realizar simulaciones en tiempo real, hará las veces de proceso. Dicho simulador tomará las señales provenientes de la unidad de regulación tal y como le llegarán a los actuadores en el proceso real, y devolverá las señales de salida, presentándoselas al controlador de igual forma que lo harán más adelante los sensores en la planta. Este mismo tipo de técnicas han sido aplicadas en [4], [5] y [6], con resultados satisfactorios.

Como puede deducirse, esta forma de proceder ofrece la gran ventaja de validar al mismo tiempo los algoritmos diseñados, la unidad de control escogida, y las interfaces hardware entre sistemas. Los resultados de las simulaciones HIL se acercarán a la realidad tanto como lo hagan los modelos de representación del proceso a regular. Por supuesto, ese paso de modelo a realidad seguirá estando presente, sin embargo, se consigue así acotar al mínimo las fuentes de error [6].

Este artículo presenta el desarrollo y la implementación de un sistema HIL para la validación de algoritmos de control avanzados, diseñados con el fin de controlar una aeronave en todas sus fases de vuelo. Se enmarca en un proyecto de investigación para la experimentación de controladores predictivos y controladores fuzzy, que empleen a su vez herramientas de optimización multiobjetivo[1]. Todo ello sobre una plataforma que representa un reto por características como: fuerte no linealidad, dinámica rápida, gran variabilidad del entorno (perturbaciones) y dificultad de obtener mediciones precisas (deriva de sensores inerciales). Dentro de dicho marco, la disposición de un método HIL para validar el conjunto de estrategias de control se presenta como una herramienta de gran valor [7].

El artículo está dividido en 5 partes. En el apartado 2 hablaremos de la plataforma de vuelo, así como de los componentes hardware y software que conforman el conjunto HIL. Se expondrán en 3 las expresiones matemáticas necesarias para implantar un modelo del proceso, que será el que simule el comportamiento de la aeronave ante las acciones propuestas por la unidad de control. Una vez obtenido el modelo, este debe ser integrado en una plataforma de simulación de tiempo real, que disponga de las herramientas para suplantar, de manera transparente a la unidad de control, las comunicaciones entre ésta y el proceso. La herramienta utilizada a tal efecto será presentada y explicada en el punto 4. Se muestra en el punto 5 uno de los ensayos realizados tras el montaje de la plataforma. Y, finalmente, se exponen en el apartado 6 las conclusiones extraídas, además de introducirse brevemente una serie de posibles actuaciones futuras.

2 Descripción de la plataforma

Como componente principal de la plataforma de vuelo encontramos al avión Kadett 2400 fabricado por la empresa Graupner (ver figura 1).



Figura 1: Kadett 2400 (Graupner) [8]

En su interior alberga todos los dispositivos necesarios para su control, tanto en modo manual como en autónomo. Incorpora 7 servomotores para controlar las 4 superficies de control de que dispone: timón de cola, elevadores, alerones y flaps. Como unidad de propulsión integra un motor brushless de corriente continua, alimentado a través de un variador. Tanto los servomotores como el variador del motor son controlados a través de señales PWM.

Existe un dispositivo puente entre la actuación manual y la autónoma. El SSC (Servo Switch Controller), es capaz de realizar la conmutación entre las diferentes fuentes de entrada, además de encargarse de transformar paquetes binarios provenientes de un puerto serie a señales PWM.

Las acciones de control son enviadas vía puerto serie desde la FCS (Flight Control Station), constituida por un PC-104. Es esta unidad la que albergará los algoritmos de control en su interior, y se encargará por tanto, de todas las tareas necesarias en cada fase de vuelo. El bucle lo cierra la unidad IG500N. Esta unidad aúna los esfuerzos de diferentes sensores como acelerómetros, giróscopos y magnetómetros, para, gracias al filtro de Kalman ubicado en su interior, ofrecer medidas precisas de posición, orientación, velocidad lineal y angular, y aceleración,

[1] Blog proyecto. http://uavupv.blogs.upv.es/

en los 3 ejes de la aeronave. Todo ello a través del puerto serie que constituye su interfaz de comunicaciones. Esta misma plataforma fue presentada y en [9] junto con el resultado de los primeros ensayos de vuelo.

La figura 2 muestra el esquema de la plataforma explicada, con todos lo componentes mencionados. En ella se hace una distinción entre la FCS y el resto de dispositivos, (delimitados por un rectángulo y denotados con la palabra UAV). Estos últimos serán sustituidos en el sistema HIL por el computador que corre la simulación.

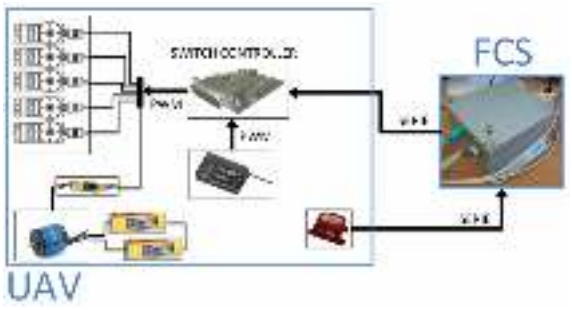


Figura 2: Esquema integral plataforma de vuelo.

Precisamente la Figura 3 muestra el esquema que finalmente adoptará el HIL. El ordenador incorpora dos puertos serie RS-232 a través de los que envía y recibe las medidas de la unidad IG500N y las acciones de control del PC-104 respectivamente. En el modelo del avión se han programado las subrutinas encargadas de transformar los valores del proceso en tramas de datos binarias, tal y como especifican los protocolos de comunicación del SSC e IG500N. Dichos protocolos fueron suministrados por los fabricantes de ambos productos.



Figura 3: Esquema HIL.

3 Formulación matemática del modelo

Toda metodología de simulación está basada en la utilización modelos, que representan de forma aproximada el comportamiento del proceso real. En este caso, se desea obtener un modelo para la simulación del control de una aeronave. Se trata pues de obtener las expresiones que relacionan las variables de entrada: variación de los ángulos en las superficies de control y revoluciones por minuto del motor; respecto de una serie de variables de salida: velocidades lineales y angulares, aceleraciones y posición en un espacio 3D.

En la búsqueda de esas expresiones partimos, tal y como se expresa en [10], de los principios de conservación de la cantidad de movimiento y del momento cinético, que se pueden expresar como:

$$\sum_{ext} F = \frac{d}{dt}(mV) \quad (1)$$

$$\sum_{ext} M = \frac{d}{dt}(I\omega) \quad (2)$$

Donde $\sum_{ext} F$ y $\sum_{ext} M$ son el sumatorio de fuerzas y momentos externos respectivamente, m e I son la masa y tensor de inercias del avión, y V y ω son los vectores de velocidades lineales y angulares. En particular son 3 los tipos de fuerzas externas que afectan al comportamiento del avión. Estos son: las fuerzas aerodinámicas (F_A), la fuerza aplicada por el motor (F_T) y la fuerza gravitatoria (F_G). A su vez, únicamente las fuerzas aerodinámicas generan pares aerodinámicos (M_A) . Así, las ecuaciones 1 y 2 quedan:

$$F_A + F_T + F_G = m\dot{V} + \omega \times mV \quad (3)$$

$$M_A = I\dot{\omega} + \omega \times I\omega \quad (4)$$

Las dos ecuaciones anteriores son en realidad ecuaciones vectoriales, por lo que se tienen un total de 6 ecuaciones que corresponden a los 6 grados de libertad de un sólido rígido en el espacio. Desarrollando 3 y 4 se llega a las siguientes expresiones:

$$\bar{q}S\begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} + \begin{bmatrix} -g\sin\theta \\ g\sin\phi\cos\theta \\ g\cos\phi\cos\theta \end{bmatrix} + \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} = m\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times m \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (5)$$

$$\bar{q}S\begin{bmatrix} bC_l \\ \bar{c}C_m \\ bC_n \end{bmatrix} = \begin{bmatrix} -I_x & 0 & -I_{xz} \\ 0 & -I_y & 0 \\ -I_{zx} & 0 & -I_z \end{bmatrix}\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} -I_x & 0 & -I_{xz} \\ 0 & -I_y & 0 \\ -I_{zx} & 0 & -I_z \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (6)$$

Donde *u, v,* y *w* son las componentes de la velocidad lineal del avión en su ejes (x,y,z). Del mismo modo, *p, q,* y *r* son las 3 componentes de la velocidad angular. Es importante destacar la aparición en las ecuaciones 5 y 6 de las variables C_i, que representan las derivadas de estabilidad para cada una de las componentes de fuerza y par aerodinámicos. Dichas

derivadas, también conocidas como coeficientes aerodinámicos, son funciones que relacionan esas componentes con el resto de variables del sistema. Suelen expresarse como polinomios de grado 1 o 2 provenientes del desarrollo en series de Taylor [10]. Los coeficientes de los polinomios se obtienen bien a través de Software CFD (Computational Fluid Dynamics) o de datos experimentales provenientes de ensayos de vuelo real o en túnel de viento. En la ecuación 7 se muestra un ejemplo de la aproximación de las derivadas de estabilidad a un polinomio de grado 1. Finalmente S, b, c son constantes constructivas del avión y $\bar{q}$ es la presión dinámica del aire.

$$C_i(x_i, \delta_i) = C_{i0} + C_{ix_1}x_1 + C_{ix_2}x_2 + \cdots + C_{ix_n}x_n + C_{i\delta_1}\delta_1 + \cdots + C_{i\delta_h}\delta_h \quad (7)$$

Como se mencionaba en el párrafo anterior, la ecuación 7 muestra una aproximación de la dependencia de los coeficientes aerodinámicos con el resto de variables del sistema. $x_1, \ldots, x_n$ son variables tales como las componentes de velocidad angular y lineal, entre otras. Las variables $\delta_1, \ldots, \delta_h$ representan la deflexión de las h superficies de control tales como alerones, elevadores o flaps. Son pues los coeficientes aerodinámicos los que relacionan las acciones de control con el comportamiento de la aeronave en vuelo. En la figura 4 vemos representadas esas superficies de control y su posición en una aeronave. En esta imagen se aprecian también el sistema de ejes local y un sistema de referencia inercial alineado con el Norte y Este magnéticos.



Figura 4: Superficies de control [11]

La orientación de un sólido en el espacio cartesiano suele expresarse a través de los denominados ángulos de Euler. Estos ángulos expresan los 3 giros que deben realizarse para la transformación desde un sistema inercial de ejes hasta el sistema de ejes locales del avión. Ambos mostrados en la figura anterior. La figura 5 explica el significado de cada uno de esos giros.



Figura 5: Representación de ángulos de Euler [11]

Las ecuaciones cinemáticas que relacionan velocidades angulares del avión con los ángulos de Euler son:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\theta & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (8)$$

Para terminar, el conjunto de expresiones 9 a 17 son el resultado de reordenar las ecuaciones 5, 6, y 8, de forma que sirvan directamente para el cálculo de los valores de salida del modelo. Dichos valores serán los mismos que se extraen del sensor IG500N presentado en el apartado 2.

Ecuaciones de fuerzas:

$$\dot{u} = rv - qw + \frac{\bar{q}S}{m}C_X(\delta) - g\sin\theta + \frac{T}{m} \quad (9)$$

$$\dot{v} = pw - ru + \frac{\bar{q}S}{m}C_Y(\delta) + g\cos\theta\sin\phi \quad (10)$$

$$\dot{w} = qu - pv + \frac{\bar{q}S}{m}C_Z(\delta) + g\cos\theta\sin\phi \quad (11)$$

Ecuaciones de momentos:

$$\dot{p} - \frac{I_{xz}}{I_x}\dot{r} = \frac{\bar{q}Sb}{I_x}C_l(\delta) - \frac{I_z - I_y}{I_x}qr + \frac{I_{xz}}{I_x}qp \quad (12)$$

$$\dot{q} = \frac{\bar{q}S\bar{c}}{I_y}C_m(\delta) - \frac{I_z - I_z}{I_y}pr - \frac{I_{xz}}{I_y}(p^2 - r^2) \quad (13)$$

$$\dot{r} - \frac{I_{xz}}{I_z}\dot{p} = \frac{\bar{q}Sb}{I_z}C_n(\delta) - \frac{I_y - I_x}{I_z}pq - \frac{I_{xz}}{I_z}qr \quad (14)$$

Ecuaciones cinemáticas:

$$\dot{\phi} = p + \tan\theta\,(q\sin\phi + r\cos\phi) \quad (15)$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \quad (16)$$

$$\dot{\psi} = \frac{q\sin\phi - r\sin\phi}{\cos\theta} \quad (17)$$

Como puede deducirse, el paso más crítico para obtener un modelo que se asemeje al comportamiento del avión es el de encontrar las expresiones correctas de las derivadas de estabilidad. En el caso de este proyecto, esas expresiones se obtuvieron por CFD.

4 Integración en Simulink

4.1 Real-time Windows target

La herramienta escogida para la ejecución de las simulaciones HIL, ha sido Simulink. Entre los paquetes que este programa ofrece se encuentra uno de especial interés para la puesta en funcionamiento de este tipo de sistemas. El denominado Real-Time Windows Target (RTWT), provee a los usuarios de un motor de tiempo real para la ejecución de modelos Simulink en PCs con sistema operativo de Microsoft Windows. Además, contiene los bloques necesarios para conectarse a un amplio abanico de placas E/S [12].

Tal y como vimos en el punto 2 de este artículo, las comunicaciones entre el computador de control, la unidad de sensores, y el SSC se establecen a través de puerto serie. Gracias las herramientas que RTWT ofrece, podemos dotar al modelo del avión de su dinámica real, y mandarle las acciones de control externamente desde la FCS, sin que esta note el cambio.

4.2 Integración de sistemas

Tras tomarse la decisión de cual va a ser el entorno de simulación, pasamos a integrar todos los componentes del HIL. Se tiene en primer lugar un modelo implementado en Simulink a partir de las ecuaciones del apartado 3. Dicho modelo puede verse en la figura 6, en la que cada bloque representa un subconjunto de ecuaciones.

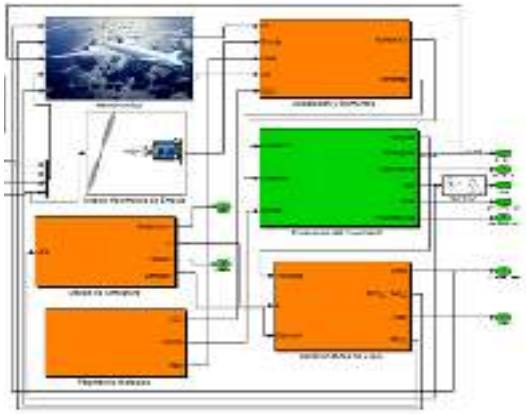


Figura 6: Modelo Simulink Kadett 2400

En segundo lugar, se han utilizado algunas de las herramientas que RTWT deja a disposición del usuario. Los bloques "Packet Output" y "Packet Input", posibilitan la lectura y el envío de paquetes de datos a través del puerto serie. Para el HIL implementado se han usado los dos puertos serie suplantando al sensor IG500N y al SSC. El diagrama completo está disponible en la figura 7.

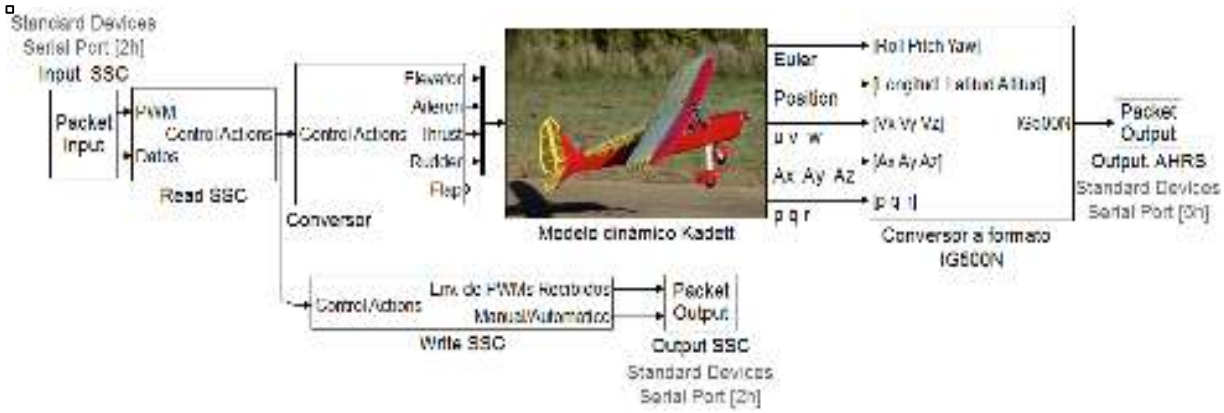


Figura 7: HIL en Simulink

En ella se aprecian varios bloques que se interponen entre el modelo propiamente dicho, y las entradas y salidas del puerto serie. Obviamente han tenido que desarrollarse los algoritmos para transformar los paquetes de datos de entrada a sus formatos correspondientes. Por un lado, el PC-104 envía las tramas de control tal y como el fabricante del SSC especifica. Es por tanto tarea nuestra interpretar dichas tramas y pasarlas a los valores PWM igual que lo haría éste componente. Nótese que solo así el cambio entre modelo y avión será completamente transparente a la FCS. En el lado opuesto, los valores de las variables calculadas en la simulación deberán transformarse al formato de trama binaria que el PC espera recibir del sensor.

Falta, para cerrar el lazo del sistema HIL, un paso final. Las variables manipuladas que se presentaban en las ecuaciones 9 a 17 eran las deflexiones de las superficies de control (en radianes) y la fuerza propulsora (en Newtons). Sin embargo, lo que el PC-104 enviará serán ciertos valores PWM a los servomotores y al variador del motor. Por ello ha habido que realizar las mediciones oportunas para

comprobar la correspondencia entre radianes de deflexión en cada superficie de control, y valor de señal PWM en cada servomotor. Del mismo modo se procedió para obtener la curva de propulsión del motor respecto de los valores de entrada al variador. Quedando la estructura así, completamente definida.

5 Pruebas

Con la intención de evaluar los componentes del sistema Hardware-In-the-Loop, se incluye en el artículo el resultado de uno de los múltiples ensayos llevados a cabo. Por motivos de espacio, el experimento que se muestra pertenece a la evaluación de uno de los subsistemas del modelo. Concretamente, la prueba consiste en controlar un modelo SISO que simula la respuesta del ángulo pitch ϕ, ante variaciones en la deflexión del elevador δ_e. Se ha programado para ello un control de tipo PD en la FCS. Las ecuaciones 18 y 19 representan, respectivamente, la función de transferencia del sistema y el controlador programado.

$$\frac{\phi(s)}{\delta_e(s)} = \frac{4}{s^2 + 2s} \tag{18}$$

$$PD(t) = 0.5\left(e_\phi(t) + \frac{de_\phi(t)}{dt}0.2\right) \tag{19}$$

donde $e_\phi(t)$ es el error de pitch, en radianes. El punto de consigna se ha programado directamente en el PC-104 y es, en este caso, igual a 1 rad. El resultado de la prueba se muestra en la figura 8. Se observan en ella las acciones de control en valor PWM, tal y como son enviadas por la FCS, y su conversión a radianes. La tercera gráfica, muestra la evolución del pitch hacia el punto de consigna.

6 Conclusiones

En un comienzo se exponía la importancia de emplear estrategias de simulación HIL a la hora de abaratar costes en la fase de desarrollo de proyectos de control para UAVs. Se ha explicado a lo largo de este artículo la metodología seguida para la implementación de un sistema de este tipo en un proyecto que envuelve la evaluación de algoritmos avanzados para el desarrollo de nuevos pilotos automáticos.

En el apartado 2 se presentaba la plataforma de vuelo empleada con los componentes que la integran y las interfaces de comunicación entre ellos. Se explican en el tercer punto las ecuaciones que intervienen en la definición del modelo, imprescindibles para cualquier simulación de procesos. Finalmente se explicaba como se han integrado e interconectado cada una de las partes que intervienen en este tipo de simulaciones.

A partir de lo expuesto se concluye que existen herramientas como RTWT para la puesta en marcha de simulaciones HIL de una forma barata y rápida, que ofrecen muchos beneficios a la hora de validar diseños de control, tanto a nivel de software como de hardware.

Se deja como trabajo futuro la mejora de las ecuaciones del modelo a partir de un proceso de identificación mediante datos de vuelo reales. Además, se desea evolucionar el hardware de la FCS hacia algún componente de tamaño más reducido. Esta evolución se apoyará fuertemente en la herramienta expuesta en este artículo.

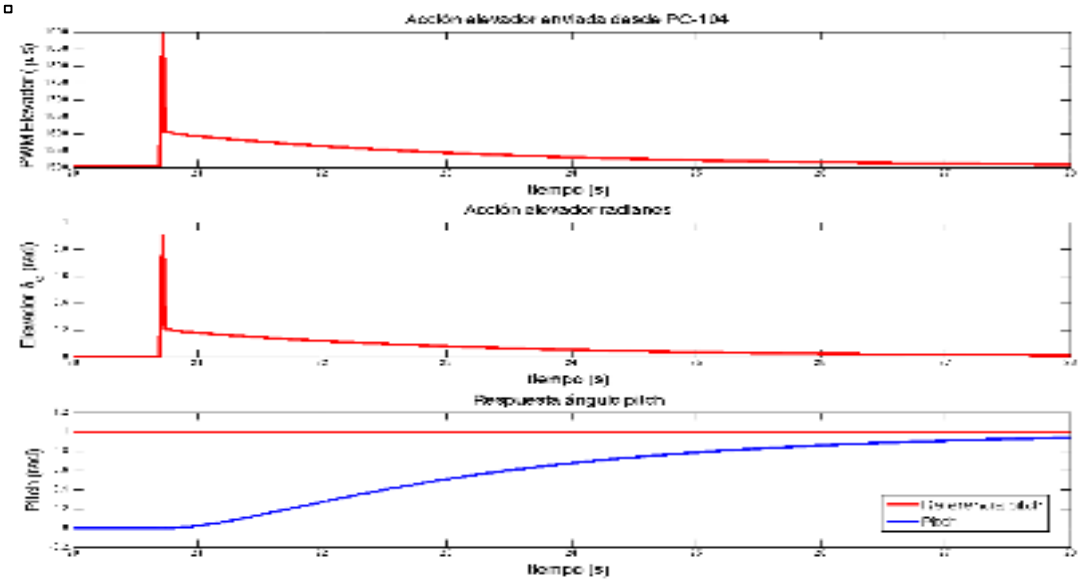


Figura 8: Control con HIL

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Ministerio de Ciencia e Innovación, gobierno de España a través del proyecto TIN2011-28082, por el Ministerio de Economía y Competitividad a través de la beca FPI BES-2012-056210; por la Universitat Politècnica de València a través del proyecto PAID-06-11 y la beca FPI 2010/19.

Referencias

[1] Peter van Blyenburgh, (1999). UAVs: an Overview. Air&Space Europe. Vol. I. No 5/6.

[2] W. Krüll, R. Tobera, I. Willms, H. Essen, N. Wahl, (2012). Early forest fire detection and verification using optical smoke, gas and microwave sensors. In: International Symposium on Safety Science and Technology. Procedia Engineering, Elsevier.

[3] H. Xiang, L. Tian, (2011). Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV). Biosystems Engineering, Elsevier.

[4] H. Wenbo, Z. Qiang, (2012). The Hardware-in-the-loop Simulation on the Control System of a Small Launch Vehicle. Procedia Engineering, Elsevier.

[5] D. Jung, J. Ratti, P. Tsiotras, (2009). Real-time Implementation and Validation of a New Hierarchical Path Planning Scheme of UAVs via Hardware-in-the-Loop Simulation. J. Intell Robot Systems, Springer.

[6] G. Cai, M.Chen, T. H. Lee, M. Dong, (2009). Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters. Mechatronics, Elsevier.

[7] Eric Mueller (2007). Hardware-in-the-loop Simulation Design for Evaluation of Unmanned Aerial Vehicle Control Systems. AIAA Guidance, Navigation, and Control and Co-located Conferences. In: AIAA Modeling and Simulation Technologies Conference and Exhibit.

[8] Graupner. Specifications and instructions for KADETT 2400. Disponible en: http://www.graupner.de/

[9] J.Velasco, S.García-Nieto, G. Reynoso-Meza y J. Sanchis, (2012). Desarrollo y evaluación de una estación de control de tierra para vehículos aéreos no tripulados. XXXIII Jornadas de Automática.

[10] Vladislav Klein, Eugene A. Morelli, (2009). Aircraft System Identification. Theory and Practice. AIAA Education Series. Edición 2006.

[11] Dr. G. Ducard. Unmanned Aircraft Design, Modeling and Control. Lecture Notes. Introduction to Navigation Systems. ETH.

[12] MathWorks Real-Time Windows Target DataSheet. Disponible en: http://www.mathworks.es/products/datasheets/pdf/real-time-windows-target.pdf

ARQUITECTURA DE TIEMPO REAL PARA EL CONTROL DE ACTITUD EN UN SATÉLITE EXPERIMENTAL*

Juan Zamorano, Ángel Esquinas, Peter Bradley, Daniel Brosnan,
Jorge Garrido, Emilio Salazar, Alejandro Alonso, Juan A. de la Puente

Grupo de Sistemas de Tiempo Real y Arquitectura de Servicios Telemáticos (STRAST)
Universidad Politécnica de Madrid (UPM)

Resumen

La misión UPMSat-2 tiene por objetivo la construcción y el lanzamiento de un microsatélite experimental que sirva como plataforma de educación e investigación en diversos aspectos de la ingeniería de sistemas espaciales. En este artículo se describe la arquitectura y el diseño del software de tiempo real que realiza el control de actitud del satélite. La arquitectura está basada en modelos, y las herramientas utilizadas permiten la validación de las propiedades de tiempo real y la generación de código de forma casi totalmente automática.

Palabras clave: Sistemas de control por computador, sistemas de tiempo real, diseño de software basado en modelos.

1. INTRODUCCIÓN

UPMSat-2 es un proyecto de la Universidad Politécnica de Madrid (UPM) que tiene como objetivo la construcción, puesta en órbita y operación de un satélite experimental que sirva como demostrador tecnológico y plataforma educativa en el ámbito de los sistemas espaciales. El proyecto está liderado por el Instituto Ignacio Da Riva(IDR/UPM), y se lleva a cabo con la colaboración del grupo STRAST/UPM,la empresa TECNOBIT,y otros grupos universitarios y empresas.

UPMSat-2 es un microsatélite de unos 50 kg de masa, con una envolvente geométrica de $0,5 \times 0,5 \times 0,6$ m. La órbita prevista es polar heliosíncrona a unos 600 km de altura. El lanzamiento está previsto para 2015, y la vida útil de la misión se estima en unos dos años.

El grupo de Sistemas de Tiempo Real y Arquitectura de Servicios Telemáticos (STRAST) se ocupa de desarrollar el software para el computador embarcado y la estación de tierra de la misión. En este artículo se describen los aspectos relacionados con el software de tiempo real que efectúa el control de actitud del satélite. En el apartado 2 se resumen las principales características de la arquitectura de hardware y software del computador embarcado. Los elementos principales del sistema de control de actitud se describen en el apartado 3, y el software de tiempo real correspondiente se describe en el apartado 4. Por último, el apartado 5 resume las principales conclusiones del trabajo.

*Trabajo financiado parcialmente por el Plan Nacional de I+D+I, proyecto TIN2011-28567-C03-01 (HI-PARTES).

2. COMPUTADOR EMBARCADO

El computador embarcado (OBC, *on-board computer*) del satélite realiza las siguientes funciones:

- Control de actitud (ADCS, *Attitude determination and control)*;
- supervisión de la plataforma del satélite (*housekeeping*);
- gestión de mensajes de telemando (TC, *telecommand*) y telemedida (TM, *telemetry*);
- gestión de los experimentos que constituyen la carga útil del satélite.

La figura 1 muestra las interfaces del computador con los sensores de que se dispone para realizar estas funciones.

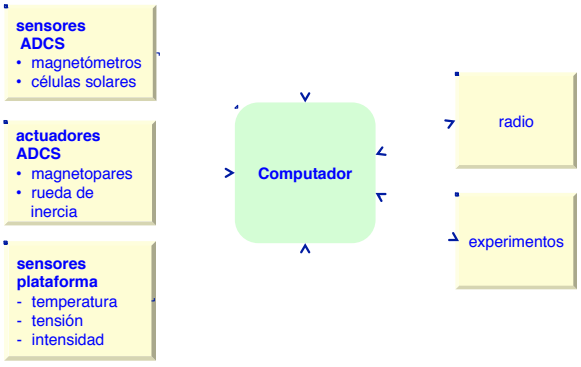


Figura 1: Diagrama de contexto del computador embarcado.

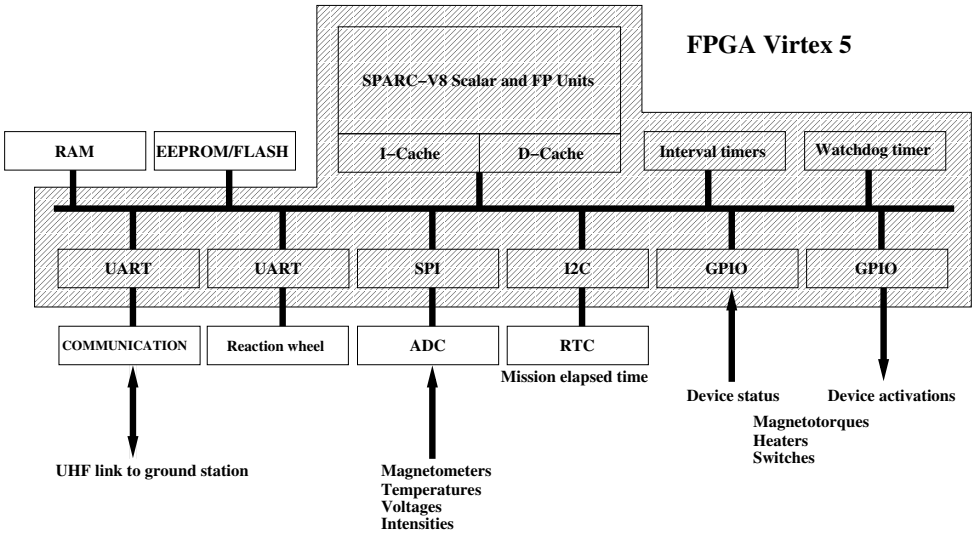


Figura 2: Computador embarcado.

2.1. ARQUITECTURA DE HARDWARE

El hardware del computador embarcado está basado en un procesador LEON3 con arquitectura SPARC v8 [8, 14], con memoria RAM y memoria no volátil tipo EEPROM para almacenar los datos de configuración y de telemedida para su posterior envío a la estación de tierra. El computador incluye también interfaces de entrada y salida analógica y digital para la interacción con los sensores, así como interfaces RS-232 y RS-422 para la radio, las conexiones de depuración y pruebas, y algunos de los experimentos [4].

La versión de ingeniería de la OBC se ha desarrollando a partir de la biblioteca de *IP cores* en VHDL GRLIB. Esta biblioteca contiene el modelo sintetizable del procesador de 32 bits LEON3 junto con controladores de buses de sistema AMBA, controladores de memoria así como controladores de buses auxiliares y dispositivos periféricos. La biblioteca está disponible en código fuente VHDL y se distribuye con licencia GNU GPL. Por lo tanto, es posible desarrollar (SOC, *Systems On a Chip*) para su uso en computadores de a bordo con una adecuada configuración y síntesis sobre dispositivos programables como FPGAs.

La figura 2 muestra la estructura del computador embarcado que se ha sintetizado en una FPGA, la versión de ingeniería del computador usa una tarjeta de evaluación ML507 de la FPGA de Xilinx Virtex-5 FXT. Esta versión de ingeniería se usa para el desarrollo del software embarcado y sirve de base para la versión de vuelo del computador que contendrá *chips* de memoria y FPGA resistentes a la radiación. El coste de estos componentes y los requisitos de consumo de energía limitan la cantidad de memoria disponible a 4 MB de SRAM y 1 MB de EEPROM. Por motivos similares, la velocidad del procesador LEON3 es de 50 MHz.

2.2. ARQUITECTURA DE SOFTWARE

El software del computador embarcado se ha diseñado siguiendo un enfoque basado en modelos (MDE) [1]. Los distintos subsistemas se modelan en lenguajes de alto nivel como Simulink® o el perfil MARTE de UML [11, 12]. A partir de los modelos de alto nivel se genera automáticamente el código fuente en C o Ada utilizando herramientas asociadas a los lenguajes de modelado, junto con otras desarrolladas por el grupo [13].

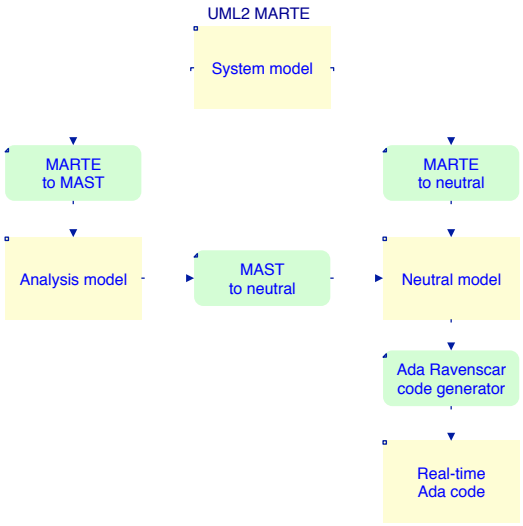


Figura 3: Proceso de desarrollo de software.

La figura 3 muestra de forma esquemática el proceso de desarrollo utilizado para el software.

El software embarcado tiene requisitos de tiempo real y se hace necesario la realización de un análisis de planificabilidad para verificar que las acciones de los subsistemas se realizan en el momento correcto [6]. Por lo tanto, las herramientas incluyen restricciones para asegurar que los modelos sean compatibles con el perfil de Ravenscar [3]. El perfil de Ravenscar se definió como un conjunto de restricciones a la parte concurrente del lenguaje de programación Ada, con el objetivo que la arquitectura de software resultante se pueda analizar temporalmente. Aunque el perfil se definió originalmente para el lenguaje Ada, su modelo computacional se puede extrapolar a otros paradigmas de programación concurrente como POSIX. La plataforma de ejecución está basada en el núcleo de tiempo real ORK+ [5], que da soporte al perfil de Ravenscar. Así que tanto el compilador como el núcleo de tiempo real realizan comprobaciones del código generado para verificar que se cumplan las restricciones.

3. CONTROL DE ACTITUD

La actitud del satélite está representada por la orientación de los ejes del satélite con respecto a un sistema de referencia local vertical. Esta orientación viene dada por la matriz de transformación entre ambos sistemas de referencia, construida a partir de los ángulos de Euler, ϕ, θ y ψ. El objetivo del sistema de control de actitud es mantener el eje Z del satélite orientado hacia el centro de gravedad de la Tierra, mientras que su eje Y se orienta en una dirección normal al plano de la órbita. El satélite debe girar entorno al eje X con una velocidad angular de $0,01\,\mathrm{rad/s}$ con el objetivo de mantener una temperatura homogénea en las caras del satélite.

La actitud del satélite se puede determinar a partir de las medidas obtenidas por tres magnetómetros ortogonales, que permiten determinar el valor y la dirección del campo magnético terrestre en el sistema de referencia local del satélite. Para modificar la actitud del satélite se dispone de tres magnetopares, que generan un momento de rotación al interaccionar con el campo magnético terrestre.

Para diseñar el algoritmo de control se ha simulado la dinámica del satélite y las fuerzas externas que actúan sobre él mediante un modelo en Simulink (figura 4).

Las variables del modelo representan los distintos momentos de fuerzas que actúan sobre el satélite, los cuaterniones que representan la actitud del satélite respecto al sistema de referencia vertical, la matriz de rotación desde el sistema de ejes del satélite ($\mathbf{C}_{BV}$), y los componentes de la velocidad

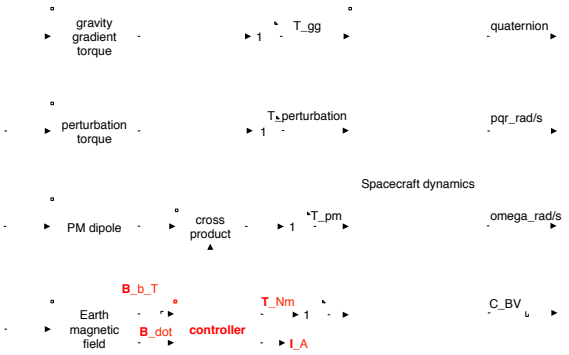


Figura 4: Modelo de la dinámica del satélite.

angular (**pqr**).

El bloque *controller* representa el algoritmo de control, que tiene como entradas las medidas del campo magnético proporcionadas por los magnetómetros ($\mathbf{B}_{b_T}$) y sus derivadas en el tiempo ($\mathbf{B}_{\mathrm{dot}}$). Las salidas del controlador son las intensidades que se suministran a los magnetopares ($\mathbf{I}_A$) y los pares que ejercen éstos ($\mathbf{T}_{\mathrm{Nm}}$). Los sensores y actuadores propiamente dichos están incluidos en el bloque controlador.

Los detalles del algoritmo y su justificación pueden verse en el documento [7]. El periodo de muestreo del algoritmo de control es de un segundo. La adquisición de los datos de los magnetómetros se realiza mediante el conversor analógico digital del OBC (figura 2) con una resolución de 12-bits y la actuación sobre los magnetopares se hace mediante modulación de ancho de impulso (PWM, *Pulse Width Modulation*) mediante salidas digitales del OBC.

4. SOFTWARE DE CONTROL

El software del sistema de control de actitud (ADCS) tiene varios componentes, que se representan en el diagrama MARTE-UML de la figura 5.

El componente AttitudeControl se ejecuta periódicamente e invoca la función ControlAlgorithm, cuyo código fuente en C se genera automáticamente a partir del modelo de simulación anterior. Los demás componentes permiten ajustar los parámetros del controlador, proporcionar medidas al sistema de telemetría, y ajustar el controlador en función de los cambios de modos del sistema. Estos componentes se deben ejecutar concurrentemente con otros subsistemas de software.

El entorno de desarrollo dispone de herramientas para calcular el peor caso de tiempo de cómputo (WCET, *Worst Case Execution Time*) [2] de

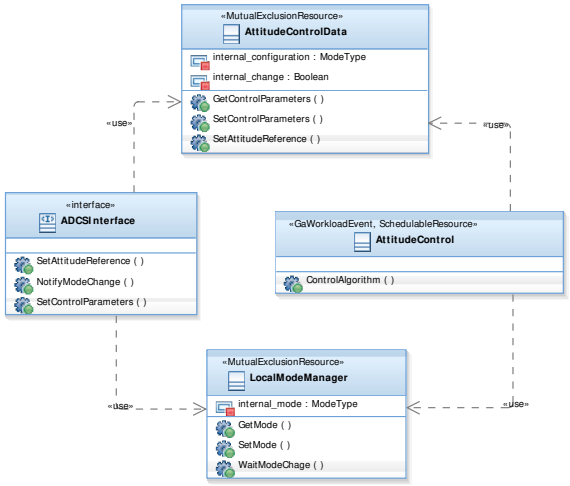


Figura 5: Estructura del software del sistema ADCS.

las funciones del sistema. Entre estas funciones está el control de actitud, los detalles del cálculo del WCET pueden verse en el documento [9]. Con la arquitectura de software y los WCET de las funciones del sistema se analizan los requisitos de tiempo real mediante herramientas específicas [10].

Una vez comprobado el comportamiento temporal, se genera todo el esqueleto del código concurrente y de tiempo real en Ada a partir del modelo MARTE-UML [13]. La compilación e implantación en el computador embarcado se realizan de forma inmediata usando la cadena de compilación GNAT/ORK [15].

5. CONCLUSIONES

El satélite UPMSat-2 (figura 6)es una plataforma de gran interés para la experimentación de nuevas técnicas por parte de los grupos de investigación de la UPM.



Figura 6: Vista general del satélite UPMSat-2.

En particular, las técnicas de desarrollo de software de tiempo real en las que trabaja el grupo STRAST se han aplicado al diseño del software de control del satélite. Los resultados obtenidos hasta ahora son satisfactorios, y por tanto está previsto continuar con el desarrollo del resto del software embarcado utilizando estas mismas técnicas en los próximos meses.

Agradecimientos

Los autores agradecen la colaboración del resto del equipo del proyecto UPMSat2, especialmente la ayuda prestada pro Assal Farrahi, Javier Cubas y Ángel Sanz.

Referencias

[1] Alejandro Alonso, Emilio Salazar, and Juan A. de la Puente. Design of on-board software for an experimental satellite. In *Jornadas de Tiempo Real — JTR-2013*, 2103. Available at `www.dit.upm.es/~str/papers/pdf/alonso&13a.pdf`.

[2] Esteban Asensio, Jorge López, Juan A. de la Puente, and Juan Zamorano. Herramientas de análisis temporal para el desarrollo de sistemas de tiempo real críticos. In *Proc. CEDI 2010*, September 2010. (In Spanish). Submitted for publication.

[3] Alan Burns, Brian Dobbing, and Tullio Vardanega. Guide for the use of the Ada Ravenscar profile in high integrity systems. Technical Report YCS-2003-348, University of York, 2003.

[4] Juan A. de la Puente, Juan Zamorano, Alejandro Alonso, and Daniel Brosnan. A real-time computer control platform for an experimental satellite. In *Jornadas de Tiempo Real — JTR-2012*, 2012. Available at `http://www.ctr.unican.es/jtr12/programme.html`.

[5] Juan A. de la Puente, Juan Zamorano, José A. Pulido, and Santiago Urueña. The ASSERT Virtual Machine: A predictable platform for real-time systems. In Myung Jin Chung and Pradeep Misra, editors, *Proceedings of the 17th IFAC World Congress*. IFAC-PapersOnLine, 2008.

[6] European Cooperation for Space Standardization. *ECSS-E-ST-40C Space engineering — Software*, March 2009. Available from ESA.

[7] Assal Farrahi, Javier Cubas, and Ángel Sanz. Design of the attitude control subsystem of the upmsat-2 satellite. Technical report, Instituto de Microgravedad Ignacio da Riva, 2013.

[8] Gaisler Research. *LEON3 - High-performance SPARC V8 32-bit Processor. GRLIB IP Core User's Manual*, 2012.

[9] Jorge Garrido, Daniel Brosnan, Juan A. de la Puente, Alejandro Alonso, and Juan Zamorano. Analysis of WCET in an experimental satellite software development. In Tullio Vardanega, editor, *12th International Workshop on Worst-Case Execution Time Analysis*, volume 23 of *OpenAccess Series in Informatics (OASIcs)*, pages 81–90. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012.

[10] Michael González Harbour, J. Javier Gutiérrez, J. Carlos Palencia, and José María Drake. MAST modeling and analysis suite for real time applications. In *Proceedings of 13th Euromicro Conference on Real-Time Systems*, pages 125–134, Delft, The Netherlands, June 2001. IEEE Computer Society Press.

[11] *OMG Unified Modeling Language (UML)*, 2011. Version 2.4.1.

[12] *OMG UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems*, 2011. Version 1.1.

[13] Emilio Salazar, Alejandro Alonso, Miguel A. de Miguel, and Juan A. de la Puente. A model-based framework for developing real-time safety Ada systems. In *Reliable Software Technologies - Ada-Europe 2013*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013.

[14] SPARC International, Upper Saddle River, NJ, USA. *The SPARC architecture manual: Version 8*, 1992.

[15] Juan Zamorano and José F. Ruiz. GNAT/ORK: An open cross-development environment for embedded Ravenscar-Ada software. In Eduardo F. Camacho, Luis Basañez, and Juan Antonio de la Puente, editors, *Proceedings of the 15th IFAC World Congress*. Elsevier Press, 2003.

EVALUATION OF SHAPE AND COLOR DESCRIPTORS BY USING BAG OF WORD TECHNIQUES WITH ONE VS ALL CLASSIFICATION

Fidalgo E. [1], Fernández-Robles L.[2], García-Ordás M. [3], García-Olalla O. [4], Alegre E. [5]
[1,3,4,5] Department of Electrical, Systems and Automatic Engineering
[2] Department of Mechanical, IT and Aerospatial Engineering
Campus de Vegazana S/N 24071, University of León, León, Spain
{eduardo.fidalgo.fernandez[1]}@gmail.com , {l.fernandez[2], mgaro[3], ogaro[4], enrique.alegre[5]}@unileon.es

Abstract

This paper presents the evaluation of several shape and color descriptors on two different DataSets, the "Soccer" dataset and a new Dataset called "Karinas_Dataset1". To represent an image by its corresponding shape (or color) descriptor, a standard BoW creation technique has been applied. In this way, all the images in both DataSets are represented by a descriptor whose dimension depends on the visual dictionary previously created. Afterwards, a one-vs-all classifier has been trained and tested using random images from the Datasets. To avoid spurious data the tests have been repeated 10 times and an average result has been taken as a reference.

Key Words: shape descriptor, color descriptor, bag of words, one-vs-all.

1. INTRODUCTION

Object classification in digital images is one of the most challenging tasks in computer vision. Advances in the last decade have produced methods to extract and describe distinctive local features in natural images. In order to apply machine learning techniques in computer vision systems, a representation based on these features is needed.

The typical object recognition process is composed of the following steps: (i) extraction of local image features (e.g., Hue descriptors), (ii) encoding of the local features in an image descriptor (e.g., a histogram of the quantized local features), and (iii) classification of the image descriptor (e.g., by a support vector machine).

During the past few years, Bag-of-Words (BoW) approaches have allowed significant advances in image classification [1].

Local features are an efficient tool for image classification due to their robustness with respect to occlusion and geometrical transformations [14]. Among the multiple approaches that have been proposed to describe the shape of local features, the SIFT descriptor [5] has proven to be one of the best [13]. But recently people have started to enrich local image descriptors with color information [1, 9, 10, 4, 11, 12]. The main challenge in color description is to obtain robustness with respect to photometric variations, as they are common in the real world, like shadow and shading variations and changes of the light source color. In order to achieve this purpose, the color descriptors are generally based on photometric invariants [6, 10, 16], such as hue and normalized RGB.

The purpose of this article is to compare the results of applying three different descriptors (SIFT, Hue Histogram and Color Name) using two Datasets (Soccer and Karinas_Dataset1). The descriptors are initially compared by plotting the results of the one-vs-all logistic regression classification with different values of iterations and regularization factor.

A summary of the used descriptors is explained in Section 2. Section 3 comments the methods for evaluating the previous descriptors once they are obtained for each image on the Data Set. Section 4 details the Datasets used in this article. Section 5 shows and discusses the results obtained after applying the evaluation method over the different descriptors. The conclusion of our work and possible future directions are stated in Section 6.

2. DESCRIPTORS USED

2.1 SHAPE DESCRIPTORS

2.1.1 SIFT

Scale Invariant Feature Transform (SIFT) is an algorithm published by David Lowe in 1999 [3] and further explained in 2004 [5]. It consists of extracting

distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. For an object in an image, interesting points on the object can be extracted to provide a description of the object based on the features of its interesting points and then they are stored in a database. A new image is matched by individually comparing each feature from the new image to this previous database and finding candidate matching features based on measuring the distance of their feature vectors. The correct matches can be filtered from the full set of matches by identifying subsets of key points that agree on the object and its location, scale and orientation in the new image.

The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination which is essential to perform reliable recognition. This method has proved to robustly identify objects even among clutter and under partial occlusion.

2.1.1.1 Scale-space extrema detection

The first stage of keypoint detection is to identify locations and scales that can be repeatable assigned under differing views of the same object. For each octave of scale space σ, the image is convolved with Gaussian filters (which is a scale-space kernel) at different scales, and then the Difference-of-Gaussian (DoG) function of two successive Gaussian-blurred images separated by a constant multiplicative factor k are taken. A DoG image $D(x, y, \sigma)$ is computed by

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \qquad (1)$$

where $L(x, y, k\sigma)$ is the convolution of the original image $I(x, y)$ with the Gaussian filter $G(x, y, k\sigma)$ at scale $k\sigma$. After each octave, the image is down-sampled by a factor of 2. The value of σ is divided by 2 by taking every second pixel in each row and column. This process is schematized in Figure 1. In order to detect the local maxima and minima of $D(x, y, \sigma)$, each pixel is compared to its eight neighbours in the current image and nine neighbours in the scale above and below. It is selected as a keypoint only if it is lager than all of these neighbours or smaller than all of them.

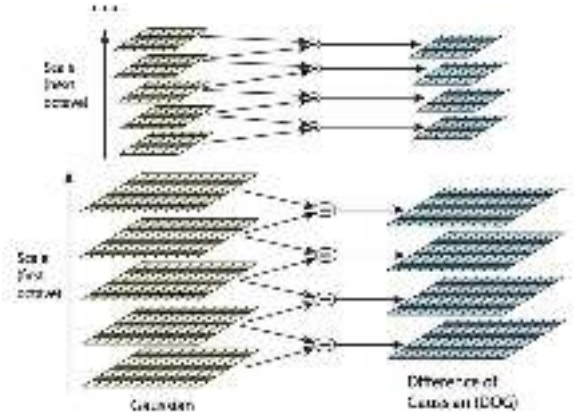


Figure 1. Scale-space extrema detection sketch.

2.1.1.2 Keypoint localization

Scale-space extreme detection produces many keypoint candidates and some of them are unstable. The next step consists of performing a detailed fit to the nearby data for location, scale and ratio of principal curvatures. This allows points with low contrast (and therefore sensitive to noise) or poorly localized along an edge to be rejected.

In order to discard low contrast key points, this approach calculates the interpolated location of the extreme by using the quadratic Taylor expansion of the DoG scale-space function, with the candidate keypoint as the origin. Afterwards, strong responses along edges are suppressed by computing the principal curvatures from a 2x2 Hessian matrix computed at the location and scale of the keypoint. A poorly defined peak in the DoG function will have a large principal curvature across the edge but a small one in the perpendicular direction.

2.1.1.3 Orientation assignment

One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations. For an image sample at scale σ, the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, are pre-computed using pixel differences. An orientation histogram with 36 bins is formed, with each bin covering 10 degrees. Each sample in the neighbouring window added to a histogram bin is weighted by its gradient magnitude and by a Gaussian-weighted circular window with σ that is 1.5 times that of the scale of the keypoint. The peaks in this histogram correspond to dominant orientations. Once the histogram is filled, the orientations corresponding to the highest peak and local peaks

that are within 80% of the highest peaks are assigned to the keypoint.

2.1.1.4 Keypoint descriptor

The local image gradients are measured at the selected scale in the region around each keypoint. First a set of orientation histograms is created on 4x4 pixel neighbourhoods with 8 bins each. These histograms are computed from magnitude and orientation values of samples in a 16x16 region around the keypoint such that each histogram contains samples from a 4x4 sub-region of the original neighbourhood region. Then, the magnitudes are weighted by a Gaussian function. The descriptor becomes a vector of all the values of these histograms. Since there are 4x4=16 histograms each with 8 bins the descriptor has 128 elements. A representation of this process can be seen in Figure 2. This vector is then normalized to unit length in order to enhance invariance to affine changes in illumination. Effects of non-linear illumination are avoided applying a threshold of 0.2 and then the vector is again normalized.

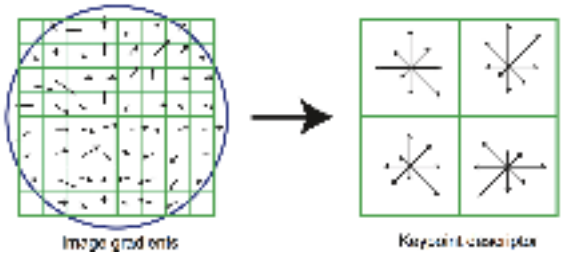


Figure 2. Keypoint descriptor composition.

2.2 COLOR DESCRIPTORS

2.2.1 Hue

Hue descriptor is composed by 36 features.

In the HSV color space, it is known that the hue becomes unstable near the gray axis. To this end, van de Weijer et al. [7] applied an error propagation analysis to the hue transformation. The analysis shows that the certainty of the hue is inversely proportional to the saturation. Therefore, the hue histogram becomes more robust by weighing each sample of the hue by its saturation. The H color model is scale-invariant and shift-invariant with respect to light intensity [7].

That is to say, for hue descriptor, it is computed the hue and saturation at each position; these can also be represented as a vector, where the hue is the angle and the saturation the length. It is computed the hue histogram of the patch where the strength of the update is equal to the saturation of the measurement. This ensures that pixels with low saturation (black-grey-white), where the hue is undefined, have no influence on the final color descriptor.

2.2.2 Color Name

The set of color names used in English is huge and includes several labels like ”white”, ”green”, ”pastel”, and ”light blue”. For this descriptor, the 11 basic color terms of the English language: black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow are used. The basic color terms were defined in the influential work on color naming of Berlin and Kay [2]. A basic color term of a language is defined as being not subsumable to other basic color terms (e.g. turquoise can be said to be a light greenish blue, and therefore it is not a basic color term). The color descriptor K is described as the vector containing the probability of the color names given an image region R

$$K = \{p(n_1 \mid R), ..., p(n_{11} \mid R)\} \qquad (2)$$

With

$$p(n_1 \mid R) = \frac{1}{N} \sum_{x \in R} p(n_{i1} \mid f(x)) \qquad (3)$$

where n_i is the i-th color name, x are the spatial coordinates of the N pixels in region R, f = {L*, a*, b*} and p ($n_i \mid f$) is the probability of a color name given a pixel value. The probabilities p ($n_i \mid f$) are computed from a set of manually annotated images.

3. BOW PROCESS

Bag of Words method requires the input format shown in Figure 3 so a normalization process is carried out with the image points descriptors obtained with SIFT.

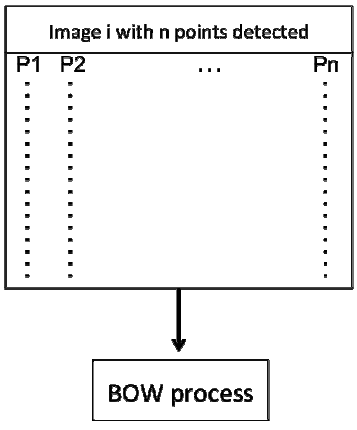


Figure 3. Input format required by BOW.

The Bag of Words method consists mainly on two steps:
First, n centres are obtained taking into account all the image points using a clustering algorithm. This process is called *CalculatedDictionary*. In our case, we have evaluated Bag of Words with k-means as clustering method.

Secondly, each point is assigned to its corresponding centre taking in consideration the distance between points and centres. This process is called *Assignment*. Both processes are shown in Figure 4.

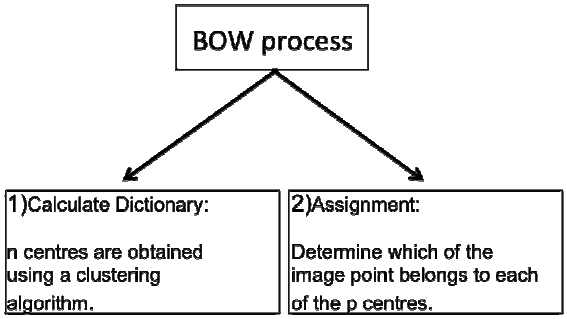


Figure 4. Bow process steps.

Both methods are carried out for each image (See Figure 5) so at the end of the process, a matrix like the one shown in Figure 6 is obtained.

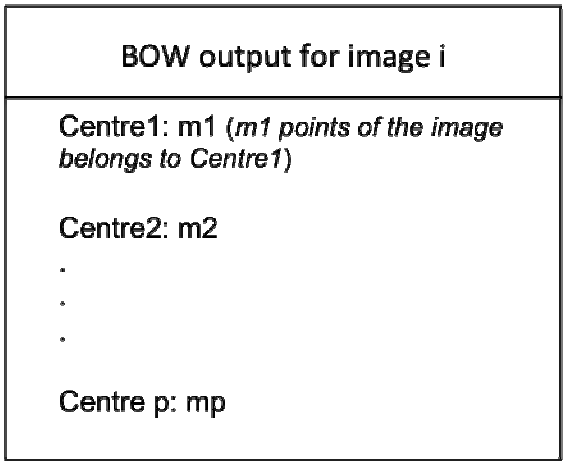


Figure 5. For each image, BOW returns the number of image points belonging to each centre.

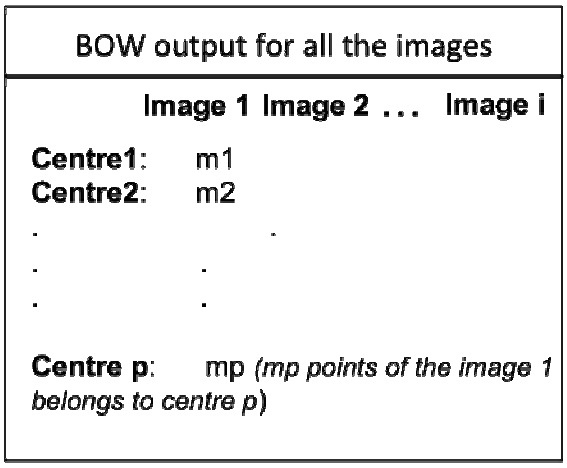


Figure 6. Matrix with the number of image points belonging to each centre for all the images.

4. ONE-VS-ALL EVALUATION METHOD

A one versus all classification based on logistic regression have been carried out in order to evaluate these descriptors. The logistic regression classification has been performed for each class, considering one class as positive and all the rest as negatives. Once all the logistic classifiers have been trained, each element is assigned to the class that gives the best response. In order to find the best configuration, several values for the regularization parameter (λ) have been tested just to avoid over fitting (for high values of λ) and under fitting (small values of λ) problems. $\lambda=0$ means that no regularization is applied. Small values of λ are referred to values close to $\lambda=0$.

5. DATASETS USED

Two datasets were used for the evaluation, the well-known Soccer Dataset (color dominant) and a new dataset created by VARP Group called "Karinas_Dataset1".

5.1 SOCCER

This data set [17] contains images from 7 soccer teams (see Figure 7) taken from the web, comprehending 40 images per class, divided into 25 training and 15 testing images per class. Although, players of other teams were allowed to appear in the images, no players being a member of the other classes in the data set were accepted.

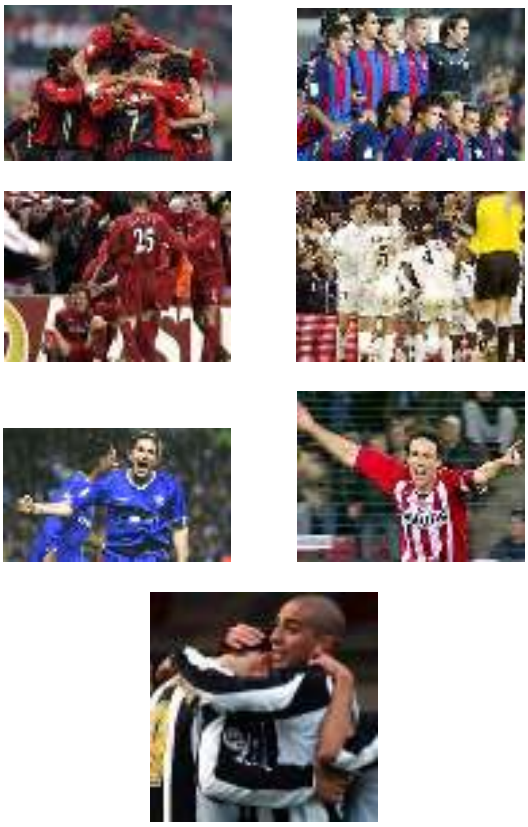


Figure 7: Soccer Dataset: One sample per class.

5.2 KARINAS_DATASET1

VARP Group of the University of León has created Karinas_Dataset1 [15], composed by 614 frames of 640x480 pixels that come from 3 videos recorded under different conditions. All videos were recorded in different bedrooms and with different distributions, illumination, textures, etc., making the object retrieval a challenging task. Nevertheless some objects are present in all videos such as two toy cars, some clothespins, a stuffed bee, some pens, some cups or a child book together with a big doll. The doll is usually the principal actor in the videos and helps us to simulate partial occlusions of the objects and a more realistic scenario. Although these objects are present in every video, they do not appear in every frame. For image classification in this article has been used the book, the blue and yellow car, and the pink, blue and green clothespin shown in Figure 8. The total number of query objects present among the 614 frames that we used in this paper can be found in Table 1.

Table 1: Karinas_Dataset1 Dataset information.

Object	Number
Book	115
Blue car	102
Green car	138
Pink clothespin	125
Blue clothespin	92
Green clothespin	42

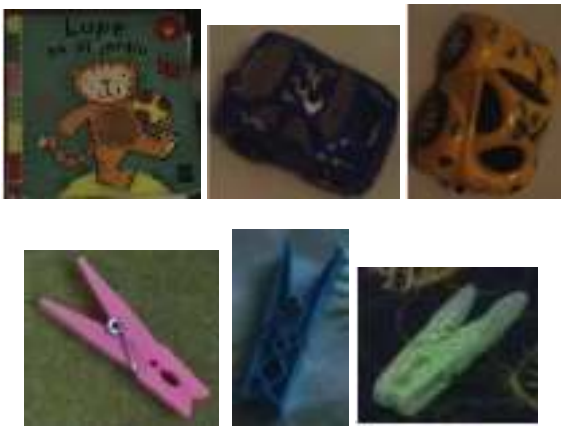


Figure 8: Karinas_Dataset1: One sample per class.

6. RESULTS

6.1 EXPERIMENT ARRANGEMENT

In the experiments, the BoW creation explained in chapter 3 has been used, in which the key points are extracted through Harris-Laplace keypoint detector. The visual dictionaries are constructed using the K-means clustering algorithm. Feature vectors (bag-of-word histograms) are built for each cue separately by using the corresponding visual dictionary. In this series of experiments, two visual cues were used: the shape and the color. Shape information is extracted from SIFT descriptors, and color information is extracted from Hue-histograms [9] and Color Name (CN) [8] descriptors.
The visual dictionary size of the different descriptors remains as follows:

- SIFT: 400 visual words.
- Hue: 300 visual words.
- Color Name: 300 visual words.

All the reported results are obtained using a one-vs-all classifier learned from the feature vectors of the different methods. As an evaluation criterion the accuracy (%) has been used.

The following table represents a sample of the tests performed. Each descriptor has been tested with different values of regularization parameter ($\lambda = 0$ means no regularization at all) and different number of iterations. Since training and test images for each experiment are randomly chosen (train 63,5% and test 37,5%), the experiment for each descriptor, with a fixed value of λ and a fixed number of iterations is repeated ten times and the mean value and the

standard deviation of the obtained results are computed.

Table 2: Experiments Table Sample.

	λ = 0 Iter = 100		λ = 0 Iter = 1000	
	time (s)	% Accu	time (s)	% Accu
Test 1	6,06	35,29	55,20	28,43
Test 2	5,25	31,05	52,66	33,66
Test 3	5,12	35,95	52,88	30,72
Test 4	5,10	31,70	52,68	32,03
Test 5	5,22	35,29	53,28	31,70
Test 6	5,27	34,31	51,63	31,70
Test 7	5,42	31,05	53,64	30,72
Test 8	5,16	34,64	52,72	31,05
Test 9	5,28	33,99	52,83	31,05
Test 10	5,12	33,01	54,55	30,07
Mean	**5,30**	**33,63**	**53,21**	**31,11**
STD	0,28	1,83	1,03	1,36

To avoid an excessive number of tables (8 tables per descriptor) we created two plots, one per Dataset, to represent the mean values obtained under every condition (values of λ and Iterations). One example of the results with λ equal 0 and Iterations equal to 100 and 1000 is shown in Table 2.

6.2 SOCCER RESULTS

The results obtained for this dataset are showed in Figure 9.

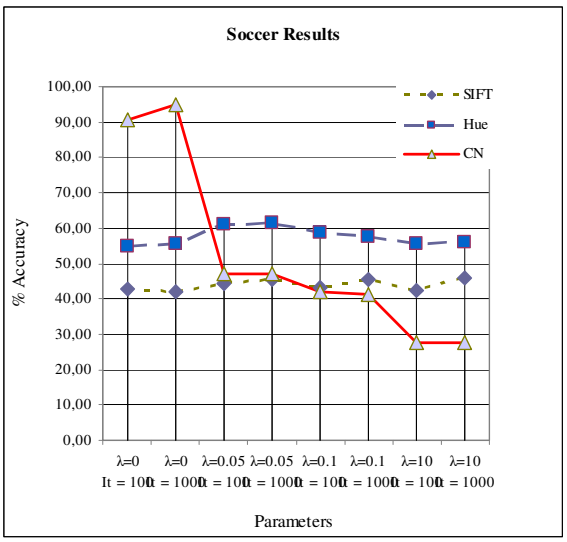


Figure 9: Soccer Dataset Plot Results.

On the one hand, as expected, the SIFT descriptors are less relevant than the Color ones (Hue and CN) for classifying soccer images.

On the other hand, the obtained results are different when using or not using regularization depending on the descriptor. It can be easily seen that with no regularization (λ=0) the results are quite better than even with a small value of lambda for the case of Color Name descriptors. However, for SIFT and Hue, when adding this regularization factor the results remains practically the same despite the value of lambda used in the case. More precisely, they are slightly better with a small factor of regularization.

Moreover, an increase on the iterations used to calculate the outputs of the LR model implies a slightly increase in the accuracy of the classification. Nevertheless, in some cases it is not worth it since the rise of time invested due to the increase of iterations needed is not proportional related with the improvement on the accuracy results.

6.3 KARINA_DATASET1 RESULTS

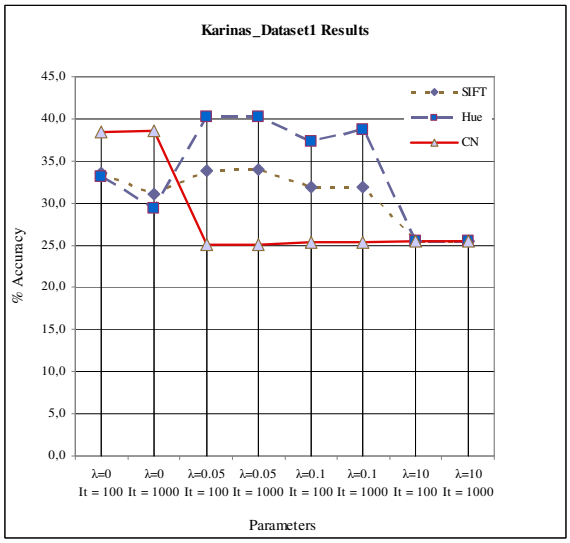


Figure 10: Karinas_Dataset1 Dataset Plot Results.

Unlike the Soccer dataset that is color dominant, Karinas_Dataset1 is considered both shape and color dominant due the complexity of the images extracted. Results obtained with no regularization confirm the previous statement in the case of no regularization classification, due the fact that the three descriptors obtained similar results (all in a range of 30-40%). These poor accuracy results may derive from the fact that the dataset is too small compared with the objects to be identified (6 query objects and 614 images for training and test).

The use of a small factor of regularization causes a slight increase in the SIFT accuracy and it also causes a noticeable improvement in the case of Hue descriptors results. But the Color Name descriptors are severely penalized by the used of the regularization term.

An increase on the iterations used to calculate the outputs of the LR model implies a slight increase in the accuracy of the classification, but again the time invested is also increased.

Like in the Soccer dataset, increase the regularization factor causes the poorest results, but Color Name keeps practically equal the accuracy with even a small value of regularization factor.

7. CONCLUSIONS

This paper presents an evaluation among three different descriptors applied on two different Datasets (Soccer and Karinas_Dataset1). Values obtained after image classification have been commented according to the dataset, descriptor and parameters used for the classification. It has been proved that colour descriptors achieved better results on a dataset that is colour dominant (Soccer), and poorer results have been obtained for a complex dataset (Karinas_Dataset1) because of the small amount of images compared with the number of objects that have been tried to identify.

Future Works will include the increase of the size of the Karina Dataset, the use of additional descriptors for image classification and SVM as the way to classify the images of the datasets. Since Karina Dataset proves to be very challenging for the task of image classification, the fusion of the previous descriptors will be tested as a way to obtain better results.

Acknowledgements

This work has been supported by the ASASEC (www.asasec.eu) European Project with reference HOME/2010/ISEC/AG/043 and by the grant DPI2012-36166 from the Spanish Government.

References

[1] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA,", *ProcP of the European Conference on Computer Vision*, 2006.

[2] B. Berlin and P. Kay, "Basic color terms: their universality and evolution", *Berkeley: University of California*, 1969.

[3] D. G. Lowe. "Object recognition from local scale invariant features". *ICCV*, volume 2, pages 1150–1157 vol.2. IEEE Computer Society, August 1999.

[4] D. Vigo, F. Khan, J. van deWeijer, and T. Gevers. "The impact of color on bag-of-words based object recognition". *ICPR*, pages 1549 – 1553, 2010.

[5] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] J. van de Weijer and C. Schmid, "Coloring local feature extraction," *Proc. of the European Conference on Computer Vision*, Graz, Austria, 2006, vol. 2, pp. 334–348.

[7] J. van de Weijer, T. Gevers, and A. Bagdanov, "Boosting Color Saliency in Image Feature Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 150-156, Jan. 2006.

[8] J. van deWeijer and C. Schmid. "Applying color names to image description". *ICIP*, pages 493–496, 2007.

[9] J. van deWeijer and C. Schmid. "Coloring local feature extraction", *ECCV*, volume 3952, pages 334–348. Springer, 2006.

[10] J.M. Geusebroek, "Compact object descriptors from local colour invariant histograms", *British Machine Vision Conference*, 2006.

[11] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. "The devil is in the details: an evaluation of recent feature encoding methods". *BMVC*, 2011

[12] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. "Evaluating color descriptors for object and scene recognition". *PAMI*, 32(9):1582–1596, 2010.

[13] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[14] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 62–86, 2004.

[15] Karinas_Dataset1
http://pitia.unileon.es/varp/galleries

[16] Schmid, C., and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE PAMI*, 19, 5 (1997), pp. 530–534.

[17] Soccer Dataset
http://lear.inrialpes.fr/people/vandeweijer/data.html

Comparison of different Local Binary Pattern variants for material recognition using KTH-TIPS 2a dataset

Oscar García-Olalla, María Teresa García-Ordás , Laura Fernández-Robles, Diego García-Ordás, Enrique Alegre
{ogaro,mgaro,l.fernandez,dgaro,enrique.alegre}@unileon.es
University of León.

Abstract

In this paper, a comparison between several variants of local binary pattern developed in the last years has been carried out for material recognition at digital images. For that, we have used the KTH-tips 2a dataset which contains 11 material images taken at 3 poses, 4 different illumination conditions and 9 scales. For the retrieval classification process, three nearest neighbours classifiers have been assessed using cosine and square distances. The best results were achieved using ALBP with a weighted k-NN method obtaining a 58.02% of hit rate using the Square distance metric.

1 Introduction

Texture analysis is a challenging open problem in computer vision. It refers to a set of processes applied to detect and describe spatial variations of the gray level of all the pixels in an image.

Texture features derived from the gray level co-occurrence matrix (GLCM) by Haralick[12] are one of the global texture features more frequently used. These features represent the probability of occurrence of a pixel pair with a given gray level difference at varying distances. There are numerous works that rely on this method. In [20], Sundaram presented an Iris recognition system based on the Haralick features on the GLCM and Neural Networks, obtaining an accuracy of 97.00%. In the medical field, some works use these techniques to detect colon cancer cells [4] or to describe the mammography images better [13].

Adaptive algorithms have been widely used in the recent years, and applied to solve lots of problems in all the stages of the image analysis. Recently, some works have used adaptive methods to describe images obtaining feature vectors more robust to changes in lighting, scale, noise, etc. One example can be found in [6], where Chen proposed a new algorithm to describe retinal images with intensity invariance. González-Castro et al propose an adaptive Pattern Spectrum for texture classification in [8].

In the last years, local feature descriptors have been used in image analysis with satisfactory results. Local Binary Pattern (LBP) [16] has been repeatedly modified in order to make it adaptive. In [19], Shen et al proposed a new adaptive LBP for 3D Face recognition whereas Li et al introduced a new LBP method invariant to scale and rotation using adaptive textons [14].

There are multiple fields that require texture analysis to classify images. For example, in fabrics, there are a lot of tasks that can be automatized using texture analysis. In [2], Behravan et al proposed a hybrid scheme for online detection and classification of textural fabric defects. Paniagua-Paniagua et al developed a method to outperform the quality detection in the cork industry [18]. In medical image, Alegre et al [1] proposed a method based on texture features to classify sperm heads according to the state of their acrosomes.

Several works have evaluated their descriptors using the KTH-TIPS 2a dataset due to the high challenge that this database implies. In 2013, Maani et al proposed in [15] a new noise robust rotation invariant descriptor testing it with KTH-TIPS 2a dataset among others. In the same year, Zhang et al [22], evaluated a continuous rotation invariant local descriptor method with this dataset. Another example of its use can be found in the work carried out by Zhang et al [21] in which they developed a new local energy pattern for texture classification. Recently, García-Olalla et al [7] demonstrated the good performance of Adaptive Local Binary Pattern with oriented standard deviation information using this dataset.

In this paper we compared some of the most important variants of LBP for material recognition such us Adaptive Local Binary Pattern (ALBP), Complete Local Binary Pattern (CLBP), Local Binary Pattern Variance (LBPV) and the original LBP using KTH-Tips 2a dataset to study their performance with a very challenging and widely used dataset.

The rest of the paper is organized as follows: Section 2 describes the followed methodology. The selected dataset, experiment setup and achieved results with the different descriptors are presented in section 3. Finally, section 4 shows our conclusions.

2 Methodology

2.1 Texture description

2.1.1 Local Binary Pattern

Local Binary Pattern (LBP) [17] is an algorithm used to describe the texture of grayscale images extracting their local spatial structure. Given a pixel, a pattern code is computed by comparing this pixel with the value of its neighbours:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \ , \qquad s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{1}$$

where g_c is the value of the central pixel, g_p is the value of its neighbour p, P are the number of neighbours and R is the radius of the neighbourhood.

After LBP is obtained for each pixel, in this work, a histogram with just $P + 2$ bins is built in order to describe the whole image instead of the typical 256 element histogram, yielding a lower dimensional feature vector of the image. The pattern extraction process for one pixel is shown in figure 1.

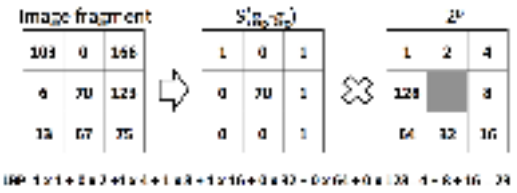


Figure 1: Local Binary Pattern process for one gray scale pixel with P=8 and R=1. LBP code assigned to the central pixel is calculated by multiplying the output of the threshold function by the term 2^p for each neighbour pixel and then summing all those values.

2.1.2 Adaptive Local Binary Pattern

In [9], Guo et al. proposed an adaptive descriptor based on Local Binary Pattern motivated by the lack of information about the orientation in the classical Local Binary Pattern algorithm. The method that they presented takes into account the mean and the standard deviation along different orientations over all the pixels in order to make the matching more robust against local spatial structure changes. To minimize the variations of the mean and standard deviation of the directional differences, Guo et al. proposed a scheme to minimize the directional difference $|g_c - w_p * g_p|$ adding to equation 1 an extra parameter w.

The objective function is defined as follows:

$$w_p = arg_w \min \left\{ \sum_{i=1}^{N} \sum_{j=1}^{M} |g_c(i,j) - w \cdot g_p(i,j)|^2 \right\} \tag{2}$$

where w_p is the weight element used to minimize the directional difference and N and M are the number of rows and columns in the image respectively. Each weight w_p is estimated along one orientation $2p\pi/P$ for the whole image.

Therefore, adding this extra parameter w the ALBP method can be defined as:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - w_p \cdot g_c)2^p \ , \qquad s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{3}$$

2.1.3 Local Binary Pattern Variance

LBPV [11] is a proposal by Guo et al which consists of a combination of LBP and a contrast distribution method. First, the uniform LBP is calculated on the whole image. Then, the variance of the image is used as an adaptive weight to adjust the contribution of the LBP code in the histogram calculation. The LBPV histogram is computed as:

$$LBPV_{P,R}(k) = \sum_{i=1}^{N} \sum_{j=1}^{M} w(LBP_{P,R}(i,j), k), k \in [0, K] \tag{4}$$

where k is each bin of the histogram, K the maximum value of LBP and w is defined in equation 5.

2.1.4 Completed Local Binary Pattern

In [10], Guo et al proposed another method called CLBP to generalize and complete the classical LBP. In that method, a local region is represented by its center pixel and a local difference sign - magnitude transform called LDSMT. LDSMT decomposes the image local structure into two complementary components: the difference signs and the difference magnitudes. In order to code both components, they proposed two operators, CLBP-Sign (CLBP_S) and CLBP-Magnitude (CLBP_M). Since all of them are in binary format, they can be combined to form the final CLBP histogram. CLBP_S is equal to the classical LBP histogram defined in 1, and CLBP_M is defined in the equation 7.

$$CLBP_M_{P,R} = \sum_{p=0}^{P-1} t(m_p, c)2^p \ , \qquad t(x,c) = \begin{cases} 1 & \text{if } x \geq c \\ 0 & \text{if } x < c \end{cases} \tag{7}$$

where c is a threshold determined adaptively.

$$w(LBP_{P,R}(i,j),k) = \begin{cases} VAR_{P,R}(i,j), & LBP_{P,R}(i,j) = k \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $VAR_{P,R}$ is the variance of the neighbourhood defined in equation 6 .

$$VAR_{P,R} = \frac{1}{P}\sum_{p=0}^{P-1}(g_p - u)^2 \ , \quad u = 1/P\sum_{p=0}^{P-1} g_p \quad (6)$$

Finally, CLBP is obtained either concatenating or merging both operators. In our case, we have chosen the concatenation due to its higher similarity with ALBPS method.

2.2 Nearest Neighbours variants

Apart from the standard kNN, two variants of the original kNN have been used in this work, mean distance kNN and weighted kNN. In the figure 2 we can see an example of a nearest neighbour classification using k=5. In the original kNN, the query element would be classified as **X** due to the higher number of elements of that class in the neighbourhood.

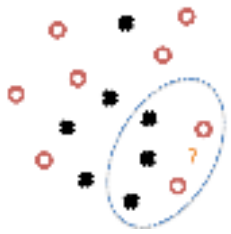


Figure 2: Example of a nearest neighbourhood of k=5 for the query element **?**.

2.2.1 Mean distance kNN

The main idea of this metric is to assign the query sample to the class whose k nearest neighbours are closer to it. To do that, the k nearest neighbours are selected obtaining the mean distance of each class. Finally, we assign the query sample to the class of the set with lower mean distance. In the figure 2 example, using mean distance the query element would be assigned to the **o** class due to the lower mean distance of that class.

2.2.2 Weighted kNN

In weighted kNN, each sample has a weight w_p which is the difference between the farthest sample from the query and the proper distance from the query sample. See equation 8.

$$w_p = |d_{max} - d_p| \quad (8)$$

Each class is weighted adding the weight of each sample of the class. Finally, the query sample belongs to the heaviest class.

3 Experiments

3.1 KTH-tips 2a dataset

Our method has been evaluated with the KTH-tips 2a texture dataset [3] used for material recognition. This dataset is composed by several images for material categorization [3]. It contains 11 materials (lettuce, brown bread, white bread, aluminium, corduroy, cork, cotton, cracker, linen, wood and wool) with a number of images for four different samples from each material. All the samples were taken at 3 poses, 4 different illumination conditions and 9 scales. All these variations make a very challenging dataset. In figure 3 we can see some examples of some textures on different conditions.

3.2 Experimental Setup

The experimental setup used for the KTH-tips 2a dataset follows the standard protocol developed by Caputo et al, used in several works [3, 5]. The method consists of taking one of the samples of each material to test and the rest for training, which conforms a more challenging setup than using random images for test and training. In this work, we carried out four classifications using this method to increase its robustness, one classification for each sample in the test class. The final hit rate is obtained as the mean of the hit rates in each iteration. We show the comparison of the three k-NN approaches shown previously (standard kNN, mean distance kNN and Weighted kNN).

3.3 Experimental results

3.3.1 NN variants evaluation.

Three different classifiers based on the Nearest Neighbours method have been implemented to evaluate the LBPs variants shown in this paper. Two distances have been selected to compare the results: Cosine (see equation 9) and Square distance (see equation 10).

$$cosineDist(a,b) = \frac{a \cdot b}{||a|| \cdot ||b||} \quad (9)$$

$$SqDist(a,b) = ||a-b||^2 \quad (10)$$

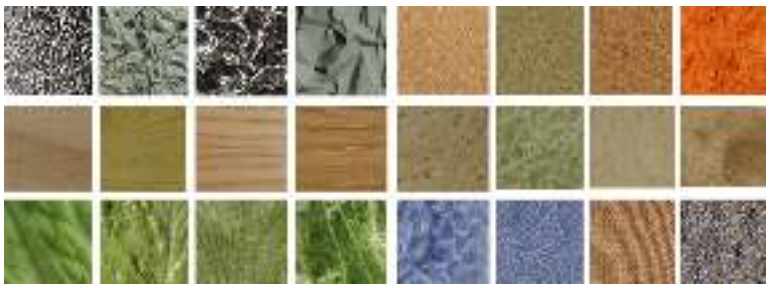


Figure 3: Examples of the KTH-tips 2a dataset.

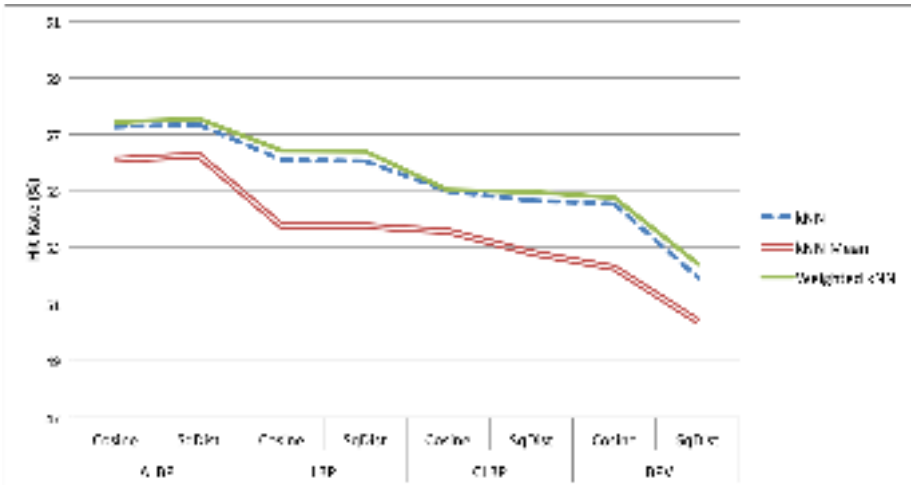


Figure 4: Results achieved on the KTH-tips 2a dataset using different classifiers. The hit rate value corresponds with the mean hit rate along different values of k (1, 3, 5, 7, 9, 11) for all the pairs Descriptor-Distance.

Table 1: Hit rate in % using kNN variants with cosine and Square distances on the KTH tips 2a dataset showing the best K value for each method.

Classifier method		ALBP	LBP	CLBP	LBPV
kNN	Cosine	57.72	56.38	55.41	54.84
	SqDist	58.00	56.59	55.01	52.97
kNN mean	Cosine	57.58	56.04	55.41	54.02
	SqDist	58.00	56.00	55.01	52.97
Weighted kNN	Cosine	57.95	56.57	55.41	55.09
	SqDist	**58.02**	56.92	55.18	52.97

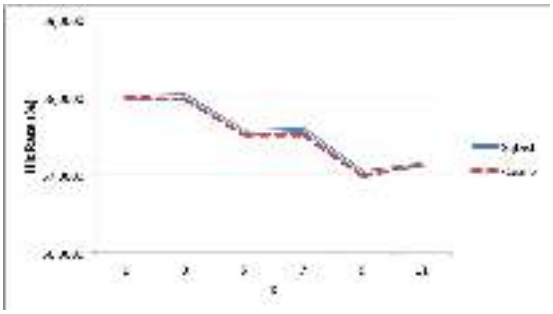


Figure 5: Results achieved on the KTH-tips 2a dataset using weighted kNN, ALBP descriptor and different values of k and distances.

In the figure 4 we can observe the mean hit rate along different values of k (1, 3, 5, 7, 9, 11) of all the descriptors using each classifier with the two distances. As we can see, the best results were achieved using the weighted kNN, which slightly outperforms the original kNN method in all the cases except the cosine CLBP where they have the same performance. Furthermore, showing the graphic, we can determine that the kNN mean shows lower accuracy in the multi class classification process in all the experiments.

3.3.2 LBP variants performance with weighted kNN

Once we have determined the best classifier (Weighted kNN), a comparison between the LBP methods evaluated was carried out. In figure 6, we can see the mean performance along different k values of the descriptors using both distances and the weighted kNN classifier. It can be appreciated that the best performance is 57.55%, achieved using the ALBP proposed by Guo et al, followed by the original LBP. As we can see in the figure, the behaviour of the weighted kNN is very similar using Cosine and Square distances in almost all the experiments. The worst result was achieved using SqDist and LBPV with less than a 53% of hit rate.

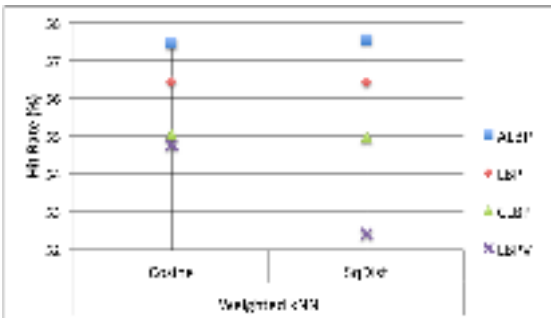


Figure 6: Results achieved on the KTH-tips 2a dataset using different descriptors and the weighted kNN classifier. The hit rate value corresponds with the mean hit rate along different values of k (1, 3, 5, 7, 9, 11) for each distance.

3.3.3 Best k neighbour value

As it has been proven, the best result was achieved using SqDist and the ALBP variant. However, all the results shown previously are obtained as the mean value along different numbers of neighbours. In figure 5, we can see the performance of ALBP for all the k values and all the distances evaluated. The results confirm that the best configuration is the one formed by ALBP and SqDist metric obtaining a 58.02% of hit rate though both distances have very similar performance for these experiments.

In table 1, we can see the results achieved using the three kNN methods for all the distances highlighting the best result.

4 Conclusions

In this paper we have evaluated several recent methods based on Local Binary Pattern for materials recognition. The KTH-TIPS 2a dataset has been used due to the high challenge it implies using eleven materials with four samples each one at different poses, illumination conditions and scales. The classification was carried out using nearest neighbours and two variants: kNN mean distance and weighted kNN. The last one, Weighted kNN, improves the results of the other two methods in all the experiments obtaining a 58.02% of hit rate with ALBP and Square Distance taking three neighbours into account. In conclusion, the ALBP descriptor outperforms LBP in 1.93%, CLBP in 5,15% and LBPV in 9,53% using the best configuration which is the one composed by weighted kNN with Square Distance.

Acknowledgments

This work has been supported by the ASASEC (www.asasec.eu) European Project with reference HOME/2010/ISEC/AG/043 and by grants DPI2012-36166, DPI2009-08424 and via the pre-doctoral FPU fellowship program from the Spanish government.

References

[1] Enrique Alegre, Victor Gonzalez-Castro, Rocio Alaiz-Rodriguez, and Maria Teresa Garcia-Ordas. Texture and moments-based classification of the acrosome integrity of boar spermatozoa images. *Computer Methods and Programs in Biomedicine*, 108(2):873 – 881, 2012.

[2] M. Behravan, R. Boostani, F. Tajeripour, and Z. Azimifar. A hybrid scheme for online detection and classification of textural fabric defects. In *Machine Vision, 2009. ICMV '09. Second International Conference on*, pages 118 –122, dec. 2009.

[3] B. Caputo, E. Hayman, and P Mallikarjuna. Class-specific material categorisation. In *ICCV*, 2005.

[4] A. Chaddad, C. Tanougast, A. Dandache, and A. Bouridane. Extraction of haralick features from segmented texture multispectral bio-images for detection of colon cancer cells. In *Informatics and Computational Intelligence (ICI), 2011 First International Conference on*, pages 55 –59, dec. 2011.

[5] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao. Wld: A robust local image descriptor. In *PAMI*, 2010.

[6] Jian Chen, Jie Tian, N. Lee, Jian Zheng, R.T. Smith, and A.F. Laine. A partial intensity invariant feature descriptor for multimodal retinal image registration. *Biomedical Engineering, IEEE Transactions on*, 57(7):1707 –1718, july 2010.

[7] Oscar Garcia-Olalla, Enrique Alegre, Laura Fernandez-Robles, Maria Teresa Garcia-Ordas, and Diego Garcia-Ordas. Adaptive local binary pattern with oriented standard deviation (albps) for texture classification. *EURASIP Journal on Image and Video Processing*, 2013(1):31, 2013.

[8] V. Gonzalez-Castro, E. Alegre, O. Garcia-Olalla, L. Fernandez-Robles, and M.T. Garcia-Ordas. Adaptive pattern spectrum image description using euclidean and geodesic distance without training for texture classification. *Computer Vision, IET*, 6(6):581–589, 2012.

[9] Zhenhua Guo, Lei Zhang, D. Zhang, and Su Zhang. Rotation invariant texture classification using adaptive lbp with directional statistical features. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 285 –288, sept. 2010.

[10] Zhenhua Guo, Lei Zhang, and David Zhang. A completed modeling of local binary pattern operator for texture classification. *Image Processing, IEEE Transactions on*, 19(6):1657–1663, 2010.

[11] Zhenhua Guo, Lei Zhang, and David Zhang. Rotation invariant texture classification using lbp variance (lbpv) with global matching. *Pattern Recognition*, 43(3):706–719, 2010.

[12] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, November 1973.

[13] K.T.R. Keerthana and K. Thangavel. Feature selection in mammogram image using rough set approach. In *Innovations in Emerging Technology (NCOIET), 2011 National Conference on*, pages 147 –152, feb. 2011.

[14] Zhi Li, Guizhong Liu, Yang Yang, and Junyong You. Scale- and rotation-invariant local binary pattern using scale-adaptive texton and subuniform-based circular shift. *Image Processing, IEEE Transactions on*, 21(4):2130 –2140, april 2012.

[15] Rouzbeh Maani, Sanjay Kalra, and Yee-Hong Yang. Noise robust rotation invariant features for texture classification. *Pattern Recognition*, 46(8):2103 – 2116, 2013.

[16] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR 1994)*, 1994.

[17] Timo Ojala, Matti Pietikainen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

[18] B. Paniagua-Paniagua, M.A. Vega-Rodriguez, P. Bustos-Garcia, J.A. Gomez-Pulido, and J.M. Sanchez-Perez. Advanced texture analysis in cork quality detection. In *Industrial Informatics, 2007 5th IEEE International Conference on*, volume 1, pages 311 –315, june 2007.

[19] Haihong Shen, Qishan Zhang, and Dongkai Yang. Adaptive local binary patterns for 3d face recognition. In *Pattern Recognition, 2009. CCPR 2009. Chinese Conference on*, pages 1 –4, nov. 2009.

[20] R.M. Sundaram and B.C. Dhara. Neural network based iris recognition system using haralick features. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 3, pages 19 –23, april 2011.

[21] Jun Zhang, Jimin Liang, and Heng Zhao. Local energy pattern for texture classification using self-adaptive quantization thresholds. *Image Processing, IEEE Transactions on*, 22(1):31–42, 2013.

[22] Jun Zhang, Heng Zhao, and Jimin Liang. Continuous rotation invariant local descriptors for texton dictionary-based texture classification. *Computer Vision and Image Understanding*, 117(1):56 – 75, 2013.

Rotation Invariant Contour Points Descriptor Histogram for shape based image retrieval

María-Teresa García Ordás, Oscar García-Olalla, Laura Fernández-Robles, Diego García-Ordás, Enrique Alegre
{mgaro, ogaro, l.fernandez, dgaro, e.alegre}@unileon.es
Escuela de ingenierías industrial e informática

Abstract

In this paper, a new method RCPDH (Rotation Invariant Contour Points Descriptor Histogram) based on the original CPDH was proposed in order to carry out an image shape retrieval. Shapes are described based on the contour points description obtaining a feature vector composed of 108 elements. The main advantage of this new descriptor is that it is intrinsically rotational invariant, solving the problems of the original method CPDH and avoiding the use of the EMD (Earth Movers Distance) classifier used by the authors of the original method to solve the problem of the invariance to rotation. The retrieval method was tested using three datasets: Kimia25, Kimia99 and MPEG2 and it was carried out using kNN and two variants: kNN mean distance and weighted kNN with Intersect and Cityblock distances obtaining a 76% of success rate in Kimia25 dataset, 91.91% in Kimia99 and 74% using MPEG2 against the 68%, 91.91% and 74.75% respectively obtained with the original method using Cityblock distance in both cases. Although the results are very similar, the greatest advantage of our retrieval is the computational cost which is clearly better than the original Earth Movers Distance classifier used in the CPDH original method which has a computational cost of n^3.

Palabras clave: Retrieval, Shapes, SVM, RCPDH, Contour, Points

1 Introduction

Shape description and recognition has become an important task in multiple fields because of the proliferation of digital image storage sites, resulting in large amounts of databases. Nowadays, shape description and recognition can be used for many applications. For example, in the medicine field, Hao et al. [13] proposed a method to classify the invertebral disks into healthy and degenerated automatically using an active learning and saving much human effort by avoiding labelling all training data manually while Severa et al. [15] evaluated the variability of the sperm head shape for 10 different stallions using Fourier descriptors and inverse Fourier transform. In the same way, Petkov et al [12] assessed the acrosome state in boar spermatozoa heads using n-contours descriptor and RLVQ and in [2], an automatic classification of the acrosome status of boar spermatozoa using digital image processing and LVQ is carried out. In the biometric field, Gholameza et al. [3] developed a hand-based verification method using Zernike moments to represent the hand geometry with an efficient algorithm that uses arbitrary precision arithmetic, a look-up table, and avoids recomputing the same terms multiple times and Proen et al. in [14] are focused on iris recognition technologies. In current manufacturing industry, shape retrieval is used to optimize the tool life in metal machining and for tool wear monitoring [1] [5]. In the same field, Castejón et al proposed a new method to estimate the wear level in cutting inserts in order to identify the time for its replacement [8].

In a different way, far from the medicine and industrial field, Anuar et al. [4] developed a retrieval system capable of comparing the visual similarity of trademarks using a global descriptor based on Zernike moments and a local descriptor based on the edge-gradient co-occurrence matrix derived from the contour information and a retrieval system based on Bull's eye score. Another example is the work developed by Zagoris et al. [18] in which an MPEG-like descriptor is proposed that contains conventional contour and region shape features with a wide applicability from any arbitrary shape to document retrieval through word spotting using Support Vector Machines.

All these approaches are two dimensional methods, used to classify images but there are also three dimensional image description techniques such as the one developed by Blenkinsopp et al. [7] in which a digital image correlation system to measure dynamic foot shape during running is developed.The main process of their proposed method is to use a set of scanning lines that rotate around the shape centroid to collect information. During the rotating process, three feature functions are computed to reveal the inner structures of the candidate shape. Another example of a three dimensional image description method is the work developed by Smeets [17] in which the geodesic distance matrix is used as an isometry-invariant shape representation. They described two approaches to arrive at a sampling order invariant shape descriptor: the histogram of geodesic distance matrix values and the set

of largest singular values of the geodesic distance matrix and finally, the shape matching is performed by comparison of the shape descriptors using the x^2 distance as dissimilarity measure.

As we can see, shape retrieval is widely used for multiple purposes and in the last years, a lot of new methods have been developed. For example, in [9], a new method for describing and classifying images using kernel based optimization is carried out and in [10] a new multiscale Fourier-based object description in 2-D space using a low-pass Gaussian filter (LPGF) and a high-pass Gaussian filter (HPGF), separately has been developed. The LPGF is used at different scales (standard deviation) representing the inner and central part of an object more than the boundary. On the other hand using the HPGF at different scales represents the boundary and exterior parts of an object. Another novel approach is the one proposed by Ebrahim et al. [11] to shape representation and description in which the object shape is captured by scanning the object image using a space-filling curve. The resulting vector is smoothed, using wavelet approximation, and sampled.

Our descriptor, RCPDH is based on the one developed by Shu et al. [16] which is a shape contour descriptor for shape matching and retrieval. They used the EMD (Earth Movers Distance) metric as matching method but in our case, original kNN matching and two variants with Intersect and Cityblock distances are used saving a lot of time in the retrieval process.

The rest of the paper is organised as follows. In section 2, CPDH original method and our contribution are described. Used classifiers are described in section 3. Datasets and results are shown in section 4 and finally, in section 5, our conclusions are shown.

2 Methods

2.1 CPDH method

The original method, Contour Points Distribution Histogram (CPDH), was proposed by Shu et al. [16]. The first step is to obtain the points representing the contour of the image under polar coordinates. The contour points are used to describe the image and they can be obtained by different methods. In this case, they used the standard Canny operator to detect the object boundary (see [6]). Once the points are extracted, the centroid is set as the origin. After that, they construct the minimum circumscribed circumference and the region defined by that circumference is divided into several bins using some concentric circumferences and equal interval angles. The final step is to construct the CPDH descriptor taking into account the number of points belonging to each division, and the angle and radius of each division.

2.2 RCPDH method

In our case, we used the method proposed by Belongie et al. [6], obtaining a contour like the one shown in figure 1.

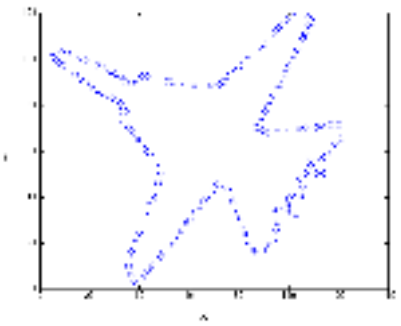


Figure 1: Contour points extraction.

This contour can be represented as a points collection in this way:

$$\begin{aligned} P =& \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} \\ & /(x_i, y_i) \in \mathbb{R}^2 \end{aligned} \tag{1}$$

where n denotes the number of points on the contour. After that, the centroid of the previous points is obtained using the expression 2 in order to obtain the minimum shape circumscribed circumference.

$$(x_c, y_c) = \frac{\sum_{i=1}^{n}(x_i, y_i)}{n} \tag{2}$$

The centroid is set as the origin and the points collection, P, is translated into polar coordinates using ec. 3

$$P = \{(\rho_1, \alpha_1), (\rho_2, \alpha_2), ..., (\rho_n, \alpha_n)\} \in \mathbb{R}^2 \tag{3}$$

where

$$\rho_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{4}$$

and

$$\alpha_i = \begin{cases} \arctan(\frac{(y_i - y_c)}{(x_i - x_c)}) & if x > 0,\ y \geq 0 \\ \arctan(\frac{(y_i - y_c)}{(x_i - x_c)}) + 2\pi & if x > 0,\ y < 0 \\ \arctan(\frac{(y_i - y_c)}{(x_i - x_c)}) + \pi & if x < 0 \\ \arctan(\frac{\pi}{2}) & if x = 0,\ y > 0 \\ \arctan(\frac{3\pi}{2}) & if x = 0,\ y < 0 \end{cases} \tag{5}$$

so the circumscribed circumference has as centre the centroid C (x_c, y_c) and radius $\rho_{max} = max\rho_i, i = 1, 2, ..., n$. The result of this process is shown in figure 2.

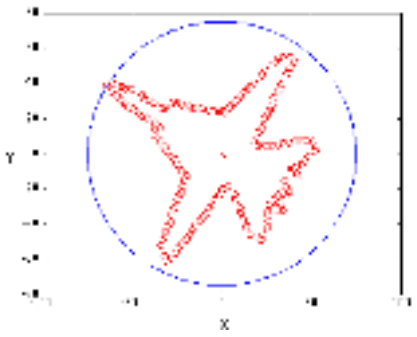


Figure 2: Circumscribed circle.

Then, the circle defined by the circumscribed circumference is divided into several bins partitioning its area into $u \times v$ bins being u the parts in which ρmax is divided and v the sectors in which the circle is divided. The result of this process is shown in figure 3.

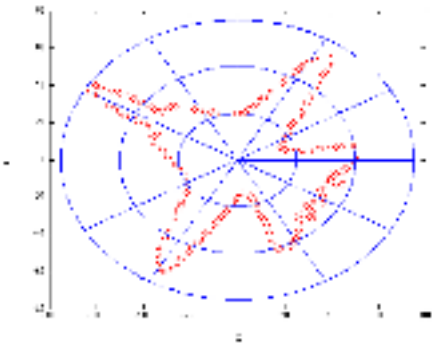


Figure 3: Circle partitioned in 36 bins.

After these previous steps, we can build the CPDH descriptor by counting the number of points belonging to each bin. The CPDH is composed by a triplet for each bin $H_i = (\rho_i, \theta_i, n_i)$ where ρ_i denotes the radius of the concentric circumferences, θ_i denotes the angle space and n_i denotes the number of points located in the bin r_i.

The authors of CPDH, used the EMD distance in order to make their descriptor invariant to rotation so our motivation to develop RCPDH was the necessity of making the descriptor invariant to rotation so that we could classify it using our retrieval system and saving a lot of time, because the main disadvantage of EMD classifier is that it is a really time consuming process.

Basically, Rotational Invariant Contour Points Distribution Histogram (RCPDH) consists of making the CPDH descriptor intrinsically invariant and avoid the use of EMD distance because of its time cost.

Once we have obtained the descriptor CPDH original as is described in the previous section using equations 2 and 3, the third element of each triplet (number of points in the bin) is taking into account to construct the minimum module sequence. This small sequence is called CPDH36. See figure 4.



Figure 4: First step to construct RCPDH: CPDH36 extraction.

Then, we search for the minimum module sequence in the CPDH36 descriptor shifting the vector along all the 36 positions following the expression 6. See figure 5.

$$RCPDH = cat(CPDH(\rho_i, \theta_i, n_i : \rho_{u\times v}, \theta_{u\times v}, n_{u\times v}), CPDH(\rho_1, \theta_1, n_1 : \rho_{i-1}, \theta_{i-1}, n_{i-1})) \quad (6)$$

where i satisfies equation 7 and $cat(a, b)$ is the concatenate operator between a and b.

$$arg_i \min\{cat(CPDH36(i : u \times v), CPDH36(1 : i-1))\forall i \in (1..u \times v)\} \quad (7)$$

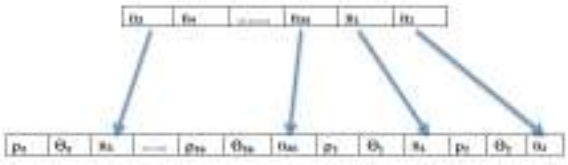


Figure 5: RCPDH descriptor.

Using this new descriptor, it is not necessary the use of a classifier to resolve the rotation invariance problem so we can use several classifiers solving the problem of the cost of time derived from EMD method. As shown in section 3, this proposal combined with kNN outperforms the CPDH original descriptor with EMD.

3 kNN classifiers

Three different kNN classifiers were used. Fistly, original kNN. Secondly, mean distance kNN in which the average distances between the query and the samples are calculated and the class is assigned taking into account these distances. And finally, weighted kNN in which we want to weight the nearest neighbours taking into account its assigned weight.

3.1 Mean distance kNN

The main idea of using mean distance kNN is to assign the query sample to the class whose mean distance to the query sample is smaller, instead of assigning it to the most represented class. In figure 6, we can see that although the most represented class is the X class, three samples against two, the query sample belongs to '+' class because the mean distance is smaller.

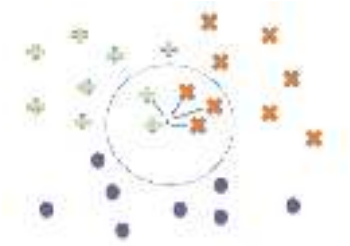


Figure 6: kNN mean distance example taking k=5.

3.2 Weighted kNN

In weighted kNN, a weight is assigned to each sample. This weight w_i is the difference between the farthest sample from the query and the proper distance from the query sample. See equation 8

$$w_i = |d_{max} - d_i| \tag{8}$$

Each class is weighted adding the weight of each sample of the class. Finally, the query sample belongs to the heavier class.

4 Experiments and results

4.1 Datasets

Three datasets were used to test our descriptor. Kimia25 and Kimia99 datasets, and a subset of MPEG7 dataset composed of 400 images from MPEG7 called MPEG2. Kimia25 is composed of 6 classes. Each of them contains four shapes except the "hand class" which is composed of 5 samples. Kimia99 is composed of 9 classes, each one containing 11 shapes. Both Kimia25 and Kimia99 datasets are shown in figure 7. MPEG7 contains 70 classes with 20 samples for each of them. MPEG2 is an MPEG7 subset composed for 20 of the classes of MPEG7 dataset. Some examples of the shapes contained in MPEG2 dataset are shown in figure 8.

4.2 Results

Our proposal was classified using kNN, kNN Mean Distance and Weigthed kNN with Intersect and Cityblock distances and they were tested on three datasets: Kimia25, Kimia99 and MPEG2. Classification results

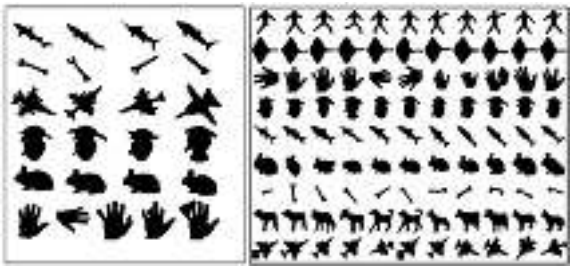


Figure 7: a)Kimia25 database composed by 25 samples divided in 6 classes. b)Kimia99 database composed by 9 classes with 11 samples per class.

for each dataset using the original kNN are shown in tables 1 and 2. Only for Kimia25 dataset better results were achieved using EMD, certainly because of the small number of samples (see table 1). In this case, a 90% of success rate was obtained against our 76% with k=1 and 80% with k=9 (see figure 9), but with the disadvantage of the time consumption. In table 1, we can see that results are better testing the descriptors against Kimia99 dataset than against kimia25 again due to the small number of samples in the former and they are slightly better with our proposal RCPDH than with the original CPDH. In table 2 we can see that with MPEG2 dataset, once again, our proposal RCPDH outperforms the other when using EMD. For this dataset, our method performs better when classifying with intersect distance and achieves a 69.75% of success rate but it is slightly worse using cityblock distance. Although on some datasets the original CPDH with EMD offered better results than our proposal, RCPDH has obtained a significant improvement of the processing time.

Table 2: MPEG2 classification using CPDH original and RCPDH method.

	MPEG7	
Classifier	CPDH	RCPDH
kNN Intersect	67.50%	69.75%
kNN Cityblock	**74.75%**	**74%**
EMD	44.25%	-

In figure 9 we can see the results with the other two classifiers, kNN mean distance and weighted kNN, for different values of k and Cityblock distances. As it is shown, using original kNN, the retrieval hit rate is decreasing for higher values of k in all cases. This problem is solved by using kNN mean distance or weighted kNN in which the hit rate variations are much smoother than using the original kNN. In figure 9, using Kimia25 dataset, we can observe that the original kNN performance is really bad for high values of k and in all datasets, it is shown that the most stable results are achieved using kNN mean distance.



Figure 8: Example of 4 classes with 20 samples of MPEG2 dataset.

Table 1: Kimia25 and Kimia99 classification using CPDH original and RCPDH method.

	Kimia25		Kimia99	
Classifier	CPDH	RCPDH	CPDH	RCPDH
kNN Intersect	60%	64%	83.83%	89.89%
kNN Cityblock	68%	**76%**	**91.91%**	**91.91%**
EMD	**90%**	-	86.41%	-

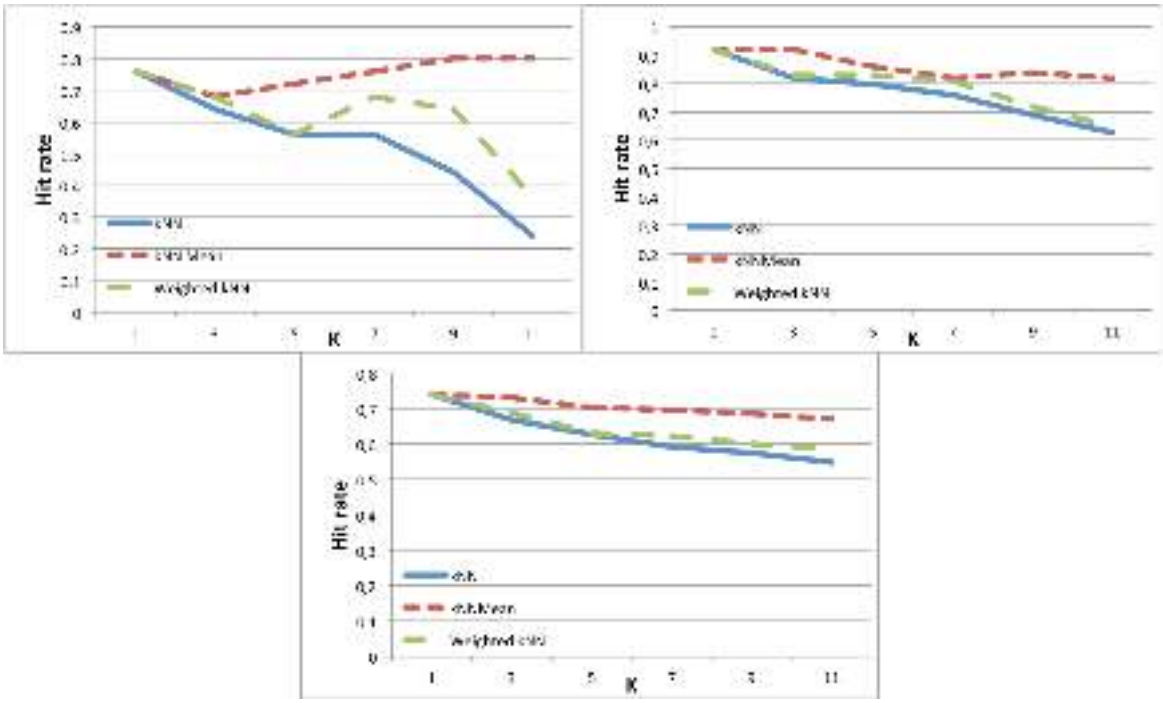


Figure 9: Hit rate of the different kNN methods applied on the RCPDH descriptor using Kimia25 dataset (upper-left), Kimia99 dataset (upper-right) and MPEG2 (down).

5 Conclusions

In this paper, a new method was proposed in order to retrieve shape images. The original method, CPDH descriptor, was classified using EMD distance to achieve invariance to rotation becoming a very slow and time consuming method. Our descriptor can be classified using kNN with classical metrics such as Intersect and Cityblock distances obtaining results within a few seconds because we made it rotational invariant. The tests carried out using the original CPDH method together with the classical metrics offered sometimes worse results than our proposal RCPDH. Our proposal was tested using 3 datasets: Kimia25, Kimia99 and MPEG20 obtaining really good results above all according to the computational cost which clearly improves the original Earth Movers Distance classifier used in the CPDH original method which has a complexity of n^3 against the n^2 complexity of kNN leave one out.

Acknowledgement

This work has been supported by the ASASEC (www.asasec.eu) European Project with reference HOME/2010/ISEC/AG/043 and by grants DPI2012-36166, DPI2009-08424 and via the pre-doctoral FPU fellowship program from the Spanish government.

References

[1] Enrique Alegre, Rocío Alaiz-Rodríguez, Joaquín Barreiro, and Jonatan Ruiz. Use of contour signatures and classification methods to optimize the tool life in metal machining. *Estonian J. Eng*, 15:3–12, 2009.

[2] Enrique Alegre, Michael Biehl, Nicolai Petkov, and Lidia Sánchez. Automatic classification of the acrosome status of boar spermatozoa using digital image processing and lvq. *Computers in Biology and Medicine*, 38(4):461–468, 2008.

[3] Gholamreza Amayeh, George Bebis, Ali Erol, and Mircea Nicolescu. Peg-free hand shape verification using high order zernike moments. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 40, 2006.

[4] Fatahiyah Mohd Anuar, Rossitza Setchi, and Yu kun Lai. Trademark image retrieval using an integrated shape descriptor. *Expert Systems with Applications*, 40(1):105 – 121, 2013.

[5] J Barreiro, M Castejón, E Alegre, and LK Hernández. Use of descriptors based on moments from digital images for tool wear monitoring. *International Journal of Machine Tools and Manufacture*, 48(9):1005–1013, 2008.

[6] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2001.

[7] Robert Blenkinsopp, Andy Harland, Dan Price, Tim Lucas, and Jonathan Roberts. A method to measure dynamic dorsal foot surface shape and deformation during linear running using digital image correlation. *Procedia Engineering*, 34(0):266 – 271, 2012. ENGINEERING OF SPORT CONFERENCE 2012.

[8] M Castejón, E Alegre, J Barreiro, and LK Hernández. On-line tool wear monitoring using geometric descriptors from digital images. *International Journal of Machine Tools and Manufacture*, 47(12-13):1847–1853, 2007.

[9] J.J. Corso and G.D. Hager. Image description with features that summarize. *Computer Vision and Image Understanding*, 113(4):446 – 458, 2009.

[10] Cem DirekoAYlu and Mark S. Nixon. Shape classification via image-based multiscale description. *Pattern Recognition*, 44(9):2134 – 2146, 2011.

[11] Yasser Ebrahim, Maher Ahmed, Wegdan Abdelsalam, and Siu-Cheung Chau. Shape representation and description using the hilbert curve. *Pattern Recognition Letters*, 30(4):348 – 358, 2009.

[12] Nicolai Petkov Lidia Sanchez. Enrique Alegre, Michael Biehl. Assessment of acrosome state in boar spermatozoa heads using n-contours descriptor and rlvq. *Computer Methods and Programs in Biomedicine*, 2013.

[13] Shijie Hao, Jianguo Jiang, Yanrong Guo, and Hong Li. Active learning based intervertebral disk classification combining shape and texture similarities. *Neurocomputing*, 101(0):252 – 257, 2013.

[14] Hugo Proen and Gil Santos. Fusing color and shape descriptors in the recognition of degraded iris images acquired at visible wavelengths. *Computer Vision and Image Understanding*, 116(2):167 – 178, 2012.

[15] L. Severa, L. Machal, L. A Svabova, and O. Mamica. Evaluation of shape variability of stallion sperm heads by means of image analysis and fourier descriptors. *Animal Reproduction Science*, 119:50 – 55, 2010.

[16] Xin Shu and Xiao-Jun Wu. A novel contour descriptor for 2d shape matching and its application to image retrieval. *Image and Vision Computing*, 29(4):286 – 294, 2011.

[17] Dirk Smeets, Jeroen Hermans, Dirk Vandermeulen, and Paul Suetens. Isometric deformation invariant 3d shape recognition. *Pattern Recognition*, 45(7):2817 – 2831, 2012.

[18] Konstantinos Zagoris, Kavallieratou Ergina, and Nikos Papamarkos. Image retrieval systems based on compact shape descriptor and relevance feedback information. *Journal of Visual Communication and Image Representation*, 22(5):378 – 390, 2011.

ROBUSTNESS TO ROTATION IN PERCEPTUAL HASHING METHODS VIA DOMINANT ORIENTATION

Diego García-Ordás, Enrique Alegre, María Teresa García-Ordás,
Oscar García-Olalla, Laura Fernández

{dgaro,enrique.alegre,magro,ogaro,l.fernandez}@unileon.es

Abstract

Lots of pictures are taken on vertical or landscape position and then they are rotated by software procedures, therefore a classification technique to detect this kind of manipulation could be very useful. There are several perceptual hashing techniques robust against rotations that can deal with this specific problem but perform poor with other kinds of tampering. In this paper, we introduce a method to confer robustness against rotations to a number of perceptual hashing algorithms, only by adding a preprocessing step. The proposed method calculates the dominant orientation of the image along four different positions and rotates it to the principal orientation before the hash extraction. It was tested using four common perceptual hashing methods in 300 *images. The dominant orientation was obtained correctly for the 100% of the images, adding in this way rotation robustness to perceptual hashes designed to deal with other specific kinds of attacks.*

Keywords: Perceptual, Robust, Hashing, Rotation, Invariant

1 INTRODUCTION

Nowadays, it is common to find the same image on the internet on different sources, with different formats, qualities, compressions, orientations and even filtered in different ways, but it is still the same image from the human perceptual system. It is difficult for a machine to determine whether or not two images are considered the same for a human and for this purpose, perceptual hashing techniques are commonly used.

While cryptographic hashes are useful to perform *content integrity verification*, they are useless for the previous problem because a little change in one bit changes completely the resulting hash. On the contrary, perceptual hashes offer robustness against several types of manipulations or attacks that alter the image with minor or imperceptible changes for the human perceptual system.

This work is very important for many applications, for example, for the ASASEC (Advisory System Against Sexual Exploitation of Children) European Project. This project needs a perceptual hashing solution so the Police can automatically classify new image and video evidences of child sexual exploitation. This kind of pictures suffer lots of modifications to avoid traceability: face obfuscation, watermarking, rotations, blur, border adding, etc. Therefore, a robust method against all this kind of attacks is needed.

There are lots of different perceptual hashing methods in the literature. In [5] a robust method against rotations using Gabor filters is proposed while in [10] a robust method against compressions and resized is developed. Other authors like Zuo et al. [12], Visha et al. [7] or Brenden et al. [2] propose methods robust against compressions, resizes and rotations respectively. There are also perceptual hashing approaches capable of tampering detection and some others for more specific problems like the one developed in [8] that is applied to biometric face recognition or the trainable spectral hashing method proposed by Weiss et al. in [9].

Recently, some methods that deal with rotation robustness for perceptual hashing have been published. In [4] Lai et al. used Hough transform achieving robustness to severe linear transformations. On other line, shape context and local features points such as SIFT-Harris detector were applied in Lv et al. work [6] to yield better identification performances under geometric attacks.

Obviously, there is no a universal hashing method robust against all kinds of attacks. We can, however, obtain final hashes robust against all these attacks just by applying different methods, each one robust against one or several types of attacks, and then concatenating their results, but that will also enlarge the final hash string. Another solution, which we decided to pursue, is the inclusion of preprocessing steps that modify the image according to our needs before hashing calculation.

The rest of the paper is organized as follows. In the next section we will describe our preprocessing step solution, followed by a description of the

tested perceptual hashing methods in section 3 and our testing datasets in section 4. After that, the experiment results are presented in section 5. In section 6, our conclusions are presented.

2 PROPOSED PREPROCESISING STEP

In this work, we propose a preprocessing method that confers robustness against rotation to a hashing extraction process. This method consists of locating the image at its dominant orientation before extracting the hash. In this work, this process is just described for rotations of 0º, 90º, 180º and 270º, fulfilling the needs of our project. Nonetheless, it can be easily extended to different rotation intervals at the expenses of greater computational cost.

The first thing we want to find is the dominant orientation of an image I so the image can be rotated into its dominant orientation. To do so, the image is resized to a fixed size and rotated in different θ angles:

$$I_\theta = rot(I, \theta), \forall \theta \in \{0, 90, 180, 270\} \quad (1)$$

Now, for each I_θ, its orientation weight W_θ is obtained:

$$W_\theta = \sum_{i=1}^{nRows} I_\theta(i, C) \quad (2)$$

where $nRows$ is the number of rows in the matrix I and C is a constant whose value is restricted to the following range:

$$C \in \mathbb{N} \ / \ C \subset [1, floor(\frac{nColumns}{2})] \cup (ceiling(\frac{nColumns}{2}), nColumns] \quad (3)$$

where $nColumns$ is the number of columns in the matrix I. This restriction over C is applied to avoid selecting a column that will be invariant over rotations, for example the middle column of a matrix A is the same column as the middle column of the matrix $A_{180} = rot(A, 180)$ (despite the reverse order).

Let $\{W_\theta\}$ be the vector composed of all the W_θ,

$$W_{dom} = Max(\{W_\theta\}) \quad (4)$$

$$\theta_{dom} = \theta \ / \ W_\theta = W_{dom} \quad (5)$$

$$I_{dom} = rot(I, \theta_{dom}) \quad (6)$$

where W_{dom} is the dominant weight, θ_{dom} is the dominant angle and I_{dom} is the resulting image placed at its dominant orientation.

After this process, a crop over the resulting image is applied to avoid possible black/white borders created by the rotation function. This step can be omitted when rotating by steps of 90º because the resulting image will always have the same number of pixels than the original one, but it can be necessary when rotating by different angles where new blank content is created near the corners of the image.

After this preprocessing step, the image becomes positioned at its dominant orientation and now we can proceed to extract its robust hash using our chosen perceptual hashing method.

3 EVALUATED PERCEPTUAL HASHING METHODS

For this evaluation, we used three different algorithms included in the free software Perceptual Hashing library "pHash"[1] and one more basic hashing method.

This basic method is the simplest perceptual hashing algorithm. It extracts the lowest frequencies of an image, containing its basic structure. This algorithm proceeds as follows: First, it scales the image to 8x8 pixels, transforms it to grayscale, calculates the mean gray value on the grayscale image and uses this value as a threshold to binarize the image. The 64 bit hash is obtained through the binarized rows of the image.

The second method evaluated was the Discrete Cosine Transform [1]. This is well-known method to obtain perceptual hashes [11] provided from pHash library. It directly extracts the hash from frequency spectrum data using the Fast Fourier Transform.

The third tested method was Marr-Hildreth (MH) hashing method that uses Marr-Hildreth operator to detect edges in the image using the Laplacian of Gaussian.

The last method tested in this work is the radial hashing method [3], based on Radon Transform.

Despite pHash benchmark offers more methods to obtain different types of perceptual hashes, we selected these four due to the fixed length of their resulting hash strings. Therefore, their Hamming distances can be normalized providing accurate comparison results. These four baseline al-

[1] http://www.phash.org



Figure 1: Example of the used dataset. In the first column we can see the original image. In the second the frame addition attack. The flip attack appears in the third column. In fourth column the rotation 90º attack is shown and the last column shows the rotation 10º and crop attacked images.

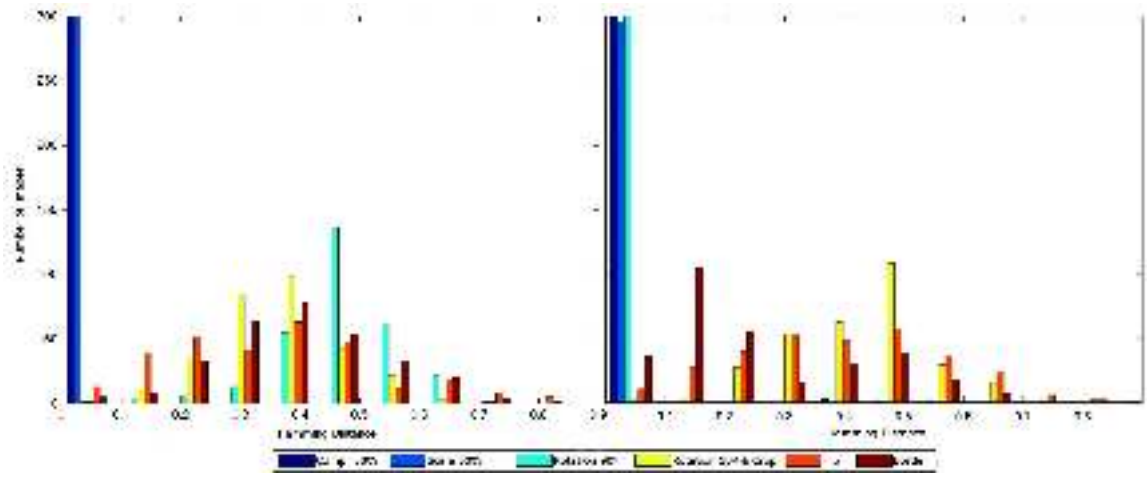


Figure 2: Number of modified images at each normalized Hamming distance using **Basic** method over different modifications. Left: original method. Right: Using proposed preprocessing step. Image best viewed in color.

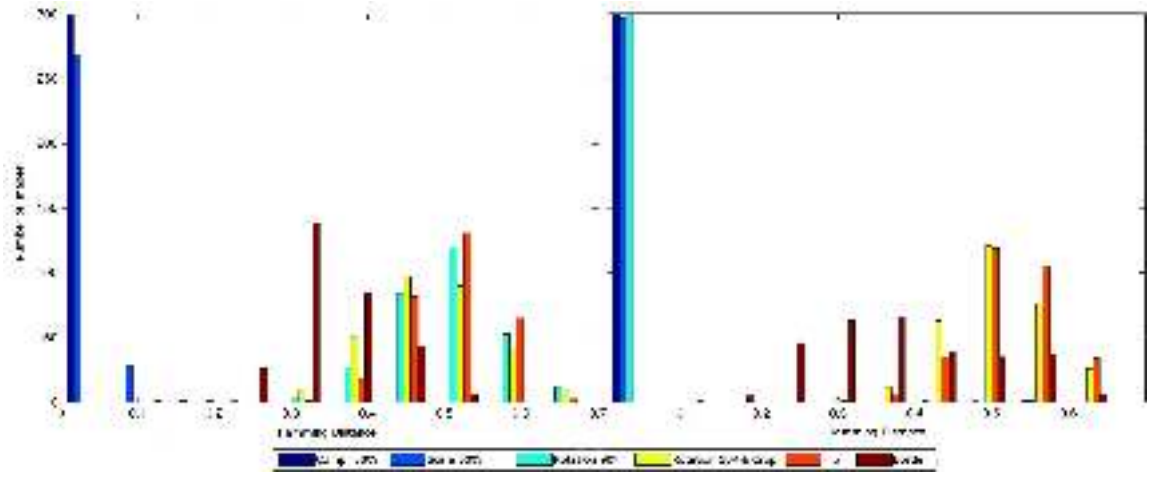


Figure 3: Number of modified images at each normalized Hamming distance using **DCT** method over different modifications. Left: original method. Right: Using proposed preprocessing step. Image best viewed in color.

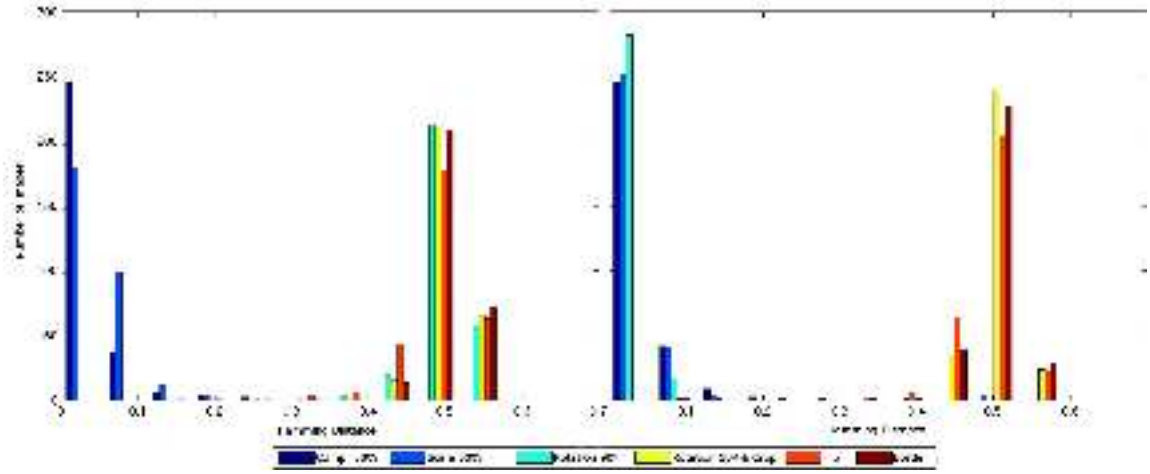


Figure 4: Number of modified images at each normalized Hamming distance using **MH** method over different modifications. Left: original method. Right: Using proposed preprocessing step. Image best viewed in color.

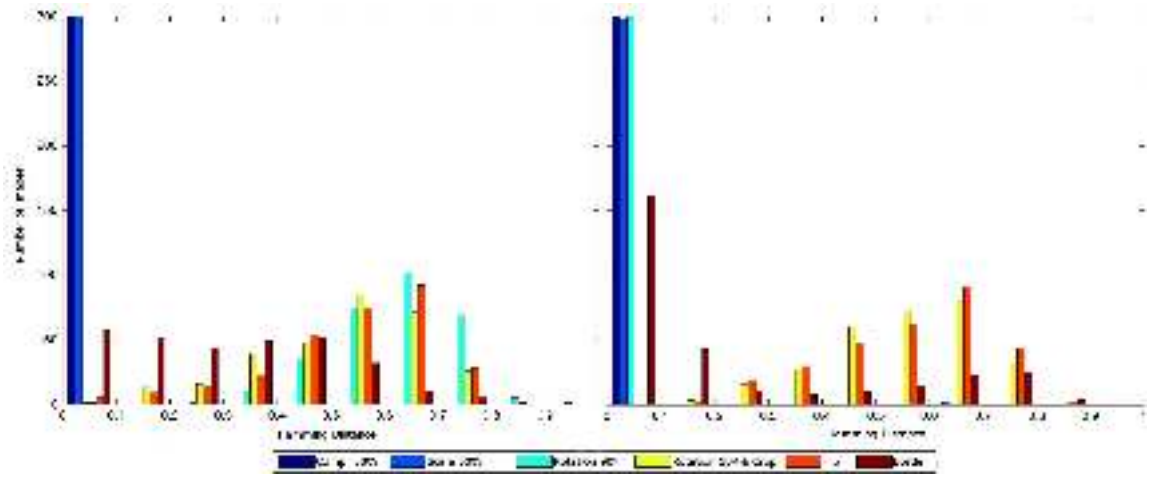


Figure 5: Number of modified images at each normalized Hamming distance using **Radial** method over different modifications. Left: original method. Right: Using proposed preprocessing step. Image best viewed in color.

gorithms were tested in our dataset to obtain a control group, and then they were tested again adding our proposed preprocessing step in order to determine the improvement.

4 DATASETS

The dataset used for this evaluation and the MatLab code can be downloaded from VARP web site[2] and it contains a collection of 300 images extracted from different sources like Wikimedia and Flickr, containing people, animals, objects and landscapes. An automatic software generated 6 attacks for each image: frame addition, compression at 50%, resize at 50%, flip, rotation 10º and rotation 90º, resulting in a final dataset of 1800 images.

An example of these images can be shown in the figure 1.

5 RESULTS

We can observe in this section, four figures comparing results for four different perceptual hashing methods (Basic in Fig. 2, DCT in Fig. 3, MH in Fig. 4 and Radial in Fig. 5) against the different kinds of attacks in our testing dataset, each of them with and without the proposed preprocessing step.

To obtain these results, we measured the normalized Hamming distances between the hashes of 300 original images and six different attacks over each of them. Given that an image and the result of an attack over it are two similar images (despite of the manipulation), a good perceptual hash should be able to reflect this similarity, resulting in a Hamming distance near zero between their two hashes.

As we can see in the graphs, these four original hashing methods (left graphs) offer great results against 'compression 50%' and 'scale 50%' manipulations (dark blue and middle blue bars) since almost the 300 images appear within a distance between 0 and 0.1. Nevertheless, the 'rotation 90º' manipulation (light blue bar) shows a pretty big distance around 0.5 in most of the cases. When using the proposed preprocessing step (right graphs) the performance against the '90º rotation' modification is highly improved compared with their original counterparts. In this case, most of the differences are in the range form 0 to 0.1. Besides, due to the crop carried out during the preprocessing step, we can also observe a general improvement on 'Scale 50%' with 'DCT' and 'MH' methods and a little improvement on 'Border' with 'Radial', 'DCT' and 'Basic'. However, slightly worse results can be observed in 10º rotation and flop attacks.

[2] http://pitia.unileon.es/varp

6 CONCLUSIONS

Perceptual hashes are usually robust against several kinds of attacks, but they behave better against some attacks than against others. Robustness against different attacks can be conferred to a hashing method just by processing the image in a previous step to the hash extraction. Our proposed method gave robustness against rotations with angles of 90º to four common hashing techniques, increasing their effectiveness against this attack in about 100% in most of the cases. As future work, this preprocessing method can be extended to rotations with angles along 360º and flip invariance. Regarding these results, we can conclude that besides improving methods, we can also improve data according to our needs.

Acknowledgements

This work has been supported by the ASASEC (www.asasec.eu) European Project with reference HOME/2010/ISEC/AG/043, by grants DPI2012-36166 and via the pre-doctoral FPU Spanish Government fellowship program.

References

[1] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, C-23(1):90 –93, jan. 1974.

[2] B. Chen and V. Chandran. Robust image hashing using higher order spectral features. In *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, pages 100 –104, dec. 2010.

[3] C. De Roover, C. De Vleeschouwer, F. Lefebvre, and B. Macq. Robust image hashing based on radial variance of pixels. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III – 77–80, sept. 2005.

[4] Shuhua Lai and Nian Zhang. Image hashing via hough transform. In *CISS*, pages 1–4, 2012.

[5] Yuenan Li, Zheming Lu, Ce Zhu, and Xiamu Niu. Robust image hashing based on random gabor filtering and dithered lattice vector quantization. *Image Processing, IEEE Transactions on*, 21(4):1963 –1980, april 2012.

[6] Xudong Lv and Z. Jane Wang. Perceptual image hashing based on shape contexts and local feature points. *IEEE Transactions on Information Forensics and Security*, 7(3):1081–1093, 2012.

[7] V. Monga and M.K. Mhcak. Robust and secure image hashing via non-negative matrix factorizations. *Information Forensics and Security, IEEE Transactions on*, 2(3):376 –390, sept. 2007.

[8] K. Senel, M.K. Mihç andak, and V. Monga. A learning framework for robust hashing of face images. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 197 –200, sept. 2010.

[9] Y. Weiss, A. Torralba, and R. Fergus. *Spectral hashing*, volume 21. NIPS, 2008.

[10] Longjiang Yu and Shenghe Sun. Image robust hashing based on dct sign. In *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP '06. International Conference on*, pages 131 –134, dec. 2006.

[11] Longjiang Yu and Shenghe Sun. Image robust hashing based on dct sign. In *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP '06. International Conference on*, pages 131 –134, dec. 2006.

[12] Jinglong Zuo and Delong Cui. Retrieval oriented robust image hashing. In *Industrial Mechatronics and Automation, 2009. ICIMA 2009. International Conference on*, pages 379 –381, may 2009.

COMBINING SURF WITH GLOBAL TEXTURE DESCRIPTORS FOR CLASSIFYING BOAR SPERM

Fernández-Robles[1], L., Alegre[2], E., García-Ordás[3], M.T., García-Olalla[4], O., García-Ordás[5] D. and Fidalgo Fernández[6] E.
1 Department of Mechanical, IT and Aerospace Engineering
2,3,4,5,6 Department of Electrical, Systems and Automatics Engineering
Campus de Vegazana S/N 24071, University of León, León, Spain.
{l.fernandez, enrique.alegre, mgaro, ogaro, dgaro}@unileon.es, {eduardo.fidalgo.fernandez@gmail.com}

Abstract

ABSTRACT. In this work we propose a new image descriptor combining Invariant Local Features (ILF) with global texture descriptors for the classification of boar sperm acrosomes as intact or damaged. We use SIFT (Scale Invariant Feature Transform) and SURF (Speeded-Up Robust Features) as ILF and Haralick features extracted from the original image, Laws masks, Legendre and Zernike moments as global texture descriptors. A k-Nearest Neighbours (kNN) based algorithm, Support Vector Machine (SVM), Neural Networks (NN) and Bag Of Words (BOW) are used in the classification in order to optimize and compare our results. Our method outperforms not only pure ILF and pure global texture approaches but also previous works. We achieve an accuracy of 95.56% with the combination of SURF and Legendre descriptors classified through a kNN-based algorithm, leading to a successful method for the classification of the state of boar sperm cells in the artificial reproduction industry.

Key Words: ILF, SIFT, SURF, Laws masks, Legendre and Zernike moments, kNN, SVM, NN, BOW.

1 INTRODUCTION

Estimation of the fertilization potential of a sperm sample for artificial insemination is a key factor in medical and veterinarian communities. One of the most important fields is the porcine industry, where it is applied with the purpose of obtaining better individuals for human consumption.

In the last years, the Computer-Assisted Semen Analysis (CASA) systems have been applied to the assessment of the seminal quality [6]. However, three criteria used by veterinarian experts cannot be measured automatically with these systems. Those are the vitality of the sample based on the presence of dead or alive spermatozoa, the integrity of the acrosome membrane and the number and presence of proximal and distal droplets. In this work, we will deal with the integrity of the acrosome membrane.

Currently, the evaluation of acrosome integrity of spermatozoa's heads is carried out manually, using stains. This manual assessment has several disadvantages such as its lack of objectivity, its high cost in terms of time or the requirement of specialised veterinarian staff and equipment. Hence, it would be very interesting to use an automatic classification of the acrosomes as intact or damaged.

There are few computer vision works that deal with boar sperm analysis in the field of texture analysis and classification. These approaches should reliably segment the heads of the spermatozoa, extract the patterns that characterise them and finally classify those patterns in order to estimate how many damaged acrosomes are present in the sample. In this way, a statistic texture description to classify boar sperm images applying discriminant analysis was carried out by Suárez et al. in [21]. The best results were achieved using QDA classifier with 86.19% of hit rate. In [12], González et al. used a discrete wavelet Transform (DWT) and several texture descriptors to classify the acrosome integrity achieving a hit rate of 92.19%. Alegre et al. [2] used Learning Vector Quantization (LVQ) with four prototypes to classify the acrosome integrity obtaining a hit rate of 93.2%. The list of works related with texture analyzed can be completed with [9, 10, 17, 18]

Furthermore, Enrique et al. [3] proposed a method to characterise acrosomes by means of texture features. The texture were described using first order statistics and features derived from the co-occurrence matrix of the image, both computed from the original image and from the coefficients yielded by the DWT. Experimental results reached a hit rate of 94.93% with a Multilayer Perceptron classifier outperforming moments-based descriptors and kNN classifiers.

Recent object description approaches rely on local features since segmentation, which represents by itself an unsolved problem, is no longer needed. The development of image matching by using a set of local interest points was definitively efficient when Lowe [16] presented SIFT, introducing invariance to the local feature approach. Later, Bay et al. [4] outperformed previously proposed schemes with respect to repeatability, distinctiveness, robustness and speed. Recently, Özuysaletal et al. [19] showed a fast key point recognition method using random Ferns which avoid the computationally expensive patch preprocessing by using hundreds of simple binary features.

However, using exclusively ILF can cause complementary information lost and ambiguities may occur locally when an image presents several similar regions, producing mismatches among key points. As a consequence, some recent works deal with the combination of ILF and global description of the image. Some approaches [5], [22], [24] firstly extracted ILF and, after that, they evaluated some global features over the previous local key points. On the contrary, Zhang et al. [25] calculated global features to determine interest regions and afterwards, they studied local description in these areas of the images. Finally, there are works that rely on the combination of local and global descriptors to perform the matching. Cao et al. [7] calculated affine-invariant SIFT feature descriptors and a global context vector that adds curvilinear shape information on various images obtaining better results when using the combination than without.

Euclidean distances among global descriptors as well as among local features were computed and averaged to carry out the matching step. Lim and Gallogahi [15] carried out a similar matching approach, but instead, they weighted the global and the local contributions when addressing the shape classification problem using Histogram of Oriented Gradient (HOG) in both local and global feature description. Li and Ma [13] improved SIFT descriptor by adding invariance through an elliptical neighbouring region and then integrated colour and global information with it. In the matching process, they discarded matches with a distance above a set threshold in a three-step comparison: the improved SIFT descriptors through a simple Nearest Neighbours approach, three colour invariance components and the global component.

The rest of the paper is organised as follows: Section 2 describes the methods used to determine the acrosome quality. The classification methods used are introduced in section 3. Results achieved with the different classifiers are presented in section 4. Finally, section 5 shows our conclusions.

2 METHODS

2.1 IMAGE ADQUISITION AND PREPROCESSING.

The boar semen images were captured with a resolution of 780 × 580 pixels using a digital camera connected to a phase-contrast microscope with a 100x magnification. Therefore, most of the spermatozoa come from different takings, which means that illumination is not completely constant, leading to a robust method to illumination changes [20].

Two snapshots of the same spermatozoa are taken without moving the position of the sample: one in real colour under fluorescent illumination and the other one in grey scale under positive phase contrast illumination. The heads of the phase-contrast images are then cropped and labelled as damaged or intact based on the real colour fluorescent image. Overlapped heads cannot be analysed, so they are discarded from the set of images. Luckily, overlapped heads do not appear frequently due to the conditions under which the sample is obtained.

After they are labelled, each head is automatically registered and cropped leading to equal orientations of the heads in order to assure scale and rotation invariance. Once all images have the same orientation, our goal is to modify them to achieve the same dimensions. In this case, we have used the median dimension that proved better results than other statistics. Finally, bad registered images are manually discarded.

After all this process we have a set of 856 intact and 861 damaged heads. Figure 1 shows 4 intact and damaged registered spermatozoa heads.

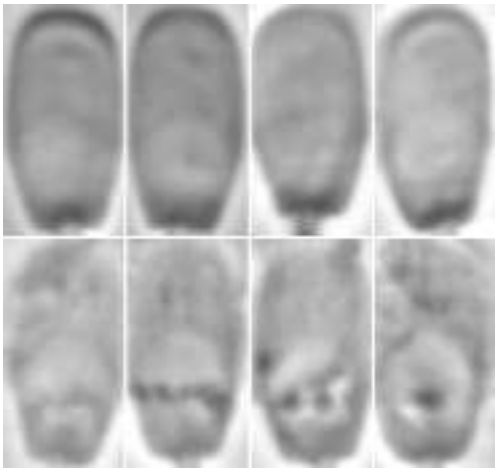


Figure 1: Intact (upper row) and damaged (lower row) registered heads.

2.2 IMAGE DESCRIPTION.

Global texture descriptors, ILF descriptors and a combination of both were evaluated. Although shape may be considered at first sight as an appropriate descriptor, it has been proved in previous works [3] that hit rates achieved were much lower.

2.2.1 Texture Descriptors

In Figure 1, it can be observed the high textural similarity presented among the images of each class (intact class or damaged class) whereas there is a significant difference with respect to image texture between the two classes. This assertion led to study the texture global description of the dataset. Haralick features, Laws masks, Legendre and Zernike moments were computed leading to acceptable results. Other texture descriptors were discard due to its insufficient success in previous works [2] [3] [12].

The first descriptor used is made up of 13 out of the 14 features proposed by Haralick, being Maximal Correlation Coefficient the excluded one. This method is called Haralick henceforth.

The second one relies on Laws masks and consists of applying convolutions with several filters to images, yielding as many images as convolutions are carried out. Let be I the initial image and g1, g2, ..., gn a set of filters. A generic image resulted after the convolution is defined by Jn = I * gn. Kernels used are defined for neighbourhoods of 5 × 5, so a 16 feature vector is obtained for each image pixel. Subsequently, vectors are reduced from 16 to 9 features. Let be Fk [i, j] the result of applying the k-th mask on the pixel (i, j), then the map Ek of the texture energy for the filter k is defined as:

$$E_k(r,c) = \sum_{i=c-7}^{c+7} \sum_{i=r-7}^{r+7} |F_k(i,j)| \qquad (1)$$

As third descriptor, Legendre moments were computed, allowing new applications such as the reconstruction of an image from the mathematical features provided by these moments defined by:

$$\lambda_{p,q} = \frac{(2p+1)(2q+1)}{4} \int_{-1}^{1} \int_{-1}^{1} P_p(x) P_q(y) f(x,y)\, dx\, dy \qquad (2)$$

where P_p y P_q are the Legendre polynomials. This led into a descriptor of 9 features.

For the last method, Zernike moments were used due to its rotation, scale and translation invariance. As Zernike moments are orthogonal, the reconstruction of the original function from the obtained one is simplified. Low order moments represent the global shape of a pattern and higher order ones represent the detail. This descriptor has a 27-feature dimensionality.

2.2.2 Invariant Local Feature Descriptors

Spermatozoa with damaged acrosomes heads show regions such as black dots that could be easily located as corners (Figure 1). This hypothesis means that damage class visually presents potential keypoints and therefore, we implemented an ILF approach. We used SIFT and SURF detectors / descriptors. SIFT was the first method to achieve invariant description through the detection of robust local features. Its good performance has been widely proven in recognition and classification tasks [8] [16] [23]. We have also tried SURF since it entails an improvement over SIFT according to their authors [4].

On the one hand, SIFT transforms image data into scale-invariant coordinates relative to local features. It can be described in 4 main stages. First, it searches over all scales and image locations by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation. Secondly, at each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability (see Figure 2). Later, one or more orientations are assigned to each keypoint location based on local image gradient directions, providing invariance to orientation, scale and location transformations. Finally, the local image gradients are measured at the selected scale in the region around each keypoint. Feature descriptors have a 128 dimensionality for each keypoint of the image: 8 directions for each histogram orientation in 4 × 4 subregions.

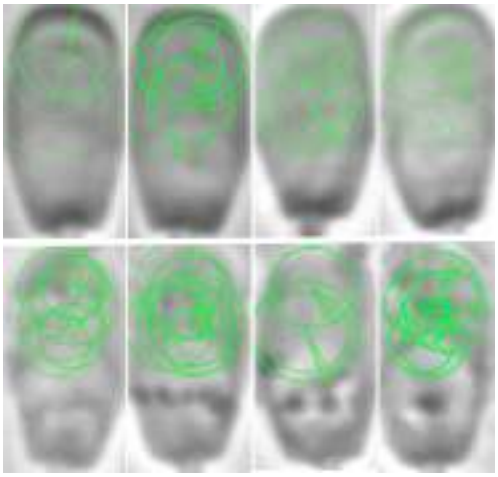


Figure 2: Intact (upper row) and damaged (lower row) registered heads with SIFT detected points.

Keypoints are displayed as circles indicating scale, orientation, and location.

On the other hand, SURF (Speeded-Up Robust Features) [16] method, based on local features, was used. SURF can be computed and compared much faster by relying mainly on integral images for image convolutions. A Hessian matrix-based measure (H(x,σ)) for the interest point detection, defined in equation 3, has been used.

$$H(x,o) = \begin{bmatrix} L_{xx}(x,o) & L_{xy}(x,o) \\ L_{xy}(x,o) & L_{yy}(x,o) \end{bmatrix} \quad (3)$$

Concatenating the descriptor vectors for all 4 × 4 subregions, this result in a features vector of length 64 for each key point of the image.

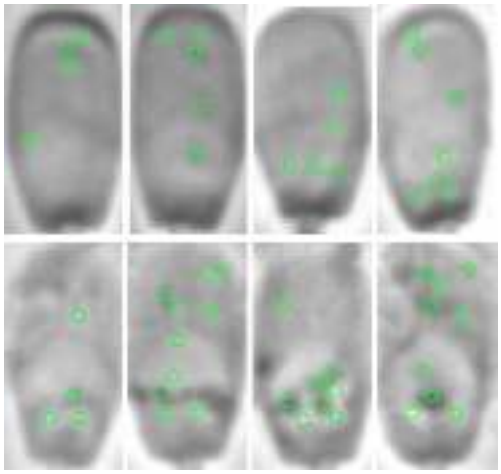


Figure 3: Intact (upper row) and damaged (lower row) registered heads with SURF detected points.

2.2.3. Combining ILF descriptors with global texture descriptors

Studying the results of the previous methods, it is noticeable that, in general, while texture global descriptors classify better the intact class than the damage one, ILF descriptors yielded the opposite situation instead (Figure 4). Thus, better results with the combination of both descriptions were expected.

We had previously calculated one global texture vector descriptor and as many ILF vectors descriptors as key points for each image of our data set. In each image, we chose to concatenate its global vector with each of the key points descriptors. Therefore, the matching process is directly affected by the dimensionality of the original descriptors.

3 CLASSIFICATION

Classification was carried out through different well-known methods.

The goal is to compare results and achieve the best possible hit rate.

3.1 k-Nearest Neighbours

Following Evans [11], a classification method based on kNN was evaluated for SURF and SIFT descriptors. In this approach, key point features are matched between two images. In a test image, each keypoint is related with the nearest keypoint of a training image using a Euclidean distance. The sum of all distances gives an insight of the likeness between both images. This process is repeated with every training image. Our test image will be assigned to the most repeated class along the k higher resemblance images (k-Nearest Neighbours).

In order to compare this method with the classical descriptors whose feature vector has only one dimension, a common kNN method was used to classify them through several values of k.
kNN matching was repeated 10 times and error rates over the 10 runs averaged.

3.2 Neuronal Network and Support Vector Machine

Global texture descriptors were classified using NN and SVM since these algorithms often provide better results than a simple kNN.

Our Neural Network was trained using a back-propagation algorithm with one hidden layer and a logistic sigmoid transfer function for the hidden and the output layer. Data was normalised with zero mean and standard deviation equal to one. Learning was executed with a momentum and adaptive learning rate algorithm. 10-fold cross validation with several different combinations of neurons in the hidden layer and training cycles was carried out in terms of the classification so the most accurate configuration could be found out.

The SVM training for the classic descriptors was performed using a Gaussian kernel with σ = 1 and Sequential Minimal Optimization (SMO) algorithm.

3.3 Bag of Words

Aiming to compare the special kNN matching with other method that deals with several feature descriptors for image, we implemented the Bag of Words (BOW) model [1, 14]. Consequently, BOW

was implemented with SIFT, SURF and the combined descriptors.

BOW allows a dictionary-based modelling. Hence, each image descriptor looks like a bag that contains some local points from the dictionary, thus the order is not considered. It consists mainly in two steps: First, it obtains "n" centres using a clustering algorithm taking into account all the image points. In our case, we have evaluated BOW with k-means cluster method. Secondly, each point is assigned to its corresponding centre taking into consideration the distance between points and centres. The output of BOW method for an image is a vector of size "n" where each element n_i shows the number of its points assigned to the centre i. This output was then classified using SVM method.

This process is repeated 10 times in order to achieve robustness to random choices. The final error rate is the average of the error rates during those 10 runs. The proximity between the patterns is computed using the Euclidean distance.

4 RESULTS

4.1 k-Nearest Neighbours

Figure 4 shows the global, intact class and damage class hit rates obtained with ILF and global texture descriptors separately. The best result is achieved by SURF with an accuracy of 94.88% followed by Legendre (87.55%). SIFT and SURF methods produce higher hit rates in damaged than in intact heads classification while global descriptors behave conversely, except for Haralick whose results are quite balanced. The homogeneous texture of the intact class in contrast with the high variability of the damaged one, could lead to more significant key points for the ILF description in the last set of images. Thus, the combination of both description techniques could take advantage of this situation.

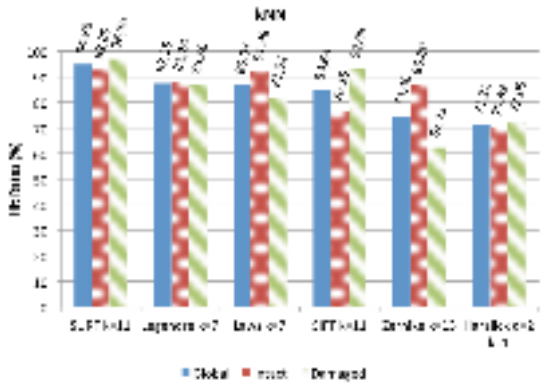


Figure 4: Global, intact and damaged hit rates for the combination of SURF with our global texture descriptors classified with kNN.

Hit rates of the combination of SURF and SIFT with our four global descriptors are shown in Figure 5 and Figure 6 respectively and confirm our expectations. Best results are obtained when combining ILF descriptors with Legendre, with a best overall result of 95.56% with SURF and Legendre. The combination of SIFT and Legendre yields a hit rate of 88.98%.

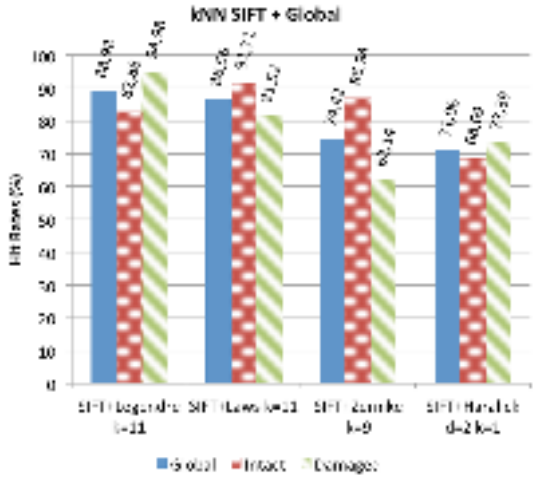


Figure 5: Global, intact and damaged hit rates for the combination of SIFT and our global texture descriptors classified with kNN.

4.2 Neuronal Network and Support Vector Machine

NN and SVM classification results for global texture descriptors can be seen in Figure 7 and Figure 8 respectively. Haralick descriptor outperforms the rest with a hit rate of 93.43% using NN and 92.14% using SVM. NN and SVM outperform kNN when classifying global texture descriptors as we anticipated.

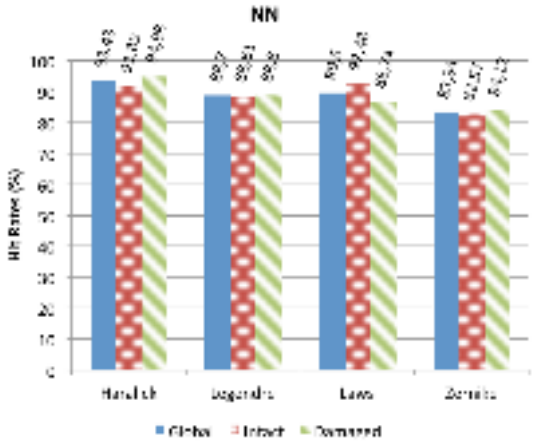


Figure 6: Global, intact and damaged hit rates classified with Neural Networks.

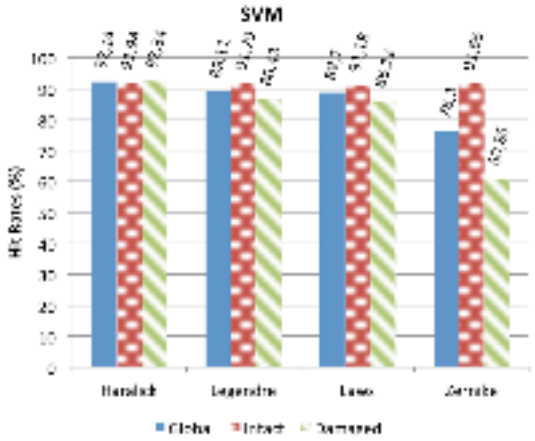


Figure 7: Global, intact and damaged hit rates classified with SVM.

4.3 Bag of Words

Lastly, Figure 9 shows BOW classification results for SURF and its combination with our global texture descriptors whereas Figure 10 does it for SIFT and its combinations. We have tested several sizes of the dictionary in order to assure the best possible results. Combining SURF and Laws, a hit rate of 75.66% with 4 elements in the dictionary is yielded. SIFT and Laws combination obtains the best hit rate (75.97%) with 4 centres, yet very low in relation with kNN matching results. Again, an improved performance when combining global and local description is noticed, proving the efficiency of our combination theory. The low number of points detected in this kind of images can explain the bad performance obtained, which implies that this method is not discriminatory enough.

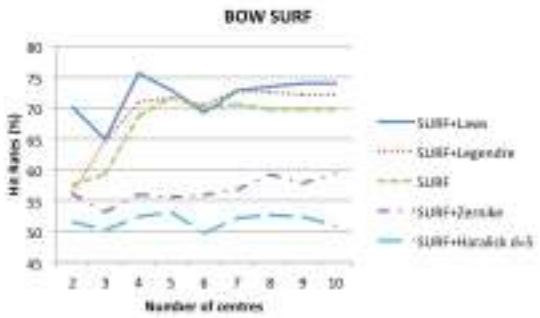


Figure 8: Global hit rates of SURF and its combinations classified with BOW.

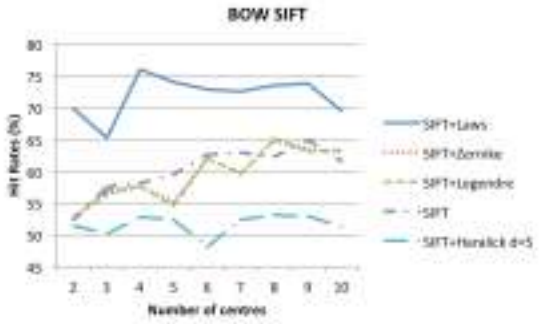


Figure 9: Global hit rates of SIFT and its combinations classified with BOW.

5 CONCLUSIONS

This work proves the success of applying a computer vision approach to classify images of boar spermatozoa heads as intact or damaged acrosome.

SURF and SIFT Invariant Local Features descriptors were computed and compared with traditional global texture descriptors such as Legendre and Zernike moments, Laws masks and Haralick features. Classification was carried out through a kNN-based algorithm, NN, SVM and BOW methods. SURF classified with kNN proved to obtain better accuracy (94.88%) than the rest. It was remarkable that ILF methods classified better the damaged than the intact class while the opposite situation was achieved by global texture descriptors.

Furthermore and more importantly, a novel combination of ILF with global texture descriptors was presented improving previous works in this area of knowledge [2] [3] [12] [21]. Both, kNN-based matching algorithm and BOW approach, reached a better performance when combining descriptors than with these descriptors separately. Best overall result was yielded by the combination of SURF with Legendre descriptors and kNN-based classifier,

reaching a hit rate of 95.56%. This result is satisfactory enough for the purpose of the paper according with the veterinarian community.

Acknowledgement

This work has been supported by grants DPI2009-08424, PR2009-0280 and DPI2012- 36166 and via the pre-doctoral FPU fellowship program from the Spanish government.
The authors would like to thank CENTROTEC for providing the semen samples and for their collaboration in the image acquisition.

References

[1] Aldavert, D., Ramisa, A., López de Mantaras, R., and Toledo, R.: 'Real-time object segmentation using a bag of features approach'. CCIA, 2010.

[2] Alegre, E., Biehl M., Petkov N., and Sanchez L.: 'Automatic classification of the acrosome status of boar spermatozoa using digital image processing and lvq'. Comput. Biol. Med, 2008, 38, pp. 461–468.

[3] Alegre, E., González-Castro, V., Alaiz-Rodríguez, R., and García-Ordás, M. T.: 'Texture and moments-based classification of the acrosome integrity of boar spermatozoa images'. *Comput. Methods Programs Biomed*, 2012, doi:10.1016/j.cmpb.2012.01.004.

[4] Bay, H., Tuytelaars, T., and Van Gool, L.: 'Surf: Speeded up robust features'. *European Conference on Computer Vision*, 2006.

[5] Besbes, B., Apatean, A., Rogozan, A., and Bensrhair, A.: 'Combining SURF-based local and global features for road obstacle recognition in far infrared images'. *Intelligent Transportation Systems (ITSC), 13th International IEEE Conference on*, 2010, pp. 1869–1874.

[6] Broekhuijse, M., Sostarik, E., Feitsma, H., and Gadella, B.M.: 'Additional value of computer assisted semen analysis (casa) compared to conventional motility assessments in pig artificial insemination'. *Theriogenology*, 2011, 76, (8), pp. 1473–1486.

[7] Cao, B., Ma, C. & Liu, Z. (2010). Affine-invariant SIFT descriptor with global context. Image and Signal Processing (CISP), *3rd International Congress on*, vol.1, 68–71.

[8] Chang, L., Duarte, M. M., Sucar, L.E., and Morales, E. F.: 'A Bayesian approach for object classification based on clusters of SIFT local features'. *Expert Systems with Applications*, 2012, 39, (2), pp. 1679–1686.

[9] E Alegre, M Garcia-Ordas, V Gonzalez-Castro, S Karthikeyan. Vitality assessment of boar sperm using ncsr texture descriptor in digital images. *Iberian Conference on Pattern Recognition and Image Analysis proceedings*. 2011.

[10] E Alegre, V Gonzalez-Castro, S Suarez, M Castejon. Comparison of supervised and unsupervised methods to classify boar acrosomes using texture descriptors. *ELMAR, 2009. International Symposium*, 65-70.

[11] Evans, C.: 'Notes on the OpenSURF Library'. (Bristol: University of Bristol, 2005).

[12] Gonzalez-Castro, V., Alaiz-Rodriguez, R., Fernandez Robles, L., Guzman-Martinez, R., and Alegre, E.: 'Estimating class proportions in boar semen analysis using the hellinger distance'. *XIII The Twenty Third International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*, 2010.

[13] Li, C., and Ma, L.: 'A new framework for feature descriptor based on SIFT'. Pattern Recogn. Lett., 2009, 30, (5), pp. 544–557.

[14] Li, T., Mei, T., Kweon, I-S., and Hua, X-S.: 'Contextual bag-of-words for visual categorization'. IEEE Transactions on Circuits and Systems for Video Technology, 2011.

[15] Lim, K.-L., and Galoogahi, H. K.: 'Shape Classification Using Local and Global Features'. *Proceedings of the 2010 Fourth Pacific-Rim Symposium on Image and Video Technology*, IEEE Computer Society, 2010, pp. 115–120.

[16] Lowe, D.: 'Distinctive image features from scale invariant keypoints'. *International Journal of Computer Vision*, 2004, 2, pp. 91–111.

[17] M González, E Alegre, R Alaiz, L Sánchez. Acrosome integrity classification of boar spermatozoon images using dwt and texture techniques. VipIMAGE-*Computational Vision and Medical Image Processing*, 2007, 165-168.

[18] Oscar Garcia-Olalla; Enrique Alegre; Laura Fernandez-Robles; Maria Teresa Garcia-Ordas; Diego Garcia-Ordas .Adaptive local binary pattern with oriented standard deviation (ALBPS) for texture classification. *EURASIP Journal on Image and Video Processing.Springer*, 2013.

[19] Ozuysal, M., Calonder, M., Lepetit, V., and Fua, P.: 'Fast keypoint recognition using random ferns'. *Pattern analysis and Machine Intelligence*, 2010, 32, pp. 448–461.

[20] Sanchez, L., Petkov, N., and Alegre, E.: 'Classification of boar spermatozoid head images using a model intracellular density distribution'. *Progress in Pattern Recognition, Image Analysis and Applications: Proceedings of the 10th Iberoamerican Congress on Pattern Recognition*, 2005.

[21] Suarez, S. A., Alegre, E., Castejon-Limas, M., and Sanchez, L.: 'Use of statistic texture descriptors to classify boar sperm images applying discriminant analysis'. *Computational Vision and Medical Imaging Processing*, England, 2008, pp.197–202.

[22] Sun, L., and Liu, G.: 'Visual Object Tracking Based on Combination of Local Description and Global Representation'. *Circuits and Systems for Video Technology, IEEE Transactions on*, 2011, 21, (4), pp. 408–420.

[23] Tuytelaars, T., and Mikolajczyk, K.: 'Local Invariant Feature Detectors: A Survey'. *Foundation and Trends in Computer Graphics and Vision*, 2007, 3, (3), pp. 177–280.

[24] Xu, Y., Huang, S.-B., Ji, H., and Fermuller, C.: 'Combining powerful local and global statistics for texture description'. *Computer Vision and Pattern Recognition* (CVPR) IEEE Conference on, 2009, pp. 573–580.

[25] Zhang, H., Jia, W., He, X., and Wu, Q.: 'Learning-Based License Plate Detection Using Global and Local Features'. *Pattern Recognition, ICPR 18th International Conference on*, 2006, 2, pp. 1102–1105.

Detección y estimación de la orientación del rostro del conductor basada en modelos 3D

Gustavo A. Peláez C., Fernando García F., Arturo de la Escalera, José M. Armingol.
Universidad Carlos III de Madrid, {gpelaez,fgarcia,escalera,armingol}@ing.uc3m.es

Resumen

Se presenta un sistema de detección y análisis del rostro del conductor. A partir de un análisis bidimensional, se ha obtenido una reconstrucción tridimensional del rostro con detección de las regiones de interés (ojos y nariz). Para, posteriormente, analizar los ángulos de inclinación con la intención de conocer la orientación de la mirada.
La detección de rostros se ha obtenido mediante el algoritmo haar-like. Una vez obtenido, se realiza un modelado en tres dimensiones que permitirá, por medio de matrices de transformación, conocer los ángulos de rotación de Euler de la cara. Por último, las zonas de interés, se obtienen por medio de restricciones geométricas del rostro y algoritmos de visión.
Esta aplicación es el primer paso para un sistema completo de monitorización del conductor que permitirá conocer, en todo momento, la orientación de la mirada, el estado de somnolencia y el nivel de atención de este.

Palabras Clave: detección de rostros, visión por computador, sistemas de ayuda a la conducción.

1 INTRODUCCIÓN

Los estudios demuestran que la mayoría de accidentes de tráfico se producen por errores del conductor o faltas de atención. Para intentar mitigar estos accidentes, durante los últimos años han surgido numerosas aplicaciones que tratan de llevar los avances en los sistemas de información y comunicaciones al mundo de la seguridad vial. Estos tipos de sistemas, son los denominados Sistemas de Ayuda a la Conducción (ADAS, por sus siglas en inglés) y están englobados en los Sistemas Inteligentes de Transporte (SIT). En este documento se presenta una aplicación novedosa, para el estudio y monitorización del comportamiento del conductor, permitiendo detectar, en su fase inicial, la cara del conductor, así como su ángulo con relación al mundo real. Además de permitir la identificación de rasgos faciales, fundamentales en futuros trabajos de cara a una monitorización exhaustiva del conductor, permitiendo detectar conductas o movimientos de distracción peligrosa así como efectos de la fatiga al conducir.
El sistema que se presenta hace uso de uno de los sensores más en boga en la actualidad, la Kinect de Microsoft, sistema que incluye diversos sensores, permitiendo una reconstrucción tridimensional del conductor y del entorno del mismo (interior del coche) así como la creación de un modelo de la cabeza, que permite el cálculo de los ángulos de giro de la misma, así como la identificación de las diversas características de la misma (ojos, nariz, etcétera). De esta forma es posible monitorizar gestos, estados de fatiga o distracciones con gran nivel de detalle. En la figura 1 se describe el sistema general empleado.

2 ESTADO DEL ARTE

Numerosos trabajos han experimentado con la posibilidad de la detección de caras/rostros, la mayor parte de ellos centrados en sistemas basados en imágenes bidimensionales. Pese a que la aplicación práctica que buscan y los sensores empleados difieren en gran medida, todas ellas comparten muchos aspectos. En [7] se emplea un sistema de modelado rápido de caras, basado en dos cámaras. La segunda cámara permite trabajar con información de profundidad más fiable. Gracias a la fusión de la información de las dos cámaras, se puede obtener la forma en 3D de la cara. Por último, la intensidad de la imagen frontal es mapeada en la reconstrucción 3D, obteniendo una reconstrucción de gran realismo. Otro ejemplo es el propuesto en [8] basada en un algoritmo para extracción de características propias de la cara humana y su uso para reconocimiento, basado en la transformada Curvelet y en una descomposición de valores singulares. Después de hacer la transformara Curvelet, la energía de las bajas frecuencias es extraída primero y posteriormente la de las altas. A continuación se realiza una descomposición de valores singulares, para finalmente fusionar las características faciales con estas componentes realizando una clasificación basado en el algoritmo de Nearest Neighbor (NN) empleando una base de datos de caras para confirmar la validez del algoritmo. Por último, en [2] se realiza detección en tiempo real de la somnolencia en el

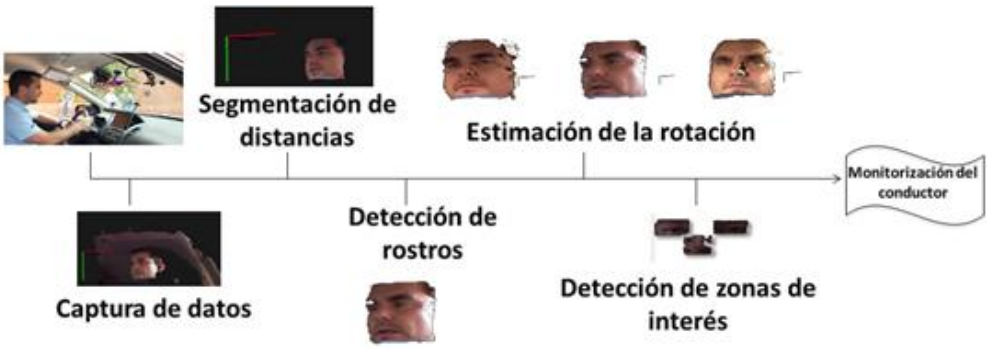


Figura 1: Esquema general del sistema

conductor, mediante la detección de la apertura o no de los ojos y midiendo la frecuencia de parpadeo.

La Kinect se ha convertido en uno de los sensores referentes en los sistemas de percepción, gracias a su gran versatilidad y su escaso precio. La posibilidad de fusionar información de diversas fuentes hace que este sistema sea un referente en los sensores de percepción modernos, surgiendo numerosos proyectos que hacen uso de las posibilidades que Kinect y su comunidad de desarrolladores han abierto. En [3] el seguimiento de una mano permite su uso en aplicaciones orientadas a la realidad virtual, por otro lado otros trabajos, como el presentado en [10] ayudan a interaccionar con objetos virtuales en aplicaciones de realidad aumentada. También es posible utilizar el sensor para detectar la presencia humana [11], basado en información de profundidad y aprovechando el contorno de la cabeza. Por último, cabe destacar que la Kinect ha supuesto una herramienta de gran utilidad en el mundo de la robótica, como sensor que permite una interactuación a un nivel mayor, como en [4] que presenta un robot móvil, dotado de un sistema Kinect que es empleado para la localización del mismo.

Como se ha comentado, la aplicación que se presenta trata de emplear el sensor Kinect, para mejorar los sistemas de ayuda a la conducción, algunos de los cuales ya están disponibles en la actualidad, no solo monitorizando al conductor, sino también detectando peatones [6], vehículos [5] e incluso mejorando sistemas ya existentes, como aumento de la cobertura GPS [9] o detección y clasificación de carriles viales [1].

3 DESCRIPCIÓN DEL SISTEMA

El sistema está desarrollado sobre la Kinect que es una barra de sensores para la consola XBOX 360 de Microsoft. Kinect está compuesta por un conjunto de diversos sensores que permiten la captura de imágenes en 2 dimensiones a color y con la proyección de puntos infrarrojos, se puede determinar además la profundidad dando así un mapa en 3 dimensiones del entorno. Además, cuenta con 4 micrófonos configurados de tal forma que recorre toda la estructura para captar comandos de audio para determinadas aplicaciones que así lo requieran.
La barra de sensores se ha colocado enfrente del conductor de tal forma que no obstruya el campo de visión de este como se ve en la figura 2. Al estar montada justo sobre el salpicadero, la parte frontal del cristal que corresponde al campo de visión del conductor está despejada y el sensor tiene un campo de visión adecuado para realizar su tarea sin problema alguno(figura 2). La colocación del sensor se ha configurador de tal forma que no se evitan oclusiones de zonas de importancia para la monitorización del mismo, como la boca o los ojos, además de permitir una reconstrucción completa del contorno de la cabeza del conductor.

4 FUNCIONAMIENTO DEL SISTEMA



Figura 2: Ubicación de los sensores en el coche inteligente

El sensor captura una nube *N* de puntos que se describe con la ecuación (1) donde *x*, *y*, *z* son dimensiones de la nube incluyendo la profundidad y *c* es el color del pixel en rojo, verde y azul (RGB por sus siglas en inglés) cuyas dimensiones son 640 pixeles por 480 pixeles en ancho y alto respectivamente.

$$N = \sum_{x=0}^{w} \sum_{y=0}^{h} \sum_{z=0}^{d} \sum_{c=1}^{3} p_{xyzc} \quad (1)$$

De esta nube se extraerá la información a color de forma ordenada de manera tal que se pueda construir la matriz de las mismas dimensiones pero solo con el color como se describe en (2). Se aplicará la restricción de que un punto es válido para ser extraído si la distancia *z* es menor a 1 metro. Esto parte de la suposición de que el conductor estará colocado a una distancia del sensor entre 50 centímetros y 1 metro. Al aplicar esta restricción, se eliminará el espacio que se encuentra detrás del conductor y se reducirá el área de búsqueda del rostro, haciendo más rápida la aplicación en general. También servirá para reducir la posibilidad de falsos positivos.

$$M = \sum_{x=0}^{w} \sum_{y=0}^{h} \sum_{c=0}^{3} p_{xyc} \quad (2)$$

Una vez obtenida la matriz, se realiza una búsqueda del rostro con el uso de características del tipo haar-like [12]. Este método se encargará de recorrer toda la imagen buscando características comunes y con una morfología fácil de identificar, como nariz, boca, etcétera.

Después de haber detectado el rostro, se guardarán los puntos que definen el rectángulo que lo rodea. Luego, los puntos que pertenecen a la zona interior del rectángulo son copiados de la nube *N* a otra nube *Nc* (3) que será la representación en 3 dimensiones de la cara.

$$Nc = \sum_{x=0}^{wc} \sum_{y=0}^{hc} \sum_{z=0}^{dc} \sum_{c=1}^{3} p_{xyzc} \quad (3)$$

Donde *wc, hc* y *dc* son los valores del ancho, alto y profundidad de la cara respectivamente.

Además de detectar la cara, también resulta posible detectar los ojos y la nariz. Esto se logra una vez el rostro haya sido encontrado y se utilizarán, al igual que anteriormente se hiciera con el rostro, las características del tipo haar-like con un entrenamiento específico para estas regiones de la cara. Un conjunto de restricciones geométricas se aplicarán previamente a hacer la búsqueda de los ojos y la nariz. Esto con el objetivo de que la búsqueda sea más rápida. Se ha decidido utilizar estas 3 zonas y no la boca debido a que el conductor podría estar haciendo diversas expresiones faciales que alterarían la detección de la boca, mientras que la nariz y ojos presentan menores alteraciones. Las restricciones para los ojos especifican una zona rectangular colocada en el centro del rostro con ancho igual al total del rostro y de alto un 25% del total del rostro. De forma similar se crea la ventana de búsqueda para la nariz. Estas restricciones se ejemplifican mejor en la figura 3.

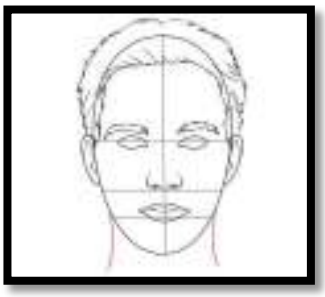


Figura 3: Líneas de orientación para el dibujo de un rostro.

A manera de hacer más rápidos los algoritmos a efectuarse posteriormente, se aplicará un filtrado del tipo voxel que se encargará de reducir el número de partículas de *Nc*. Este método reduce la cantidad de puntos de una nube haciéndola menos densa. Para lograr esto, se crea un conjunto de voxels, basado en cajas 3D, de todo el objeto. Una vez creados los voxels, se reduce todo el objeto al considerar únicamente el punto central de este (Figura 4).

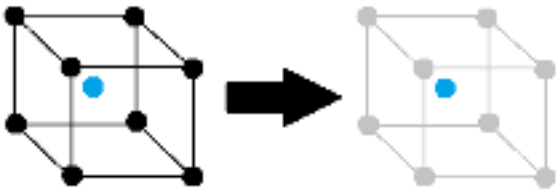


Figura 4: Ejemplo de reducción de la nube por voxel.

El programa tomará la matriz *Nc* inicial como referencia para calcular la matriz de transformación respecto a las siguientes a ser capturadas. De esta matriz interesan los ángulos de Euler para determinar hacia donde está orientada la nube actual respecto a la primera. Para obtener dicha matriz de transformación se utilizará el método de punto cercano iterativo (ICP por sus siglas en inglés). Este algoritmo ajusta los puntos de entrada a un modelo predeterminado por medio de la minimización de la suma de errores cuadráticos y la salida es la matriz de transformación entre las dos nubes los pasos son los siguientes:

1. Asociar los puntos por medio del criterio del vecino más cercano con la ecuación (4).

$$d(p_1, p_2) = \|p_1 - p_2\| \quad (4)$$

Donde p_1 es el punto de la nube modelo y $p_2$es el punto de la nube a comparar (Figura 5).

2. Esto asume que los dos objetos tienen la misma cantidad de puntos, de no ser así, se tomaría un segmento de la nube más grande.

3. Se calculan los parámetros de transformación entre los dos objetos previamente asociados utilizando la ecuación de mínimos cuadrados descrita en (5).

$$MC = \frac{1}{n}\sum_{i=1}^{n}(p_1 - p_2)^2 \qquad (5)$$

4. Aplicar la transformación de los puntos de la nube a evaluar con el modelo establecido.

5. Determinar el error de ajuste entre las dos nubes.

6. Se vuelve al paso 1 hasta que el error obtenido esté dentro de los límites determinados.

Una vez filtrada la nube con voxel, se aplica el ICP para obtener la matriz de transformación *Mt* (6) para determinar los nuevos valores *x, y, z* de la nube proyectada(Figura 5).



Figura 5: Nube modelo en la izquierda y nube a evaluar en la derecha.

$$Mt = R\left(\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + T\right) \qquad (6)$$

Donde *T* está definida como el vector de traslación en los tres ejes $\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}$

Mientras que *R* está definida como la matriz de rotación que se obtiene de la multiplicación de las tres matrices de rotación en cada eje *R1* (7), *R2* (8) y *R3* (9) en los ejes Z, Y, X respectivamente

$$R1 = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & 0 \\ sin(\Delta\theta) & cos(\Delta\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$R2 = \begin{bmatrix} \cos(\Delta\delta) & 0 & \sin(\Delta\delta) \\ 0 & 1 & 0 \\ -\sin(\Delta\delta) & 0 & \cos(\Delta\delta) \end{bmatrix} \quad (8)$$

$$R3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & \cos(\Delta\varphi) & -\sin(\Delta\varphi) \\ 0 & \sin(\Delta\varphi) & \cos(\Delta\varphi) \end{bmatrix} \quad (9)$$

De ICP se obtiene la matriz *Mt* y es de esta matriz de donde hay que obtener los ángulos de Euler de acuerdo a las ecuaciones (10) (11) y (12).

$$\Delta\theta = arcsin(R[1,0]) \qquad (10)$$

$$\Delta\delta = arctan2(-R[2,0], R[0,0]) \qquad (11)$$

$$\Delta\varphi = arctan2(-R[1,2], R[1,1]) \qquad (12)$$

Donde $\Delta\theta, \Delta\delta$ y $\Delta\varphi$ son los ángulos de rotación respecto a los ejes Z, Y, X respectivamente.
Utilizando un rango de valores para cada ángulo se puede desplegar en pantalla si la cara detectada está orientada hacia arriba, abajo, izquierda, derecha y la rotación que puede ser en sentido horario o anti-horario (Figura 6). Permitiendo monitorizar el movimiento del conductor en todo momento y detectar situaciones de distracción.



Figura 6: nube detectada y rotada.

En el caso del análisis para ojos y nariz, la figura 7 muestra la nube resultante de la segmentación de estas regiones de interés. Esta información es útil para el estudio y monitorización del conductor, detalles como la apertura de los ojos o la dirección de la mirada son útiles para conocer el estado de fatiga o la atención del conductor.

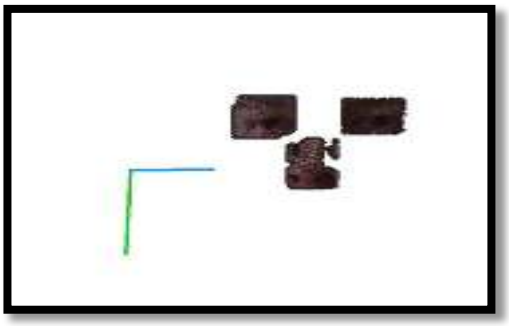


Figura 7: Nube de puntos con ojos y caras.

5 RESULTADOS

El sistema fue probado con diferentes condiciones lumínicas y posiciones de la cara. En todos los casos donde la cara estaba presente, se ha logrado obtener el modelo 3D de la cara y sus tres ángulos de rotación. La velocidad dependerá de la configuración de los parámetros de las funciones como ICP o voxel. En promedio, se obtuvo un resultado completo cada 80 milisegundos en un procesador Intel Core i7 a 2Ghz y 8Gb de RAM.

Aunque el dispositivo fue diseñado con una resolución de profundidad pensada para analizar objetos del tamaño del cuerpo humano, se ha logrado un buen resultado como se puede ver en la figura 8 que muestra un resultado de la captura. En esta figura, se pueden distinguir diversas características del rostro.

Los ángulos se han obtenido de la matriz de transformación resultante del método ICP. Las pequeñas perturbaciones se han eliminado al hacer un promedio entre las últimas medidas, en este caso, las últimas 5. No se han obtenido falsos positivos ya que las restricciones de los buscadores del tipo haar-like indican que la cara debe ser de un tamaño mínimo para ser considerada como válida. Esto, agregado a la filtración por distancia, permiten que la detección de una cara, por consiguiente su modelación en 3 dimensiones, sea bastante fiable.

El resultado final es un modelo en 3 dimensiones de la cara que tiene la misma información que una detección realizada de forma exitosa en 2 dimensiones pero además también agrega la variable de profundidad que sería de utilidad para otras herramientas que así consideren necesario hacer uso de dicho valor, además de la detección de regiones e interés para la monitorización del conductor, en concreto ojos y nariz.



Figura 8: Nube de puntos del rostro

6 CONCLUSIONES

Se ha presentado un sistema de detección y modelación 3D del rostro del conductor por medio de herramientas de visión. El hecho de que sea por medio de visión, hace de este método una solución no invasiva por lo que no obstruye al conductor en ninguna manera y no requiere de ningún dispositivo adicional colocado sobre el conductor.

Probado en diversas condiciones lumínicas, el sistema ha sido capaz de detectar tanto la cara, como los ojos y la nariz. Los ángulos obtenidos se corresponden a escala con el movimiento real de la cabeza del conductor por lo que es posible determinar la dirección hacia donde el conductor está mirando. Además, el sistema, con un tiempo de ejecución de 80 milisegundos, ha demostrado que es posible realizar un algoritmo de monitorización del conductor en tiempo real, gracias en gran medida a la correcta configuración de otras técnicas adicionales utilizadas, como filtrado por voxel y distancia. Este último tipo de filtrado resulta ser muy útil, permitiendo eliminar la parte no relevante de la imagen, al contrario de otras soluciones, que utilizan cámaras de diversos tipos, pero siempre haciendo un análisis más complejo en 2 dimensiones. Se ha planteado la idea de efectuar el análisis posterior de los objetos detectados, es decir, los ojos y la cara en general, permitiendo obtener resultados más refinados y rápidos, proporcionando información de alto nivel a otros sistemas a bordo del vehículo. Por último, cabe destacar que en la actualidad se está investigando la viabilidad de incluir en el sistema la capacidad de trabajar bajo condiciones de luminosidad más complejas, haciendo uso del sensor infrarrojo que dispone el sensor, que permite trabajar en condiciones de baja o nula visibilidad.

Agradecimientos

Proyecto financiado gracias a los proyectos CICYT (Grand TRA2010-20225-C03-01) y (Grand TRA 2011-29454-C03-02) y al proyecto SEGVAUTO-II (S2009/DPI-1509), financiado por la CAM.

Referencias

[1] Collado J.M., Hilario C., de la Escalera A. and Armingol J.M. Multi-lane Visual Perception for Lane Departure Warning Systems. Third International Conference on Computer Vision Theory and Applications. 2008 .

[2] Flores, M. J., Armingol, J. M., & Escalera, A. (2009). Real-Time Warning System for Driver Drowsiness Detection Using Visual Information. Journal of Intelligent Robotic Systems, 59(2), 103–125.

[3] Frati, V., & Prattichizzo, D. Using Kinect for hand tracking and rendering in wearable haptics. 2011 IEEE World Haptics Conference 317–321 (2011).

[4] Ganganath, N., & Leung, H. Mobile robot localization using odometry and kinect sensor. 2012 IEEE International Conference on Emerging Signal Processing Applications 91–94 (2012).

[5] García, F., Jiménez, F., Naranjo, J. E., Zato, J. G., Aparicio, F., Armingol, J. M., & De la Escalera, A. (2011). Environment perception based on LIDAR sensors for real road applications. Robotica.

[6] Garcia, F., Musleh, B., Escalera, A. D. La, & Armingol, J. M. Fusion procedure for pedestrian detection based on laser scanner and computer vision. 2011 14th International IEEE Conference on Intelligent Transportation Systems ITSC 1325–1330 (2011).

[7] Heo, J., & Savvides, M. Rapid 3D face modeling using a frontal face and a profile face for accurate 2D pose synthesis. , Face and Gesture 2011 632–638 (2011).

[8] Jia, H. J. H., & Xiangfeng, Z. X. Z. Facial feature extraction and recognition based on Curvelet transform and SVD. 2009 International Conference on Apperceiving Computing and Intelligence Analysis 104–107 (2009).

[9] Martí, E. D., Martín, D., García, J., De la Escalera, A., Molina, J. M., & Armingol, J. M. (2012). Context-aided sensor fusion for enhanced urban navigation. Sensors (Basel, Switzerland), 12(12), 16802–37.

[10] Santos, E. S., Lamounier, E. A., & Cardoso, A. Interaction in Augmented Reality Environments Using Kinect. , 2011 XIII Symposium on Virtual Reality 112–121 (2011).

[11] Xia, L., Chen, C.-C., & Aggarwal, J. K. Human detection using depth information by Kinect. , CVPR 2011 workshops 15–22 (2011).

[12] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I–511).

Determinación del estado de batido de la pasta de aceituna empleando visión por computador

D. Martínez Gila, P. Cano Marchal, J. Gámez García, J. Gómez Ortega
Grupo de Robótica, Automática y Visión por Computador
Universidad de Jaén
{dmgila,pcano,jggarcia,juango}@ujaen.es

Resumen

En este trabajo se propone una nueva metodología para controlar el estado de batido de la pasta, basada en la integración de un sistema de visión por computador en el interior de la termobatidora. Su finalidad es obtener determinadas características de la pasta relacionadas con su color y textura; en concreto, las características extraídas han sido: granularidad de la superficie de la pasta, viscosidad y presencia de aceite sobrenadante. Para tal fin se ha empleado el parámetro homogeneidad obtenido de la matriz de co-ocurrencia de niveles de grises (GLCM) y la técnica de segmentación basada en umbral. Además se ha diseñado un controlador difuso basado en el conocimiento experto del maestro, para ajustar de forma eficiente las variables que intervienen en el proceso de batido. Esta ventaja se puede traducir en mejoras del rendimiento industrial, en la calidad del aceite elaborado y en la mejora del consumo energético de la planta.

Palabras clave: Visión por computador, Control de procesos, Aceite de oliva virgen

1. Introducción

Actualmente, el grado de automatización del proceso de elaboración del aceite de oliva virgen se encuentra en una fase preliminar [1]. Particularmente, el control del estado de batido de la pasta lo realiza el almazarero observando periódicamente la pasta en el interior de la batidora y actuando en consecuencia sobre las variables de forma manual. Este modo de actuar depende de la frecuencia con la que el almazarero mira la pasta y de la subjetividad del mismo, lo que provoca poca homogeneidad en la toma de decisiones y falta de robustez.

Controlar automáticamente, en tiempo real, el estado de batido de la pasta, podría permitir ajustar al mínimo el tiempo de batido necesario para su preparación, lo que permite incrementar la capacidad productiva de la instalación. También permitiría la detección rápida de pastas difíciles adecuando el batido a sus características, evitando pérdidas de aceite en el orujo debidas a procesar pasta en condiciones deficientes de preparación. Además, la reducción al mínimo de este tiempo de batido, unida a la optimización del empleo de agua de calefacción, supone una mejora importante en el consumo energético de la instalación.

El subproceso de batido de la pasta de aceituna es, dentro del proceso de elaboración del aceite de oliva virgen, uno de los subprocesos más importante junto con el de centrifugación. El estado inicial de la pasta de aceituna que entra al proceso de batido depende fundamentalmente de las condiciones físicas y químicas de la aceituna molturada. Condiciones que dependen del periodo de recolección del fruto y de la variedad. Un batido incompleto o defectuoso afecta de forma negativa al proceso de extracción posterior, mermando tanto en calidad como en cantidad el aceite elaborado.

Existen trabajos publicados que han estudiado la influencia de las variables que intervienen en el proceso de batido sobre la calidad y el agotamiento del aceite de oliva virgen [4]. En concreto, el aumento del tiempo de batido incrementa el rendimiento en la extracción hasta un cierto umbral temporal, a partir del cual, si se sobrepasa, se pueden ver decrementadas las características nutricionales del aceite elaborado. Así mismo, el incremento de la temperatura de batido tiene efectos positivos en el agotamiento pero puede repercutir de forma negativa en el contenido en polifenoles y en la presencia de componentes volátiles.

Por otro lado, las técnicas de control basadas en lógica difusa ya han sido probadas en la industria alimentaria para supervisar procesos y para alcanzar determinados objetivos de calidad. Estas técnicas permiten modelar el conocimiento experto de los operadores y utilizar este conocimiento como un sensor software para escenarios donde no existe un sensor hardware disponible [7].

La contribución de este artículo está en la implementación de un sistema de control automático del estado de batido de la pasta de aceituna, integrando cámaras de visión por computador en el interior de la termobatidora. Para ello, se preten-



Figura 1: Pasta de aceituna batiéndose en el interior de una termobatidora.

de estimar la cantidad de aceite sobrenadante, la viscosidad de la pasta y la granularidad de su superficie. La presencia de aceite sobrenadante es el indicador mas obvio, ya que este hecho facilitara el proceso posterior de separación; la emulsión de las gotas de aceite con el resto de componentes de la aceituna aumenta la viscosidad de la pasta [3], y este estado es indicativo de pastas poco preparadas o difíciles; la coalescencia de las gotas de aceite provoca el agrupamiento de las partículas de la fase sólida de la pasta, y esto implica que el tamaño de los grumos de pasta formados en su superficie es un indicador del estado de la pasta.

El sistema resultante tiene un doble objetivo: ayudar al maestro almazarero a caracterizar en todo momento la pasta y cerrar el lazo de control que garantiza su estado. A nuestro entender, es la primera vez que se emplea visión por computador para conseguir los objetivos mencionados.

El resto del trabajo se estructura de la siguiente forma: la sección 2 plantea el problema que se trata de resolver; a continuación, en la sección 3 se explica el diseño de la solución propuesta; en la sección 4 se explica el procedimiento seguido para la implementación del sensor de estado de batido de la pasta; por otro lado, en la sección 5 se detalla el diseño del controlador; en la sección 6 se muestran los resultados obtenidos en simulación; y por ultimo, en la sección 7 se comentarán las conclusiones.

2. Descripción del problema

Dentro del proceso de elaboración del aceite de oliva virgen, el batido de la pasta de aceituna es el subproceso posterior a la fase de molienda y el anterior al subproceso de centriguración (véase [3] para una descripción completa del proceso). Su finalidad es conseguir una primera separación de las diferentes fases que constituyen la pasta de aceituna de tal forma que quede preparada para el proceso de extracción posterior. En esta subfase de preparación de la pasta, se completa el cizallamiento de las partes insuficientemente tratadas en la fase de molienda y se incrementa su temperatura para producir la agregación de las gotas de aceite dispersas en la pasta molida, aumentando de este modo la proporción de mosto suelto a partir de mosto normal.

El proceso de batido se lleva a cabo en las máquinas termobatidoras. En la industria de elaboración del aceite de oliva se emplean básicamente dos tipos de termobatidoras, las de palas giratorias sobre eje vertical y las de palas giratorias sobre eje horizontal (Fig. 1), siendo estas últimas las más usadas en la actualidad. Las termobatidoras suelen estar distribuidas en diferentes cuerpos apilados verticalmente, donde el numero de cuerpos depende principalmente de la capacidad de molturación de la linea de extracción. Estos cuerpos están provistos en su interior de un sistema de palas que al mismo tiempo que cumplen su función de batido, conducen de forma gradual a la pasta desde el extremo por donde entra hacia el extremo contrario por el cual sale la pasa a través de un rebosadero. Las paredes de la termobatidora son huecas de forma que permiten la circulación de agua caliente por su interior para producir el calentamiento de la pasta de aceituna. Para aumentar la superficie de intercambio de calor, cada cuerpo de la batidora se suele dividir en varios compartimentos, separados por intercambiadores de calor, por las que circula también el agua caliente. El hecho de que existan placas de intercambio de calor dentro de la batidora consigue que el calentamiento de la masa sea gradual, ya que una variación brusca de temperatura repercutiría negativamente en la calidad final del aceite.

Actualmente la valoración del grado de dificultad de la pasta la estima el almazarero de forma visual observando de forma periódica el proceso de batido en el interior de la termobatidora. Los aspectos en los que se basa para esta valoración son, principalmente, el grado de suciedad de las palas, la granularidad de la pasta y la presencia de aceite flotante en la superficie de la pasta. Así, si la pasta presenta una baja granularidad, las palas de la termobatidora están completamente sucias y no existe aceite flotante sobre la masa, ésta tendrá la consideración de pasta difícil. Ante esta situación y bajo la condición de que el objetivo sea aumentar el rendimiento en la extracción, el maestro actuaría sobre la válvula de entrada de agua caliente para conseguir aumentar su caudal y por consiguiente la temperatura de la pasta. De modo contrario, si la pasta esta muy granulada,

las palas están completamente limpias y la fase oleosa de la pasta se puede diferenciar, la pasta sera considerada como pasta fácil.

Los problemas que trata de solventar este trabajo son todos los inherentes al control manual por parte de los operarios de almazara. Por ello se propone la implementación de un sistema de control del batido de la pasta basado en visión por computador con el que poder determinar en cualquier momento las características de la pasta y actuar de forma automática en base a las necesidades que prevea el maestro de almazara.

3. Solución propuesta



Figura 2: Esquema del sistema de control del estado de batido de la pasta.

La solución que se propone consiste en el diseño de un sistema de control como el de la Figura 2. El sistema consta de dos partes diferenciadas. La primera se basa en la implementación de un sensor software que obtiene las características de la pasta a partir de la imágenes capturadas de la pasta de aceituna. Y la segunda consiste en el diseño de un controlador basado en lógica difusa que a partir de las características obtenidas por el sensor, consiga el estado de elaboración establecido por el maestro de almazara.

En las siguientes dos secciones primero se detallará la implementación del sensor software de estado de batido y posteriormente el diseño del controlador.

4. Diseño del sensor

El diseño del sensor de estado de batido de la pasta se divide fundamentalmente en dos tareas. La primera es hardware y consiste en la instalación de todo el sistema de adquisición de imágenes sobre la máquina. La segunda tarea consiste en el procesado de las capturas y obtención de características. Ambas tareas serán explicadas en los siguientes subapartados.

4.1. Hardware empleado

El despliegue hardware se ha realizado en una de las lineas de producción de aceite de oliva virgen de la almazara San Bartolomé, perteneciente al grupo Sociedad Cooperativa Oleocampo (Jaén, España).

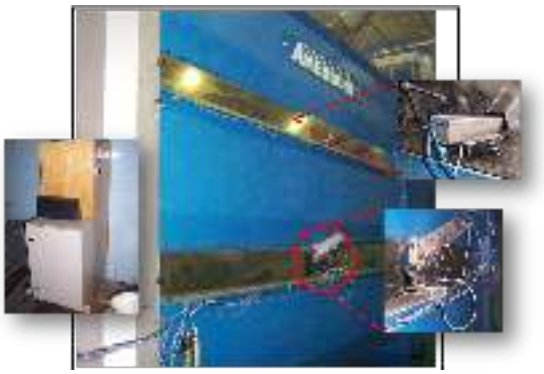


Figura 3: Instalación del sistema de visión por computador en la termobatidora. En ella se puede observar el armario que alberga el PC de procesado y las dos cámaras de visión ubicadas en dos de los tres cuerpos de la termobatidora.

La figura 3 muestra la termobatidora horizontal de tres cuerpos utilizada para la experimentación. En ella se puede observar la instalación de dos cámaras de visión, una de ellas para monitorizar la pasta de aceituna en el cuerpo intermedio de la termobatidora y otra para monitorizarla en el cuerpo inferior. La inexistencia de un sistema de visión en el cuerpo superior se debe a la falta de viabilidad en su ubicación. Por otra parte se instaló un equipo PC para la recepción y almacenamiento de imágenes, ademas de toda la electrónica asociada.

El sistema de visión del cuerpo inferior esta formado por un foco de iluminación halógena de 100W y una cámara de visión multiespectral JAI AD-080 GE con óptica F1.8/6mm. Esta cámara se emplea para capturar información visual de la pasta de aceituna en el espectro infrarrojo y visible. El sistema de visión del cuerpo intermedio esta formado por un foco de iluminación halógena de 100W y por una cámara industrial DALSA Genie C1400 con óptica F1.4/3.5mm. Ambas cámaras están protegidas por medio de un encapsulamiento en acero inoxidable. Dada la temperatura que alcanza la termobatidora, estos encapsulados están provistos de un sistema de ventilación que evita el sobrecalentamiento de las cámaras.

El software de captura de imágenes se ha programado en Python y se ha ejecutado sobre un equipo PC convencional con capacidad suficiente como para almacenar las imágenes capturadas durante

toda la campaña de recolección.

4.2. Granularidad de la pasta

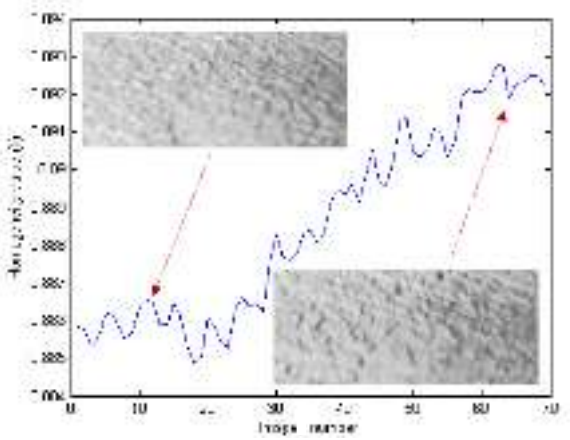


Figura 4: Homogeneidad para una secuencia de imágenes donde la pasta evoluciona desde poco a muy granulada.

Para la determinación de la granularidad de la pasta de aceituna se han analizado las imágenes capturadas por la cámara de visión DALSA Genie C1400. Dado que en este cuerpo el nivel de pasta de aceituna se mantiene por encima de las palas de la termobatidora, la región de interés y el tiempo de muestreo seleccionado es independiente de la presencia de estas palas en las imágenes.

En este caso, para obtener la granularidad superficial de la pasta, primero se transforma la imagen RGB capturada al espacio de color CIELAB [8]. El canal de luminancia L del espacio de color CIELAB [8] aporta para cada pixel de la muestra la luminancia reflejada. Para el área de interés seleccionada, la variación del valor de los píxeles contenidos en el área aporta la información necesaria para aplicar el método de análisis estadístico de texturas basado en la matriz de coocurrencia de niveles de grises GLCM [5] [6].

La matriz de coocurrencia de niveles de grises es uno de los métodos estadísticos mas usado en análisis de texturas [2]. Se basa en la construcción de una matriz $p(k,l)$ que contiene información del número de veces que el par de niveles de grises k y l aparecen en la imagen, en una dirección determinada θ y a una distancia determinada d. Para obtener la matriz GLCM de la pasta de aceituna, primero se ha promediado la matriz para todas las direcciones posibles (0 , 45, 90, 135, 180, 225, 270 y 315 grados) y para una distancia igual a 1. Y segundo, se ha normalizado con la siguiente función:

$$P(k,l) = \frac{p(k,l)}{R} \tag{1}$$

donde R es la suma de todos los elementos de la matriz. A partir de esta matriz de coocurrencia se pueden obtener las características de contraste, correlación, energía y homogeneidad. Para cuantificar la granularidad de la pasta tiene interés el calculo de la homogeneidad. Este parámetro se obtiene con la ecuación 2.

$$h = \sum_k \sum_l \frac{1}{1+|k-l|} P(k,l) \tag{2}$$

El valor h de homogeneidad sera mayor cuanto mayor sea el tamaño de los grumos de pasta, y se verá reducido cuando el tamaño de los grumos sea pequeño, aumentando la varianza de niveles de intensidad de pixel en la región de interés. En la Figura 4 se puede observar el valor de la característica homogeneidad para una secuencia de imágenes capturadas a lo largo de la campaña, donde la textura de la pasta de aceituna evoluciona desde poco granulada (izquierda) hacia muy granulada (derecha).

4.3. Viscosidad de la pasta

La viscosidad de la pasta de aceituna durante el subproceso de batido se puede estimar visualizando la cantidad de pasta que se adhiere a las palas de la termobatidora. De modo que la pasta será mas viscosa cuanto más sucias estén las palas y viceversa. La cantidad de pasta adherida a las palas se ha determinado analizando las imágenes capturadas por el dispositivo de visión JAI AD-080 GE en el espectro del infrarrojo (IR). Se optó por este cuerpo porque es el único en el que las palas se mantienen siempre visibles por encima del nivel de la pasta.

La segunda imagen de la Figura 5 muestra la diferencia entre el índice de absorción de la pasta de aceituna y el del acero inoxidable de la pala, para el espectro del infrarrojo. Esta diferencia contrasta claramente la suciedad de las palas y permite determinarla empleando uno de los métodos de segmentación basado en el histograma de la imagen. En concreto el método de los dos picos. Este método consiste en buscar los dos niveles de intensidad que se repiten con mayor frecuencia en el histograma de la imagen, y obtiene como umbral el valor mínimo que se encuentra entre los dos niveles anteriormente seleccionados. Los píxeles con valores superiores al umbral serán considerados como suciedad de la pala.

En nuestro caso, los histogramas de las imágenes presentan tres máximos. Para aplicar el método

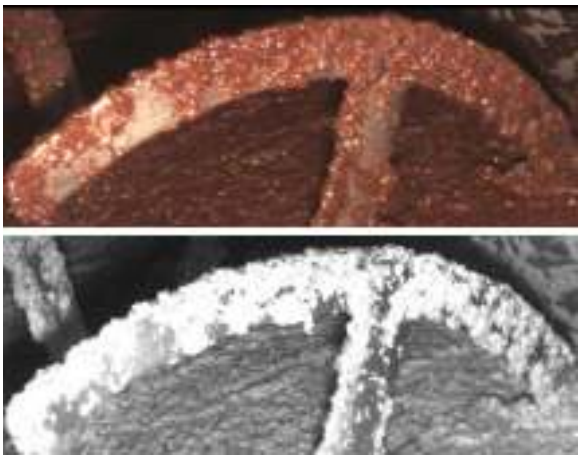
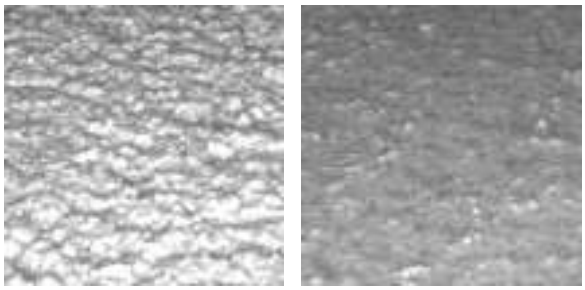


Figura 5: Imagen en la que se puede observar pasta de aceituna adherida a una de las palas de la termobatidora. La figura superior muestra la pala en el espectro visible y la figura inferior la misma pala en el espectro infrarrojo.

descrito, los picos que se seleccionaron fueron los de mayor valor de nivel de píxel. Es decir, el máximo del histograma que indica presencia de pasta adherida y el máximo que indica tonos medios de grises en la imagen.

4.4. Presencia de aceite sobrenadante



(a) Ausencia de aceite sobrenadante (b) Presencia de aceite sobrenadante

Figura 6: Imagen en el espectro del infrarrojo de la superficie de la pasta de aceituna en el interior de la termobatidora. La imagen de la izquierda es una muestra de pasta sin aceite y la de la izquierda con aceite sobrenadante. Se puede apreciar la diferencia de tonos.

Para la detección del aceite flotante sobre la pasta de aceituna se analizaron las imágenes capturadas por el dispositivo de visión JAI AD-080 GE en el espectro del infrarrojo (IR). Se seleccionó una región de interés y un periodo de adquisición de imágenes tal que para un régimen de revoluciones del eje de las palas constante, siempre se obtiene una imagen de la superficie de la pasta de aceituna libre de la presencia de palas.

La Figura 6 muestra dos imagenes, una sin presencia de aceite flotante y otra con presencia de aceite flotante. En esta misma imagen se puede apreciar como la cantidad de luz infrarroja que absorbe la pasta de aceituna difiere en función de la presencia o no del fluido. Esta característica es útil para determinar la cantidad de aceite sobrenadante presente en el área de interés seleccionada.

Para extraer de la región de interés la cantidad de pasta de aceituna que no esta inundada por el aceite sobrenadante, se ha aplicado la metodología de segmentación basada en umbral [8]. El umbral en este caso será un determinado nivel de intensidad de pixel. Serán considerados pasta de aceituna en fase sólida los píxeles con nivel de intensidad superior al umbral, y aceite flotante los píxeles con nivel de intensidad inferior al umbral. El valor del umbral se ha obtenido experimentalmente procesando una serie de 20 imágenes que evolucionan desde prácticamente ausencia de aceite sobrenadante hasta pastas de aceituna inundadas. Este procesado se ha repetido para cada uno de los 256 posibles umbrales, siendo el nivel de intensidad 168 el que mayor gradiente ofrece.

Es de considerar que esta experimentación se ha realizado con agua en vez de aceite de oliva dada la escasa presencia de aceite sobrenadante en las pastas de aceituna de la última campaña de recolección 2012/2013. Este hecho no incide en los resultados obtenidos, pues la respuesta de ambos fluidos a la luz infrarroja contrasta de igual forma con respecto a la pasta de aceituna.

5. Diseño del controlador

Para el diseño del controlador basado en lógica difusa primero se definen las funciones miembro de entrada al controlador (Fig 7). El rango de variación de los parámetros que reciben estas funciones se establecen según las características obtenidas tras el procesamiento de las imágenes comentado en la sección 4. Estos parámetros de entrada al controlador son: granularidad de la pasta, cantidad de aceite sobrenadante y suciedad de las palas de la termobatidora.

El rango de variación de la característica granularidad (G) está relacionado con el rango de variación del índice de homogeneidad de la imagen. Experimentalmente se ha obtenido que para la pasta de aceituna este parámetro oscila entre 0.5 y 1. Por otro lado, para la variable que indica la cantidad de aceite presente en la superficie de la pasta (AS), su rango varía entre 0 % y 100 % de aceite detectado. De esta misma forma el parámetro que representa la limpieza de las palas (LP) varía entre 0 % y 100 %.

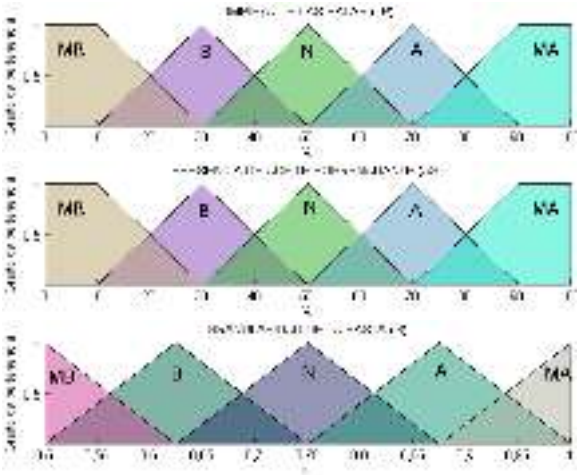


Figura 7: Funciones de pertenencia de las variables de entrada del controlador

Para relacionar la percepción visual humana con los valores de las variables obtenidos teóricamente, se han definido cinco niveles: Muy bajo (MB), bajo (B), normal (N), alto (A) y muy alto (MA).

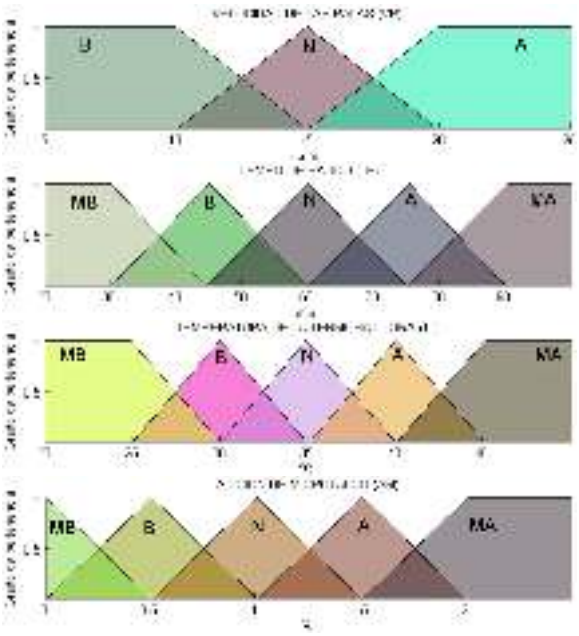


Figura 8: Funciones de pertenencia de las variables de salida del controlador

Las funciones de pertenencia de las variables de salida se muestran en la Figura 8. Los rangos de variación de estas variables se justifican con lo siguiente:

- Referencia de velocidad de las palas de la termobatidora (VP): Esta referencia se transmite al rotovariador de la termobatidora para actuar sobre el régimen de revoluciones. Su valor oscila entre 10 y 20 r.p.m., siendo superior cuando la pasta incrementa en dificultad.

- Referencia de tiempo de batido (TB): Actuando sobre el caudal de entrada y de salida de pasta de aceituna se puede variar el tiempo de batido. Este tiempo oscila entre 20 y 100 minutos en función del tipo de pasta.

- Referencia de temperatura de la pasta (TT): El control de la temperatura de la pasta se puede realizar actuando sobre la servoválvula que regula el caudal de entrada de agua caliente. En función del tipo de pasta y de los objetivos de elaboración, la temperatura de referencia oscila entre los 20 y los 40 °C.

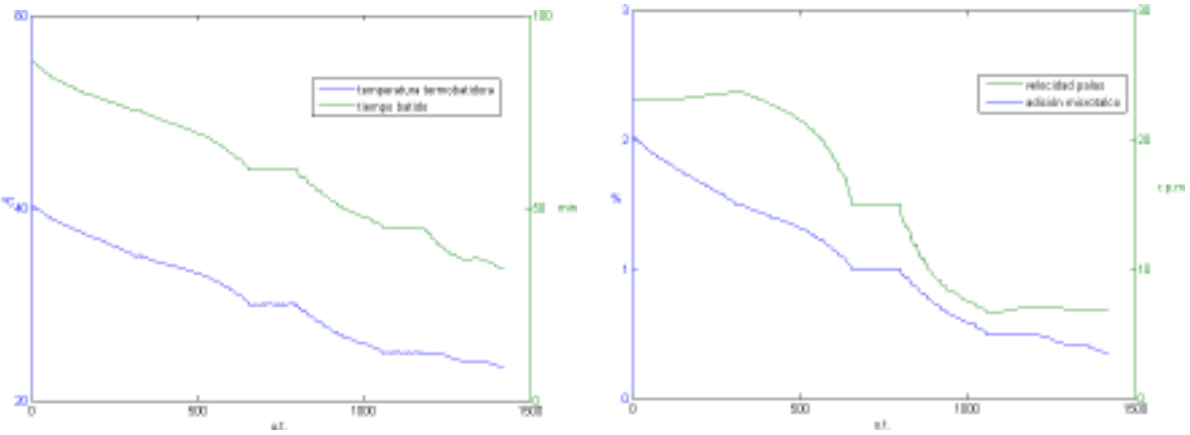
- Referencia de adición de microtalco (AM): La adición de microtalco natural se realiza con una bomba impulsora. Controlando el régimen de revoluciones de esta bomba se puede variar el porcentaje de producto añadido. El rango de variación está comprendido entre el 0 % y el 4 %.

La lógica combinacional del controlador, basada en la experiencia del maestro almazarero, se muestra en las siguientes condiciones:

If :	LP *es* MA & AS *es* MA & G *es* MA
Then :	VP *es* B & TB *es* MB & TT *es* MB & AM *es* MB
If :	LP *es* A & AS *es* A & G *es* A
Then :	VP *es* B & TB *es* B & TT *es* B & AM *es* B
If :	LP *es* N & AS *es* N & G *es* N
Then :	VP *es* N & TB *es* N & TT *es* N & AM *es* N
If :	LP *es* B & AS *es* B & G *es* B
Then :	VP *es* A & TB *es* A & TT *es* A & AM *es* A
If :	LP *es* MB & AS *es* MB & G *es* MB
Then :	VP *es* A & TB *es* MA & TT *es* MA & AM *es* MA

6. Resultados experimentales

Las imágenes de pasta de aceituna capturadas en el interior de la termobatidora en la última campaña de recolección de aceituna 2012/2013, muestran en su mayoría pastas difíciles. Este hecho es debido a la brevedad de la campaña, producida por la poca cantidad de aceituna recolectada. Por lo tanto las características extraídas de estas imágenes no son representativas de todos los estados de la pasta de aceituna que ocurren típicamente en una campaña. Por este motivo se ha optado por emular las variables de entrada al controlador con una posible evolución de los parámetros característicos de la pasta de aceituna, como se muestra en la Figura 10. Las entradas generadas evolucionan desde pastas poco preparadas (palas



(a) Referencia de temperatura y de tiempo de batido

(b) Referencia de velocidad de palas y de adición de microtalco

Figura 9: Evolución de las referencias obtenidas por controlador simulado para una campaña de recolección. Las entradas del controlador son las variables de la Figura 10.

muy sucias, baja granularidad y muy poca presencia de aceite sobrenadante) hacia pastas preparadas (palas limpias, alta granularidad y presencia abundante de aceite sobrenadante).

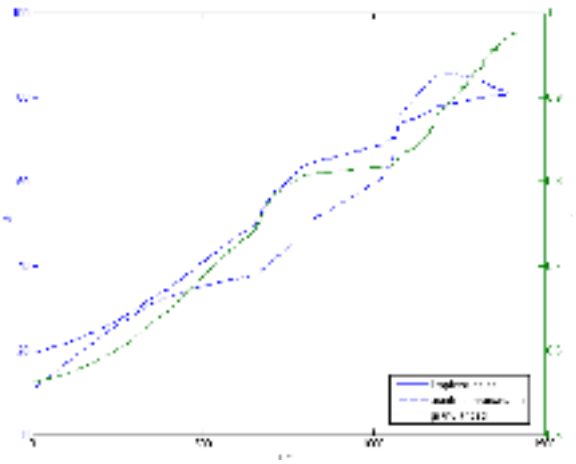


Figura 10: Variables de entrada al controlador.

Las acciones de control obtenidas para las entradas anteriormente comentadas se pueden ver en la Figura 9. Partiendo de pastas difíciles, el controlador obtiene una referencia de temperatura alta, un tiempo de batido elevado (Fig. 9(a)), un régimen alto de revoluciones de las palas de la batidora y una consigna elevada de porcentaje de adición de microtalco (Fig. 9(b)). Por otro lado, se puede observar como con pastas más sencillas, la evolución de las referencias es considerablemente decreciente.

7. Conclusiones y líneas futuras

En este trabajo se ha diseñado un sistema de control del proceso de batido de la pasta de aceituna para ser aplicado a lo largo del periodo de elaboración del aceite de oliva virgen. El lazo de control del sistema se ha cerrado mediante la implementación un sensor de estado de batido, basado en visión por computador, en el que a partir de las imágenes capturadas de la masa en el interior de la termobatidora se obtienen determinadas características de la pasta, como viscosidad, granularidad y presencia de aceite sobrenadante. Estas características determinarán el estado de la pasta en base a la experiencia del maestro de almazara. Además el sistema permite la detección rápida de pastas difíciles adecuando el batido a sus características, evitando pérdidas de aceite en el orujo debidas a procesar pastas en condiciones deficientes de preparación. Por otro lado, el establecimiento de las referencias sobre el proceso de batido optimiza el empleo de agua de calefacción, lo que supone una mejora importante en el consumo energético de la instalación.

Para trabajos futuros se pretende desarrollar un sistema software para la adquisición del conocimiento del maestro y validación online del sistema, utilizando formularios interactivos. Además se desea implementar el controlador simulado en fábrica.

Agradecimientos

Los autores quieren agradecer la subvención parcial de esta investigación a través de los proyectos DPI2011-27284, TEP2009-5363 and AGR-6429. Además nos complace agradecer a Oleocampo S.C.A. por poner sus instalaciones a nuestra

disposición.

Referencias

[1] Cano Marchal, P., Gómez Ortega, J., Aguilera Puerto, D., & Gámez García, J. (2011). Situación actual y perspectivas futuras del control del proceso de elaboración del aceite de oliva virgen. *Revista Iberoamericana de Automática e Informática Industrial RIAI, 8*, 258 - 269.

[2] Chaoxin Zheng, Da-Wen Sun and Liyun Zheng, (2006). Recent applications of image texture for evaluation of food qualities: A review. *Trends in Food Science & Technology, 17*, 113 - 128.

[3] Civantos, L., (1998). Obtención del aceite de oliva virgen. *Editorial Agrícola Española, S.A.*

[4] Clodoveo, M. L. (2012). Malaxation: Influence on virgin olive oil quality. Past, present and future - An overview. *Trends in Food Science & Technology, 25*, 13 - 23.

[5] Haralick, R.M., Shanmugan, K., and Dinstein, I., (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-3*, 610 - 621.

[6] Haralick, R.M. and Shapiro, L.G., (1992). *Computer and Robot Vision: Vol. 1, Addison-Wesley*, p. 459.

[7] Perrot, N., Ioannou, I., Allais, I., Curt, C., Hossenlopp, J., & Trystram, G., (2006). Fuzzy concepts applied to food product quality control: A review. *Fuzzy Sets and Systems, 157*, 1145 - 1154.

[8] Sangwine, S. J., Horne, R. E. N., 1998. The Colour Image Processing Handbook. Springer.

Construcción de mapas topológicos y estimación de trayectorias usando descriptores de apariencia visual global

Francisco Amorós, Luis Payá, Lorenzo Fernández, Óscar Reinoso, Mónica Ballesta y Miguel Juliá
Departamento de Ingeniería de Sistemas y Automática.
Universidad Miguel Hernández. Avda. de la Universidad s/n. 03202, Elche (Alicante), Spain
famoros@umh.es, lpaya@umh.es, l.fernandez@umh.es, o.reinoso@umh.es, m.ballesta@umh.es, mjulia@umh.es

Resumen

La navegación basada en información visual es una fuente de numerosas investigaciones en el campo de la robótica móvil. En este trabajo se presenta un algoritmo de construcción de mapas y localización topológicos. Se utilizan descriptores basados en la apariencia global de imágenes para representar de forma óptima la información visual. Primero, se construye un mapa topológico para representar el espacio de navegación del robot. Los nodos corresponden a posiciones del entorno, y están compuestos por una colección de imágenes que cubren el campo de visión completo con respecto al plano del suelo. Usamos la información proporcionada por una cámara que captura información a través de algunas rutas que pasan entre los nodos para estimar su posición. Una vez construido el mapa, se presenta un sistema de localización que es capaz de estimar la posición del robot no sólo en los nodos sino también en posiciones intermedias usando la información visual. Estos algoritmos han sido evaluados y presentan buenos resultados en ambientes de interior reales.

Palabras clave: Creación de mapas, descriptores visuales basados en apariencia global, zoom artificial, navegación topológica.

1. INTRODUCCIÓN

La navegación autónoma de un robot móvil en un cierto entorno suele requerir de una representación interna del área, es decir, un mapa, que el robot utiliza con el objetivo de estimar su posición usando la información proporcionada por los sensores con los que está equipado. Entre todos los posibles sensores que un robot puede usar para conseguir este propósito, en este trabajo vamos a centrarnos en un sistema de adquisición de información visual. Los sistemas de visión proporcionan una gran cantidad de información del entorno con un coste relativamente bajo. Concretamente, nuestra información de partida son imágenes adquiridas con una cámara con lente de ojo de pez, las cuales abarcan un amplio ángulo de visión del entorno.

Una tarea clave en navegación basada en visión es la asociación entre las imágenes. A través de la comparación entre la imagen capturada en ese momento por el robot y la información almacenada en el mapa del entorno, es posible llevar a cabo la localización del móvil en dicho entorno. Normalmente, esta tarea se logra extrayendo características concretas de las imágenes para crear un descriptor útil de la escena, siendo deseable una baja dimensionalidad que permita usar la información en operación de navegación en tiempo real.

Es posible encontrar dos categorías principales de descriptores: basados en características locales y basados en la apariencia de las escenas. La primera aproximación se basa en la extracción de puntos o regiones significativas de la imagen. Ejemplos muy destacados pueden ser SIFT [6], o SURF [9]. Por otro lado, los descriptores basados en la apariencia global describen las escenas en su conjunto, sin extraer características locales de las mismas. Como ejemplo, [5] usa PCA (Análisis de Componentes principales) para procesar las imágenes y reducir el tamaño de las mismas, y [7] aprovecha las propiedades de la Transformada de Fourier Discreta para aplicarla sobre imágenes panorámicas.

Con respecto a la representación del entorno en un mapa, la bibliografía actual puede ser categorizada en dos estrategias principales: métrica y topológica. Los mapas métricos incluyen información correspondiente a distancias respecto de un sistema de coordenadas definido. En este sentido, es posible encontrar ejemplos como [8], que presenta un sistema basado en sónar para navegación robótica, y [4], que describe una aproximación para resolver problemas de SLAM mediante el uso de un equipo de robots con un mapa representado por la posición tridimensional de puntos característicos de las imágenes. Por el contrario, los mapas topológicos representan el entorno a través de grafos. Los nodos corresponden con características o zonas del entorno, mientras que los ejes representan la conectividad entre nodos. En [3] se muestra un sistema de navegación visual que usa una cámara omnidireccional y un mapa topológico como representación en un entorno interior estructurado.

[2] describe un sistema similar usando una única cámara no omnidireccional.

En [1] se presenta un algoritmo de SLAM llamado FAB-MAP. La descripción de la escena se basa en extracción de puntos característicos. Específicamente, hacen uso de características SURF, y el conjunto de datos experimentales es una extensa colección de imágenes omnidireccionales capturadas en entornos de exterior. El propósito de nuestro trabajo es desarrollar un sistema similar usando una cámara con lente de ojo de pez, más económica que los sistemas catadióptricos o las cámaras esféricas. Otra diferencia reside en la información usada para representar las imágenes, ya que vamos a usar un descriptores basados en la apariencia global de la información visual.

El primer paso de nuestro trabajo consiste en la construcción de un mapa del entorno. Vamos a usar un mapa topológico, en el que cada nodo está compuesto por 8 imágenes que cubren el ángulo de visión completo desde un cierto punto del entorno, mientras que las aristas representan la conectividad entre nodos.

Con el objetivo de estimar las relaciones topológicas entre nodos, se hace uso de información extraída de rutas de imágenes capturadas a lo largo del entorno que pasan a través de los nodos. Como contribución de este trabajo, se aplica un análisis multiescala de zooms artificiales para aumentar la similitud entre las imágenes de los nodos y las de las rutas, especialmente cuando trabajamos con imágenes alejadas de los nodos. De este análisis se obtiene por un lado un aumento de la precisión en el matching de las imágenes de la ruta con las del mapa, y por otro una medida de la posición relativa entre las escenas comparadas.

Una vez se ha construido el grafo que representa el entorno, se ha diseñado un algoritmo para representar la trayectoria del robot en dicho mapa. Este algoritmo también aprovecha la información obtenida con las escalas de los zooms para estimar la posición del robot no sólo a las posiciones de los nodos, sino también en puntos intermedios. También se introduce una función de pesado de las distancias imágenes para mejorar la precisión en la localización.

El siguiente glosario incluye alguno de los términos usados en el texto:

- *Nodo*: Colección de 8 imágenes capturadas en una posición en el plano del suelo con una rotación aproximada de 45° entre imágenes.
- *Base de datos del Mapa*: Conjunto de descriptores de las imágenes de los nodos.
- *Mapa*: Grafo que representa la distribución topológica de los nodos.
- *Distancia Topológica*: Posición relativa entre imágenes o nodos en el mapa.
- *Distancia Imagen* (d): Distancia Euclídea entre descriptores de dos imágenes.

El resto del documento está estructurado de la siguiente forma: La sección 2 presenta las características de la base de datos utilizada en la parte experimental, y el descriptor elegido para representar las escenas. La sección 3 introduce el algoritmo desarrollado para construir el mapa topológico. En la sección 4 se explica el sistema de estimación de la trayectoria, y los resultados experimentales. Finalmente, en la sección 5 se resumen las principales ideas obtenidas en este trabajo.

2. BASES DE IMÁGENES Y DESCRIPTOR

En esta sección se presentan las características de las bases de imágenes usadas para llevar a cabo los experimentos, y el descriptor basado en apariencia global utilizado para describir la información de las escenas.

Las imágenes han sido capturadas usando una cámara con lente de ojo de pez. Se elige este tipo de lente debido a su amplio ángulo de visión. Específicamente, el modelo usado es el Hero2 de GoPro [13]. El ángulo de visión de las escenas es de 127º. Debido al tipo de lente, las imágenes presentan una distorsión que hacen imposible obtener directamente información útil a través de descriptores basados en apariencia global, ya que estos está basados en la distribución espacial y disposición de los elementos en la escena, y la distorsión hace que los elementos aparezcan alterados. Por esta razón, usamos la Toolbox de Matlab *OCamCalib* para calibrar la cámara y corregir la distorsión de las imágenes originales [11]. En el resto del documento, el término *imagen* hará referencia a la escena con la distorsión corregida.

Como nuestro propósito es resolver el problema de localización usando descriptores basados en apariencia global, es necesario usar técnicas que concentren la información visual de la escena en su conjunto, siendo también interesante la robustez antes cambios de iluminación y la capacidad de trabajar con pequeños cambios en la orientación de las escenas. Algunos trabajos, como [10] han comparado el comportamiento de descriptores basados en apariencia global. Tomando estos trabajos en cuenta, se ha decidido elegir el descriptor Gist-Gabor [12], ya que presenta buenos resultados en tareas de búsqueda de correspondencias

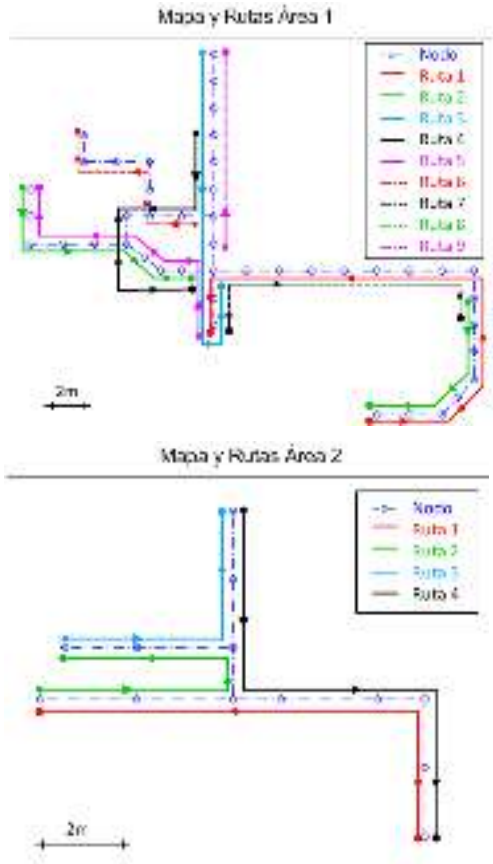


Figura 1: Distribución sintética de los nodos y las rutas en el (a)Área 1 y (b)Área 2.

entre imágenes en entornos de interior. También muestra un coste computacional razonable. Con un tamaño de imagen de 64x32 píxeles, el algoritmo tarda 0.0442 segundos para calcular el descripto de la imagen usando Matlab R2009b sobre un procesador Quad-Core Intel Xeon de 2.8 GHz, con un tamaño de 4.096 bytes de memoria por escena.

La base de imágenes está dividía en dos áreas diferentes. El Área 1 se compone de 44 nodos, mientras que el Área 2 de 13. Los nodos han sido capturados cada 2 metros, pero en lugares en los que se produce un cambio importante de apariencia, como en el cruce de una puerta o en un cambio de dirección, se ha capturado un nuevo nodo independientemente de la distancia con el nodo anterior. Tal y como se ha comentado en la introducción, los nodos se componen de 8 imágenes, capturadas con un desfase de aproximadamente 45º sobre el plano del suelo, cubriendo el ángulo de visión completo.

También se han capturado imágenes a lo largo de rutas en ambas áreas. La información de dichas rutas se usa para encontrar las relaciones topológicas entre los nodos, y con ello construir el grafo que representa el mapa. Con respecto a las rutas, las imágenes se han capturado cada 0.5 metros en el Área 1, y cada 0.2 metros en el Área 2. En los cambios de dirección, se incrementa la frecuencia de captura de imágenes, con un mínimo de 4 imágenes por posición en los giros. En el Área 2, esta frecuencia aumenta hasta un mínimo de 6 imágenes por rotación. En la Fig. 1 se muestra la distribución de los nodos y las rutas en una representación sintética.

Cuadro 1: Número de imágenes por área

	#Imágenes Area1	#Imágenes Area2
Nodos	352	52
Ruta1	110	100
Ruta2	50	72
Ruta3	67	66
Ruta4	58	125
Ruta5	62	-
Ruta6	46	-
Ruta7	69	-
Ruta8	67	-
Ruta9	40	-

3. CONSTRUCCIÓN DEL MAPA

En este apartado se describe el algoritmo desarrollado con el propósito de crear la representación interna del entorno que utilizará el robot durante la navegación. Dicha representación estará basada en un mapa topológico, por lo que el proceso de construcción del mapa consiste en encontrar las relaciones de adyacencia y posición relativa entre los nodos. Para este propósito, se usarán las rutas de imágenes. Así pues, el mapa se construye como un grafo en el cual los nodos representan posiciones del entorno, mientras que las aristas proporcionan información sobre la distancia topológica y la dirección de nodos adyacentes.

No tenemos conocimiento previo ni del orden de los nodos ni de su disposición. Sin embargo, se conoce en qué orden han sido capturadas las 8 imágenes de un mismo nodo, aunque no su orientación absoluta con respecto a un sistema global de coordenadas. Las imágenes de los nodos componen la base del mapa en la cual se buscará la correspondencia de las imágenes de las rutas. Las imágenes de las rutas se consideran ordenadas, por lo que se va añadiendo información al mapa de forma incremental a través de la búsqueda de correspondencias con la base de los nodos.

En este proceso, se utilizan zooms artificiales de las imágenes para incrementar la probabilidad de asociación correcta entre imágenes de los nodos y de las rutas. En la Fig. 2 podemos ver un ejemplo de la disposición de dos escenas de las rutas y las

correspondientes a los nodos más cercanos. La Fig. 2(b) es una imagen de la ruta, la Fig. 2(a) es la imagen de nodo más cercana, y la Fig. 2(a') es una ampliación de la Fig. 2(a). Comparando la imagen de la ruta con la imagen del nodo original y su correspondiente ampliación, podemos percatarnos que se produce un aumento de la similitud entre escenas cuando se amplia la imagen del nodo. Esto es especialmente útil para hallar correspondencias con imágenes de ruta situadas en puntos intermedios entre nodos, en los cuales la similitud entre imágenes se reduce notablemente con respecto a los nodos más cercanos. El empleo de zooms nos permite incrementar la asociación correcta entre las imágenes de los nodos y de las rutas.

En nuestro algoritmo, se usan tanto ampliaciones de las imágenes de los nodos como de las rutas. Dada una escena de una ruta, el algoritmo compara diferentes escalas de ampliación de esa imagen con distintos zooms de las imágenes de los nodos. Tras esa comparación, se elige el caso con una mínima distancia imagen, es decir, una mínima distancia entre descriptores. s_n y s_r son la escala de ampliación de la imagen de nodo y de la ruta respectivamente en las que se obtiene la mínima distancia imagen. La distancia topológica l entre la imagen de la ruta y la del nodo puede ser calculada mediante:

$$l = s_n - s_r \tag{1}$$

Siguiendo con el ejemplo de la Fig. 2, si una imagen de la ruta está delante de un nodo, la mínima distancia imagen corresponderá a la existente entre la imagen original de la ruta Fig. 2(b) y una ampliación de la escena del nodo Fig. 2(a'). En ese caso, obtendríamos un valor positivo de l. Por otro lado, si la posición actual del robot está por detrás de un nodo, la mayor similitud se produce entre una ampliación de la imagen de la ruta Fig. 2(c') y la imagen original del nodo Fig. 2(d), lo que se traduce en un valor negativo de l.

El proceso de construcción del mapa comienza con el cálculo de la base $\mathbf{Z}$ que contiene los descriptores de las imágenes de los nodos, incluyendo distintas ampliaciones de las mismas. Esta base se usará para hallar la asociación entre los nodos y las escenas de las rutas. Por lo tanto, calculamos los descriptores $z^n \in \mathbb{R}^{1\times y}$ de las imágenes de los nodos, siendo y el número de componentes de cada descriptor. Los descriptores se almacenen como columnas de la matriz $\mathbf{Z} = [z_1^n \quad z_2^n \dots z_i^n \dots z_m^n]$, siendo m el número de imágenes incluídas en la base, que corresponden al número de nodos multiplicado por el número de orientaciones por nodo y por el número de ampliaciones por imagen.

Como los descriptores z^n se almacenan en la base siguiendo un orden conocido, es posible hallar el nodo n correspondiente, la orientación θ y la escala de ampliación s_n de cada imagen almacenada en la base conociendo su posición en la matriz $\mathbf{Z}$. Considerando i como el número de columna de $\mathbf{Z}$:

$$[n, \theta, s_n] = f(i). \tag{2}$$

Cuando llega una nueva imagen de la ruta, buscamos la correspondencia con la imagen más similar incluida en el mapa. Para ello, calculamos su descriptor correspondiente, y estimamos la distancia imagen con cada una de las imágenes incluidas en $\mathbf{Z}$:

$$d_i^r = \sqrt{\sum_{a=1}^{y} (z_{i,a}^n - z_a^r)^2}, i = 1 \dots m. \tag{3}$$

d_i^r aporta información sobre la similitud entre imágenes, y se usa como clasificador. El algoritmo selecciona el vecino más cercano, y asocia a la distancia d correspondiente el valor de n, θ y s_n de la imagen de la base seleccionada. Este proceso se repite para diferentes ampliaciones de la imagen de ruta. Obtenidos los resultados para las diferentes s_r, se clasifican los casos seleccionados usando de nuevo d, y se selecciona el caso con una menor distancia imagen. De esta forma, al buscar la correspondencia entre una imagen de la ruta y la base obtenemos un vector de información que incluye los siguientes datos:

$$[n \quad d \quad \theta \quad s_n \quad s_r]. \tag{4}$$

Hasta este punto se ha presentado el proceso de búsqueda de correspondencias entre las imágenes del mapa y de las rutas usando zooms para conseguir un aumento en la precisión y una medida de la posición relativa entre imágenes. Ahora continuamos describiendo el proceso de creación del grafo que representa el mapa topológico. Para ello, se usa la información incluida en los vectores de información (4).

La decisión de añadir un nuevo nodo al mapa tiene en cuenta los resultados del las correspondencias de las últimas imágenes de la ruta. Específicamente, se estudian los últimos cinco vectores de información. Primero se calcula la moda de los nodos más cercanos n_m de dichos vectores. Siendo M el número de veces que se repite n_m, y μ y σ la media y desviación estándar de todas las distancias imágenes d incluidas en los vectores calculados en una cierta ruta hasta esa imagen (no sólo las cinco últimas), se incluye un nuevo nodo en el grafo si una de las siguientes dos condiciones se alcanza:

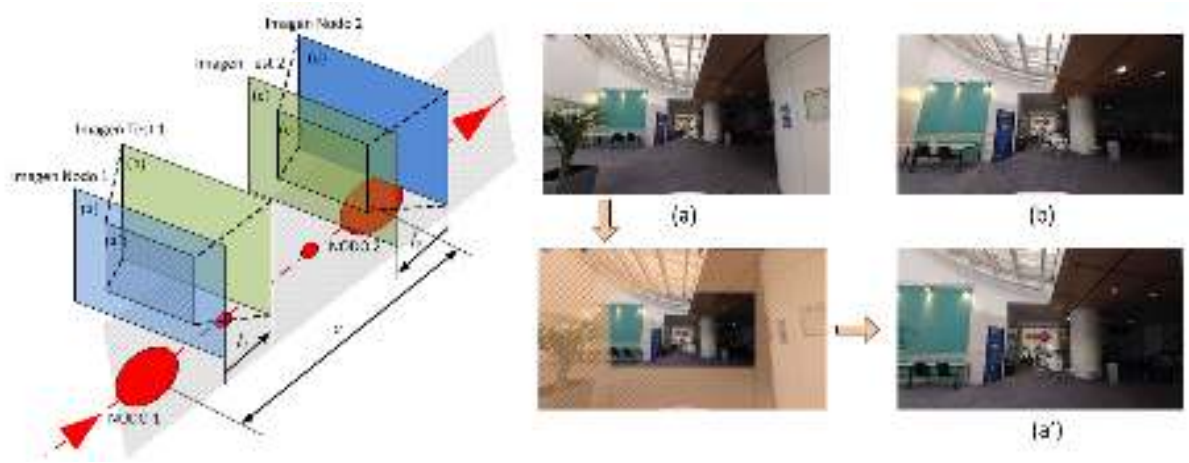


Figura 2: Ejemplos de imágenes de test y de nodos con distintas localizaciones. La imagen de Test 1 tiene el Nodo 1 como nodo más cercano, y se encuentra por delante de él. La imagen de Test 2 tiene como nodo más cercano al 2, y se encuentra por detrás de él. (a) Imagen del Nodo 1, (a') Zoom de la imagen del Nodo 1, (b) Imagen de Test 1, (c) Imagen de Test 2, (c') Zoom de la Imagen de Test 2 y (d) Imagen del Nodo 2. En la parte derecha se incluyen ejemplos de las imágenes (a), (a') y (b) para mostrar cómo se lleva a cabo el proceso de zoom y el aumento de la similitud entre escenas.

- $M \geq 3$
- $M = 2 \quad \text{y} \quad d_{n_m} < \mu - \sigma$

Cuando el vector de información de una asociación de imágenes tiene una distancia imagen $d > \mu + 2\sigma$, no se tiene en cuenta a la hora de valorar la inclusión de un nuevo nodo, ya que un valor demasiado alto de d indica baja fiabilidad en la asociación.

Para estimar las relaciones topológicas entre nodos, se crea la matriz de adyacencia $A \in \mathbb{R}^{N \times N}$, siendo N el número de nodos. A es una matriz dispersa, cuyas filas y columnas corresponden con los nodos del grafo. Con 1 se denota relación de adyacencia entre los nodos correspondientes a la posición matricial en la que se encuentra, y 0 en caso contrario. La distancia topológica entre nodos del grafo se calcula mediante los factores de escala de las ampliación de las escenas. Siendo l^p y l^u las distancias topológicas entre la primera y la última imagen de una ruta en la cual se selecciona un mismo nodo como más cercano, la distancia topológica c entre ese nodo y el siguiente puede calcularse de la siguiente forma:

$$c_{i,i+1} = l_i^u - l_{i+1}^p \tag{5}$$

Para construir el grafo, también es necesario calcular la orientación relativa entre nodos. θ_i^p representa la orientación asociada con la la primera imagen de la ruta en la que se detecta el nodo i, y θ_i^u es la orientación de la última imagen antes que se detecte un nuevo nodo. El desfase de la ruta en ese nodo se calcula como:

$$\Delta\theta_i = \theta_i^u - \theta_i^p \tag{6}$$

La orientación del grafo se fija mediante la definición del ángulo de salida del primer nodo del mapa, que se hace de forma arbitraria. Dicha dirección determinará la orientación del sistema global del mapa. La orientación del grafo se va actualizando en cada nodo mediante $\Delta\theta_i$. Aunque el mapa y los nodos pueden tener sistemas de referencia con diferente orientación, esto no afecta a la construcción del mapa, ya que la actualización de la dirección del grafo se realiza mediante incrementos de ángulo. Una vez definida la orientación del sistema global de coordenadas, es posible definir la diferencia de fase con el sistema local de cada uno de los nodos. Esta información se usará durante la estimación de la trayectoria de las rutas,

Cuando se estudia una nueva ruta, el algoritmo inicializa un nuevo sistema de coordenadas. Dicha ruta se analizará independientemente del grafo global hasta que que se encuentre un nodo común. Usando la posición y orientación del nodo común con respecto a ambos sistemas de coordenadas, el algoritmo es capaz de añadir los nuevos nodos al grafo global de forma correcta. Si la trayectoria de una nueva ruta coincide con una previa ya estudiada, la distancia topológica c entre los nodos se vuelve a estimar, y los resultados se tienen en consideración mediante el cálculo de la media ponderada con las distancias estimadas previamente.

En la Fig. 3 se pueden ver los grafos obtenidos tras la construcción del mapa. En los experimentos, se usa un número elevado de escalas de ampliación con el objetivo de incrementar la precisión en la

estimación de la distancia topológica entre nodos. En concreto, s_n tiene un rango de valores de 1 a 2.5 con un paso de 0.1, mientras que s_r va de 1 a 1.4 con un incremento de 0.05 entre escalas consecutivas. Con esos parámetros, el tiempo empleado por imagen de ruta es de 720 ms.

Es posible apreciar que el algoritmo es capaz de estimar las relaciones de adyacencia entre nodos manteniendo una distribución de nodos en el mapa similar a la real. El Área 1 presenta mayor dificultad en la estimación del mapa debido a su mayor número de nodos, la inclusión de distintas estancias y el bucle en el mapa. El grafo perteneciente al Área 1 presenta alguna diferencia de la disposición real, especialmente en el ángulo de la parte derecha inferior del bucle. En esos nodos se produce el cierre del bucle, y la diferencia angular se produce por el error acumulado en la distancia de los nodos del bucle, no en la estimación del ángulo. Por tanto, como la estimación de la fase entre nodos es correcta, la navegación no se vería afectada a pesar de esa inexactitud en la representación del grafo, pues se conoce la dirección a seguir para llegar de un nodo al siguiente.

Como el algoritmo necesita un número mínimo de correspondencias en las que se encuentra un mismo nodo para incluirlo en el mapa, si no hay suficiente distancia entre nodos adyacentes, o la frecuencia de captura de imágenes en la ruta es baja, un nodo podría no ser incluido en el mapa. El sistema diseñado es especialmente sensible en el cálculo de la orientación del grafo, ya que se basa en el cálculo del desfase entre la orientación de la imagen de entrada y de salida de un nodo, es decir, la orientación de la primera y última imagen de la ruta en la que un nodo es detectado. Por esa razón, es aconsejable aumentar la frecuencia de adquisición de imágenes en los nodos en los puntos de la ruta en los que se produce un cambios de dirección.

4. ESTIMACIÓN DE LA TRAYECTORIA

Una vez creado el mapa del entorno, el siguiente objetivo es estimar la trayectoria de rutas en el mapa. Este problema puede ser abordado como un problema de localización usando la información visual. Sin embargo, si basamos la localización de las trayectoria sólo en la asociación de las imágenes de las rutas con las de los nodos, nuestro conocimiento de la posición del robot se limitará a la posición de los nodos. Nuestro algoritmo mejora la capacidad de localización a puntos intermedios entre nodos del mapa. Para ello, se utiliza el análisis de las escalas de ampliación para llevar a cabo esta tarea, de una forma similar a la descrita en la sección 3.

El algoritmo usa la matriz de adyacencia A y la base de descriptores de las imágenes del mapa $\mathbf{Z}$. La comparación del descriptor z^r asociado a una nueva imagen de ruta con $\mathbf{Z}$ proporciona una medida de su similitud con todas las imágenes de los nodos usando la ecuación (3). Por otro lado, la matriz A contiene información sobre la distancia entre los nodos, haciendo posible encontrar la mínima distancia topológica entre dos nodos cualesquiera del mapa.

Por lo tanto, el primer paso en el proceso de localización es calcular la distancia Euclídea entre el descriptor de la nueva imagen y los de la base, obteniendo d_i^r para $i = 1 \dots m$. Se asocia cada d con el nodo n, dirección θ y escala de ampliación s_n de las base de imágenes. Luego, se clasifican los resultados con respecto a d, y se seleccionan los k vecinos más cercanos. Se repite el proceso para diferentes escalas de la imagen de ruta s_r. Concretamente, en los experimentos llevados a cabo se fija $k = 10$ y s_r adquiere valores de 1 a 1.5 con pasos de 0.1.

Seleccionados los distintos vecinos para cada s_r, usamos una función de pesado de la distancia imagen d usando la posición y orientación asociadas a cada uno. El propósito es penalizar la probabilidad de los vecinos seleccionados que están geométricamente lejos de la última pose conocida del robot. Como nuestro clasificador usa la distancia imagen como criterio, definimos una función que incrementa la d conforme la pose asociada aumenta la distancia topológica o el desfase entre con respecto a la última posición del robot.

Para llevar a cabo este proceso, primero estimamos la distancia topológica entre la posición anterior y la posición asociada a todos los vecinos seleccionados durante el proceso de matching.

Como se ha comentado anteriormente, somos capaces de calcular el camino más corto entre dos nodos cualesquiera del mapa usando A, ya que tenemos un grafo conexo. Siendo $c_{n1,n2}$ el coste asociado al camino entre nodos adyacentes $n_1, n_2 \in A$ definido en (5), y $P_{i,j} = [n_i, \dots n_j]$ la secuencia de nodos del camino más corto que conecta n_i y n_j, el coste $C_{i,j}$ asociado con $P_{i,j}$ puede definirse como:

$$C_{i,j} = \sum_{n_i}^{n_j} c_{n_i,n_{i+1}} \tag{7}$$

Considerando la imagen i-ésima de una ruta, el valor de d de cada vecino seleccionado se actualiza mediante la función de pesado definida de la siguiente forma:

$$d' = d \times [w_1 \cdot C_{i,i-1} + w_2 \cdot \Delta\theta_{i,i-1}] \qquad (8)$$

donde w_1 y w_2 son las constantes de pesado. Concretamente, w_1 pondera la distancia topológica, mientras w_2 pondera los cambios en la orientación. La función actualiza las distancias imágenes de cada uno de los vecinos seleccionados. El término multiplicador se incrementa conforme aumenta la distancia topológica o el desfase con respecto la última posición de la ruta conocida. Esto produce un incremento de la distancia imagen, que reduce la probabilidad del vecino asociado como posición elegida. De esa forma, la función penaliza los cambios importantes en la posición y la orientación entre escenas estimaciones consecutivas en las escenas de las rutas. Hay que tener en cuenta que el termino multiplicador de la función de pesado puede ser distinto para cada vecino seleccionado.

El valor de las constantes de pesado se ha determinado experimentalmente, siendo $w_1 = 0{,}15$ y $w_2 = 0{,}1$.

Una vez la distancia imagen ha sido actualizada, clasificamos los vecinos seleccionados con respecto d', y elegimos el que menor distancia presenta. Como resultado, obtenemos el nodo n más cercano a la posición actual, su orientación θ, y las escalas tanto de la ampliación de la escena del nodo s_n y de la imagen e la ruta s_r. Tenemos que tener en cuenta que el ángulo θ que obtenemos está en el sistema de referencia del nodo, por lo que debe ser corregida con el desfase existente entre el sistema de referencia del nodo y el sistema de referencia global, estimado previamente en la construcción del mapa.

La posición actual del robot se calcula usando el nodo más cercano y la posición relativa entre el nodo y la imagen de la ruta (l) usando (1).

La Fig. 3 muestra ejemplos de rutas estimadas en las dos Áreas consideradas. Los puntos representan las posiciones de las diferentes escenas.

Es posible comparar los resultados con los mapas sintéticos mostrados en Fig. 1. Tal y como se puede apreciar, el algoritmo es capaz de obtener una interpolación en la estimación de la posición en puntos intermedios entre nodos usando la información de las escalas de ampliación de las imágenes. En general, la precisión de la localización en puntos donde se produce un cambio de dirección es menor. Es también importante remarcar que, aunque introducimos una función de pesado, el algoritmo es capaz de encontrar de nuevo la posición correcta aunque la posición anterior fuera errónea, como se muestra en Fig. 3(a) y (c). Los resultados de la cuarta ruta del Área 1(Fig. 3(b)) son también interesantes. Comparando la ruta estimada con la real (Fig. 1(b)), la ruta 4 presenta una trayectoria que no coincide con la distribución de los nodos, y a pesar de ello, el algoritmo es capaz de estimar su posición de forma aceptable.

Así pues, los resultados muestran que nuestro algoritmo es capaz de estimar trayectorias las rutas en puntos intermedios de los nodos, y corregir falsas asociaciones a pesar de utilizar una función de pesado.

5. CONCLUSIONES

En este trabajo se ha estudiado el problema de la construcción de mapa y navegación topológicos usando la apariencia global de imágenes. El algoritmo de construcción del mapa desarrollado es capar de estimar las relaciones entre nodos y crear un grafo usando la información capturada a lo largo de rutas en el entorno. Además, se introducen diversas escalas de ampliaciones artificiales de las imágenes con el propósito de hallar una medida de distancia topológica entre imágenes, la cual puede ser considerada como una medida aproximada de la distancia geométrica proporcional. Los resultados presentan una alta precisión en la búsqueda de correspondencias entre las imágenes de los nodos que forman la base y las imágenes de las rutas. Esto nos permite hallar las relaciones de adyacencia entre nodos y su orientación relativa. Toda esta información se recoge en un grafo que representa el mapa del entorno.

También se ha mostrado un algoritmo de estimación de trayectoria de rutas en el entorno. Este algoritmo se vale también del estudio de escalas de ampliación para mejorar la localización del robot en posiciones intermedias entre los nodos. Además, se mejora la precisión en la localización introduciendo una función de pesado que penaliza los cambios de posición y orientación entre imágenes de consecutivas en una ruta. A pesar del pesado, el algoritmo es capaz de reubicar al robot de forma correcta aunque una localización previa fuera errónea.

Los resultados obtenidos tanto en la construcción del mapa como en la estimación de la trayectoria de las rutas animan a continuar estudiando las posibilidades de la aplicación de descriptores basados en apariencia global a tareas de navegación topológica. Sería interesante extender este estudio para buscar la minimización de la cantidad de información necesaria que debe contener el mapa para una navegación óptima, la aplicación de nuevos descriptores basados en apariencia global, o la mejora en la estimación de la fase para conseguir que el algoritmo corrija pequeños errores en la orientación del robot.

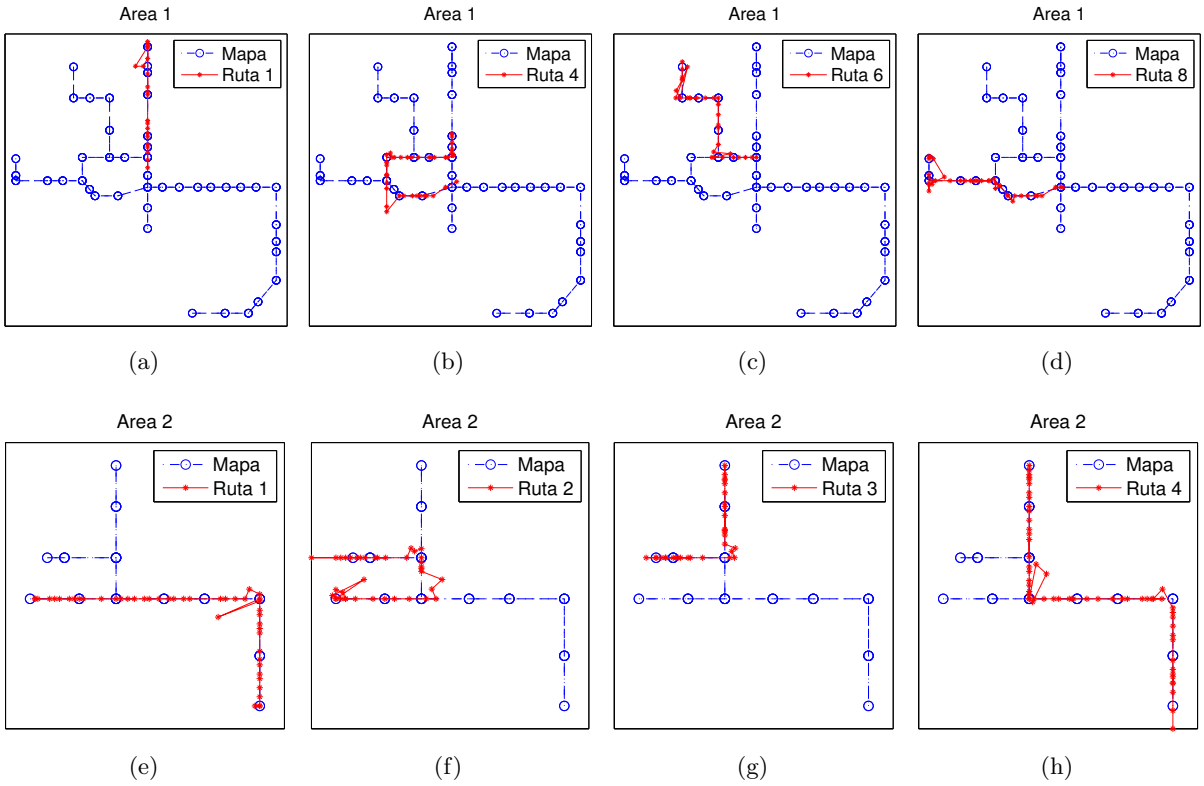


Figura 3: Estimación de la trayectoria de (a) Ruta 1, (b) Ruta 4, (c) Ruta 6 y (d) Ruta 8 del Área 1 sobre el grafo obtenido en la construcción del mapa, y estimación de la trayectoria de la (e) Ruta 1, (f) Ruta 2, (g) Ruta 3 y (h) Ruta 4 del Área 2 sobre el grafo obtenido en la construcción del mapa

Agradecimientos

Este trabajo ha sido apoyado por el gobierno español a través del proyecto DPI2010-15308.

Referencias

[1] Mark Cummins and Paul Newman. Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[2] R. Frizera, Schneebeli H., and J. Santos-Victor. Visual navigation: Combining visual servoing and appearance based methods. 1998.

[3] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omnidirectional camera. *Robotics and Automation, IEEE Transactions on*, 16(6):890 –898, dec 2000.

[4] A. Gil, O.“ Reinoso, M. Ballesta, M. Julia, and L. Paya. Estimation of visual maps with a robot network equipped with vision sensors. *Sensors*, 10(5):5209–5232, 2010.

[5] B. Krose, R. Bunschoten, S. Hagen, B. Terwijn, and N. Vlassis. Visual homing in enviroments with anisotropic landmark distrubution. In *Autonomous Robots, 23(3), 2007, pp. 231-245*, 2007.

[6] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[7] E. Menegatti, T. Maeda, and H. Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251 – 267, 2004.

[8] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 116 – 121, mar 1985.

[9] A.C. Murillo, J.J. Guerrero, and C. Sagues. Surf features for efficient robot localization with omnidirectional images. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3901 –3907, april 2007.

[10] L. Paya, L. Fenandez, O. Reinoso, A. Gil, and D. Ubeda. Appearance-based dense maps creation: Comparison of compression techniques with panoramic images. In *6th Int Conf on Informatics in Control, Automation and Robotics*, 2009.

[11] D. Scaramuzza, A. Martinelli, and R. Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on*, page 45, jan. 2006.

[12] A. Torralba. Contextual priming for object detection. In *International Journal of Computer Vision, Vol. 53(2), 169-191*, 2003.

[13] Inc Woodman Labs. http://gopro.com/hd-hero2-cameras/, 2013.

OBJECT RETRIEVAL UNDER DIFFERENT ILLUMINATION DIRECTIONS USING INVARIANT LOCAL FEATURES

Gorgojo[1], N., Fernández-Robles[2], L. Alegre[3], E.
[1,3] Department of Electrical, Systems and Automatic Engineering
[2] Department of Mechanical, IT and Aerospatial Engineering
Campus de Vegazana S/N 24071, University of León, León, Spain.
nuria.gorgojo@gmail.com,{l.fernandez, enrique.alegre}@unileon.es

Abstract

In this paper we analyzed several retrieval methods, using an image dataset and obtaining results in the form of hit probabilities. The most efficient method for retrieving images under different illumination angles is determined in this work. In this way, we can observe how the illumination affects the object retrieval. We present an analysis of the object retrieval methods based on invariant local features. Invariant local features are explained, and in particular, Surf, Sift and Cosfire methods are described together with the metrics that can be used to measure reliability. Moreover, similarity and distance measures applied for object recognition are presented. In order to show the efficiency of the studied methods, an experiment with a set of 192 images under different illumination angles belonging to 8 different objects is carried out, and it is proven that Sift performs better than the other two methods.

Key Words: ILF, SIFT, SURF, COSFIRE, illumination direction, object retrieval.

1 INTRODUCTION

Currently, computer vision is gaining much prominence. More and more systems and methods are able to recognize objects in images or videos. To carry out this object recognition, it is necessary to compute matches among objects on the different images, and in many cases lighting, orientation or other invariances should be taken into account.

Object retrieval is used to designate those methods and techniques used to search for an object in a collection of images or videos so that these methods return the images in which it is considered that the object is located.

The process is as follows [4]:

- Training, detection and description of interest points of the pattern image.
- Detection and description of interest points of test images.
- Calculation of matches between descriptors: matching.
- Decision on whether the set of points detected in test images belong to the pattern image or not.

The retrieval of objects has many applications in a wide range of fields as diverse as industry, security, medicine or robotics, among others.

For the goal of retrieving objects, we have decided to use local features. While global features are too rigid to represent an image, local features extract a set of features for each keypoint and its neighborhood. Thus obtaining, for example, increased robustness to occlusions of the object.

We evaluated SIFT descriptor [7] since it was the first method that got invariance for the local features approach. Also SURF descriptor [2], an improved SIFT method, that can achieve the same results by accelerating the process. Finally, we have also tested COSFIRE [1] method since this new approach achieves a good performance from a different perspective.

The objective of this work is to evaluate the performance of the above methods for object retrieval under different lighting conditions. We are not aware of previous works that have a detailed study with appropriate metrics, such as precision at n or average precision, of the effect of changes in illumination of objects for retrieval evaluation.

The rest of the paper is organized as follows: Section 2 outlines the dataset used. Section 3 explains the methods SIFT, SURF and COSFIRE. Section 4 details the evaluation metrics for object retrieval. Finally, we show the experimental results in section 5 whereas conclusions will be explained in section 6.

2 DATASET

For this Project we have chosen the database "Amsterdam Library of Object Images" (ALOI) which is available at http://staff.science.uva.nl/~aloi/.
ALOI is a collection of a thousand images of small objects, recorded for scientific purposes.

It contains various images, which vary the viewing angle, illumination angle, and illumination color for each object, together with some wide-baseline stereo images. It contains more than a hundred images of each object, giving a total of 110,250 images.

We have chosen the set of images with different illumination directions. In this case each object varies in 24 configurations as the example in Figure 1:

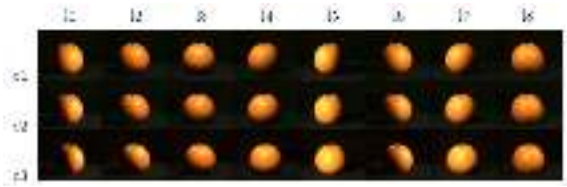


Figure 1. Illumination conditions.

Each object is photographed with one out of five different lights turned on, producing five different lighting angles (L1-L5). At positions c1, c2, c3 is made a small change, changing 15 and 30 degrees in the direction of light. This results in 15 different illumination angles.

Turning on two lights at the sides of the object yielded an oblique illumination from right (L6) and left (L). Finally, turning on all lights yields positions L8. In total, 24 different lighting conditions were obtained as shown in Figure 1.

Specifically, for this project we used "full color", "illumination direction" dataset from where we chose eight objects (objects number 32, 50, 51, 63, 64, 447, 752, 814) shown in Figure 2.


Object1


Object2


Object3


Object4


Object5


Object6


Object7


Objetct8

Figure 2. Groups of objects used for retrieval.

3 DESCRIPTION OF THE IMAGES AND MATCHING PROCESS

We evaluated three methods: SIFT, SURF and COSFIRE, each with recommended metrics.

3.1 SIFT

Using local features is the most effective method of visual recognition. The first person that used it was Christoph von der Malsburg through oriented Gabor filters at different scales on the same graph.

It was later when David Lowe mixed these scale features with SIFT scale [7].

SIFT stands for Scale-Invariant Feature Transform. It provides an approach to look for distinctive local keypoints in the image and describe them. Through these local features, we seek invariance to scale, orientation, illumination changes, etc. It is usually applied to find matches between different views of the same scene.

This algorithm is based on four stages described by David Lowe:

Detection of maxima and minima in the space scale: Search on different scales and dimensions of the image to identify possible interesting points, which are invariant to changes in orientation and scaling, using Difference of Gaussian function.

Location of the key: To select the final interest points, a measure of stability is applied on them in order to discard those that are not suitable.

Assigning orientation: Orientations are assigned to each point of interest extracted from the image based on local directions in the image gradient. The following operations are performed with the data processed according to the orientation, scale and location allocated at this stage, providing invariance.

Interest point descriptor: They represent the interest points as a measure of local image gradients in the neighborhood of these points. Each interesting point comprises a feature vector of 128 elements,

which provides shape deformations and lighting changes invariance.

SIFT descriptor is constructed from Gaussian scale space of the original image, which can effectively detect the positions of the interesting points, invariant to changes in image scale.

The algorithm uses the DoG (Difference of Gaussian) formed from the resulting scalar Gaussian. This function *D(x,y,σ)* is obtained by subtracting subsequent scales in each octave:

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (1)$$

where k is a constant multiplicative scale factor. To detect local maximum and minimum of each point of the image D (x, y, σ), this value is compared with its 26 neighboring points, 8 nearest neighbors in D image and 9 neighboring points in each of the upper and lower level images. If the value is higher or lower than all its neighbors, it identifies the point as a local maximum or minimum.

The following figure shows the SIFT keypoints of one of the images:

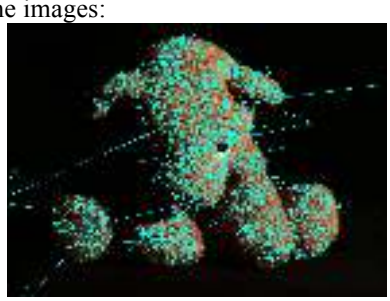


Figure 3. SIFT keypoints.

The matching between each pair of keypoints is performed using the Euclidean distance between their feature vectors. This is useful in order to determine which pair of points in the input image belong to its counterpart in the query image.

Figure 4 shows the matches between two images with different objects.

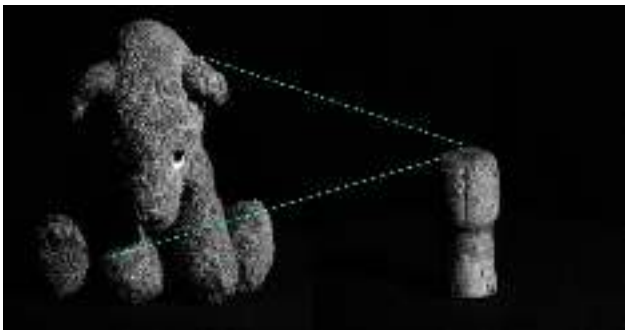


Figure 4. SIFT matches.

3.2 SURF

Bay et al. [2] developed SURF descriptor, speeded-Up Robust Features, as an interest points detector and robust descriptor. The SURF descriptor bears some similarity to the SIFT descriptor, although there are significant differences. Bay et al. assert that this detector and descriptor has three main differences (improvements) with respect to SIFT are:

- The length of the feature vectors of the interest points is smaller. 64-dimensional feature vectors, which means a reduction in length by 2.
- SURF descriptor always uses the same image, the original without scaling it.
- It uses the determinant of the Hessian matrix to estimate the position and the scale of the interesting points, simplifying the computing process.

As to the last point, SURF descriptor uses the value of the determinant of the Hessian matrix to obtain the location and scale information of the interest points. The reason for using this matrix is the higher performance in terms of computing speed and precision.

Figure 5 shows the most significant SURF interest points for the same object shown in Figure 4 with SIFT.



Figure 5. SURF keypoints.

The method used to find matches among images is the same as with SIFT descriptor, nearest neighbor technique.

3.3 COSFIRE

COSFIRE filter, Combination Of Shifted Filter Responses, proposed by Azzopardi and Petkov [1], uses the offset feature of simple filters responses. Supports of Gabor filters are combined for different locations to obtain a sophisticated filter.

The function used to combine the filter response is weighted geometric mean primarily the multiplication, which has specific advantages over

the recognition of shapes and robustness to contrast variations.

Due to the multiplicative nature of the output function, a filter COSFIRE produces a response only when all the constituent parts of the interesting pattern are present.

The following example shows the main idea of the method. Figure 6a shows an input image containing three vertices. The rounded apex, which is shown enlarged in Figure 6b, is considered as a interesting pattern to automatically configure a COSFIRE filter.

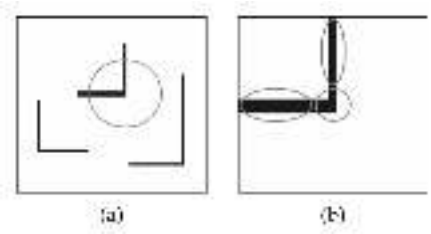


Figure 6. COSFIRE configuration.

The two ellipses shown in Figure 6b represent the dominant orientations in the vicinity of specified interest point. Lines are detected by symmetrical Gabor filters. The central circle represents the superposed supports of a group of filters. COSFIRE response descriptor is calculated by combining the responses of Gabor filters in the centers of the respective ellipses by multiplication. The preferred orientations of these filters and the locations of the responses are determined by analyzing the pattern used for COSFIRE filter settings.

In our case, taking the midpoint of the image, the function trace 5 concentric circles to that point, searching points that are local maximum to describe the object from the information in these points and to create the filter.

In some images, the object is not in the center, so the function traces these circles and it does not find any maximum. We could draw the larger circles to find maximum, but the filter would be poorly defined, so this option was discarded. Instead, we cropped the images so that the object is in the center of the image.

Figure 7 shows an example of the COSFIRE Filter:



Figure 7. COSFIRE filter configuration.

4 RESULT METRICS

To evaluate the performance of the three methods we used Average Precision and Precision at N [10].

The level of accuracy can be measured by the precision at different cut points. In an object retrieval system, hit-n objects are the first n objects in the ranking. Precision at N or P @ n is the ratio of the hit-n relevant objects. The value of n can be chosen based in assumption that the user will see. For example, if the first 10 objects are relevant and not the next 10 in a P @ 10, it would have an accuracy of 100%, while a P @ 20, would be 50%.
Precision at n is defined by:

$$Precision\ at\ n = \frac{r}{n} \qquad (2)$$

Where n is the selected section, and r is the number of relevant documents.

Average Precision is a measure which combines the recovery and precision of results. It is the average of the accuracy scores after retrieving relevant objects.

$$Average\ Precision = \frac{\sum_{j=1..n}(P@j * rel(j)}{R} \qquad (3)$$

Where R is the total relevant objects and rel (j) will be one is if the document j is 1 and 0 otherwise.

In this work we compute the distance (or likelihood) between two images and rely on the minimum distance (or maximum likelihood). This value is considered as the strength of the matching. In case of SIFT and SURF, it will be the strongest matching between a pair of keypoints whereas for COSFIRE we look for the filter in every pixel of the image and take the maximum strength over all pixels.

So, taking for example the first image of Object 1, and computing the matches with all the 192 images, the strength between the query object (Object 1) and each image in the dataset will be stored. By sorting the images from highest to lowest strength, we get a ranking. In order to achieve perfect precisions, all images labeled as containing the query object should have been retrieved in the first positions of the ranking.

In this experiment we found the precisions at different hits, P @ n. We have calculated P@5, P@10, P@15, P@20 and finally P@24, which is the maximum number of images in each kind of object. We also calculated the Average Precision for each of the 192 objects.

Besides, the average probability of success for each of the 8 objects was computed, and finally the general average of each method. In this way we can provide results about which the best method or which object is best recovered with each method.

5 RESULTS

In this section we present the results of the different methods used. SURF was computed with different distance metrics: Surf SAD (Sum of Absolute Differences), Surf SSD (Sum of Squared Differences) and Surf Normx (Normalized cross-correlation). Then we computed SIFT and two versions of COSFIRE: Cosfire Max (Max blurring [1]) and Cosfire Sum (Sum blurring [1]).

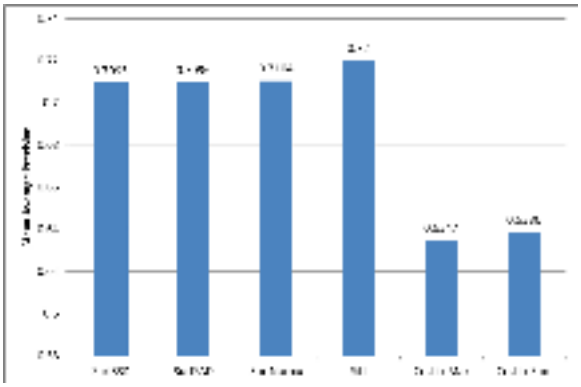


Figure 8.Mean Average Precision

Figure 8 shows the mean Average Precision for each method. It is observed that all methods are in the range 63-72% of Average Precision, Cosfire Sum and Cosfire Max perform worst achieving only a 63%.

Tables 1, 2 and 3 show the obtained results for precisions at 10, 24 and average respectively. The results marked in bold show the best methods for retrieving each object, and the results marked in red and underlined highlight the best-recovered objects for each method.

Table 1. Precision at 10.

	P@10					
	SurfSSD	SurfSAD	SurfNormx	Sift	CMax	Csum
Objeto1	0.81	0.93	0.88	***0.97***	0.58	0.92
Objeto2	0.75	***0.88***	0.83	0.78	0.83	0.78
Objeto3	0.86	0.92	0.89	***0.96***	0.09	0.03
Objeto4	***1.00***	0.97	0.98	0.81	0.68	0.57
Objeto5	0.70	0.85	0.79	0.74	0.87	***0.90***
Objeto6	0.91	0.91	0.92	1.00	***1.00***	0.99
Objeto7	0.81	***0.95***	0.89	0.91	0.83	0.80
Objeto8	0.74	0.78	0.75	0.84	***0.96***	0.96

In P@10 some methods reach 100% precision, as in Surf SSD with Object 4 or Cosfire Max with Object 6. The only method that is not the best for retrieving any object is Normx Surf. Objects 4 (yellow sponge) and 6 (large cup) are the best recovered.

Table 2. Precision at 24.

	P@24					
	SurfSSD	SurfSAD	SurfNormx	Sift	CMax	CSum
Objeto1	0.55	0.58	0.59	0.65	0.48	***0.65***
Objeto2	0.49	0.55	0.56	0.47	***0.57***	0.52
Objeto3	0.59	0.69	0.67	***0.73***	0.08	0.03
Objeto4	***0.93***	0.61	0.73	0.63	0.50	0.59
Objeto5	0.60	0.62	0.58	0.61	0.75	***0.80***
Objeto6	0.81	0.82	0.82	***0.94***	0.87	0.65
Objeto7	0.57	0.65	0.63	***0.66***	0.59	0.62
Objeto8	0.56	0.59	0.59	0.51	***0.66***	0.63

In P@24 the probabilities have dropped, only reaching around 60% average precision for some cases. Object 6 (large cup) is the best retrieved with four methods.

Table 3. Average Precision.

	Average Precision					
	SurfSSD	SurfSAD	SurfNormx	Sift	CMax	CSum
Objeto1	0.63	0.66	0.64	0.75	0.51	***0.79***
Objeto2	0.57	***0.66***	0.63	0.53	0.63	0.61
Objeto3	0.68	0.75	0.74	***0.80***	0.13	0.10
Objeto4	***0.98***	0.66	0.77	0.72	0.60	0.56
Objeto5	0.65	0.68	0.63	0.69	0.83	***0.87***
Objeto6	0.89	0.89	0.90	***0.97***	0.94	0.77
Objeto7	0.68	0.72	0.71	***0.76***	0.69	0.71
Objeto8	0.60	0.66	0.66	0.53	**0.74**	0.70

Finally, the Average Precision shows that Sift is the best method in most cases, achieving 97% for Object 6 (large cup). The Object 4 (yellow sponge), 5 (pink sponge) and 6 (large cup) are the ones that get better retrieved in any of the cases. Object 3 (cork) gets very low scores on Cosfire cases but not in others.

Figure 9 and 10 show the Average Precision with different methods and objects.

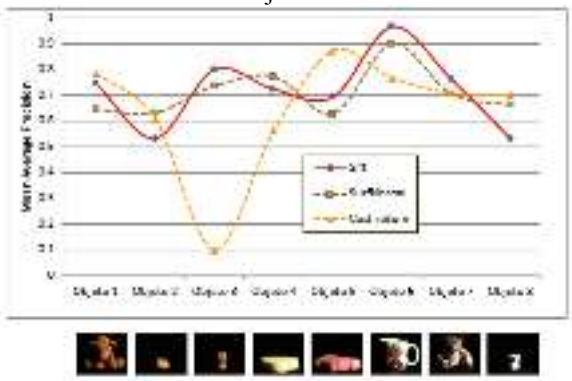


Figure 9. Average Precision of groups of objects with Sift, Surf Normx and Cosfire Sum.

Figure 9 shows Sift, SurfNormx and CosfireSum methods. Sift and Surf are fairly regular for all objects, however Cosfire Sum, has a very low probability in the Object 3 (cork). It can be seen that for poorly defined objects Cosfire filters do not perform well. In the rest of the cases Cosfire works similarly to Sift and Surf.

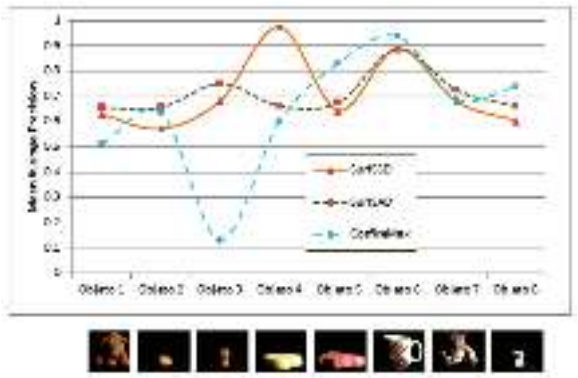


Figure 10. Average Precision of groups of objects with Surf SSD, Surf SAD and Cosfire Max.

Figure 10 shows the results for Surf SSD, Surf SAD and Cosfire Max for each of the 24 views of Object 1. As in the previous graph, Cosfire Max do not recognize well the Object 3, however, it recognizes the Object 6 with a high probability. The other two methods are quite more regular. Surf SSD method has very high probability, practically 100% in the Object 4 (yellow sponge).

Figures 11 and 12 show the graph of one of the individual objects as an example, Object 1. The images of the x-axis charts are ordered from best to worst retrieved, to see the results more clearly.

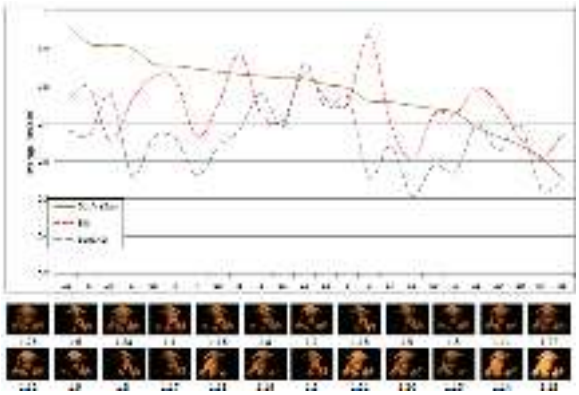


Figure 11. 24 Average Precision of each object in Object 1 with Cosfire Sum, Sift and Surf SAD.

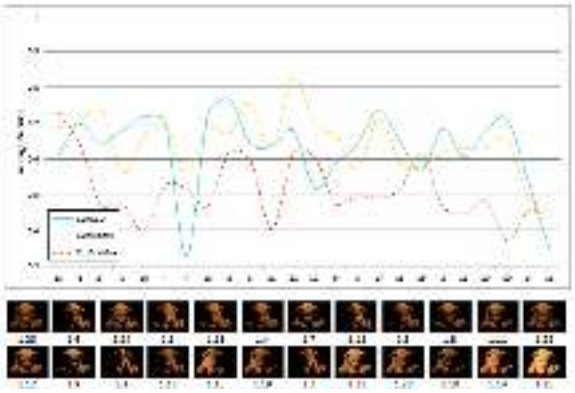


Figure 12. Average Precision of each object in Object 1 with Surf SSD, Surf Normx and Cosfire Max.

In this case, the best method is Cosfire Sum, and the worst one Cosfire Max. From the ranking of images, we see that the clearest images are the worst recovered.

We can affirm, in general terms, that the best descriptor is SIFT, because it gets the best average precision. If it was not for Object 3, COSFIRE would have outperformed SIFT. So COSFIRE can be very useful for well defined objects.

The three Surf approaches have been fairly even, with little better results Normx compared to the other two, Surf SAD and Surf SSD.

Looking to the best-retrieved objects, the Object 6 is the easiest to retrieve (large cup), probably because the representative characteristics of the drawing that contains the cup.

Finally, with respect to the illumination, in many cases clear images are not retrieved correctly. On the contrary, a more neutral illumination is more suitable for our task.

6 CONCLUSIONS

In the present, local invariant features are a widely used tool for object retrieval. In this paper, we have made a study and evaluation of some descriptors based on local features.

Three methods were chosen, SURF and SIFT descriptors, quite similar, since SURF is based on SIFT, and COSFIRE filters, broadening the evaluation.

Our experiment helps to understand which of these methods is most suitable for retrieving object under different illumination directions, not leading to a single solution. There are some specific objects in which some of the methods are efficient while others

achieve a low probability. Future work could be to use a combination of the three methods, to obtain a higher probability of success.

As regard to the illumination, one of the characteristics in which the methods SIFT and SURF are partially invariant, in most cases the items have been retrieved regardless of the illumination. However, in some others, the clearer images have taken up the last positions in the ranking of retrieved images.

This project has verified that the descriptors based on local features suitable for object retrieval.

Acknowledgement

This work has been supported by the ASASEC (www.asasec.eu) European Project with reference HOME/2010/ISEC/AG/043 and by the grant DPI2012-36166 from the Spanish Government.

References

[1] Azzopardi, G., Petkov, N.: 'Trainable COSFIRE Filters for Keypoint detection and pattern recognition' *IEEE Transactions on pattern analysis and machine intelligence Vol.35*, 2013.

[2] Bay, H., Tuytelaars, T., and Van Gool, L.: 'Surf: Speeded up robust features'. *European Conference on Computer Vision*, 2006.

[3] Boullosa, O. 'Estudio comparativo de descriptores visuales para la detección de escena cuasi-duplicadas' [En línea], 2011. http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20110318OscarBoullosa.pdf [Consulta: mayo de 2013]

[4] Enebrail Gonzalez, J. 'Detección y asociación automática de puntos característicos para diferentes aplicaciones' [En línea], 2009 http://upcommons.upc.edu/pfc/bitstream/2099.1/8052/1/memoria.pdf [Consulta: mayo de 2013]

[5] Gomez Diaz, R. 'La evaluación en recuperación de la información' [En línea], 2003, http://www.upf.edu/hipertextnet/numero-1/evaluacion_ri.html#6 [Consulta: mayo de 2013]

[6] Lowe, D.: 'Distinctive image features from scale invariant keypoints'. *International Journal of Computer Vision*, 2004, 2, pp. 91–111.

[7] Lowe, D.: 'Object Recognition from Local Scale-Invariant Features', Computer Science Department, University of British Columbia, 1999.

[8] Pedroza, H.: 'Sistema de análisis estadístico con SPSS', Instituto Nicaragüense de Tecnología Agropecuaria (INTA), 2006.

[9] Tuytelaars, T., and Mikolajczyk, K.: 'Local Invariant Feature Detectors: A Survey'. *Foundation and Trends in Computer Graphics and Vision*, 2007, 3, (3), pp. 177–280.

[10] Zhang, Y.,Wang, X.: 'Demostration of Precision and Recall', University of Pittsburgh [En línea], 2001,http://www2.sis.pitt.edu/~ir/Projects/Sum01/StudentPageSum01.html [Consulta: mayo de 2013]

Seguimiento de objetos sin marcas fiduciales con cámaras ToF

S. Satorres Martínez, J. Gómez Ortega, J. Gámez García, A. Sánchez García, J. de la Casa Cárdenas
Grupo de Robótica, Automática y Visión por Computador
Universidad de Jaén
{satorres, juango, jggarcia, asgarcia, casacar}@ujaen.es

Resumen

Una línea de investigación activa en el ámbito de la robótica manipuladora es el desarrollo de algoritmos de seguimiento de objetos. Disponer de un entorno controlado, o incluir marcas de referencia en los objetos a seguir, simplifica, en gran medida, la tarea de control visual. En este artículo, se presenta un procedimiento para la localización y el seguimiento de objetos sin marcas de referencia. La información de la escena es obtenida mediante el procesado de medidas de profundidad proporcionadas por una cámara de tiempo de vuelo. Se ha implementado un controlador predictivo para controlar la posición relativa entre el efector final del robot y el objeto. En la validación experimental de la propuesta de control visual se ha utilizado una plataforma basada en un robot Meka.

Palabras clave: ToF, segmentación, control en robots manipuladores, MPC.

1 INTRODUCCIÓN

El desarrollo de algoritmos de seguimiento de objectos es un tema de investigación de creciente interés en aplicaciones de control visual dentro de la robótica manipuladora. Se conoce como control visual a la tarea de guiado de un robot utilizando información obtenida a partir de un sistema de visión [1]. Atendiendo a la naturaleza de las características utilizadas, como entrada a la ley de control, es posible distinguir entre dos tipos de control visual: el basado en posición y el basado en imagen [2]. En el basado en posición, la señal del error es obtenida comparando la localización de un objeto, estimada en el espacio Cartesiano, con la deseada; mientras, que en el basado en imagen, la señal de error se define como la diferencia entre ciertas características de un objeto, medidas de la imagen, y las deseadas. No obstante, la matriz de interacción, utilizada por este último tipo de sistema, requiere de la información de profundidad de las características 2D extraídas de la imagen.

En la mayoría de los sistemas de control visual la información 3D de la escena es obtenida a partir de los resultados del procesado de imágenes 2D [3]. Sin embargo, en los últimos años, las cámaras ToF se presentan como una interesante alternativa a las técnicas estándar de obtención de medidas de profundidad [4]. Con las cámaras ToF es posible adquirir, de manera simultánea, información de profundidad y de intensidad a una elevada velocidad de captura [5]. Estas características hacen que las cámaras ToF comiencen a utilizarse en control visual. Por ejemplo en [6], la información de profundidad se integra en la matriz de interacción para un control visual basado en imagen; mientras que en [7], se presenta un sistema 3D de bajo coste, basado en el principio de triangulación para obtener la imagen de profundidad, para un control en posición de un manipulador.

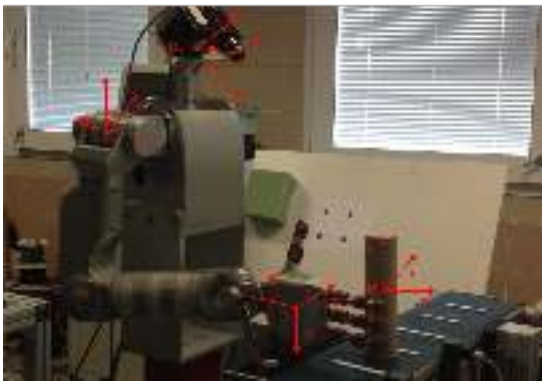


Figura 1: Plataforma de experimentación.

Otro aspecto relevante para los sistemas de control visual es la robustez y la estabilidad. Para garantizar un comportamiento estable y convergente en este tipo de sistemas, trabajos recientes han utilizado técnicas de Control Predictivo basado en Modelo (MPC). Por ejemplo en [8], se proponen tres arquitecturas distintas de control predictivo para un robot de 6-d.o.f. con configuración de cámara en el extremo del robot; mientras que en [9], se emplea esta técnica de control en un sistema robótico sujeto a restricciones mecánicas y de visibilidad.

La principal contribución de este trabajo radica en el desarrollo de un método de procesado de la imagen de profundidad capaz de localizar objetos sin marcas de referencia o fidu-

ciales. Esta propuesta será de utilidad en aplicaciones de control visual de seguimiento de objetos en robots manipuladores. Además, se ha implementado un controlador predictivo para asegurar que el robot antropomórfico sigue a la referencia evitando obstáculos fijos incluidos en la escena. La validación de la propuesta se realiza en una plataforma de experimentación que incluye un robot antropomórfico Meka (Figura 1).

La organización de artículo es como sigue. La sección 2 trata todos los aspectos relevantes en el sistema de visión 3D. En la sección 3 se describen los componentes principales del controlador predictivo. Los resultados experimentales para la validación del controlador se presentan en la sección 4. En la última sección se exponen las conclusiones del trabajo.

2 OBTENCIÓN DE LA INFORMACIÓN 3D

El cometido del sistema de visión 3D es la localización del objeto a seguir en el sistema de referencia mundo. En esta sección se analiza cómo la cámara ToF obtiene la información de profundidad, la calibración de la misma y el procesado de la imagen que proporciona las medidas de profundidad.

2.1 PRINCIPIO DE FUNCIONAMIENTO

La obtención de la medida de profundidad está basada en el conocido principio de tiempo de vuelo [10]. Un sistema de iluminación LED—con una longitud de onda cercana al infrarrojo e incluido en la cámara ToF—emite luz que es reflejada por los objetos de la escena y captada, a su vez, por el sensor de la ToF. El desfase entre la luz emitida y la recibida se utiliza para estimar la medida de profundidad. De esta forma, la distancia radial en cada uno de los píxeles del sensor, en el sistema de referencia de la cámara, ${}^{C}d$, se obtiene a partir de:

$$ {}^{C}d = \frac{c \cdot \varphi}{4 \cdot \pi \cdot f_m} \tag{1} $$

donde c es la velocidad de la luz en el vacío, f_m es la frecuencia de modulación de la luz emitida y φ es el desfase muestreado en cuatro puntos equiespaciados en cada periodo de la señal ($m_0(0^o)$, $m_1(90^o)$, $m_2(180^o)$, $m_3(270^o)$) calculados de la siguiente manera:

$$ \varphi = \arctan(\frac{m_3 - m_1}{m_0 - m_2}) \tag{2} $$

Los datos de amplitud (A), que representan la calidad de la medida de profundidad, también se entregan de manera simultánea para cada píxel como:

$$ A = \frac{\sqrt{[m_3 - m_1]^2 + [m_0 - m_2]^2}}{2} \tag{3} $$

2.2 CALIBRACIÓN

La exactitud y la precisión en las medidas de profundidad de una cámara ToF dependen, en gran medida, del procedimiento de calibración de la misma. En dicho procedimiento se consideran tres tipos distintos de parámetros: intrínsecos, extrínsecos y de profundidad. Los dos primeros son los que habitualmente se estiman en la calibración de una cámara 2D y el tercero permite corregir los errores en las medidas de profundidad ofrecidas por la cámara ToF. Se enumera, a continuación, la metodología seguida en este trabajo para el cálculo de cada uno de ellos.

- Parámetros intrínsecos:

  Los fabricantes de ToF suministran métodos de calibración válidos para la totalidad del rango de funcionamiento de la cámara [11]. Por regla general, ofrecen funciones para convertir las medidas radiales a coordenadas Cartesianas—expresadas en metros—en el sistema de referencia de la cámara. Estas transformaciones suelen incluir una compensación para las distorsiones radiales provocadas por la lente. En este trabajo se han utilizado las funciones del fabricante en la obtención de dicha conversión.

- Parámetros extrínsecos:

  Estos parámetros se determinan a partir de la imagen de amplitud y considerando el procedimiento de calibración como un problema de control visual basado en imagen [12]. Por tanto, mediante esta propuesta de calibración, se tratará de minimizar el error entre las distancias reales de las características observadas en el patrón de calibración y las distancias de las mismas calculadas por retroproyección, $\mathbf{p}$, a partir de los parámetros intrínsecos y extrínsecos actuales, como:

  $$ \mathbf{p} = pr_{\xi}({}^{C}\mathbf{P}) = pr_{\xi}({}^{C}M_w \, {}^{w}\mathbf{P}) \tag{4} $$

  donde $pr_{\xi}(\cdot)$ es el modelo de proyección correspondiente a los parámetros intrínsecos ξ, considerados como constantes y obtenidos a partir de las funciones del fabricante de la cámara ToF, ${}^{C}M_w$ es la matriz homogénea de 4×4 que relaciona el sistema de referencia mundo y el sistema de referencia cámara, ${}^{C}\mathbf{P}$ son las posiciones de las características

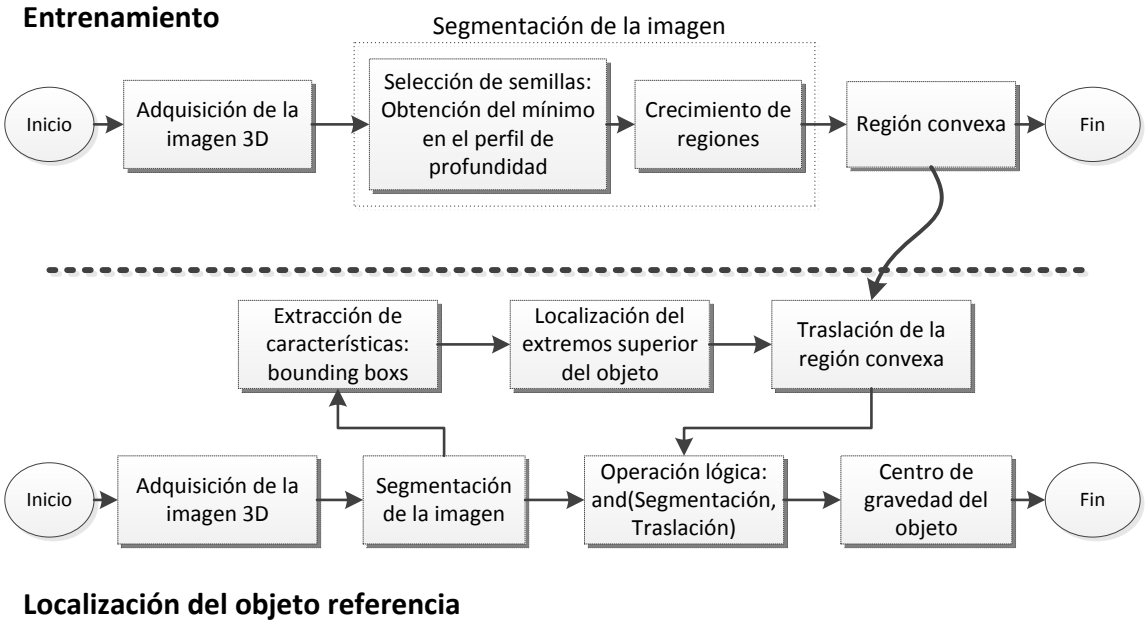


Figura 2: Esquema seguido para el procesado de la imagen de profundidad para la obtención de las coordenadas 3D del objeto referencia.

en el sistema de referencia cámara y ${}^{w}\mathbf{P}$ es el valor de estas características en el sistema de referencia mundo. Para asegurar la minimización del error, la cámara virtual se desplaza—inicialmente en ${}^{Ci}\mathbf{M}_{w}$—utilizando la ley clásica de control visual. Cuando se consigue la minimización del error, los parámetros de la cámara virtual serán ${}^{Cf}\mathbf{M}_{w}$.

- Corrección de las medidas de profundidad:

  La estimación de la medida de distancia está influenciada por el tiempo de integración—o tiempo que requiere el sensor de la cámara ToF para recibir la señal de luz reflejada—y por el índice de reflexión de los objetos de la escena. Será fundamental evitar que los píxeles del sensor estén saturados o capten poca información lumínica. Mediante el ajuste adecuado del tiempo de integración se evita la saturación y se consigue un compromiso entre el ruido de la imagen y la velocidad de adquisición. El error relacionado con el tiempo de integración es uno de los cinco tipos de errores sistemáticos, o aquellos que pueden ser corregidos mediante calibración, identificados en [5].

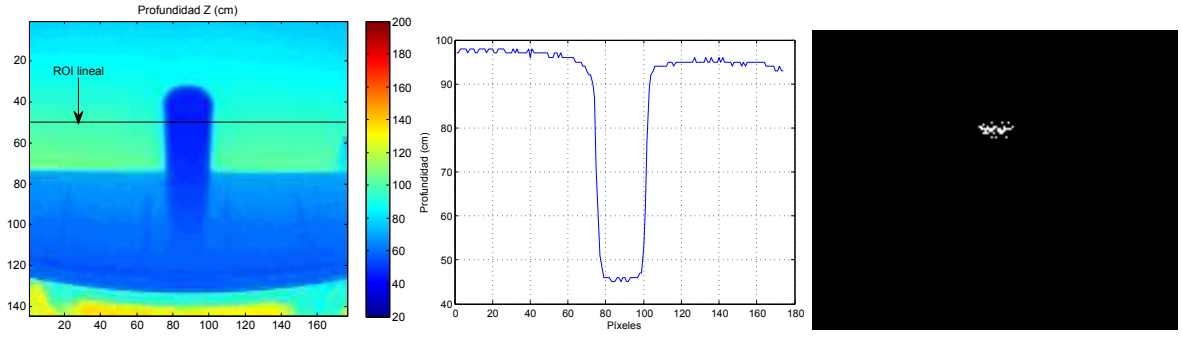
  Puesto que en la aplicación propuesta la medida de distancia en el eje Z—del sistema de referencia cámara—permanece prácticamente invariante, el tiempo de integración se ajusta al inicio del experimento y permanece constante durante la ejecución de la tarea.

2.3 PROCESADO DE LA INFORMACIÓN 3D

La figura 2 muestra las etapas llevadas a cabo en el procesado de la imagen para la obtención de las coordenadas 3D del centro de gravedad del objeto a seguir. Tal y como puede observarse, la localización del objeto de referencia está basada en una fase previa de entrenamiento. El objetivo de esta fase será el de extraer información acerca del contorno 2D del objeto empleado como referencia—región convexa—. Dicha información es fundamental en la fase de seguimiento del objeto en tiempo real ya que, una vez finalizada la segmentación de la imagen, tanto el objeto como el robot pueden ser extraídos del fondo.

Con el fin de eliminar cualquier región distinta al objeto de referencia, se realiza una operación lógica "and" entre la imagen segmentada y la región convexa del objeto aprendido, realizando una traslación previa de la misma a la localización en la cual se encuentra el objeto. La localización del extremo superior del objeto de referencia es inmediata, al disponerse de un conocimiento previo de la altura y anchura del mismo. De esta forma, será posible localizar al objeto durante la tarea de seguimiento, incluso cuando el brazo robótico aparezca en el campo de visión de la cámara.

La figura 3 muestra la imagen de profundidad utilizada durante la fase de entrenamiento. En ella, solamente el objeto de referencia y la cinta trans-



(a) Imagen de profundidad para la fase de entrenamiento. (b) Perfil de profundidad de la ROI. (c) Selección de semillas.

Figura 3: Fase de entrenamiento

portadora aparecen en la escena. El objeto es localizado correctamente posicionando una región de interés lineal, ROI, en la parte superior del mismo. El valor mínimo de este perfil de distancias se utiliza como semillas en el proceso de segmentación de la imagen.

Tanto en la fase de entrenamiento como en la de operación en tiempo real se utiliza el mismo algoritmo de segmentación. Se trata de un método convencional basado en el crecimiento de regiones pero aplicado a la imagen de profundidad. Tal y como su nombre indica, este método agrupa píxeles o subregiones en regiones de mayor tamaño basándose en un criterio predefinido de crecimiento [13]. Partiendo de un conjunto de "semillas", obtenidas del perfil de profundidad (Figura3(c)), el algoritmo divide los datos de profundidad en n regiones, $R_1, R_2, ..., R_n$, de tal forma que:

1. $\bigcup_{i=1}^{n} R_i = R$.
2. R_i es una región conexa, $i = 1, 2, ..., n$.
3. $R_i \bigcap R_j = \emptyset$ para todos i y j, $i \neq j$.
4. $P(R_i) = TRUE$ para $i = 1, 2, ..., n$.
5. $P(R_i \bigcup R_j) = FALSE$ para cualquier región adyacente R_i y R_j

Donde, $P(R_i)$, es el predicado lógico que deben satisfacer todos los píxeles de la región. Por tanto, un píxel con un valor de profundidad z_k se incluye en la región R_i si satisface:

$$|z_k - S| < T \tag{5}$$

siendo S el valor de profundidad de la "semilla" y T es un umbral seleccionado empíricamente. La figura 5 muestra los resultados del procesado de la imagen demostrando cómo es posible localizar el objeto de referencia incluso cuando el brazo aparece en el campo de visión de la cámara (Fig.4).

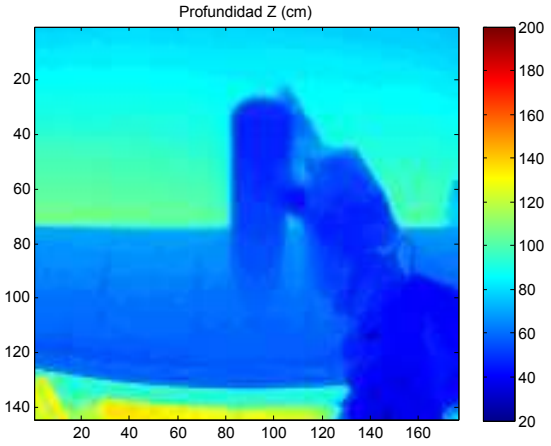


Figura 4: Imagen de profundidad.

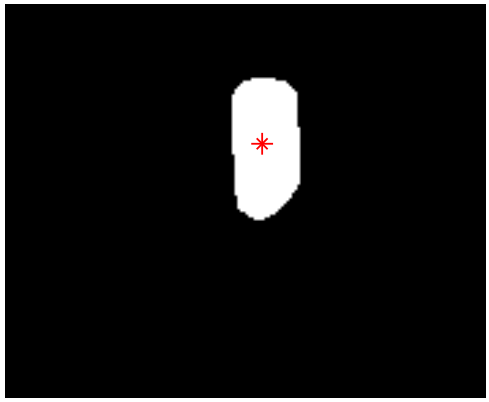


Figura 5: Localización del objeto.

La metodología propuesta es una manera robusta y eficaz de localizar y seguir cualquier tipo de objeto previamente aprendido. Sin embargo, es necesario asegurar que en el campo de visión de la cámara ToF no aparezcan, de manera simultánea, objetos con la misma geometría que el objeto aprendido. Tal y como se mostrará en la sección de resultados el seguimiento del objeto es 3D.

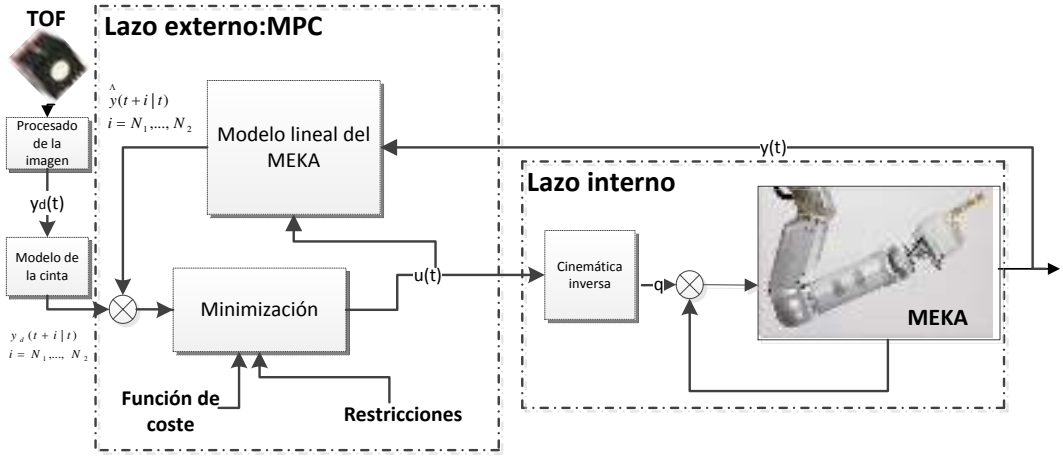


Figura 6: Estructura del controlador predictivo.

3 CONTROL PREDICTIVO BASADO EN MODELO

Un esquema del controlador predictivo implementado se presenta en la figura 6. Teniendo en cuenta el periodo de ejecución, se distinguen dos lazos de control: interno y externo. El lazo de control interno es el más rápido y su periodo de muestro es de $1ms$. Para este lazo, dada la futura posición del efector final del robot en el espacio Cartesiano y en el sistema de referencia mundo—$u(t)$—, se determinan los ángulos de cada articulación, q, utilizando la cinemática pseudo inversa del manipulador. Mediante el modelo cinemático se establece la relación entre el movimiento de cada articulación del brazo y la posición del efector final o mano del robot, $y(t)$.

El lazo de control externo implementa el controlador predictivo, el cual obtiene los valores $u(t+i|t); i = 1, ..., N_u$ mediante la minimización de la función de coste. Al utilizar la técnica conocida como *horizonte deslizante*, solamente se aplicará la primera variable de control de la secuencia calculada $u(t)$, desechando el resto.

Mediante el procesado de la imagen de profundidad ofrecida por la cámara ToF es posible localizar la posición deseada del objeto a seguir, $y_d(t)$. La futura evolución de la referencia, $y_d(t + i|t); i = N_1, ..., N_2$, se obtiene teniendo en cuenta la posición actual y las posiciones previas del objeto y la velocidad de la cinta transportadora sobre la que éste se encuentra ubicado y se desplaza.

Finalmente, haciendo uso de la posición actual del efector final—$y(t)$—y del modelo lineal del robot, se predicen las salidas futuras del sistema $\hat{y}(t + i|t); i = N_1, ..., N_2$. En las siguientes subsecciones se describen los elementos principales del controlador predictivo.

3.1 IDENTIFICACIÓN DEL SISTEMA

El modelo del robot ha sido obtenido mediante modelado experimental y utilizando la System Identification Toolbox™ de Matlab. En la etapa de selección del modelo, se analizaron varios modelos matemáticos: ARX, ARMAX, OE, BJ y espacio de estados. Finalmente, la mejor correlación de los datos se obtuvo con el modelo Output Error (OE); este modelo se define como:

$$w(t) + f_1 w(t-1) + ... + f_{n_f} w(t-n_f) = b_1 u(t-1) + ... + b_{n_b} u(t-n_k-n_b+1) \quad (6)$$

$$y(t) = w(t) + e(t) \quad (7)$$

donde $w(t)$ es la salida sin perturbación, $y(t)$ es la salida en el instante t, $f_1 ... f_{n_f}$ y $b_1 ... b_{n_b}$ son los parámetros a estimar, n_f es el número de polos de la función de transferencia que relaciona la entrada, $u(t)$, y la salida sin perturbación, $n_b + 1$ se corresponde con el número de ceros en la función de transferencia anterior, n_k es el número de muestras de entrada que tienen lugar antes de que las entradas afecten a la salida, $u(t-1) + ... + u(t-n_k-n_b+1)$ son las entradas previas de las que depende la salida y $e(t)$ es una perturbación de tipo ruido blanco.

Puesto que el modelo OE es sólo válido en sistemas SISO, ha sido necesario realizar una identificación para cada uno de los ejes cartesianos. La figura 7 muestra una comparativa para el eje X entre la trayectoria de referencia, la real y la ofrecida por el modelo, validándose, de esta forma, los modelos OE identificados.

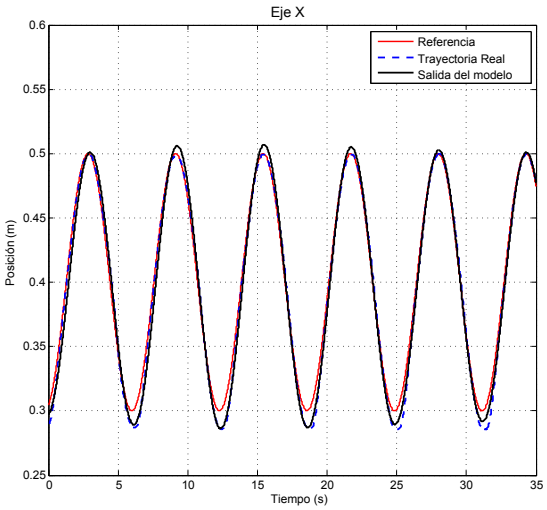


Figura 7: Trayectorias en el eje X utilizadas para la identificación del modelo lineal del robot.

3.2 FUNCIÓN DE COSTE

La función de coste a minimizar, J, es una función cuadrática de la suma de los errores futuros en el seguimiento de la referencia, predichos hasta un determinado horizonte de predicción N, así como las señales de control que se prevén para el futuro, hasta un horizonte de control N_u, expresándose como:

$$J(N, N_u) = \sum_{i=1}^{N} \delta[\hat{y}(t+i|t) - y_d(t+i)]^2 + \quad (8)$$

$$\sum_{i=1}^{N_u} \lambda[\Delta u(t+i-1)]^2$$

$$s.a.$$

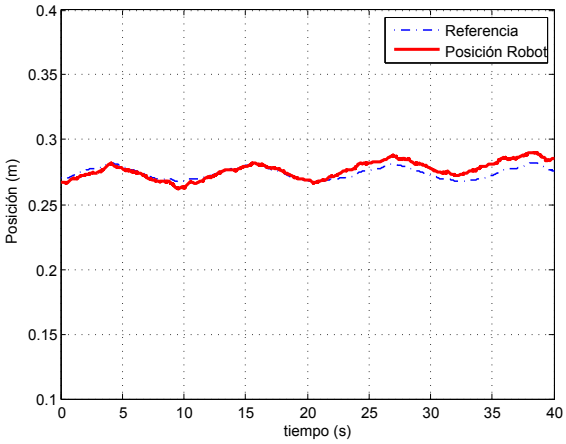
$$u_{min} \leq u(t) \leq u_{max}$$

donde:

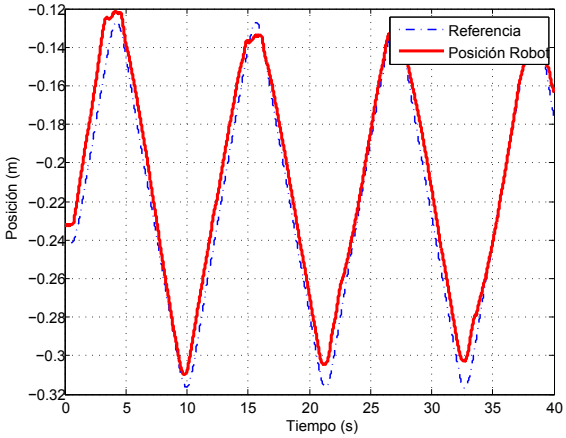
- δ and λ son los pesos de ponderación.
- $\hat{y}(t+i|t)$ son las salidas futuras del sistema, es decir, las posiciones 3D futuras del efector final del robot, predichas a partir del modelo de robot.
- $y_d(t+i)$ son las salidas futuras deseadas o, lo que es lo mismo, la posición futura del objeto a seguir.
- $\Delta u(t+i-1) = u(t+i) - u(t+i-1)$ es la secuencia futura de esfuerzos control.

La minimización de la función J está sujeta a ciertas restricciones que deberán ser tenidas en cuenta. En este sentido, y para evitar colisiones en el movimiento del brazo, se ha incluido la ubicación de obstáculos fijos que puedan aparecer en la escena, como por ejemplo la posición de la cinta transportadora empleada para mover el objeto a seguir.

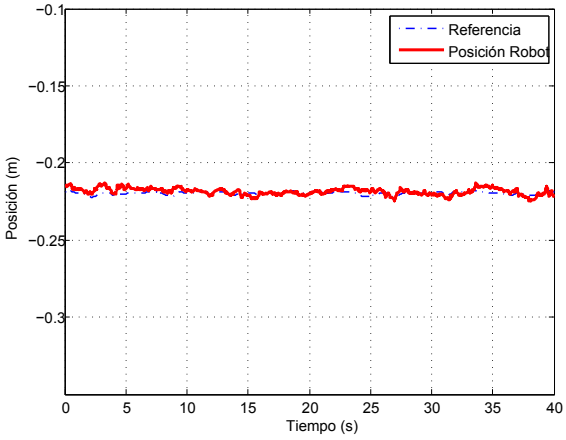
4 RESULTADOS



(a) Comparativa de posiciones para el eje X.



(b) Comparativa de posiciones para el eje Y.



(c) Comparativa de posiciones para el eje Z.

Figura 8: Resultados del MPC.

A fin de llevar a cabo pruebas reales se ha utilizado la plataforma robótica mostrada en la figura 1. Dicha plataforma está constituida, principalmente, por un robot antropomórfico Meka de 7-d.o.f. y una cámara ToF SR4000, ubicada en la parte superior del torso del robot. La cámara ToF tiene una resolución de 176 × 144 píxeles con un

rango de medida de hasta $10m$. La exactitud en la medida está en torno a $+/-15mm$ y la precisión es de $6mm$ dependiendo, ambos parámetros, de la distancia de trabajo y del índice de reflexión de los objetos incluidos en la escena. El objeto a seguir es de un material no reflectante y está posicionado sobre una cinta transportadora con inversión del sentido de giro.

El experimento seleccionado, para validar el diseño del controlador predictivo, es el seguimiento del objeto de referencia mientras éste efectúa un movimiento de vaivén.

La figura 8 muestra, para cada uno de los ejes del sistema de referencia mundo, O_w, una comparativa entre la posición de referencia, $y_d(t+i|t); i = N_1, ..., N_2$, y la posición del efector final del robot $y(t)$. Tal y como puede observarse el error máximo entre la referencia y la posición del robot se alcanza en el eje Y cuando la cinta invierte su sentido de giro siendo de $20mm$, en el peor de los casos.

5 CONCLUSIONES

En este artículo se ha presentado un procedimiento para la localización y el seguimiento de objetos, sin marcas fiduciales, empleando una cámara de tiempo de vuelo. La propuesta de procesado de la imagen de profundidad permite extraer únicamente el centro de gravedad del objeto de referencia, incluso si otros elementos, como por ejemplo el manipulador, apareciesen en la escena 3D. El guiado del robot, en el seguimiento de la referencia, se realiza mediante un controlador predictivo. Como se ha comprobado experimentalmente, la estrategia de control planteada y los algoritmos de procesado de la imagen de profundidad permiten realizar un control visual satisfactorio sobre el robot antropomórfico Meka.

Agradecimientos

Los autores quieren agradecer la subvención parcial de este trabajo a través de los proyectos DPI2011-27284, TEP2009-5363 y AGR-6429.

Referencias

[1] S. Hutchinson, G.D. Hager and P.I. Corke. *A tutorial on visual servoing control*, IEEE Transactions on Robotics and Automation, vol.12, no.5, pp.651-670, 1996.

[2] F. Chaumette anf S. Hutchinson. *Visual Servo Control. Part I: Basic Approaches*, IEEE Robotics and Automation Magazine, vol.13, no.4, pp.82-90, 2006.

[3] E. Marchand and F. Chaumette. *Feature tracking for visual servoing purposes*, Robotics and Autonomous Systems, vol.52, pp.53-70, 2005.

[4] U. Reiser, J. Kubacki. *Using a 3D time-of-flight range camera for visual tracking*, IFAC Proceedings Volumes, vol.6 (PART 1), pp. 355-360, 2007.

[5] S. Foix, G. Alenyà and C. Torras. *Lock-in Time-of-Flight (ToF) Cameras: A Survey*, IEEE Sensors Journal, vol.11, no.9, pp.1917-1926, 2011.

[6] J. Pomares, P. Gil and F. Torres. *Visual Control of Robots using Range Images*, Sensors, vol.10, pp.7303-7322, 2010.

[7] I. Siradjuddin, L. Behera, T.M. McGinnity and S. Coleman. *A position based visual tracking system for a 7 DOF robot manipulator using a Kinect camera*, IEEE World Congress on Computational Intelligence, 2012.

[8] C. Copot, C. Lazar and A. Burlacu. *Predictive control of nonlinear visual servoing systems using image moments*, IET Control Theory Applications, vol.6, iss.10, pp.1486-1496, 2012.

[9] G. Allibert, E. Courtial and Y. Toure. *Visual predictive control for manipulators with catadioptric camera*, Proc. of IEEE International Conference on Robotics and Automation, pp.510-515, 2008.

[10] S. Hussmann and T. Edeler. *Pseudo-Four-Phase-Shift Algorithm for Performance Enhancement of 3D-TOD Vision Systems*, IEEE Transactions on Instrumentation and Measurement, vol.59, no.5, pp.1175-1181, 2010.

[11] S. Fuchs and G. Hirzinger. *Extrinsic and Depth Calibration of ToF-cameras*, IEEE Conference on Computer Vision and Pattern Recognition, CVPR , art. no. 4587828, 2008.

[12] E. Marchand and F. Chaumette. *A new formulation for non-linear camera calibration using virtual visual servoing*, IRISA Technical Report, 2001.

[13] J.C. Russ. *The Image Processing Handbook*, CRC Press Taylor & Francis Group, ISBN:0-8493-7254-2, 2007.

[14] E.F. Camacho and C. Bordons. *Model Predictive Control*, Springer-Verlag Berlin Heidelberg New York, ISBN: 3-540-76241-8, 2000.

SISTEMA DE ALERTA Y REGISTRO DE INFRACCIONES DE TRÁFICO

Nourdine Aliane, Javier Fernández y Sergio Bemposta
Universidad Europea de Madrid, Villaviciosa de Odón 28670, Madrid, España
Dpto: Informática, Automática y Comunicaciones.
{nourdine.aliane;javier.fernandez;sergio.bemposta}@uem.es

Resumen

En este artículo se presenta un sistema de ayuda a la conducción capaz de alertar a los conductores de situaciones de riesgo asociadas a la señalización vertical y registrar determinadas infracciones de tráfico en una caja negra, que posteriormente, se pretende usar para proporcionar a los conductores un feedback sobre su conducción. El prototipo desarrollado se basa en un sistema de visión artificial capaz de identificar señales de tráfico y en una instrumentación capaz de capturar el estado del vehículo así como identificar a los conductores.

Palabras Clave: Sistemas Empotrados, Visión por Computador, Sistemas Inteligentes de Transporte, Seguridad Vial.

1 INTRODUCCION

Uno de los mayores desafíos a los que se enfrentan los países desarrollados es la necesidad de reducir el número de accidentes de carretera y de disminuir su alto coste en vidas humanas. Un informe del 2009 de la Comisión Europea sobre la seguridad vial [1] alerta de la muerte anualmente de unas 35.000 personas involucradas en más de 1.5 millones de accidentes de tráfico, y lo que económicamente se traduce en un coste de más de 160.000 M€. A nivel nacional, un informe de la Aseguradora Mapfre [2] alerta que el número de accidentes de tráfico con víctimas mortales registrados en España en el 2010 asciende a 1.548 causando la muerte de 1730 personas y provocando 7954 heridos graves. Estas cifras de fallecimientos son superiores si se comparan con otros medios de trasporte.

Varios estudios como [3, 4] afirman que en el 90% de los accidentes de tráfico está presente el factor humano. Las principales causas de estos accidentes son la distracción al volante [4, 5], la somnolencia [6, 7], la fatiga [8], o la desatención [9, 10]. Estos factores conllevan una pérdida de concentración al volante y reducen de forma considerable la percepción y la interpretación de las señales de tráfico. En el trabajo [11] los autores afirman que un porcentaje alto de accidentes de tráfico está directamente relacionado con el no respeto de las señales de tráfico, siendo el exceso de velocidad el factor más común en los accidentes de vehículos [12]. Es importante señalar que la mayor parte de los conductores no cometen infracciones de tráfico de forma deliberada, y gran parte de estas infracciones se producen como consecuencia de la pérdida de concentración en la conducción. Las señales de tráfico transmiten información de obligatoriedad, de peligro y otras indicaciones que los conductores no deben de obviar. Por tanto, el uso de un sistema de ayuda a la conducción que avise a los conductores de los peligros potenciales en carretera, y hacerles ver sus infracciones de tráfico debe considerarse como un medio que puede concienciar a los conductores y hacerles reflexionar sobre sus errores.

Para mejorar la seguridad vial, en los últimos años se ha realizado una importante labor de investigación y desarrollo enfocada al diseño de sistemas de ayuda a la conducción. Prueba de ello es que la Comisión Europea pone de manifiesto su apuesta por la integración de la tecnología en los automóviles con varias iniciativas como [3, 13, 14] cuyas directrices están orientadas a diseñar vehículos inteligentes que contribuyan a mejorar la seguridad vial.

Los sistemas de ayuda a la conducción, que sin llegar a tomar control del vehículo, tienen por objetivo alertar a los conductores de los potenciales peligros en carretera mediante la emisión de advertencias sonoras y/o visuales. En la literatura existen varios trabajos que describen diferentes sistemas de ayuda a conducción. Por ejemplo, en [6] se propone un sistema basado en visión artificial capaz de detectar la somnolencia y la distracción del conductor. En [8] se propone el uso de un sistema de visión que analiza la localización del rostro y de los ojos para prevenir la fatiga en los conductores. En [10] se propone un método basado en la lógica borrosa para detectar la inatención del conductor mediante la toma de imágenes de la cara del conductor y de algunos parámetros como la posición de la cabeza, la frecuencia del parpadeo, etc. En [15] se ha investigado el problema de la distracción de los

conductores como consecuencia del manejo de los dispositivos como el teléfono móvil, los navegadores, la radio, etc, mediante técnicas de minería de datos. En [16] se propone un sistema basado en visión artificial con múltiples funcionalidades como la detección y reconocimiento de señales de tráfico, alerta en cambios de carril no intencionadas y la detección de peatones.

Otra línea de investigación está enfocada a la caracterización del comportamiento de los conductores a partir de datos obtenidos usando simuladores. Es el caso del trabajo [17] donde se propone modelar la conducción centrada en cómo evitar accidentes haciendo uso de modelos estocásticos. El trabajo [15] propone un método para detectar la distracción asociada al manejo de dispositivos electrónicos mediante un sistema de visión que hace un seguimiento del movimiento del rostro y de los ojos. De la misma manera, [18] propone un método para detectar la somnolencia en los conductores en base a los movimientos del volante. Existen también trabajos que se basan en datos reales de conducción haciendo uso de dispositivos EDR "Event Data Recorder", conocidos comúnmente como cajas negras [19, 20]. La versión básica de los EDR se ha utilizado para registrar principalmente la información sobre las circunstancias que rodea a los accidentes [21, 22]. En los último años estos dispositivos se han utilizado con otros fines, como por ejemplo, para evaluar el riesgo de la conducción registrando los distintos tipos de maniobras que los conductores realizan sobre el vehículo [23], como un instrumento para disuadir a los conductores de sus comportamientos temerosos [24], usados para medir la eficiencia de la conducción en términos de consumo de combustible [25], y recientemente se han usado para registrar datos y proporcionar a los conductores un feedback sobre su conducción [23, 26].

El objetivo de este trabajo es presentar un sistema de ayuda a la conducción basado en visión artificial capaz de reconocer las señales de tráfico e identificar determinadas infracciones de tráfico como el "exceso de velocidad" o el no respeto de las señales de "stop" y de "giro prohibido". El sistema tiene por objetivo alertar en tiempo real a los conductores de determinadas situaciones de riesgo, y registrar determinadas infracciones de tráfico en una caja negra, que posteriormente se pretende usar para proporcionar a los conductores un feedback sobre su conducción. El sistema, por tanto, puede ayudar a los conductores a darse cuenta de sus errores y permitirles hacer un auto-diagnóstico de su conducción y, por ende, ayudarles a practicar una conducción responsable.

El resto del artículo está organizado de la siguiente manera: la sección 2 describe la plataforma experimental usada para el desarrollo de este sistema así como su arquitectura hardware. La sección 3 describe los módulos funcionales más importantes del sistema experimental. La sección 4 presenta el sistema de alerta y registro de infracciones de tráfico. Finalmente, la sección 5 concluye el presente trabajo.

2 PLATAFORMA EXPERIMENTAL

El desarrollo de un sistema de ayuda a la conducción capaz de alertar a los conductores de los posibles riesgos en la carretera y registrar las infracciones de tráfico requiere un vehículo con una instrumentación capaz de captar su entorno así como su estado interno. La plataforma experimental utilizada en este trabajo se basa en un vehículo convencional Nissan Note como se muestra en la Figura 1.



Figura 1: Fotografías de la plataforma experimental.

El hardware embebido del sistema consta de un PC, conectado a dos cámaras, dedicado exclusivamente al procesamiento de imágenes, y de una placa Mini-ITX como ordenador central. Ambos ordenadores están montados en un rack ubicado en el maletero del vehículo. La placa Mini-ITX, a través de sus diferentes chipsets, permite integrar de forma sencilla todos los dispositivos que dan soporte a las diferentes funcionalidades del sistema. En efecto, la placa dispone de un adaptador para tarjetas de memoria de tipo "compaq-flash" para dar soporte a una base de datos local (o caja negra), la placa está también conectada a un lector de tarjetas inteligentes para la identificación de los conductores, está conectada a una pantalla táctil para ejecutar la interfaz HMI del sistema, e integra una instrumentación variada como un GPS, la interfaz ODB-II y otros sensores que se van a detallar más adelante. Todos los dispositivos electrónicos que componen la plataforma se alimentan mediante un inversor de continua/alterna, transformando los 12 Voltios de la batería del vehículo en una corriente alterna a 220 Voltios con una potencia de 800 W. La arquitectura hardware, resumida en la figura 2, permite el desarrollo y el despliegue de otras funcionalidades relacionadas con la seguridad vial de forma sencilla.

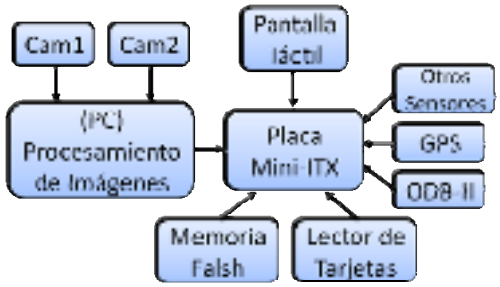


Figura 2: Arquitectura hardware del sistema.

3 ARQITECTURA FUNCIONAL

Desde el punto de vista funcional, el sistema de alerta y registro de infracciones de tráfico está compuesto de cinco módulos interconectados: un sistema de visión artificial, un sistema de adquisición de datos basado en varios sensores, un sistema para la identificación de conductores, una caja negra para el registro de eventos asociados a la conducción así como las infracciones de tráfico, y un módulo para ejecutar una interfaz hombre máquina. La arquitectura software del sistema, mostrando las relaciones entre los diferentes módulos, queda resumida en la Figura 3.

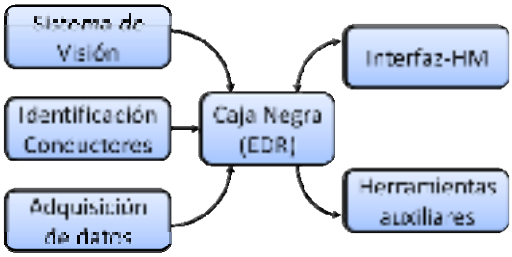


Figura 3: Arquitectura software del sistema.

3.1 DETECCION Y RECONOCIMIENTO DE LA SEÑALIZACIÓN VERTICAL

El sistema de visión está compuesto por dos cámaras digitales de color de resolución (1392×1040) montadas en el techo del vehículo y se utilizan para escanear la carretera para detectar las señales de tráfico. La distancia focal de estas cámaras es de 12-mm, lo que les proporciona un campo de visión que permite detectar señales de tráfico a una distancia de hasta 80 metros.

El proceso de detección e identificación de las señales de tráfico se efectúa en varios pasos [27, 28]. En primer lugar, a las imágenes capturadas se le aplica una segmentación en el espacio de color HLS empleando umbrales dinámicos. A continuación, las regiones susceptibles de ser señales se someten a un análisis geométrico de forma basado en el ajuste de tramos rectos y circulares a las posibles señales. Los objetos que cumplen con las restricciones de color y de forma son contrastados con la señalización vertical legal mediante correlación normalizada. Los objetos no reconocidos en este paso son descartados.
En las pruebas realizadas hasta ahora, se han notado algunas confusiones con la primera cifra en las señales de "limitación de velocidad", dado que los números 5, 6, 8 y 9 son similares en el diseño vigente, y que por otra parte, las señales no están pensadas para su reconocimiento mediante visión por computador. Sobre estas señales de limitación de velocidad, se realiza un análisis posterior empleando una segmentación mediante un umbral adaptable recursivamente hasta conseguir una cifra nítida. De esta manera, se reduce la confusión entre límites de velocidad a menos del 1%. El resultado final es una señal correctamente interpretada que se emplea en la política de aviso al conductor o de registro de la infracción en su caso. El sistema de visión artificial es capaz de procesar hasta 4 imágenes por segundo. Así, a una velocidad de 100 km/h, el sistema dispone de al menos de 2 oportunidades de detectar las señales a las distancias de 50 y 30 metros respectivamente antes de rebasarlas, y de más oportunidades a velocidades menores.

El sistema ha sido probado en condiciones reales en más de 2000-km circulando a velocidades normales por entornos urbanos e interurbanos tanto en autovías como en carreteras secundarias, en condiciones atmosféricas variables y a diferentes horas del día. Obviamente la mayor dificultad para el sistema es la adaptación a las distintas condiciones de iluminación que ocurren en estos entornos. Se ha dispuesto de un sistema de ajuste dinámico de los parámetros de captura para responder en la medida de lo posible a estos cambios. En condiciones de iluminación favorables (sol alto o cielo completamente cubierto, que es el caso que ocasiona una iluminación difusa), el reconocimiento de señales se produce en el 90% de los intentos, estando la mayor parte de los errores en los intentos "lejanos" (a 50m). El acierto crece hasta el 98% en los intentos "cercanos" (30m o menos). Contando con al menos 2 intentos por señal, la tasa de acierto global se sitúa en el 96%. No se producen falsos positivos, ya que el sistema rechaza los casos dudosos. La tasa de reconocimiento cae al 85% durante situaciones de lluvia, dado que elimina prácticamente la posibilidad de la lectura "lejana" y dificulta la lectura "cercana". Finalmente, las condiciones que hacen caer drásticamente las prestaciones del sistema son en trazados urbanos, los giros del vehículo hace que en ocasiones la señalización quede fuera del área de captura de la cámara, o la falta de luz en las salidas de los túneles. Las iluminaciones con fuerte contraluz que provocan la saturación de la cámara, y por tanto, quedándose las señales excesivamente oscuras. Esto ocurre también en las circunstancias de un sol con un ángulo

bajo frente a la cámara, del reflejo del sol sobre un suelo mojado, o cuando las señales están colocadas en un cambio de rasante lo suficientemente elevado, provocando todos estos casos una saturación de la cámara.

3.2 SISTEMA DE ADQUISICIÓN DE DATOS

El sistema de adquisición de datos está formado por una serie de sensores encargados de recoger los datos asociados al estado del vehículo. Por ejemplo, la velocidad, las revoluciones, o la aceleración se leen directamente de la unidad electrónica del vehículo "ECU" a través del puerto ODB-II. Para tener acceso a estos datos desde un programa de ordenador, se ha utilizado una interfaz hardware "ELM-327", que es un dispositivo que convierte el puerto ODB-II en una interfaz RS-232 convencional. Este circuito permite ver la ECU del vehículo como un módem, y desde el punto de vista de programación, la comunicación entre la placa Mini-ITX y la ECU del vehículo se hace mediante envío de comandos AT a través de la línea serie.

Por otra parte, para medir el giro del volante se ha utilizado un codificador óptico absoluto de 7-bits montado sobre el eje de dirección. El giro del volante se mide con respecto de una posición neutral que se define mediante en la fase de calibración apretando un botón. La carrera completa del volante es de 1.5 vueltas en cada sentido (-540º a 540º), por lo que el codificador ofrece una resolución de 8.5º por bit del codificador (o lo que es equivale a un giro en las ruedas de aproximadamente 1.2º).

La placa Mini-ITX está también conectada a un receptor GPS con una frecuencia de muestreo de 1-Hz, que es suficiente para la navegación de un vehículo. Finalmente, la placa Mini-ITX integra varios sensores de contacto para registrar las interacción del conductor con los pedales, los cinturones de seguridad, los indicadores, etc.

3.3 IDENTIFICACIÓN DE CONDUCTORES

El sistema experimental identifica a los conductores y diferentes usuarios mediante el uso de tarjetas inteligentes, cuyo lector está directamente conectado a una ranura del Mini-ITX. Estas tarjetas llevan grabados datos de interés como la identificación de los conductores, direcciones, información médica como el grupo sanguíneo, huellas dactilares, etc. Por otra parte, el sistema limita el acceso a la información guardada en la caja negra, y el acceso a los datos se hace de forma selectiva acorde a los diferentes perfiles de los usuarios. En efecto, el sistema distingue tres tipos de usuarios: los conductores como usuarios finales, los administradores del sistema, que podrían ser los responsables flotas, y los agentes de tráfico (reservado para futuras aplicaciones). Estos perfiles otorgan permisos diferentes y permiten accesos selectivos a los registros de la caja negra. Por ejemplo, los conductores solo pueden acceder a sus registros personales y no pueden modificar los datos grabados. El agente de tráfico tiene un acceso restringido y solo tiene acceso a la información del tacógrafo digital que se basa en el histórico de los tiempos de descanso y de conducción y las infracciones. Finalmente, si no se usa ninguna tarjeta inteligente, el sistema registra datos a cuenta de un usuario de carácter anónimo.

3.4 CAJA NEGRA

La caja negra es una base de datos local al vehículo que está físicamente soportada por una memoria de tipo "compact-flash" para el almacenamiento de eventos sucedidos durante la conducción y de un disco externo para el almacenamiento de imágenes de las señales de tráfico. La información guardada en la caja negra procede de varias fuentes y está estructurada en varios campos (velocidad, revoluciones, aceleración, posición GPS, estado de los cinturones de seguridad, acciones sobre los pedales, etc). Estos datos se recogen durante un viaje de manera continuada a una frecuencia de cuatro (4) registros por segundo. La información guardada en la caja negra está vincula al calendario y a los conductores que se identifican con sus tarjetas electrónicas. Por otra parte, en el contexto del registro de infracciones, se guarda la información asociada al tipo de infracción cometida, su posición GPS y una fotografía de la señal de tráfico correspondiente y su entorno.

Desde el punto de vista de la implementación, la adquisición de datos es de naturaleza paralela, cuando la escritura en la caja negra es secuencial. En este aspecto, la solución adoptada consiste en una toma de datos desde los diferentes dispositivos mediante un programa multi-hilo y uno de ellos se encarga de la escritura final en el soporte físico.

En la actualidad, la caja negra está diseñada con el objetivo de proporcionar un feed-back a los conductores, aunque se le puede dar otros usos como en el análisis del comportamiento de los conductores, optimización de trayectos e itinerarios en la gestión de las flotas, etc.

3.5 INTERFAZ HOMBRE-MAQUINA

La plataforma experimental está equipada con una pantalla táctil situada en el salpicadero, y ejecuta la aplicación de visualización (HMI) como se muestra en la Figura 4.



Figura 4: La pantalla táctil en el salpicadero.

Su interfaz principal está organizado en varias pestañas que dan acceso a las distintas funcionalidades del sistema así como a las tareas de administración. En la pantalla principal se muestra las 4 últimas señales de tráfico reconocidas. En el caso de una infracción de tráfico, las alertas se emiten en forma de mensajes acústicos a través de los altavoces del vehículo. A través de su sistema de navegación, se puede tener acceso a los datos de la caja negra como el registro de las infracciones, el histórico de la velocidad del vehículo, etc. La Figura 5 muestra una captura de la interfaz del programa de gestión del sistema.

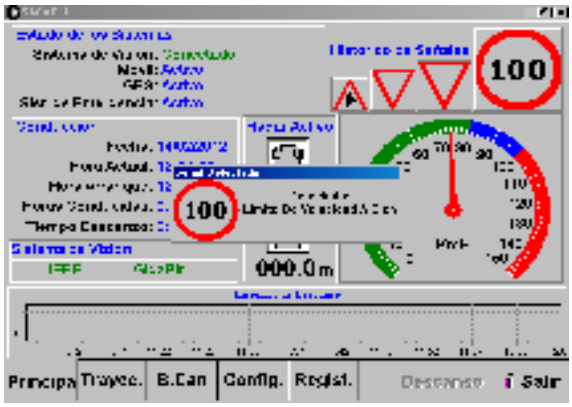


Figura 5: Interfaz del programa principal.

El desarrollo software del sistema se ha realizado principalmente con el entorno de programación C++Builder. En cuanto al procesamiento de las imágenes, se han utilizado las MIL "Matrox Imaging Library", la cual es una librería de funciones optimizadas para el procesamiento de imágenes y desarrollo de aplicaciones relacionados con visión artificial.

4 REGISTRO DE INFRACCIONES

El sistema de alerta y registro de infracciones tiene por objetivo avisar a los conductores en determinadas situaciones de riesgo que puedan surgir en carretera. El sistema se encarga de la detección y reconocimiento de la señalización vertical presente en la vía y generar avisos para el conductor. Los avisos son configurables pudiendo emitir las advertencias mediante mensajes sonoros a través de los altavoces del vehículo y/o visuales a través de una pantalla colocada en el salpicadero.

Si bien el modulo de visón es capaz de reconocer hasta 64 señales de tráfico, los avisos al conductor están restringidos a las situaciones críticas como las señales de "limitación de velocidad", "adelantamiento prohibido", las aproximaciones a señales de cruce como el "ceda el paso", el "stop", "prohibido el giro" y las señales relacionadas con las obras en carreteras. Estos avisos repercuten directamente en mejorar la atención del conductor sobre la vía, la velocidad de su vehículo, y en prestar más atención en los puntos peligrosos como los cruces o las incorporaciones.
Es importante señalar que un sistema de ayuda a la conducción tiene una mayor eficacia en la identificación de señales de tráfico en comparación a un conductor en estado de cansancio, de somnolencia o de fatiga. Así pues, el sistema permite proporcionar a los conductores información que puede ser útil o incluso vital para el buen manejo del vehículo y prevención de accidentes. Por ejemplo, un sistema que puede avisar a los conductores de reducir la velocidad en lugares donde un exceso de velocidad puede terminar en un accidente es, naturalmente, un medio necesario para la seguridad vial.

Por otro lado, algunas de las advertencias comentadas anteriormente pueden convertirse en infracciones de tráfico. Por ahora, el sistema solo contempla tres tipos de infracciones: "el exceso de velocidad", no respetar las señales de "stop", y "giros prohibidos". Cuando el sistema emite una alerta, el conductor dispone de un tiempo de espera de 2-4 segundos antes de registrarse como infracción. Este tiempo es suficiente para que el conductor reaccione a la situación de tráfico indicada por la señal. En cuanto al funcionamiento, cuando se detecta las señales correspondientes a las infracciones, el sistema las contrasta con el estado del vehículo para determinar si efectivamente se ha cometido la infracción o no, y en caso positivo, la infracción se registra en la caja negra. Cada una de las infracciones soportadas por el sistema tiene una política de gestión específica. En el caso de un exceso de velocidad, el sistema emite el aviso correspondiente, y pasado el tiempo de espera se hace una comparación con la velocidad real del vehículo para determinar si hay una infracción de tráfico o no. De la misma forma, cuando se detecta una señal de "Stop", el sistema comprueba si la velocidad del vehículo ha decaído hasta cero. En cuanto a los giros prohibidos, su procesamiento se realiza mediante la medición del giro del volante del vehículo. Las infracciones registradas en la caja negra constan de varios campo: la posición GPS del lugar de infracción, una fotografía de la señal de tráfico y de

su entorno, la velocidad del vehículo, la fecha y el instante de cuando se cometió la infracción y la identificación del conductor. El registro de las infracciones tiene por objetivo principal proporcionar un feedback a los conductores, dándoles la posibilidad de consultar sus infracciones, y permitiéndoles hacer un auto-diagnóstico de sus faltas. La Figura 6 muestra una captura de la herramienta usada para recuperar el registro de las infracciones de tráfico cometidas por un conductor.



Figura 6: Registro de infracciones.

Así mismo, se ha desarrollado una aplicación auxiliar que explota los datos almacenados en la caja negra para hacer una visualización gráfica de las infracciones cometidas sobre Google Earth. Esta aplicación permite visualizar gráficamente el recorrido del vehículo, su velocidad a lo largo del trayecto, y la distribución geográfica de las infracciones cometidas en el mapa. La Figura 7 muestra un ejemplo de visualización sobre Google Earth, permitiendo al usuario visualizar sobre el mapa las fotografías de los lugares donde se han cometido las infracciones. Finalmente, en lo que se refiere a la implementación, los datos almacenadas en la caja se guardan en una archivo de marcas de tipo "KML" que se utiliza para modelar las características geográficas como puntos, líneas, imágenes, polígonos y modelos que se mostrarán en Google Earth.

5 CONCLUSIONES

En este artículo se ha presentado un sistema de ayuda a la conducción capaz de alertar a los conductores de los posibles peligros en carretera así como registrar determinadas infracciones de tráfico en una caja negra. El objetivo del sistema es recordar a los conductores acerca de su comportamiento negligente al volante y hacerles ver sus infracciones de tráfico. El sistema es, por tanto, un medio que podría ayudar a concienciar a los conductores y hacerles reflexionar sobre sus errores. En cuanto a trabajos futuros, se está considerando complementar el sistema presentado en este artículo con un estudio de aceptabilidad de su uso, y otro trabajo que consiste en usar la plataforma para la toma de datos reales de conducción e investigar aspectos relacionados con el cumplimiento y respeto de la señalización vertical presente en carretera.

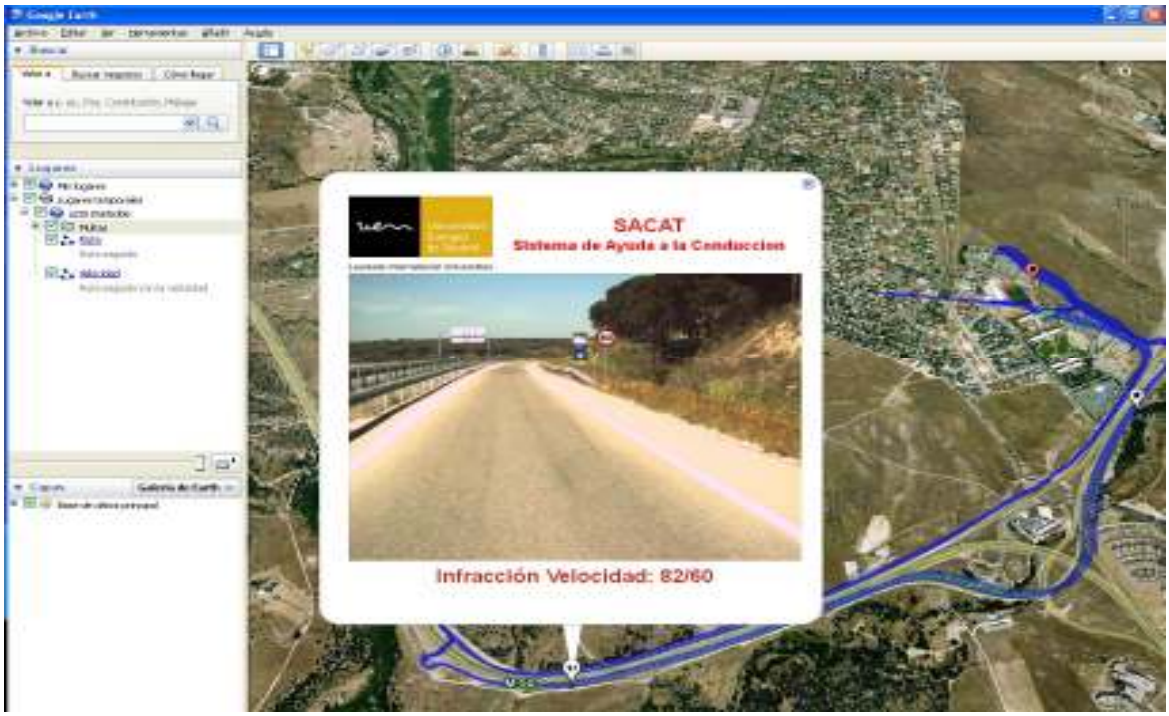


Figura 7: vista de Google Earth desplegando la evidencia de una infracción.

Agradecimientos

Este trabajo ha sido financiado en parte por la Comunidad de Madrid bajo el proyecto (S2009/DPI-1509-SEGVAUTO) y por el ministerio de ciencias e innovación bajo el proyecto (TRA2010-20225-C03-02 SAMPLER).

Referencias

[1] SEC: Security European Commission (2011), White paper 'On support for an EU-wide e-Call service". Retrieved January, 30, 2012, from: http://ec.europa.eu/governance/impact/ia_carried_out/docs/ia_2011/sec_2011_1019_en.pdf

[2] SIMS: Spanish Insurance Market Service (2010). Retrieved January, 30, 2012, from: http://www.mapfre.com/ccm/content/documentos/fundacion/cs-seguro/informes/The-Spanish-Insurance-Market-in-2010.pdf

[3] eSafety: Retrieved January, 30, 2012, from: http://ec.europa.eu/information_society/activities/esafety/index_en.htm

[4] Regan, M. A., Hallett, C., Gordon, C. P., 2011. Driver distraction and driver inattention: definition, relationship and taxonomy. Accident Analysis and Prevention. 43(5), 1771-1781.

[5] Horrey, W.J., Lesch, M. F. 2009. Driver-initiated distractions: Examining strategic adaptation for in-vehicle task initiation. Accident Analysis and Prevention. 41(1), 115-122.

[6] Flores, M. J., Armingol, J.M, Escalera, M.A., 2011. Sistema avanzado de asistencia a la conducción para la detección de la somnolencia. RIAI, 8(3), 216-228.

[7] Flores, M. J., Escalera, A., Armingol, J. M., 2010. Real-time warning system for driver drowsiness detection using visual information. Journal of Intelligent and Robotic Systems. 59(2), 103-125.

[8] Tran, C., Trivedi, M.M. 2011. Vision for driver assistance: Looking at people in a vehicle, guide to visual analysis of humans. Springer, 597-614.

[9] Nuevo, J., Bergasa, L. M., Llorca, D. L., Ocaña, M., 2011. Face tracking with automatic model construction. Image and Vision Computing. 29, 209-218.

[10] Bergasa, L. M., Nuevo, J., Sotelo, M. A., Barea, R., Lopez, M. E., 2006. Real-time system for monitoring driver vigilance. IEEE trans. Intelligent Transportation Systems, 7(1), pp.63-77.

[11] Lawton, R., Parker, D., Manstead, A. S. R, Stradling, S. G., 1997. The Role of affect in predicting social behaviors: The case of road traffic violations. Journal of Applied Social Psychology. 27(14), 1258–1276.

[12] Afukaar, F. K., 2003. Speed control in developing countries: issues, challenges and opportunities in reducing road traffic injuries, Injury Control and Safety Promotion. 10(1 &2), 77-81.

[13] Intelligent-Car: Retrieved January, 30, 2012, from: http://ec.europa.eu/information_society/activities/intelligentcar/index_en.htm

[14] Ertrac Research Framework, European Road Transport Research Advisory Council (2008). Retrieved January, 30, 2012, from: http://www.ertrac.org/en/content/ertrac-publications_10/

[15] Liang, Y., Reyes, M. L., Lee, J. D., 2007. Real-time detection of driver cognitive distraction using support vector machines. IEEE Trans. Intelligent Transportation Systems., 8(2), 340–350.

[16] Armingol, J. M., Escalera, A., Hilario, C., Collado, J.M., Carrasco, J.P., Flores, M.J., Pastor, J.M., Rodríguez, F.J., 2007. IVVI: Intelligent vehicle based on visual information. Robotics and Autonomous Systems, 55, 904-916.

[17] Sekizawa, S. Inagaki, S., Suzuki, T., Hayakawa, S., Tsuchida, N., Tsuda, T., Fujinami, H., 2007. Modeling and recognition of driving behavior based on stochastic switched ARX model. IEEE Trans. Intelligent Transportation Systems. 8(4), 593–606.

[18] Giusti, A., Zocchi, C., Rovetta, A., 2009. A non invasive system for evaluating driver vigilance level examining both physiological and mechanical data. IEEE trans. Intelligent Transportation Systems. 10(1), 127–134.

[19] Perez, A., Garcia, M.I., Nieto, M., Pedraza, J.L., Rodriguez, S., Zamorano, J., 2010. Argos: An advanced in-vehicle data recorder on a massively sensorized vehicle for car driver behavior experimentation, IEEE trans. Intelligent Transportation Systems. 11(5), 463 – 473.

[20] Ma, X. Andreasson, I., 2007. Behavior measurement, analysis, and regime classification in car following. IEEE Trans. Intelligent Transportation Systems. 8(1), 144-156.

[21] Lee, W., Han, I., 2004. Development and test of a motor vehicle event data recorder. Inst, Mech. Eng. Journal of Automobile Engineering, 218(D), 977-985.

[22] Niehoff, P. Gabler, H. C., Brophy, J., Chidester, C., Hinch, J., Ragland, C., 2005. Evaluation of event data recorders in full systems crash tests. 19th Int. Technical Conf. on the Enhanced Safety of Vehicles.

[23] Toledo, T., Musicant, O., Lotanc, T., 2008. In-vehicle data recorders for monitoring and feedback on drivers behaviour, Transportation Research Part C: Emerging Technologies. 16(3), 320-331.

[24] Wouters, P. I. J., Bos, J.M.J., 2000. Traffic accident reduction by monitoring driver behaviour with in-car data recorders. Accident Analysis and Prevention. 32(5), 643-650.

[25] Mundke, R., Malewar, S., Rekhi, K., 2006. Use of data recorder for driver rating. SAE International, 06AE-273.

[26] Takeda, K., Miyajima, C., Suzuki, T., Kurumida, K., Kuroyanagi, Y., Ishikawa, H., 2011. Improving driving behavior by allowing drivers to browse their own recorded driving data. 14th Int. IEEE-Conference on Intelligent Transportation Systems, Washington, 44-49.

[27] Bahlmann, C., Zhu, Y., Armes, V. , Pellkofer, M. Koehler, T., 2005. A System for Traffic Sign Detection, Tracking, and recognition using color, shape, and motion information. IEEE Intelligent Vehicles Symposium. 6(8), 255-260.

[28] Escalera, A, Armingol, J. M. Mata, M., 2003. Traffic sign recognition and analysis for intelligent vehicles. Image and Vision Computing. 12(3), 247-258

Análisis del descriptor VFH para el reconocimiento 3D de múltiples vistas de objetos domésticos

Francisco Martínez, Carlos M. Mateo, Pablo Gil, Fernando Torres
Grupo de Automática, Robótica y Visión Artificial
Dpto. Física, Ingeniería de Sistemas y Teoría de la señal
Universidad de Alicante
franciscomartinez@outlook.com, cmmateo86@gmail.com,{pablo.gil, fernando.torres}@ua.es

Resumen

Los nuevos sensores de bajo coste como Kinect y la implementación de nuevos detectores y descriptores 3D están permitiendo importantes avances en aplicaciones de percepción robótica, tales como reconocimiento tridimensional de objetos, mapeado de entornos para la navegación de robots o la construcción de interfaces para la interacción hombre-máquina. En este artículo, se presenta un método de reconocimiento de objetos domésticos 3D de propiedades distintas: en forma, dimensiones y material de fabricación. En este método se analiza el uso de un descriptor de tipo global de reciente aparición denominado Histograma de Características de punto de vista (VFH), el cual además de proporcionar información de la forma de un objeto es capaz de trabajar con información de pose o punto de vista en el que ese objeto fue registrado. Ambas características son muy interesantes para aplicaciones de manipulado robóticos en el que se requiere reconfigurar posición y orientación de la garra de sujeción.

Palabras Clave: RGB-D, reconocimiento 3D, descriptor VFH.

1 INTRODUCCIÓN

De todas las tareas de visión artificial, analizar una escena reconociendo todos los objetos que aparecen en ella sigue siendo una de las más complejas. En la actualidad, la tecnología ha avanzado hasta poder reconstruir de manera precisa escenas 3D a partir de imágenes 2D [1]. No obstante, todavía no es fácil aprender a reconocer todos los objetos presentes en una escena, independientemente de cual sea la posición y orientación espacial que éstos ocupen en una secuencia de imágenes, o cual sea el punto de vista con el que éstas se han tomado. En este sentido, por un lado, los nuevos sensores con capacidad de medir profundidad, como ToF y/o RGBD [2] han contribuido favorablemente para identificar planos y volumenes, trabajando a partir de nubes de puntos [3]; por otro lado, la implementación de nuevos detectores de características y en especial de nuevos descriptores han contribuido a mejorar los procesos de reconocimiento de objetos 3D. Así, destacan en la literatura los descriptores locales como NARF [4], SURF (Speeded-Up Robust Features) [5] y ORB (Oriented FAST and Rotated BRIEF) [6] que combina el detector de puntos de interes de gradiente máximo de FAST [7] con el descriptor de BRIEF [8]. Todos ellos son descriptores que son invariantes a escala y orientación, no tanto al punto de vista, pero que son fuertemente dependientes de información de textura en la imagen. En el caso de que haya demasiada homogeneidad entre píxeles de vecindad se imposibilita la extracción de características haciendo que el descriptor se vuelva inconsistente para el reconocimiento. Esta falta de zonas representativas en la imagen, caso que surge cuando hay demasiada homogeneidad, requiere el uso de descriptores globales, entre los más novedosos destacan PFH (Point Features Histograms) [9], FPFH (Fast Point Feature Histogram) [10], SHOT [11] o VFH (Viewpoint Feature Histogram) [12].

Dentro del grupo de descriptores globales destaca FPFH, por su robustez, con una complejidad computacional del orden de $O(nk^2)$, donde n, es el número de puntos de la superficie y k la vecindad de cada puntos. VFH por otro lado consigue una complejidad computacional lineal, $O(n)$, en detrimendo de la precisión. Las elecciones principales del uso de VFH para este trabajo son dos. Primero, se hace uso de un sensor de bajo coste con precisión limitada, por lo que no es necesaria una excesiva precisión en el algoritmo de clasificación. Segundo, se requiere un algoritmo lo más rapido posible, ya que se trata de reconocer el objeto en el menor tiempo posible, para su posterior manipulado. Este trabajo consiste en analizar el funcionamiento y precisión del descriptor VFH para el reconocimiento de formas de objetos 3D.

La percepción sensorial de un entorno y en particular, la identificación y reconocimiento de objetos en éste, se hace una tarea imprescindible en sistemas

inteligentes de manipulación robótica. Así, en aplicaciones de manipulación robótica se requiere determinar qué objeto u objetos hay presentes en el entorno, y las características de éstos: forma, volumen, textura, flexiblidad, etc. La forma permite definir número y posición de los puntos iniciales de sujección para el agarre y manipulado. El volumen, también, contribuye a determinar los puntos iniciales de contacto, pero además, determina el tipo de garra, mano multidedo o pinza y la configuración de apertura-cierre de ésta para asegurar el agarre correctamente. Otros, como la textura del objeto permite determinar el tipo de superficie y nivel de rozamiento en el contacto para evitar deslizamiento y permitir la reconfiguración del agarre en caso de que este se produzca. Y la flexibilidad, determina la capacidad de deformación del objeto. En ocasiones, los objetos se deforman variando sus centros de masas y algunas otras propiedades, si durante el proceso de sujección se les aplica demasiada fuerza. Es una característica interesante de un sistema de visión, ser capaz de poder medir la deformación que se produce en el objeto durante el proceso de manipulado, o si un objeto ha quedado deformado después de un proceso de manipulación robótica. Es decir, hay diferencia entre su forma inicial y su forma final, antes y después del proceso de manipulado respectivamente.

La incorporación de un sistema de percepción para identificar y reconocer objetos empieza hacerse imprescindible si se quiere dotar de flexibilidad y reconfigurabilidad a un sistema de manipulado robótico. En la bibliografía existen trabajos en ese sentido, por ejemplo para ser aplicados a un robot doméstico que mediante visión reconoce los objetos [13] comunes que puedan encontrarse en una estancia doméstica [14], y por tanto, tomar decisiones de agarre de acuerdo con el tipo y naturaleza de dichos objetos

Este artículo se ha organizado de la siguiente manera. En la sección 2 se comenta el descriptor VFH. En la sección 3, se describe el método propuesto para el reconocimiento de objetos 3D. En la sección 4 se muestra y discuten las pruebas y resultados experimentales de distintos objetos. Finalmente, en la sección 5 se presentan las conclusiones.

2 DESCRIPTOR VFH

Para la realización de este trabajo se ha implementado el descriptor global Histograma de Características de Punto de Vista (VFH: *Viewpoint Feature Histogram*). El descriptor VFH consiste en un histograma de un conjunto de características de forma y perspectiva (1) obtenidas desde vectores normales a superficies. Hay un histograma por cada *imagen-i* de una secuencia de *n* imágenes que representan un objeto. Así, cada histograma representa información de forma de un objeto específico desde un punto de vista concreto, donde *n* es el número de posibles puntos de vista.

$$VFH = \{h1, h2, \dots, hn\} \quad (1)$$

De esta manera, es posible construir una base de datos como modelo de objetos distintos. Cada modelo de objeto que se pretende reconocer, estará modelado matemáticamente por un VFH. Esto como se ha visto implica tomar imágenes del objeto-modelo de distintos puntos de vista, y almacenar el histograma para cada una de esas imágenes que representa un punto de vista distinto. De esta forma, se captura todas las formas del objeto posibles y se hacen dependientes del punto de vista con el que se observa.

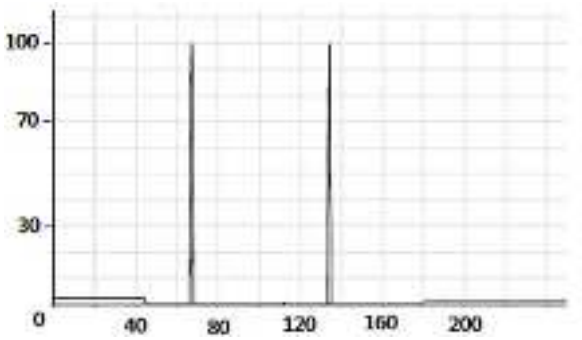


Figura 1: Histograma VFH de una nube de puntos con distribución esférica (véase Figura 2 y 3)

Para calcular cada uno de los histogramas h_i del descriptor VFH (Figura 1), es necesario computar los vectores normales a la superficie de un objeto, donde el objeto se representa en 3D como una nube de puntos (*x, y, z*). Así, dada una superficie geométrica, suele ser trivial inferir la dirección de la normal de un punto cualquiera $p_i=(x_i,y_i,z_i)$ en dicha superficie como el vector perpendicular al plano tangente en dicho punto de la superficie (Figura 2).

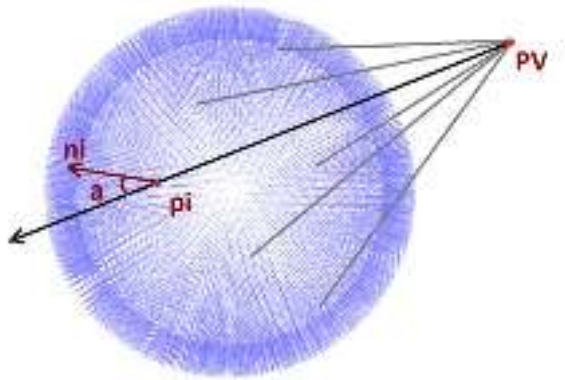


Figura 2: Vectores normales a una superficie esférica.

Así, el vector normal n_i, es perpendicular a cualquier otro vector contenido en el plano tangente. Por lo tanto, si se toman dos o más vectores diferentes y

tangentes a la superficie en dicho punto, su producto vectorial sería siempre perpendicular a todos ellos, y por tanto perpendicular a cualquier combinación lineal de dos de ellos, es decir perpendicular al plano generado por éstos. No obstante, cabe destacar que para superficies cerradas fruto de objetos volumétricos, se dan dos vectores normales, uno exterior y otro interior, y que para el cálculo del histograma sólo interesa los vectores exteriores (Figura 2).

Una vez calculados los vectores normales, como vector exterior, es posible calcular el descriptor VFH. Para ello, se calculan los ángulos de giro (2), inclinación (3) y guiñada (4) que forman los vectores normales de cada punto de la superficie, para después registrar la información en formato de histograma. En general, para una superficie de un objeto, S_o, compuesta por un conjunto de k-puntos tal que, $S_o = \{p_1, \ldots p_k\}$, siendo $p_i = (x_i, yi, zi)$ los puntos de la superficie, C el centroide del objeto, y n_i el vector normal a cada punto p_i, es posible calcular los ángulos de orientación α, ϕ y θ como:

$$\propto = v \cdot n_i \tag{2}$$

$$\phi = u \cdot \frac{p_i - C}{d} \tag{3}$$

$$\theta = \arctan(w \cdot n_i, u \cdot n_i) \tag{4}$$

Donde u,v,w representa un marco de Darboux elegido en p_i y d_i la traslación (Figura 3).

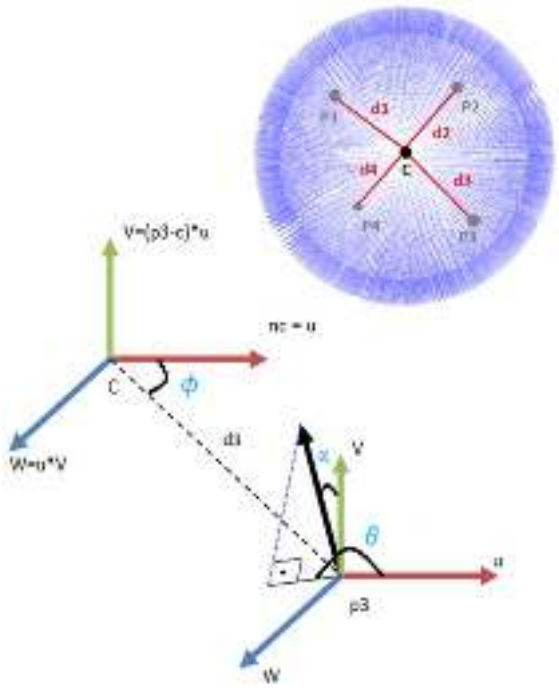


Figura 3: Marco de Darboux

Así, si el conjunto de las normales en los puntos de la superficie describen la forma del objeto, la incorporación al descriptor de estos ángulos permite definir la forma desde un punto de vista concreto. Esta información adicional resulta útil para poder reconocer la '*pose*' en la que se encuentra el objeto. Toda esta información se distribuye en un histograma de la siguiente manera: 60 subdivisiones para cada uno de los tres ángulos α, ϕ, θ y 128 subdivisiones para la componente del punto de vista. Lo que hace un total de 308 componentes por cada histograma. En la figura 1 se ha mostrado una representación de dicho histograma.

3 RECONOCIMIENTO CON VFH

El método implementado en este trabajo, se ha realizado en dos etapas. Una primera etapa de entrenamiento, en la que se crea una base de datos de histogramas VFH, cada uno de los cuales describen un objeto de entre varios objetos posibles a identificar, y una segunda etapa que consiste en un testeo o prueba en la que se obtiene un histograma en tiempo real de lo que percibe el sistema de visión en ese mismo instante, se computan los descriptores VFH y se compara con la información modelo almacenada para distintos objetos en la base de datos, con intención de buscar un grado de semejanza con alguno de ellos.

3.1 BASE DE DATOS DE MODELOS

En primer lugar es necesario crear una base de datos en la que se almacenan los descriptores VFH de los objetos que se buscarán reconocer después como modelos. A mayor número de imágenes, por lo tanto, mayor número de puntos de vista distintos para cada objeto modelo, mejor quedará descrito éste en la base de datos. Al mismo tiempo, en contraposición, una elevada representación del objeto en la base de datos conlleva un mayor tiempo en la búsqueda de correspondencia puesto que hay que realizar mayor número de comparaciones entre el VFH almacenado para el modelo y el histograma computado para el objeto que se percibe. En este trabajo, se ha optado por trabajar con 20 muestras por cada objeto, y en cada muestra el objeto se encuentra girado unos grados sobre sí mismo sin traslación alguna. De este modo, se extrae la forma del objeto en un giro completo de 360º. No obstante, el sistema es flexible y podrían incorporarse, fácilmente, nuevos modelos o nuevas vistas para modelos almacenados. Cabe destacar que además, en todos los casos los objetos poseen un eje de simetría bilateral (caso de las figuritas), y además algunos de los objetos poseen más de un eje de simetría (caso del tetrabrik) o simetría radial (caso del balón).

3.2 FILTRADO Y EXTRACCIÓN DEL DESCRIPTOR

En la figura 4 se describe gráficamente el proceso que se sigue desde la adquisición con Kinect de una

imagen de un objeto hasta el reconocimiento del objeto que se representa en dicha imagen.



Figura 4: Etapas del proceso de reconocimiento

La imagen de entrada (Figura 5), se pueden modelar como una nube de puntos. Esa nube de puntos registra toda la información espacial, (x,y.z), de los objetos presentes en la imagen. En ocasiones, la complejidad de la escena hace que aparezcan varios objetos en la nube, siendo ésta de una gran densidad de puntos. Esto implica un aumento de los costes computacionales asociados al proceso de segmentación para aislar aquellos puntos de la nube que forman parte de los objetos de interés y desechar aquellos que no interesan.

Para realizar el proceso de segmentación se lleva a cabo dos etapas: una de filtrado por distancia (Figura 6) y otra de eliminación de planos (Figura 7). El filtro por distancia, permite mediante un filtrado paso banda eliminar todos aquellos puntos de la nube que espacialmente se sitúan fuera de un rango de trabajo, previamente definido. En este caso se han eliminado todos aquellos puntos cuyo valor de z esté por debajo de 0.5m o por encima de 1.4m. Posteriormente, el filtrado por planos permite aislar los objetos sobre los que apoyan o se sitúan.



Figura 5: Nube de puntos del objeto Caja



Figura 6: Filtrado por distancia



Figura 7: Objeto segmentado

Para la eliminación de estos planos se emplea el algoritmo de RANSAC [15], que permite ajustar puntos de la nube que no pertenecen a un determinado plano. Se emplea como valor de tolerancia, distancia máxima permitida entre un punto y el plano que se está ajustando, 0.03m.

A continuación, la nube de puntos resultante, ya sin los puntos que pertenece a fondo, u objetos de apoyo como suelo o paredes, requiere ser sometida a un proceso de clusterizado. En este proceso se divide y reorganiza la nube de puntos en varios conjuntos de puntos, cada uno de los cuales se agrupa en una subnube de puntos. A su vez, cada una de estas subnubes de puntos forma una misma entidad susceptible de ser objeto, denominado clúster. Este proceso se ha realizado siguiendo los siguientes pasos:

1. Crear una representación *kd-tree* para la nube de puntos de entrada P;
2. Crear una lista de clústeres C, y una cola de puntos que necesitan ser comprobados Q;
3. Entonces, para cada punto $p_i \in P$ hacer:
   - Añadir p_i a la cola actual Q;
   - Para cada punto $p_i \in Q$ realizar:
     - Buscar el conjunto P_k^i de vecinos de p_i en una esfera con radio $r < d$;
     - Para cada vecino $p_k^i \in P_k^i$ comprobar si el punto ya ha sido procesado y si no añadirlo a Q;
   - Cuando la lista de todos los puntos en Q se han procesado, añadir Q a la lista de clusters C, y resetear Q a una lista vacía;
4. El algoritmo finaliza cuando todos los puntos $p_i \in P$ han sido procesados y son parte de una lista de clusters C;

Si la lista de clústeres es mayor que 1 o bien hay varios objetos presentes en la escena o se ha producido un proceso de sobre segmentación, en el que un mismo objeto ha quedado dividido en varias regiones espaciales, formando subnubes de puntos independientes. Esto puede ser debido a ruido durante el proceso de adquisición, oclusión parcial de la superficie de un objeto, o simplemente porque la

superficie del objeto es muy heterogénea, y éste se pueda separar en varias partes. Es común, en los procesos de segmentación, fijar unos valores mínimo y máximo de tolerancia que permitan definir cuando un clúster es o no representativo. Un estudio de la distancia euclídea, entre los centroides de los clústeres detectados, y el número de puntos que componen cada clúster, permite eliminar clústeres no representativos, fusionar clústeres o identificar cuál es el clúster más representativo.

3.3 CORRESPONDENCIA Y NIVEL DE SEMEJANZA

En esta etapa se debe comparar el histograma *VFH*, que describe la instancia del objeto a reconocer con todos los que están almacenados en la base de datos (sección 3.1). Para realizar esta comparación se emplea un sistema de búsqueda basado en una estructura de tipo árbol k-dimensional, que proporciona un coste computacional logarítmico con coste dependiente del número de elementos en la base de datos. Este árbol kd almacena en sus hojas, los *VFHm* de cada modelo. Para realizar el análisis de correspondencias entre dos variables, en este caso el descriptor *VFHm* del modelo, y un descriptor *VFH* de la instancia del objeto testeado, ambos de 308 categorías cada uno, se emplea la distancia Euclídea ponderada Chi-Cuadrado, muy usada en estadística descriptiva para comparar dependencias y semejanzas entre distribuciones de variables. Así, se define la "distancia" (no es una distancia como tal ya que no guarda simetría) Chi-Cuadrado del descriptor *VFH* computado con respecto a *VFHm*, y se denota como $D\chi^2(VFHm,VFH)$, al valor:

$$D\chi^2 = \sum_i \frac{(VFHm_i - VFH_i)^2}{VFHm_i} \quad (5)$$

siendo $VFHm_i$ el descriptor del modelo almacenado en la base de datos, y VFH_i el descriptor del objeto a reconocer. Por lo tanto, la "distancia" de Chi-Cuadrado es una media ponderada de las diferencias cuadráticas de $VFHm_i$ y VFH_i. Contra más pequeño sea el valor de Chi-Cuadrado computado, más se asemejará la instancia del objeto registrado con el modelo que lo representa y que se almacena en la base de datos.

4 EXPERIMENTACIÓN

Para la realización de este trabajo se ha escogido un número limitado de objetos potenciales de ser reconocidos. En particular, se han escogido 10 objetos cuya morfología es bastante distinta. Entre los 10 objetos, destacan objetos fabricados con diferentes materiales: piel, plástico (3 variedades), porcelana (2 variedad), cartón (2 variedades), metal (2 variedades); y con diferente forma tales como: esférica (1 objeto), prisma (3 objetos), antropomórfica (2 objetos), cilíndrica (4 objetos). Estos objetos de forma, textura, volumen y simetría distinta son: una pelota, un bote de gel, un tetrabrik de leche, un muñeco de Mickey, una caja, un bote de refresco, un bol de cocina, una figurita de un coche, una taza y un muñeco de Pooh. En la figura 8 se muestran de izquierda a derecha y de arriba abajo, estos objetos.



Figura 8: Modelo de objetos

En la base de datos se ha contado con 200 muestras (20 instancias por cada uno de los 10 objetos) y se han llevado a cabo un total de 101 pruebas de test con los distintos objetos. Consiguiendo una tasa de reconocimiento del 96,04%. Así en 97 de las pruebas de test se ha producido un reconocimiento positivo y en 4 de las 101, se ha producido un falso reconocimiento o se ha emparejado instancia de objeto y modelo candidato erróneamente. El proceso seguido ha sido descrito en el apartado anterior, en él se calcula y almacena el descriptor VFH para cada modelo de objeto, siguiendo los pasos indicados en el esquema de la figura 4, donde se requería capturar una secuencia de imágenes con Kinect, computar las

nubes de puntos a partir de la información RGBD del sensor y filtrar éstas para quedarse exclusivamente con la nube de puntos que representan el objeto significativo en la imagen. Un ejemplo, de cada uno de los objetos segmentados, una vez aislada su nube de puntos, así como el cálculo de su descriptor VFH se muestra en la figura 9.

Una vez se dispone de la base de datos de objetos, se ejecuta el algoritmo de reconocimiento para testear si es posible identificar el objeto (Figura 10). Para ello, se toma una imagen con Kinect de la escena, se segmenta e identifica la nube de puntos del objeto presente en la escena mediante los procesos de filtrado y clusterizado, comentados en el apartado anterior, y se computa un descriptor VFH. A continuación mediante la distancia Chi-Cuadrado se mide el grado de semejanza entre el VFH computado y el VFH de los modelos almacenados en la base de datos. En la figura 10, se puede ver un ejemplo de test de reconocimiento para cada uno de los 10 objetos. Para ello, se parte de una instancia de muestra adquirida con Kinect de cada objeto y se compara con los modelos registrados en la base de datos de cada clase de objeto. Así, se muestra cómo la pelota ha sido identificada con un distancia mínima de 6.78 con su modelo específico, guardando el segundo mayor parecido con la figurita del coche para la que alcanza un valor de semejanza de 175 muy distinto con el logrado para su modelo específico. De modo similar, el valor de semejanza del bote de gel es de 6.78 con su modelo específico, guardando el segundo mayor parecido con el tetrabrik de leche para el que se obtiene un valor de semejanza de 92.78, bastante lejos, también, del logrado para su modelo específico. El tetrabrik tiene un valor de semejanza de 28.65 con su modelo, frente a 94.65 de similitud con la taza que se mantiene como segundo modelo al que más se le parece. La figurita de Mickey guarda semejanza con un valor de 29 con su modelo, frente a 78 con el modelo de Pooh. La caja de Kinect se asemeja a su modelo en 47 unidades frente a 78 de similitud del modelo de bote de gel, con el que la muestra escogida, en función del punto de vista, también parece guardar cierto parecido. El bote de refresco guarda una semejanza de 2.74 con su modelo frente a las 73.41 que tiene respecto a la taza. La distancia del bol a su modelo es 10.62, mientras que la segunda distancia más cercana sería del bote de refresco con 147.49, también bastante alejado. La figurita del coche guarda una distancia a su modelo específico de 29.74, frente al bote de refresco con 103.42 que se encuentra en segundo lugar. Para la taza se obtiene 35.51 como distancia mínima con respeto a su modelo y el segundo resultado más cercano es la figurita de Mickey con 123 unidades. Finalmente, para la figurita de Pooh se ha obtenido una semejanza de 28, frente a los 58 del segundo candidato que es el modelo de Mickey.

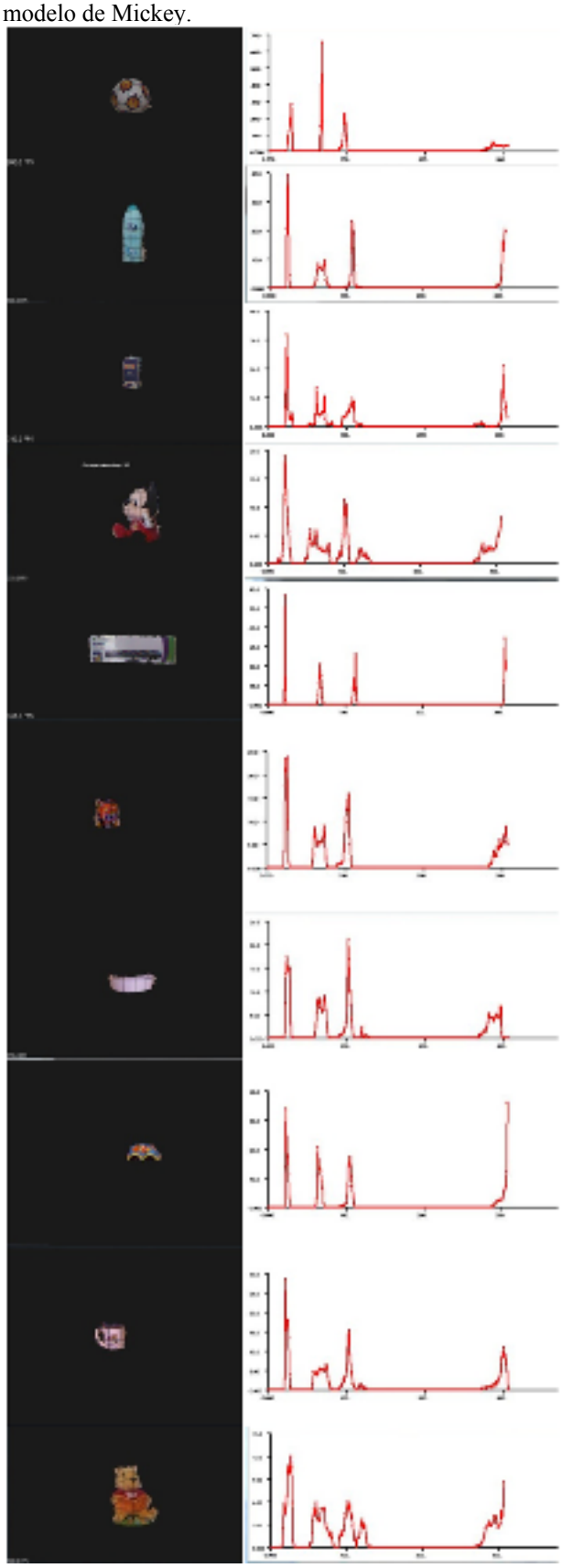


Figura 9: Cálculo descriptor VFH

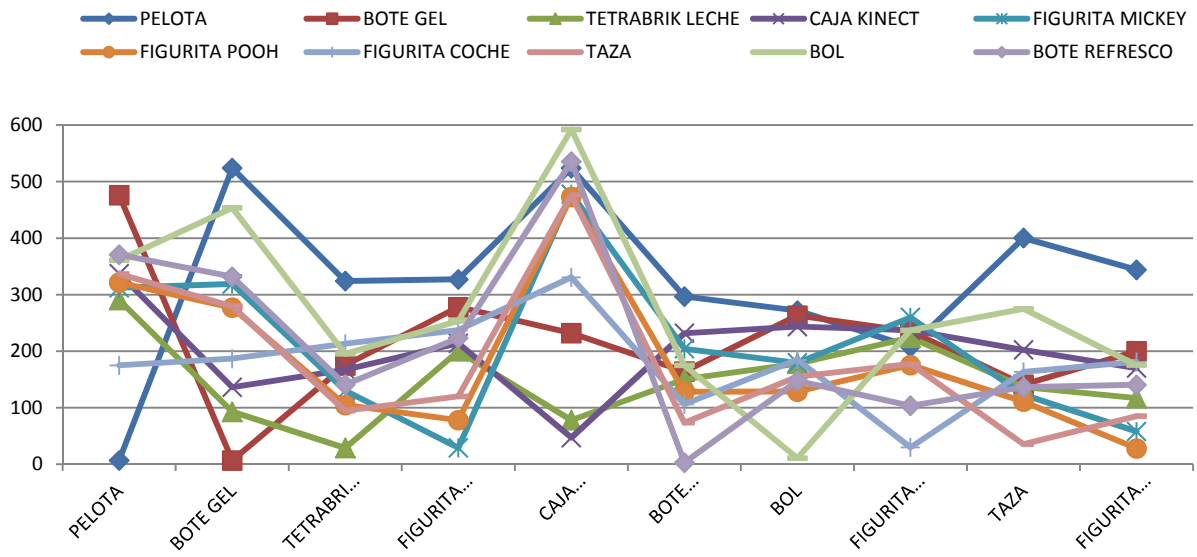


Figura 10: Test de reconocimiento

En la tabla 1, se muestra información resumen de los 200 test realizados, 1 test para cada una de las 20 instancias de cada objeto que se han tomado. En esta tabla se muestran los valores de media de Chi-Cuadrado promediado para los 20 test/objeto, así como su desviación típica.

Tabla 1: Distancia promedio y desviación típica

Objeto	Promedio(μ)	Desviación típica(σ)
Pelota	27,08	16,11
Gel	74,25	38,61
Leche	112,22	35,79
Mickey	65,85	21,54
Caja	156,98	92,69
Bote	23,56	9,72
Bol	21,28	7,05
Coche	89,84	29,12
Taza	82,33	30,51
Pooh	50,56	16,87

Tabla 2: Tiempo medio en la etapa de decisión

Objeto	Promedio tiempo (s.)
Pelota	0,076
Gel	0,060
Leche	0,071
Mickey	0,096
Caja	0,064
Bote	0,070
Bol	0,066
Coche	0,071
Taza	0,076
Pooh	0,086

En la tabla 2, se muestra el promedio de tiempo que se ha tardado, para cada objeto, en el proceso de correspondencia y nivel de semejanza, en el que se compara el modelo test con todos los elementos de la base de tiempo, los valores están en segundos.

4 CONCLUSIONES

En este artículo, se ha presentado un método de reconocimiento de objetos de distintas formas haciendo uso del descriptor VFH. La tasa de acierto se sitúa por encima del 90% aunque se han detectado problemas en el proceso de reconocimiento en determinadas circunstancias, principalmente:

- Similitud geométrica entre modelo e instancia de objeto a reconocer. Por ejemplo, ambos tienen forma prismática, o forma esférica, etc.
- Similitud de forma, en determinadas condiciones de perspectiva. A veces el punto de vista con la que se observa un objeto hace que pueda identificarse la silueta del objeto con otro cuya geometría realmente difiere bastante.

Estas circunstancias, por ejemplo, se observan cuando se pretende comparar un objeto con un modelo candidato con importantes semejanzas geométricas o formas, por ejemplo el tetrabrik de leche y la caja de Kinect si se sitúan ambos en su posición vertical (eje mayor, perpendicular a la superficie de apoyo). En este caso, la forma rectangular de las caras de ambos y sus geometrías prismáticas, determinan que los VFH que describen modelo y objeto sean similares. Este problema podría subsanarse incorporando un segundo descriptor que complementara el uso de VFH, por ejemplo SURF, SIFT, GLOH que hacen uso de textura, detectando cambios de gradiente en superficies heterogéneas.

Por otro lado, en ocasiones, la complejidad del objeto a reconocer, es el caso de la geometría antropomórfica de éste, por ejemplo en los muñecos Mickey y Pooh, se pueden provocar problemas en el reconocimiento dependiendo de la vista con la que se adquieran las imágenes. En estas ocasiones, el algoritmo de identificación de la nube de puntos puede provocar una sobre segmentación de la escena detectando varias nubes de puntos para distintas partes de un mismo objeto, generándose VFH fragmentados que pueden imposibilitar minimizar la distancia entre muestra y modelo.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio Español de Economía, Fondos Europeos FEDER y el Gobierno Regional Valenciano, a través de los proyectos DPI2012-3290, PROMETEO/2013/085 y GV/2012/102.

Referencias

[1] Bylow, E. Sturm, J., Kerl, C., Khal, F., Cremers, D. (2013). "Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions". *Proc. of Conference on Robotics, Science and Systems (RSS)*. Berlin, Germany.

[2] Smisek, J., Jancosek, M., Pajdla, T. (2011). "3D with Kinect". *Proc. of IEEE International Conference on Computer Vision Workshops (ICCVW)*, Barcelona, Spain, pp. 1154-1160.

[3] Rusu, R.B. and Cousins, S., (2011) "3D is here: Point Cloud Library (PCL)". *Proc. of International Conference on Robotics and Automation (ICRA)*, Shanghai, China, IEEE Press, pp. 1-4

[4] Steder, B., Rusu, R.B., Kolige, K., Burgard, W.(2010). "NARF: 3D Range Image Feature for Object Recognition". *Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), workshop on Defining and Solving Realistic Perception Problems inPersonal Robotics*, Taipei, Taiwan.

[5] Bay, H., Tuytelaars, T., Van Gool, L. (2006) "Speedy up robust features". *Proc of European Conference on Computer Vision (ECCV)*. Graz, Austria.

[6] Rublee, E., Rabaud, V., Konolige, K., Bradki, G. (2011). "ORB: an efficiente alternative to SIFT or SURF". *Proc of International Conference on Computer Vision (CVPR)*. Springs, USA.

[7] Rosten, E., Drummond, T. (2006). "Machine learning for highspeed corner detection". *Proc of European Conference on Computer Vision (ECCV)*. Graz, Austria.

[8] Calonder, M., Lepetit, V., Strecha, C. (2010). "Brief: Binary robust independent elementary features". *Proc of European Conference on Computer Vision (ECCV)*. Crete, Greece.

[9] Rusu, R.B., Blodow, N., Marton, Z., Beetz, M. (2008). "Aligning point cloud views using persistent feature histograms". *Proc. Of International Conference on Intelligent Robots and Systems (IROS)*. Niza, Francia.

[10] Rusu, R.B., Blodow, N., Beetz, M. (2009). "Fast Point Feature Histograms (FPFH) for 3D Registration". P*roc. of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan.

[11] Tombari, F., Salti, S., Stefano, L.D. (2010). "Unique signatures of histograms for local surface description". *Proc of European Conference on Computer Vision (ECCV)*, Crete, Greece, pp. 356-369.

[12] Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J. (2010). "Fast 3D recognition and pose using the viewpoint feature histogram". *Proc. of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Taipei, Taiwan.

[13] Lai, K., Bo, L., Ren, X., Fox, D. (2011). "A Large-scale Hierarchical Multi-view RGBD Object Dataset" *Proc. of International Conference on Robotics and Automation (ICRA)*. Shanghai, China, pp. 1817-1824.

[14] Siberman, N., Fergus, R. (). "Indoor Scene Segmentation using Structured Light Sensor". *Proc. of IEEE International Conference on Computer Vision Workshops (ICCVW)*, Barcelona, Spain, pp. 601-608.

[15] Fischler, M.A., Bolles R.C. (1981). "Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Communications of the ACM*, vol.24(6).

CONTROL VISUO-INERCIAL DE 6 GRADOS DE LIBERTAD PARA EL POSICIONAMIENTO AÉREO DE UNA CÁMARA

Jesús G. Villagómez, Manuel Vargas, Francisco R. Rubio
(*villagomez, mvargas, rubio@us.es*)
Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla

Resumen

Se presenta el desarrollo en curso de técnicas de fusión sensorial y control visuo-inercial de los 6-dof de una plataforma multirotor dotada de un sistema de visión, el cual se encuentra ubicado sobre una estructura que permite cambios de orientación en roll *y* tilt*. Estos dos elementos conjuntamente serán considerados como una plataforma ampliada de 6-dof, permitiendo el posicionamiento deseado del sensor óptico. Por medio de la integración de información visual y medidas inerciales, se buscará lograr una estimación del estado más robusta, para el control de dicha plataforma ampliada.*

Palabras clave: Fly-the-camera, PBIVS, sensor visuo-inercial.

1. INTRODUCCIÓN

Hoy en día, es posible imaginar innumerables aplicaciones para las cuales resulte útil montar una cámara a bordo de un sistema aéreo multirotor, desde aplicaciones tales como inspección de tendidos eléctricos hasta filmaciones en el ámbito de la fotografía o cinematografía. De hecho, en relación con estas últimas aplicaciones, recientemente se ha despertado un gran interés por explotar las posibilidades que ofrece el hecho de hacer volar una cámara con total libertad. Es en este ámbito donde han comenzado a comercializarse sistemas de balancín motorizados (aunque también los hay pasivos) de dos o tres grados de libertad, preparados para ser montados en sistemas multirotor. Gracias a ellos se pretende desacoplar o independizar la orientación de la cámara de las maniobras que, en cada momento, esté realizando la plataforma (figura 1). Abusando de la terminología inglesa, para hacer referencia a este elemento electromecánico usado como balancín, frecuentemente se utiliza el término *camera gimbal* o *gimbal*, a secas. Por su propio origen, estos sistemas están pensados para ser doblemente tele-controlados mediante la intervención simultánea de dos operadores: uno de ellos se encarga del guiado de la plataforma multirotor, mientras que el otro se encarga de orientar la cámara a voluntad.

En el contexto de los sistemas aéreos no tripulados (UAS), en los que se cuenta con una automatización completa del guiado, en la inmensa mayoría de los casos en los que se dispone de una cámara a bordo, ésta se encuentra completamente a merced de las maniobras que la plataforma aérea realice para su desplazamiento, al encontrarse la cámara rígidamente unida a dicha plataforma. Ello impediría, por ejemplo, mantener a un objeto a la vista de forma continua, mientras se realiza una maniobra de desplazamiento en el plano horizontal.

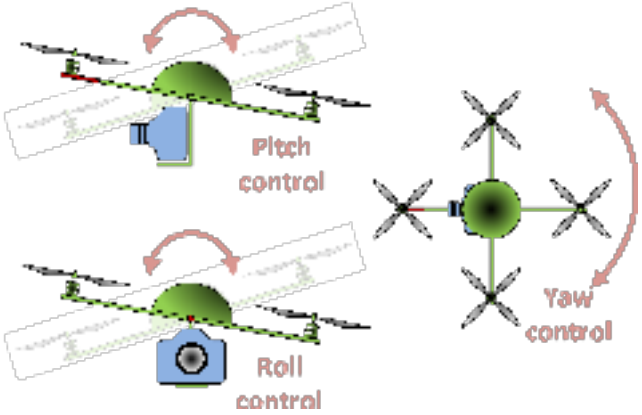


Figura 1: Distintos escenarios de compensación del movimiento de la plataforma utilizando el gimbal para mantener constante la orientación del eje optico.

El objetivo pues de este trabajo, a grandes rasgos, podría interpretarse como la automatización completa del guiado del sistema electromecánico formado por la plataforma multirotor más el gimbal motorizado, permitiendo a la cámara poder recuperar en cierta medida los 6-dof para un posicionamiento libre en el espacio, sin las restricciones impuestas por el acoplamiento rígido a la plataforma multirotor subactuada; esto es, básicamente, dotar de "alas" a la cámara. Desde el punto de vista del control, los 6-dof se descompondrían en dos grupos: los tres grados de libertad de traslación, junto con la guiñada (*yaw*), serían controlados mediante la plataforma multirotor, y por otra parte, los dos grados de libertad de cabeceo (*roll*) y balanceo (*pitch*) de la cámara serían regulados a voluntad mediante el gimbal 2-DoF.

El presente artículo queda estructurado como sigue: en la sección 2 se presenta un breve contexto introductorio de la utilización de los sensores óptico e inercial en la actualidad. Las secciones 3 y 4 contemplan el esquema de control y fusión sensorial propuestos y la arquitectura hardware utilizada, respectivamente. La sección 5 recoge las primeras conclusiones recopiladas tras el estudio en progreso presentado en la propuesta actual.

2. TRABAJO RELACIONADO

Recuperar los 6 grados de libertad de una plataforma aérea (sistemas multivariables, acoplados y subactuados [4, 11]) combinando información visual y medidas provenientes de unidades de medida inerciales para el posicionamiento espacial de cámaras de altas prestaciones es el objetivo central del estudio en curso. La consideración conjunta de ambos elementos se ha dado en llamar muy gráficamente por algunos autores *fly-the-camera* [7, 10, 14], y consiste básicamente en la utilización conjunta (en el mismo vector de estados) de los 4 grados de libertad proporcionados por la plataforma, en combinación con otros 2 grados de libertad que pueden ser obtenidos de un gimbal específico que soporta al sistema óptico. Dicho estudio incluye el aumento del vector de estados para contemplar la identificación en línea tanto del bias de los sensores (en función temporal de la temperatura) como la corrección de posibles deficiencias en la identificación de transformaciones entre sistemas de referencia (figura 2). Si bien la técnica *fly-the-camera* no contempla la fusión sensorial de las medidas provenientes de un sensor óptico y sensores inerciales externos al propio vehículo, en esta propuesta se plantea su inclusión como un único sensor, desde el punto de vista del esquema de control, denominado sensor visuo-inercial.

La idea del uso de cámaras como sensores de alto nivel para realimentación de velocidad o posición en el control de vehículos aéreos no es reciente [2, 9, 17]. La técnica de control a emplear dependerá del conocimiento geométrico previo que se tenga del objeto de interés para la estimación de la relación en posición o velocidad entre cámara y éste, y del espacio de trabajo de la tarea [12], bien sea que se defina el error en el espacio de la imagen (IBVS Image Based Visual Servo) [17] o en el espacio cartesiano (PBVS Position Based Visual Servo) [18].

Respecto al número de lentes o cámaras que son utilizadas, en la actualidad es posible identificar dos grandes tendencias: el uso de sistemas estereoscópicos o RGB-D proporcionan mucha más información para tareas de navegación y localización simultáneas (SLAM); en estas propuestas se obtiene una reconstrucción tridimensional del entorno gracias al cálculo de profundidad proporcionado por la propia cámara, bien por la duplicidad de cámaras o por la inclusión de sensores de profundidad en la medida. Todo ello proporciona información más completa que la que se puede obtener con un sistema monocular convencional, el cual permite, sin embargo, en combinación con otro tipo de información sensorial (desde sensores de distancia por ultrasonidos o infrarrojos hasta sensores de presión barométricos) tareas de navegación y posicionamiento en el espacio[1].

Si para la extracción de información visual del entorno se considera el seguimiento de algún objeto de referencia, las técnicas dependerán del conocimiento previo del modelo geométrico del mismo; aquellas que así lo contemplen permitirán la reconstrucción parcial de la disposición relativa cámara-objeto, siempre y cuando el objeto sea completamente visible por parte de la cámara. Sin embargo, para entornos no estructurados o previamente reconstruidos, otras técnicas contemplan la posibilidad de mantener una relación geométrica denominada homografía entre un marco o *frame* de referencia y la imagen actual. Mediante el emparejamiento continuo de características o puntos relevantes, restringido a objetos planos, es posible obtener una reconstrucción parcial de la orientación y posición (hasta un factor de escala) respecto una referencia determinada [19]. Con técnicas similares, es posible el cierre del lazo de control externo mediante consignas de velocidad, en lugar de estimación en tiempo real de la posición relativa del vehículo, considerando de este modo al sensor óptico como un sensor estimador de velocidad [5].

En la literatura, se pueden encontrar numerosos trabajos de fusión de información estrictamente inercial (giróscopo, acelerómetro, o ampliada con magnetómetro), aunque se trata de un campo bastante maduro hoy en día [3, 16]. Es muy típico encontrar en cualquier plataforma robótica sensores de medidas inerciales, tales como acelerómetros y giróscopos, que combinados con marcos de referencia comunes se conocen ampliamente como IMU (Inertial Measurement Unit).

En los últimos años, gracias a la disminución de costes de producción y al avance en la tecnología, existen otras alternativas más complejas a la citada anteriormente. Buen ejemplo de ello son las denominadas AHRS (Attitude Heading Reference System): son dispositivos que integran diversos sensores de bajo nivel, tales como acelerómetro, giróscopo, magnetómetro o barómetro. La principal diferencia con las IMU consiste en la inclusión de sistemas de procesamiento en placa que proporcionan soluciones de orientación y actitud del

vehículo, medidas más complejas y de mayor nivel que aquellas en bruto proporcionadas de manera individual directamente por los sensores de bajo nivel.

En lo que respecta a la combinación de información visual e inercial, sin embargo, los desarrollos son bastante más recientes y sigue siendo un interesante campo de investigación. En particular, resultan de especial relevancia los trabajos en los que se busca la formulación de observadores visuo-inerciales en el espacio especial Euclídeo $\mathbb{SE}3$ [13].

Actualmente, hasta donde alcanza el conocimiento de los autores, no existen trabajos donde específicamente se combinen el control de una plataforma conjunta multirotor-gimbal con fusión visuo-inercial de información. El objetivo de este trabajo es contribuir al avance en este ámbito, mediante la descripción de un proyecto en fase de desarrollo, en el que se plantea un esquema de control visuo-inercial, inicialmente haciendo uso de una estrategia de fusión muy simple, y describiendo la arquitectura hardware propuesta.

3. ESQUEMA DE CONTROL

En la presente sección, se plantea el esquema general de control en espacio cartesiano de una plataforma aérea, considerando la inclusión del sensor visuo-inercial y recuperando los 6-dof para el posicionamiento aéreo de una cámara con modelo de objeto conocido. A partir de ésta, y en base al conocimiento y tipo del modelo geométrico del objeto, se obtendrá la disposición relativa entre la cámara y el objeto usado como referencia, mediante la identificación de puntos característicos en la imagen.

En trabajos futuros se plantearán otras técnicas, similares a trabajos recientes, donde se considera el uso de niveles de intensidad de los píxeles en la imagen, en lugar del cálculo de descriptores o la extracción de características de más alto nivel, para tareas de localización y navegación (*Visual SLAM* [20]). Estas propuestas son robustas frente a cambios de iluminación u oclusiones parciales, resultan computacionalmente eficientes y permiten la utilización de toda la información codificada en la imagen.

En el caso concreto aquí tratado, como una primera aproximación al problema, se considera el espacio cartesiano como espacio de trabajo y modelo geométrico de objeto conocido para el control visuo-inercial basado en posición (*PBIVS Position Based Inertio-Visual Servoing*) de la plataforma multirotor.

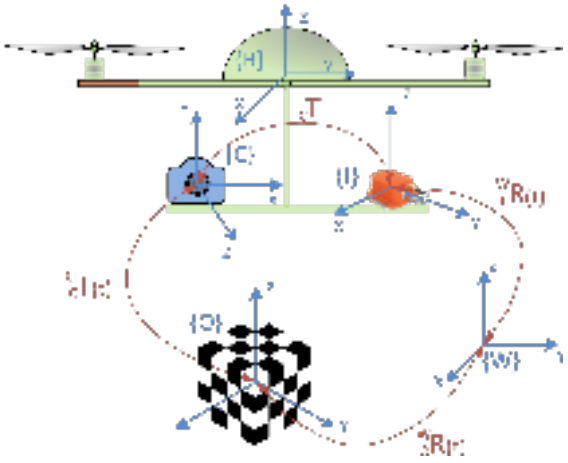


Figura 2: Sistemas de coordenadas involucrados.

3.1. NOTACIÓN Y SISTEMAS DE REFERENCIA

Para el caso particular de un esquema de control en el espacio cartesiano, en base a objeto de referencia visual cuyo modelo geométrico se asume conocido *a priori*, es posible componer un diagrama de interrelaciones geométricas como el mostrado en la figura 2, donde se muestran todos los sistemas de referencia involucrados y las transformaciones que pueden ser requeridas *a priori*.

$\{\mathcal{I}\}$: *IMU coordinate frame*. Se dispondrán de dos IMUs diferentes: la que forma parte del sensor visuo-inercial, que será considerada como IMU principal,$\{\mathcal{I}\}$, y la que se emplea para la mera estabilización de la plataforma, $\{\mathcal{I}_2\}$, no mostrada en la figura.

$\{\mathcal{W}\}$: *World coordinate frame*. Si se hace coincidente con el sistema de referencia inercial al que se refieren las medidas de la IMU, también se puede denominar *Inertial frame* o *Earth reference frame*. La posición absoluta de este sistema de referencia puede fijarse arbitrariamente, al sólo estar definida su orientación.

$\{\mathcal{O}\}$: *Reference-object coordinate frame*. Ligado al objeto, necesario para esquemas de control visual basado en modelo de objeto conocido, en base al cual se supone definido el modelo geométrico del mismo. En principio, se asumirá objeto inmóvil.

$\{\mathcal{C}\}$: *Camera coordinate frame*. Sistema de coordenadas ligado a la cámara.

$\{\mathcal{B}\}$: *Body coordinate frame*. Sistema de coordenadas ligado a la plataforma, cuya posición se asume coincidente con el centro de masas del mismo.

Notación para vectores y matrices: $\boldsymbol{t} \in \mathbb{R}^3$: vector de posición; $\boldsymbol{v} \in \mathbb{R}^3$(alternativamente $\dot{\boldsymbol{t}}$): vector velocidad lineal; $\boldsymbol{a} \in \mathbb{R}^3$(alternativamente $\ddot{\boldsymbol{t}}$):

vector aceleración lineal; $\boldsymbol{R} \in \mathbb{SO}3$: matriz de rotación indicando la orientación relativa entre dos sistemas de referencia $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \in \mathbb{SE}3$: matriz de transformación homogénea; $\boldsymbol{\theta} = [\phi\ \theta\ \psi]^T$: tripleta de ángulos *roll*, *pitch* y *yaw* respectivamente; $\boldsymbol{\omega} \in \mathbb{R}^3$: vector velocidad angular.

Así, por ejemplo, la posición del sistema de coordenadas de la IMU principal $\{\mathcal{I}\}$, respecto al universal $\{\mathcal{W}\}$, sería: ${}^{\mathcal{W}}_{\mathcal{I}}\boldsymbol{t}$, mientras que la orientación vendría dada por ${}^{\mathcal{W}}_{\mathcal{I}}\boldsymbol{R}$, mientras que para hacer referencia a ambos de manera conjunta se usaría ${}^{\mathcal{W}}_{\mathcal{I}}\boldsymbol{T}$.

3.2. FUSIÓN SENSORIAL

Los algoritmos de visión empleados en este caso permitirán estimar la posición y orientación del objeto respecto al sistema de referencia ligado a la cámara (o de forma equivalente, la posición y orientación de la cámara respecto al sistema de coordenadas ligado al objeto). Como premisas, es necesario que se haya resuelto fuera de línea el problema de la *calibración interna* de la cámara. Gracias a esta calibración, se dispondrá de una estimación de los parámetros intrínsecos de la cámara, tales como distancia focal, dimensión del píxel, punto central, etc.

Del mismo modo se puede suponer que se dispone de una estimación de la transformación existente entre los sistemas de coordenadas $\{\mathcal{C}\}$ y $\{\mathcal{I}\}$: ${}^{\mathcal{I}}_{\mathcal{C}}\hat{\boldsymbol{T}}$. Esta matriz se habrá obtenido como parte de un proceso de calibración IMU-cámara a realizar fuera de línea. Inicialmente, esta relación sería constante, una vez que se hubiera establecido el acoplamiento mecánico rígido entre la cámara y la IMU. Sin embargo, se puede proponer la estimación *online* de esta transformación, por la importante repercusión que puedan tener los eventuales errores resultantes de dicha calibración [13]. En este caso, la citada transformación sería dependiente del tiempo: ${}^{\mathcal{I}}_{\mathcal{C}}\hat{\boldsymbol{T}}(t)$. Cuando se usa un modelo de objeto conocido, se suele denominar proceso de *calibración externa* al proceso de estimación en tiempo real de la disposición relativa entre cámara y objeto ${}^{\mathcal{O}}_{\mathcal{C}}\hat{\boldsymbol{T}}(t)$ en cada instante. Para el planteamiento del problema de control en el espacio cartesiano, se asume que se dispondrá de una posición y orientación de referencia absoluta para la cámara en cada instante: ${}^{\mathcal{W}}_{\mathcal{C}}\boldsymbol{T}^*(t)$. El objetivo consistirá en regular la disposición actual que en cada momento se estime, ${}^{\mathcal{W}}_{\mathcal{C}}\hat{\boldsymbol{T}}(t)$, para llevarla a que coincida con la citada referencia.

La estimación que, en cada momento, se haga de ${}^{\mathcal{W}}_{\mathcal{C}}\hat{\boldsymbol{T}}(t)$, será el resultado de fusionar medidas de la cámara y de la IMU. En particular, la componente de rotación de esta transformación, ${}^{\mathcal{W}}_{\mathcal{C}}\hat{\boldsymbol{R}}(t)$, se obtendría como:

$$ {}^{\mathcal{W}}_{\mathcal{C}}\hat{\boldsymbol{R}} = {}^{\mathcal{W}}_{\mathcal{I}}\hat{\boldsymbol{R}} \cdot {}^{\mathcal{I}}_{\mathcal{C}}\hat{\boldsymbol{R}} \tag{1} $$

En lo que respecta a la componente de traslación, ${}^{\mathcal{W}}_{\mathcal{C}}\hat{\boldsymbol{t}}(t)$, dado que la posición del sistema de coordenadas inercial no está determinada y puede ubicarse arbitrariamente (no así su orientación), puede plantearse que el origen de dicho sistema de coordenadas $\{\mathcal{W}\}$ sea coincidente con el del objeto $\{\mathcal{O}\}$. Para que esta suposición sea coherente con la información de aceleración lineal que se obtiene de la IMU (en caso de que ésta fuera usada), el objeto debe permanecer estacionario. De esta forma, es posible establecer la relación

$$ {}^{\mathcal{W}}_{\mathcal{C}}\hat{\boldsymbol{t}} \triangleq {}^{\mathcal{W}}\left({}^{\mathcal{O}}_{\mathcal{C}}\hat{\boldsymbol{t}}\right) = {}^{\mathcal{W}}_{\mathcal{O}}\hat{\boldsymbol{R}} \cdot {}^{\mathcal{O}}_{\mathcal{C}}\hat{\boldsymbol{t}} \tag{2} $$

donde ${}^{\mathcal{W}}_{\mathcal{O}}\hat{\boldsymbol{R}}(t)$ puede obtenerse de la siguiente forma:

$$ {}^{\mathcal{W}}_{\mathcal{O}}\hat{\boldsymbol{R}} = {}^{\mathcal{W}}_{\mathcal{C}}\hat{\boldsymbol{R}} \cdot {}^{\mathcal{O}}_{\mathcal{C}}\hat{\boldsymbol{R}}^T \tag{3} $$

matriz que se puede asumir que es constante si el objeto efectivamente permaneciera inmóvil y no hubiera derivas apreciables en la estimación de orientación proporcionada por la IMU. En la figura 3, se muestra un esquema elemental de fusión, en el que se ignoran las medidas nativas de la IMU y se asume que ésta proporciona medidas elaboradas en forma de posición angular, mediante hardware específico de fusión inercial (Sección 4.1.1).

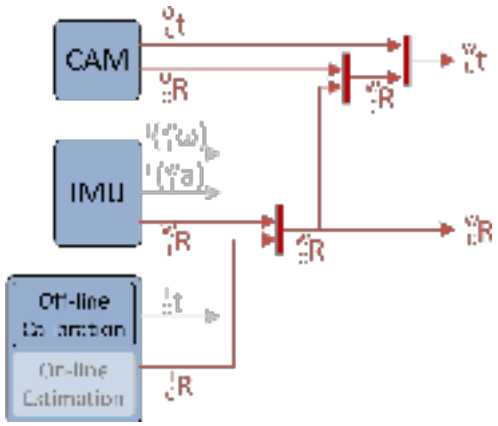


Figura 3: Esquema para la estimación elemental de la disposición absoluta de la cámara.

Dado que la posición del objeto $\{\mathcal{O}\}$ se usa como única referencia absoluta de posición para el control, si éste dejara de ser inmóvil en algún momento, podría considerarse como una perturbación a compensar por el el sistema de control. Desde el punto de vista práctico, para una referencia ${}^{\mathcal{W}}_{\mathcal{C}}\boldsymbol{T}^*$ constante, se apreciaría que la cámara se desplazaría para mantener la posición relativa (no así la orientación) respecto al objeto.

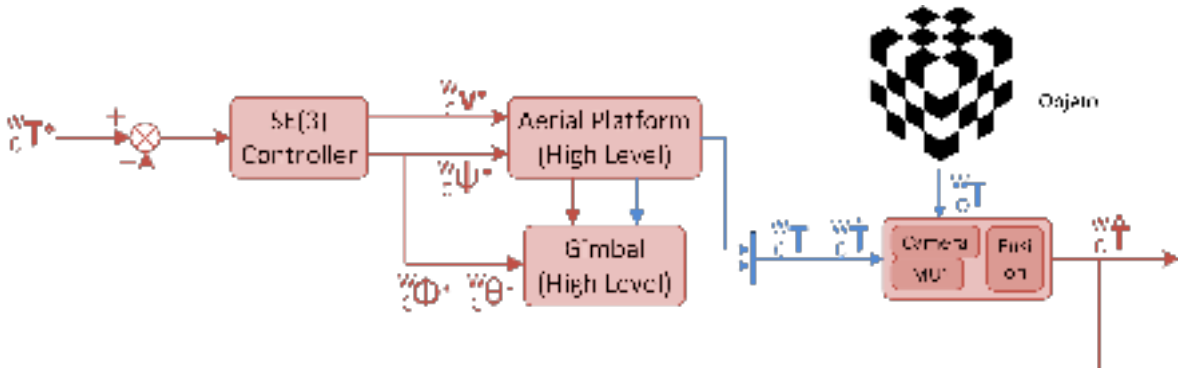


Figura 4: Diagrama de bloques mostrando el lazo de control de alto nivel en su variante control cartesiano basado en modelo geométrico de objeto.

3.3. ESQUEMA DE CONTROL

En la figura 4, se muestra el diagrama de bloques de alto nivel siguiendo la estrategia indicada: en rojo las señales que se transmiten a través del lazo de control, mientras que en azul quedan representadas las posibles dependencias cinemáticas (el movimiento al que estén sometidos en cada momento la plataforma aérea y el gimbal influyen obviamente en las observaciones o medidas de la IMU y la cámara). Aunque es posible plantear el control externo sólo con información visual, en el caso de que eventualmente no fuera posible encontrar al objeto de referencia en el campo de visión, bastaría con mantener ${}^{\mathcal{W}}_{c}\hat{\boldsymbol{t}}$ igual al último ciclo en el que se obtuvieran medidas de la disposición del objeto e ir actualizando dicho valor con predicciones en base a los datos provenientes de la IMU.

La descomposición y vista interna del control de los dos sistemas de alto nivel que componen la propuesta actual se presenta en detalle en la figura 5: los bloques *Aerial Platform (High Level)* y *Gimbal (High Level)*. El primero de ellos implementaría el control desde el más alto nivel del sistema multirotor, visto como un dispositivo al que se le pasan referencias de velocidad lineal y de orientación en *yaw*. Ciertamente se trata de un esquema de control anidado, donde el lazo más externo controla la traslación, proporcionando empuje y consignas de ángulos para el lazo interno de estabilización, el cual también se encarga del control de guiñada (*yaw*).

Como señales de actuación producidas por ambos lazos resultan el empuje T y los pares de estabilización en *roll*, *pitch* y *yaw*: $\boldsymbol{\tau}_{\mathcal{B}} = [\tau_\phi\ \tau_\theta\ \tau_\psi]^T$. Estos dos constituirán la actuación a bajo nivel sobre la plataforma, conformando un vector de actuación $\boldsymbol{u}_{\mathcal{B}} = \begin{bmatrix} T & \boldsymbol{\tau}_{\mathcal{B}} \end{bmatrix}^T \in \mathbb{R}^4$. Como se ilustra en la figura, en respuesta a ese empuje y vector par, es necesario el cálculo de la combinación correspondiente de las velocidades de los rotores, $\Omega_1 \dots \Omega_n$ (siendo n el número de rotores) que conformarán la referencia en velocidad para el control en cascada implementado en los ESC (Electronic Speed Controller) a bajo nivel.

En lo que respecta al bloque *Gimbal (High Level)*, a partir de los ángulos de orientación deseados para la cámara, se controlan los dos grados de libertad de la plataforma para satisfacer dichas referencias. Como se ve en la figura, este bloque recibe la referencia angular para los dos grados de libertad de la cámara que pueden corregirse mediante el gimbal. Asimismo, se reciben los ángulos de referencia fijados para la propia plataforma, dado que pueden suponer una información de prealimentación notablemente útil. La señal de actuación de bajo nivel en este caso puede asumirse también que será en forma de par aplicado a cada motor, $\boldsymbol{u}_{\mathcal{G}}$.

Conviene discutir un aspecto particular, como es el hecho de que la IMU principal se emplea tanto para el control a bajo nivel del gimbal, como para el control de alto nivel de todo el sistema, en este último caso, en combinación con la cámara. No obstante, cabe esperar que este lazo interno use la información proporcionada por esta IMU a una frecuencia superior a la empleada en el lazo de realimentación combinada cámara+IMU. La alternativa que evitara hacer este doble uso de la IMU principal sería que el balancín no tuviera un lazo de control interno, sino que se controlara directamente mediante el lazo externo. Dado que, en este primer esquema, se plantea un control de doble lazo para la plataforma aérea, se ha preferido mantener esa misma filosofía para el otro elemento.

3.3.1. Variante con control cartesiano directo

Manteniendo el mismo esquema de control en espacio cartesiano, en base a un objeto de geo-

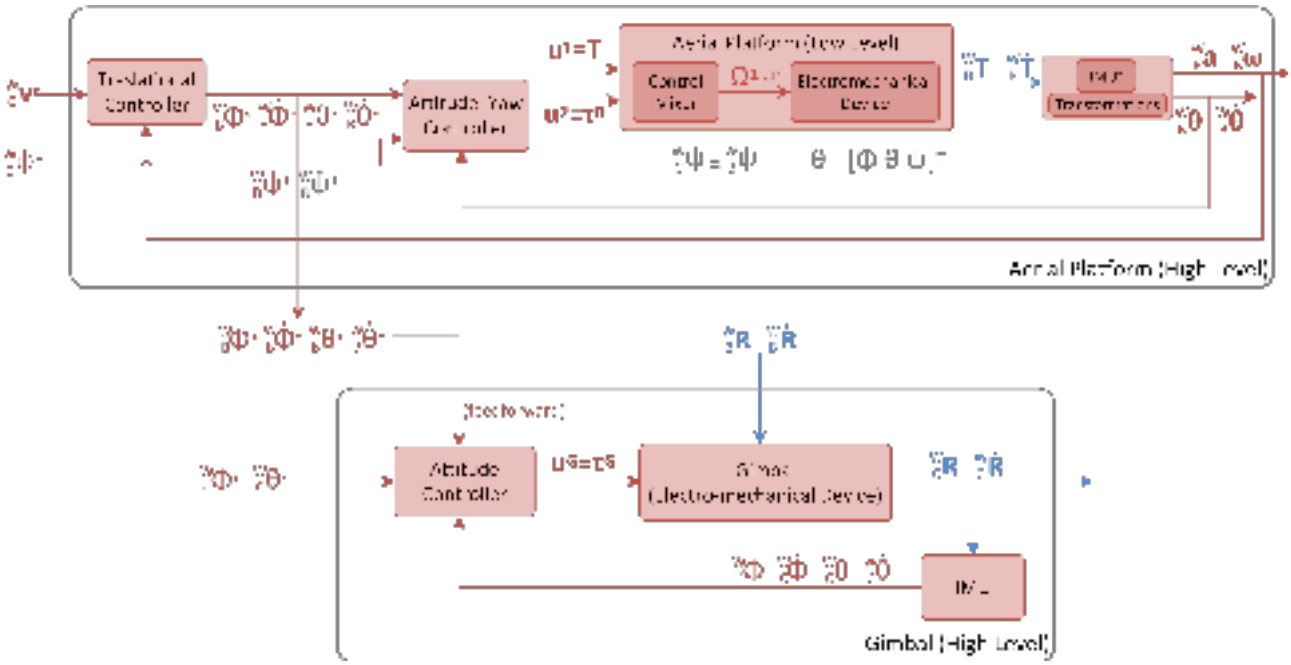


Figura 5: Vista interna de los bloques *Aerial Platform (High Level)* y *Gimbal (High Level)*.

metría conocida, puede plantearse una alternativa más ambiciosa, consistente en implementar un único lazo de control que, directamente a partir de las medidas sensoriales, actúe sobre los pares del sistema electromecánico conjunto plataforma multirotor-gimbal. En este caso, en que se emplea un esquema de control unificado de los dos elementos electromecánicos, sería posible definir un vector de actuación completo, $\boldsymbol{u}$, mediante la combinación de las actuaciones de bajo nivel ya definidas $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_{\mathcal{B}} & \boldsymbol{u}_{\mathcal{G}} \end{bmatrix}$.

4. ARQUITECTURA HARDWARE

4.1. DESCRIPCIÓN GENERAL

Partiendo del esquema de la figura 6, se procede a describir someramente cada uno de los niveles funcionales de la arquitectura modular propuesta, siguiendo un planteamiento *bottom-up*.

4.1.1. Controlador de gimbal

En el nivel más bajo de la arquitectura se dispone de una placa de control de motores para el posicionamiento angular mediante motores eléctricos BLDC (*Brushless DC*) de una plataforma en la cual está dispuesta la cámara y la IMU principal del sistema $\{\mathcal{I}\}$. Gracias al sensado de medidas filtradas de la posición angular, utilizando el DMP (*Digital Motion Processor*) de la IMU *MPU6050* de Invesense empleada (acoplada de manera adecuada a la cámara, y comunicación mediante bus digital I^2C), no es necesario la inclusión de sensores adicionales para cerrar el lazo y controlar la posición o velocidad de los motores, como sensores de efecto *Hall*, tacómetros o *encoders*. La implementación de las señales sinusoidales para cada una de las fases del motor, con un desplazamiento de fase entre sí de 120° utilizando un procesador de 8 bits a 16 Mhz *ATmega* de *Atmel*, se lleva a cabo mediante la generación de señales de pulsos PWM.

Los motores utilizados son típicos brushless que, sin embargo, no trabajan en un régimen de operación típico. Tienen la particularidad de un devanado extra de cada una de las bobinas con cables más finos, para aumentar su resistencia y par que permita mantener la posición de la cámara. A diferencia de los rotores, que a bajo nivel son controlados en velocidad mediante los ESC, estos motores se controlan en posición directamente a través del microcontrolador y *drivers* hardware específicos.

4.1.2. Unidad de control de vuelo

Aunque es posible en la actualidad adquirir una FCU (Flight Control Unit) por costes muy asequibles, todas las alternativas existentes están muy limitadas en cuanto a la capacidad computacional requerida para el proyecto. Si bien la implementación más extendida, Arducopter [8, 15], hace uso del procesador más elevado de la familia de Arduino (ATMega 2560), sigue siendo un microprocesador con especificaciones muy limitadas que incluso utiliza emulación de tipo de dato flotante para las operaciones más comunes, disminu-

yendo de forma drástica el rendimiento deseado. Así, para la implementación del proyecto, se ha tomado como referencia una placa de evaluación de la familia de microprocesadores ARM® de Texas Instruments, más concretamente el modelo Stellaris EK-LM4F120XL, con un procesador ARM Cortex-M4 de 32 bits a una frecuencia de funcionamiento de 80 Mhz con unidad de punto flotante.

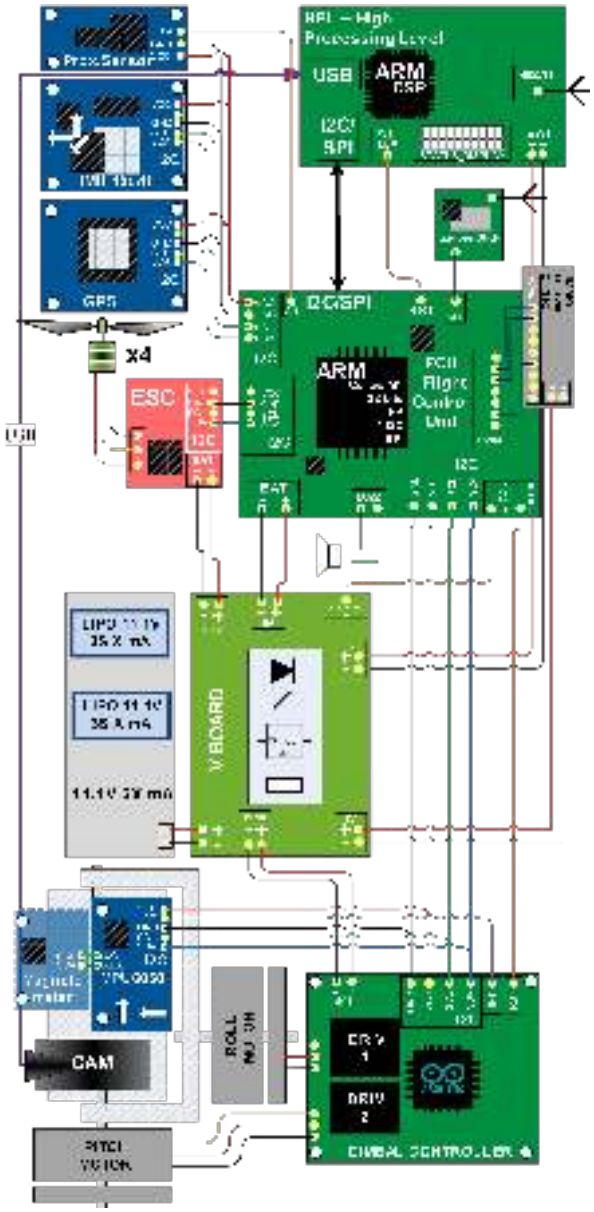


Figura 6: Esquema de conexionado del hardware de la plataforma propuesta.

Uno de los requisitos que se satisfacen con creces es el número de *GPIOS* (*General Programmable Input Outputs*), con lo que para este nivel se estima óptima por el compromiso entre la cantidad de sensores y actuadores controlables a bajo nivel y su gran capacidad computacional. Para cuestiones de telemetría y configuración, se hace uso de un adaptador Zigbee para comunicación con una estación base, dado no es necesario un radioenlace de grandes prestaciones. La IMU secundaria $\{\mathcal{I}_2\}$ utilizada en este nivel es una placa de bajo coste que integra hasta 4 sensores diferentes: acelerómetro, giróscopo, magnetómetro y barómetro, hasta llegar a los 10-DoF, con acceso desde el microprocesador de la FCU mediante bus digital I^2C.

4.1.3. HPL High Processing Level

Este nivel de la arquitectura es el responsable del cierre del lazo de control a alto nivel mediante consignas de posición en el espacio utilizando para ello información visuo-inercial, obtenida del conjunto cámara-IMU. Es deseable que este nivel implemente hardware de aceleración 3D específico para el procesamiento de instrucciones multimedia necesario para el tratamiento en tiempo real de las imágenes. La tendencia durante los últimos años ha provocado un cambio en el tipo de hardware utilizado en este nivel: arquitecturas clásicas como PC104 han sido relegadas por procesadores mucho más potentes, pequeños y eficientes o incluso por otros estándares similares como los *QSeven*. Ejemplos de esta categoría son la *Gumstix Overo*® ARM Cortex-A8,que incluyen DSP y hardware de aceleración 3D; o la serie de procesadores ARM *NEON*™ de TI, que permiten comunicación tanto analógica como digital a bajo nivel. Aunque existen otras alternativas como la *BeagleBoard*, con muy buena relación consumo-rendimiento, no son aptas para este tipo de proyectos debido a la escasez de canales de comunicación de bajo nivel.

Respecto a las comunicaciones, se utiliza un radio enlace Wifi 802.11g, el cual permite un gran ancho de banda para el envío ocasional de imágenes de gran tamaño y largas distancias del canal para futuros usos complementarios, como comunicaciones con otros equipos aéreos.

4.1.4. Software

Respecto al software utilizado en cada uno de los niveles, tanto la FCU como la placa de control de gimbal ejecutan código estándar ANSI C/C++ compilado para cada microprocesador en particular. Un elemento clave en el diseño del proyecto es la definición de interfaces de comunicación y acceso a los distintos servicios de bajo nivel, de manera que, desde alto nivel o incluso desde una estación en tierra, cualquier elemento de la arquitectura pueda ser accedido y evaluado, desde medidas en bruto de los sensores inerciales hasta los diversos parámetros de los controladores implementados.

Como SO de alto nivel se propone una distribución de Linux específica para la plataforma (ARM) con un conjunto de librerías específicas para ROS (*Robot Operating System*), necesarias para la monito-

rización de nodos o *topics* específicos, el estado de los distintos métodos de fusión y la implantación de código de control de alto nivel, entre otros.

Tras un estudio exhaustivo de las distintas librerías de visión artificial que en la actualidad es posible implantar en sistemas de tiempo real (*OpenCV*, *Open Cloud Library u Open Tracking Library* entre otras), finalmente se decidió por utilizar la implementación del algoritmo de visual SLAM [6] para la estimación parcial de la posición y orientación de la plataforma mediante la utilización de un único sensor óptico. PTAM (*Parallel Tracking and Mapping*), denominación de la implementación de dicho algoritmo, está disponible para ROS y su uso está cada vez más extendido en aplicaciones de sistemas monoculares. Son varias las ventajas que supone la utilización de esta librería para esta aplicación en el futuro: permite dividir la complejidad del problema de SLAM en dos hilos simultáneos separados (reconstrucción del mapa y estimación de la posición); puede ser utilizada en plataformas limitadas por potencia y consumo y dispone de paquetes disponibles para el *framework* de ROS.

5. CONCLUSIONES

En el presente trabajo, se ha presentado el desarrollo en curso de una técnica para el control de una plataforma multirotor subactuada, complementada con un gimbal, a través de las medidas provenientes de un sensor visuo-inercial, instalado sobre dicho gimbal, con el objetivo del posicionamiento 6-dof de una cámara en el aire. El conjunto de información visual e inercial servirá para la estimación de posición y orientación relativa que mejorará el proceso de estimación del estado, haciendo uso de una arquitectura modular de bajo coste. Varias líneas de trabajo quedan abiertas: desde el análisis y diseño de observadores más sofisticados, hasta el control unificado de la plataforma ampliada, pasando por la resolución de problemas de implementación utilizando librerías de visión por computador y su integración en tiempo real.

Agradecimientos

Los autores quieren expresar su agradecimiento al MECD por la financiación de este trabajo, a través de los proyectos DPI2012-37580-C02-02 y DPI2010-19154.

Referencias

[1] M. Achtelik, M. Achtelik, S. Weiss, R.Siegwart. "Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments". *2011 IEEE International Conference on Robotics and Automation.* Shanghai International Conference Center May 9-13, 2011, Shanghai, China.

[2] E. Altug, J.P. Ostrowski, R. Mahony. Control of a "Quadro-tor Helicopter Using Visual Feedback". *Robotics and Automation, 2002. Proceedings. ICRA '02*. IEEE International Conference on Vol.1 72-77, 2002.

[3] G. Baldwin, R.Mahony, J. Trumpf, T. Hamel, T Cheviron. "Complementary filter design on the Special Euclidean group SE (3)." *2005 Conference on Decision and Control* and *2005 European Control Conference*. CDC-ECC '05. 44th IEEE.

[4] P. Castillo, P. Garcia, R. Lozano, P. Albertos. "Modelado y Estabilizacion de un Helicóptero con cuatro rotores". *RIAI,* ISSN: 1697-7912 4:41-57, 2007.

[5] D. Honegger, P. Greisen, L. Meier, P. Tanskanen, M. Pollefeys. "Real-time Velocity Estimation Based on Optical Flow and Disparity Matching". *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura, Portugal. Octubre 2012.

[6] G. Klein, D. Murray. "Parallel tracking and mapping for small ar workspaces". *International Symposium on Mixed and Augmented Reality*, Nov. 2007, pp. 225–234.

[7] D. Lee, V. Chitrakaran, T. Burg, D. Dawson, B. Xian. "Control of a Remotely Operated Quadrotor Aerial Vehicle and Camera Unit Using a Fly-The-Camera Perspective". *Proceedings of the 46th IEEE Conference on Decision and Control.* New Orleans, LA, USA, Dec. 12-14, 2007.

[8] C. Liu, O. McAree, W. Chen. "Path following for small UAVs in the presence of wind disturbance". *UKACC International Conference on Control 2012.* Cardiff, UK. Septiembre 2012.

[9] L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys. "PIXHAWK Project: A System for Autonomous Flight using Onboard Computer Vision". ETH Zurich.

[10] A.E. Neff, D. Lee, V.K. Chitrakaran, D. M. Dawson, T. C. Burg. "Velocity Control for a Quad-Rotor UAV Fly-By-CameraInterface". Clemson University. 2007.

[11] G. V. Rao, M. G. Ortega, F. R. Rubio. "Robust Backstepping/Nonlinear H_∞ control for path tracking of a quadrotor unmanned aerial vehicle". *Proceedings American Control Conference (Acc'08)* Seattle. Wa. Usa, 2008.

[12] C. Samson, M. Leborgne, B. Espiau. "Robot control. The Task-Function approach." *In Oxford Engineering Science Series, vol. 22,* Oxford University Press, 1991.

[13] G. G. Scandaroli, P. Morin, G. Silveira. "A Nonlinear Observer Approach for Concurrent Estimation of Pose, IMU Bias and Camera-to-IMU Rotation". *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, CA, USA. Septiembre 2011.

[14] S. M. Weiss. "Vision Based Navigation for Micro Helicopters". Tésis. *DISS. ETH NO. 20305*. 2012.

[15] C.B. Yigit, E. Altug. "Visual Attitude Stabilization of a Unmanned Helicopter in Unknown Environments with an Embedded Single-board Computer". Dept. of Mech.Eng., Instanbul Tech Univ. Instanbul. Turkey.

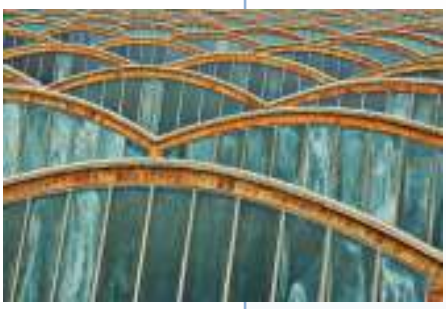
[16] Multisensor Data Fusion. Cap. 25. Springer Handbook of Robotics. 2008.

[17] N. Guenard, T. Hamel, R. Mahony. "A Practical Visual Servo Control for an Unmanned Aerial Vehicle". *IEEE Transactions on Robotics*, Vol.24, Abril 2008.

[18] E. Malis. "Survey of Vision-Based Robot Control". *ENSIETA European Naval Ship Design Short Course*, Brest, Francia, Abril, 2002.

[19] S. Benhimane, E. Malis. "Homography-based 2D Visual Servoing". *Proceedings of the 2006 IEEE International Conference on Robotics and Automation.* Orlando, Florida - Mayo 2006.

[20] G. Silveira, E. Malis, P. Rives. "An Efficient Direct Approach to Visual SLAM". *IEEE Transactions on Robotics.* Vol. 24. NO5. Octubre 2008.











Patrocinadores:

ISBN: 978-84-616-5063-7