



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de La Paz



Robótica y Computación. Nuevos Avances



**TECNOLÓGICO
NACIONAL DE MÉXICO**

Coordinadores de la Edición
Iliana Castro Liera
Mario Cortés Larrinaga



ISBN: 978-607-98174-6-6



9 786079 817466

Esta página fue dejada en blanco intencionalmente.



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de La Paz

Robótica y Computación. Nuevos Avances

D. R. ©INSTITUTO TECNOLÓGICO DE LA PAZ
Blvd. Forjadores de B.C.S. No. 4720
Col. 8 de octubre
C.P. 23080
La Paz Baja California Sur, México

Primera edición mayo de 2020
ISBN: 978-607-98174-6-6



Reservados todos los derechos. Ninguna parte de este libro puede ser reproducida, archivada o transmitida, en cualquier sistema –electrónico, mecánico, de fotorreproducción, de almacenamiento en memoria o cualquier otro– sin hacerse acreedor a las sanciones establecidas en las leyes, salvo con el permiso escrito del titular de los derechos de la obra.

Diseño de la portada: Iliana Castro Liera

Esta página fue dejada en blanco intencionalmente.

Coordinador General

M.C. Juan Pablo Morales Álvarez - Tecnológico Nacional de México / I. T. de La Paz

Coordinador del Programa Técnico

Dr. Marco Antonio Castro Liera - Tecnológico Nacional de México / I. T. de La Paz

Comité Revisor

Dr. Alejandro Israel Barranco Gutiérrez - Tecnológico Nacional de México / I. T. de Celaya

Dr. Eduardo Javier Moreno Valenzuela - CITEDI Tijuana

Dr. Gustavo Alberto Cordero Hernández - Tecnológico Nacional de México / I. T. de La Laguna

Dr. Rubén Alejandro Garrido Moctezuma, CINVESTAV

Dr. Alejandro Enrique Dzul López - Tecnológico Nacional de México / I. T. de La Laguna

Dr. Eusebio Bugarín Carlos - Tecnológico Nacional de México / I. T. de Ensenada

Dr. Miguel Ángel Llama Leal - Tecnológico Nacional de México / I. T. de La Laguna

Dra. Ana Yaveni Aguilar Bustos - Tecnológico Nacional de México / I. T. de Ensenada

Dr. José Armando Sáenz Esqueda - Universidad Juárez del Estado de Durango

Dr. Isaac Gandarilla Esparza - Tecnológico Nacional de México / I. T. de La Laguna

Dr. Emmanuel Nuño Ortega - Universidad de Guadalajara

Dr. Ricardo Emmanuel Campa Cocom - Tecnológico Nacional de México / I. T. de La Laguna

Dr. Víctor Adrián Santibáñez Dávila - Tecnológico Nacional de México / I. T. de La Laguna

Dr. Jesús Alberto Sandoval Galarza - Tecnológico Nacional de México / I. T. de La Paz

M.S.C. Iliana Castro Liera - Tecnológico Nacional de México / I. T. de La Paz

Dr. Saúl Martínez Díaz - Tecnológico Nacional de México / I. T. de La Paz

Dr. Israel Marcos Santillán Méndez - Tecnológico Nacional de México / I. T. de La Paz

Dr. Marco Antonio Castro Liera - Tecnológico Nacional de México / I. T. de La Paz

M.S.C. José Luis Gómez Torres - Tecnológico Nacional de México / I. T. de La Paz

M.S.C. Joel Artemio Morales Viscaya - Tecnológico Nacional de México / I. T. de La Paz

M.S.C. César Higuera Verdugo - Tecnológico Nacional de México / I. T. de La Paz

Coordinadores de la edición

M.S.C. Iliana Castro Liera - Tecnológico Nacional de México / I. T. de La Paz

Dr. Mario Cortés Larrinaga - Tecnológico Nacional de México / I. T. de La Paz

Índice general

Portada	I
Comités	V
Índice	VI
Diseño de un controlador I-LOS para el seguimiento de trayectoria en un ASV . .	1
Parameter Estimation and Delay-Based Control of a Linear Ultrasonic Motor . . .	9
Using Active Disturbance Rejection for Chaotizing a class of Linear Systems . . .	16
Control Conexionista de un Vehículo Autónomo	22
Estudio comparativo de servomotores de CD orientados a la construcción de pro- totipos educativos	32
Control Robusto a Perturbaciones Basado en Redes Neuronales para Seguimiento de Trayectorias de Quadrotores	41
Control de Orden Fraccionario para el seguimiento de trayectorias en un Robot Móvil con Ruedas	49
Acotamiento de soluciones en sistemas lineales: Un enfoque de control difuso . . .	56
Solución a la cinemática directa con cuaterniones duales de un péndulo doble y simulación gráfica en un sistema embebido	62
Modelado, control y optimización de ganancias de un vehículo aéreo no tripulado	67
Control IDA-PBC en voltaje para la regulación de posición de un sistema ca- rro-péndulo	80
Modelado de sistemas mediante regresión basada en un AG	86
Diseño de una Red Neuronal en FPGA para el Modelado de Sistemas Mecatrónicos	94
Diseño de un Perfilador Hidrográfico Autónomo Costero	104
Implementacion electrónica y análisis de la dinámica de un sistema regulador glucosa-insulina de orden fraccionario con retardos temporales	110
An embedded platform for Internet of Things Applications	119
Monitorización de la integridad estructural para aerogeneradores offshore con es- trutura tipo jacket mediante métodos de aprendizaje automático	128
A Genetic Algorithm Applied to the Nesting of Cutting Patterns in Finite Materials	135
Diversificación de un portafolio de inversión por medio de un Algoritmo Genético con codificación real	141
Clasificación de un Disparo de Arma de Fuego Corta por Medio de los Coeficientes del Coseno de Fourier	148
Red Neuronal para la Detección de Datos Anómalos en Dispositivos IoT	153
Marca de Agua Invisible en Imágenes Digitales Utilizando IWT-DCT y Mapas Caóticos	158

Sistema Esteganográfico Reversible en Imágenes Estereoscópicas por Modificación del Histograma	163
Esteganografía en Imágenes a Color mediante DWT y Método de Tinkerbelle . . .	169
Generación de modelos digitales de elevación mediante Interferometría de Radar de Apertura Sintética	175
Monitoreo de sargazo en el Mar Caribe	182
SCI-HI: Pruebas de un radio-interferómetro de dos elementos implementado en Roach2 con ancho de banda 40-120MHz	186
Diseño de un Gateway Lora/WiFi como punto de acceso a una red de largo alcance para mensajería de texto	191
Emulación del movimiento del cuello humano con un robot de 3 grados de libertad	198
Diseño en Arduino de un Sistema de Seguimiento de Trayectorias Caóticas Esféricas para Robot Móvil	205
Attitude Estimation of a Mobile Robot by Data Fusion and a Lower Heading Reference System	211
Aplicación de las Ecuaciones de Poincare-Chetaev-Rumyantsev al Modelado Dinámico de Mecanismos Robóticos	217
Seguimiento de Trayectoria de un Robot Móvil Omnidireccional en Plano Inclinado	223
Determinación de los Parámetros Geométricos de un Mecanismo Plano de Cinco Barras Mediante una Técnica de Calibración de Cámara	229
Análisis e Implementación de una Red Neuronal Convolutiva para la Estimación de Pose Utilizando un Sistema de Visión Monocular	236
Implementación parcial del algoritmo SURF en FPGA	241
Aplicación web para la obtención y visualización de datos provenientes de perfiladores autónomos Argo	248
Diseño de una aplicación web para control de expedientes médicos del Hospital General Progreso CAAPS	256
Diseño de una Aplicación Móvil para el control de una flotilla vehicular de Recolección de Residuos Sólidos Urbanos	263
Patrones de acceso de una aplicación basada en Scrum para la gestión de proyectos de desarrollo de software	273
Diseño de una aplicación web para el control del historial clínico, implementando el estándar HL7	278
Implementación del backend de una aplicación web para el mantenimiento de infraestructura y equipo en el ITSM	286
Diseño de sistema de cobro por medio de transferencias para el uso en instituciones públicas	291
Diseño de un sistema Web para toma de decisiones en la asignación de operadores en el SIT ACABÚS	299

Diseño de un controlador I-LOS para el seguimiento de trayectoria en un ASV

Jorge Ernesto Fernández Gómez, Víctor Manuel Ramírez Rivera ¹, Francisco del Río Rivera

Sistemas Híbridos de Energía. Unidad de Energía Renovable
Centro de Investigación Científica de Yucatán, Mérida, México.

¹ Autor de correspondencia: victor.ramirez@cicy.mx
email: jorgeefg1991@gmail.com, francisco.delrio.rivera@gmail.com

Resumen—: El desarrollo de los vehículos autónomos de superficie es un tema de gran interés y actualidad en la comunidad científica internacional. Dotarlos de un sistema de control de movimiento es de vital importancia para el cumplimiento de misiones sin intervención humana. Con ese objetivo, esta investigación emplea la estrategia de control basada en la ley de guiado LOS sobre el modelo dinámico del ASV – *Cheemkin*. Una acción integral es incorporada al controlador para contrarrestar el efecto de las perturbaciones marinas sobre la embarcación. Todo lo anterior se efectúa mediante simulación para demostrar la validez del controlador propuesto en el seguimiento de una trayectoria en línea recta previamente definida.

Keywords—Control Integral, Vehículo autónomo marino, Line-of-sight

I. INTRODUCCIÓN

El desarrollo de vehículos autónomos que navegan en la superficie del mar (*ASV*, Autonomous Surface Vehicle), ha sido un tema extensamente estudiado y tiene plena vigencia en el desarrollo de las investigaciones por parte de la comunidad científica.

La característica autónoma de una embarcación marina se la ofrece una fuente de energía y una unidad de cómputo a bordo, donde se ejecutan soluciones de software y control, que le proporcionan la posibilidad de cumplir tareas preprogramadas sin la intervención de un piloto humano [1]. Los *ASV* tienen gran variedad de sensores para monitorear las condiciones del medio ambiente marino. Estas herramientas, hacen de los *ASV* una opción ventajosa pues reducen los gastos de personal y combustible en comparación con las misiones de seguimiento de un barco tradicional y en consecuencia disminuyen el impacto de la contaminación ambiental [2].

Los primeros prototipos de *ASV* con una autonomía considerable fueron desarrollados en la década de los años 90 del siglo XX por el Instituto Tecnológico de Massachusetts (MIT); *ARTEMIS* se construyó en conjunto con las autoridades militares estadounidenses [3]. Precisamente el sector militar destacó como iniciador de estos prototipos, por ejemplo, durante las batallas marítimas de la Segunda Guerra Mundial se emplearon botes teleoperados para la detección del enemigo y la toma de muestras de agua radiactiva luego de las explosiones nucleares en Hiroshima y Nagasaki.

Hoy en día, los *ASV* han sido desarrollados e implementados por laboratorios pertenecientes a instituciones

académicas, corporaciones empresariales y centros gubernamentales [4]–[7]. Varios prototipos han sido construidos para el aprovechamiento de las fuentes renovables de energía como la solar, la eólica y la maremotriz. Un ejemplo es el SailDrone que funciona de forma autónoma propulsado por el viento [8]. De igual forma, el Instituto Superior Técnico de Lisboa posee las embarcaciones *DELFIN* y *CARABELA*, capaces de navegar autónomamente, implementando un sistema de control para el seguimiento de trayectorias basado en el conocimiento del modelo dinámico del vehículo. *CARABELA* tiene incorporado un radar como sensor para la evasión de obstáculos [9].

Otros ejemplos de estas embarcaciones pueden ser encontrados en la Universidad de Oporto en Portugal [10]–[12] y la Universidad de Ciencias y Tecnología de Noruega (NTNU) [2], [13]–[15].

Entre las aplicaciones de los *ASV* se encuentran: el estudio de la diversidad microbiana marina, la exploración y conservación del océano y sus recursos, el monitoreo, mantenimiento y protección de las comunicaciones, la navegación marítima y portuaria, entre otras. Verfuss [16] recomienda la utilización de estas embarcaciones autónomas en el monitoreo de especies marinas y aclara que el conteo de la densidad poblacional de las especies marinas y la evaluación de sus patrones espaciales debe hacerse siguiendo ciertos puntos en la superficie del mar. El mapeado del área bajo investigación puede determinarse en forma de polígonos conformado por líneas rectas lo cual conlleva a un menor gasto de energía. Es común que para estas misiones el vehículo navegue a bajas velocidades, inferiores a 2 m/s.

Para garantizar un seguimiento de estas trayectorias de manera autónoma se ha popularizado, entre los vehículos marinos, la ley de guiado Línea de Visión o Line of Sight (*LOS*), por sus siglas en inglés. *LOS* es una estrategia que se define como simple e intuitiva, y que imita el comportamiento de un timón que va corrigiendo el rumbo al seguir cierta trayectoria, evitando los movimientos de balanceo. Børhaug y otros autores [17], pertenecientes al NTNU, abordan la temática y presentan una ley de orientación *LOS* modificada con una acción integral para contrarrestar las perturbaciones ambientales sobre la embarcación objeto de estudio. El control garantiza el seguimiento de una ruta en línea recta ante la

presencia de dichas perturbaciones.

En los trabajos realizados por Caharija *et al* [15] [18], se efectúan varias simulaciones empleando este controlador tanto para vehículos en superficie como subacuáticos. Se demuestra que a pesar de la influencia de perturbaciones la acción integral logra mantener el curso de la embarcación.

En el trabajo publicado por Bibuli *et al* [19], se realiza una prueba experimental en simuladores de oleaje y brisa en superficie, con el objetivo de evaluar el controlador $I - LOS$ con una acción integral diseñado para el vehículo CART USSV. El comportamiento del CART USSV se analiza para diferentes valores de velocidad de propulsión y variando los parámetros de sintonización (distancia de seguimiento y ganancia de la acción integral). Los resultados obtenidos señalan que con ciertos valores de estos parámetros se logra una convergencia mucho más rápida hacia el punto deseado.

La presente investigación aborda el diseño de un controlador automático para el seguimiento de trayectorias en líneas rectas, empleando la ley de guiado (LOS) con la incorporación de una acción integral, para minimizar el efecto de las perturbaciones marinas sobre el vehículo. Además, se toman medidas para atenuar el efecto *wind-up*¹ causado por la incorporación de la acción integral.

II. SISTEMA DE CONTROL DE MOVIMIENTO

Para la ejecución de las misiones de manera autónoma, se requiere de un Sistema de Control de Movimiento (SCM), constituido a su vez por tres sistemas que interactúan entre sí a través de la transmisión de datos y señales [13].

a) El sistema de guiado provee una referencia con el cálculo de la posición y orientación deseadas. Estas coordenadas se calculan continuamente, con base en la información dada por el sistema de navegación GPS, utilizado generalmente en los vehículos de superficie.

b) El rumbo deseado alimenta al sistema de control, que es el encargado de manipular los actuadores, para posicionar la embarcación en las coordenadas pretendidas. Por ejemplo, seguimiento de trayectoria, de ruta o control de velocidad [20].

c) La trayectoria a seguir implementada en el subsistema de planificación de la ruta puede ser escogida atendiendo a diferentes criterios:

- Misión: El vehículo parte desde una posición inicial (x_0, y_0, z_0) y hasta una posición final (x_f, y_f, z_f) pasando por puntos intermedios.
- Información del medioambiente: La información del comportamiento del viento, las olas y las corrientes marinas puede ser utilizada para un uso óptimo de la energía.
- Datos geográficos: Los datos sobre la profundidad del mar y la presencia de accidentes geográficos son tenidos en cuenta para navegar en condiciones de seguridad.

¹Este efecto es característico de los controladores que presentan acción integral, suele producirse cuando en un sistema de control con un amplio rango de operación, la variable de control alcanza los límites prefijados del actuador; cuando esto sucede, la salida del lazo de control permanece en su límite independientemente de la salida del proceso.

- Evitar colisiones: Conocer la presencia de obstáculos, como otros vehículos marinos o construcciones flotantes, cercanos a la ruta planificada.
- Viabilidad: La configuración de los puntos sobre el plano debe efectuarse en correspondencia con la maniobrabilidad del vehículo y sin exceder su velocidad máxima.

Si la trayectoria a seguir está conformada por n segmentos en línea recta conectados por $n + 1$ puntos del camino, es necesario emplear una estrategia para conocer cuándo un punto (*waypoint* WP) ha sido alcanzado por el vehículo. Cuando el *ASV* está dentro de los límites de ese punto, entonces se cambia el objetivo de la ley de guiado hacia el siguiente punto del camino. Fossen [13], propone una vía para hacerlo, la cual está asociada a un círculo de aceptación que se establece con cada WP, cuyo radio será de valor $R_{k+1} > 0$ para el punto del camino $k + 1$. Por lo tanto, el criterio para que el algoritmo cambie de dirección hacia el siguiente WP queda establecido de la siguiente manera:

$$(x_{k+1} - x(t))^2 + (y_{k+1} - y(t))^2 \leq R_{k+1}^2 \quad (1)$$

donde x y y son las coordenadas de los puntos del camino establecido.

Como se puede observar, para que se produzca el cambio de un punto a otro, la posición del vehículo debe estar dentro del círculo de aceptación, teniéndose que definir un valor de $r \geq R_{k+1}$. El valor del radio es escogido de acuerdo con criterios de exactitud y maniobrabilidad de la embarcación [21]. Sin embargo, es conocido que en los *ASV* de inercia considerable se manifiesta un sobre impulso cuando se cambia de dirección al arribar al WP. Estos factores influyen en la precisión con la cual se logra seguir la trayectoria previamente definida.

III. ASV-CHEEMKIN

El *ASV - Cheemkin*, construido por el Laboratorio de Sistemas Híbridos del Centro de Investigación Científica de Yucatán (CICY) y objeto de estudio de esta investigación, tiene una configuración tipo catamarán multicasco. La estructura seleccionada permite la fácil instalación de las diferentes fuentes de recolección de energía renovables, destacando la energía solar por medio de dos paneles ubicados en la parte superior del vehículo. Los materiales de fabricación utilizados para la embarcación tipo catamarán fueron fibra de vidrio Advantex con ligante soluble de estireno tratada con Poliéster PP-250 y catalizador Metil-etil-cetona 10 %. El puente cuenta con un alma de aluminio de 6 mm de espesor para montaje de instrumentación e instalación de paneles solares. El prototipo del *ASV - Cheemkin* se muestra en la Figura 1.

La Tabla I muestra las características principales del *ASV - Cheemkin*.

La energía proporcionada por los paneles es almacenada en baterías que funcionan como fuente de alimentación para los motores eléctricos, sensores y otros componentes electrónicos instalados a bordo. El elemento central de todo el sistema de control es una mini computadora que

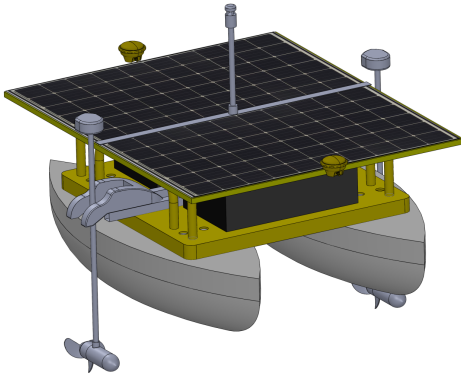


Fig. 1. Modelado en CAD del ASV – Cheemkin.

 TABLA I
 PRINCIPALES DIMENSIONES DEL ASV – Cheemkin

Parámetros	Valor	Unidades
Longitud Máxima	2.250	m
Altura	1.700	m
Altura casco	0.430	m
Dragado	0.1	m
Área sección media	0.103	m ²
Desplazamiento	630	kg

recibe las señales, mediante puertos seriales, del sistema de posicionamiento global (GPS-Trimble) para determinar la posición del vehículo; de la unidad de movimiento inercial que está integrada por: acelerómetro, giroscopio y magnetómetro; del sensor de viento ultrasónico (Airmar) que mide la velocidad y dirección aparente del viento, temperatura y humedad del aire, todo por medio de cuatro transductores; y de un microcontrolador que varía la velocidad de las propelas y actuadores como servomotores para cambiar la dirección del vehículo. Las propelas tienen controladores para prevenir la sobre corriente y evitar que se dañen por bloqueo. Otra de las funcionalidades de la mini computadora a bordo es el envío de la información vía inalámbrica a la estación remota en tierra.

IV. MODELADO DEL VEHÍCULO MARINO

El modelado de un vehículo marino involucra un estudio profundo del mismo, tanto en régimen estático como dinámico. Las características estáticas corresponden al estado de equilibrio de los cuerpos cuando se encuentran en reposo, mientras que las dinámicas abarcan el comportamiento de los cuerpos al moverse de manera acelerada [13], [22]. Para describir la trayectoria de un ASV es conveniente introducir dos sistemas de referencia: uno inercial con respecto a la tierra U y otro con respecto al propio vehículo. Figura 2.

Sistema de referencia con respecto a tierra U :

- El plano $x_u y_u$ coincide con el plano de superficie del agua.
- El eje z_u se ubica en el sentido del campo gravitacional local.

Sistema de referencia con respecto al vehículo C :

- El entorno matemático O_c coincide con el centro de gravedad del catamarán.
- El plano xy es paralelo al catamarán.
- El entorno matemático z apunta en el sentido del campo gravitacional local.

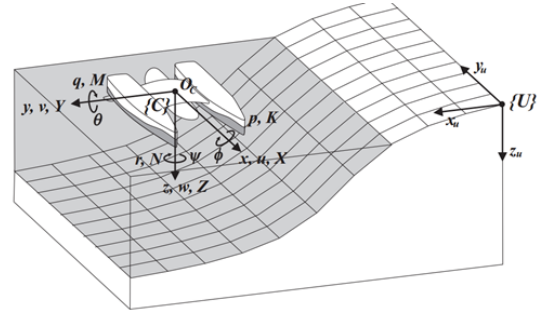


Fig. 2. Sistema de Coordenadas.

La embarcación se supone como un cuerpo rígido que se mueve en un medio líquido, y teniendo como premisa que el ASV se desplaza solo en el plano horizontal xy , se puede representar su modelo en un sistema de tres grados de libertad (3 GDL). Para ello la segunda ley de Newton puede plantearse para un vehículo marino con sistema de coordenada fijo al cuerpo, a partir de la siguiente ecuación:

$$M_c \dot{\nu} + C_c(\nu)\nu + D(\nu)\nu + g(\eta) = \tau_c + \tau_d \quad (2)$$

donde

$$M_c = M_{RB} + M_A \quad (3)$$

y

$$C_c \dot{\nu} = C_{RB} \dot{\nu} + C_A \dot{\nu} \quad (4)$$

además la matriz de transformación entre el vehículo y eje de coordenadas respecto a tierra es:

$$\dot{\eta} = J(\eta)\nu \quad (5)$$

- M_c =Matriz de inercia (incluye las masas añadidas M_A)
- $C_c(\nu)$ = Matriz de Coriolis y de términos centrípedos (incluye las masas añadidas $C_A(\nu)$)
- M_{RB} = Matriz del cuerpo rígido
- $C_{RB}(\nu)$ = Matriz del cuerpo rígido de coriolis
- τ_c =Fuerzas de propulsión y momentos
- τ_d = Fuerzas externas (incluye factores medioambientales como las corrientes, las olas y el viento)
- $D(\nu) = D^T(\nu) > 0$ Matriz de fuerzas hidrodinámicas
- $g(\eta)$ = es el vector de fuerzas generalizadas debido a la gravedad y la flotabilidad. Las fuerzas gravitacionales actúan en el centro de gravedad del vehículo de superficie

(O_c). Estas fuerzas son usualmente asociadas a una estabilidad metacéntrica. Se entiende por metacentro consiste en el punto teórico originado por la intersección de una línea vertical imaginaria que corta al centro de flotabilidad con otra línea imaginaria que atraviesa al nuevo centro de flotabilidad; el cual aparece cuando el vehículo es desplazado o inclinado.

Los vectores de posición y velocidad están dados por:

$$\eta = [x, y, \psi]^T \in \mathbb{R}^2 \times \mathcal{S} \quad (6)$$

$$\nu = [u, v, r]^T \in \mathbb{R}^3 \quad (7)$$

De acuerdo con 2 y 5, la matriz M_{RB} y la matriz de masas añadidas M_A quedan definidas como:

$$M_{RB} = \begin{bmatrix} 530 & 0 & 0 \\ 0 & 530 & 0 \\ 0 & 0 & 523.4 \end{bmatrix} \quad (8)$$

$$M_A = \begin{bmatrix} -23.7 & 0 & 0 \\ 0 & -702.09 & -30.01 \\ 0 & -30.01 & -409.3 \end{bmatrix} \quad (9)$$

Los elementos en M_A son parámetros hidrodinámicos: $X_{\dot{u}} = -23.7kg$, $Y_{\dot{v}} = -702.09kg$, $N_{\dot{r}} = -409.3kgm^2$, $Y_{\dot{r}} = N_{\dot{v}} = -30.01kgm$. Estos valores se obtuvieron basados en el desarrollo de un vehículo similar construido por el Instituto Superior Técnico de Lisboa [23]. Tal como recomienda la literatura se pueden utilizar métodos semi empíricos validados posteriormente con análisis de CFD del *ASV - Cheemkin* [24].

Las matrices que representan la fuerza de Coriolis y los efectos de la aceleración centrípeta quedan definidas en el *ASV* objeto de estudio de la presente investigación como: C_{RB} , que se relaciona con el peso y la velocidad angular y C_A que está determinada por los componentes hidrodinámicos mostrados en M_A , pero en este caso C_A es una matriz simétrica oblicua que representa el producto cruzado entre la aceleración y la interacción de masas añadidas.

$$C_{RB} = \begin{bmatrix} 0 & -530r & 0 \\ 530r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (10)$$

$$C_A = \begin{bmatrix} 0 & 0 & -702.09v - 30.01r \\ 0 & 0 & -23.7u \\ 702.09v + 30.01r & 23.7u & 0 \end{bmatrix} \quad (11)$$

En la diagonal de la matriz D se encuentran los parámetros que representan la energía de disipación debido a la interacción del *ASV* con la masa de agua en la cual está inmerso.

$$D = \begin{bmatrix} -3.9u & 0 & 0 \\ 0 & -601.02v + 339|r| & 0 \\ 0 & 0 & 51.6v + 1903.1|r| \end{bmatrix} \quad (12)$$

Además de la dinámica del vehículo es importante tener en cuenta los efectos de factores ambientales, como olas,

corrientes marinas y el viento. Con el fin de simular estos efectos se incluyen perturbaciones en los componentes de velocidad y dirección tanto para el viento como para las olas y corrientes marinas. Por la complejidad de reflejar el compartamiento real de estas variables se opta por adjudicarles valores constantes que se mencionarán en la simulación.

V. CONTROLADOR I – LOS

Los escenarios para el control de movimiento se seleccionan a partir de los objetivos de control deseados y mantienen una estrecha relación con el sistema de guiado [14].

Para concretar estos escenarios, la literatura señala tres de las estrategias de guiado más utilizadas en los vehículos autónomos. Estas son:

- Persecución (PP. Pure Pursuit)
- Dirección Constante (CB. Constant Bearing)
- Línea de Visión (LOS. Line-of-Sight)

La línea de visión toma en cuenta los requerimientos espaciales y se acompaña de una ley de dirección basada en la distancia (*lookahead*, Δ), definida por la distancia que marca la proyección de la posición del vehículo y el vector *LOS*. Las principales variables del algoritmo de guiado se pueden apreciar en la Figura 3, donde:

- P_k y P_{k+1} es el punto inicial y final de la trayectoria a seguir
- r es el vector *LOS* que se origina en el vehículo (p) y finaliza en el punto P_{int}
- Δ distancia a la que está situado el punto P_{int} de la proyección de la posición del vehículo sobre el camino
- s error de seguimiento a lo largo del camino
- e error de seguimiento perpendicular al camino
- α_k ángulo de rotación del camino respecto al sistema de referencia en tierra
- χ_r ángulo de curso deseado del vehículo

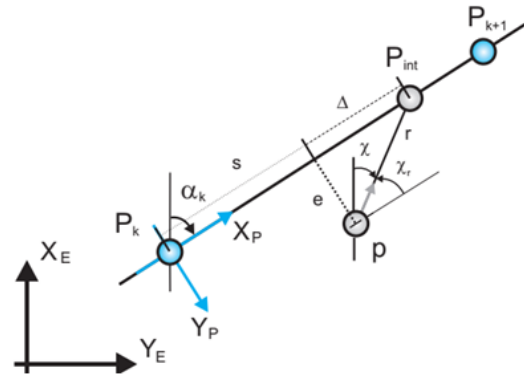


Fig. 3. Principales variables del algoritmo de guiado basado en la distancia de la proyección del vehículo sobre el camino.

Esta ley propone determinar la dirección que debe seguir el vehículo a partir de la suma de dos ángulos:

$$\chi(e) = \chi_r(e) + \chi_p \quad (13)$$

donde:

$$\chi_p = \alpha_k \quad (14)$$

siendo χ_p el ángulo tangente al camino y $\chi_r(e)$ un ángulo de corrección, el cual asegura que la velocidad del vehículo esté en dirección al punto del camino hacia el cual la embarcación se dirige.

El propósito del algoritmo es que el error perpendicular al camino (e) sea cero cuando el tiempo tienda a infinito [13], [14].

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (15)$$

$$\chi_r(e) = \arctan\left(\frac{-e(t)}{\Delta}\right) \quad (16)$$

donde

$$\Delta = \frac{1}{k_p} \quad (17)$$

De la ecuación (16) se puede incluir el efecto integral para garantizar que el error se a cero en estado estable:

$$\chi_r(e) = -\arctan(k_p e(t) + k_i y_{int}) \quad (18)$$

donde k_i es la ganancia integral y k_p la proporcional. Por tanto y_{int} queda definida por:

$$\dot{y}_{int} = \frac{e\Delta}{\Delta^2 + (e + k_i y_{int})^2} \quad (19)$$

Todo lo anterior es utilizado cuando el algoritmo de control se emplea con la variable Δ constante; para la utilización de una ganancia proporcional en dependencia del cambio del error perpendicular al camino, se emplea la ecuación (20), correspondiente a Δ variable. Durante el proceso de simulación no es posible conmutar entre ambos algoritmos. A lo largo de las simulaciones se emplean ambos algoritmos a modo de comparación.

$$\Delta(e) = (\Delta_{m\acute{a}x} - \Delta_{m\acute{i}n}) e^{-K_\Delta(e)^2} + \Delta_{m\acute{i}n} \quad (20)$$

Como se observa en la configuración de la ley de guiado es importante el parámetro Δ . A partir del valor de Δ seleccionado, se calcula la ganancia proporcional k_p , ya que ambos términos son inversamente proporcionales. Todo lo anterior implica que la ley de dirección basada en la distancia *lookahead* sea equivalente a una ley de control proporcional [14]. Los valores máximos ($\Delta_{m\acute{a}x}$) y mínimos ($\Delta_{m\acute{i}n}$), entre los cuales varía el parámetro Δ , son definidos por el usuario y deben depender de las características de maniobrabilidad (largo) de la embarcación. Por lo tanto, se recomienda ajustar desde 2 a n veces, de la longitud. El restante parámetro de diseño K_Δ , se indica seleccionar siempre mayor que cero. Su incidencia en la ecuación es de actuar como un peso de la magnitud del error, provocando que exista prevalencia por los valores límites de Δ .

El diseño del controlador mostrado en (19) tiene la propiedad de que cuando $y \rightarrow \infty$, $\dot{y}_{int} \rightarrow 0$. Esto significa que la tasa de integración disminuirá con errores grandes de la variable y . Es decir, el término integral será menos dominante cuando el error perpendicular al camino sea cada vez más grande [15]. Todo ello garantiza contrarrestar el efecto *wind-up*, característico de los controladores que tienen una acción integral, y generalmente, ocurre cuando en un sistema de control con un amplio rango de operación, la variable de control alcanza los límites predeterminados del actuador. Cuando esto sucede, la salida del bucle de control permanece en su límite independientemente de la salida del proceso.

VI. RESULTADOS EN SIMULACIÓN

Todas las pruebas y simulaciones realizadas en este trabajo se desarrollan en el software *Matlab – Simulink*. Para tal efecto se construye un archivo *ILOS – ASV – Cheemkin.slx*, capaz de recrear de forma virtual todo el sistema de control de movimiento. Además se emplea el toolbox de libre acceso *Marine Systems Simulator (MSS)* [25].

El principal criterio para evaluar el desempeño del controlador diseñado fue el error perpendicular al camino trazado.

$$Error = \frac{\sum_{i=0}^{i=T} e(t)^2}{T} \quad (21)$$

Además se evalúa la acción de mando hacia los propulsores eléctricos del vehículo teniendo en cuenta que sus patrones de consumo energético afectan la autonomía de la misión.

$$Mando = \frac{\sum_{i=0}^{i=T} \psi(t)^2}{T} \quad (22)$$

La literatura consultada recomienda el empleo del parámetro *lookahead* variable tal como muestra la ecuación (20), para los tramos del camino donde una maniobra más complicada es exigida, como en los giros de 90° que tiene la trayectoria. Con ese fin se procedió a implementar esta ecuación que varía el valor de la ganancia k_p en función de una realimentación del error.

TABLA II
PUNTOS DE LA TRAYECTORIA

x (m)	0	50	50	200	300	50
y (m)	0	0	250	250	0	-150

Las simulaciones se realizaron bajo las siguientes condiciones ambientales: la velocidad de las corrientes marinas se estimó en $V_c = 0.1$ m/s y su ángulo de dirección $\beta_c = 30^\circ$, el oleaje se establece utilizando el espectro *Joint North Sea Wave Project (JONSWAP)*, para un valor de frecuencia fundamental $w_0 = 6$ rad/s y con un $\sigma = 0.5$. Por el momento, no se tiene configurada la acción del viento dada la complejidad de modelación e incertidumbre en la incidencia de su comportamiento sobre el vehículo marino.

La velocidad de giro del motor del *ASV – Cheemkin* es de 600 rpm y el período de muestreo se fija en $T = 0.01$

segundos. Se considera además que el radio para el círculo de aceptación es de 20 metros. A partir de este valor, y utilizando la ecuación (1), se establece cuando un punto de la trayectoria está alcanzado.

Con estos supuestos se realizan las primeras simulaciones para sintonizar el controlador, cuyos dos parámetros fundamentales son k_i y k_p . Este último es el inverso de Δ .

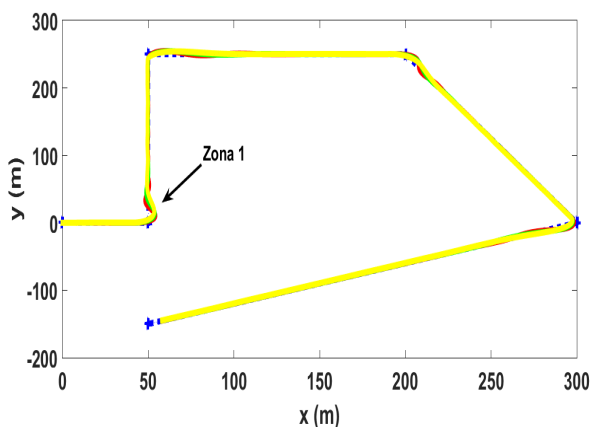


Fig. 4. Controlador I-LOS. Seguimiento de trayectoria para 3 diferentes valores de Δ .

La Figura 4 muestra el seguimiento de la trayectoria definida por los puntos representados en la Tabla II. Aquí el parámetro Δ se evalúa en 5 m (línea roja), 9 m (línea verde) y 15 m (línea amarilla). La ganancia integral k_i se selecciona, en un primer momento igual a 0.1.

De modo general, todas siguen la trayectoria deseada marcada en azul, sin embargo, en los puntos de giro brusco como el de la Zona 1 existen diferencias en el comportamiento del sobre impulso cuando alcanza el primer punto (50;0). Más detalle de este particular se puede ver en la Figura 5.

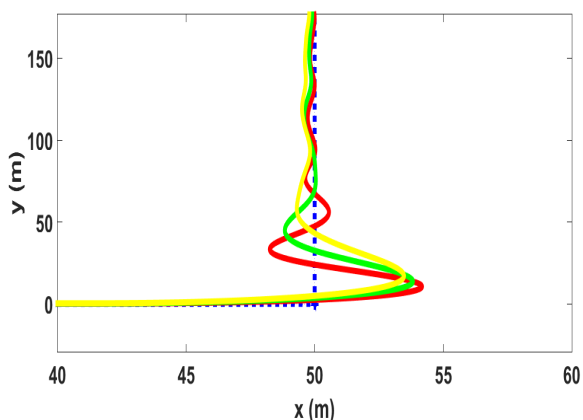


Fig. 5. Acercamiento sobre la Zona 1.

En la Zona 1, el valor de $\Delta = 15m$ tiene un mejor comportamiento pero a lo largo de la trayectoria $\Delta = 5m$

ofrece menor valor de error perpendicular al camino por ligera diferencia.

TABLA III
COMPARACIÓN DEL ERROR Y LA DESVIACIÓN ESTÁNDAR PARA 3 VALORES DIFERENTES DE Δ

	$\Delta = 15m$	$\Delta = 9m$	$\Delta = 5m$
Error (m)	2.17	2.08	2.06
σ_{error}	1.4733	1.448	1.4353

Seguidamente se procedió a variar la ganancia integral mantenido en $k_p = 0.2$, pues fue la que mejor resultados arrojó. Al realizar la comparación entre valores de $k_i = 0.1$, $k_i = 0.5$ y $k_i = 0.05$ se obtienen los siguientes resultados:

TABLA IV
COMPARACIÓN DEL ERROR Y LA DESVIACIÓN ESTÁNDAR PARA 3 VALORES DIFERENTES DE k_i .

	$k_i = 0.1$	$k_i = 0.5$	$k_i = 0.05$
Error (m)	2.06	2.052	2.55
σ_{error}	1.4353	1.4326	1.4330

Como puede apreciarse la mejor opción es elevar la ganancia integral a 0.5, sin que se incurra en un comportamiento oscilatorio durante la trayectoria. Con este valor y $k_p = 0.2$ se obtiene la Figura 6, que muestra el comportamiento del error a lo largo del tiempo de simulación. Los picos abruptos en el error alrededor de los 2, 11, 17 y 27 segundos, coinciden con la transición de un tramo del camino hacia el siguiente y la consecuente discontinuidad matemática.

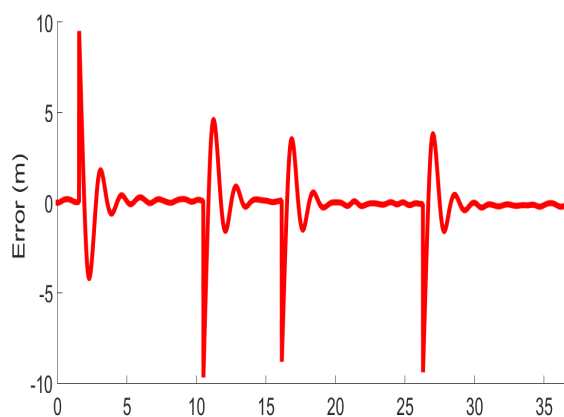


Fig. 6. Comportamiento del error con el controlador I-LOS. $k_p = 0.2$ y $k_i = 0.5$.

Finalmente, se prueba el empleo del algoritmo con Δ variable para comprobar si efectivamente se produce un mejor comportamiento en los tramos de la trayectoria donde se requiere cambios de rumbo en el vehículo. Aquí es preciso advertir que las características dinámicas del ASV – *Cheemkin*, al ser una embarcación pesada, son lentas. Por lo tanto, le toma más tiempo converger al rumbo deseado, existiendo un desfase entre el ángulo de rumbo calculado por el controlador y el ángulo real que experimenta el ASV.

Para la simulación se tiene en cuenta la literatura consultada que recomienda asignar valores de $6L$ (L longitud del vehículo) a $\Delta_{m\acute{a}x}$ y $2L$ a $\Delta_{m\acute{i}n}$. Tal como refleja la ecuación (20). El restante parámetro de diseño K_{Δ} se indica seleccionar siempre mayor que cero. El propósito de K_{Δ} es actuar como un peso de la magnitud del error provocando que exista prevalencia por los valores límites de Δ .

En general, la respuesta del vehículo es muy similar con ambos algoritmos, por lo que se omite la comparación. Sin embargo, para el caso de la señal de mando que se le exige a los actuadores (motores eléctricos) se obtiene la Figura 7. La línea azul corresponde a $4 \leq \Delta \leq 6$ mientras que la línea roja es la señal de mando correspondiente a $\Delta = 5$: además los datos comparativos de la señal y su desviación estándar se pueden observar en la Tabla V.

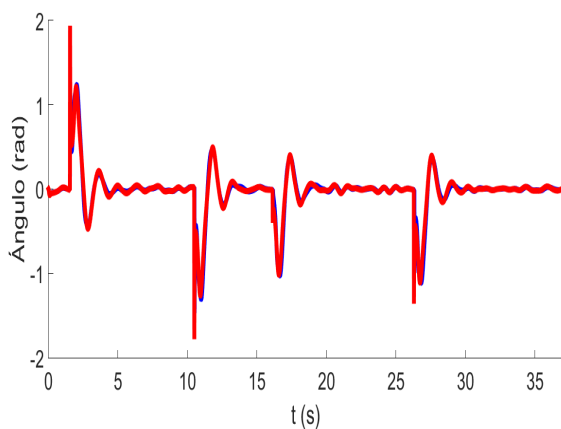


Fig. 7. Comportamiento del error con el controlador $I - LOS$. $k_p = 0.2$ y $k_i = 0.5$.

TABLA V
SEÑAL DE MANDO Y DESVIACIÓN ESTÁNDAR.

	Δ variable	Δ constante
Mando (rad)	0.0733	0.0689
σ_{mando}	0.2696	0.2614

Por lo tanto, se puede afirmar que la mejor opción para el controlador $I - LOS$ es emplear $\Delta = 5$ pues la otra variante propuesta provoca un aumento de la señal de mando a los motores eléctricos (mayor consumo); además con la elección se evita un mayor gasto de recursos de cómputo que exigen la programación y ejecución del algoritmo iterativo Δ variable.

VII. CONCLUSIONES

En esta investigación se diseñó un controlador con efecto integral basado en el algoritmo de guiado LOS para el vehículo marino *ASV-Cheemkin*. Los resultados obtenidos mediante simulación demostraron que la embarcación puede seguir la trayectoria planificada de forma satisfactoria. Tal circunstancia está en consonancia con la teoría desarrollada por otros autores.

El controlador $I - LOS$ podría programarse en el hardware que se encuentra disponible en la embarcación, sin embargo, es recomendable que futuros trabajos realicen una comparación con otros métodos de control. De igual forma se debe incluir un modelo mejorado de las perturbaciones marinas y del viento.

VIII. AGRADECIMIENTOS

Los autores desean agradecer al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo financiero en el marco del proyecto 2015-01-786 (Convocatoria Problemas Nacionales). De igual forma, se agradece al ingeniero Juan Enrique Méndez Magaña miembro del grupo de trabajo del Centro de Investigación Científica de Yucatán.

REFERENCIAS

- [1] J. M. Cruz García, J. Aranda, and J. M. Girón, "Automática marina: Una revisión desde el punto de vista del control," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 9, no. 3, pp. 205–218, 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.riai.2012.05.001>
- [2] M. Breivik, "Topics in Guided Motion Control of Marine Vehicles," Ph.D. dissertation, NTNU, 2010.
- [3] J. Manley, "Development of the autonomous surface craft ACES," in *Oceans '97. MTS/IEEE Conference Proceedings*, vol. 2. IEEE, 1997, pp. 827–832. [Online]. Available: <http://ieeexplore.ieee.org/document/624102/>
- [4] M. Caccia, "Autonomous surface craft: Prototypes and basic research issues," *14th Mediterranean Conference on Control and Automation, MED'06*, 2006.
- [5] G. Papadopoulos, "Underwater Vehicle Localization Using Range Measurements," Ph.D. dissertation, Massachusetts Institute of Technology, 2010.
- [6] H. Niu, Y. Lu, A. Savvaris, and A. Tsourdos, "Efficient Path Planning Algorithms for Unmanned Surface Vehicle," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 121–126, 2016.
- [7] D. Moreno-Salinas, N. Crasta, M. Ribeiro, B. Bayat, A. M. Pascoal, and J. Aranda, "Integrated Motion Planning, Control, and Estimation for Range-Based Marine Vehicle Positioning and Target Localization," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 34–40, 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.ifacol.2016.10.318>
- [8] C. Meinig, N. Lawrence-Slavas, R. Jenkins, and H. M. Tabisola, "The use of Saildrones to examine spring conditions in the Bering Sea: Vehicle specification and mission performance," *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–6, 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7404348/>
- [9] J. Alves, R. Oliveira, and A. Pascoal, "Vehicle and Mission Control of the DELFIM Autonomous Surface Craft," *Canadian Journal of Remote Sensing*, vol. 16, no. 1, pp. 2–13, 2006.
- [10] J. P. Martins Pires, "Guidance of an Autonomous Surface Vehicle for Underwater Navigation Aid," Ph.D. dissertation, Universidad de Ingeniería de Oporto, Portugal, 2018.
- [11] N. A. Cruz, "Adaptive Ocean Sampling with Modular Robotic Platforms," Ph.D. dissertation, Universidad de Ingeniería de Oporto, Portugal, 2016.
- [12] J. Valente, "Real-Time Passive Acoustic Tracking of Underwater Vehicles," Ph.D. dissertation, Universidad de Ingeniería de Oporto, Portugal, 2016. [Online]. Available: <https://repositorio-aberto.up.pt/handle/10216/85105>
- [13] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*, first edit ed. Trondheim, Norway: John Wiley & Sons, 2011. [Online]. Available: www.wiley.com.
- [14] A. M. Lekkas, "Guidance and Path-Planning Systems for Autonomous Vehicles," Ph.D. dissertation, NTNU, 2014.
- [15] W. Caharija, K. Y. Pettersen, M. Bibuli, P. Calado, E. Zereik, J. Braga, J. T. Gravdahl, A. J. Sorensen, M. Milovanovic, and G. Bruzzone, "Integral Line-of-Sight Guidance and Control of Underactuated Marine Vehicles: Theory, Simulations, and Experiments," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1623–1642, 2016.
- [16] U. K. Verfuss, A. S. Aniceto, D. V. Harris, D. Gillespie, S. Fielding, G. Jiménez, P. Johnston, R. R. Sinclair, A. Sivertsen, S. A. Solbø, R. Storvold, M. Biuw, and R. Wyatt, "A review of unmanned vehicles for the detection and monitoring of marine fauna," *Marine Pollution Bulletin*, vol. 140, no. December 2018, pp. 17–29, 2019. [Online]. Available: <https://doi.org/10.1016/j.marpolbul.2019.01.009>
- [17] E. Børhaug, A. Pavlov, and K. Y. Pettersen, "Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents," *Proceedings of the IEEE Conference on Decision and Control*, pp. 4984–4991, 2008.
- [18] W. Caharija, "Integral Line-of-Sight Guidance and Control of Underactuated Marine Vehicles," Ph.D. dissertation, NTNU, 2014. [Online]. Available: <http://www.diva-portal.org/smash/record.jsf?pid=diva2:761266&dswid=9190>
- [19] M. Bibuli, W. Caharija, K. Y. Pettersen, G. Bruzzone, M. Caccia, and E. Zereik, "LOS guidance - Experiments and tuning," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 19, pp. 4209–4214, 2014.
- [20] G. Conte, D. Scaradozzi, D. Mannoichi, P. Raspa, L. Panebianco, and L. Screpanti, "Experimental testing of a cooperative ASV-ROV multi-agent system," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 347–354, 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.ifacol.2016.10.428>
- [21] M. H. Majid and M. R. Arshad, "Control of autonomous surface vehicle for acoustic source tracking," *Proceedings - 2016 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2016*, no. October, pp. 101–106, 2017.
- [22] Fossen, "Nonlinear Modelling and Control of Underwater Vehicles," Ph.D. dissertation, NTNU, 1991.
- [23] J. Alves, P. Oliveira, R. Oliveira, A. Pascoal, M. Rufino, L. Sebastiao, and C. Silvestre, "Vehicle and mission control of the delfim autonomous surface craft," in *2006 14th Mediterranean Conference on Control and Automation*. IEEE, 2006, pp. 1–6.
- [24] J. M. De la Cruz García, J. A. Almansa, and J. M. Girón Sierra, "Automática marina: Una revisión desde el punto de vista del control," *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 9, no. 3, pp. 205–218, 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.riai.2012.05.001>
- [25] T. Fossen and T. Perez, "Marine Systems Simulator (MSS)," Noruega, 2009. [Online]. Available: <https://github.com/cybergalactic/MSS>

Parameter Estimation and Delay-Based Control of a Linear Ultrasonic Motor

Luis Luna, Kevin Lopez, Luis Cantera, Rubén Garrido, Cristóbal Vargas
Automatic Control Department
CINVESTAV-IPN

Email: jluna, klopez, lcantera, garrido, cvargas@ctrl.cinvestav.mx

Abstract—This paper presents the parameter estimation and position control of a Linear Ultrasonic Motors (LUM). A nominal linear model is used to perform parametric estimation using the standard Least Squares and the Least Squares Orthogonal Distances methods. Subsequently, the parameter estimates permits tuning a Cascade Proportional-Integral Retarded (CPIR) controller which is designed to carry out the positioning of a LUM. Experiments with a laboratory prototype show that this controller is able to produce a steady-state position error of 50 nm.

I. INTRODUCTION

Linear Ultrasonic Motors (LUMs) have attracted attention in the last few years in different fields of industry and research due to their characteristics such as a simple mechanical structure, relatively large forces or torques without gear mechanisms, insensitivity to electromagnetic fields and negligible electromagnetic emission [1]. The working principles of a LUM are based on the ability of piezoelectric materials to vibrate in the ultrasonic frequency band and in the inverse piezoelectric effect. The latter happens when applying a voltage on a piezoelectric crystal. Under this condition a deformation takes place transforming electrical energy into mechanical movement [2]. This principle produces movement in a LUM through the frictional contact force between its fixed and mobile parts.

A set of commercially available LUMs are the so-called micro-thrust devices, manufactured by Nanomotion Ltd. In this case, the elliptical movements of the surface contact points between the fixed and mobile parts of the LUM are generated by the superposition of longitudinal and bending oscillations modes. These microscopic movements are transformed into a macroscopic linear movement through frictional contact forces. The positioning accuracy of the micro-thrust motors is mainly influenced by the measuring system and the

control strategy and, of course, it is limited by the sliding effects due to the frictional contact forces [3].

It is worth noting that mechanical friction has been identified as one of the main challenges faced when controlling a LUM. On the other hand, the mathematical model for this kind of motor depends on different models of friction as well as the wave equation model that describes the movement of the piezoelectric crystals which correspond to a fourth-order partial differential equation [1]. It is worth noting that some mathematical models describing the dynamics of LUMs reported in the literature are rather complex since the friction phenomena and electromechanical hysteresis present in this type of motors are difficult to model [4] and the parameters are time-varying due to changes in temperature and operating conditions. Some of the control strategies applied to the LUM reported in the literature are the Proportional Integral Derivative (PID) Control law with fixed and variable gains[5], adaptive control techniques [6], and sliding modes control [7].

Nowadays, delay-based controllers have been proposed in the literature for position and velocity control of servo systems including the Proportional Retarded (PR) [8] and the Proportional-Integral-Retarded (PIR) [9] algorithms. Recently, a Cascade Proportional Integral Retarded (CPIR) algorithm has been proposed [10]. It corresponds to the time-delay version of the Cascade Proportional-Integral (P-PI) algorithm widely used in practice for controlling servomotors [11]. In the P-PI controller, a proportional controller closes an outer position loop, and an inner PI controller regulates a velocity loop. Interestingly enough, delay-based controllers exhibit measurement noise filtering capabilities thus making unnecessary the use of filters as it has been shown in [9] for the Integral retarded (IR) controller applied to the velocity control of servodrives. As far as the authors know, these types of controllers have not

been previously applied to ultrasonic motors, and due to the advantages they present, they are a good option to carry out nano-positioning tasks.

In order to obtain a model adequately describing the behavior of a plant, its parameters are adjusted to obtain the best fit to the experimental data through a parameter estimation algorithm. One of the most popular methods for parameter estimation of dynamical systems is the Least Squares (LS) algorithm, which consists in minimizing the sum of the squares of the errors between the experimental data and the data generated through the model [12], [13]. On the other hand, the Least Squares of Orthogonal Distances (LSOD) method takes into account errors in the dependent and the independent variables [14], [15]. The main advantage of the LSOD algorithm over the LS method is that the former avoids computing the pseudo inverse of a matrix, which may be ill-conditioned.

This work presents the nano-positioning control of a LUM using a Cascade Proportional-Integral Retarded (CPIR) controller. The procedure for tuning of this controller is straightforward and it is based on a simple root finding method. On the other hand, the parameters of a dynamic model of the LUM provided by the manufacturer are estimated through the LS and LSOD algorithms. The structure of this work is as follows: Section 2 presents the mathematical model of the LUM. In Section 3 the parameter estimation of a LUM model are carried out using the LS and LSOD methods. In Section 4, the CPIR control law is introduced as well as its tuning methodology, and Section 5 describes the corresponding experimental results. Finally, Section 6 mentions some concluding remarks.

II. LINEAR ULTRASONIC MOTOR MODEL

The linear ultrasonic motor (LUM) considered in this work is the HR4-K-S3 motor driven by an AB5 amplifier, both produced by the Nanomotion company. The motor has a rectilinear optical position encoder with a resolution of 50nm that allows measuring its displacement. When this motor works with the Nanomotion AB5 amplifier, their dynamics are described as follows [16]

$$m\ddot{q}(t) + K_{fv}\dot{q}(t) = K_f(v_{in}(t) - v_{offset}(t)) + \varphi(t), \quad (1)$$

where m is total mass of the moving part of the motor including the load, K_{fv} is a damping factor, $\ddot{q}(t)$ is the motor acceleration, $\dot{q}(t)$ is the motor velocity, K_f is a gain, $v_{offset}(t)$ is the voltage applied to compensate for a parasitic voltage generated by the AB5 amplifier, $\varphi(t)$

represents internal and external constant perturbations, and $v_{in}(t)$ is the input voltage defined as follows

$$v_{in}(t) = u(t) + \hat{v}_{offset}(t), \quad (2)$$

where $u(t)$ is the control signal and $\hat{v}_{offset}(t)$ is an estimate of $v_{offset}(t)$. Model (1) is a modification of the model provided by Nanomotion. Substituting (2) into (1) yields

$$m\ddot{q}(t) + K_{fv}\dot{q}(t) = K_f u(t) + K_f(\hat{v}_{offset}(t) - v_{offset}(t)) + \varphi(t), \quad (3)$$

Defining $\bar{d}(t) := K_f(\hat{v}_{offset} - v_{offset}) + \varphi(t)$, allows rewriting (3) as

$$m\ddot{q}(t) + K_{fv}\dot{q}(t) = K_f u(t) + \bar{d}(t),$$

Dividing by m and defining $a := \frac{K_{fv}}{m}$, $b := \frac{K_f}{m}$ and $d = \frac{\bar{d}(t)}{m}$ finally yields

$$\ddot{q}(t) + a\dot{q}(t) = bu(t) + d. \quad (4)$$

The above equation (4) describes the behavior of the LUM with unknown parameters a and b and disturbance term d .

III. PARAMETER ESTIMATION

The LSOD method consists in minimizing the sum of the squares of the orthogonal distances $\|\mathbf{d}_i\|$ from the experimental data Q_i to an approximation plane as it is shown in Figure 1.

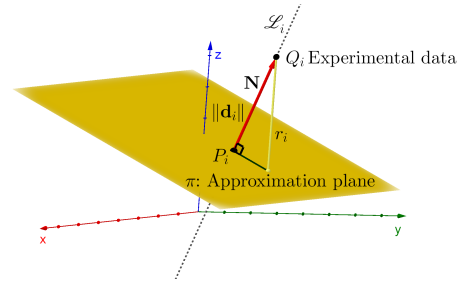


Figure 1. Approximation plane.

In Figure 1, the approximation plane in \mathbb{R}^3 is defined as

$$\pi = \{(x, y, z) : Ax + By + Cz + D = 0\}, \quad (5)$$

with normal vector to the plane $\mathbf{N} = [A \ B \ C]^\top$. Given \mathbf{N} and the experimental data $Q_i = (x_i, y_i, z_i)$ with $i = 1, 2, \dots, m$. The orthogonal distance $\|\mathbf{d}_i\|$ is defined as follows, where $\|\cdot\|$ denote the Euclidean norm of a vector.

Definition 1. The orthogonal distance $\|\mathbf{d}_i\|$ from the point Q_i to the plane π is the distance from Q_i to the intersection point P_i between the plane π and the straight line \mathcal{L}_i , which is defined along \mathbf{N} and passes through Q_i .

On the other hand, LS method consists in minimizing the sum of the squares of the errors from experimental data to an approximation curve using only the dependent variable (see Fig. 1). To clarify this point, let an experimental data set (x_i, y_i, z_i) with $i = 1, 2, \dots, m$ and the approximation plane $f(x, y)$. The errors from the dependent variable z_i in the experimental data to the approximation plane are $r_i = z_i - f(x_i, y_i)$, which is different from $\|\mathbf{d}_i\|$.

A. Parameter Estimation of a LUM

To stabilize the linear ultrasonic motor in closed loop without prior knowledge on its parameters, a Proportional-Derivative (PD) control algorithm is used for this purpose and it is given by

$$u = K_p e - K_d \dot{q}_e, \quad (6)$$

where e denotes the position error defined as $e := r - q$, r is the reference, q the position measured by the optical position encoder and \dot{q}_e an estimate of the time derivative \dot{q} . The following transfer function composed of a high-pass filter in cascade with a low-pass filter allows obtaining estimates of the velocity \dot{q}_e from the position measurements q .

$$G(s) = \left[\frac{450s}{s+450} \right] \left[\frac{400}{s+400} \right] \quad (7)$$

The equation (4) is written using the following change of coordinates $(x, y, z) = (-\dot{q}, u, \ddot{q})$ while neglecting the perturbation term, i.e.

$$\pi = \{(x, y, z) : ax + by + z = 0\}, \quad (8)$$

with normal vector to the plane $\mathbf{N} = [a \ b \ 1]^T$, where the experimental data will be acceleration, velocity and the control law signals. Note that acceleration and velocity signals are estimated through filtering of the position measurements. The functions to be minimize for the LS and LSOD method are [17]

$$f_{LS} := \sum_{i=1}^m r_i^2, \quad (9)$$

$$f_{LSOD} := \|\mathbf{d}\|^2 = \frac{1}{a^2 + b^2 + 1} \sum_{i=1}^m r_i^2. \quad (10)$$

B. Experimental setup

The laboratory prototype consists of a linear ultrasonic motor, model HR4-K-S3 powered by an AB5 amplifier, model AB5-TP HR E4 configured in servo mode, which operates in a range of ± 10 V, both from Nanomotion Ltd. An optical linear encoder Renishaw RGH24H30D30A with 50 nm resolution per pulse allows measuring the motor position. The control algorithms are implemented using the MatLab/Simulink programming platform under the QUARC real-time control environment and a QPID data acquisition card, the latter two from Quanser Consulting. The control signal produced by the QPID card passes through a galvanic isolation box before feeding the AB5 amplifier. Figure 2 shows the configuration used in the experiments. All the Simulink diagrams use a sampling period of 1 ms and the Euler-ode1 integration method.

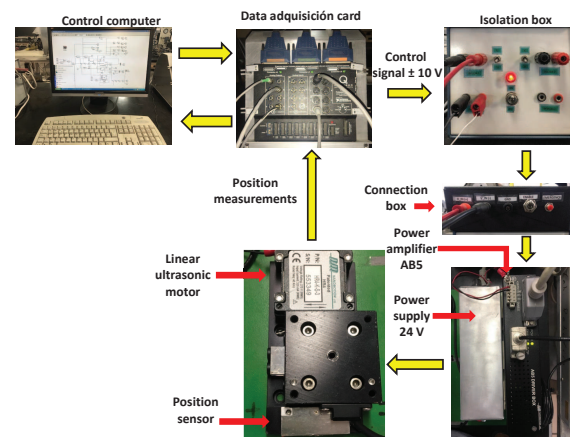


Figure 2. Experimental Setup

The parameter estimation obtained through the LS and the LSOD methods can be summarized in Table I together with the tuning of the PD controller. A filtered white noise signal is used as reference for identification purposes.

Table I
PARAMETRIC IDENTIFICATION RESULTS

Method	K_p	K_d	\hat{a}	\hat{b}
LS	4225	13	43.9652	0.9360
LSOD	4225	13	253.7334	5.1266

IV. DELAY CONTROLLER FOR POSITION CONTROL

Inspired by the P-PI controller, the delay-based algorithm called the CPIR controller, is composed of two

loops, an IR controller closing an inner velocity loop, and an outer loop corresponding to a Proportional (P) controller that regulates the angular position.

One advantage of the CPIR controller over the P-PI controller is the fact that the use of time delays in the former precludes velocity measurements, thus avoiding the shortcomings associated with finite-differences methods and high-pass filters. On the other hand, the CPIR controller have several advantages. Principal features include the functioning using only position measurements. The CPIR has two advantages inherited from the P-PI controller, namely, the fact that the inner velocity loop may work alone when the outer position loop is disengaged, and that it is possible to add a saturation function in the outer loop. This last feature has two advantages when a positioning task requires large position set points. First, the saturation function sets a prescribed value of the LUM angular velocity. Second, it also avoids overshoots which otherwise happen without introducing this function. This variant of the CPIR controller is called the Cascade Nonlinear Proportional Integral Retarded (CNPIR) controller.

The equation describing the Cascade Proportional Integral Retarded (CPIR) controller is

$$\dot{u}(t) = (K_i - K_{ir})K_p e(t) - K_{ir}\dot{q}(t-h) + K_i\dot{q}(t). \quad (11)$$

Consider the dynamics of a LUM (4), with $a > 0$, $b \neq 0$ and neglecting the perturbation term d . The position error is $e(t) = r - q$, where r is a desired constant position reference. The tuning rules for the Cascade Proportional Integral Retarded (CPIR) controller proposed in [10] are as follows

$$h = \frac{1}{\sigma_{int}}, \quad (12)$$

$$K_i = \frac{\sigma_{int}^2 + \frac{a^2}{4}}{b}, \quad (13)$$

$$K_{ir} = \frac{2\sigma_{int}^2 e^{-\beta}}{b}, \quad (14)$$

with

$$\sigma_{ext} = l\frac{a}{2}, \quad (15)$$

$$\sigma_{int} = \left(\frac{l-1}{l}\right)k\sigma_{ext}, \quad (16)$$

$$\beta = \frac{\sigma_{int} + \frac{a}{2}}{\sigma_{int}}. \quad (17)$$

The terms σ_{int} and σ_{ext} are the maximum exponential decay rate for the inner velocity and outer position loops

respectively. The term $l > 1$ is a scaling factor representing how fast is the desired position loop response with respect to the LUM open-loop response, defined as follows

$$l = \frac{e + ek^2 - 2e^{1/k}k^2}{3e - 2e^{1/k}k + ek^2 - 2e^{1/k}k^2} \quad (18)$$

where k is a scaling factor between σ_{int} and σ_{ext} .

Therefore, given the value of σ_{ext} , a value of l is computed using (15) and the estimated value of a . Then, substituting l into (18) produces a transcendental equation to be solved for k using a numerical method. Solving for k allows computing β , σ_{int} and subsequently the gain K_p

$$K_p = \frac{\sigma_{ext}(\sigma_{ext}^2 - a\sigma_{ext} + \gamma - 2\sigma_{int}^2 e^{-\beta + \frac{\sigma_{ext}}{\sigma_{int}}})}{\gamma - 2\sigma_{int}^2 e^{-\beta}}, \quad (19)$$

with

$$\gamma = \frac{a^2}{4} + \sigma_{int}^2. \quad (20)$$

Further details on this tuning are found in [10].

V. EXPERIMENTAL RESULTS

A. Experiments using the parameter estimates obtained through the LS and LSOD methods

In this section, the results obtained using the CPIR controller (11) with the parameter estimates obtained through the LS and LSOD methods are presented. Fig. 3, 4 and 5 depict the positions, the control signals and the error signals respectively. The performance of the closed-loop system is evaluated through the Integral Squared Error (*ISE*), the Integral of the Absolute value of the Error (*IAE*), the Integral of the Absolute value of the Control (*IAC*) and the Integral of the Absolute value of the Control Variation (*IACV*) indexes that are defined as

$$ISE = 100 \int_{\tau_1}^{\tau_2} [e(t)]^2 dt, \quad IAE = 100 \int_{\tau_1}^{\tau_2} |e(t)| dt,$$

$$IACV = \int_{\tau_1}^{\tau_2} \left| \frac{du(t)}{dt} \right| dt, \quad IAC = \int_{\tau_1}^{\tau_2} |u(t)| dt,$$

which are evaluated at $\tau_1 = 1$ and $\tau_2 = 5$ s.

Table II resumes the outcomes of the experiments. Table III shows the gain values used in the CPIR controller (11).

From Table II, it is clear that good performance is obtained using the LSOD parameter estimates for tuning the CPIR controller according to the ISE and

Table II
 EXPERIMENTAL RESULTS USING LS AND LSOD.

	ISE	IAE	$IACV$	IAC	e_{pulses}
LS	0.8318	0.5315	3.2533	1.0062	1
LSOD	0.0357	0.0904	14.2492	0.9449	1

 Table III
 GAIN VALUES USING LS AND LSOD.

	h	K_i	K_{ir}	σ_{ext}	σ_{int}	l	K_p	k
LS	0.0159	4.7434e+03	2.1928e+03	0.51a	84.8842	1.02	9.6295	143.0713
LSOD	0.0028	2.8849e+04	1.3336e+04	0.51a	489.8863	1.02	55.5741	143.0713

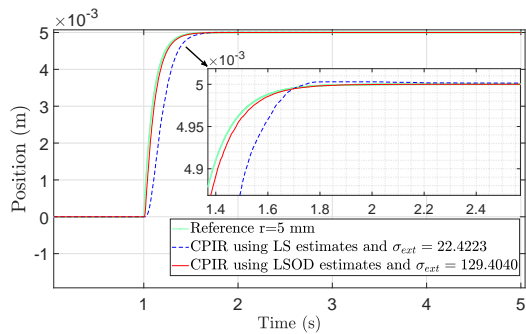


Figure 3. Position responses for a filtered step reference of 5 mm.

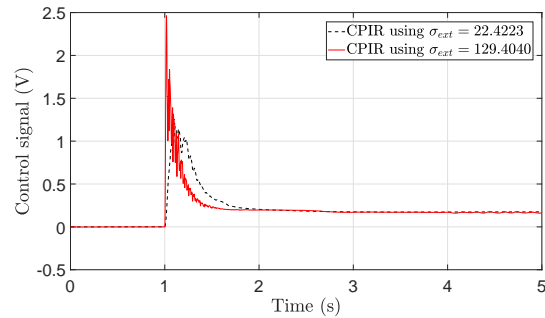


Figure 5. Control signals for a step reference of 5 mm

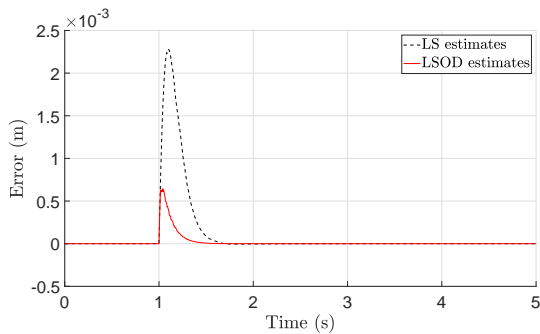


Figure 4. Error positions for a step reference of 5 mm.

IAE indexes. On the other hand, for the for LS method, those indexes have higher values. These results suggest that the best parameter estimates would correspond to the LSOD method. Moreover, it is worth noting that the IACV index is higher with a tuning performed using the LSOD estimates compared with the corresponding index for the LS method. However, the energy consumption measured with the IAC index, is lower when employing the LSOD parameter estimates. Finally, using the LSOD

parameters, a steady state error of 1 pulse equivalent to 50 nm is obtained.

B. Experiments with different values of σ_{ext} using the LSOD method

In this section, the results of using the controller (11) with the parameters obtained through the LSOD method are presented for different values of σ_{ext} . The Fig. 6, 7 and 8 represent the positions, the control signals and the error signals respectively. Table IV resumes the outcomes of the experiments. Table V shows the gain values used in the controller (11).

From Table IV, it is clear that good performance is obtained using the LSOD parameter estimation method according to ISE and IAE indexes. On the other hand, note that the IACV index is higher when a large value of σ_{ext} is used. Finally, using the LSOD parameters, a steady state error of 1 pulse equivalent to 50 nm is obtained. The table also shows that when the σ_{ext} value is higher, the value of the IEC and IAC indexes have a smaller value. However, the control signal oscillates more and has a higher value according to the IACV and IAC indexes.

Table IV
EXPERIMENTAL RESULTS USING LSOD AND DIFFERENT VALUES OF σ_{ext} .

σ_{ext}	ISE	IAE	IACV	IAC	e_{pulses}
1	0.0357	0.0904	14.2492	0.9449	0
1.5	0.0263	0.0768	43.2610	1.0194	0
2	0.0227	0.0709	74.8586	1.0863	0

Table V
GAIN VALUES FOR LSOD.

h	K_i	K_{ir}	σ_{ext}	σ_{int}	l	K_p
0.0028	2.8845e+04	1.3335e+04	0.51a	489.8863	1.0200	55.5741
0.0022	4.2792e+04	2.2019e+04	0.60a	577.7323	1.2	65.3323
0.0020	5.1832e+04	2.7792e+04	0.65a	626.4936	1.3	70.7550

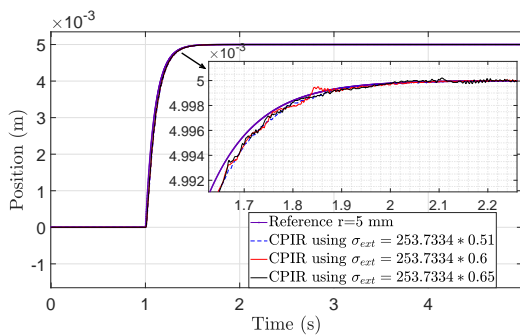


Figure 6. Position response for a filtered step reference of 5 mm for different values of σ_{ext} .

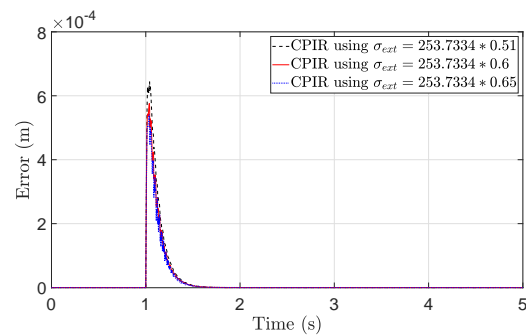


Figure 8. Error position response for a filtered step reference of 5 mm for different values of σ_{ext} .

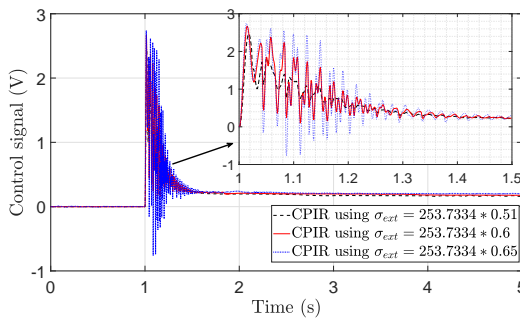


Figure 7. Control signal for a filtered step reference of 5 mm for different values of σ_{ext} .

VI. CONCLUSION

This work presents preliminary results on parametric identification using the LSOD and LS methods applied to a linear ultrasonic motor. The parameter estimates are subsequently used to tune a Cascade Proportional Integral Retarded (CPIR). The experimental results show

that using the estimates obtained through the LSOD method for tuning the CPIR controller allow obtaining better performance and a steady state error of 50 nm for a filtered step reference. These outcomes indicate that despite using a model of a linear ultrasonic motor that does not include complex friction phenomena, the LSOD parameter identification method combined with the CPIR controller yield good performance in nanopositioning tasks.

ACKNOWLEDGMENT

The authors want to thank the support of Gerardo Castro Zavala and Jesús Meza Serrano in setting up the laboratory testbed.

REFERENCES

- [1] J. M. Fernandez and Y. Perriard, "Comparative analysis and modeling of both standing and travelling wave ultrasonic linear motor," in *Ultrasonics, 2003 IEEE Symposium on*. IEEE, 2003.
- [2] J. Tichy, J. Erhart, and E. Kittinger, *Fundamentals of piezoelectric sensorics*. Springer., 2010.
- [3] T. Hemsell and J. Wallaschek, "State of the art and development trends of ultrasonic linear motors," in *Ultrasonics Symposium, 2000 IEEE*. IEEE, 2000.

-
- [4] S.-t. Ho, "Modelling of the linear ultrasonic motor using an elliptical shape stator," in *Mechatronics, 2006 IEEE International Conference on*. IEEE, 2006, pp. 82–87.
 - [5] I. Rusnak, "The generalized pid controller and its application to control of ultrasonic and electric motors," *IFAC Proceedings Volumes*, vol. 33, no. 4, pp. 119–124, 2000.
 - [6] F.-J. Lin, "Fuzzy adaptive model-following position control for ultrasonic motor," *IEEE Transactions on Power Electronics*, 1997.
 - [7] Y. Li and Q. Xu, "Adaptive sliding mode control with perturbation estimation and pid sliding surface for motion tracking of a piezo-driven micromanipulator," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 798–810, 2010.
 - [8] R. Villafuerte, S. Mondié, and R. Garrido, "Tuning of proportional retarded controllers: theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 983–990, 2012.
 - [9] A. Ramírez, R. Garrido, and S. Mondié, "Velocity control of servo systems using an integral retarded algorithm," *ISA transactions*, vol. 58, pp. 357–366, 2015.
 - [10] K. López, S. Mondié, and R. Garrido, "A tuning procedure for the cascade proportional integral retarded controller," *IFAC-PapersOnLine*, vol. 51, no. 14, pp. 61–65, 2018.
 - [11] G. Ellis, *Control system design guide: using your computer to understand and diagnose feedback controllers*. Butterworth-Heinemann, 2012.
 - [12] A. Abdulle and G. Wanner, "200 years of least squares method," *Elemente der Mathematik*, vol. 57, no. 2, pp. 45–60, 2002.
 - [13] Y. Bard, "Nonlinear parameter estimation," 1974.
 - [14] R. J. Adcock, "A problem in least squares," *The Analyst*, vol. 5, no. 1, pp. 53–54, 1878.
 - [15] I. Petras and I. Podlubny, "Least squares or least circles? a comparison of classical regression and orthogonal regression," *Chance*, vol. 23, no. 2, pp. 38–42, 2010.
 - [16] <http://studylib.net/doc/5770643/nanomotion-customer-presentation>, accesado : 22-FEBRERO-2018.
 - [17] L. A. C. Cantera, L. Luna, C. Vargas-Jarillo, and R. Garrido, "Parameter estimation of a linear ultrasonic motor using the least squares of orthogonal distances algorithm," in *2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. IEEE, 2019, pp. 1–6.

Using Active Disturbance Rejection for Chaotizing a class of Linear Systems

Erick Asiain, Adrian Lopez, Luis Luna, Rubén Garrido
Automatic Control Department
CINVESTAV-IPN
Email: easiain, mlopezc, jluna, garrido@ctrl.cinvestav.mx

Abstract—This article introduces active disturbance rejection for inducing a chaotic behavior in third order linear systems. A control law composed of a state feedback controller and a Disturbance Observer (DOB) compensate for the unknown terms and perturbations in the linear system. Moreover, the control law also relies on a jerk function of a known chaotic system. In this way, the linear system together with the proposed control law is able to track the chaotic system. Using the Largest Lyapunov Exponent criterion allows verifying that the system acquires a chaotic behavior. A stability proof is given to ensure that the closed-loop system trajectories are Uniformly Ultimately Bounded (UUB). Numerical simulations show the effectiveness of the proposed methodology.

I. INTRODUCTION

The chaotic behavior of some systems has attracted the attention of several scientific communities for almost four decades because they appear in systems and models in biology, chemistry, physics, economics among others [1], [2], [3], [4]. In robotics, some applications are trajectory planning of mobile robots, chaos in bipedal locomotion [5] and the tracking trajectory task of a two-link flexible robot manipulator where the desired path is a chaotic signal [6].

On the other hand, in the chemical industry the *chaotic stirring*, particularly of fluids is a relevant area of research on the use of controlled chaos [7], [8]. Fluid mixing aims to thoroughly mix two or more fluids of different kinds using the least amount of control energy possible, which is easy to achieve if the particle motion dynamics are strongly chaotic [9].

In [10], the authors propose and implement the chaotization of a Direct Current (DC) motor using time-delay feedback control to produce chaotic mixing processes. Chaotic processes also appear in electromechanical systems like in a permanent-magnet synchronous motor [11]. In this case the authors show that this kind of motors may exhibit different dynamics such as steady states, limit cycles and chaos. Reference [12] describes the

dynamics of a system consisting of an electrical Duffing oscillator coupled to a linear mechanical oscillator, and uses feedback controllers to make the electromechanical device to evolve from a chaotic trajectory to a regular target orbit.

Chaotification methods of linear dynamical systems have been studied in [13], where a tracking trajectory task is carried out through indirect and direct adaptive control and state feedback control. Synchronization of chaotic systems has been studied in [14], where an Active Disturbance Rejection Control (ADRC) approach is applied. For this purpose a Generalized Proportional Integral (GPI) Observer is employed and it allows to approximately estimate, on the basis of the measured output, the states of the nonlinear system, as well as to closely estimate an unknown perturbation.

Another ADRC approach that is robust and popular due to its simplicity, flexibility and efficacy is the Disturbance Observer (DOB) because it deals with plant uncertainties as well as exogenous and endogenous disturbances [15]. In DOB-based control, internal and external disturbances are estimated by using a nominal model and measurable states of the plant. This control technique has been experimentally verified in many different engineering applications, such as in robotics and servo systems [16], [17].

The DOB employs the inverse of a nominal model of a minimum phase plant together with a filter to estimate the disturbance. Hence, the DOB closes an inner loop, and a controller designed for the undisturbed nominal model of the plant closes an outer loop. It is worth remarking that in most past work on the DOB for linear plants, the analysis and design are accomplished in the frequency domain. As far as the authors know, these types of controllers have not been used for chaotic trajectory tracking tasks. By its nature, simplicity and easy implementation, the use of DOB based controllers

for the chaotization is attractive.

The objective of this work is the chaotization of a third order linear system through a state feedback controller endowed with a Disturbance Observer, plus a jerk function that corresponds to a known third order chaotic system.

The overview of the work is as follows. Section 2 briefly introduces the main features of chaotic systems. Section 3 presents an overview of the DOB. Section 4 describes the methodology for chaotization of a third order linear system. Section 5 describes a stability proof that allows concluding that the trajectories of the closed loop system are Uniformly Ultimately Bounded (UUB). Section 6 reports some simulation results. Finally, Section 7 gives conclusions and describes the future work to be developed.

II. CHAOTIC SYSTEMS

A system whose output highly depends on its initial conditions is called chaotic [18], [19]. In this kind of systems, two trajectories arising from two nearby initial conditions separate exponentially over time [20]. The practical implication is that long-term prediction becomes impossible in a chaotic system, where small uncertainties amplify very quickly [21].

In this work the control goal is chaotization, which is defined as a process wherein an originally non-chaotic dynamical system acquires chaotic behavior using a control scheme, or when a chaotic system improves existing chaos [22]. A criterion to know if a system exhibits chaotic behavior is through the Lyapunov exponents. They measure the average convergence or divergence rates of nearby orbits in the phase space. For regular motions the Largest Lyapunov Exponent (LLE) is negative, but for chaotic movement the LLE is positive [23].

III. DISTURBANCE BASED CONTROL

In DOB-based control, input and output plant measurements are used to obtain an estimate of the disturbance to later compensate for its effects.

A scheme of DOB-based control for linear systems is shown in Fig. 1. Some assumptions in this control technique are that the plant is minimum phase and the state is assumed available. To reconstruct the disturbance, an inverse model of the plant is necessary, however, a problem is that the plant inverse $P^{-1}(s)$ is not proper. In order to circumvent the above issue a filter $F(s)$ is added to guarantee that the transfer function $P^{-1}(s)F(s)$ is proper or strictly proper.

According to this scheme the disturbance may be interpreted as an input-output error modeling term. It is

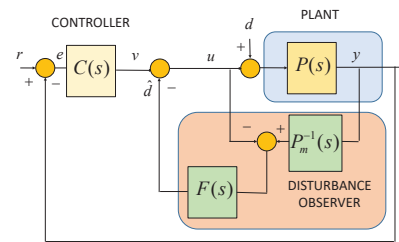


Figure 1. Disturbance Observer block diagram.

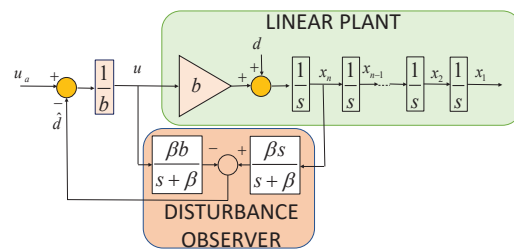


Figure 2. Disturbance Observer applied to a chain of integrators.

also worth noting that this estimation scheme assumes that the bandwidth of the filter $F(s)$ is equal or greater than the bandwidth of the disturbance.

In the particular case of a linear perturbed integrator chain (see Fig. 2), the dynamics of the DOB is given by the next expression

$$\dot{\hat{d}} = -\beta\hat{d} + \beta(\dot{x}_n - bu) \quad (1)$$

IV. PROPOSED STATE FEEDBACK WITH A DOB-BASED CONTROLLER

A. Control law

Consider the plant model whose control signal is u , and whose states x_1 , x_2 and x_3 are available through measurements

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= bu + \rho \end{aligned} \quad (2)$$

where b is the plant input gain and ρ is a bounded disturbance. It is assumed that b is constant and known, and ρ and $\dot{\rho}$ fulfill

$$|\rho| \leq R_1, \quad |\dot{\rho}| \leq R_2 \quad (3)$$

with R_1 and R_2 positive constants.

Consider the Rössler oscillator whose model is given by [24]

$$\begin{aligned}\dot{y}_1 &= y_2 \\ \dot{y}_2 &= y_3 \\ \dot{y}_3 &= G\end{aligned}\quad (4)$$

where

$$G = -0.2y_3 - y_2 - 0.586y_1 + 0.386y_1^2 \quad (5)$$

is the jerk function of this system.

Let the control law u be described by

$$u = \frac{1}{b}(\bar{u} - \hat{\rho}) \quad (6)$$

where \bar{u} corresponds to the control law for (2) without taking into account disturbances, and $\hat{\rho}$ is an estimate of ρ . Assume that $\hat{\rho}$ completely counteracts the effects of ρ . Then, substituting (6) into (2) yields

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= \bar{u}\end{aligned}\quad (7)$$

Define the error signals as

$$\begin{aligned}e_1 &= y_1 - x_1 \\ e_2 &= y_2 - x_2 \\ e_3 &= y_3 - x_3\end{aligned}\quad (8)$$

The corresponding time derivatives of (8) are

$$\begin{aligned}\dot{e}_1 &= \dot{y}_1 - \dot{x}_1 \\ \dot{e}_2 &= \dot{y}_2 - \dot{x}_2 \\ \dot{e}_3 &= \dot{y}_3 - \dot{x}_3\end{aligned}\quad (9)$$

Substituting (4) and (7) into (9) gives

$$\begin{aligned}\dot{e}_1 &= e_2 \\ \dot{e}_2 &= e_3 \\ \dot{e}_3 &= G - \bar{u}\end{aligned}\quad (10)$$

It is desired to find a state feedback control law \bar{u} generating the following error dynamics

$$\begin{aligned}\dot{e}_1 &= e_2 \\ \dot{e}_2 &= e_3 \\ \dot{e}_3 &= -\lambda_1 e_3 - \lambda_2 e_2 - \lambda_3 e_1\end{aligned}\quad (11)$$

From (10) and (11) it follows that

$$\bar{u} = \lambda_1 e_3 + \lambda_2 e_2 + \lambda_3 e_1 + G \quad (12)$$

B. Closed-loop dynamics

Replacing (6) into (2) produces

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= \bar{u} - \hat{\rho} + \rho\end{aligned}\quad (13)$$

Defining

$$\tilde{\rho} = \hat{\rho} - \rho \quad (14)$$

allows writing (13) as

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= \bar{u} - \tilde{\rho}\end{aligned}\quad (15)$$

Substituting (4) and (15) into (9) leads to

$$\begin{aligned}\dot{e}_1 &= e_2 \\ \dot{e}_2 &= e_3 \\ \dot{e}_3 &= G - \bar{u} + \tilde{\rho}\end{aligned}\quad (16)$$

Therefore, from (12) it follows that

$$\begin{aligned}\dot{e}_1 &= e_2 \\ \dot{e}_2 &= e_3 \\ \dot{e}_3 &= -\lambda_1 e_3 - \lambda_2 e_2 - \lambda_3 e_1 + \tilde{\rho}\end{aligned}\quad (17)$$

On the other hand, the variable $\hat{\rho}$ is estimated as

$$\dot{\hat{\rho}} = -\beta \hat{\rho} + \beta(\dot{x}_3 - bu) \quad (18)$$

where $\beta > 0$ sets the cutoff frequency. This equality corresponds to a standard DOB. Now, introducing the control law (6) into the above equation leads to

$$\dot{\hat{\rho}} = -\beta \hat{\rho} + \beta(\dot{x}_3 - \bar{u} + \hat{\rho}) \quad (19)$$

Substituting \dot{x}_3 given in (15) into (19) produces

$$\dot{\hat{\rho}} = -\beta \tilde{\rho} \quad (20)$$

Consequently, the time derivative of $\tilde{\rho}$ is

$$\dot{\tilde{\rho}} = -\beta \tilde{\rho} - \dot{\rho} \quad (21)$$

Defining

$$E = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \tilde{\rho} \end{bmatrix} \quad (22)$$

allows binding the closed-loop error dynamics (17) to the dynamics of $\tilde{\rho}$, thus leading to

$$\dot{E} = AE - \Gamma \quad (23)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\lambda_3 & -\lambda_2 & -\lambda_1 & 1 \\ 0 & 0 & 0 & -\beta \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{\rho} \end{bmatrix}$$

Note that the characteristic polynomial of A is

$$\det(sI - A) = (\beta + s)(s^3 + \lambda_1 s^2 + \lambda_2 s + \lambda_3) \quad (24)$$

Hence, matrix A is Hurwitz stable when the next conditions hold

$$\begin{aligned} \beta &> 0 \\ \lambda_1, \lambda_2, \lambda_3 &> 0 \\ \lambda_1 \lambda_2 &> \lambda_3 \end{aligned} \quad (25)$$

The above implies that $\forall Q \in \mathbb{R}^{4 \times 4} > 0$, the Lyapunov equation

$$A^\top P + PA = -Q \quad (26)$$

has an unique solution $P \in \mathbb{R}^{4 \times 4} > 0$ [25].

V. STABILITY PROOF

Consider the following Lyapunov function candidate

$$V = E^\top P E \quad (27)$$

whose time derivative is

$$\dot{V} = \dot{E}^\top P E + E^\top P \dot{E} \quad (28)$$

Replacing (23) and (26) into (28) produces

$$\dot{V} = -E^\top Q E - 2E^\top P \Gamma \quad (29)$$

Let the upper bounds of the ensuing terms be

$$-E^\top Q E \leq -\lambda_{\min}(Q) \|E\|^2 \quad (30)$$

$$-2E^\top P \Gamma \leq 2\|E\| \lambda_{\max}(P) R_2 \quad (31)$$

where $\lambda_{\min}(Q)$ corresponds to the smallest eigenvalue of Q and $\lambda_{\max}(P)$ corresponds to the largest eigenvalue of P . The operator $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ denotes the Euclidean norm.

Substituting (30) and (31) into (28) yields

$$\dot{V} \leq -\lambda_{\min}(Q) \|E\| \left(\frac{\|E\| - 2\lambda_{\max}(P) R_2}{\lambda_{\min}(Q)} \right) \quad (32)$$

Consequently, $\dot{V} \leq 0$ as long as

$$\|E\| \geq \frac{2\lambda_{\max}(P) R_2}{\lambda_{\min}(Q)} \quad (33)$$

Hence, the trajectories of the closed-loop system are Uniformly Ultimately Bounded.

Therefore, according to the above, the control law applied to the plant (2) is

$$\begin{aligned} u &= \frac{1}{b} (\bar{u} - \hat{\rho}) \\ \bar{u} &= \lambda_1 e_3 + \lambda_2 e_2 + \lambda_3 e_1 + G \\ \dot{\hat{\rho}} &= -\beta \hat{\rho} + \beta (\dot{x}_3 - bu) \\ G &= -0.2y_3 - y_2 - 0.586y_1 + 0.386y_1^2 \end{aligned} \quad (34)$$

The next proposition resumes the preceding stability result

Proposition 1. *Consider the plant whose model is (2) in closed-loop with control law (34). If (33) is valid, then E remains bounded, and the closed-loop system is Ultimately Uniformly Bounded.*

VI. SIMULATION RESULTS

The next numerical simulation shows the application of the proposed control law (34) to solve the problem of tracking a chaotic system that corresponds to the Rössler oscillator 4. The plant used to accomplish the simulation corresponds to the following linear system

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= bu + \rho \end{aligned} \quad (35)$$

where $\rho = -5x_2 - 6x_3$ and $b = 50$.

The performance of the closed-loop system is evaluated through the Integral Squared Error (*ISE*), the Integral of the Absolute value of the Error (*IAE*), the Integral of the Absolute value of the Control (*IAC*) and the Integral of the Absolute value of the Control Variation (*IACV*) indexes that are defined as

$$ISE = 100 \int_{\tau_1}^{\tau_2} [e(t)]^2 dt, \quad IAE = 100 \int_{\tau_1}^{\tau_2} |e(t)| dt,$$

$$IACV = \int_{\tau_1}^{\tau_2} \left| \frac{du(t)}{dt} \right| dt, \quad IAC = \int_{\tau_1}^{\tau_2} |u(t)| dt,$$

which are evaluated at $\tau_1 = 38$ s and $\tau_2 = 40$ s. The LLE are computed through the algorithm reported in [26].

A. Simulations using different values of β

Figures 3, 4 and 5 depict the values of x_1 , the errors $e_1 = y_1 - x_1$, and the control signals u for several values of the DOB cutoff frequency β . The gain values used in the state feedback controller remain constant for all the experiments

$$\lambda_1 = 20, \quad \lambda_2 = 15, \quad \lambda_3 = 15$$

Table I resumes the outcomes of the experiments. The table also indicates the initial conditions for the chaotic system. In addition, the Largest Lyapunov Exponents are observed in the last column. These values show that the linear system exhibits a chaotic operation. Fig. 3 shows the signal of x_1 at different values of the cutoff frequency β . Note that the tracking error decreases with increasing values of β as depicted in Fig. 4. Table I shows that the IAE index decreases more than 90% when a value

Table I
EXPERIMENTAL RESULTS WITH INITIAL CONDITIONS
 $y_1(0) = 0.2, y_2(0) = 0, y_3(0) = 0$.

β	ISE	IAE	IAC	IACV	LLE
1	65.7643	0.3459	0.1366	0.1599	0.1027
20	0.4016	0.0262	0.1413	0.2065	0.1047
40	0.0977	0.0128	0.1403	0.2073	0.1069

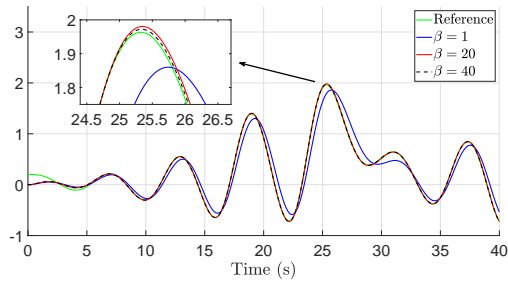


Figure 3. Linear system responses x_1 for different values of β .

Table II
EXPERIMENTAL RESULTS WITH INITIAL CONDITIONS
 $y_1(0) = 0.3, y_2(0) = 0, y_3(0) = 0$.

β	ISE	IAE	IAC	IACV	LLE
1	89.2672	0.3638	0.1901	0.0712	0.0905
20	1.1000	0.0463	0.1979	0.0983	0.1020
40	0.2715	0.0230	0.1953	0.0982	0.1015

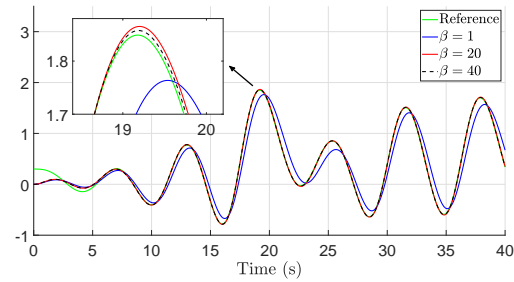


Figure 6. Linear system response x_1 for different values of β .

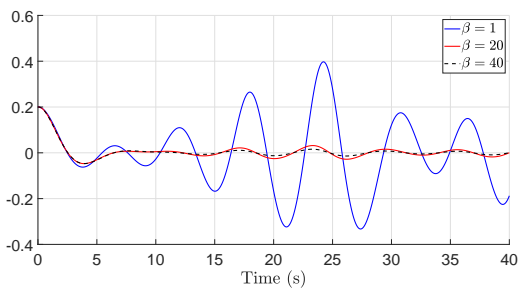


Figure 4. Error signals $e_1(t)$ for different values of β .

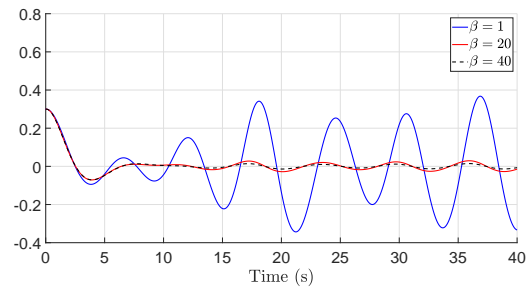


Figure 7. Error signals $e_1(t)$ for different values of β .

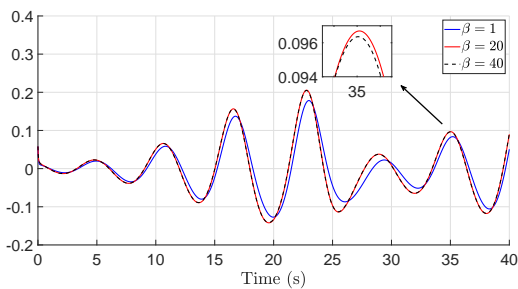


Figure 5. Control signals for different values of β

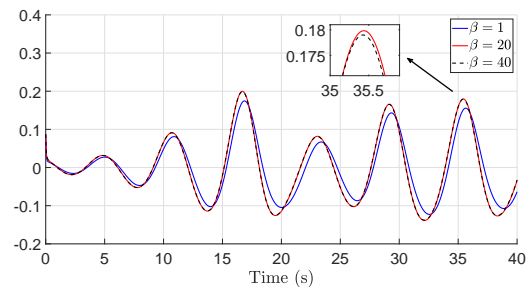


Figure 8. Control signals for different values of β .

of $\beta = 20$ is used (see Fig. 4). Interestingly enough, a slight increase in the LLE is observed for increasing values of β .

B. Simulation with another set of initial condition for the chaotic system

This subsection presents experiments with another set of initial conditions for the chaotic system. They are

depicted in Table II. Fig. 6 shows that also in this case the controller performance improves when the value of β increases. On the other hand, in Table II, the IAE index shows that the tracking error decreases more than 85% when a value of $\beta = 20$ is used (see also Fig. 7). The values of the LLE are also positive thus indicating a chaotic operation.

VII. CONCLUSIONS

This paper presents preliminary results on using a state feedback controller plus a Disturbance Observer (DOB) for chaotizing a third order linear system. The simulations and the stability analysis show that the trajectories of the closed-loop system are bounded. Moreover, the use of the DOB significantly improves the controller performance. The Largest Lyapunov Exponents corresponding to the output of the linear system are positive, then the system acquires a chaotic behavior. Future work includes the application of the proposed controller to a real plant and the extension of the results to the case of linear systems of arbitrary order.

REFERENCES

- [1] H.-K. CHEN, "Chaos and chaos synchronization of a symmetric gyro with linear-plus-cubic damping," *Journal of Sound and Vibration*, vol. 255, no. 4, pp. 719 – 740, 2002.
- [2] R. Roy, T. W. Murphy, T. D. Maier, Z. Gills, and E. R. Hunt, "Dynamical control of a chaotic laser: Experimental stabilization of a globally coupled system," *Phys. Rev. Lett.*, vol. 68, pp. 1259–1262, Mar 1992.
- [3] V. Petrov, V. Gaspar, J. Masere, and K. Showalter, "Controlling chaos in the belousov–zhabotinsky reaction," *Nature*, vol. 361, no. 6409, pp. 240–243, 1993.
- [4] E. Mosekilde and L. Mosekilde, *Complexity, chaos, and biological evolution*. Springer Science & Business Media, 2013, vol. 270.
- [5] X. Zang, S. Iqbal, Y. Zhu, X. Liu, and J. Zhao, "Applications of chaotic dynamics in robotics," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 60, 2016.
- [6] K. Lochan, J. P. Singh, B. K. Roy, and B. Subudhi, "Chapter 12 - chaotic path planning for a two-link flexible robot manipulator using a composite control technique," in *Recent Advances in Chaotic Systems and Synchronization*, ser. Emerging Methodologies and Applications in Modelling, O. Boubaker and S. Jafari, Eds. Academic Press, 2019, pp. 233 – 257.
- [7] J. M. Ottino, *The kinematics of mixing: stretching, chaos, and transport*. Cambridge University Press, 1989, vol. 3.
- [8] A. Sharma and N. Gupte, "Control methods for problems of mixing and coherence in chaotic maps and flows," *Pramana*, vol. 48, no. 1, pp. 231–248, Jan 1997.
- [9] G. Chen and X. Yu, *Chaos control: theory and applications*. Springer Science & Business Media, 2003, vol. 292.
- [10] K. T. Chau, Shuang Ye, Yuan Gao, and J. H. Chen, "Application of chaotic-motion motors to industrial mixing processes," in *Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting.*, vol. 3, Oct 2004, pp. 1874–1880 vol.3.
- [11] Zhong Li, Jin Bae Park, Young Hoon Joo, Bo Zhang, and Guanrong Chen, "Bifurcations and chaos in a permanent-magnet synchronous motor," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 3, pp. 383–387, March 2002.
- [12] R. Yamapi, J. Chabi Orou, and P. Woafu, "Harmonic oscillations, stability and chaos control in a non-linear electromechanical system," *Journal of Sound and vibration*, vol. 259, no. 5, pp. 1253–1264, 2003.
- [13] D. I. Sepúlveda, M. A. Duarte-Mermoud, and J. C. Travieso-Torres, "Chaotification of unknown linear and nonlinear systems with applications," *Electrical Engineering*, vol. 90, no. 1, pp. 3–17, 2007.
- [14] A. Luviano-Juarez, J. Cortes-Romero, and H. Sira-Ramirez, "Synchronization of chaotic oscillators by means of generalized proportional integral observers," *International Journal of Bifurcation and Chaos*, vol. 20, no. 05, pp. 1509–1517, 2010.
- [15] K. Ohishi, K. Ohnishi, and K. Miyachi, "Adaptive dc servo drive control taking force disturbance suppression into account," *IEEE transactions on industry applications*, vol. 24, no. 1, pp. 171–176, 1988.
- [16] E. Sariyildiz, H. Sekiguchi, T. Nozaki, B. Ugurlu, and K. Ohnishi, "A stability analysis for the acceleration-based robust position control of robot manipulators via disturbance observer," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 5, pp. 2369–2378, 2018.
- [17] R. Garrido and J. L. Luna, "On the equivalence between pd+ dob and pid controllers applied to servo drives," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 95–100, 2018.
- [18] S. Boccaletti, J. Kurths, G. Osipov, D. Valladares, and C. Zhou, "The synchronization of chaotic systems," *Physics Reports*, vol. 366, no. 1, pp. 1 – 101, 2002.
- [19] F. C. Moon, *Chaotic and fractal dynamics: introduction for applied scientists and engineers*. John Wiley & Sons, 2008.
- [20] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.*, vol. 64, pp. 821–824, Feb 1990.
- [21] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC Press, 2018.
- [22] J. Moreno-Valenzuela, "Lyapunov function-based adaptive chaos anti control of robot manipulators," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, June 2014, pp. 1221–1226.
- [23] J. J. Thomsen, *Vibrations and stability: advanced theory, analysis, and tools*. Springer Science & Business Media, 2013.
- [24] D. I. Sepúlveda, M. A. Duarte-Mermoud, and J. C. Travieso-Torres, "Chaotification of unknown linear and nonlinear systems with applications," *Electrical Engineering*, vol. 90, no. 1, pp. 3–17, Nov 2007.
- [25] C.-T. Chen, *Linear System Theory and Design*, 3rd ed. New York, NY, USA: Oxford University Press, Inc., 1998.
- [26] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285 – 317, 1985.

Control Conexionista de un Vehículo Autónomo

Juan Manuel Ibarra Zannatha, Oscar González Miranda, Adriana López López

Laboratorio de Robótica y Visión Artificial
Departamento de Control Automático del Cinvestav
Ciudad de México

jibarra@cinvestav.mx ogonzalez@ctrl.cinvestav.mx, alopezl1406@alumno.ipn.mx

Resumen—Se propone un esquema de control inteligente en lazo cerrado basado en tres redes neuronales, cada una de ellas realiza una de estas funciones: i) Conducción autónoma visual sobre una carretera; ii) Realización de la maniobra de rebase de vehículos lentos en carretera; y iii) Estacionarse. En el primer caso, el sistema de control de dirección del vehículo (steering) utiliza como única entrada la información visual provista por la cámara frontal del vehículo, mientras que en los dos casos restantes la información utilizada es la que proporciona el Lidar. Las características particulares de estas tres tareas son las especificadas en las reglas del Torneo Mexicano de Robótica en su categoría de Automodel Car, mientras que el vehículo a escala utilizado en este proyecto es el AutoNOMOS desarrollado por la Universidad Libre de Berlín, Alemania, que es la plataforma estándar de la competición antes señalada. Los patrones de entrenamiento y validación para la red neuronal que hace la conducción visual autónoma se obtuvieron utilizando uno de los mejores controladores visuales que se desarrollaron con anterioridad para hacer varios recorridos de la pista TMR a fin de registrar en un bag de ROS tanto el steering calculado por el controlador como el video correspondiente. Para las maniobras de rebase y de estacionamiento se diseñó una agresiva trayectoria modelo, es decir muy ajustada y por tanto riesgosa, utilizada en el vehículo en lazo abierto y en condiciones ideales en cuanto a la colocación de todos los vehículos a fin de registrar en un bag de ROS tanto el steering utilizado como la información generada por el Lidar embarcado. En los tres casos se utilizaron perceptrones multicapas con una capa oculta de nodos sigmoidales y un nodo de salida lineal entrenados con el algoritmo BP. Los resultados obtenidos por los tres controladores neuronales, en simulación, se comparan contra los obtenidos usando el controlador visual que mejores resultados ha dado para la conducción autónoma, así como las trayectorias precalculadas y ejecutadas en lazo abierto para las maniobras de rebase y estacionamiento.

Keywords—Vehículos autónomos, control de dirección, redes neuronales.

I. INTRODUCCIÓN

Una de las tendencias más importantes en la industria automotriz junto con los sistemas de propulsión alternativos (motores eléctricos, bombas de hidrógeno) son los sistemas automáticos que permiten a los vehículos una autonomía en su manejo en distintos grados o niveles. La aplicación cada vez más extensa de tecnologías de informática y de comunicación en los distintos sistemas de los automóviles posibilita cambios disruptivos en esta industria que asocia no solo a las armadoras de automóviles y a los fabricantes de autopartes, sino también a grandes empresas del sector de las TICs.

Las grandes empresas del sector automotriz y del sector de la TICs llevan prácticamente tres décadas trabajando en el desarrollo de vehículos autónomos que sea capaces de conducirse en ambientes urbanos o en carreteras sin intervención humana. Los sistemas de manejo autónomo son ya una realidad en el presente en las calles y carreteras del mundo incluyendo nuestro país, y han beneficiado de los recientes avances en la investigación en robótica móvil así como los de inteligencia artificial. Los sistemas de conducción autónoma combinan sistemas de navegación, inteligencia artificial, sistemas avanzados de percepción basados en sensores embarcados, sistemas de monitoreo del tráfico, sistemas de transporte inteligente y sistemas de comunicación entre vehículos y la infraestructura carretera y/o urbana. En este contexto una de las tecnologías empleadas con éxito son los controladores conexionistas basados en redes neuronales artificiales.

A. Controladores Conexionistas para Vehículos Autónomos

Una red neuronal artificial (RNA) puede verse como un aproximador universal de funciones cuyos nodos, llamados neuronas, están profusamente conectados entre sí siguiendo diversas topologías. Estos nodos adquieren y procesan la información de manera no lineal, están interconectados y organizados en capas y operan de forma paralela. Debido a su robustez, adaptabilidad y fácil implementación representan una metodología atractiva para resolver el problema de seguimiento de trayectorias para un vehículo autónomo [2-9]. Por ejemplo, en [5] se reporta cómo se utilizó una red neuronal recurrente para modelar la dinámica del mecanismo que cambia el ángulo de dirección de un auto de modo que, para obtener un ángulo de dirección deseado, la red proporciona un par adecuado al actuador del volante. Esta red, en conjunto con un controlador PID diseñado con la teoría del Control Predictivo, permitió a un Toyota Highlander conducir a velocidades entre 8 y 16 km/h; con valores RMS de hasta 10 km/h, al seguir una trayectoria dada. Por su parte, en [8] se utiliza una red neuronal *feedforward* para ajustar las ganancias de un controlador PID diseñado para cambiar el ángulo de dirección de un vehículo autónomo y así seguir una trayectoria dada. Sin embargo, en los resultados de simulación que presentan, el vehículo solo viaja a 1.4 km/h y los errores en la posición oscilan entre ± 2 m.

En [9] se utiliza una red neuronal con base de funciones radiales para estimar la incertidumbre en la rigidez de las curvas de los neumáticos; los cuales son un parámetro del modelo dinámico de un automóvil difícil de medir experimentalmente [1]. Esto les permitió diseñar un controlador robusto del ángulo de dirección y con este obtener resultados tanto en simulación como en la plataforma experimental HIL. En su trabajo reportan que, ante perturbaciones desconocidas, el método propuesto mantiene la estabilidad del ángulo de dirección del vehículo y un buen desempeño al seguir la trayectoria a una velocidad de hasta 54 km/h.

También existen las llamadas redes neuronales convolucionales, las cuales son una variación de las RNA y cuyas entradas son matrices de datos, tienen como principal campo de aplicación la Visión Artificial. Esto se debe a que son capaces de clasificar objetos a partir de imágenes, segmentar colores y reconocer patrones entre otras cosas [14]. Estas redes toman la imagen de entrada y utilizando operaciones de convolución y compresión, adquieren información que pasarán a las demás capas.

Recientemente se han utilizado redes neuronales convolucionales para obtener el ángulo de dirección a partir de las imágenes adquiridas por una cámara [4-7], [10-13]. Con este enfoque en [10] se condujo un vehículo a escala 1:5 en un ambiente urbano simulado; mientras que en [4] se reporta que se pudo conducir de forma autónoma al vehículo Dave 2 de la empresa NVIDIA a través de ambientes ciudadanos, en carretera, en caminos húmedos o con nieve e incluso en senderos sin señalización. Sin embargo, el vehículo sólo podía seguir la carretera y no era capaz de cambiar de carril o girar para cambiar de un camino a otro.

B. Problema a resolver

En la Categoría de Vehículos Autónomos (Automodel Car) del Torneo Mexicano de Robótica organizado cada año por la Federación Mexicana de Robótica los vehículos participantes deben realizar tres pruebas de manera autónoma: *i*) Recorrido de una carretera, *ii*) Maniobra de rebase de un vehículo lento en una recta de la misma carretera, y *iii*) Maniobra de estacionamiento entre dos vehículos. En este trabajo se pretende resolver el problema de control de la dirección del vehículo utilizando sendas redes neuronales para realizar con éxito cada una de las tres pruebas de la competición antes señalada.

C. Solución Propuesta

Para ello se propone un controlador de la dirección (*steering*) del vehículo para cada una de estas tres pruebas basado en una red neuronal multicapas que es entrenada mediante un sistema de aprendizaje supervisado con patrones entrada salida (x^d, y^d) , en donde el vector de entrada consiste en las mediciones visuales hechas sobre el video provisto por la cámara frontal del vehículo o bien de las mediciones generadas por el Lidar representativas de la posición relativa entre el vehículo y el obstáculo a rebasar (o bien, los vehículos entre los cuales debe

estacionarse). Mientras que la salida de las tres redes es el ángulo de dirección (*steering*) necesario para ejecutar la maniobra deseada. Los patrones de entrada-salida que se utilizan en el entrenamiento de los controladores conexionistas se obtienen de dos maneras: *i*) Haciendo la conducción mediante uno de los mejores controladores visuales clásicos, obtenidos, a lo largo de una carretera de prueba; y *ii*) Realizando las maniobras de rebase y estacionamiento mediante un controlador en lazo abierto de la dirección, el cual se diseña geoméricamente.

D. Organización del artículo

Después de esta Primera Sección introductoria, la Segunda Sección se dedica a la descripción de la plataforma empleada incluyendo los controladores que se han desarrollado para las tareas de interés de este artículo. La Tercera Sección muestra el diseño de los controladores Neuronales propuestos, mientras que los resultados obtenidos con su aplicación son objeto de la Cuarta Sección. Finalmente, la Quinta Sección incluye comentarios finales, algunas conclusiones y trabajos a futuro dentro de este proyecto.

II. DESCRIPCIÓN DE LA PLATAFORMA UTILIZADA

El vehículo empleado en este proyecto es el AutoNOMOS de la Universidad Libre de Berlín [22]. Se trata de un vehículo a escala 1:10 equipado con un sistema de tracción en las cuatro ruedas actuado por un motor sin escobillas, el cual permite un rango de velocidades de [-1000, +1000] rpm y un sistema de dirección Ackerman actuado por un servomotor que permite obtener una dirección en el rango $[42^\circ, 138^\circ]$. Un *steering* de 90° , denominada dirección neutra, hace que el vehículo se mueva en línea recta hacia adelante (o hacia atrás); aumentar esta consigna cuando el automóvil avanza hace girar el auto hacia la izquierda, mientras que disminuyéndolo girará hacia la derecha. Los sensores que equipan este vehículo son: Cámara RGBD, Cámara gran angular mirando hacia arriba (permite implementar un sistema visual de autolocalización), un Lidar y una IMU de 9 gdl. Su sistema computacional está formado por una tarjeta Odroid y dos Arduinos; cuenta, además, con un sistema operativo Ubuntu y librerías OpenCV, todo ello embebido en el Middleware ROS. Finalmente, el sistema de alimentación utiliza una batería LiPo de cuatro celdas de 14.8 V a 5200 mAh tipo 35 C y cuenta con un sistema de regulación para generar todos los voltajes necesarios en la plataforma. En la Fig. 1 se muestra este prototipo.

El vehículo cuenta con algunos sistemas de control visual el ángulo de dirección [15-19] que se han desarrollado para conducir en una carretera cualquiera y que, en particular, le permite conducirse de manera autónoma en una pista como la utilizada en el Torneo Mexicano de Robótica de la FMR en la Categoría Automodel Car [20], o bien en la pista Seat-Barcelona, los cuales se presentan a continuación de manera breve. Además, se han desarrollado un sistema de control en lazo abierto del ángulo de dirección capaz de realizar las maniobras de rebase y de estacionamiento.

Estos dos controladores, uno para rebasar y el otro para estacionarse, representan la primera de dos alternativas que permiten obtener los datos para entrenamiento del controlador conexionista correspondiente. Por tal motivo se procede a presentarlos con más detalle.

A. Controlador Visual de conducción

Actualmente se tienen implementados al menos cuatro diferentes sistemas de control visual de la dirección (*steering*), todos ellos basados en la información visual extraída de las imágenes que proporciona la cámara frontal del vehículo relativa a la forma de la carretera. Para ello, a partir de estas imágenes, distorsionadas por la proyección en perspectiva típica de cualquier cámara, realizamos primero la reconstrucción 3D de la carretera con escala en metros. A continuación, se hace la mejora de la imagen así como el realzado de las líneas que delimitan los carriles de la carretera mediante varios algoritmos de procesamiento de imágenes, es decir, se aplican diferentes tipos de filtraje para reducir ruidos, sombras y otros problemas de iluminación. Finalmente, se extrae la información sobre la forma de la carretera (su curvatura) a partir de dichas líneas. La Fig. 2 muestra una imagen obtenida por la cámara, con coordenadas en píxeles, y la reconstrucción 3D correspondiente, con coordenadas en centímetros.

Controlador 1. En este primer esquema de control se trata de obtener la curvatura de la carretera a fin de calcular el ángulo de dirección necesario para obtener el radio de giro que mejor aproxime la curvatura estimada [15], [16].

Controlador 2. Otro esquema de control se obtiene asignando potenciales de rechazo a las líneas que delimitan el carril, tanto con perfil lineal como cuadrático, mientras que al centro del carril se le asigna un potencial de atracción [17], [18].

Controlador 3. Este controlador, denominado *Preview Control*, es similar al segundo esquema, pero ahora se utilizan varias mediciones de la pendiente de la línea que delimita el carril a lo largo de la parte de la carretera que queda dentro del espacio visual de la cámara [19].

Controlador 4. En una segunda aproximación al problema de controlar la dirección del vehículo se estima la pendiente de la línea que delimita el carril y se ajusta con base a la distancia entre el vehículo y el lugar de la carretera en la que se mide la pendiente a fin de obtener el ángulo de dirección necesario de acuerdo con una ley de control óptimo [19]. Este controlador ofrece mejores prestaciones, por lo cual se utiliza para generar los patrones de entrenamiento para el control conexionista que realiza la tarea de conducción visual autónoma.

B. Controlador de la maniobra de rebase

La maniobra por automatizar consiste en rebasar un vehículo en movimiento que circula en el mismo carril o bien que se encuentra completamente detenido en dicho carril, utilizando para ello el carril en contraflujo, pues la carretera sólo tiene esos

dos carriles. La maniobra de rebase puede ejecutarse en dos fases: En la primera se debe pasar al carril de contraflujo y en la segunda se debe regresar al carril de circulación. En cada fase el vehículo debe cambiar a una trayectoria paralela a la trayectoria por la que venía circulando, de modo que cada una de ellas incluye dos cambios de dirección simétricos: izquierda, derecha y recto para pasar al carril de contraflujo y derecha, izquierda y recto para regresar a su carril.

Suponiendo que el vehículo se mueve a una velocidad v constante, son dos los parámetros que determinan la forma de la trayectoria resultante: el incremento θ en la dirección del auto ψ (el ángulo de *yaw*) tras el cual se cambia de dirección, ya sea de izquierda a derecha o viceversa; y el valor del ángulo de dirección (*steering*) el cual define el radio de giro del vehículo, tal como se muestra en la Fig. 3.

A fin de realizar la maniobra en el menor espacio y en el menor tiempo se propone utilizar el máximo valor de *steering* para todos los cambios de dirección, el cual es de $\pm 48^\circ$ a partir de la dirección neutra que es de 90° , lo cual equivale a un radio de giro de $r = 0.55$ m. En la Fig. 3 se puede apreciar que la separación entre las dos trayectorias paralelas es de $2L$, mientras que la longitud total de la maniobra es $2x$, en donde:

$$L = r(1 - \cos \theta), \quad x = r \sin \theta$$

Cada uno de los carriles de la carretera tiene un ancho de 40 cm., mientras que los vehículos tienen un ancho de 20 cm. De acuerdo con las reglas del TMR Categoría Automodel Car, el vehículo por rebasar se encuentra en el carril de circulación; entonces, se requiere de una separación entre trayectorias de al menos el ancho del vehículo (20 cm) más una separación entre vehículos de, digamos 10 cm. Por tanto, $2L$ debe ser al menos de 30 cm. Como el ángulo de dirección se eligió con su valor máximo (*steering* = $\pm 48^\circ$ a partir de la dirección neutra), se tiene el mínimo valor de radio de giro, $r = 0.55$ m, por lo que el cambio de dirección debe hacerse en $\psi \pm \theta$, con $\theta = 45^\circ$, el cual se obtiene de resolver las ecuaciones anteriores. Además, se obtiene una longitud total de la maniobra de $2x = 0.80$ m, mientras que $2L = 0.32$ m. Cabe mencionar que el ángulo de *yaw* se mide con ayuda de la IMU (Unidad de Medición Inercial) del vehículo

Para determinar el momento de iniciar cada una de las dos fases de la maniobra de rebase. La primera fase se inicia en cuanto se detecte que hay un obstáculo enfrente del vehículo a una distancia menor o igual a un umbral d_R (distancia para iniciar la maniobra) y en la segunda fase se debe regresar al carril de circulación en cuanto el vehículo se encuentre a una distancia igual o mayor a d_F (distancia para finalizar la maniobra) enfrente del vehículo por rebasar. Estas dos distancias se miden con ayuda del Lidar embarcado en el vehículo. La distancia d_F es función de la velocidad relativa entre los dos vehículos. Para vehículos detenidos o muy lentos (caso contemplado en el reglamento del TMR para la categoría Automodel Car) se usa $d_R = 50$ cm, $d_F = 0$ cm (es decir, cuando el referencial del Lidar y el frente del vehículo por rebasar estén alineados), y $\theta = 45^\circ$; con lo cual se obtiene la trayectoria mostrada en la Fig. 3. El

algoritmo completo (diagrama de flujo) con el que realizamos la maniobra de rebase se muestra en la Fig. 4.

Para poder realizar la maniobra sin chocar con el otro vehículo, se debe cumplir que $v \cos(\theta) > v_o$; siendo v_o la velocidad del automóvil a rebasar. Para demostrar esto se utiliza el modelo cinemático de la bicicleta [22] dado por las ecuaciones:

$$\begin{aligned} \frac{dx}{dt} &= v \cos(\psi) \\ \frac{dy}{dx} &= v \sin(\psi) \frac{d\psi}{dt} = \frac{v}{l} \tan(st) \end{aligned}$$

En donde (X, Y, ψ) es la pose del auto, medida desde algún referencial fijo $\{O\}$ y l es la separación entre los ejes trasero y delantero de las ruedas. Sin pérdida de generalidad se puede ubicar al referencial inercial $\{O\}$ en el lugar del auto al inicio de la primera fase de la maniobra; de modo que X coincide con la dirección de avance del auto a rebasar y $\psi(0) = 0^\circ$. Suponiendo que la segunda fase de la maniobra dura $2t_0$ segundos, entonces la dirección del auto $st(t)$ varía de la forma mostrada en la Fig. 5. Entonces, del modelo matemático podemos deducir que:

$$\psi(t) = \begin{cases} -\frac{v}{l} \tan(\delta) t & \text{si } 0 \leq t < t_0 \\ -\theta + \frac{v}{l} \tan(\delta) t & \text{si } t_0 \leq t \leq 2t_0 \end{cases}$$

Con $\theta = \frac{v}{l} \tan(\delta) t_0$. Así pues, la velocidad del auto en la dirección X , mientras la maniobra está en la segunda fase, está dada por: $\dot{X}(t) = v \cos(\psi(t))$ y es mínima cuando $\psi(t) = \theta$. Por lo tanto para rebasar a otro automóvil, la relación entre las velocidades está dada por: $v \cos(\theta) > v_o$

Es importante señalar que esta trayectoria se ha diseñado con muy poca tolerancia con respecto a la posición relativa entre ambos vehículos y que, al ser controlada en lazo abierto, se corre un riesgo importante de colisión. Sin embargo, mientras no se violen las suposiciones iniciales, es decir, que ambos vehículos siguen la misma trayectoria antes de la maniobra del rebase y que tienen el mismo ancho, la tolerancia lateral de 4 cm que ha sido considerada funciona muy bien. De cualquier manera, se debe recordar que el objetivo de la maniobra diseñada sirve tan solo para la recogida de datos del Lidar y del *steering* que son utilizados en el entrenamiento de un controlador conexionista

Durante los experimentos realizados en simulación para ajustar los parámetros de esta trayectoria se hizo el registro de un **bag** ROS de datos con la información del Lidar así como del *steering* correspondiente a la trayectoria que se ha diseñado por cada una de las diferentes corridas que se hicieron de la maniobra de rebase, tanto de un vehículo detenido como en movimiento lento.

C. Controlador de la maniobra de estacionamiento

La maniobra de estacionamiento prevista en el reglamento de la Categoría Automodel Car del TMR consiste en aparcar entre dos automóviles de 40 cm de largo por 20 cm de ancho, separados una distancia equivalente a dos veces el largo del vehículo, es decir, 80 cm. Esta maniobra puede realizarse en tres fases: En la primera, el vehículo debe posicionarse al lado

izquierdo del vehículo estacionado que se encuentra más adelantado de modo que queden 4 cm de separación lateral entre ambos vehículos y que la parte trasera del vehículo estacionado se encuentre a una distancia d_E detrás del referencial del vehículo a maniobrar. Para ello se usa un controlador visual que tenga como consigna conducir paralelo a la línea izquierda de los vehículos estacionados, con la consigna de 15 cm. similar a la usada para la conducción automática en la carretera, deteniéndose cuando el Lidar mida una distancia mayor o igual a d_E entre el vehículo y la parte trasera del vehículo detrás del cual nos debemos estacionar. En la segunda fase de la maniobra de estacionamiento el vehículo debe retroceder con el volante girado al máximo hacia la derecha hasta que su dirección ψ (yaw) se incremente θ grados, es decir, cuando la dirección del vehículo sea igual a $\psi + \theta$, momento en el cual el volante se gira hacia la izquierda a su máximo valor, hasta lograr que la dirección del vehículo sea la que tenía al inicio de la maniobra de reversa, es decir ψ , y sin que la distancia con el automóvil de atrás sea menor a un umbral d_T . La tercera fase consiste en avanzar recto, es decir, con el volante en posición neutra (90°) hasta quedar a una cierta distancia d_D del auto de enfrente.

Así, la trayectoria correspondiente a esta maniobra, ilustrada en la Fig. 6, se logra mediante el algoritmo mostrado por el diagrama de flujo de la Fig. 7. La trayectoria utilizada es exactamente la misma que se usa para la maniobra de rebase, es decir que se elige el valor máximo del ángulo de dirección (*steering* = $\pm 48^\circ$), correspondiente al mínimo valor de radio de giro, $r = 0.55$ m, con un cambio de dirección en $\psi - \theta$, con $\theta = 45^\circ$, utilizando los umbrales $d_E = 30$ cm, $d_T = 20$ cm y $d_D = 30$ cm. Al igual que en el caso de la maniobra de rebase, la trayectoria para estacionamiento se diseña bastante ajustada, de modo que cambios en las condiciones iniciales o errores en la colocación de los vehículos generan riesgos de colisión durante los experimentos realizados en simulación para ajustar los parámetros de esta trayectoria se hizo el registro de un **bag** ROS de datos con la información del Lidar así como del *steering* correspondiente a la trayectoria que se ha diseñado y de la velocidad para realizar las diferentes corridas que se hicieron de la maniobra de estacionamiento.

III. CONTROLADORES CONEXIONISTAS

Se desarrollaron tres controladores conexionistas (CC), uno para realizar la conducción visual autónoma, otra para realizar la maniobra de rebase y, finalmente, una para controlar la maniobra de estacionamiento. La primera se plantea como una red neuronal para la aproximación de la función de dirección, es decir para realizar el mapeo entre forma geométrica de la carretera y el ángulo de dirección (*steering*). Mientras que, la realización de las maniobras de rebase y estacionamiento se plantea como un problema de clasificación; esto es, clasificar los patrones (ρ, α_i) generados por el Lidar en tres categorías: giro a la izquierda, giro a la derecha y continuar recto. A continuación se describe el diseño de cada una de estos tres controladores conexionistas.

A. CC para la Conducción Visual Autónoma

Para diseñar el controlador conexionista correspondiente el primer paso es definir cómo serán los patrones de entrada: se tiene la imagen de la carretera, lo cual representa una gran cantidad de información redundante de la cual sólo es útil la forma que tiene la carretera. Siguiendo la idea que se utiliza en el diseño del *Preview Control* [19] la carretera se modela por cinco segmentos lineales, quedando representada por una secuencia de las rectas equivalentes tomadas cada 15 cm codificadas como su pendiente. Así, la red neuronal directa multicapas que hace la función de controlar visualmente el ángulo de dirección o *steering*, tiene cinco neuronas en la capa de entrada cada una de ellas recogiendo la pendiente de cinco tramos rectos sucesivos que modelan la forma de la carretera. Se agrega una capa oculta con 20 nodos sigmoidales y una neurona lineal en la capa de salida. Al igual que en los dos casos siguientes, aquí se utiliza un método de aprendizaje supervisado tipo BP (*backpropagation*) para ajustar los pesos en esta red, con un conjunto de patrones obtenido del **bag** de datos, con la imagen y el *steering*, registrado durante la operación con el controlador óptimo desarrollado [19].

La red se entrena con una parte de los datos guardados en el **bag** correspondiente, mientras que las muestras restantes se utilizan para validar la red ya entrenada. Se recogen dos **bags**, uno con el automóvil moviéndose a baja velocidad (0.2 m/s) a fin de generar una buena cantidad de datos para entrenamiento y validación. Mientras que, para generar datos de prueba, se hacen recorridos a la velocidad de cruce (0.6 m/s).

B. CC para la Maniobra de Rebase

Como se ha mencionado, ejecutando las trayectorias diseñadas en la sección previa se pueden realizar las maniobras de rebase y estacionamiento en lazo abierto sin problema, siempre y cuando se verifiquen exactamente las condiciones de diseño, lo cual es muy difícil de asegurar. De ahí la necesidad de desarrollar un controlador en lazo cerrado que garantice la correcta ejecución de esas maniobras independientemente de las condiciones iniciales y de sus variaciones. Para ello se propone un controlador conexionista constituido por una red neuronal directa multicapa, que permita cerrar el lazo con respecto a la tarea utilizando la medición de la posición relativa del vehículo con respecto a los vehículos, tanto al que se debe rebasar como aquellos entre los cuales debe estacionarse, proporcionada por el Lidar (radar laser) embarcado.

En un principio se propuso una red neuronal para aproximar la función correspondiente al mapeo entre los patrones proporcionados por el Lidar y el ángulo de dirección (*steering*). Sin embargo, como la estrategia de rebase propuesta se forma con los ángulos extremos de dirección el problema fue planteado como clasificación de los patrones del Lidar en tres categorías: seguir recto, girar a la izquierda y girar a la derecha. Entonces, el primer paso en el diseño del CC para la maniobra de rebase es construir el Espacio de Clasificación (EC).

El Lidar genera el patrón formado por N vectores (ρ_i, α_i) , donde la distancia ρ_i es la distancia a un obstáculo correspondiente al ángulo $\alpha_i \in [0, 2\pi]$, con $i=1,2,\dots,N$. De manera natural el EC se puede formar con un subconjunto de las N distancias ρ para ciertos valores α , digamos para los ángulos: $(0^\circ, 10^\circ, 20^\circ, \dots, 180^\circ)$, lo cual representa muchas entradas con una cantidad de información insuficiente. Entonces se propone preprocesar la información del Lidar, para lo cual se consideran dos posibilidades: *i*) codificar el objeto detectado por el Lidar por una recta obtenida mediante Regresión Lineal; y *ii*) codificar el objeto mediante el rayo más corto del Lidar, es decir la longitud ρ_{\min} y el ángulo α correspondiente, tal como se ilustra en la Fig. 7. Finalmente se elige esta segunda opción para crear el EC, el cual queda como se ilustra en la Fig. 8, en la cual se nota que, durante la fase de rebase correspondiente a circular recto sobre el carril de rebase paralelo al vehículo por rebasar los valores (ρ_{\min}, α) quedan encimados con los de girar a la derecha. Agregando el tiempo como tercera característica en este EC las clases se separan correctamente (Fig. 10). Para considerar esta característica del EC se sugieren dos opciones: *i*) agregar la razón de cambio de las dos características iniciales $(\Delta\rho_{\min}, \Delta\alpha)$ con lo cual efectivamente se separan un poco las dos clases, aunque no lo logra completamente (Fig. 11); y *ii*) agregar la característica $(\alpha/\rho)(1/\rho_{270})$ (siendo ρ_{270} la distancia medida por el Lidar a 270°) con lo cual sí se logra la separación buscada (Fig. 12). Esto se debe a que la distancia al obstáculo decrece mucho en la zona de conflicto por lo cual “levanta” la clase seguir recto de la clase giro a la derecha. En esta última figura se aprecia claramente que se requiere un mínimo de cinco hiperplanos o discriminantes lineales para separar las tres clases, debido a que al menos uno de los *clusters* correspondientes es disjunto; para tratar de obtener mejores resultados se eligen 10 nodos en la capa oculta.

De este modo, la red resultante tiene tres nodos lineales en la capa de entrada $(\rho_{\min}, \alpha, (\alpha/\rho)(1/\rho_{270}))$, una capa oculta con 10 nodos sigmoidales y una capa de salida con tres nodos *hard limiter*, cada uno correspondiente a los tres ángulos de dirección necesarios para realizar la maniobra de rebase. Finalmente, la variable *steering* se obtiene sumando las tres salidas, cada una con un coeficiente igual al ángulo que representan:

$$st = 42y_1 + 90y_2 + 138y_3$$

La red resultante se entrena con el conjunto de patrones obtenido del **bag** de datos registrado durante la aplicación de las trayectorias en lazo cerrado diseñadas en la sección previa. Aquí también se recogen datos a dos velocidades diferentes, una lenta a fin de obtener una cantidad suficiente de datos para entrenamiento y validación, así como otra más alta, similar a la utilizada como velocidad de cruce, para probar la red.

C. CC para la Maniobra de Estacionamiento

El control de la maniobra de estacionamiento se hace en tres fases: *i*) aproximación: usando el controlador conexionista de conducción visual considerando como consigna la línea recta que delimita el espacio para estacionarse; *ii*) marcha atrás: realizada en lazo cerrado con la información del Lidar en las

dos opciones consideradas en la maniobra de rebase, es decir, usando la misma red del inciso previo, pero con una condición de paro dada por la distancia entre la parte trasera del vehículo a estacionar y el auto estacionado; y *iii*) terminación: avanzar corrigiendo la posible desviación angular con el controlador neuronal de conducción visual autónoma guiándose con la línea de la banqueta usando como criterio de paro la distancia entre la parte delantera de nuestro vehículo y el auto estacionado. La red para implementar el controlador en lazo cerrado de estacionamiento es similar a la desarrollada para rebasar, es decir que tiene dos nodos de entrada lineales, 10 nodos sigmoidales en la capa oculta y tres nodos *hard limiter* en la capa de salida, con los cuales se calcula la variable *steering* de acuerdo con la ecuación anterior.

El entrenamiento de esta red se hace con los datos almacenados en el **bag** correspondiente del mismo modo que en el caso anterior.

IV. IMPLEMENTACIÓN Y RESULTADOS

Cabe mencionar que se desarrollaron un par de máquinas de estados finitos para el control global del vehículo. La primera tiene dos estados: conducción visual sobre la carretera y rebasar. Cuando el auto se encuentra conduciendo y aparece un vehículo lento se pasa al estado rebasar; cuando termina la maniobra de rebasar se regresa al estado de conducción visual; sin embargo, este método de control también funciona para esquivar obstáculos en la carretera. En el caso de la maniobra de estacionamiento se tienen tres estados, uno para cada una de las tres fases que componen la maniobra: iniciar la maniobra, es decir, avanzar para colocarse en posición de iniciar la maniobra; retroceder, terminar la maniobra. Del primer estado se pasa al segundo cuando se encuentre a la distancia especificada; se pasa del segundo al tercer estado cuando se alcanza la tolerancia en la parte trasera del vehículo y se termina cuando se alcanza la tolerancia de distancia en la parte delantera del vehículo.

Todos los controladores desarrollados se implementaron como nodos del software ROS Kinetic y se probó su desempeño en el simulador del vehículo AutoNOMOS [21]. El modelo matemático utilizado en este simulador es el modelo cinemático de la bicicleta presentado anteriormente [22]. Previamente, todas las redes neuronales utilizadas fueron entrenadas usando el *toolbox* nntool de MATLAB 2017b.

A. CC para la Conducción Visual Autónoma

De la conducción autónoma llevada a cabo por el controlador óptimo se obtuvieron 528 datos al recorrer la pista mostrada en la Fig. 13, a 0.6 m/s. De éstos, 370 se usaron para el entrenamiento de la red, 79 para la validación de la misma y 79 se usaron como conjunto de prueba. El entrenamiento se hizo con una ganancia de aprendizaje de 0.001 y como criterio de paro se usó el parámetro $max_fail = 6$; el cual es el número máximo de épocas en las que el MSE (*mean square error*) de los datos de validación, deja de ser decreciente y se vuelve creciente.

La Fig. 13. Muestra la evolución del MSE durante el entrenamiento. Después de 27 épocas, el mínimo MSE obtenido con los datos de validación fue de 0.14; mientras que con los datos de prueba se obtuvo 0.06. La Fig. 14. Compara las trayectorias generadas por el auto al conducirse usando el controlador óptimo (en azul) y la red neuronal (en rojo). Como se puede observar el desempeño es casi el mismo y el error se mantiene entre ± 1 cm.

B. CC para la Maniobra de Rebase

Del mismo modo que la red anterior, para entrenar al clasificador y obtener datos de validación y prueba; se usó un controlador en lazo abierto el cual realizaba la maniobra ideal de rebase, mientras el auto se conducía a 0.6 m/s. Entonces se utilizaron 590 datos para entrenar la red, 127 para la validación y 127 para las prueba. Al igual que en el caso anterior, el entrenamiento se hizo con una ganancia de aprendizaje de 0.001 y con $max_fail=6$ como criterio de paro. La Fig. 15. Muestra el cambio del MSE durante el entrenamiento. El mejor desempeño MSE de la red, con los datos de validación y de prueba fue de 0.0035.

La Fig. 16. Muestra en azul la trayectoria que siguió el auto al adquirir los datos de entrenamiento mientras rebasaba un obstáculo de 20x40 cm (este no se muestra en la figura); por otro lado, en rojo está el recorrido seguido por el auto conducido por el CC. En esta maniobra, el error está acotado entre ± 9 cm.

C. CC para la Maniobra de Estacionamiento

La red neuronal utilizada para realizar esta maniobra es muy similar a la anterior ya que para ambas, la trayectoria es básicamente la misma. Es por eso que esta red se entrenó con los mismos parámetros y se obtuvo un desempeño similar. La Fig. 17. Muestra tanto la trayectoria ideal para estacionarse, llevada a cabo por un controlador en lazo abierto (en azul); como la trayectoria realizada por el CC. Las paredes entre las que se estaciona el auto, no se muestran en la figura.

V. COMENTARIOS FINALES

Se diseñaron tres redes neuronales directas de tres capas (entrada, oculta y salida) entrenadas mediante el algoritmo de aprendizaje supervisado *backpropagation*, para materializar los tres controladores necesarios para que un vehículo a escala realice las tres pruebas de la competición TMR en su categoría Automodel Car: conducir de manera autónoma sobre una carretera dada usando información visual, rebasar a un vehículo lento en dicha carretera y estacionarse utilizando información de Lidar para estos dos últimos casos. Para entrenar estas tres redes se utilizaron datos obtenidos al realizar todas estas maniobras en casos ideales y mediante controladores en lazo abierto con respecto a la tarea (para rebasar y estacionarse) y un controlador visual óptimo para la tarea de conducir. La tarea más delicada fue la codificación de la información sensorial que debería alimentarse a estas RNs; así, para el primer caso se modela la carretera por una función lineal a pedazos formada

por cinco segmentos de recta, a fin de alimentar la RN con las pendientes de estas cinco rectas.

Para las dos maniobras restantes utilizamos la red neuronal como clasificador, por lo cual se procedió a diseñar el espacio de clasificación cuyas características deberán corresponder a la información sensorial aportada por el Lidar. Para ello, esta última es codificada como una recta equivalente al obstáculo, parametrizada en coordenadas polares (ρ, θ) , la cual se obtuvo considerando la recta perpendicular al rayo del Lidar correspondiente al punto más cercano, es decir la recta (ρ_{\min}, θ) , así como la razón de cambio de este ángulo con respecto a la distancia (α/ρ_{\min}) . De este modo se necesitan sólo tres nodos en la capa de entrada. Estas dos redes se entrenaron como clasificadores con tres diferentes clases, una para cada valor de *steering*. Los resultados utilizando esta codificación fueron bastante buenos, sobre todo considerando que la RN correspondiente es bastante más pequeña y su entrenamiento converge más rápidamente.

Finalmente, los controladores neuronales desarrollados utilizan el Lidar o la cámara para cerrar el lazo de control con respecto a la tarea, motivo por el cual son capaces de absorber diferencias en posición o cambios en la velocidad del vehículo por rebasar, a pesar de haber sido entrenados con información correspondiente a condiciones ideales y sin errores de colocación.

REFERENCIAS

- [1] Ackermann, J., & Sienel, W. (1990, May). Robust control for automatic steering. In 1990 American Control Conference (pp. 795-800). IEEE.
- [2] Cheng, R. M. H., Xiao, J. W., & LeQuoc, S. (1992, May). Neuromorphic controller for AGV steering. In Proceedings 1992 IEEE International Conference on Robotics and Automation (pp. 2057-2062). IEEE.
- [3] Gu, D., & Hu, H. (2002). Neural predictive control for a car-like mobile robot. *Robotics and Autonomous Systems*, 39(2), 73-86.
- [4] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zhang, X. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.
- [5] Garimella, G., Funke, J., Wang, C., & Kobilarov, M. (2017, September). Neural network modeling for steering control of an autonomous vehicle. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2609-2615). IEEE.
- [6] Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., & Muller, U. (2017). Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911.
- [7] Chen, Z., & Huang, X. (2017, June). End-to-end learning for lane keeping of self-driving cars. In 2017 IEEE Intelligent Vehicles Symposium (IV) (pp. 1856-1860). IEEE.
- [8] Han, G., Fu, W., Wang, W., & Wu, Z. (2017). The lateral tracking control for the intelligent vehicle based on adaptive PID neural network. *Sensors*, 17(6), 1244.
- [9] Ji, X., He, X., Lv, C., Liu, Y., & Wu, J. (2018). Adaptive-neural-network-based robust lateral motion control for autonomous vehicle at driving limits. *Control Engineering Practice*, 76, 41-53.
- [10] Codevilla, F., Müller, M., López, A., Koltun, V., & Dosovitskiy, A. (2018, May). End-to-end driving via conditional imitation learning. 2018 IEEE Int. Conf. on Robotics and Automation (ICRA) (pp. 1-9). IEEE.
- [11] Toromanoff, M., Wirbel, E., Wilhelm, F., Vejarano, C., Perrotton, X., & Moutarde, F. (2018, October). End to end vehicle lateral control using a single fisheye camera. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 3613-3619). IEEE.
- [12] Hsu, T. M., Wang, C. H., & Chen, Y. R. (2018, November). End-to-End Deep Learning for Autonomous Longitudinal and Lateral Control based on Vehicle Dynamics. In Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality (pp. 111-114).
- [13] Devineau, G., Polack, P., Althé, F., & Moutarde, F. (2018, November). Coupled longitudinal and lateral control of a vehicle using deep learning. In 2018 21st Int. Conf. on Intelligent Transportation Systems (pp. 642-649). IEEE.
- [14] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- [15] J.M. Ibarra Zannatha, P. Vera Bustamante, A. Cureño Ramírez, R. de la Rosa Picazo. Generación de Trayectorias para Vehículos Autónomos. CoMRob 2018: XX Congreso Mexicano de Robótica de la AMRob. CICESE, Ensenada, BC. Del 12 al 14 de octubre de 2018. Pag. 176-182.
- [16] E. Rivera Morales, D. Lechuga Rosales, P. Vera Bustamante, M. Villafuerte Bante, J.M. Ibarra Zannatha. Autonomous Vehicle's Instrumentation. CoMRob 2019, XXI Congreso Nacional de Robótica de la AMRob, Manzanillo, Col., 13 - 15 noviembre de 2019.
- [17] O. González Miranda, S.M. Orozco Soto, J.M. Ibarra Zannatha. Potential field-based preview control of an autonomous vehicle using visual feedback. CoMRob 2019, XXI Congreso Nacional de Robótica de la AMRob, Manzanillo, Col., 13 - 15 noviembre de 2019.
- [17] O. González Miranda, S.M. Orozco Soto, JM. Ibarra Zannatha. Control basado en campos potenciales para un vehículo autónomo usando realimentación visual. (ISSN 1870-4069). pp. 253-262; rec. 2019-03-21; acc. 2019-04-25 *Research in Computing Science* 148(8), 2019.
- [19] O. González Miranda. Modelado y Control de un vehículo autónomo. Tesis de Maestría en Ciencias en Control Automático. Cinvestav. 17 de diciembre de 2019.
- [20] Federación Mexicana de Robótica A.C. Federación mexicana de robótica. <https://femexrobotica.org/>, 2019. [Online; accessed 27-February-2020].
- [21] Freie Universität Berlin. Autonomos model. <https://github.com/AutoModelCar/AutoModelCarWiki/wiki>, 2016. [Online; accessed 27-February-2020].
- [22] Corke, P. (2017). *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised* (Vol. 118). Springer.

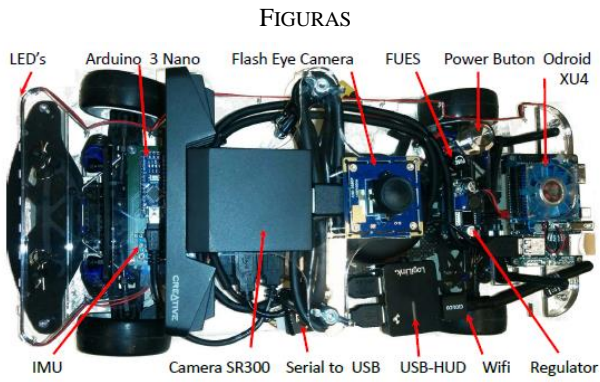


Fig. 1. El AutoNOMOS de la Universidad Libre de Berlín

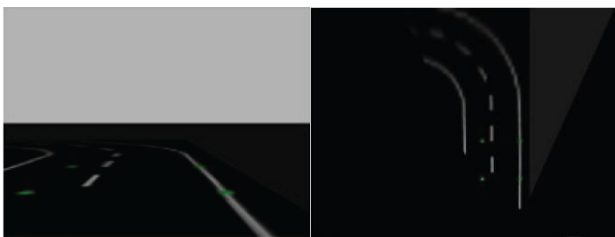


Fig. 2. a) Imagen de la carretera proporcionada por la cámara. b) Reconstrucción 3D

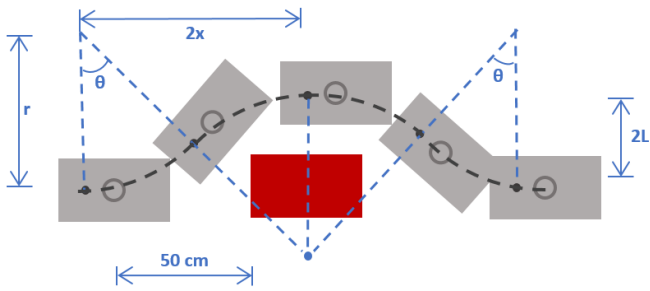


Fig. 3. Diagrama de la maniobra de rebase

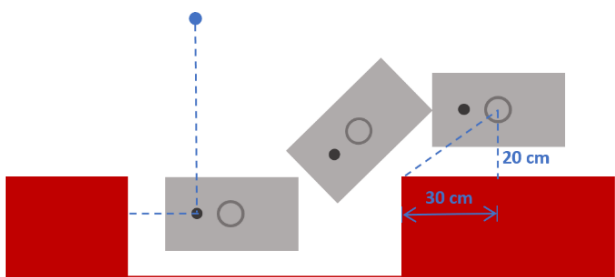


Fig. 6. Diagrama de la maniobra de estacionamiento

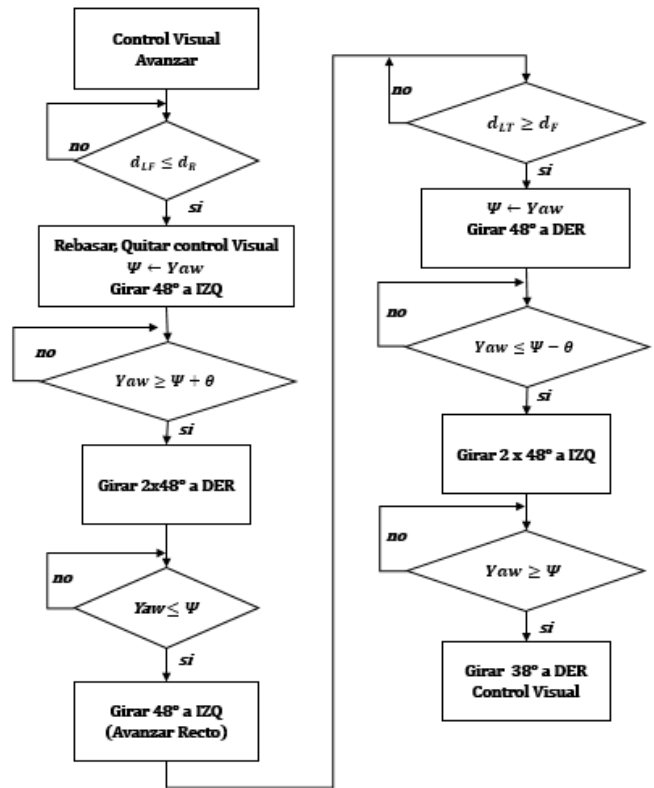


Fig. 4. Algoritmo de control para realizar la maniobra de rebase

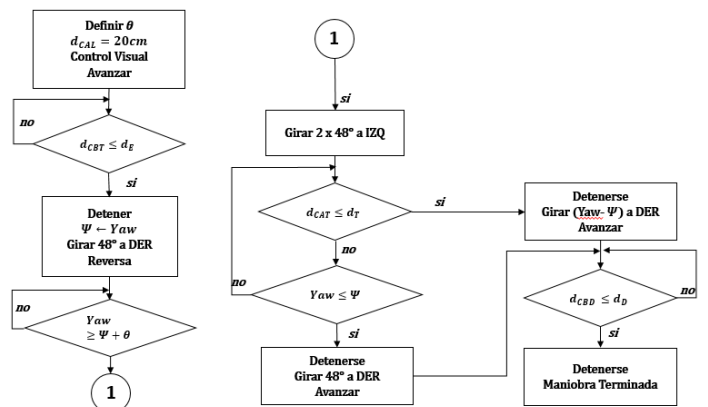


Fig. 7. Algoritmo de control para realizar la maniobra de estacionamiento

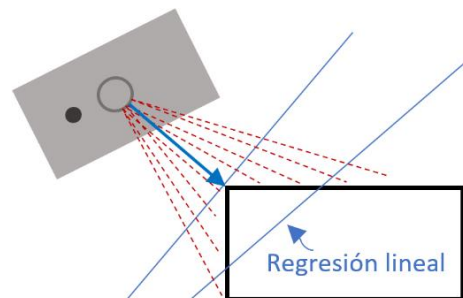


Fig. 8. Obstáculo detectado por el Lidar y recta equivalente.

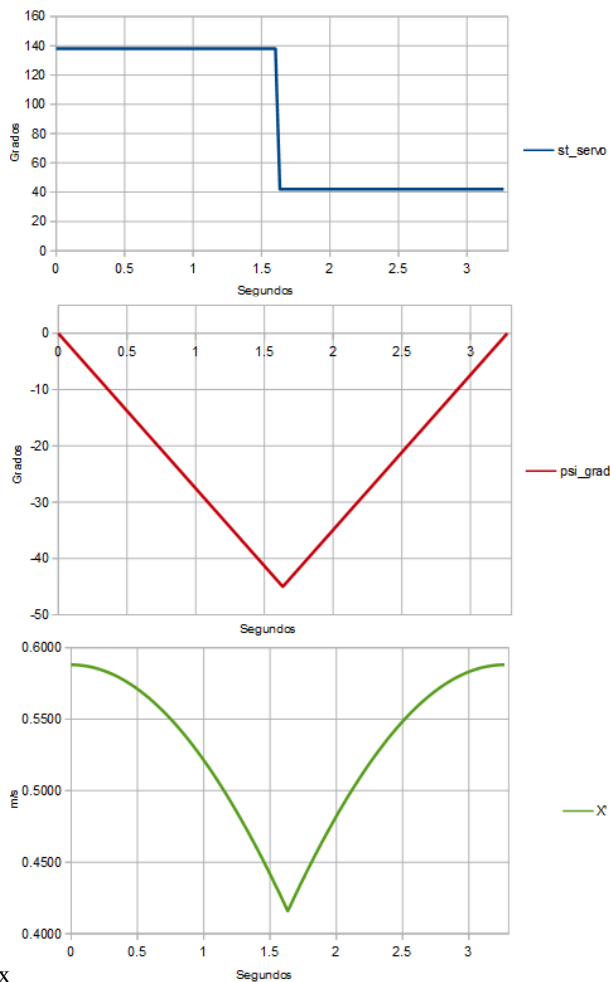


Fig. 5. Cambio en el tiempo del ángulo de dirección st (en azul), de la dirección ψ (en rojo) y de la velocidad en la dirección X (en verde), del automóvil.

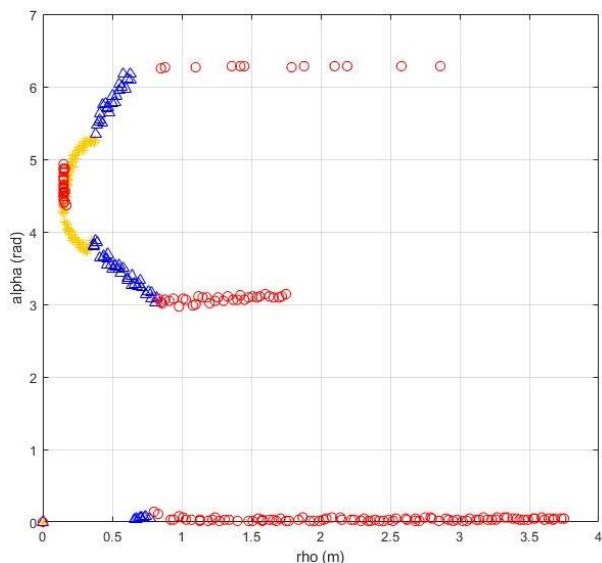


Fig. 9. Espacio de Clasificación (ρ_{\min}, α)

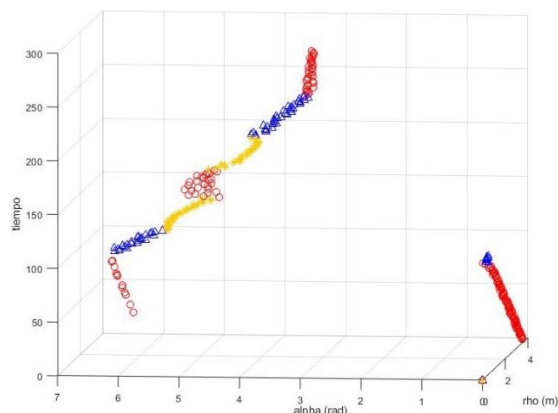


Fig. 10. Espacio de Clasificación (ρ_{\min}, α, t)

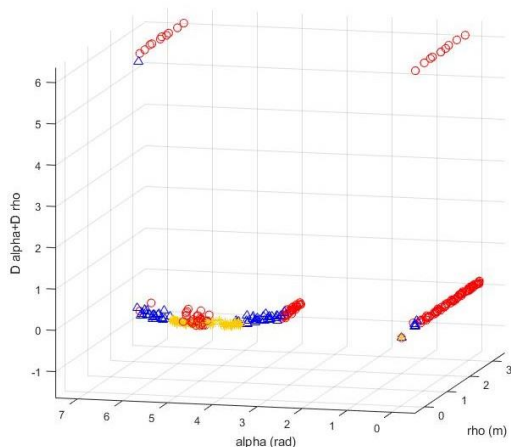


Fig. 11. Espacio de Clasificación $(\rho_{\min}, \alpha, \Delta\rho_{\min}+\Delta\alpha)$

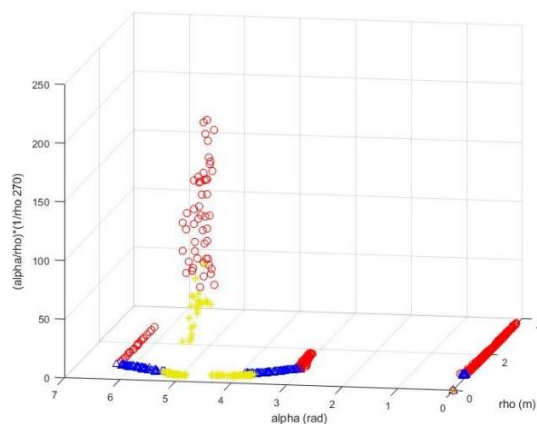


Fig. 12. Espacio de Clasificación $(\rho_{\min}, \alpha, (\alpha/\rho)*(1/\rho^{270}))$

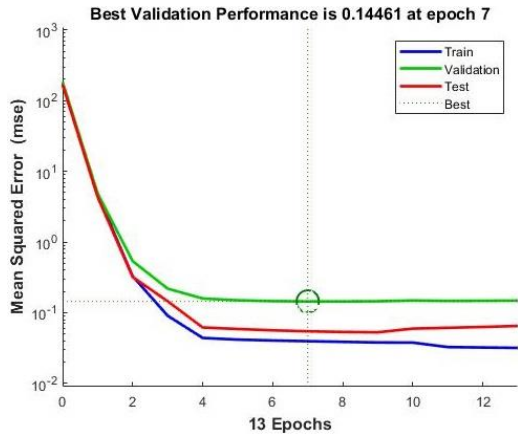


Fig. 13. Cambio del MSE de la red del CC visual, durante el entrenamiento.

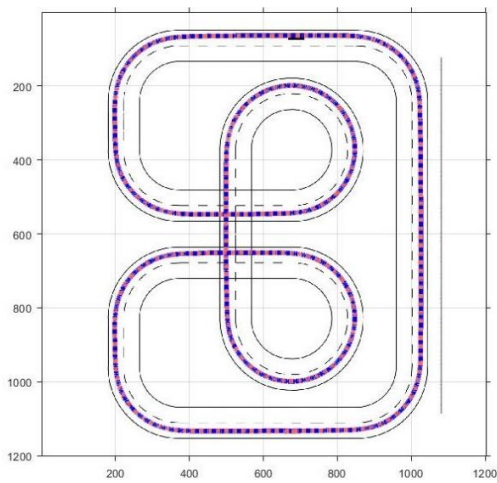


Fig 14. Pista utilizada en el TMR mostrando la trayectoria con el Controlador Óptimo y con el Controlador Conexionista para conducción visual autónoma

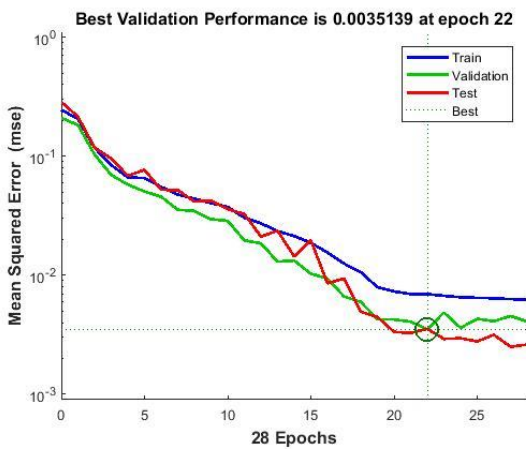


Fig. 15. Cambio del MSE del clasificador usado en la maniobra de rebase.

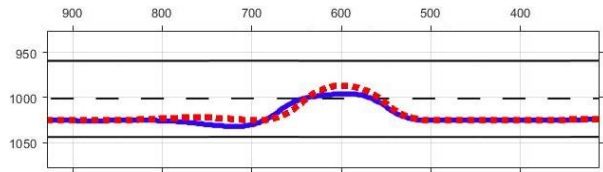


Fig. 16. Trayectoria de rebase: caso ideal con el controlador en lazo abierto (en azul) y con el Controlador Conexionista (en rojo).

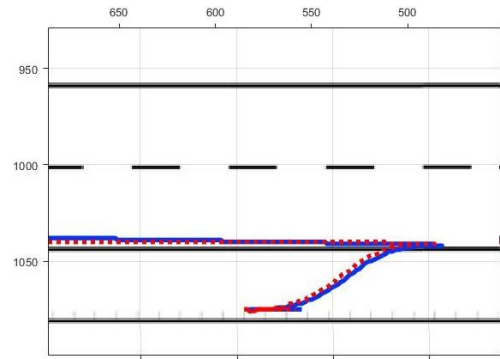


Fig. 17. Trayectoria de la maniobra de estacionamiento: caso ideal con el controlador en lazo abierto (en azul) y con el Controlador Conexionista (en rojo).

Estudio comparativo de servomotores de CD orientados a la construcción de prototipos educativos

Olga L. Jiménez, Jessica Maldonado, Rubén Garrido

Departamento de Control Automático.

CINVESTAV-IPN

Ciudad de México, México

Email: {ojimenez, jmaldonado, garrido}@ctrl.cinvestav.mx

Resumen—En este trabajo se presenta un estudio comparativo de tres servomotores de CD con el fin de que éstos sirvan como actuadores en la construcción de prototipos de enseñanza de bajo costo dentro del Control Automático, la Robótica y la Mecatrónica. Dos de ellos están dotados de sensores de posición de tipo codificador óptico, y un tercero posee un potenciómetro. Uno de los servomotores corresponden a la marca *LEGO*, modelo *EV3*, un segundo servomotor es de la marca *MakeBlock* y un tercero corresponde a un Servomotor Analógico de Radio Control (SARC) genérico. Se describen sus características técnicas, disponibilidad, costo así como también su desempeño cuando se implementa en ellos el controlador Proporcional Integral Derivativo (PID) sintonizado mediante técnicas basadas en el Regulador Cuadrático Lineal, comúnmente conocido como Linear Quadratic Regulator (LQR). Además, los parámetros de un modelo de segundo orden que describe la dinámica de estos servomotores se identifican mediante el algoritmo de Mínimos Cuadrados. Los experimentos reportados permiten apreciar aspectos no observados en simulaciones numéricas como lo son perturbaciones, fricción y ruidos de medición, entre otros. Los resultados obtenidos en el estudio comparativo ponen en evidencia las ventajas y las desventajas de cada uno de los servomotores de CD estudiados.

I. INTRODUCCIÓN

La evaluación en tiempo real de algoritmos de control desempeña un papel importante en la enseñanza y la investigación en las áreas de Control Automático, Robótica y Mecatrónica. Lo anterior permite al estudiante apreciar aspectos no observados en simulaciones numéricas como pueden ser dinámicas no modeladas, perturbaciones, incertidumbres y ruidos de medición, entre otros. La evaluación se realiza empleando prototipos de laboratorio los cuales están equipados con fuentes de alimentación, actuadores, sensores y son controlados mediante una plataforma experimental de control en tiempo real.

En la literatura existen varios prototipos comerciales orientados a la enseñanza. En las referencias [1] y [2] se describe un sistema marca QUANSER (QET por sus siglas en inglés). Este consiste de un motor de Corriente Directa (CD) equipado con un codificador óptico y un controlador integrado QUANSER (QIC) el cual contiene un microcontrolador y los periféricos necesarios para controlar al motor. En [3] se presenta un prototipo educativo el cual consta de siete módulos mecánicos

lineales y no lineales diseñados para demostrar la influencia del amortiguamiento, la elasticidad y la fricción. Este incluye un motor de CD el cual tiene acoplado directamente un taco-generator que genera señales de voltaje proporcional a la velocidad y un codificador óptico que mide la posición angular. En [4] se presenta el diseño de un sistema de control de movimiento rotacional el cual consiste de dos discos accionados por un motor de CD que se acoplan mediante bandas dentadas y poleas.

Sin embargo, el alto costo de los prototipos comerciales mencionados anteriormente ha motivado que educadores e investigadores desarrollen plataformas educativas a un costo accesible que les permita ser reproducidos en gran número. La referencia [5] presenta experimentos utilizando un prototipo de bajo costo el cual consta de un motor de imán permanente controlado mediante un microcontrolador. En el trabajo presentado en [6] se describe el desarrollo de un prototipo de laboratorio para la prueba de algoritmos de control el cual pueda ser asequible a estudiantes de nivel licenciatura. El prototipo consta de una tarjeta Raspberry Pi, un motor de CD y varios circuitos necesarios para las prácticas de laboratorio. En [7] se presenta el desarrollo de un laboratorio de aprendizaje avanzado para un motor de CD que permite a los estudiantes realizar experimentos de manera remota. Las diferencias más importantes entre los prototipos educativos existentes en la literatura y los presentados en el resumen corresponden a sus componentes, es decir, el tipo de motor de CD empleado y el tipo de sensores que requieren para obtener las señales de medición.

El objetivo de este trabajo es presentar un estudio comparativo de tres tipos de servomotores de CD de bajo costo: Un servomotor Analógico de Radio Control (SARC), un servomotor de la marca *MakeBlock* y un servomotor de la marca *LEGO* modelo *EV3*, con el fin de determinar sus diferencias tanto en costo como en su desempeño al ser controlados en lazo cerrado, con el propósito de construir un prototipo educativo de bajo costo que sea asequible para estudiantes de licenciatura y posgrado.

El artículo se divide de la siguiente manera: En la sección II se presentan las características de cada uno de los servo-

motores estudiados, en la sección III se encuentra el modelo matemático correspondiente a un servomotor de CD. En la sección IV se muestra el algoritmo de identificación paramétrica del modelo de un servomotor de CD. En la sección V se describe el algoritmo de control PID y la técnica de sintonización mediante un Regulador Cuadrático Lineal. La sección VI da detalles de la plataforma experimental utilizada para la aplicación del algoritmo de control PID y se presentan los resultados experimentales obtenidos. En la sección VII se presenta un análisis del desempeño del controlador PID aplicado a cada uno de los servomotores de CD estudiados. Finalmente se tienen las conclusiones del estudio comparativo realizado.

II. DESCRIPCIÓN DE LOS SERVOMOTORES DE CD

A. Servomotor marca MakeBlock

Este servomotor posee un codificador óptico que permite la medición de la posición angular del motor. Cuenta con un eje de salida de acero especial que permite acoplar directamente engranes piñones y correas síncronas, entre otros (Fig. 1). Está equipado con una caja de engranajes hecha de un material anti-abrasivo, lo cual se traduce en un menor ruido acústico durante su funcionamiento. En la Tabla I se muestran sus especificaciones técnicas.

B. Servomotor marca LEGO EV3

El servomotor *LEGO* modelo *EV3* mostrado en la Fig. 2 incluye un codificador óptico acoplado al eje de salida del motor a través de un engrane (ver Fig. 3) el cual permite la medición de la posición angular del motor. La Tabla II muestra las características técnicas de este servomotor. En lo sucesivo este motor se denominará simplemente *LEGO*.

C. Servomotor Analógico de Radio Control (SARC)

El Servomotor Analógico de Radio Control (SARC) marca Hitec modelo HS-485HB (Fig. 4) está constituido por un motor de CD, un juego de engranes, un potenciómetro conectado a este eje de salida y que es utilizado para medir la posición, y una tarjeta de control [8]. Para propósitos del estudio comparativo el servomotor es modificado eliminando su tarjeta de control (Fig. 5), ésto con el objetivo de tener acceso al potenciómetro y al motor de CD por separado. Dicha tarjeta es sustituida por dispositivos externos que incluyen un amplificador de potencia y un controlador programable. La Tabla III muestra las especificaciones técnicas del SARC.

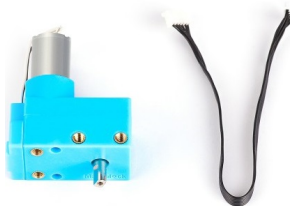


Fig. 1. Servomotor marca *MakeBlock*.

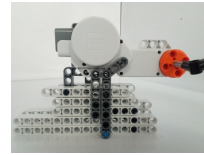


Fig. 2. Servomotor marca *LEGO* modelo *EV3* (vista lateral).

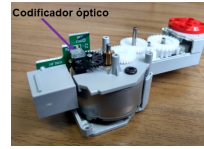


Fig. 3. Vista interior del servomotor *LEGO*.



Fig. 4. SARC HS-485HB.

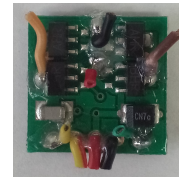


Fig. 5. Tarjeta de control del SARC HS-485HB.

TABLA I
ESPECIFICACIONES TÉCNICAS DEL SERVOMOTOR MARCA *MakeBlock* [9].

Voltaje nominal	7.4 V
Corriente de consumo sin carga	240 mA
Velocidad de rotación	178 ± 10 RPM
Par de arranque	5 kg·cm
Realimentación	codificador óptico
Rotación	ilimitada
Resolución del codificador	360 ppr
Peso	0.0615 kg

TABLA II
ESPECIFICACIONES TÉCNICAS DEL SERVOMOTOR MARCA *LEGO EV3* [10].

Voltaje nominal	9 V
Corriente de consumo sin carga	60 mA
Velocidad de rotación	170 RPM
Par de arranque	5.1 kg·cm
Realimentación	codificador óptico
Rotación	ilimitada
Resolución del codificador	180 ppr
Peso	0.080 kg

TABLA III
 ESPECIFICACIONES TÉCNICAS DEL SARC HS-485HB [11].

Rango de Voltaje nominal	4.8 V-6.0 V
Corriente de consumo sin carga (4.8V-6.0V)	150mA-180mA
Velocidad de rotación (4.8V-6.0V)	0.22-0.18 sec/60°
Par de arranque (4.8V-6.0 V)	4.8-6.0 kg·cm
Realimentación	Potenciómetro de 5 KΩ
Rotación	limitada
Rango de Movimiento	180°
Peso	0.0451 kg

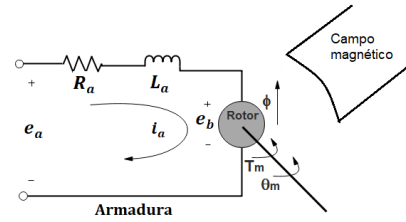


Fig. 6. Diagrama de un motor de CD de imán permanente.

III. MODELO MATEMÁTICO DE LOS SERVOMOTORES DE CD

En la Fig. 6 se muestra un esquema de un servomotor de CD de imán permanente. La armadura de éste se modela como un circuito con resistencia R_a conectada en serie a una inductancia L_a y a una fuente de voltaje e_b que representa la fuerza contraelectromotriz cuando el rotor gira. El campo magnético generado por un imán permanente se representa mediante ϕ . Las variables y parámetros del servomotor se definen en la Tabla IV.

La función de transferencia entre el desplazamiento angular del eje del servomotor $\Theta_m(s)$ y el voltaje de entrada $E_a(s)$, está descrita por:

$$\frac{\Theta_m(s)}{E_a(s)} = \frac{k_m}{s(L_a s + R_a)(J_m s + b_m) + k_c k_m s} \quad (1)$$

Además, se considera al amplificador que suministra el voltaje de entrada que alimenta al servomotor. En este caso, la relación entre el voltaje $E_a(s)$ de salida del amplificador y que alimenta al servomotor y el voltaje de control $U(s)$ aplicado al servomotor está dada por $E_a(s) = K_a U(s)$. Notar que el amplificador únicamente amplifica en voltaje y corriente la señal de control. Tomando en cuenta lo anterior, suponiendo que la inductancia es mucho más pequeña que la resistencia de armadura y considerando la relación n del sistema de engranajes que acopla mecánicamente al servomotor a la carga (ver Fig. 7), se tiene que la función de transferencia simplificada está dada por [12]:

$$\frac{Y(s)}{U(s)} = G(s) = \frac{b}{s(s+a)} \quad (2)$$

en donde:

$$Y(s) = n\Theta_m(s) \quad (3)$$

$$b = \frac{K_a k_m n}{R_a J_{eq}}$$

$$a = \frac{b_{eq} + \frac{k_m k_c}{R_a}}{J_{eq}}$$

donde

$$J_{eq} = J_m + n^2 J_L \quad y$$

$$b_{eq} = b_m + n^2 B_L.$$

 TABLA IV
 VARIABLES Y PARÁMETROS DE UN MOTOR DE CD.

Variables	Descripción (Unidad de medida)
T_m	Par electromagnético del motor (N-m)
ϕ	Flujo magnético (Wb)
i_a	Corriente de armadura (A)
L_a	Inductancia (H)
R_a	Resistencia (Ω)
k_m	Constante de par (N-m/A)
k_c	Constante de la fuerza contraelectromotriz (V-s/rad)
e_b	Fuerza contraelectromotriz (V)
e_a	Voltaje de alimentación (V)
u	Señal de control (V)
ω_m	Velocidad angular (rad/s)
θ_m	Desplazamiento angular del motor (rad).
J_m	Momento de inercia del motor ($\text{kg}\cdot\text{m}^2$)
b_m	Coefficiente de fricción viscosa del motor (N-m-s)
J_L	Momento de inercia de la carga ($\text{kg}\cdot\text{m}^2$).
B_L	Coefficiente de fricción viscosa de la carga (N-m-s)
K_a	Ganancia del amplificador de potencia (\bullet).
n	Radio de reducción de la caja de engranajes (\bullet).

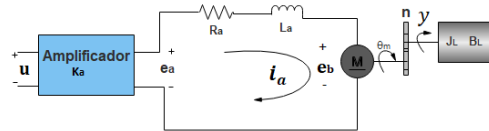


Fig. 7. Representación esquemática de un motor de CD acoplado a una carga inercial.

IV. IDENTIFICACIÓN PARAMÉTRICA

En el área de Control Automático la identificación de parámetros es un aspecto primordial para el diseño de algoritmos de control basados en un modelo de la planta a controlar. La identificación trata de inferir un modelo del sistema a partir de las variables medidas tales como entradas o variables de control, salidas o variables controladas y posiblemente perturbaciones. En la literatura existen varios métodos de identificación paramétrica [13]–[17], dentro de éstos se encuentra el algoritmo de Mínimos Cuadrados.

A. Identificación paramétrica de un servomotor de CD

Considérese el modelo matemático de un servomotor de CD (2) en su forma de ecuación diferencial:

$$\ddot{y} = -a\dot{y} + bu \quad (4)$$

Se puede apreciar que este modelo tiene un polo en el origen del plano complejo y por lo tanto el sistema es marginalmente estable. Esta característica dificulta su identificación en lazo abierto razón por la cual es necesario estimar sus parámetros en lazo cerrado. Se propone el empleo de un algoritmo de control Proporcional con Realimentación Taquimétrica (PRT). Este controlador es capaz de estabilizar al servomotor y la sintonización de sus ganancias se puede realizar sin el conocimiento de los parámetros de este último. La ley de control PRT utilizada está dada por la ecuación:

$$u = k_p e - k_d \dot{y}_{ef} \quad (5)$$

donde k_p representa la ganancia proporcional a la señal del error, que está definido por $e = r - y$, con r como señal de referencia, y la salida en posición angular, y k_d es la ganancia derivativa, ambas ganancias son constantes reales positivas. El término \dot{y}_{ef} es un estimado de la velocidad \dot{y} y es obtenido mediante el filtrado de la posición angular y utilizando un filtro (6) compuesto por un filtro pasa altas en cascada con un filtro pasa bajas.

$$y \rightarrow \left[\frac{f_{01}s}{s + f_{01}} \right] \left[\frac{f_{02}}{s + f_{02}} \right] \rightarrow \dot{y}_{ef} \quad (6)$$

con f_{01} y f_{02} constantes reales positivas.

Para aplicar el método de Mínimos Cuadrados al modelo (4) del servomotor son necesarias las mediciones de la señal de control u , la velocidad angular \dot{y} y la aceleración angular \ddot{y} en instantes de tiempo diferentes para formar un sistema sobredeterminado. Sin embargo, sólo se tiene acceso a la posición y obtenida a partir de mediciones. Para solucionar este problema se hace uso de filtros con la finalidad de obtener una expresión que contenga los parámetros desconocidos pero esta vez asociados a señales disponibles [15].

Se define el filtro:

$$F(s) = \frac{f_2}{s^2 + f_1s + f_2} \quad (7)$$

con f_1 y f_2 reales positivos.

Aplicando el filtro al modelo (2) se obtiene:

$$\begin{aligned} F(s)(s^2 + as)Y(s) &= F(s)bU(s) \\ s^2F(s)Y(s) + asF(s)Y(s) &= bF(s)U(s) \end{aligned} \quad (8)$$

De (8) se definen:

$$\begin{aligned} Y_f(s) &= F(s)Y(s) \\ U_f(s) &= F(s)U(s) \end{aligned}$$

en consecuencia la ecuación (8) se reescribe como:

$$(s^2 + as)Y_f(s) = bU_f(s) \quad (9)$$

Aplicando la transformada inversa de Laplace a (9) produce:

$$\ddot{y}_f + a\dot{y}_f = bu_f \quad (10)$$

donde $\dot{y}_f = \mathcal{L}^{-1}\{sY_f(s)\}$, $\ddot{y}_f = \mathcal{L}^{-1}\{s^2Y_f(s)\}$, $u_f = \mathcal{L}^{-1}\{U_f(s)\}$ y el operador $\mathcal{L}^{-1}\{\bullet\}$ corresponde a la transformada inversa de Laplace.

Los filtros empleados para generar las variables u_f , \dot{y}_f y \ddot{y}_f son los siguientes:

$$\begin{aligned} y &\rightarrow \frac{-f_2s}{s^2 + f_1s + f_2} \rightarrow -\dot{y}_f \\ y &\rightarrow \frac{f_2s^2}{s^2 + f_1s + f_2} \rightarrow \ddot{y}_f \\ u &\rightarrow \frac{f_2}{s^2 + f_1s + f_2} \rightarrow u_f \end{aligned} \quad (11)$$

El modelo filtrado dado por (10) permite obtener una regresión lineal en función de señales disponibles, esto es:

$$z(t) = \phi^T(t)\theta \quad (12)$$

donde:

$$\begin{aligned} z(t) &= \ddot{y}_f \\ \phi(t) &= \begin{bmatrix} -\dot{y}_f \\ u_f \end{bmatrix}, \quad \theta = \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \end{aligned}$$

el vector $\phi(t)$ es un vector de funciones conocidas y $\theta \in \mathbb{R}^2$ es un vector de parámetros constantes desconocidos.

B. Algoritmo de Mínimos Cuadrados fuera de línea

La ecuación (12) es válida para cualquier tiempo t , por lo tanto también es válida para los instantes de tiempo $T, 2T, \dots, (k-1)T, kT$, donde T es el periodo de muestreo. Lo anterior permite escribirla como:

$$z(k) = \phi^T(k)\theta \quad (13)$$

en donde se ha eliminado el periodo de muestreo T para mantener la notación simple.

Se realizan mediciones de la posición y y de u en instantes de tiempo correspondientes a los periodos de muestreo $1, 2, \dots, (k-1), k$, y mediante los filtros (11) se obtienen los correspondientes valores de \dot{y}_f , \ddot{y}_f y u_f . Estos permiten formar el siguiente sistema de ecuaciones sobredeterminado:

$$\begin{aligned} -\dot{y}_f(1)\theta_1 + u_f(1)\theta_2 &= \ddot{y}_f(1) \\ -\dot{y}_f(2)\theta_1 + u_f(2)\theta_2 &= \ddot{y}_f(2) \\ &\vdots \\ -\dot{y}_f(k)\theta_1 + u_f(k)\theta_2 &= \ddot{y}_f(k) \end{aligned}$$

que puede expresarse como:

$$\begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ \vdots \\ \phi^T(k) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} z(1) \\ z(2) \\ \vdots \\ z(k) \end{bmatrix}$$

De lo anterior se define el siguiente sistema de ecuaciones sobredeterminado:

$$A\theta = Y \quad (14)$$

donde

$$A = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ \vdots \\ \phi^T(k) \end{bmatrix} \in \mathbb{R}^{n \times 2}; \quad Y = \begin{bmatrix} z(1) \\ z(2) \\ \vdots \\ z(k) \end{bmatrix} \in \mathbb{R}^{n \times 1}$$

Finalmente, la solución en sentido de Mínimos Cuadrados para el sistema (14) corresponde a [18]:

$$\hat{\theta} = (A^T A)^{-1} A^T Y \quad (15)$$

La condición necesaria para que se lleve a cabo la identificación paramétrica es que la inversa de la matriz $(A^T A)$ exista. Lo anterior se logra si la señal de excitación cumple con la condición de excitación persistente [14]. Ejemplos de señales de este tipo son las secuencias binarias pseudoaleatorias y el ruido blanco filtrado.

V. CONTROLADOR PID SINTONIZADO MEDIANTE TÉCNICAS BASADAS EN EL REGULADOR CUADRÁTICO LINEAL

El controlador PID es una de las leyes de control más utilizadas tanto en la estabilización y regulación de procesos industriales como en la enseñanza de la Teoría de Control debido a su simplicidad de sintonización e implementación, así como a su efectividad.

La expresión matemática que define al control PID está dada por [19]:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt} \quad (16)$$

donde k_p representa la ganancia proporcional al error definido por $e = r - y$, con r una señal de referencia, y es la salida, k_i es la ganancia integral y k_d es la ganancia derivativa. En el caso de los servomotores y corresponde a la posición angular.

Para evitar problemas en la acción derivativa cuando se produce un cambio brusco de la señal de referencia r , se propone el empleo de la siguiente ley de control PID modificada:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau - k_d \dot{y} \quad (17)$$

En la implementación de este controlador, el término \dot{y} se reemplaza por su estimado \dot{y}_{ef} obtenido mediante el filtro (6).

Las ganancias del controlador PID fueron obtenidas mediante la teoría del Regulador Cuadrático Lineal. Dicha técnica utiliza la representación en variables de estado, por lo tanto, la ley de control (17) escrita como una realimentación de estado está dada por:

$$u = -KX \quad (18)$$

en donde el estado del sistema X es:

$$X = \begin{bmatrix} e \\ \int_0^t e(\tau) d\tau \\ -\dot{y} \end{bmatrix} \quad (19)$$

y:

$$K = \begin{bmatrix} k_p \\ k_i \\ k_d \end{bmatrix}$$

A. Representación en variables de estado del modelo del servomotor

La representación en variables de estado del servomotor se lleva a partir del modelo (4) y de la dinámica del error e :

$$\begin{aligned} e &= r - y \\ \dot{e} &= -\dot{y} \\ \ddot{e} &= -\ddot{y} \\ \ddot{e} &= -a\dot{e} - bu \end{aligned} \quad (20)$$

$$\begin{aligned} x_1 &= e & \dot{x}_1 &= -\dot{y} = x_3 \\ x_2 &= \int_0^t e(\tau) d\tau & \dot{x}_2 &= e = x_1 \\ x_3 &= \dot{e} = -\dot{y} & \dot{x}_3 &= -ax_3 - bu \end{aligned} \quad (21)$$

el cual tiene la siguiente escritura alternativa:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (22)$$

donde:

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & -a \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ -b \end{bmatrix} \quad C = [1 \quad 0 \quad 0]$$

B. Sintonización del controlador PID

La teoría del Regulador Cuadrático Lineal permite calcular las ganancias K de realimentación de estado minimizando el criterio de desempeño:

$$J = \int_0^{\infty} (X^T Q X + U^T R U) \quad (23)$$

en donde $Q \geq 0$ y $R > 0$ son matrices de ponderación. La matriz Q y la matriz R ponderan respectivamente la dinámica del estado X y el gasto de la energía correspondiente a la señal de control U .

La minimización del criterio de desempeño J corresponde a la solución $S = S^T > 0$ de la ecuación de Ricatti:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \quad (24)$$

la cual permite calcular la ganancia de realimentación de estado K a partir de la siguiente ecuación:

$$K = R^{-1}B^T S \quad (25)$$

Cabe mencionar que el programa Matlab dispone de un comando para obtener la ganancia K , es decir:

$$[K, S, E] = lqr(A, B, Q, R) \quad (26)$$

en donde S representa la solución de la ecuación de Ricatti (24) y E es un vector cuyas entradas corresponden a los polos del sistema en lazo cerrado.

Para el caso del servomotor el cual es un sistema una entrada-una salida, se cumple que $Q \in \mathfrak{R}^{3 \times 3}$ y $R \in \mathfrak{R}$. De lo anterior se define a la matriz Q como:

$$Q = \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{bmatrix}$$

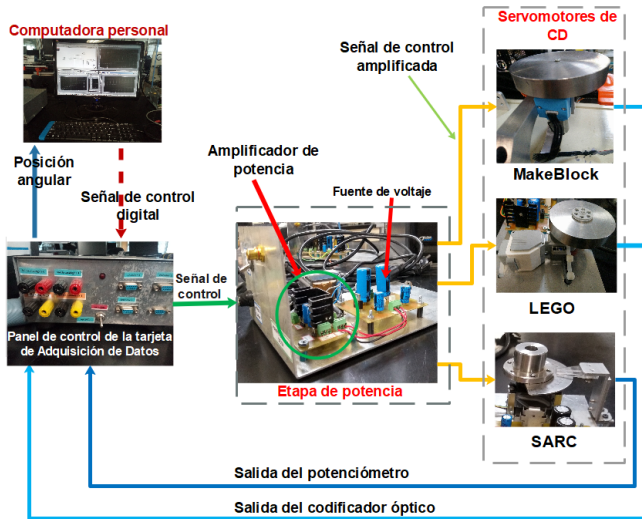


Fig. 8. Plataforma experimental utilizada para el control de los servomotores de CD.

VI. EXPERIMENTOS

A. Plataforma experimental

La plataforma experimental utilizada para evaluar a los servomotores, ilustrada en la Fig. 8, consta de una computadora personal equipada de una tarjeta de adquisición de datos marca *Sensoray* modelo 626. Los algoritmos de control se codificaron en la plataforma de programación MATLAB/SIMULINK bajo el ambiente en tiempo real *QUARC* de *Quanser Consulting* con un tiempo de muestreo de 1 ms y el método de integración Euler01. La salida de la señal de control producida por la tarjeta de adquisición de datos se procesa a través de una etapa de potencia para accionar a los servomotores. Dicha etapa de potencia compuesta por un amplificador lineal *National Semiconductor* modelo LM675 [20]. De acuerdo al diagrama eléctrico de la implementación del amplificador de potencia presentado en la Fig. 9, su ganancia es igual a $K_a=21.36$.

Se utiliza como referencia una señal de onda cuadrada con una amplitud de 0.8 rad (45°) y una frecuencia de 0.1 Hz, filtrada mediante un filtro pasa bajas de primer orden con frecuencia de corte de 10 rad/s. Cabe mencionar que cada uno de los servomotores impulsan una carga inercial.

B. Resultados experimentales

De acuerdo al algoritmo de Mínimos Cuadrados (15), los filtros utilizados para estimar las señales \dot{y} , \ddot{y} y para filtrar la señal de control u fueron los siguientes:

$$y \rightarrow \frac{-400s}{s^2 + 40s + 400} \rightarrow -\dot{y}_f$$

$$y \rightarrow \frac{400s^2}{s^2 + 40s + 400} \rightarrow \ddot{y}_f$$

$$u \rightarrow \frac{400}{s^2 + 40s + 400} \rightarrow u_f$$

Se utiliza como señal de excitación ruido blanco procesado mediante un filtro pasa bajas de primer orden de frecuencia de

corte de 10 rad/s. Los parámetros estimados de los modelos de los servomotores empleados en el estudio comparativo se presentan en la Tabla V al igual que las ganancias del controlador PRT (5) empleadas.

En la Tabla VI se presentan las ganancias de realimentación del controlador PID obtenidas a partir de la metodología del Regulador Cuadrático Lineal. Las matrices de ponderación se escogieron en cada caso para lograr el error de posicionamiento más pequeño correspondiente a cada servomotor. El filtro (6) utilizado para obtener estimaciones de la velocidad angular de los servomotores está dado por:

$$y \rightarrow \left[\frac{160s}{s + 160} \right] \left[\frac{160}{s + 160} \right] \rightarrow \dot{y}_f \quad (27)$$

En las Fig. 10, 11 y 12 se muestran la salida en posición, la señal de control y el error de posición respectivamente del servomotor *Makeblock*. Las Fig. 13, 14 y 15 muestran la salida en posición, la señal de control y el error de posición respectivamente del servomotor *LEGO*. Finalmente, en las Fig. 16, 17, 18 se muestran la salida en posición, la señal de control y el error de posición respectivamente del *SARC*. Cabe resaltar que la señal de control presentada en las Fig. 11, 14 y 17 es la señal medida antes del amplificador de potencia.

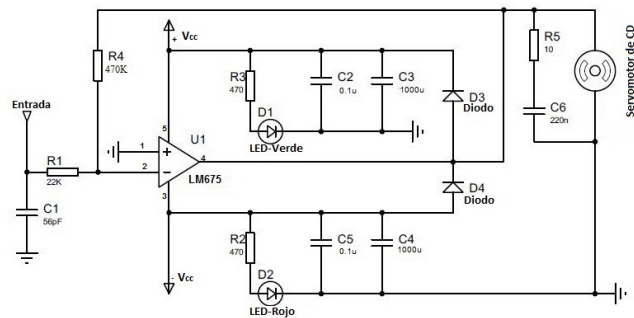


Fig. 9. Diagrama eléctrico del amplificador de potencia para alimentar a los servomotores de CD [21].

TABLA V
PARÁMETROS ESTIMADOS CORRESPONDIENTES A LOS MODELOS DE LOS SERVOMOTORES DE CD Y GANANCIAS DEL CONTROLADOR PRT (5).

Servomotor	Controlador		Parámetros estimados	
	k_p	k_d	a	b
<i>MakeBlock</i>	1.75	0,074	20.3415	135.6939
<i>LEGO</i>	2	0,1	17.3852	76.2582
<i>SARC</i>	1,72	0,006	13.2776	200.1332

TABLA VI
GANANCIAS DEL CONTROLADOR PID (17) CALCULADAS MEDIANTE LA TEORÍA DEL REGULADOR CUADRÁTICO LINEAL.

Servomotor	Q	R	k_p	k_i	k_d
<i>MakeBlock</i>	diag(170,0.02,0.0001)	1	13.043	0.141	0.313
<i>LEGO</i>	diag(370,0.07,0.001)	1.1	18.350	0.252	0.502
<i>SARC</i>	diag(150,0.05,0.005)	45	1.828	0.033	0.084

VII. ANÁLISIS DE RESULTADOS

Para medir el desempeño del controlador PID (17) aplicado a los servomotores de CD se utilizaron los criterios de desempeño siguientes: La Integral del Error Cuadrático (IEC), la Integral del Valor Absoluto del Control (IAC) y la Integral de la Derivada del Valor Absoluto del Control (IDAC). Estos índices están expresados matemáticamente como sigue:

$$IEC = \int_{T_1}^{T_2} k[e(t)]^2 dt \quad (28)$$

$$IAC = \int_{T_1}^{T_2} |u(t)| dt \quad (29)$$

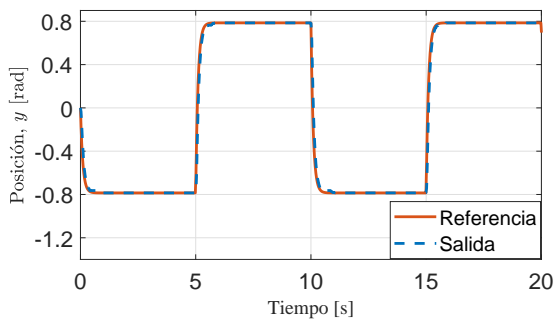


Fig. 10. Referencia r vs salida en posición y del servomotor *Makeblock*.

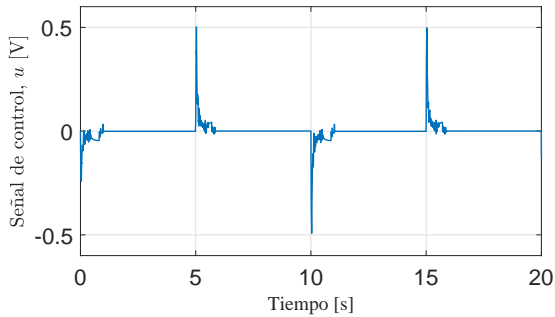


Fig. 11. Señal del controlador PID aplicado al servomotor *MakeBlock*.

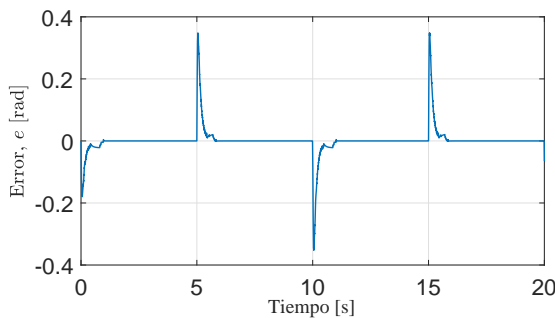


Fig. 12. Error en posición e del servomotor *MakeBlock*.

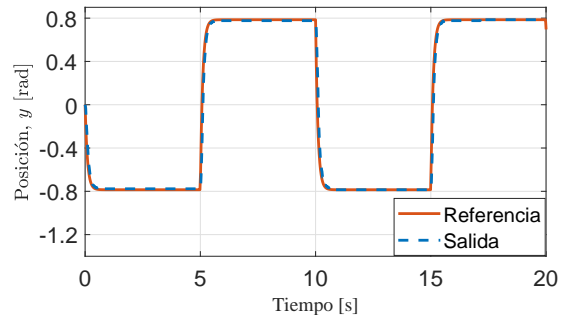


Fig. 13. Referencia r vs salida en posición y del servomotor *LEGO*.

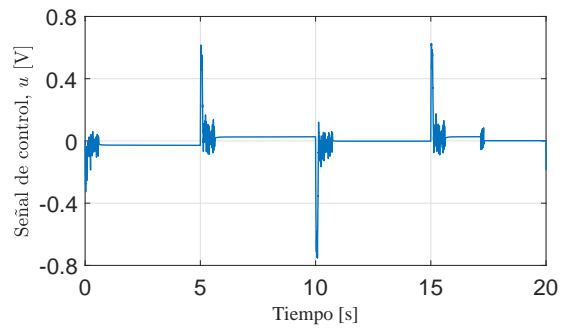


Fig. 14. Señal del controlador PID aplicado al servomotor *LEGO*.

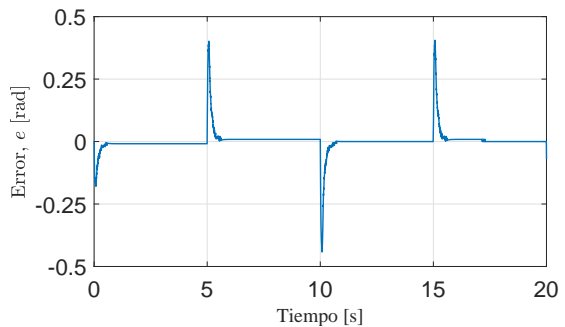


Fig. 15. Error en posición e del servomotor *LEGO*.

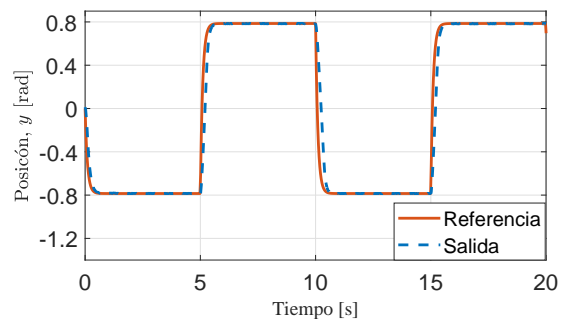


Fig. 16. Referencia r vs salida en posición y del *SARC*.

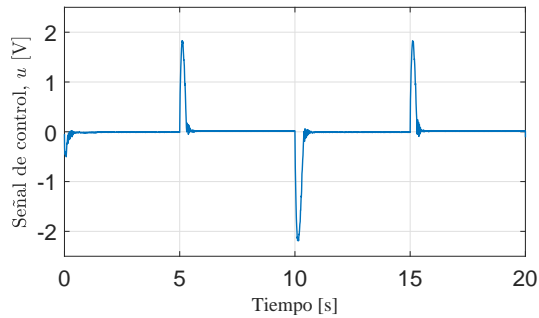
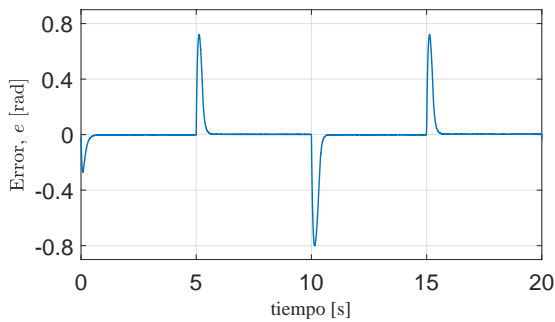


Fig. 17. Señal del controlador PID aplicado al SARC.


 Fig. 18. Error en posición e del SARC.

$$IDAC = \int_{T_1}^{T_2} \left| \frac{du(t)}{dt} \right| dt \quad (30)$$

en donde k representa un factor de escalamiento y $\{T_1, T_2\}$ definen un intervalo de tiempo durante el cual se calculan los índices de desempeño. Para el estudio comparativo se utiliza un valor de $k = 100$ con intervalos de tiempo $T_2 - T_1 = 5$ s.

En la Tabla VII se muestra el desempeño del controlador PID (17) en cada caso. Se puede observar que de los dos servomotores dotados de codificadores ópticos, los índices IEC e IAC más pequeños los produce el servomotor *MakeBlock* y el índice IDAC más pequeño lo produce el servomotor *LEGO*. Notar también que el SARC exhibe el desempeño más pobre. En particular, el índice de desempeño IDAC para el SARC, el cual mide la variación de la señal de control es el más elevado. Lo anterior puede ser el resultado de los altos niveles de ruido de medición que produce el potenciómetro que mide su posición angular. Por otro lado, los valores de los índices de desempeño pueden cambiar con una sintonización diferente de los controladores, por lo que los valores reportados se deben considerar únicamente como ejemplos y es posible que el desempeño mejore con otras sintonizaciones. Sin embargo, el empleo de un potenciómetro como sensor de posición afecta considerablemente el desempeño del SARC comparado con los servomotores *MakeBlock* y *LEGO* que emplean un decodificador óptico.

La Tabla VIII muestra el estudio comparativo de los servomotores de CD empleados en este trabajo respecto a costos,

 TABLA VII
DESEMPEÑO DEL CONTROLADOR PID.

Servomotor	IEC	IAC	IDAC
<i>MakeBlock</i>	0.1043	0.02986	0.6166
<i>LEGO</i>	0.1497	0.0893	0.1197
<i>SARC</i>	1.5724	0.3489	3.8505

 TABLA VIII
TABLA COMPARATIVA DE COSTOS EN PESOS MEXICANOS (MXN), VOLTAJE DE ALIMENTACIÓN, TIPO DE SENSOR EN POSICIÓN Y DISPONIBILIDAD DE LOS SERVOMOTORES DE CD EN EL MERCADO NACIONAL.

Servomotor	Costo	Voltaje	Sensor	Disponibilidad
<i>MakeBlock</i>	\$399.99	7.8 V	Codificador óptico	Difícilmente disponible
<i>LEGO</i>	\$549.00	9 V	Codificador óptico	Disponible
<i>SARC</i>	\$330.00	5 V	Potenciómetro	Fácilmente disponible

voltaje de alimentación, tipo de sensor de posición y su disponibilidad en el mercado nacional. El servomotor marca *LEGO* se pueden obtener a través de proveedores nacionales, el servomotor *MakeBlock* sólo se puede conseguir, al menos por el momento, a través de su importación. Los servomotores de Radio Control son de fácil adquisición y se encuentran ampliamente disponibles en el mercado nacional.

VIII. CONCLUSIONES

De acuerdo al estudio comparativo realizado en este trabajo se obtuvieron las siguientes conclusiones sobre las ventajas y desventajas de cada uno de los servomotores de CD.

Los sensores ópticos que poseen los servomotores marca *LEGO* y *MakeBlock* representan una ventaja con respecto al potenciómetro utilizado en el Servomotor Analógico de Radio Control (*SARC*), ya que los primeros no poseen ruido en las señales de medición. En este sentido el potenciómetro es un sensor con elevados niveles de ruido de medición. Además, el empleo de los codificadores ópticos permiten que el giro de la salida de un servomotor sea ilimitado a diferencia del *SARC*, el cual tiene un giro máximo menor a 360° debido al potenciómetro. Por otro lado, una ventaja en los *SARC* es el costo el cual es más bajo y se encuentran ampliamente disponibles en México. De acuerdo al trabajo desarrollado, se concluye que la elección de un servomotor de CD para la construcción de prototipos depende de la disponibilidad de los servomotores, de su costo y del tipo de sensor de posición que emplean. También se observó que es posible implementar sin dificultades particulares el algoritmo de control Proporcional Integral Derivativo (PID).

Es interesante mencionar que, a diferencia de los servomotores de CD dotados de un codificador óptico, el servomotor de Radio Control (*SARC*) tiene la característica de tener señales de voltaje tanto en la entrada como en la salida. Esto posibilita el empleo de controladores basados en circuitos analógicos para su control, aspecto de posible utilidad para que los

estudiantes empleen conocimientos de electrónica analógica en el desarrollo de prácticas de laboratorio.

AGRADECIMIENTOS.

Los autores desean agradecer a Gerardo Castro y Jesús Meza por su ayuda en la puesta en funcionamiento de los prototipos de laboratorio y el sistema de adquisición de datos usado en los experimentos. Olga Lidia Jiménez y Jessica Maldonado agradecen el apoyo otorgado por CONACyT mediante dos becas de posgrado.

REFERENCES

- [1] J. Apkarian and K. J. Åström. A laptop servo for control education. *IEEE Control Systems Magazine*, 24(5):70–73, 2004.
- [2] D. S. Bernstein. The quanser dc motor control trainer individual or team learning for hands-on control education - product review. *IEEE Control Systems Magazine*, 25(3):90–93, 2005.
- [3] Inteco. *Modular Servo System*. Inteco, ul. Katowicka 3631-351 KrakowPoland, pci edition, 2016.
- [4] Educational Products Company. Model 220 industrial plant emulator. http://www.ecpsystems.com/docs/ECP_Ind_Emulator_Model_220.pdf, 1995-2000. Consultado: Agosto 2019.
- [5] M. Gunasekaran and R. Potluri. Low cost undergraduate control systems experiments using microcontroller based control of a dc motor. *IEEE Transactions on Education*, 55(4):508–516, 2012.
- [6] R. M. Reck and R. S. Sreenivas. Developing a new affordable dc motor laboratory kit for an existing undergraduate controls course. In *2015 American Control Conference (ACC)*, pages 2801–2806. IEEE, 2015.
- [7] I. Colak, S. Demirbas, S. Sagioglu, and E. Irmak. A novel web-based laboratory for dc motor experiments. *Computer Applications in Engineering Education*, 19(1):125–135, 2011.
- [8] C. U. Onate and J. K. Molina. Characterization, simulation and implementation of a new dynamic model for a dc servomotor. *IEEE Latin America Transactions*, 12(6):997–1004, 2014.
- [9] Makeblock. 180 optical encoder motor. <https://store.makeblock.com/180-optical-encoder-motor>, 2019. Consultado: Septiembre 2018.
- [10] LEGO. 9v-technic-motors-compared-characteristics. <http://www.philohome.com/motors/motorcomp.htm>, 2006. Consultado: Septiembre 2018.
- [11] ServoDatabase.com. Hitec hs-485hb. <https://servodatabase.com/servo/hitec/hs-485hb>. Consultado: Septiembre 2018.
- [12] Katsuhiko Ogata. *Ingeniería de Control Moderna*. Pearson Education, S.A., Madrid, 2010.
- [13] R Vallejo et al. Identificación paramétrica de sistemas dinámicos. *Revista Científica Ingeniería y Desarrollo*, (2):10–22, 2011.
- [14] Petros A. Ioannou Jing Sun. *Robust Adaptive Control*. Prentice Hall, 1995.
- [15] Shankar Sastry. *Adaptive Control; Stability, Convergence and Robustness*. Prentice Hall Information and System Sciences Series, 1989.
- [16] L. Ljung. System identification. *Wiley Encyclopedia of Electrical and Electronics Engineering*, pages 1–19, 1999.
- [17] B. Wittenmark K. J. Åström. *Adaptive Control*. Addison Esley, 1989.
- [18] Rolf Isermann and Marco Münchhof. *Identification of dynamic systems: an introduction with applications*. Springer Science & Business Media, 2010.
- [19] K. Johan Åström and C. Canudas De Wit. *Revisiting the LuGre friction model*. IEEE Control systems, 2008.
- [20] Texas-Instruments. Lm675-power-operational-amplifier. <http://www.ti.com/lit/ds/symlink/lm675.pdf>, May 1999. Consultado: Octubre 2018.
- [21] J. J. Maldonado and R. Garrido. Prototipo educativo basado en un servomotor analógico de radio control. In *Memorias del Congreso Nacional de Control Automático*, pages 569–574, 2017.

Control Robusto a Perturbaciones Basado en Redes Neuronales para Seguimiento de Trayectorias de Quadrotores

Ivan Lopez-Sanchez¹, Francisco Rossomando², Ricardo Pérez-Alcocer³,
Carlos Soria², Ricardo Carelli² y Javier Moreno-Valenzuela¹

Resumen—Los quadrotores han mostrado una gran versatilidad de aplicaciones tanto en interiores como en exteriores. La mayoría de éstas se desarrollan principalmente en exteriores, exponiendo al vehículo a perturbaciones. Aunado a esto, algunos de los parámetros del sistema son desconocidos o difíciles de obtener. En este trabajo se presenta un esquema de control no lineal, constituido por un control por modos deslizantes y una red neuronal de regresión generalizada. Este esquema de control es aplicado a un quadrotor que ya posee un controlador embebido, lo que conduce a una simplificación del modelo. Se presenta una comparación en simulación del esquema de control propuesto con respecto a un controlador basado en modos deslizantes. Los resultados de esta comparación muestran la efectividad de la utilización de la red neuronal para el rechazo de perturbaciones, además de producir errores de seguimiento más pequeños.

I. INTRODUCCIÓN

Desde su primera aparición en la década de los sesenta del siglo pasado, los vehículos aéreos no tripulados de ala rotatoria han cambiado considerablemente, desde su configuración geométrica, materiales, instrumentación hasta sus aplicaciones. Inicialmente fueron concebidos para aplicaciones militares. Sin embargo, hoy en día son utilizados también como juguetes [1]. Dentro de la clasificación de vehículos aéreos no tripulados (UAV, por sus siglas en inglés) de ala rotatoria, los más utilizados son los quadrotores, debido a su estructura simple y a su manejo intuitivo. A pesar de su estructura simple, su representación matemática no es trivial, por lo que diversos documentos y trabajos se han presentado abordando el modelo dinámico de este vehículo con fines de identificación y control [2]–[7].

Como resultado de la versatilidad de este vehículo, muchas aplicaciones se han encontrado para ellos, como el monitoreo de vida silvestre [8], planeación urbana e inspección de estructuras [9]–[11], búsqueda de personas en zonas de desastre [12] y vigilancia aérea [13]. Las aplicaciones de los quadrotores son en exteriores, donde las condiciones climáticas pueden llegar a ser adversas y cambiantes. A esto, se le agrega el hecho de que estos vehículos son sistemas no lineales y subactuados, ya que poseen 6 grados de libertad

y sólo cuatro actuadores [14]. Por estas razones surge la necesidad de desarrollar controladores capaces de mantener el vehículo estable mientras se transporta de un punto a otro y de sobrellevar los cambios dentro de su espacio de trabajo.

Tal y como se describe en [15]–[19], la red neuronal de regresión generalizada es una red neuronal de un solo paso con una estructura con alto grado de paralelismo. Puede ser utilizada para resolver problemas de regresión o estimación donde no se puede suponer que el sistema es lineal. Una aplicación muy útil para esta red neuronal es la de identificar la dinámica de una planta para fines de control [16].

También se han encontrado diversas aplicaciones para esta red neuronal en el área de procesamiento de imágenes o incluso la predicción del consumo energético en alguna región. En [17] se implementó una red neuronal de regresión generalizada en conjunto con un algoritmo genético de optimización con el propósito de identificar objetos en 3D a partir de poses en 2D del objeto y para el reconocimiento de dígitos escritos a mano. Un algoritmo de optimización basado en enjambre de moscas de fruta y una red neuronal de regresión generalizada fueron usados para obtener un modelo para la previsión de la carga energética anual en [18]. También se ha implementado con fines de control en vehículos aéreos no tripulados. En [19] se utilizó una red neuronal de regresión generalizada que requería entrenamiento para controlar la altura de un vehículo aéreo no tripulado con alas capaces de aletear.

Se han presentado diversos trabajos con el objetivo de proveer de soluciones a los problemas que conlleva el control de quadrotores utilizando los sensores inerciales embebidos en el quadrotor, sistemas de visión y cámaras [20]–[22]. En [23] se presentó un controlador robusto a perturbaciones externas basado en la técnica de backstepping y modos deslizantes en conjunto con una red neuronal utilizando una función de base radial para la tarea de regulación de posición de un quadrotor. Un controlador adaptable basado en redes neuronales con funciones de base radial fue propuesto en [24] para transportar una carga suspendida por un cable. En [25] se propuso un controlador basado en la técnica de control adaptable por modelo de referencia para controlar la orientación y la altitud de un quadrotor, donde el modelo de referencia es controlado por un controlador PID. Utilizando la técnica de control por modos deslizantes adaptable y capa límite adaptable, en [26] se propuso un esquema de control robusto a fallas en los actuadores e incertidumbre paramétrica. En [27], se presentó un controlador adaptable por modos deslizantes para estabilizar un quadrotor

¹Adscritos al Instituto Politécnico Nacional–CITEDI, Av. Instituto Politécnico Nacional 1310, Mesa de Otay, Tijuana, B.C., México (ilopez@citedi.mx, moreno@citedi.mx).

²Adscritos al Instituto de Automática–CONICET–Universidad Nacional de San Juan, Av. San Martín, San Juan 1109, Argentina (frosoma@inaut.unsj.edu.ar, csoria@inaut.unsj.edu.ar, rcarelli@inaut.unsj.edu.ar).

³Adscrito al CONACYT–Instituto Politécnico Nacional–CITEDI, Av. Instituto Politécnico Nacional 1310, Mesa de Otay, Tijuana, B.C., México (rrperez@citedi.mx).

bajo el efecto de perturbaciones externas e incertidumbre paramétrica.

En este trabajo se propone un esquema de control robusto a perturbaciones e incertidumbre paramétrica basado en modos deslizantes y en una red neuronal de regresión generalizada. El esquema propuesto es aplicado a un quadrotor que ya cuenta con un controlador embebido, lo que conduce a un modelo simplificado que tiene como entrada señales normalizadas de pose y velocidad y tiene como salida la posición y el ángulo de *yaw*. La red neuronal es una parte importante del esquema propuesto para el rechazo de perturbaciones y compensación de la dinámica no modelada. La propuesta es respaldada por una comparación en simulación donde se prueba la funcionalidad del esquema y la utilidad de la red neuronal.

El resto del documento está organizado de la siguiente manera: En la sección II se presenta el modelo dinámico de un quadrotor y el controlador embebido. El esquema de control propuesto y la red neuronal se describen en la sección III. Los resultados de las simulaciones numéricas se presentan en la sección IV. Finalmente, en la sección V se presentan las conclusiones de este trabajo.

II. MODELO MATEMÁTICO DE UN QUADROTOR

El modelo dinámico de un quadrotor representado en el marco de referencia inercial está dado de la siguiente manera de acuerdo a lo discutido en [3], [5],

$$m\ddot{\mathbf{p}} + mg\mathbf{e}_z + D_p\dot{\mathbf{p}} = R(\boldsymbol{\eta})\mathbf{e}_z F, \quad (1)$$

$$M(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} = W(\boldsymbol{\eta})^{-T}\boldsymbol{\tau}, \quad (2)$$

donde la ecuación (1) representa la dinámica de posición y la ecuación (2) representa la dinámica de orientación, m es la masa del quadrotor, g es la constante de aceleración gravitacional, $\mathbf{p} = [x \ y \ z]^T \in \mathbb{R}^3$ es la posición del quadrotor, $\boldsymbol{\eta} = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3$ representa la orientación del quadrotor en el marco de referencia inercial, $\mathbf{e}_z = [0 \ 0 \ 1] \in \mathbb{R}^3$ es un vector unitario en dirección del eje z en el marco de referencia inercial, $R(\boldsymbol{\eta}) \in \mathbb{R}^{3 \times 3}$ es una matriz de rotación, $M(\boldsymbol{\eta}) \in \mathbb{R}^{3 \times 3}$ es la matriz de inercia, $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \in \mathbb{R}^{3 \times 3}$ es la matriz de Coriolis, $D_p \in \mathbb{R}^{3 \times 3}$ es una matriz diagonal con los coeficientes de arrastre aerodinámico, $W(\boldsymbol{\eta}) \in \mathbb{R}^{3 \times 3}$ es una matriz de transformación y $F \in \mathbb{R}$ y $\boldsymbol{\tau} \in \mathbb{R}^3$ son las entradas de control.

El modelo de un quadrotor representado por las ecuaciones (1) y (2) es un modelo que no contempla los efectos aerodinámicos a los cuales el quadrotor está sujeto durante su operación. De acuerdo a lo discutido en [28]–[31], diversos efectos aerodinámicos pueden ser contemplados tales como la influencia del ángulo de ataque en el empuje de los rotores. Sin embargo, esta clase de efectos aerodinámicos son más significativos en vuelos de alta velocidad y en maniobras acrobáticas. No obstante, uno de los efectos aerodinámicos que se puede considerar es la fuerza de arrastre inducida por el viento, tal efecto es incluido en el modelo del quadrotor en (1) por medio del término $D_p\dot{\mathbf{p}}$.

Según lo discutido en [32]–[35], bajo el supuesto de la presencia de un controlador embebido en el quadrotor,

capaz de estabilizarlo y mantenerlo en vuelo estacionario, es posible suponer que las entradas del quadrotor son las entradas a dicho controlador embebido. En los trabajos [32]–[35] se hace la suposición de que dicho controlador embebido es dado por

$$F = \frac{m}{c_\phi c_\theta} (g + \dot{z}^*), \quad (3)$$

$$\boldsymbol{\tau} = W(\boldsymbol{\eta})^T [M(\boldsymbol{\eta})\ddot{\boldsymbol{\tau}} + C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\tau}}], \quad (4)$$

donde

$$\dot{z}^* = \frac{1}{\tau_z} (\dot{z}_{\text{ref}} - \dot{z}), \quad (5)$$

$$\begin{bmatrix} \ddot{\tau}_\phi \\ \ddot{\tau}_\theta \\ \ddot{\tau}_\psi \end{bmatrix} = \begin{bmatrix} \omega_\phi^2 (\phi_{\text{ref}} - \phi) - 2\xi_\phi \omega_\phi \dot{\phi} \\ \omega_\theta^2 (\theta_{\text{ref}} - \theta) - 2\xi_\theta \omega_\theta \dot{\theta} \\ \frac{1}{\tau_\psi} (\dot{\psi}_{\text{ref}} - \dot{\psi}) \end{bmatrix}, \quad (6)$$

que en relación a los sistemas lineales de primer y segundo orden, ω_ϕ y ω_θ representan la frecuencia natural no amortiguada, ξ_ϕ y ξ_θ representan las constantes de amortiguamiento y τ_z y τ_ψ son constantes de tiempo.

Nótese que las señales de referencia \dot{z}_{ref} , ϕ_{ref} , θ_{ref} y $\dot{\psi}_{\text{ref}}$ son comandos para el control embebido, y son definidos de la siguiente manera

$$\dot{z}_{\text{ref}} = k_z \dot{z}_{\text{max}} u_z, \quad (7)$$

$$\theta_{\text{ref}} = k_\theta \theta_{\text{max}} u_\theta, \quad (8)$$

$$\phi_{\text{ref}} = k_\phi \phi_{\text{max}} u_\phi, \quad (9)$$

$$\dot{\psi}_{\text{ref}} = k_\psi \dot{\psi}_{\text{max}} u_\psi, \quad (10)$$

que al ser sustituidos en (5) y (6) se obtiene

$$\dot{z}^* = \frac{k_z \dot{z}_{\text{max}}}{\tau_z} u_z - \frac{1}{\tau_z} \dot{z},$$

$$\begin{bmatrix} \ddot{\tau}_\phi \\ \ddot{\tau}_\theta \\ \ddot{\tau}_\psi \end{bmatrix} = \begin{bmatrix} k_\phi \omega_\phi^2 \phi_{\text{max}} u_\phi - 2\xi_\phi \omega_\phi \dot{\phi} - \omega_\phi^2 \phi \\ k_\theta \omega_\theta^2 \theta_{\text{max}} u_\theta - 2\xi_\theta \omega_\theta \dot{\theta} - \omega_\theta^2 \theta \\ \frac{k_\psi \dot{\psi}_{\text{max}}}{\tau_\psi} u_\psi - \frac{1}{\tau_\psi} \dot{\psi} \end{bmatrix}.$$

Según lo discutido en [32]–[35], las constantes \dot{z}_{max} , ϕ_{max} , θ_{max} y $\dot{\psi}_{\text{max}} > 0$ son útiles para ajustar las unidades de las ecuaciones (7)–(10) en el caso de que \dot{z}_{ref} , ϕ_{ref} , θ_{ref} y $\dot{\psi}_{\text{ref}}$ estén normalizadas y sean adimensionales.

De esta forma, al sustituir las ecuaciones (3)–(10) en las ecuaciones (1) y (2) se obtiene

$$\ddot{x} = \frac{F}{m} (s_\psi s_\phi + c_\psi c_\phi s_\theta) - d_x \dot{x}, \quad (11)$$

$$\ddot{y} = \frac{F}{m} (-c_\psi s_\phi + s_\psi c_\phi s_\theta) - d_y \dot{y}, \quad (12)$$

$$\ddot{z} = \frac{k_z \dot{z}_{\text{max}}}{\tau_z} u_z - \left(\frac{1}{\tau_z} + \frac{d_z}{m} \right) \dot{z},$$

$$\ddot{\phi} = k_\phi \omega_\phi^2 \phi_{\text{max}} u_\phi - 2\xi_\phi \omega_\phi \dot{\phi} - \omega_\phi^2 \phi,$$

$$\ddot{\theta} = k_\theta \omega_\theta^2 \theta_{\text{max}} u_\theta - 2\xi_\theta \omega_\theta \dot{\theta} - \omega_\theta^2 \theta,$$

$$\ddot{\psi} = \frac{k_\psi \dot{\psi}_{\text{max}}}{\tau_\psi} u_\psi - \frac{1}{\tau_\psi} \dot{\psi}.$$

Linealizando las ecuaciones (11) y (12) (que describen la dinámica de posición en el plano (x, y)) alrededor del punto

de operación $\psi = \text{constante}$, $\theta = \phi = 0$ y $F = mg$, que corresponde al vuelo estacionario, se obtiene

$$\ddot{x} = g(c_\psi\theta + s_\psi\phi) - d_x\dot{x}, \quad (13)$$

$$\ddot{y} = g(s_\psi\theta - c_\psi\phi) - d_y\dot{y}. \quad (14)$$

Sumando y restando $g(c_\psi\theta + s_\psi\phi)$ y $g(s_\psi\theta - c_\psi\phi)$ a (13) y (14), respectivamente, definiendo el error de orientación de *pitch* y de *roll* como

$$\begin{aligned} \tilde{\theta} &= \theta_{\text{ref}} - \theta, \\ \tilde{\phi} &= \phi_{\text{ref}} - \phi, \end{aligned}$$

y realizando las manipulaciones algebraicas correspondientes, la dinámica del quadrotor se puede expresar como

$$\ddot{x} = c_\psi g k_\theta \theta_{\text{max}} u_\theta + s_\psi g k_\phi \phi_{\text{max}} u_\phi - d_x \dot{x} - g(c_\psi \tilde{\theta} + s_\psi \tilde{\phi}), \quad (15)$$

$$\ddot{y} = s_\psi g k_\theta \theta_{\text{max}} u_\theta - c_\psi g k_\phi \phi_{\text{max}} u_\phi - d_y \dot{y} - g(s_\psi \tilde{\theta} - c_\psi \tilde{\phi}), \quad (16)$$

$$\ddot{z} = \frac{k_z \dot{z}_{\text{max}}}{\tau_z} u_z - \left(\frac{1}{\tau_z} + \frac{d_z}{m} \right) \dot{z}, \quad (17)$$

$$\ddot{\theta} = -2\xi_\theta \omega_\theta \dot{\theta} + \omega_\theta^2 \tilde{\theta}, \quad (18)$$

$$\ddot{\phi} = -2\xi_\phi \omega_\phi \dot{\phi} + \omega_\phi^2 \tilde{\phi}, \quad (19)$$

$$\ddot{\psi} = \frac{k_\psi \dot{\psi}_{\text{max}}}{\tau_\psi} u_\psi - \frac{1}{\tau_\psi} \dot{\psi}. \quad (20)$$

Se supone de que el controlador (4) asegura que $\tilde{\theta}(t) \approx 0$ y $\tilde{\phi}(t) \approx 0$ para todo $t \geq 0$, entonces el modelo del quadrotor descrito en las ecuaciones (15)-(20) puede ser rescrito de forma matricial como en [20], [21], [36], [37]

$$\ddot{\mathbf{x}}^w = T(\psi) K_u \mathbf{u} - K_v \dot{\mathbf{x}}^w, \quad (21)$$

donde $\mathbf{x}^w = [x \ y \ z \ \psi]^T \in \mathbb{R}^4$ es un vector que contiene la posición en el espacio tridimensional (x, y, z) y el ángulo de *yaw* ψ con respecto del marco de referencia inercial (el superíndice w indica la relación al marco de referencia inercial), $K_u \in \mathbb{R}^{4 \times 4}$ y $K_v \in \mathbb{R}^{4 \times 4}$ son matrices diagonales y definidas positivas relacionadas a los parámetros del vehículo y del controlador embebido, descritas explícitamente como

$$K_u = \begin{bmatrix} g k_\theta \theta_{\text{max}} & 0 & 0 & 0 \\ 0 & g k_\phi \phi_{\text{max}} & 0 & 0 \\ 0 & 0 & \frac{k_z \dot{z}_{\text{max}}}{\tau_z} & 0 \\ 0 & 0 & 0 & \frac{k_\psi \dot{\psi}_{\text{max}}}{\tau_\psi} \end{bmatrix},$$

$$K_v = \begin{bmatrix} d_x & 0 & 0 & 0 \\ 0 & d_y & 0 & 0 \\ 0 & 0 & \left(\frac{1}{\tau_z} + \frac{d_z}{m} \right) & 0 \\ 0 & 0 & 0 & \frac{1}{\tau_\psi} \end{bmatrix}.$$

La matriz $T(\psi) \in \mathbb{R}^{4 \times 4}$ es una matriz de transformación definida explícitamente como

$$T(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 & 0 \\ \sin(\psi) & -\cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

y $\mathbf{u} = [u_\theta \ u_\phi \ u_z \ u_\psi] \in \mathbb{R}^4$ es el vector de entradas de control cuyo valor está normalizado y es adimensional, donde:

- u_θ es la entrada de posición angular relacionada al desplazamiento en el eje x del marco de referencia inercial.
- u_ϕ es la entrada de posición angular relacionada al desplazamiento en el eje y del marco de referencia inercial.
- u_z es la entrada de velocidad relacionada al movimiento en el eje z del marco de referencia inercial.
- u_ψ es la entrada de velocidad angular relacionada a la rotación sobre el eje z en el marco de referencia inercial.

El modelo en (21) representa el quadrotor en condiciones ideales. Sin embargo, para aplicaciones en exteriores y desplazamientos de alta velocidad, los parámetros del modelo pueden cambiar como consecuencia de las condiciones climáticas (ráfagas de viento, lluvia, cambios en la densidad del aire, etc.), o también, según la tarea a desempeñar, como por ejemplo, transportar una carga suspendida. Como resultado de lo antes mencionado, el modelo en (21) puede ser expresado como

$$\ddot{\mathbf{x}}^w = T(\psi) K_u(\mathbf{x}^w, \dot{\mathbf{x}}^w) \mathbf{u} - K_v(\mathbf{x}^w, \dot{\mathbf{x}}^w) \dot{\mathbf{x}}^w + \boldsymbol{\delta}(t), \quad (22)$$

donde $\boldsymbol{\delta}(t) \in \mathbb{R}^4$ representa el vector de perturbaciones externas y de dinámica no modelada, el cual puede ser acotado como

$$\|\boldsymbol{\delta}(t)\| \leq \delta_0,$$

donde δ_0 es una constante estrictamente positiva.

III. CONTROLADOR CON RECHAZO DE PERTURBACIONES

El controlador propuesto está compuesto por un esquema de control por modos deslizantes y una red neuronal de regresión generalizada. Considerando $\mathbf{x}_{\text{ref}}^w = [x_{\text{ref}} \ y_{\text{ref}} \ z_{\text{ref}} \ \psi_{\text{ref}}]^T \in \mathbb{R}^4$ como señales de referencia para la posición y el ángulo de *yaw*, el error se define como

$$\mathbf{e} = \mathbf{x}^w - \mathbf{x}_{\text{ref}}^w, \quad (23)$$

el cual define un error generalizado de posición.

Una superficie deslizante para un sistema de múltiples entradas múltiples salidas (MIMO, por sus siglas en inglés) se propone de manera siguiente [38]

$$\mathbf{r} = \dot{\mathbf{e}} + \alpha \mathbf{e}, \quad (24)$$

donde $\alpha \in \mathbb{R}^{4 \times 4}$ es una matriz de ganancias diagonal y definida positiva. Ahora, de la definición del error en (23) y la superficie deslizante (24), se obtiene la dinámica de la superficie deslizante y realizando algunas manipulaciones algebraicas, se obtiene la siguiente expresión

$$\ddot{\mathbf{x}}^w = \dot{\mathbf{r}} - \alpha(\mathbf{r} - \alpha \mathbf{e}) + \ddot{\mathbf{x}}_{\text{ref}}^w. \quad (25)$$

Sustituyendo (25) en (22) y despejando $\dot{\mathbf{r}}$, se obtiene

$$\dot{\mathbf{r}} = -K_v \dot{\mathbf{x}}^w + T(\psi) K_u \mathbf{u} + \boldsymbol{\delta}(t) + \alpha \mathbf{r} - \alpha^2 \mathbf{e} - \ddot{\mathbf{x}}_{\text{ref}}^w. \quad (26)$$

Considerando la expresión (26), se propone el siguiente controlador

$$\mathbf{u} = (T(\psi)K_u)^{-1}(\ddot{\mathbf{x}}_{\text{ref}}^w - b_1 \tanh(b_2 \mathbf{r}) - \hat{\boldsymbol{\delta}}(t) - K_r \text{sign}(\mathbf{r})), \quad (27)$$

donde, b_1 y b_2 son constantes estrictamente positivas, $K_r \in \mathbb{R}^{4 \times 4}$ es una matriz de ganancias diagonal y positiva definida, $\hat{\boldsymbol{\delta}}(t) \in \mathbb{R}^4$ es una estimación de las perturbaciones y de la dinámica no modelada. La estimación de estas perturbaciones es obtenida utilizando una red neuronal de regresión generalizada, que de acuerdo al teorema de aproximación universal [39], [40], puede ser utilizada para aproximar cualquier función continua de manera siguiente

$$\boldsymbol{\delta}(t) = \Phi^T \mathbf{H}(\bar{\mathbf{r}}, \mathbf{c}, \boldsymbol{\rho}) + \boldsymbol{\epsilon},$$

donde, $\bar{\mathbf{r}} = [1 \ \mathbf{r}^T \ \mathbf{e}^T \ \dot{\mathbf{x}}^w]^T \in \mathbb{R}^{13}$ es el vector extendido de entradas para la red neuronal, $\mathbf{H}(\bar{\mathbf{r}}, \mathbf{c}, \boldsymbol{\rho}) \in \mathbb{R}^m$ es el vector de funciones de activación basadas en una función de base radial definida como

$$H_i(\bar{\mathbf{r}}, \mathbf{c}, \boldsymbol{\rho}) = \frac{e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_i\|}{2\rho_i}}}{\sum_{j=1}^m e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_j\|}{2\rho_j}}}, \quad (28)$$

donde m es la cantidad de neuronas, el vector $\mathbf{c}_i \in \mathbb{R}^{13}$ es el vector de centros correspondiente a cada elemento de $\mathbf{H}(\bar{\mathbf{r}}, \mathbf{c}, \boldsymbol{\rho})$ de tal forma que el vector $\mathbf{c} \in \mathbb{R}^{13m}$ es un vector construido con los vectores \mathbf{c}_i ordenados de forma que $\mathbf{c} = [\mathbf{c}_1^T \ \mathbf{c}_2^T \ \dots \ \mathbf{c}_{13}^T]^T$, mientras que $\rho_i \in \mathbb{R}$ es la desviación o anchura de cada función de base radial $H_i(\bar{\mathbf{r}}, \mathbf{c}_i, \rho_i)$ en $\mathbf{H}(\bar{\mathbf{r}}, \mathbf{c}, \boldsymbol{\rho})$, por lo que $\boldsymbol{\rho} \in \mathbb{R}^m$, $\Phi \in \mathbb{R}^{m \times n}$ es la matriz de parámetros óptimos, los cuales son constantes y desconocidos, n es el número de salidas, que para este caso $n = 4$. El vector $\boldsymbol{\epsilon} \in \mathbb{R}^4$ es el error de aproximación, que puede ser acotado como $|\epsilon_i| \leq k_{r_i}$. Los elementos k_{r_i} corresponden a la matriz K_r presente en la ley de control (27) y ayudan a reducir los errores de aproximación. Según lo discutido en [15]–[19], [41], la red neuronal de regresión generalizada es una variación de la red neuronal de funciones de base radial, está compuesta por varias capas lo que la hace una red neuronal multicapa. En general se puede decir que consta de cuatro capas: capa de entrada, capa oculta, capa de suma y la capa de salida en una sola capa. Dentro de la capa oculta es donde se alojan las funciones de activación de base radial.

La función $\hat{\boldsymbol{\delta}}$ está dada entonces por una red neuronal de regresión generalizada, es decir,

$$\hat{\boldsymbol{\delta}}(t) = \hat{\Phi}^T \hat{\mathbf{H}}(\bar{\mathbf{r}}, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}). \quad (29)$$

En la Figura 1 se presenta un esquema de la arquitectura de la red neuronal de regresión generalizada de acuerdo a la ecuación (29). Los parámetros $\hat{\Phi}$, $\hat{\mathbf{c}}$ y $\hat{\boldsymbol{\rho}}$ son obtenidos por leyes de adaptación que se obtienen por medio de un análisis de convergencia del error $\mathbf{e}(t)$ que se omite y están

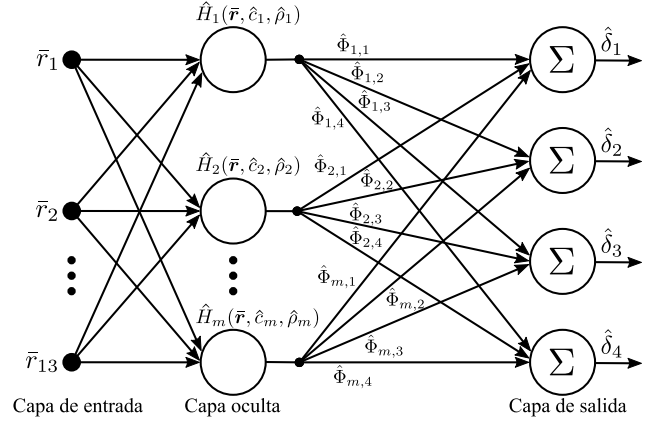


Fig. 1. Diagrama de la red neuronal de regresión generalizada para la estimación de $\hat{\boldsymbol{\delta}}$.

definidas como

$$\begin{aligned} \hat{\Phi}_i &= b_3 \hat{\mathbf{H}}(\bar{\mathbf{r}}, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}})(\tanh(b_2 \mathbf{r}_i)), \\ \hat{\mathbf{c}} &= b_4 \sum_{i=1}^n [\tanh(b_2 \mathbf{r}_i)(\Delta \hat{\mathbf{H}}_c^T \hat{\Phi}_i(t))], \\ \hat{\boldsymbol{\rho}} &= b_5 \sum_{i=1}^n [\tanh(b_2 \mathbf{r}_i)(\Delta \hat{\mathbf{H}}_\rho^T \hat{\Phi}_i(t))], \end{aligned}$$

donde $\Delta \hat{\mathbf{H}}_c \in \mathbb{R}^{m \times [m \dim(\bar{\mathbf{r}})]}$ y $\Delta \hat{\mathbf{H}}_\rho \in \mathbb{R}^{m \times m}$ representan los jacobianos de la función de activación (28) con respecto de los vectores de centros \mathbf{c} y de anchura $\boldsymbol{\rho}$, respectivamente, y se definen como

$$\Delta \hat{\mathbf{H}}_c = \left. \frac{\partial \mathbf{H}(\bar{\mathbf{r}}, \mathbf{c}, \boldsymbol{\rho})}{\partial \mathbf{c}} \right|_{\substack{\boldsymbol{\rho} = \hat{\boldsymbol{\rho}} \\ \mathbf{c} = \hat{\mathbf{c}}}}, \quad (30)$$

$$\Delta \hat{\mathbf{H}}_\rho = \left. \frac{\partial \mathbf{H}(\bar{\mathbf{r}}, \mathbf{c}, \boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \right|_{\substack{\boldsymbol{\rho} = \hat{\boldsymbol{\rho}} \\ \mathbf{c} = \hat{\mathbf{c}}}}. \quad (31)$$

Los elementos de las matrices jacobianas (30) y (31) están dados explícitamente como

$$\begin{aligned} \frac{\partial H_i(\bar{\mathbf{r}}, \mathbf{c}, \boldsymbol{\rho})}{\partial c_{ik}} &= \frac{e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_i\|}{2\rho_i}} \frac{\bar{\mathbf{r}}_k - c_{ik}}{2\rho_i \|\bar{\mathbf{r}} - \mathbf{c}_i\|}}{\left[\sum_{l=1}^m e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_l\|}{2\rho_l}} \right]^2} \\ &= \frac{e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_i\|}{2\rho_i}} \frac{\bar{\mathbf{r}}_k - c_{ik}}{2\rho_i \|\bar{\mathbf{r}} - \mathbf{c}_i\|} \sum_{l=1}^m e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_l\|}{2\rho_l}}}{\left[\sum_{l=1}^m e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_l\|}{2\rho_l}} \right]^2}, \\ \frac{\partial H_i(\bar{\mathbf{r}}, \mathbf{c}, \boldsymbol{\rho})}{\partial c_{jk}} &= \frac{e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_i\|}{2\rho_i}} \frac{\bar{\mathbf{r}}_k - c_{jk}}{2\rho_j \|\bar{\mathbf{r}} - \mathbf{c}_j\|} e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_j\|}{2\rho_j}}}{\left[\sum_{l=1}^m e^{-\frac{\|\bar{\mathbf{r}} - \mathbf{c}_l\|}{2\rho_l}} \right]^2}. \end{aligned}$$

En la Figura 2 se muestra el diagrama de bloques que describe la estructura del controlador propuesto y el controlador embebido en lazo cerrado con el quadrotor.

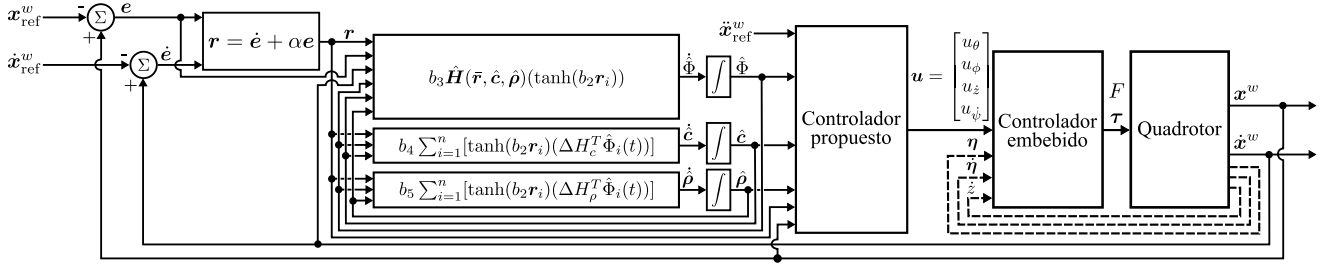


Fig. 2. Diagrama de bloques del controlador (27).

IV. RESULTADOS DE SIMULACIÓN

Para evaluar la funcionalidad del esquema de control propuesto, se han realizado simulaciones numéricas en Matlab-Simulink. Además, se presenta una comparación entre el controlador propuesto (27) y el controlador descrito por

$$u = (T(\psi)K_u)^{-1}(\ddot{x}_{ref}^w - b_1 \tanh(b_2 r) - K_r \text{sign}(r)), \quad (32)$$

donde puede observarse que (32) no incluye la red neuronal. Esto es con el objetivo de demostrar la mejora en el desempeño al incluir la red neuronal para tareas de seguimiento de trayectorias bajo la influencia de perturbaciones. El controlador (27) será denotado como CANP, que es el acrónimo de “controlador adaptable neuronal propuesto”, y el controlador (32) será llamado CMD, que significa “controlador por modos deslizantes”.

La tarea consiste en seguir una trayectoria que dibuja una lemniscata con una altura variable descrita como:

$$\begin{aligned} x_{ref} &= 0.5 \sin\left(\frac{2\pi t}{4}\right) \text{ [m]}, \\ y_{ref} &= \cos\left(\frac{2\pi t}{8}\right) \text{ [m]}, \\ z_{ref} &= 1.0 + 0.4 \sin\left(\frac{2\pi t}{4}\right) \text{ [m]}, \\ \psi_{ref} &= 0.0 \text{ [}^\circ\text{]}. \end{aligned}$$

La perturbación $\delta = [\delta_x \ \delta_y \ \delta_z \ \delta_\psi]^T$ es nula para $0 \leq t < 40$ [s] y

$$\delta(t) = \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \delta_\psi \end{bmatrix} = \begin{bmatrix} \sin\left(\frac{2\pi t}{4}\right) \\ \cos\left(\frac{2\pi t}{8}\right) \\ \sin\left(\frac{2\pi t}{4}\right) \\ \sin\left(\frac{2\pi t}{4}\right) \end{bmatrix} \text{ para } t \geq 40 \text{ [s]}$$

Las unidades de δ_x , δ_y y δ_z son [m/s²] y de δ_ψ son [°/s²]. Ver Fig. 5 para ilustración.

Los valores de las matrices K_u y K_v correspondientes a los parámetros del quadrotor descrito en (22) son los correspondientes a la plataforma AR.Drone reportados en [42], esto es,

$$\begin{aligned} K_u &= \text{diag}\{4.72 \ 6.23 \ 2.65 \ 2.38\}, \\ K_v &= \text{diag}\{0.28 \ 0.53 \ 2.58 \ 1.52\}. \end{aligned}$$

Las ganancias de los controladores (27) y (32) fueron obtenidas heurísticamente, resultando en los siguientes valores

$$\begin{aligned} m &= 10, & b_1 &= 0.75, & b_2 &= 30, \\ b_3 &= 26, & b_4 &= 0.8, & b_5 &= 0.8, \end{aligned}$$

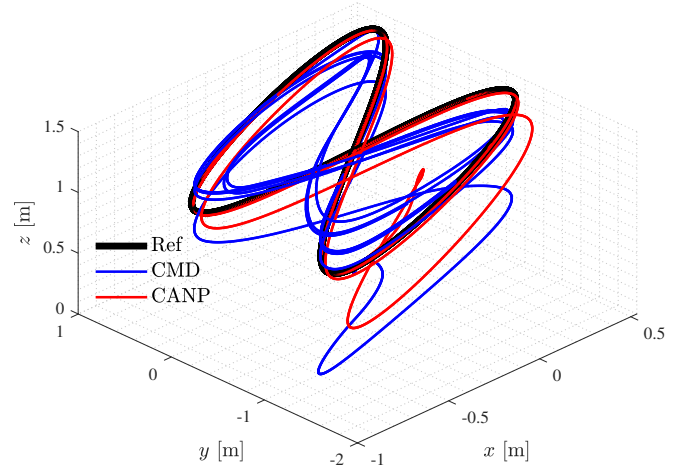


Fig. 3. Rutas realizadas por el quadrotor en simulación usando los controladores CMD y CANP.

$$\begin{aligned} K_r &= \text{diag}\{0.001 \ 0.001 \ 0.001 \ 0.001\}, \\ \alpha &= \text{diag}\{0.225 \ 0.225 \ 0.720 \ 0.225\}. \end{aligned}$$

La inicialización de los pesos se realizó con valores aleatorios entre -1 y 1 para $\hat{\Phi}$ y \hat{c} . El vector $\hat{\rho}$ se inicializó como un vector unitario.

En la Fig. 3 se muestran la ruta trazada por el quadrotor al realizar la tarea de seguimiento de trayectorias, donde se observa en color rojo la correspondiente al controlador CANP y en color azul la correspondiente al controlador CMD. Puede apreciarse fácilmente que con el controlador CANP el seguimiento de la ruta de referencia (x_{ref} , y_{ref} , z_{ref}) es mejor que el obtenido con el controlador CMD.

Las trayectorias resultantes $x(t)$, $y(t)$, y $z(t)$ así como la orientación $\psi(t)$ se presentan en la Fig. 4. Como puede observarse, el controlador CANP mantiene la posición y orientación cercana a la referencia, incluso a partir de $t \geq 40$ [s] que es cuando se presentan las perturbaciones.

La evolución de los errores de seguimiento de posición y orientación se presentan en la Fig. 6, donde se puede apreciar que los errores obtenidos con el controlador CANP se mantienen muy cercanos a cero una vez que el transitorio ha pasado. Es fácil apreciar que el controlador CANP mantiene el error muy cercano a cero, especialmente una vez que las perturbaciones están presentes. También es posible apreciar que el tiempo de respuesta del controlador CANP es menor en comparación al del controlador CMD.

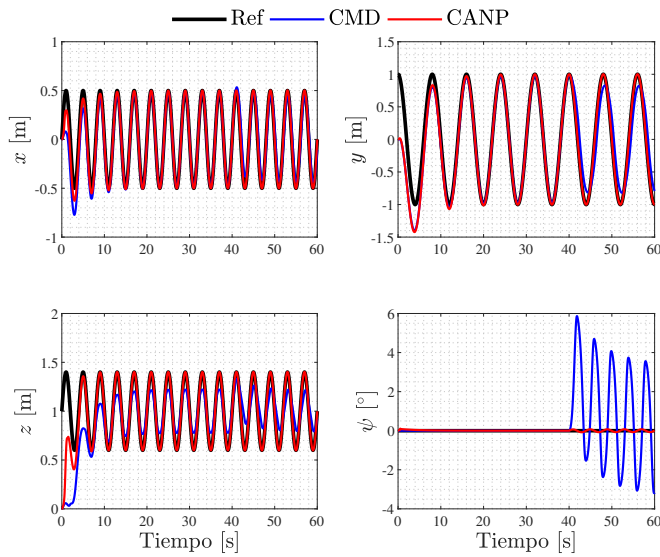


Fig. 4. Evolución temporal de la tarea de seguimiento de trayectorias en simulación.

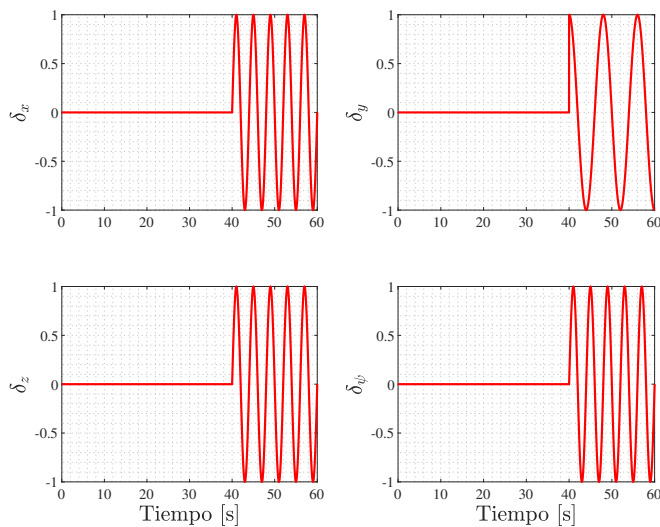


Fig. 5. Perturbaciones $\delta(t)$ durante la tarea de seguimiento de trayectorias.

En la Fig. 7 se muestran las acciones de control generadas por el controlador CANP y el controlador CMD. Note que las señales de control son adimensionales.

Con la finalidad de mostrar el aporte de la red neuronal de regresión generalizada $\hat{\delta}(t)$ en la ecuación (29) al controlador propuesto (27), en la Fig. 8 se presenta el aporte individual de la red neuronal en color rojo y del resto del controlador en color azul, de tal forma que $\mathbf{u}_{\text{CANP}} = \mathbf{u}_{\text{CMD}} + \mathbf{u}_{\text{RN}}$, donde \mathbf{u}_{CMD} está dada por la expresión (32) y $\mathbf{u}_{\text{RN}} = (RK_u)^{-1}[-\hat{\delta}(t)]$.

Los pesos de salida $\hat{\Phi}_i(t)$ de la red neuronal (29) se muestran en la Fig. 9. Como se puede observar, algunos de los pesos $\hat{\Phi}_i(t)$ presentan cambios abruptos durante los primeros instantes de la simulación, algunos se mantienen constantes, y otros oscilan. Una vez que las perturbaciones se hacen presentes en el sistema se puede observar como los pesos de salida se adaptan nuevamente a las nuevas

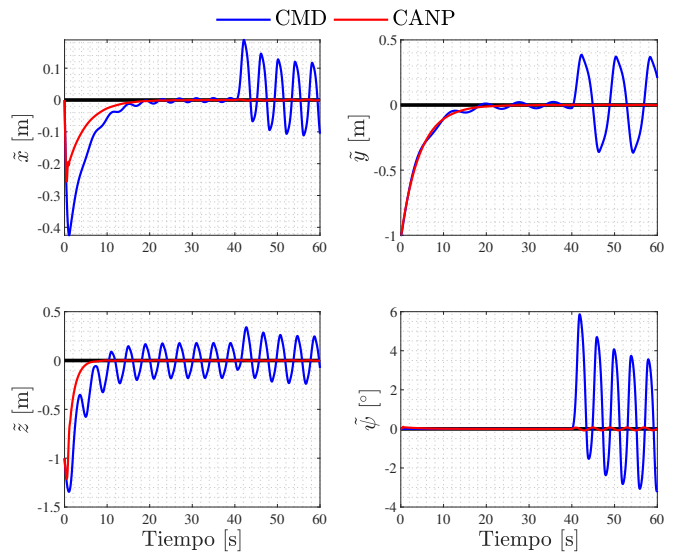


Fig. 6. Evolución temporal de los errores de seguimiento en simulación utilizando el controlador propuesto CANP y el controlador CMD.

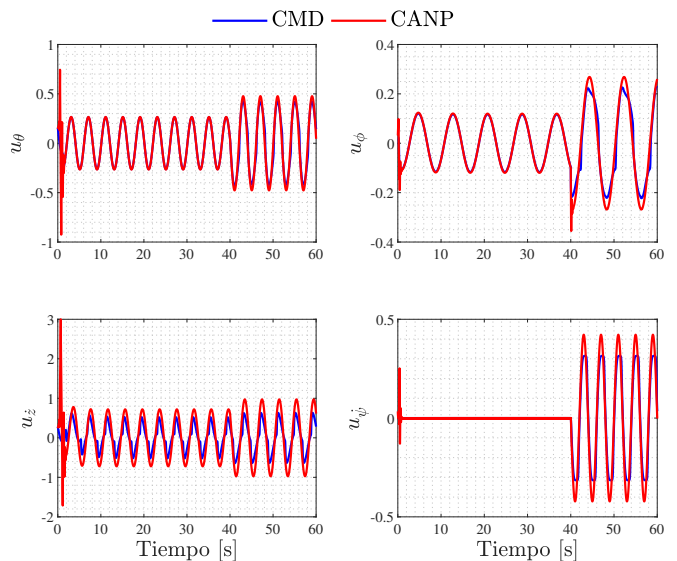


Fig. 7. Acciones de control provistas por los controladores en simulación.

condiciones de operación.

El índice de desempeño seleccionado para la comparación de los algoritmos CMD y CANP es el valor cuadrático medio (RMS por sus siglas en inglés) del error de seguimiento $e(t)$. El valor RMS de los errores de seguimiento se calculó en dos segmentos de tiempo. El primer segmento es $20 [s] \leq t < 40 [s]$ que corresponde al periodo en que el transitorio se ha desvanecido y no se presentan perturbaciones, el segundo segmento de $40 [s] \leq t \leq 60 [s]$ es donde se presentan las perturbaciones. Los resultados de tales índices de desempeño así como el porcentaje de mejora obtenido con el controlador CANP se presentan en la Tabla I. Como puede apreciarse, la mayoría de los valores más bajos se obtuvieron con el controlador CANP, llegando a alcanzar una mejora promedio

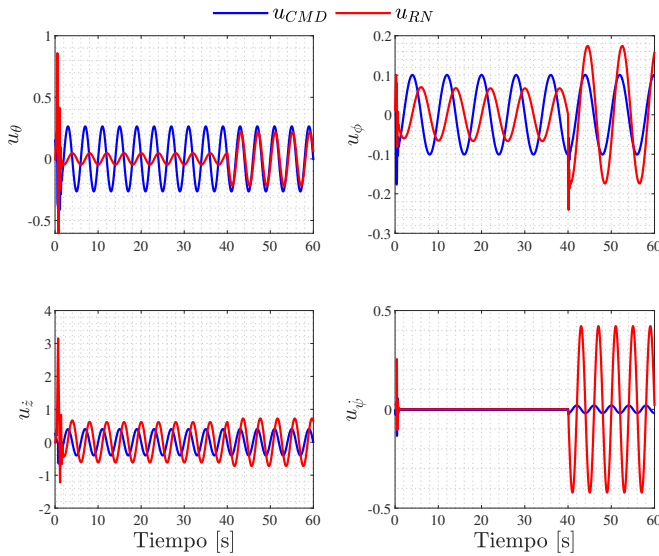


Fig. 8. Aporte de las acciones de control u_{CMD} y u_{RN} , correspondientes al controlador propuesto CANP parametrizado como $u_{CANP} = u_{CMD} + u_{RN}$

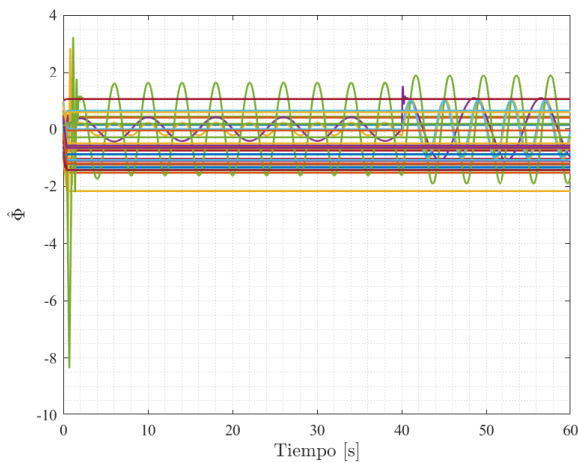


Fig. 9. Evolución temporal de los pesos de salida de la red neuronal de regresión generalizada.

Tabla I

VALOR RMS DEL ERROR $e(t)$ DURANTE LA TAREA DE SEGUIMIENTO DE TRAYECTORIAS EN SIMULACIÓN.

Señal	Sin perturbación			Con perturbación		
	CMD	CANP	(*%)	CMD	CANP	(*%)
\tilde{x} [m]	0.0051	0.0010	81.30	0.0879	0.0009	98.97
\tilde{y} [m]	0.0176	0.0039	77.83	0.2573	0.0010	99.63
\tilde{z} [m]	0.1225	0.0013	98.91	0.1688	0.0016	99.07
$\tilde{\psi}$ [°]	0.0000	0.0004	— inf	2.7525	0.0510	98.15

de posición de 86.01% en el periodo sin perturbaciones y de hasta un 99.22% en el periodo con perturbaciones.

V. CONCLUSIONES

En este trabajo, se presenta un controlador robusto a perturbaciones basando en el esquema de control por modos deslizantes y una red neuronal de regresión generalizada. El controlador propuesto es aplicado al modelo de un quadrotor con un controlador embebido. Una función de base radial en la que se adaptan los centros y la anchura es utilizada como función de activación para la red neuronal. Las simulaciones numéricas para la tarea de seguimiento de trayectorias muestran la funcionalidad del esquema de control propuesto. Además, la comparación en simulación con un controlador sin la red neuronal respaldan el uso de la propuesta para este tipo de tareas, especialmente en presencia de perturbaciones. En otras palabras, los resultados obtenidos en simulación muestran que el esquema de control propuesto provee de un buen desempeño para la tarea de seguimiento de trayectorias y que es robusto a perturbaciones.

REFERENCIAS

- [1] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, "Aviation history and unmanned flight," in *On integrating unmanned aircraft systems into the national airspace system*. Dordrecht, Netherlands: Springer, 2012, pp. 11–42.
- [2] J. Kim, M.-S. Kang, and S. Park, "Accurate modeling and robust hovering control for a quad-rotor VTOL aircraft," in *Selected papers from the 2nd International Symposium on UAVs*, Reno, NV, USA, June 8–10 2009, pp. 9–26.
- [3] F. Kendoul, Z. Yu, and K. Nonami, "Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles," *Journal of Field Robotics*, vol. 27, no. 3, pp. 311–334, 2010.
- [4] L. D. Minh and C. Ha, "Modeling and control of quadrotor MAV using vision-based measurement," in *Proc. International Forum on Strategic Technology*, Ulsan, South Korea, October 13–15 2010, pp. 70–75.
- [5] L. R. García Carrillo, A. E. Dzul López, R. Lozano, and C. Pégard, *Quad rotorcraft control: vision-based hovering and navigation*. London, UK: Springer-Verlag, 2012.
- [6] X. Zhang, X. Li, K. Wang, and Y. Lu, "A survey of modelling and identification of quadrotor robot," *Abstract and Applied Analysis*, vol. 2014, 2014.
- [7] A. Chovancová, T. Fico, L. Chovanec, and P. Hubinsk, "Mathematical modelling and parameter identification of quadrotor (a survey)," *Procedia Engineering*, vol. 96, no. 2014, pp. 172–181, 2014.
- [8] M. A. Olivares-Mendez, C. Fu, P. Ludvig, T. F. Bissyandé, S. Kannan, M. Zurad, A. Annaiyan, H. Voos, and P. Campoy, "Towards an autonomous vision-based unmanned aerial system against wildlife poachers," *Sensors*, vol. 15, no. 12, pp. 31362–31391, 2015.
- [9] A. Bulgakov, A. Evgenov, and C. Weller, "Automation of 3D building model generation using quadrotor," *Procedia Engineering*, vol. 123, pp. 101–109, 2015.
- [10] A. Bulgakov, S. Emelianov, T. Bock, and D. Sayfeddine, "Inspection of flyover bridges using quadrotor," in *Proc. International Symposium on Automation and Robotics in Construction*, vol. 32, Oulu, Finland, June 15–18 2015, pp. 1–7.
- [11] R. A. S. Fernández, S. Dominguez, and P. Campoy, " \mathcal{L}_1 adaptive control for wind gust rejection in quad-rotor UAV wind turbine inspection," in *Proc. International Conference on Unmanned Aircraft Systems*, Miami, FL, USA, June 13–16 2017, pp. 1840–1849.
- [12] S. Lee, D. Har, and D. Kum, "Drone-assisted disaster management: Finding victims via infrared camera and lidar sensor fusion," in *Proc. 3rd Asia-Pacific World Congress on Computer Science and Engineering*, Fiji, Melanesia, December 5–6 2016, pp. 84–89.
- [13] H. B. Lee, S. W. Moon, W. J. Kim, and H. J. Kim, "Cooperative surveillance and boundary tracking with multiple quadrotor UAVs," *Journal of Institute of Control, Robotics and Systems*, vol. 19, no. 5, pp. 423–428, 2013.

- [14] J. Moreno-Valenzuela, R. Pérez-Alcocer, M. Guerrero-Medina, and A. Dzul, "Nonlinear PID-type controller for quadrotor trajectory tracking," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 5, pp. 2436–2447, 2018.
- [15] D. F. Specht *et al.*, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568–576, 1991.
- [16] T. L. Seng, M. Khalid, and R. Yusof, "Adaptive GRNN for the modelling of dynamic plants," in *Proc. IEEE International Symposium on Intelligent Control*, Vancouver, Canada, 30 October 2002, pp. 217–222.
- [17] Ö. Polat and T. Yıldırım, "Genetic optimization of GRNN for pattern recognition without feature extraction," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2444–2448, 2008.
- [18] H.-Z. Li, S. Guo, C.-J. Li, and J.-Q. Sun, "A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm," *Knowledge-Based Systems*, vol. 37, pp. 378–387, 2013.
- [19] A. J. Al-Mahasneh, S. G. Anavatti, and M. A. Garratt, "Altitude identification and intelligent control of a flapping wing micro aerial vehicle using modified generalized regression neural networks," in *Proc. IEEE Symposium Series on Computational Intelligence*, Honolulu, Hawaii, USA, November 27–December 1 2017, pp. 1–6.
- [20] M. C. Santos, L. V. Santana, M. M. Martins, A. S. Brandão, and M. Sarcinelli-Filho, "Estimating and controlling UAV position using RGB-D/IMU data fusion with decentralized information/Kalman filter," in *Proc. IEEE International Conference on Industrial Technology*, Seville, Spain, March 17–19 2015, pp. 232–239.
- [21] L. V. Santana, A. S. Brandao, M. Sarcinelli-Filho, and R. Carelli, "A trajectory tracking and 3D positioning controller for the AR.Drone quadrotor," in *Proc. International Conference on Unmanned Aircraft Systems*, Orlando, FL, USA, May 27–30 2014, pp. 756–767.
- [22] R. Pérez-Alcocer and J. Moreno-Valenzuela, "Adaptive control for quadrotor trajectory tracking with accurate parametrization," *IEEE Access*, vol. 7, pp. 53 236–53 247, 2019.
- [23] W. Shen and Z. Li, "Backstepping sliding mode RBF network adaptive control for quadrotor UAV," in *Proc. Chinese Automation Congress*, Hangzhou, China, November 22–24 2019, pp. 4086–4091.
- [24] C. Luo, J. Song, and C. Lv, "Robust adaptive control design based on RBFNN with an unmanned quadrotor for transporting tasks," in *Proc. 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference*, Chongqing, China, October 11–13 2019, pp. 614–618.
- [25] R. Thusoo, S. Jain, and S. Bangia, "PID and MRAC control of a quadrotor," in *Proc. 3rd International Conference on Recent Developments in Control, Automation & Power Engineering*, Noida, India, October 10–11 2019, pp. 197–201.
- [26] B. Wang, X. Yu, L. Mu, and Y. Zhang, "A dual adaptive fault-tolerant control for a quadrotor helicopter against actuator faults and model uncertainties without overestimation," *Aerospace Science and Technology*, vol. 99, pp. 105 744–105 758, 2020.
- [27] Z. Li, X. Ma, and Y. Li, "Finite-time attitude stabilization control of a quadrotor with parametric uncertainties and disturbances," in *Proc. International Conference on Robotics and Biomimetics*, Dali, Yunnan, China, December 6–8 2019, pp. 2654–2659.
- [28] H. Bouadi and M. Tadjine, "Nonlinear observer design and sliding mode control of four rotors helicopter," *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, vol. 1, no. 7, pp. 354–359, 2007.
- [29] H. Bouadi, M. Bouchoucha, and M. Tadjine, "Sliding mode control based on backstepping approach for an UAV type-quadrotor," *World Academy of Science, Engineering and Technology*, vol. 26, no. 5, pp. 22–27, 2007.
- [30] T. Luukkonen, "Modelling and control of quadcopter," Aalto University, Espoo, Finland, Int. Rep. 22, 2011.
- [31] M. Bangura and R. Mahony, "Nonlinear dynamic modeling for high performance control of a quadrotor," in *Proc. Australasian Conference on Robotics and Automation*, Wellington, New Zealand, December 3–5 2012, pp. 115–124.
- [32] K. T. V. Vonásek, D. Fišer, and J. Faigl, "AR-Drone as a robotic platform for research and education," in *Proc. International Conference on Research and Education in Robotics*, Prague, Czech Republic, June 15–17 2011, pp. 1–15.
- [33] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadcopter," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, October 7–12 2012, pp. 2815–2821.
- [34] A. Hernandez, C. Copot, R. De Keyser, T. Vlas, and I. Nascu, "Identification and path following control of an AR.Drone quadrotor," in *Proc. 17th International Conference on System Theory, Control and Computing*, Sinaia, Romania, October 11–13 2013, pp. 583–588.
- [35] P. Falcón, A. Barreiro, and M. D. Cacho, "Modeling of Parrot Ardrone and passivity-based reset control," in *Proc. 9th Asian Control Conference*, Istanbul, Turkey, June 23–26 2013, pp. 1–6.
- [36] M. C. Santos, M. Sarcinelli-Filho, and R. Carelli, "Trajectory tracking for UAV with saturation of velocities," in *Proc. International Conference on Unmanned Aircraft Systems*, Arlington, VA, USA, June 7–10 2016, pp. 643–648.
- [37] C. Rosales, C. M. Soria, and F. G. Rossomando, "Identification and adaptive PID control of a hexacopter UAV based on neural networks," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 1, pp. 74–91, 2019.
- [38] F. Capraro, F. G. Rossomando, C. Soria, and G. Scaglia, "Cascade sliding control for trajectory tracking of a nonholonomic mobile robot with adaptive neural compensator," *Mathematical Problems in Engineering*, vol. 2017, 2017.
- [39] T. Chen and H. Chen, "Approximations of continuous functionals by neural networks with application to dynamic systems," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 910–918, 1993.
- [40] F. L. Lewis, J. Campos, and R. Selmic, *Neuro-fuzzy control of industrial systems with actuator nonlinearities*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002.
- [41] Z. Liu and C. P. Chen, "Broad learning system: Structural extensions on single-layer and multi-layer neural networks," in *Proc. International Conference on Security, Pattern Analysis, and Cybernetics*, Shenzhen, China, December 15–17 2017, pp. 136–141.
- [42] J. A. Sarapura, F. Roberti, J. M. Toibero, J. M. Sebastián, and R. Carelli, "Visual servo controllers for an UAV tracking vegetal paths," in *Machine Vision and Navigation*. Cham, Switzerland: Springer, 2020, pp. 597–625.

Control de Orden Fraccionario para el seguimiento de trayectorias en un Robot Móvil con Ruedas

Ulises Vázquez
Tecnológico Nacional de México/
Instituto Tecnológico de la Laguna
Torreón, Coahuila, México
ulises.vqz7@gmail.com

Alejandro Dzul
Tecnológico Nacional de México/
Instituto Tecnológico de la Laguna
Torreón, Coahuila, México
alejandrodzul@gmail.com

Jaime González-Sierra
Universidad Politécnica de Pachuca
Zempoala, Hidalgo, México
jamesgsjr@hotmail.com

Resumen—En este trabajo se presenta el diseño de un controlador proporcional-derivativo (PD) de orden fraccionario aplicado al seguimiento de trayectorias de un Robot Móvil con Ruedas (RMR) del tipo diferencial. Las ecuaciones del modelo utilizadas en el controlador corresponden al modelo cinemático extendido del RMR diferencial. Los términos fraccionarios del controlador son obtenidos a partir del operador de Caputo. Los resultados obtenidos se presentan a través de simulaciones, las cuales se realizaron considerando distintos órdenes de las derivadas del PD fraccionario, comparándolas también con un controlador PD de orden entero.

Index Terms—Control fraccionario, robot móvil con ruedas, control de seguimiento.

I. INTRODUCCIÓN

Un robot móvil se define como un dispositivo capaz de desplazarse en su entorno, es decir, sin estar fijo en una ubicación física. Los robots móviles pueden clasificarse de distintas formas: terrestres, aéreos, acuáticos; dentro de la clasificación de terrestres, se encuentran los robots móviles con ruedas (RMR), los cuales han ganado popularidad en las últimas décadas, debido a la gran cantidad de aplicaciones en las que pueden emplearse. A los RMR se les han diseñado diversos tipos de controladores, por ejemplo de estabilización, para estacionarse, así como para el seguimiento de trayectorias. Estos últimos, son los más utilizados, dada la misma naturaleza de las posibles aplicaciones de los RMR.

En los últimos años han aparecido múltiples publicaciones que tratan dicho problema, por ejemplo en [1] se diseña un controlador para el seguimiento de trayectorias de un robot diferencial usando el modelo cinemático extendido, mientras que en [2] se diseña un control por backstepping sobre el modelo dinámico de un RMR diferencial; en [3] se realiza el control de un robot con restricciones no holónomas y evasión de obstáculos, asimismo [4] y [5] abordan el mismo problema, pero haciendo uso de redes neuronales.

En el contexto de diseño de controladores de seguimiento para robots móviles, se pueden también encontrar trabajos desarrollados en la temática de control fraccionario. El concepto de cálculo fraccionario surgió por primera vez en el siglo XVII, y es la forma en que se denominó a la extensión del cálculo que permite considerar la integración y derivación usando órdenes no necesariamente enteros. Sin embargo, fue hasta finales del siglo XX que su desarrollo comenzó a ser

aplicado en distintas áreas de la ciencia y la ingeniería, por ejemplo en ciencia de materiales, reología, fractales, física teórica, mecánica y electrónica; múltiples aplicaciones del cálculo fraccionario pueden encontrarse en [6], [7] y [8].

En el ámbito de control, las primeras aplicaciones del cálculo fraccionario se dieron a principios de los años 60s, mismas que hacían uso de un operador integral de orden no entero. Años más tarde, el estudio de cálculo fraccionario resultó en el desarrollo del sistema CRONE (Control Robusto de Orden No Entero) en la universidad de Burdeos, Francia [9]. A partir de entonces, el cálculo fraccionario ha tenido un gran desarrollo teórico, abarcando tanto sistemas lineales [10] - [11] como no lineales [12], llegando al diseño de controladores usando distintas técnicas [13]. En el caso de los RMR, en [14] - [15] diseñan un control PD^μ de seguimiento para un RMR diferencial, ajustando las ganancias en el dominio de la frecuencia y usando filtros para calcular los términos fraccionarios; en [16] se presenta un PI^λ aplicado al control de velocidad y orientación; en [17] se implementa un backstepping de orden fraccionario utilizando el operador de Grünwald-Letnikov en la obtención de las derivadas fraccionarias. En [18] diseñan un $PI\lambda D^\mu$ para seguimiento de trayectorias en un robot diferencial controlando la cinemática inversa del robot, las ganancias se sintonizan mediante numerosos métodos y no se hace hincapié en el análisis de estabilidad del sistema. Por otro lado, en [19] también se utiliza un $PI\lambda D^\mu$ para el seguimiento de trayectorias, sin embargo, al usar el operador de Grünwald-Letnikov, utilizan el principio de memoria corta, de igual forma el análisis de estabilidad es omitido. En [20], [21] y [22] se presentan controladores tipo $PI^\lambda D^\mu$, cuyas ganancias son optimizadas por distintos tipos de algoritmos genéticos. Por otra parte, en [23] se presenta el diseño de un controlador por modos deslizantes de orden fraccionario para un sistema multi-agente.

De los controladores de orden fraccionario mencionados anteriormente, cabe resaltar los siguientes aspectos:

1. No todos realizan seguimiento de trayectorias en el plano (X, Y) , en su lugar, se realiza el seguimiento de referencias de orientación o velocidad.
2. Algunos de los trabajos que hacen seguimiento en el plano, utilizan filtros para la obtención de los términos de orden fraccionario, lo cual ocasiona que las señales

presenten ruido.

3. Existen trabajos que utilizan otros operadores (diferentes a los clásicos de Caputo o Riemann-Liouville), por lo que sus condiciones iniciales no tienen una interpretación física.
4. Algunos controladores son presentados de forma general y no para un tipo de RMR en específico, por lo tanto no consideran las posibles restricciones físicas del sistema.

Debido a lo anterior, en este artículo se presenta una propuesta alternativa para el control de seguimiento de un RMR tipo diferencial, donde se utilizará el operador de Caputo para obtener los términos fraccionarios, con lo cual se garantiza que las condiciones iniciales de las simulaciones tendrán una interpretación física, y por lo tanto su aplicación futura es una opción factible. El algoritmo de control propuesto es del tipo PD^μ (donde μ es el orden fraccionario de la parte derivativa del controlador), con el que se garantiza el seguimiento de trayectorias en el plano (X, Y) , utilizando el modelo cinemático extendido del RMR, con lo cual se elimina el uso en el dominio de la frecuencia y trabajar con la dinámica de cada motor. La principal motivación de este trabajo consiste en estudiar el comportamiento del RMR ante el controlador de orden fraccionario, con el fin de evaluar su desempeño en comparación con el controlador de orden entero y de esta manera evaluar la factibilidad de aplicar este tipo de controladores a robots móviles y/o continuar con su estudio.

El artículo está dividido de la siguiente forma: la sección II está dedicada al modelo cinemático del RMR. La sección III resume preliminares de cálculo fraccionario y definiciones de los operadores; posteriormente, en la sección IV se presenta el desarrollo del diseño del controlador PD^μ . En la sección V se muestran los resultados obtenidos en simulación. Para finalizar, la sección VI presenta las conclusiones generales sobre los resultados obtenidos.

II. MODELADO DEL ROBOT MÓVIL CON RUEDAS

Considere un robot móvil con ruedas tipo diferencial (también llamado robot $(2,0)$), como se muestra en la Figura 1, el cual cuenta con dos ruedas motorizadas unidas al mismo eje, y cuyo modelo cinemático de postura está dado por [24]:

$$\dot{x} = v \cos \theta, \quad (1a)$$

$$\dot{y} = v \sin \theta, \quad (1b)$$

$$\dot{\theta} = \omega, \quad (1c)$$

donde $[v \ \omega]^T \in \mathbb{R}^2$ son sus entradas de control, siendo la velocidad lineal y angular del robot, respectivamente; $\theta \in \mathbb{R}$ representa el ángulo de orientación y está medido en sentido contrario a las manecillas del reloj a partir del eje X , mientras que la posición del punto medio del eje de las ruedas del robot está dada por $[x \ y]^T \in \mathbb{R}^2$.

Es bien sabido que el sistema (1) no puede controlarse utilizando el enfoque de linealización por retroalimentación para controlar el punto $[x \ y]^T$, por lo tanto, para introducir el modelo cinemático extendido, se realiza un cambio de variables, donde la velocidad lineal estará ahora representada

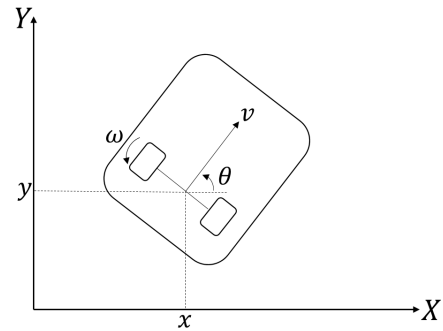


Figura 1. Robot móvil tipo diferencial.

por φ , u será la aceleración lineal, y ω seguirá siendo la velocidad angular. A continuación se presentan las ecuaciones del modelo cinemático de postura extendido del robot:

$$\dot{x} = \varphi \cos \theta, \quad (2a)$$

$$\dot{y} = \varphi \sin \theta, \quad (2b)$$

$$\dot{\varphi} = u, \quad (2c)$$

$$\dot{\theta} = \omega. \quad (2d)$$

El modelo cinemático de postura extendido será utilizado para el diseño del controlador de seguimiento, dado que permite trabajar con las posiciones x - y del robot.

III. OPERADORES DE ORDEN FRACCIONARIO

El cálculo de orden fraccionario es una generalización del cálculo de orden entero a orden real. Para dicha generalización se utiliza en la literatura un operador, el cual puede denotarse de la siguiente forma: ${}_a D_t^\alpha$, donde a y t son los límites de la operación, mientras que α es el orden. Asimismo, la integración y diferenciación de orden fraccionario se definen de la siguiente manera:

$${}_a D_t^\alpha f(t) = \begin{cases} \frac{d^\alpha}{dt^\alpha} f(t), & \alpha > 0, \\ f(t), & \alpha = 0, \\ \int_a^t f(\tau) d\tau^{-\alpha}, & \alpha < 0. \end{cases} \quad (3)$$

Existen numerosas definiciones matemáticas para describir tanto derivadas como integrales de orden fraccionario ([6], [7] y [10], sólo por citar algunas). De entre estas múltiples definiciones, aquí se presentan dos de las que han sido mayormente utilizadas en el área de control. La primera es la definición de Riemann-Liouville [6], cuya integral de orden fraccionario se define como:

$${}_a^{RL} D_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t - \tau)^{\alpha-1} f(\tau) d\tau, \quad (4)$$

donde $0 < \alpha < 1$, y a es el valor inicial. $\Gamma(\alpha) = \int_0^\infty \exp^{-t} t^{\alpha-1} dt$. Del mismo modo, se define la derivada como:

$${}_a^{RL} D_t^\alpha f(t) = \frac{1}{\Gamma(n - \alpha)} \frac{d^n}{dt^n} \left[\int_a^t \frac{f(\tau)}{(t - \tau)^{\alpha-n+1}} d\tau \right], \quad (5)$$

donde $n - 1 < \alpha \leq n$.

Por su parte, el operador de Caputo [6] es una definición alternativa a la de Riemann-Liouville, y está definido como

$${}_a^C D_t^\alpha f(t) \equiv \frac{1}{\Gamma(m - \alpha)} \int_a^t \frac{f^{(m)}(\tau)}{(t - \tau)^{\alpha - m + 1}} d\tau, \quad (6)$$

donde $m - 1 < \alpha < m$.

La integral por el operador de Caputo queda descrita a continuación;

$${}_0^C D_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t \frac{f(\tau)}{(t - \tau)^{1 - \alpha}} d\tau, \alpha > 0. \quad (7)$$

Cabe destacar que el operador de Caputo cuenta con condiciones iniciales que son interpretables físicamente, por ello se utiliza preferentemente en sistemas físicos, sin embargo, bajo condiciones iniciales homogéneas, ambas definiciones son equivalentes de acuerdo a la siguiente expresión:

$${}^R D_t^\alpha f(t) = {}^C D_t^\alpha f(t) + \sum_{k=0}^{n-1} \frac{(t-a)^{k-\alpha}}{\Gamma(k-\alpha+1)} f^{(k)}(a), \quad (8)$$

para $f^{(k-a)}(a) = 0, (k = 0, 1, \dots, n-1)$.

Debido a lo mencionado anteriormente, en lo subsecuente, al utilizar la notación D^α , se estará haciendo referencia a la derivada utilizando el operador de Caputo.

IV. CONTROLADOR DE ORDEN FRACCIONARIO

Para lidiar con el problema de seguimiento de trayectorias, la estrategia consistirá en implementar un controlador tipo PD^μ [6], para ello se considerará el modelo cinemático extendido del robot diferencial visto en (2).

Utilizando la técnica de linealización por realimentación de estados, se cancelarán los términos no lineales en el modelo del robot [25]. Eligiendo como la salida a controlar a $h = [x \ y]^T$, se deriva ésta las veces que sea necesario, hasta que las entradas de control del sistema (u y ω) aparezcan de forma explícita. Entonces

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = A(\theta, \varphi) \begin{bmatrix} u \\ \omega \end{bmatrix}, \quad (9)$$

donde

$$A(\theta, \varphi) = \begin{bmatrix} \cos(\theta) & -\varphi \sin(\theta) \\ \sin(\theta) & \varphi \cos(\theta) \end{bmatrix}. \quad (10)$$

Antes de seguir adelante, debe tomarse en cuenta la restricción que se presenta en el sistema, ya que si se invierte (10) (que es la matriz de desacoplamiento), se obtiene

$$A^{-1}(\theta, \varphi) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\frac{\sin(\theta)}{\varphi} & \frac{\cos(\theta)}{\varphi} \end{bmatrix}. \quad (11)$$

Comentario 1. Note que de [23], el $\det(A) = \varphi$. Esto quiere decir que la inversa de la matriz A existe si y solo si $\varphi \neq 0$. Físicamente, esto significa que el robot está en movimiento. Por lo tanto, se asumirá que el robot se encuentra en movimiento por acción de un controlador que permitió su arranque, posterior a esto, el controlador aquí presentado comienza a

actuar. De igual forma, se asumirá que la conmutación entre los dos controladores es estable.

Para obtener el controlador de seguimiento de trayectorias del robot diferencial, considere (9), donde las entradas de control $[u \ \omega]^T$ se definen como

$$\begin{bmatrix} u \\ \omega \end{bmatrix} = A^{-1}(\theta, \varphi) \begin{bmatrix} \bar{u} \\ \bar{\omega} \end{bmatrix}. \quad (12)$$

donde \bar{u} y $\bar{\omega}$ son los controles auxiliares, los cuales tienen como propósito eliminar las no linealidades del sistema, lográndose al invertir la matriz $A(\theta, \varphi)$ la cual existe siempre y cuando $\varphi \neq 0$. Por lo tanto, al desarrollar las ecuaciones se obtiene lo siguiente:

$$u = \bar{u} \cos(\theta) + \bar{\omega} \sin(\theta), \quad (13a)$$

$$\omega = -\frac{\bar{u} \sin(\theta)}{\varphi} + \frac{\bar{\omega} \cos(\theta)}{\varphi}, \quad (13b)$$

y la dinámica de (9) con (12) da como resultado el siguiente sistema

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \bar{u} \\ \bar{\omega} \end{bmatrix}. \quad (14)$$

Las entradas de control han quedado entonces establecidas, sin embargo las variables auxiliares introducidas en (12) aún deben definirse. Para ello, se proponen los siguientes errores de seguimiento:

$$e_x = x - x_d, \quad e_y = y - y_d, \quad (15a)$$

$$\dot{e}_x = \dot{x} - \dot{x}_d, \quad \dot{e}_y = \dot{y} - \dot{y}_d, \quad (15b)$$

$$\ddot{e}_x = \ddot{x} - \ddot{x}_d, \quad \ddot{e}_y = \ddot{y} - \ddot{y}_d. \quad (15c)$$

Sustituyendo (15c) en la dinámica del error, se puede definir el valor de los controles auxiliares dados por

$$\bar{u} = \ddot{x}_d - D^\alpha e_x k_{dx} - e_x k_{px}, \quad (16a)$$

$$\bar{\omega} = \ddot{y}_d - D^\alpha e_y k_{dy} - e_y k_{py}. \quad (16b)$$

donde (16a) y (16b) tienen la estructura de un PD^μ , con k_{dx} , k_{px} , k_{dy} y k_{py} como las ganancias de los controladores. Además, el controlador se ajusta para que compense la aceleración deseada del robot en (X, Y) , dado que ésta se conoce. Dicho todo lo anterior, aún queda pendiente la noción de estabilidad sobre el sistema, para lo cual se hará referencia en primera instancia al caso de un controlador similar pero de orden entero, para ello considérense las ecuaciones (16a) y (16b) sustituyendo α por derivadas de orden entero:

$$\bar{u} = \ddot{x}_d - \dot{e}_x k_{dx} - e_x k_{px}, \quad (17a)$$

$$\bar{\omega} = \ddot{y}_d - \dot{e}_y k_{dy} - e_y k_{py}. \quad (17b)$$

Para demostrar que las entradas de control propuestas son estables, se hace el siguiente análisis para el caso de e_x :

$$\varepsilon_1 = e_x, \quad \dot{\varepsilon}_1 = \dot{e}_x, \quad (18a)$$

$$\varepsilon_2 = \dot{e}_x, \quad \dot{\varepsilon}_2 = -k_{dx} \varepsilon_2 - k_{px} \varepsilon_1. \quad (18b)$$

Expresando lo anterior en forma matricial se tiene,

$$\dot{\varepsilon} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k_{px} & -k_{dx} \end{bmatrix}}_{K_x} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix}. \quad (19)$$

Después, se analiza el polinomio característico de (19):

$$\det(sI - K_x) = \det \left(\begin{bmatrix} s & -1 \\ k_{px} & s + k_{dx} \end{bmatrix} \right) = s^2 + sk_{dx} + k_{px}. \quad (20)$$

De forma que al asignar polos estables a (20) se prueba la estabilidad del sistema, esto se logra cuando $k_{px}, k_{dx} > 0$. Si se realiza el mismo análisis para e_y se encontrará la misma condición para las ganancias. Ahora tomemos el análisis anterior como punto de partida para el caso de (16a) y (16b).

$$\varepsilon_3 = e_x, \quad D^\alpha \varepsilon_3 = \varepsilon_4, \quad (21a)$$

$$\varepsilon_4 = D^\alpha e_x, \quad D^\alpha \varepsilon_4 = -D^\alpha e_x k_{dx} - e_x k_{px}. \quad (21b)$$

Ahora se expresan las ecuaciones anteriores de forma matricial.

$$D^\alpha \varepsilon = \underbrace{\begin{bmatrix} 0 & 1 \\ -k_{px} & -k_{dx} \end{bmatrix}}_{K_x} \begin{bmatrix} \varepsilon_3 \\ \varepsilon_4 \end{bmatrix}. \quad (22)$$

Al igual que como se muestra en (20), se obtendrá un pseudopolinomio.

$$\det(s^\lambda I - K_x) = \det \left(\begin{bmatrix} s^\lambda & -1 \\ k_{px} & s^\lambda + k_{dx} \end{bmatrix} \right), \quad (23a)$$

$$\det(s^\lambda I - K_x) = s^{2\lambda} + s^\lambda k_{dx} + k_{px} \quad (23b)$$

donde (23b) describe la parte que corresponde a e_x . Ahora usando el teorema de Matignon [26], [27], que afirma que los argumentos de las raíces deben cumplir que

$$| \arg(\lambda_i) | > \alpha \frac{\pi}{2}. \quad (24)$$

Con base en lo anterior y siguiendo el mismo razonamiento que el utilizado en la prueba de estabilidad del controlador de orden fraccionario que se presenta en [28], se tiene que la condición de estabilidad del controlador presentado en (16a) y (16b) está dada por las ganancias k_{dx}, k_{dy}, k_{px} y k_{py} , donde si $\alpha = 1$, basta con que las matrices de ganancias sean Hurwitz para garantizar la estabilidad del sistema.

Por último, la Figura 2 ilustra el diagrama a bloques del sistema en lazo cerrado, donde se aprecia más fácilmente cómo las consignas de posición, velocidad y aceleración son generadas a partir de la trayectoria deseada, tal como suele ser cuando se emplea el modelo cinemático. El controlador recibe las señales deseadas y actúa sobre el modelo cinemático extendido del robot diferencial, del cual se obtienen las aceleraciones en x y y , mismas que se integran dos veces para extraer la posición del robot y poder realimentarla al controlador, generando así los errores de seguimiento.

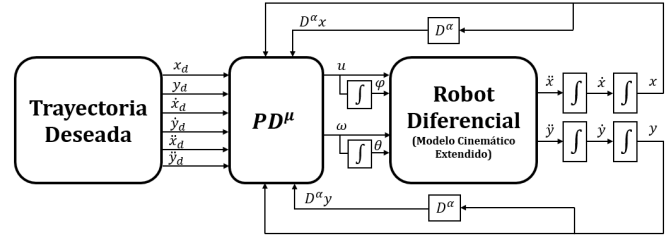


Figura 2. Diagrama de bloques del sistema.

V. SIMULACIONES NUMÉRICAS

Con el fin de evaluar el desempeño del controlador propuesto, se realizaron distintas simulaciones utilizando Matlab-Simulink. El objetivo consiste en que el robot siga una trayectoria formada por una lemniscata en un periodo de 20 [s]. Cabe destacar que las derivadas de orden fraccionario se obtuvieron con el uso del toolbox *FOMCON (Fractional-order Modeling and Control)*. A continuación se presentan las ecuaciones de la trayectoria deseada:

$$x_d = \cos(\pi t/10), \quad (25)$$

$$y_d = 0.5 \sin(\pi t/5). \quad (26)$$

Las ganancias del controlador fueron sintonizadas heurísticamente, teniendo como objetivo que la trayectoria del robot converja a la trayectoria deseada con la mayor rapidez posible, pero sin presentar sobre impulsos. Se realizaron cinco simulaciones con distintos órdenes de derivación, con el fin de mostrar las variaciones según la elección del orden. Con el propósito de comparar el controlador propuesto, se simuló también el comportamiento del sistema con un *PD* de orden entero. Las condiciones iniciales fueron $x = 1$ [m], $y = -0.35$ [m], $\theta = 0$ [rad]. Las ganancias de los controladores fraccionarios son: $k_{dx} = 12$, $k_{dy} = 9$, $k_{px} = 20$ y $k_{py} = 14$, mientras que las del *PD* entero son: $k_{dx} = 4.95$, $k_{dy} = 4.50$, $k_{px} = 19$ y $k_{py} = 8$.

La Figura 3 ilustra la trayectoria en el plano realizada por el robot diferencial. Se puede apreciar que el orden del operador afecta en gran medida la respuesta del sistema. Concretamente, cuando $\mu = 0.6$ y $\mu = 0.7$, el sistema presenta sobre impulsos, los cuales son atenuados, mientras que para los demás casos, esto no ocurre. Sin embargo, una vez realizada la etapa de transición, en todos los casos se logra la tarea de seguimiento.

En las Figuras 4 y 5 se muestra el seguimiento de la trayectoria en cada eje; dado que la condición inicial es $x = 1$, el seguimiento no presenta mayor inconveniente en su fase de transición en el eje X . Caso contrario al eje Y , donde la convergencia a la trayectoria deseada varía dependiendo del orden del controlador. No obstante, en los primeros 3 segundos de la simulación el controlador de orden entero se encuentra ligeramente por encima de la señal de referencia, por su parte, los controladores de orden fraccionario se encuentran por debajo de la referencia, pero en todos los casos se logra el seguimiento.

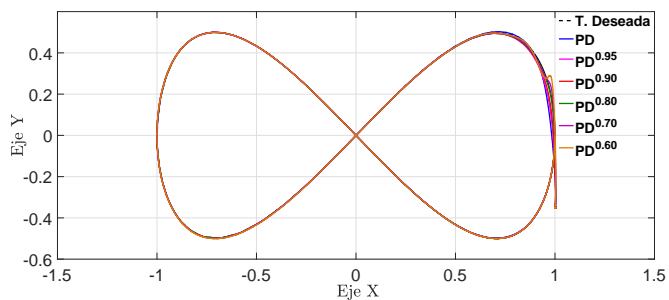


Figura 3. Seguimiento de la trayectoria en el plano (x, y) .

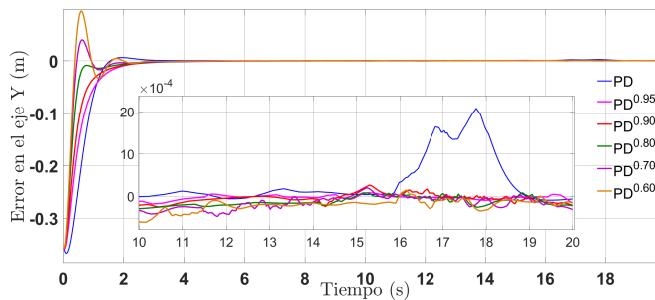


Figura 7. Error en el eje y .

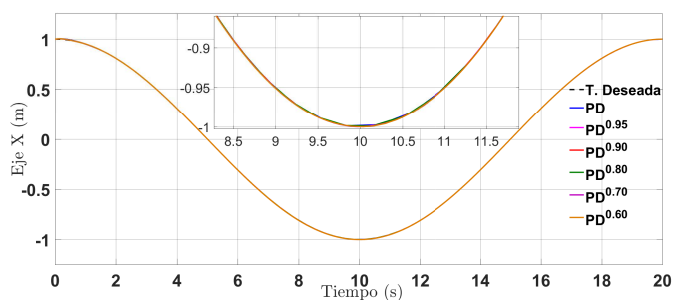


Figura 4. Seguimiento de la trayectoria en el eje x .

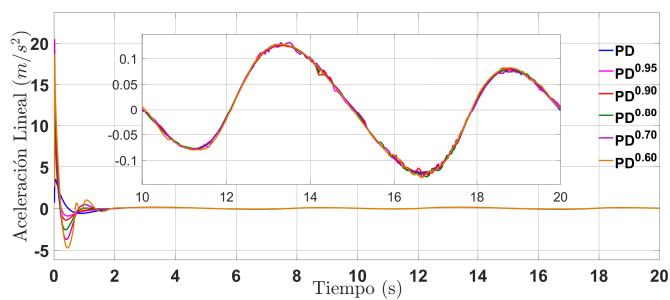


Figura 8. Aceleración lineal u .

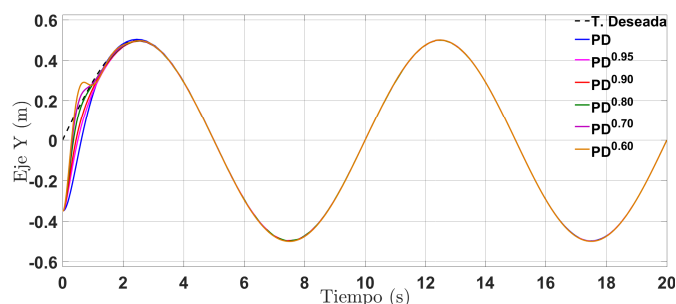


Figura 5. Seguimiento de la trayectoria en el eje y .

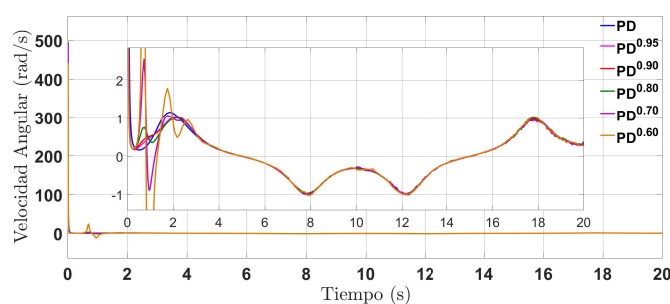


Figura 9. Velocidad angular ω .

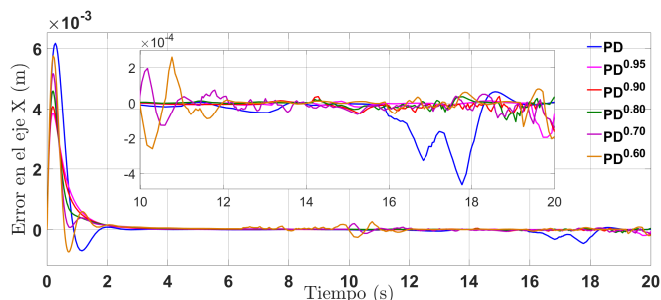


Figura 6. Error en el eje x .

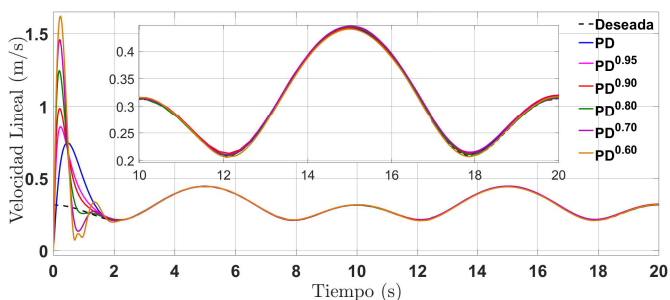


Figura 10. Velocidad lineal φ .

Las Figuras 6 y 7 resultan muy interesantes, ya que en ellas se aprecian los errores de posición en los ejes X y Y , respectivamente. El error en el eje X , en todos los casos, comienza con un crecimiento que se atenúa antes de transcurridos 2 segundos, dado que el RMR está tratando de alcanzar la referencia en el eje Y . Dicho error es más grande en el controlador entero, mientras que para el controlador fraccionario, el error será más pequeño conforme el orden de derivación sea más cercano a uno. Durante el resto de la simulación los errores se mantienen en una región cercana al cero, siendo el $PD^{0.80}$ y $PD^{0.90}$ los controladores que presentan las variaciones más pequeñas y el $PD^{0.95}$ el que se mantiene más cercano al cero durante la mayor parte de la simulación. El error en el eje Y tiene un comportamiento ligeramente distinto. Cuando $\mu = 0.6, 0.7$ y 1 , se presentan sobre impulsos que son rápidamente atenuados. Una vez que se tiene la convergencia en la trayectoria deseada, el seguimiento es realizado y es sólo entre los segundos 16 y 18 que se observa cómo la señal del PD entero presenta un error significativamente mayor al resto de los controladores. En el acercamiento de la imagen, visualmente es notorio que una vez más el mejor seguimiento lo realizan $PD^{0.80}$, $PD^{0.90}$ y $PD^{0.95}$, y es aquí donde se observa de forma clara que entre los segundos 16 y 18 la señal de error del PD entero es al menos 3 veces mayor que el resto de los controladores. El crecimiento de los errores de posición tanto en el eje X como en el eje Y es atribuido a que durante este segmento de la simulación, la curva de la trayectoria deseada es muy pronunciada.

El decremento en los errores de posición para los controladores de orden fraccionario que fue descrito con anterioridad, se ve reflejado en las Figuras 8 y 9, las cuales presentan las señales de control. Note que, al inicio, las señales presentan picos, los cuales no son vistos en la señal de control entero, así como las señales tienden a ser más ruidosas conforme el orden del controlador esté más alejado de 1. Por último, la Figura 10 muestra la velocidad lineal del robot. Es evidente que el controlador entero presenta la menor amplitud de la velocidad lineal, la cual aumenta conforme el orden del controlador es menor.

VI. CONCLUSIONES

En este artículo se presentó una alternativa para el seguimiento de trayectorias de un robot móvil con ruedas, de tipo diferencial; para ello se utilizó un controlador PD^μ para el seguimiento de trayectorias en los ejes (X, Y) , utilizando el modelo cinemático extendido del robot.

Para obtener las derivadas de orden fraccionario se empleó el operador de Caputo. Las ganancias se obtuvieron heurísticamente, siguiendo [26] y [27] con el fin de garantizar la estabilidad del sistema. Las simulaciones se realizaron en *Matlab-Simulink* usando el toolbox *FOMCON* para las derivadas de orden fraccionario, obteniendo resultados con distintos órdenes en el controlador para evaluar su desempeño. Adicionalmente se compararon los resultados obtenidos con los de un PD de orden entero, esto con el propósito de evaluar

los resultados del controlador de orden fraccionario. Los controladores de orden fraccionario cercanos a 1 presentaron errores de posición menores que el controlador entero, tanto en X como en Y ; asimismo, dichos errores se tradujeron en señales de control con amplitudes más grandes y ruidosas para los controladores fraccionarios (según el orden elegido, creciendo entre más pequeño sea el orden), sin embargo, se logra el seguimiento después de la fase de transición.

Luego de analizar los resultados de las simulaciones se encontró que es muy útil el tener la posibilidad de alterar el orden de las derivadas, ya que con ello se tiene la posibilidad de sintonizar más parámetros y por ende, mejorar los posibles resultados que se obtengan, sin embargo se pudiese tener la desventaja de complicar más el análisis de estabilidad, y en general la forma de atacar el problema, así como hacer la labor de sintonización más tardía.

Como trabajo futuro se busca extender el análisis de estabilidad del sistema, de igual forma se esperan obtener resultados experimentales empleando como plataforma de trabajo el robot móvil de tipo diferencial *Qbot2e* de la empresa Quanser. También, se espera mejorar los resultados con algún algoritmo de optimización que permita reducir, ya sea los errores de seguimiento, o el tiempo de convergencia del sistema a las trayectorias deseadas.

REFERENCIAS

- [1] Contreras, J. C. M., Herrera, D., Toibero, J. M., & Carelli, R. "Controllers design for differential drive mobile robots based on extended kinematic modeling". 2017 European Conference on Mobile Robots (ECMR), IEEE, pp. 1-6, September 6-8, Paris, Francia, 2017.
- [2] Chwa, Dongkyoung. "Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization". IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 40, no 6, pp. 1285-1295, 2010.
- [3] Yang, H., Fan, X., Shi, P., & Hua, C. "Nonlinear control for tracking and obstacle avoidance of a wheeled mobile robot with nonholonomic constraint". IEEE Transactions on Control Systems Technology, vol. 24, no 2, pp. 741-746, 2015.
- [4] Park, B. S., Yoo, S. J., Park, J. B., & Choi, Y. H. "Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty". IEEE Transactions on Control Systems Technology, vol. 17, no 1, pp. 207-214, 2008.
- [5] Fierro, R., & Lewis, F. L. "Control of a nonholonomic mobile robot using neural networks". IEEE transactions on neural networks, vol. 9, no 4, pp. 589-600, 1998.
- [6] Podlubny, I. "Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications". Elsevier, 1998.
- [7] Miller, K. S., & Ross, B. "An introduction to the fractional calculus and fractional differential equation". Wiley- 1993.
- [8] Hilfer, R. "Applications of fractional calculus in physics". Singapore: World scientific, vol. 35, no 12, pp. 87-130, 2000.
- [9] Oustaloup, A. "La dérivation non entière: Théorie". Synthèse et Applications. 1995.
- [10] Valério, D., & Da Costa, J. S. "An introduction to fractional control". IET, vol. 91, pp. 32-208, 2013.
- [11] Padula, F., & Visioli, A. "Advances in robust fractional control". Switzerland: Springer International Publishing, pp. 1-176, 2015.
- [12] Petráš, I. "Fractional-order nonlinear systems: modeling, analysis and simulation". Springer Science & Business Media, 2011.
- [13] Luo, Y., & Chen, Y. "Fractional order motion controls" John Wiley & Sons Limited, 2012.
- [14] Rojas-Moreno, A., & Perez-Valenzuela, G. "Fractional order tracking control of a wheeled mobile robot". 2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON). IEEE, pp. 1-4, August 15-18, Lima, Perú, 2017.

- [15] Lichuan Z., Liu L., & Zhang S. "Design, Implementation, and Validation of Robust Fractional-Order PD Controller for Wheeled Mobile Robot Trajectory Tracking". *Complexity*, Volume 2020, Article ID 9523549, 2020.
- [16] Orman, K., Basci, A., & Derdiyok, A. "Speed and direction angle control of four wheel drive skid-steered mobile robot by using fractional order PI controller". *Elektronika ir Elektrotechnika*, vol. 22, no 5, pp. 14-19, 2016.
- [17] Zhao, Y., Chen, N., & Tai, Y. "Trajectory tracking control of wheeled mobile robot based on fractional order backstepping". 2016 Chinese Control and Decision Conference (CCDC). IEEE, pp. 6730-6734, May 28-30, Yinchuan, China, 2016.
- [18] Ammar H.H., & Azar A.T. "Robust Path Tracking of Mobile Robot Using Fractional Order PID Controller". In: Hassanien A., Azar A., Gaber T., Bhatnagar R., F. Tolba M. (eds). *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019)*. AMLTA 2019. *Advances in Intelligent Systems and Computing*, vol 921. Springer, Cham
- [19] Ostalczyk P., & Stolarski M. "Fractional-order PID controllers in a mobile robot control". *IFAC Proceedings Volumes*, Vol 42, Issue 13, pp. 268-271, 2009
- [20] Tawfik, M. A., Abdulwahb, E. N., & Swadi, S. M. "Trajectory tracking control for a wheeled mobile robot using fractional order $PI^{\alpha}D^{\beta}$ controller". *Al-Khwarizmi Engineering Journal*, vol. 10, no 3, pp. 39-52, 2014.
- [21] Al-Mayyahi, A., Wang, W., & Birch, P. "Design of fractional-order controller for trajectory tracking control of a non-holonomic autonomous ground vehicle". *Journal of Control, Automation and Electrical Systems*, vol. 27, no 1, pp. 29-42, 2016.
- [22] Rasheed, L. T., & Al-Araji, A. S. "A cognitive nonlinear fractional order PID neural controller design for wheeled mobile robot based on bacterial foraging optimization algorithm". *Engineering and Technology Journal*, vol. 35, no 3 Part (A) Engineering, pp. 289-300, 2017.
- [23] Bandyopadhyay, B., & Kamal, S. "Stabilization and control of fractional order systems: a sliding mode approach". Switzerland: Springer International Publishing, vol. 317, 2015.
- [24] Campion, G., Bastin, G., & Dandrea-Novet, B. "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots". *IEEE transactions on robotics and automation*, vol. 12, no 1, pp. 47-62, 1996.
- [25] Khalil, H. K., & Grizzle, J. W. "Nonlinear systems". Upper Saddle River, NJ: Prentice hall, vol. 3, 2002.
- [26] Matignon, D. "Stability results for fractional differential equations with applications to control processing". *Computational engineering in systems applications*, pp. 963-968, July 9-12, Lille, Francia, 1996.
- [27] Petráš, I. "Stability of fractional-order system with rational orders: A Survey". *An International Journal for Theory and Applications*, Vol. 12, Number 3, pp. 269-298, 2009.
- [28] Munoz Vázquez, A.J., Parra Vega, V., Sánchez Orta A., García, O. & Izaguirre Espinoza C. "Attitude tracking control of a quadrotor based on absolutely continuous fractional integral sliding modes". 2014 IEEE Conference on control applications (CCA). IEEE, pp. 717-722, October 8-10, Antibes/Nice, Francia, 2014.

Acotamiento de soluciones en sistemas lineales: Un enfoque de control difuso

Jesus Rogelio Pulido-Luna, Angel Andres Osuna-Arrazola,
Selene Lilette Cardenas-Maciel, Nohe Ramon Cazarez-Castro
Instituto Tecnológico de Tijuana
Tecnológico Nacional de México
Tijuana, México
{angel.osuna18,jesus.pulido19}@tectijuana.edu.mx,
{lilettecardenas,nohe}@ieee.org

Jorge Antonio Lopez-Renteria
Instituto Tecnológico de Tijuana
Consejo Nacional de Ciencia y Tecnología-
Tecnológico Nacional de México
Tijuana, México
jorge.lopez@tectijuana.edu.mx

Abstract—En este artículo se presenta una metodología que permite exhibir en un sistema en lazo cerrado trayectorias de salida acotadas por dos funciones exponenciales. La metodología consiste en determinar una señal de referencia y diseñar un controlador difuso y ganancias tal que force al sistema de errores a tener un comportamiento exponencial acotado que es definido de acuerdo a las necesidades del usuario. Se reporta la verificación numérica del enfoque propuesto al aplicarlo en un modelo matemático de un motor de corriente directa, mostrando que tanto las trayectorias de salida del sistema en lazo cerrado, así como la del sistema de errores permanecen acotadas mientras que se logra el objetivo de control.

Index Terms—acotamiento de soluciones, control difuso, sistemas lineales

I. INTRODUCTION

A menudo se ha clasificado al control difuso como un control inteligente [1] debido a que únicamente se requiere la clasificación de datos obtenidos de la evolución dinámica del sistema que se desea controlar. Fue concebido por Zadeh [2] y originalmente no se presentó como una metodología de control, pero se establecieron las bases matemáticas para los conjuntos difusos. Fue hasta 1974 cuando Assilian y Mamdani [3] desarrollaron en Reino Unido el primer controlador difuso diseñado para la máquina de vapor. La implementación real de un controlador de este tipo no fue realizada hasta 1980 en una planta cementera en Dinamarca de la compañía FL Smidth and Co [4].

En los últimos años las aplicaciones del control en actuadores o procesos delicados han crecido, por lo que es importante poder garantizar que la solución de un sistema dado estará acotada en una región adecuada para su correcto funcionamiento. El acotar las soluciones de un sistema que es controlado mediante lógica difusa puede ser de gran utilidad

Esta investigación fue apoyada por CONACyT mediante el proyecto A1-S-32341 y por el Tecnológico Nacional de México mediante los proyectos “ESTUDIO DE LA DINÁMICA EN ESTRUCTURAS CONSTRUCTIVAS VERTICALES - PARTE 3: MODELADO MATEMÁTICO E INSTRUMENTACIÓN DE UNA ESTRUCTURA CONSTRUCTIVA VERTICAL”, “Predicción de compuestos nitrogenados de efluente de planta escala de tratamiento de agua residual” y “DESARROLLO DE UNA TOOL BOX DE MATLAB PARA LA CONSTRUCCIÓN DE COEFICIENTES DIFUSOS EN MODELOS MATEMÁTICOS”.

en una gran cantidad de aplicaciones en ciencias e ingeniería, particularmente en aquellas aplicaciones en las que un comportamiento no deseado pueda causar daños en actuadores como las reportadas en [5]–[8] o procesos sensibles como los de [9]–[11]. En la literatura relacionada al tema es posible encontrar trabajos como [12] y [13] donde se han propuesto diferentes enfoques para el acotamiento de las salidas de un sistema. Sin embargo, es un tema que aún no ha sido completamente explorado, por lo que la principal aportación de este artículo es definir una metodología para acotar soluciones de un sistema acotando el sistema de errores.

El resto del artículo está dividido de la siguiente manera: en la Sección II se presenta la teoría necesaria para comprender la metodología que se propone; en la Sección III se presenta la descripción del sistema con el que se trabajará y se define el problema que se considera en el presente artículo; en la Sección IV se desarrolla la metodología propuesta, mientras que en la Sección V se muestran resultados numéricos del método aplicado a un motor de corriente directa (CD). Por último, en la Sección VI se presentan conclusiones.

II. PRELIMINARES

En esta sección, se presentan definiciones que son necesarias para comprender la metodología que se propone.

A. Conjuntos difusos

A continuación, se presenta la definición de conjunto difuso y función de membresía que se usará durante el resto del artículo:

Definición 1 (L.A. Zadeh [2]): Un conjunto difuso A es una pareja ordenada de dos elementos definida como $A = \{(x, \mu_A(x)) \mid x \in X, \mu_A(x) \in [0, 1]\}$, donde $X \subseteq \mathbb{R}$ y $\mu_A(x)$ es denominada función de membresía del conjunto difuso A . La función de membresía μ_A de un conjunto difuso no vacío A es una función: $\mu_A : X \rightarrow [0, 1]$. La función μ_A es interpretada como el grado de pertenencia de cada elemento x al conjunto difuso A , para cada $x \in X$, donde X es el soporte del conjunto difuso A .

Tomando en cuenta la Definición 1, es posible construir la función de membresía tipo triangular que se utilizará durante el resto del artículo.

Definición 2 (Jant et al. [14]): Se define como función de membresía triangular de un conjunto difuso A a la siguiente expresión:

$$\mu_A(x; a, b, c) = \begin{cases} 0 & \text{si } x \leq a. \\ \frac{x-a}{b-a} & \text{si } a \leq x \leq b. \\ \frac{c-x}{c-b} & \text{si } b \leq x \leq c. \\ 0 & \text{si } c \leq x. \end{cases} \quad (1)$$

También se puede expresar en una forma mas compacta usando los operadores min y max como:

$$\mu_A(x; a, b, c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (2)$$

Los parámetros a , b y c cumplen $a \leq b \leq c$, donde b es el núcleo del triángulo.

La operación de defusificación sobre un conjunto difuso permite convertir la información difusa en dura. Uno de los métodos de defusificación más utilizados es la del centro de gravedad (COG por sus siglas en inglés), la cual se define como:

Definición 3 (Y.Bai y D.Wang [15]): El COG de un conjunto A se define como:

$$v = \frac{\sum_{x_i} \mu_A(x_i) x_i}{\sum_{x_i} \mu_A(x_i)} \quad (3)$$

B. Sistemas Difusos

Existen diferentes técnicas para definir un sistema difuso [3], [16], [17]. Este trabajo se basa en sistemas difusos de tipo Mamdani, los cuales se definen de la siguiente forma:

Definición 4 (Gorrín-Ortega et al. [18]): Un sistema de inferencia difusa tipo Mamdani es una terna $M = (E, R, S)$, donde:

- E es el conjunto de variables lingüísticas de la entrada.
- R es el conjunto de reglas que relacionan la entrada con la salida.
- S es el conjunto de variables lingüísticas de la salida.

Las variables lingüísticas están formadas por etiquetas lingüísticas que facilitan asociar un significado lingüístico a subintervalos en el dominio de cada variable. Una variable puede tener una o mas etiquetas lingüísticas, matemáticamente estas se representan por funciones de membresía y en consecuencia por conjuntos difusos.

La relación entre las variables de entrada y salida se hace por medio de reglas de la forma Si-Entonces, en este artículo se asume que el sistema difuso puede tener una o mas entradas y solo una salida, por lo tanto las reglas tienen la forma:

$$R_i : \begin{array}{l} \text{Si } x_1 \text{ Es } A_{k1}, x_2 \text{ Es } A_{k2}, \dots, x_n \text{ Es } A_{kh}, \\ \text{Entonces } y_m \text{ Es } B_{pj}, \end{array} \quad (4)$$

donde $R_i \subset R$ representa las reglas, en este caso de tipo Mamdani, debido que relaciona conjuntos difusos en la entrada

con conjuntos difusos a la salida. Las variables $x_n \in E$ son las variables lingüísticas de la entrada, A_{kh} son las etiquetas de las variables de entrada, $y_m \in S$ son las variables de salida y B_{pj} sus respectivas etiquetas.

El proceso de inferencia utilizado en este artículo toma el mínimo de los conjuntos difusos (que se forman mediante las activaciones de la reglas) y el máximo al operar los conjuntos que están a las salidas de las reglas formando un conjunto difuso que se somete a la etapa de defusificación [19].

III. DESCRIPCIÓN DEL SISTEMA Y PLANTEAMIENTO DEL PROBLEMA

Considérese el sistema

$$\ddot{\theta} + \varphi \dot{\theta} + \lambda \theta = u, \quad (5)$$

donde $\varphi, \lambda \in \mathbb{R}$ son constantes fijas, θ representa posición, $\dot{\theta} \in \mathcal{C}^1$ es la variable de salida y representa velocidad, $\ddot{\theta}$ representa aceleración y u es la señal de entrada del sistema.

Se realiza el cambio de coordenadas

$$x_1 = \theta \quad (6)$$

y

$$x_2 = \dot{\theta} \quad (7)$$

para representar el sistema (5) en el modelo de espacio de estados

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\lambda & \varphi \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u, \quad (8)$$

$$y = (0 \ 1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \quad (9)$$

El objetivo es acotar la salida y del sistema mediante el acotamiento del error

$$\xi = r - y, \quad (10)$$

del sistema de control en lazo cerrado, donde r es una señal de referencia; esto con la finalidad de que dicho error permanezca dentro de dos funciones exponenciales

$$h_i = \pm \beta e^{-\alpha t} \quad (11)$$

definidas por el usuario, donde con $i = 1, 2$ y $\beta, \alpha \in \mathbb{R}^+$. Es decir,

$$h_2 \leq \xi \leq h_1. \quad (12)$$

IV. DISEÑO DE LA METODOLOGÍA

A continuación, en la Sección IV-A se presentan las condiciones que se deben satisfacer para lograr el acotamiento de soluciones según se presenta en la Sección IV-B.

A. Condiciones

Sea el error (10) de forma exponencial.

En caso de no serlo, se induce este comportamiento definiendo r como:

$$r = a + \delta e^{-\gamma t}, \quad (13)$$

donde $a \in \mathbb{R}$ es una compensación cuyo valor es el objetivo de regulación para el sistema (8)-(9) y, $\delta e^{-\gamma t}$, con $\delta, \gamma \in \mathbb{R}$, es el comportamiento exponencial a inducir a (10).

Nótese que definiendo r como en (13), el error (10) tendrá un comportamiento exponencial si se resuelve el problema de control que tiene como objetivo

$$\lim_{t \rightarrow \infty} \|\xi\| = 0. \quad (14)$$

B. Acotamiento

Con el error (10) con un comportamiento exponencial, ya sea de forma natural o inducido por la definición (13) para r y resolviendo (14), es posible acotarlo.

El acotamiento se lleva a cabo extrayendo los parámetros α y β de (11), generando con ellos una ganancia $\alpha\beta$ para escalar la salida del control u , por lo que la nueva ley de control se expresa como:

$$\tilde{u} = |\alpha\beta|u. \quad (15)$$

Así, (5) puede escribirse como

$$\ddot{\theta} + \varphi\dot{\theta} + \lambda\theta = \tilde{u}, \quad (16)$$

y (8)-(9) como

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\lambda & \varphi \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tilde{u}, \quad (17)$$

$$y = (0 \ 1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \quad (18)$$

V. RESULTADOS NUMÉRICOS

Para verificar la efectividad de la metodología propuesta, se obtuvieron resultados numéricos mediante la simulación en un motor de corriente directa definido en [20] como:

$$\frac{d^2 q_m}{dt^2} = \frac{1}{J_m} \left(K_a i_a - f_m \frac{dq_m}{dt} \right), \quad (19)$$

$$\frac{di_a}{dt} = \frac{1}{L_a} \left(-R_a i_a + V - K_b \frac{dq_m}{dt} \right), \quad (20)$$

donde V representa el voltaje de entrada, es decir, entrada de control. Los parámetros del motor están dados en la Tabla I.

TABLE I
PARÁMETROS DEL MOTOR DE CORRIENTE DIRECTA.

Constante	Parámetro	Valor
J_m	Momento de inercia del motor	0.01 kg · m ²
f_m	Fricción viscosa	0.1 N · m · s
K_b	Fuerza electromotriz	0.01 v · s/rad
K_a	Torque	0.01 N · m/Amp
R_a	Resistencia eléctrica	1 Ohm
L_a	Inductancia eléctrica	0.5 H

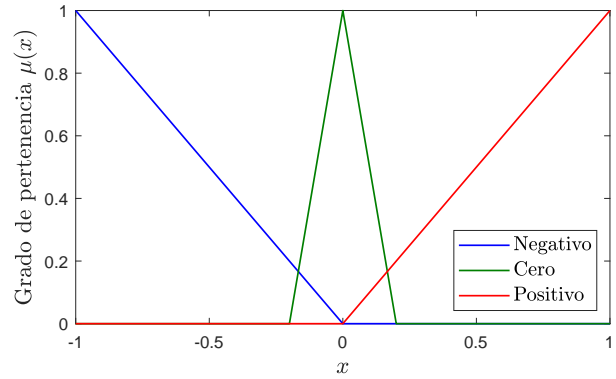


Fig. 1. Funciones de membresía para la variable de entrada y de salida del controlador difuso tipo P.

Considérese el sistema (19)-(20), mismo que escrito de la forma de (8)-(9) queda como

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -\frac{f_m}{J_m} & \frac{K_a}{J_m} \\ \frac{K_b}{L_a} & -\frac{R_a}{L_a} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{L_a} \end{pmatrix} V, \quad (21)$$

$$y = (0 \ 1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \quad (22)$$

Considérese como objetivo de control alcanzar una velocidad de 2 rad/s . A continuación se presentan resultados (i) considerando el problema clásico de regulación en la Sección V-A, (ii) forzando la trayectoria del sistema de errores en la Sección V-B, y (iii) varios casos forzando y acotando la trayectoria del error en la Sección V-C.

A. Problema de clásico de regulación

Esta configuración trata sobre la solución del problema que tiene como objetivo de control a (14) con $r = 2 \text{ rad/s}$.

Para resolver dicho problema es necesario diseñar un esquema de control que logre dicho cometido, para propósito de ejemplificar, en este artículo se propone un controlador difuso proporcional.

La base de reglas que definen al controlador difuso proporcional es

R_1 : Si ξ Es *negativo* Entonces u Es *negativo*,

R_2 : Si ξ Es *cero* Entonces u Es *cero*,

R_3 : Si ξ Es *positivo* Entonces u Es *positivo*,

donde ξ es la variable de entrada, y u es la variable de salida que representa la señal de control, que para efectos del sistema (21)-(22), $V = u$.

Las funciones de membresía consideradas tanto para la variable de entrada como para la variable de salida son de tipo triangular, y pueden verse en la Fig. 1.

La trayectoria para la salida y es la mostrada la Fig. 2(a), mientras que en la Fig. 2(b) se muestra la trayectoria del error. Se puede apreciar que el error no tiene un comportamiento

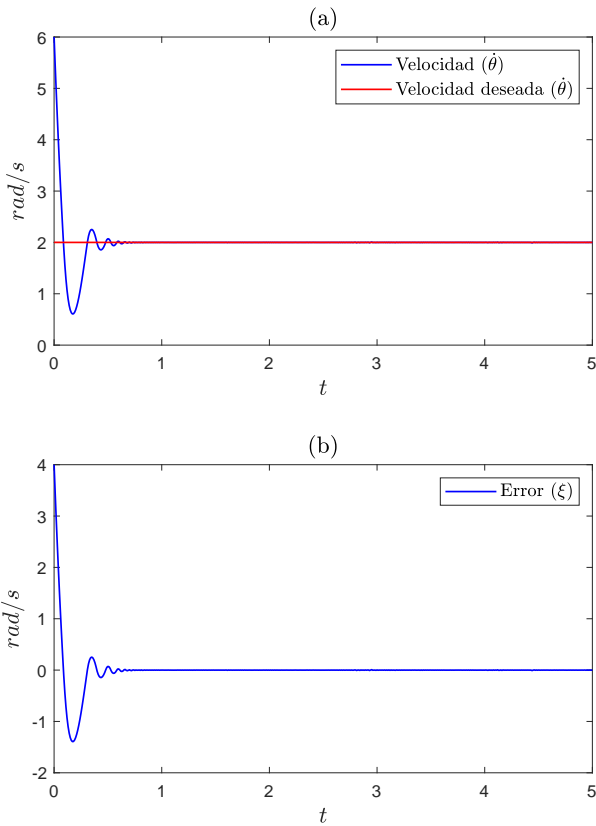


Fig. 2. (a) Regulación de velocidad del motor de CD con control proporcional sin modificar la trayectoria del error. (b) Error de velocidad sin comportamiento exponencial.

exponencial, por lo cual se puede usar la metodología expuesta en la Sección IV para inducir a que lo tenga.

B. Resultado forzando el error

Considérense los resultados obtenidos en la Sección V-A.

Considérese que se desea que el error manifieste un perfil como el de (13) con $\delta = -2$ y $\gamma = 1$.

Nótese que el problema ahora es lograr el objetivo de control (14) para (13) con $\delta = -2$ y $\gamma = 1$, es decir

$$r = (2 - e^{-t})\text{rad/s}. \quad (23)$$

Para lograr dicho objetivo de control, considérese el mismo controlador difuso proporcional diseñado en la Sección V-A.

En la Fig. 3(b) se puede observar que la trayectoria del error tiene un comportamiento de tipo exponencial, forzando a que la salida y del sistema tenga también dicho comportamiento, sin dejar de resolver el problema de regulación.

Nótese que este resultado ya cumple con las condiciones establecidas en la Sección IV-A.

C. Resultados forzando y acotando el error

Considérense los resultados de la Sección V-B, mismos que ya cumplen las condiciones establecidas en la Sección IV-A.

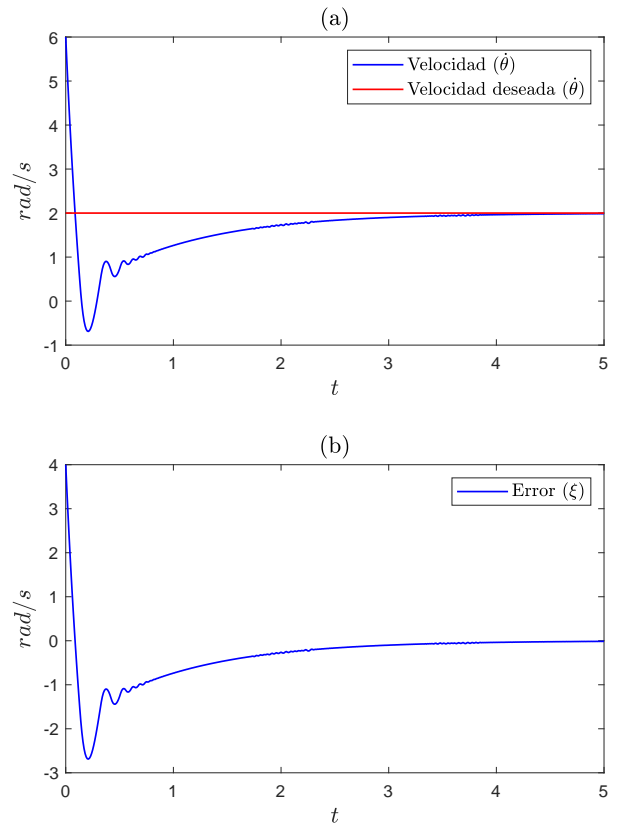


Fig. 3. (a) Regulación de velocidad del motor de CD utilizando un controlador difuso proporcional para forzar al error a seguir una trayectoria exponencial. (b) Error de velocidad con comportamiento exponencial.

Considérese que se desea que el error de los resultados obtenidos en la Sección V-B permanezcan acotados entre dos funciones exponenciales como las definidas en (11) con $\beta = \pm 4$ y $\alpha = 0.5$.

Entonces, la ley de control usada en la Sección V-B debe modificarse de tal forma que ahora se debe usar la ley de control definida en (15) con $\beta = \pm 4$ y $\alpha = 0.5$, es decir

$$\tilde{u} = |(\pm 4)(0.5)|u = 2u. \quad (24)$$

En la Fig. 4(b) se puede observar que una vez que el error ingresa a la zona de acotamiento, este permanece en dicha zona permanentemente, mientras que en la Fig. 4(a) se puede observar la salida del sistema, observándose que el problema de regulación es resuelto.

Nótese que el error puede ser acotado en zonas tan estrictas o tan laxas como se desee. Con la finalidad de mostrar que el esquema propuesto permite ajustar la configuración a las necesidades del usuario. Un caso con cotas más estrictas puede ser considerando $\beta = \pm 2.3$ y $\alpha = 0.75$, por lo que los valores de la señal de referencia se ajustan a $\delta = -0.75$ y $\gamma = 8$. La Fig. 5(a) muestra la salida del sistema, mismo que alcanza el objetivo de control en menor tiempo, mientras que en la Fig.

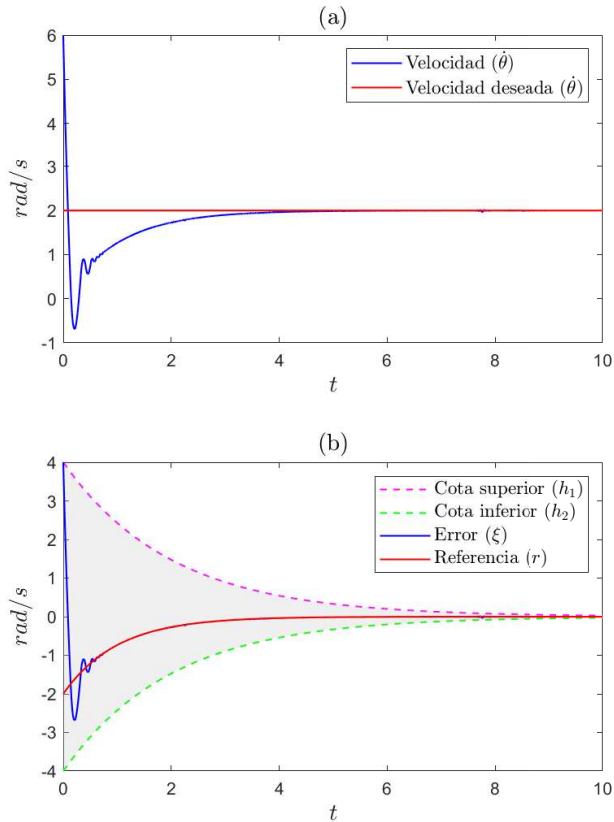


Fig. 4. (a) Regulación de velocidad del motor de CD utilizando un controlador difuso proporcional para forzar al error a seguir una trayectoria exponencial y acotando dicho error. (b) Error de velocidad acotado entre las funciones exponenciales con parámetros $\beta = \pm 4$ y $\alpha = 0.5$.

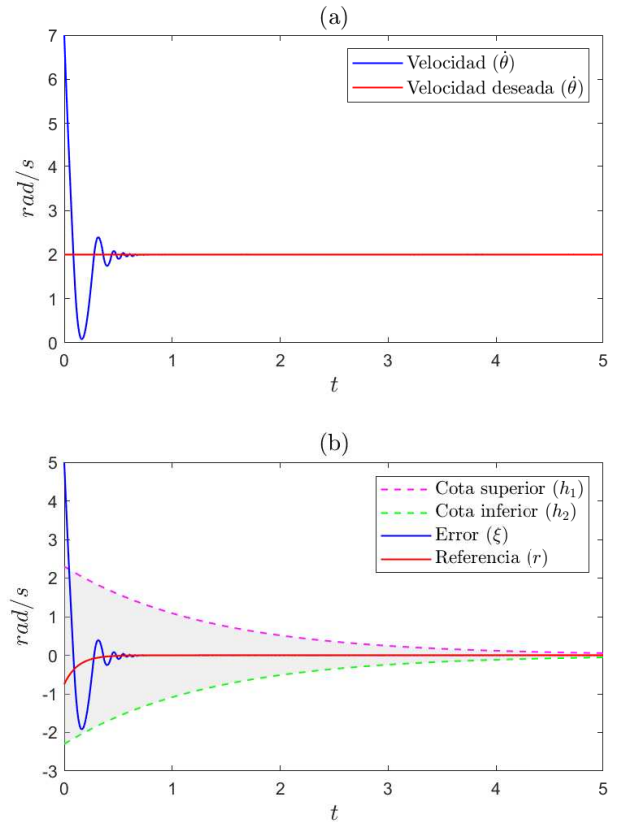


Fig. 5. (a) Regulación de velocidad del motor de CD utilizando un controlador difuso proporcional para forzar al error a seguir una trayectoria exponencial y acotando dicho error. (b) Error de velocidad acotado entre las funciones exponenciales con parámetros $\beta = \pm 2.3$ y $\alpha = 0.75$.

5(b) se muestra el error del sistema, mismo que se mantiene contenido dentro de la zona de acotamiento.

Otro ejemplo puede observarse en la Fig. 6, en la cual puede ser observado en el cual la zona de acotamiento es mas laxo, lo cual se logra considerando $\beta = \pm 5$ y $\alpha = 2$, por lo cual, se ajustan los parámetros de la señal de referencia como $\delta = 3$ y $\gamma = 6$. En la Fig. 6(b) puede apreciarse la trayectoria del error.

Como último ejemplo, en la Fig. 7 se aprecian las trayectorias de los sistemas considerando $\beta = \pm 1$ y $\alpha = 1$, por lo que la trayectoria exponencial se ajusta usando los valores $\delta = -0.75$ y $\gamma = 2$.

VI. CONCLUSION

La metodología propuesta crea un sistema en lazo cerrado que resuelve el problema de control como un seguimiento de trayectorias, ya que el mecanismo de compensación introducido en la señal de referencia permite satisfacer el requisito de mantener un comportamiento en forma exponencial, y el controlador difuso diseñado que funge como controlador proporcional dirige la evolución de las trayectorias de los

errores de seguimiento mostrando que permanecen dentro de una región acotada por funciones de tipo exponencial.

La rapidez de convergencia del comportamiento del sistema de errores esta relacionado con la selección de las ganancias de acotamiento que se aplican a la señal de control que arroja el controlador difuso, por lo que motiva un desarrollo a futuro dedicado al análisis de las ganancias tal que preserve el comportamiento acotado deseado.

REFERENCES

- [1] L. F. Escobar-Dávila, O. D. Montoya-Giraldo, and D. Giraldo-Buitrago, "Control global del péndulo de furuta empleando redes neuronales artificiales y realimentación de variables de estado," *TecnoLógicas*, pp. 71–94, 2013.
- [2] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.
- [3] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," in *Proceedings of the institution of electrical engineers*, vol. 121, no. 12. IET, 1974, pp. 1585–1588.
- [4] L. Holmblad and J. Ostergaard, "Control of cement kiln by fuzzy logic," *Fuzzy Inf and Decis Processes*, 01 1982.
- [5] F. Fujitec, "Flex-8800 series elevator group control system," *Fujitec Co., Ltd., Osaka, Japan*, vol. 433, 1988.

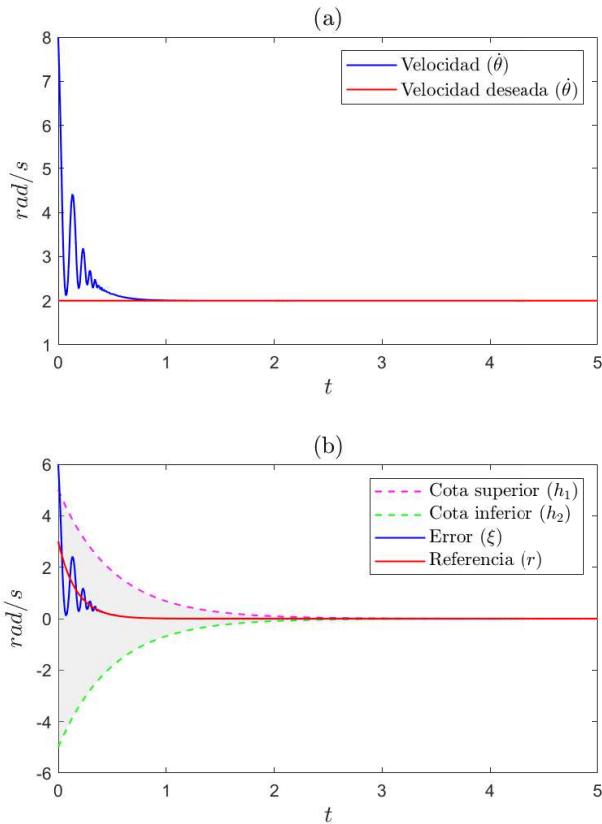


Fig. 6. (a) Regulación de velocidad del motor de CD utilizando un controlador difuso proporcional para forzar al error a seguir una trayectoria exponencial y acotando dicho error. (b) Error de velocidad acotado entre las funciones exponenciales con valores $\beta = \pm 5$ y $\alpha = 2$.

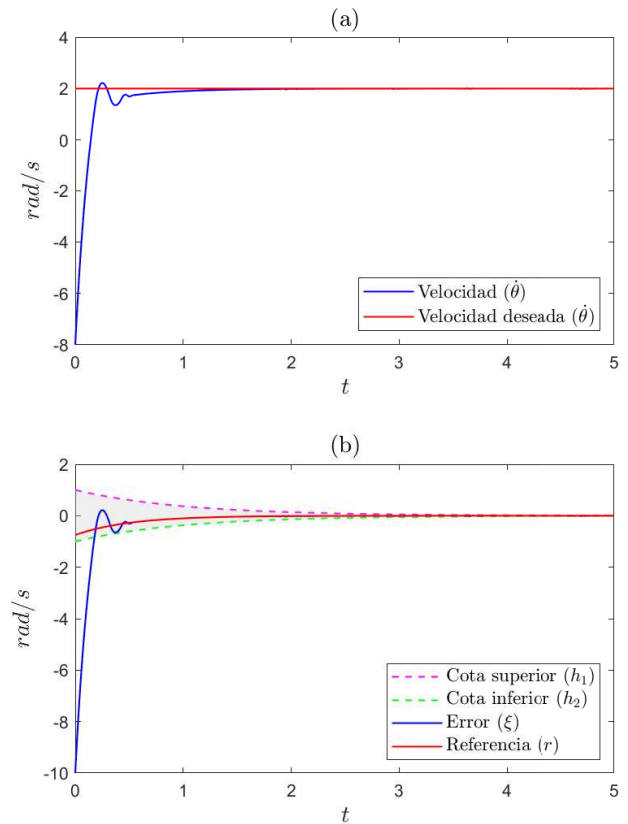


Fig. 7. (a) Regulación de velocidad del motor de CD utilizando un controlador difuso proporcional para forzar al error a seguir una trayectoria exponencial y acotando dicho error. (b) Error de velocidad acotado entre las funciones exponenciales con valores $\beta = \pm 1$ y $\alpha = 1$.

- [6] S. Yasunobu, S. Miyamoto, and H. Ihara, "Fuzzy control for automatic train operation system," *IFAC Proceedings Volumes*, vol. 16, no. 4, pp. 33–39, 1983.
- [7] S. Yasunobu, S. Sekino, and T. Hasegawa, "Automatic train operation and automatic crane operation systems based on predictive fuzzy control," in *Proc. 2nd IFSA Congress*, 1987, pp. 835–838.
- [8] Y. Kasai and Y. Morimoto, "Electronically controlled continuously variable transmission (ecvt-ii)," in *International Congress on Transportation Electronics*, IEEE, 1988, pp. 33–42.
- [9] J. A. Bernard, "Use of a rule-based system for process control," *IEEE Control Systems Magazine*, vol. 8, no. 5, pp. 3–13, 1988.
- [10] O. Yagishita, "Application of fuzzy reasoning to the water purification process," *Industrial applications of fuzzy control*, pp. 19–40, 1985.
- [11] O. Itoh, K. Gotoh, T. Nakayama, and S. Takamizawa, "Application of fuzzy control to activated sludge process," in *Proc. 2nd IFSA Congress*, 1987, pp. 282–285.
- [12] A. Ajorlou, A. Momeni, and A. G. Aghdam, "A class of bounded distributed control strategies for connectivity preservation in multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2828–2833, 2010.
- [13] W. Wang and S. Tong, "Adaptive fuzzy bounded control for consensus of multiple strict-feedback nonlinear systems," *IEEE transactions on cybernetics*, vol. 48, no. 2, pp. 522–531, 2017.
- [14] J. R. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.
- [15] Y. Bai and D. Wang, "Fundamentals of fuzzy logic control – fuzzy sets, fuzzy rules and defuzzifications," in *Advanced Fuzzy Logic Technologies in Industrial Applications*, Y. Bai, H. Zhuang, and D. Wang, Eds. Springer London, 2006, pp. 17–36.
- [16] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985.
- [17] J. Li, L. Yang, Y. Qu, and G. Sexton, "An extended takagi–sugeno–kang inference system (tsk+) with fuzzy interpolation and its rule base generation," *Soft Computing*, vol. 22, no. 10, pp. 3155–3170, 2017.
- [18] Y. Gorrín-Ortega, N. R. Cazarez-Castro, S. L. Cárdenas-Maciél, and J. A. López-Rentería, "Coeficiente en f_R basado en sistema difuso para ecuaciones diferenciales difusas," in *Robótica y Computación. Investigación y Desarrollo*, I. Castro-Liera and M. Cortés-Larrinaga, Eds. Instituto Tecnológico de La Paz, 05 2019, pp. 158–163.
- [19] J. H. Lilly, *Fuzzy Control and Identification*. Wiley, 2010.
- [20] R. Kelly, V. Santibañez, and A. Loria, *Control of Robot Manipulators in Joint Space*. Springer, 2005.

Solución a la cinemática directa con cuaterniones duales de un péndulo doble y simulación gráfica en un sistema embebido

Gerardo Hernandez Valencia
 División de Robótica
 UPTex
 Texcoco, México
 ghernandezv0402@gmail.com

Angel David Barrera Garcia
 División de Robótica
 UPTex
 Texcoco, México
 angel.barrera@uptex.edu.mx

Dolores Carina Gonzalez Barrera
 División de Mecatrónica
 UPVM
 Tultitlán, México
 lolagonzalez7185@gmail.com

Luis Rodriguez Mendez
 División de Robótica
 UPTex
 Texcoco, México
 luisrodriguezmenendez@gmail.com

Jacqueline Soto Hernández
 División de Robótica
 UPTex
 Texcoco, México
 sotohernandezjacque@gmail.com

Resumen— En este trabajo se presenta la simulación gráfica de un mecanismo planar de dos grados de libertad con articulaciones rotacionales en un sistema embebido basado en un Arduino ATmega2560 y un display de 7 pulgadas TFT SSD1963. Los sistemas de coordenadas se asignaron de acuerdo a la convención de Denavit-Hartenberg (D-H) y se obtuvo con cuaterniones duales unitarios la parametrización de la posición y orientación relativa entre los sistemas de coordenadas adyacentes, para posteriormente por medio de la multiplicación entre cuaterniones duales obtener el modelo cinemático directo que permite conocer la posición y orientación del extremo del mecanismo a partir de las variables articulares. Finalmente se realizó una transformación de cuaternion dual a matriz de transformación homogénea para comparar y validar la posición y orientación resultantes.

Palabras clave—Cuaternion dual, Cinemática directa, Sistema embebido

I. INTRODUCCIÓN

La cinemática de un robot se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo, y en particular, por las relaciones entre la posición y orientación del extremo del robot con respecto a los valores que toman sus coordenadas articulares [1].

Denavit-Hartenberg [2] propusieron un método sistemático para describir la geometría espacial de los elementos de una cadena cinemática con respecto a un sistema de referencia fijo, de tal manera, que una vez asignados los sistemas de coordenadas convenientemente, se obtienen las relaciones entre cada sistema de coordenadas adyacentes por medio de la matriz de transformación homogénea (1), la cual depende de cuatro parámetros: " θ ", " α ", " a " y " d " asociados a cada eslabón.

$$A_i^{i-1} = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Donde: $c = \text{coseno}$, $s = \text{seno}$. La matriz del robot T_n^0 se obtiene multiplicando las matrices.

$$T_n^0 = A_1^0 A_2^1 \dots A_n^{n-1} \quad (2)$$

La matriz resultante del robot se define de manera general en [3], y especifica la posición (vector " d ") y orientación (vectores " n ", " s " y " a ") del extremo con respecto a la base.

$$T_n^0 = \begin{bmatrix} n & s & a & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Un cuaternion es una extensión de un número complejo el cual puede ser definido como $q = a + bi + cj + dk$, donde a, b, c, d son números reales e i, j, k son las tres unidades imaginarias [4], y se tiene que $i^2 = j^2 = k^2 = -1$. La norma de q se define por $\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2}$, y su conjugado como $q^* = a - bi - cj - dk$. Para el caso en que $\|q\| = 1$, el cuaternion es unitario y se cumple $q^{-1} = q^*$. El producto de dos cuaterniones $q_0 = a_0 + b_0i + c_0j + d_0k$ y $q_1 = a_1 + b_1i + c_1j + d_1k$, se da por medio de la siguiente ecuación.

$$\begin{aligned} q_0 \cdot q_1 = & a_0 * a_1 - b_0 * b_1 - c_0 * c_1 - d_0 * d_1 \\ & + (a_0 * b_1 + b_0 * a_1 + c_0 * d_1 - d_0 * c_1)i \\ & + (a_0 * c_1 - b_0 * d_1 + c_0 * a_1 - d_0 * b_1)j \\ & + (a_0 * d_1 + b_0 * c_1 - c_0 * b_1 + d_0 * a_1)k \end{aligned} \quad (4)$$

Un número dual es definido como $\hat{a} = a + \epsilon b$ con $\epsilon^2 = 0$ pero $\epsilon \neq 0$, donde a y b son números reales y ϵ es nilpotente, de manera que, un cuaternion dual se define por $\hat{q} = q_r + \epsilon q_d$, donde q_r y q_d son cuaterniones, llamados la parte real y la parte dual respectivamente [5]. El conjugado de un cuaternion dual es $\hat{q}^* = q_r^* + \epsilon q_d^*$. Si $\|\hat{q}\|^2 = 1 + \epsilon 0$, entonces se tiene un cuaternion dual unitario para el cual su inverso es igual a su conjugado $\hat{q}^{-1} = \hat{q}^*$. El producto de dos cuaterniones duales $\hat{q}_0 = q_{r0} + \epsilon q_{d0}$ y $\hat{q}_1 = q_{r1} + \epsilon q_{d1}$ se obtiene por medio de la siguiente ecuación.

$$\hat{q}_0 \cdot \hat{q}_1 = q_0 \cdot q_{r1} + \epsilon(q_{r0} \cdot q_{d1} + q_{d0} \cdot q_{r1}) \quad (5)$$

Un cuaternion unitario puede ser usado para describir la rotación de un sistema de coordenadas sobre un vector unitario \mathbf{n} con un ángulo $|\theta|$.

$$q = \left[\cos\left(\frac{|\theta|}{2}\right), \sin\left(\frac{|\theta|}{2}\right) \mathbf{n} \right] \quad (6)$$

Por otro lado, el cuaternion dual unitario permite representar rotaciones y traslaciones de manera simultánea, de manera que una rotación del cuaternion q seguida de una traslación del cuaternion p , se representa de la siguiente manera.

$$\hat{q} = q + \frac{\epsilon}{2} q \cdot p \quad (7)$$

En la ecuación (8) se presenta la parametrización de la posición y orientación relativa entre dos sistemas de coordenadas adyacentes con cuaterniones duales unitarios [6].

$$DQ_i^{i-1} = \begin{bmatrix} \cos\frac{\theta_i}{2} * \cos\frac{\alpha_i}{2} \\ \cos\frac{\theta_i}{2} * \sin\frac{\alpha_i}{2} * i \\ \sin\frac{\theta_i}{2} * \sin\frac{\alpha_i}{2} * j \\ \sin\frac{\theta_i}{2} * \cos\frac{\alpha_i}{2} * k \\ \left(\frac{-d_i}{2} * \sin\frac{\theta_i}{2} * \cos\frac{\alpha_i}{2} - \frac{a_i}{2} * \cos\frac{\theta_i}{2} * \sin\frac{\alpha_i}{2}\right) * \epsilon \\ \left(\frac{-d_i}{2} * \sin\frac{\theta_i}{2} * \sin\frac{\alpha_i}{2} + \frac{a_i}{2} * \cos\frac{\theta_i}{2} * \cos\frac{\alpha_i}{2}\right) * i\epsilon \\ \left(\frac{d_i}{2} * \cos\frac{\theta_i}{2} * \sin\frac{\alpha_i}{2} + \frac{a_i}{2} * \sin\frac{\theta_i}{2} * \cos\frac{\alpha_i}{2}\right) * j\epsilon \\ \left(\frac{d_i}{2} * \cos\frac{\theta_i}{2} * \cos\frac{\alpha_i}{2} - \frac{a_i}{2} * \sin\frac{\theta_i}{2} * \sin\frac{\alpha_i}{2}\right) * k\epsilon \end{bmatrix} \quad (8)$$

Donde: θ_i, α_i, a_i y d_i representan los parámetros de D-H.

De manera análoga a la ecuación (2), el cuaternion dual que relaciona el extremo del péndulo doble con la base, se obtiene multiplicando los cuaterniones duales que relacionan a cada par de sistemas de coordenadas adyacentes de acuerdo a:

$$DQ_n^0 = DQ_1^0 \cdot DQ_2^1 \dots DQ_n^{n-1} \quad (9)$$

Finalmente, para validar los resultados de posición y orientación se realiza una transformación de cuaternion dual a matriz de transformación homogénea [7] por medio de las ecuaciones (10) y (11).

$$DQ_n^0 = [q_x \quad q_y i \quad q_z j \quad q_w k \quad q_{x\epsilon} \quad q_{y\epsilon} \quad q_{z\epsilon} \quad q_{w\epsilon} \epsilon]^T$$

$$T_n^0 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Donde:

$$\begin{aligned} n_x &= 1 - 2(q_y^2 + q_z^2) \\ s_x &= 2(q_x q_y - q_z q_w) \\ a_x &= 2(q_x q_z + q_y q_w) \\ p_x &= 2(-q_x q_{w\epsilon} + q_w q_{x\epsilon} - q_z q_{y\epsilon} + q_y q_{z\epsilon}) \\ n_y &= 2(q_x q_y + q_z q_w) \\ s_y &= 1 - 2(q_x^2 + q_z^2) \\ a_y &= 2(q_y q_z - q_x q_w) \\ p_y &= 2(-q_y q_{w\epsilon} + q_z q_{x\epsilon} + q_w q_{y\epsilon} - q_x q_{z\epsilon}) \\ n_z &= 2(q_x q_z - q_y q_w) \\ s_z &= 2(q_y q_z + q_x q_w) \\ a_z &= 1 - 2(q_x^2 + q_y^2) \\ p_z &= 2(-q_z q_{w\epsilon} - q_y q_{x\epsilon} + q_x q_{y\epsilon} + q_w q_{z\epsilon}) \end{aligned} \quad (11)$$

II. CINEMÁTICA DIRECTA CON CUATERNIONES DUALES

En primer lugar, se asignaron los sistemas de coordenadas (Fig.1) para el péndulo doble y se obtuvieron sus parámetros de Denavit-Hartenberg (Tab.1).

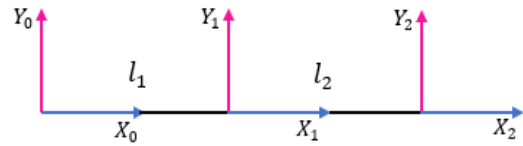


Fig. 1. Sistemas de coordenadas asignados para el péndulo doble.

TABLE I.

Parámetros de D.H.				
i	θ	α	a	d
1	q_1	0	l_1	0
2	q_2	0	l_2	0

Los parámetros de la Tab.1 se sustituyen en (8) y se determinan los cuaterniones duales DQ_1^0 y DQ_2^1 para cada eslabón.

$$DQ_1^0 = \begin{bmatrix} \cos\left(\frac{q_1}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{q_1}{2}\right) \\ 0 \\ \frac{l_1 \cdot \cos\left(\frac{q_1}{2}\right)}{2} \\ \frac{l_1 \cdot \sin\left(\frac{q_1}{2}\right)}{2} \\ 0 \end{bmatrix} \quad (12)$$

$$DQ_2^1 = \begin{bmatrix} \cos\left(\frac{q_2}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{q_2}{2}\right) \\ 0 \\ \frac{l_1 \cdot \cos\left(\frac{q_2}{2}\right)}{2} \\ \frac{l_1 \cdot \sin\left(\frac{q_2}{2}\right)}{2} \\ 0 \end{bmatrix} \quad (13)$$

Estos cuaterniones duales se multiplican de acuerdo a (9) para encontrar DQ_2^0 , el resultado representa el modelo cinemático directo del péndulo doble, el cual depende de las variables articulares y las longitudes de los eslabones.

$$DQ_2^0 = \begin{bmatrix} \cos\left(\frac{q_1+q_2}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{q_1+q_2}{2}\right) \\ 0 \\ \frac{l_1 \cdot \cos\left(\frac{q_1-q_2}{2}\right) + l_2 \cdot \cos\left(\frac{q_1+q_2}{2}\right)}{2} \\ \frac{l_1 \cdot \sin\left(\frac{q_1-q_2}{2}\right) + l_2 \cdot \sin\left(\frac{q_1+q_2}{2}\right)}{2} \\ 0 \end{bmatrix} \quad (14)$$

Para el caso en el que $q_1 = q_2 = 0$ y $l_1 = l_2 = 1$, se obtiene el cuaternión dual numérico siguiente:

$$DQ_2^0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (15)$$

Sustituyendo en (10) y (11) se obtiene la matriz de transformación homogénea equivalente, donde se verifica que la matriz de rotación que se obtuvo es igual a la matriz identidad, lo cual concuerda con la Fig. 2 donde se tiene la misma orientación entre el sistema de coordenadas del extremo s_2 y el

de la base s_0 , en el caso de la posición solo se tiene desplazamiento a lo largo del eje x, lo que equivale a la suma de $l_1 + l_2$, resultado que concuerda con el valor del elemento p_z de la matriz T_2^0 que se muestra a continuación.

$$T_2^0 = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

Para el caso en el que $q_1 = q_2 = 90$ y $l_1 = l_2 = 1$, la configuración de los eslabones es la mostrada en la Fig. 2.

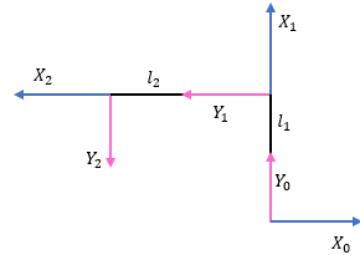


Fig. 2. Configuración del péndulo doble para $q_1 = q_2 = 90$.

Sustituyendo los nuevos valores en (14) se obtiene la posición y orientación del extremo para la nueva configuración.

$$DQ_2^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0.5 \\ 0.5 \\ 0 \end{bmatrix} \quad (17)$$

Realizando la transformación a matriz, por inspección se verifica que concuerda con la relación existente entre los sistemas s_0 y s_2 para los valores de las coordenadas articulares y las longitudes de los eslabones establecidos.

$$T_2^0 = \begin{bmatrix} -1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

III. SIMULACIÓN GRÁFICA EN MATLAB Y ARDUINO

Una vez que se obtuvo el modelo cinemático y que se validaron los resultados de posición y orientación obtenidos por el modelo cinemático basado en cuaterniones duales, se programaron las ecuaciones en una interfaz gráfica de usuario en el software MATLAB, de manera que se calcula y despliegan los resultados de DQ_2^0 y T_2^0 así como también se realiza una simulación gráfica unifilar de los eslabones del péndulo doble de acuerdo a la configuración establecida por medio de las coordenadas articulares cuyos valores varían por medio de dos sliders. En la Fig. 3 y Fig. 4 se muestran los resultados obtenidos para la primera y segunda configuración respectivamente.

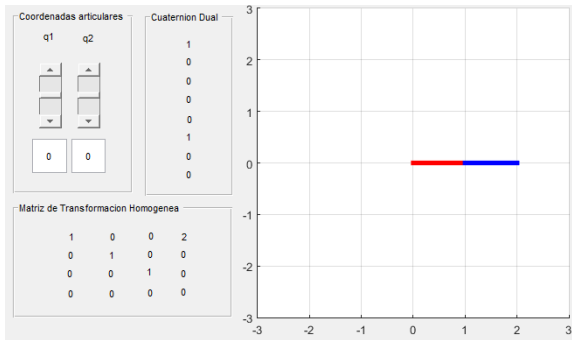


Fig. 3. Resultados de la interfaz cuando $q_1 = q_2 = 0$.

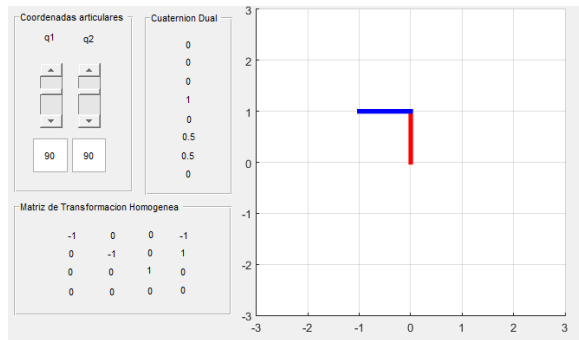


Fig. 4. Resultados de la interfaz cuando $q_1 = q_2 = 90$.

Estas ecuaciones posteriormente se programaron en un sistema embebido basado en Arduino de acuerdo al diagrama de flujo de la fig. 5, se utilizo un Microcontrolador ATmega2560, un módulo LCDTFT01 Arduino Mega Shield, una pantalla LCD TFT SSD1963 y 2 trimpots para variar los valores de los ángulos como se aprecia en la fig.6.

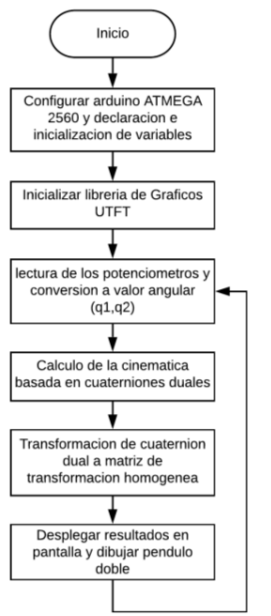


Fig. 5. Diagrama de flujo para la implementación del sistema embebido.

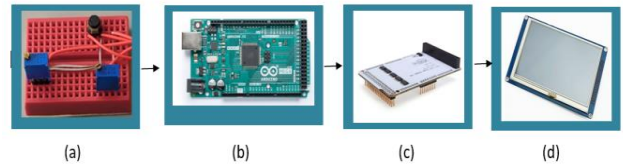


Fig. 6. Componentes del sistema: (a) Trimpots, (b) Arduino ATmega2560, (c) LCDTFT01 Arduino Mega Shield y (d) Pantalla LCD TFTSSD1963.

Finalmente el sistema desarrollado y el despliegue de la información en la pantalla se muestra en la fig. 7, los resultados para la primera y segunda configuración del péndulo doble se ilustran en la Fig. 8 y Fig. 9, respectivamente.

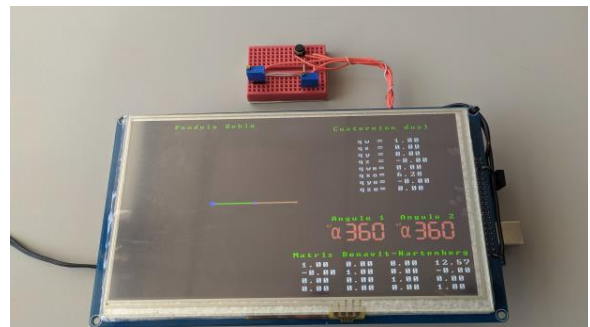


Fig. 7. Simulación gráfica y despliegue de resultados en el sistema embebido.

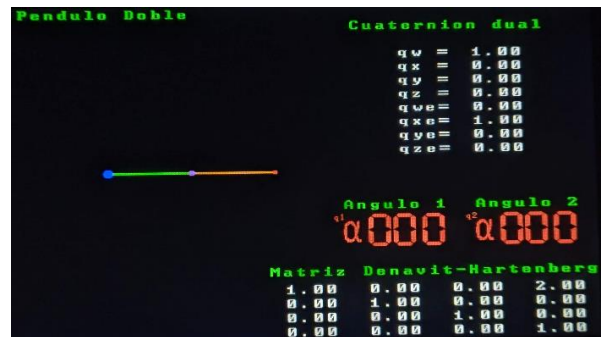


Fig. 8. Resultados para la primera configuración $q_1 = q_2 = 0$.

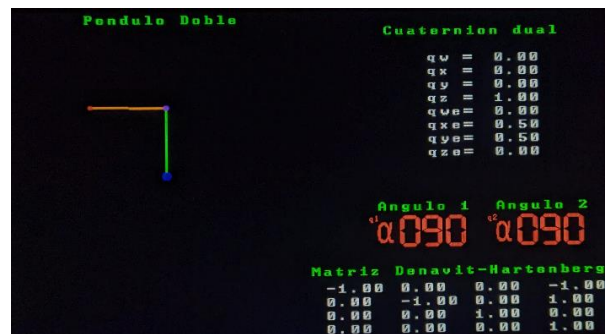


Fig. 9. Resultados para la segunda configuración $q_1 = q_2 = 90$.

IV. CONCLUSIONES

Los cuaterniones duales son una herramienta compacta, robusta y computacionalmente eficiente para representar la posición y orientación, así como, realizar transformaciones sobre marcos de referencia de rotación y traslación. En este trabajo se presentó un modelo matemático basado en cuaterniones duales unitarios de un mecanismo planar de dos grados de libertad con articulaciones rotacionales, el cual emplea los parámetros de Denavit-Hartenberg para parametrizar la posición y orientación relativa entre los sistemas de coordenadas adyacentes. Los resultados de estas ecuaciones se validaron y simularon gráficamente en el sistema embebido como una alternativa viable a las matrices de transformación homogénea que es una de las herramientas matemáticas comúnmente empleadas en robótica y en la simulación gráfica.

REFERENCIAS

- [1] K. Fu, R. Gonzalez, and C. Lee, *Robotica*, 1st ed., Espana: Mc. Graw Hill, 1988.
- [2] J. Denavit and R. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Journal of Applied Mechanics*, pp. 215–221, June 1955.
- [3] W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 1st ed., Oxford: Wiley, 2005, pp.56.
- [4] D. P. Mandic, C. Jahanchahi and C. Cheong, "A Quaternion gradient operator and its applications", *IEEE Signal Processing Letters*, pp. 47–50, 2011.
- [5] X. Wang and C. Yu, "Unit-Dual-Quaternion-Based PID Control Scheme for Rigid-Body Transformation," *Proceedings of the 18th World Congress the International Federation of Automatic Control*, Milano Italy, 2011.
- [6] J. Ramirez, E. Merchan, E. Lugo, R. Rodriguez and G. Urriolagoitia, "Desarrollo de una nueva solucion compacta a la cinematica de manipuladores roboticos basada en cuaterniones duales," *Revista iberoamericana de automatica e informatica industrial*, pp. 334–344, 2011.
- [7] L. Radavelli, R. Simoni and D. Martins, "A comparative study of the kinematics of robots manipulators by Denavit-Hartenberg and Dual quaternion," *Mecanica Computacional Vol. XXXI*, pp.2833–2848, 2012

Modelado, control, y optimización de ganancias de un vehículo aéreo no tripulado

Ing. Merlín Octavio Maravilla Castro
 Instituto Tecnológico de La Paz
 La Paz, Baja California Sur
 Email: merlinomc96@gmail.com

M.S.C. Joel Artemio Morales Viscaya
 División de Estudios de Posgrado e Investigación, ITLP
 La Paz, Baja California Sur
 Email: iscviscaya@gmail.com

Resumen—En este trabajo se desarrolla un modelo matemático de un cuadcóptero, empleando transformaciones lineales basadas en los ángulos de Tait-Bryan para obtener las relaciones cinemáticas, así como la segunda ley de Newton y la ecuación de Euler para cuerpos rígidos para definir el modelo dinámico. Se diseña un controlador LQR para realizar control de posición. Se sintonizan las ganancias del controlador mediante la optimización de dos funciones objetivo, empleando el método de descenso de gradiente. La primera función está definida por la sumatoria de la energía cinética y potencial del vehículo, y la norma del error en cada instante de tiempo. La segunda se define como el producto de la sumatoria anterior y el tiempo. El desempeño de las leyes de control se analiza simulando el sistema por medio de MATLAB, utilizando los parámetros correspondientes al cuadcóptero *Rolling Spider*, de la compañía Parrot, que se encuentran en la documentación de MATLAB.

I. INTRODUCCIÓN

El estudio de vehículos aéreos no tripulados está documentado en libros como el de [1] que, a pesar de enfocarse en los VANT de ala fija, presenta el modelado de las fuerzas aerodinámicas que actúan sobre una aeronave de tamaño pequeño, así como modelos cinemático y dinámico, que pueden adaptarse para un cuadcóptero. El modelo de un cuadcóptero se ha realizado de acuerdo con distintas metodologías, empleando la formulación de Newton-Euler como en [2], [3], [4] o la de Lagrange-Euler como en [5]. La diferencia entre los modelos descritos por estos trabajos radica en la definición de distintos marcos coordinados y las consideraciones que se establecen durante su desarrollo. El control de vehículos de varios rotores se ha estudiado ampliamente. Debido a que los modelos que se obtienen son, por lo general, altamente no lineales, una práctica común consiste en linealizar el sistema respecto al punto de operación y evaluar el desempeño de controladores lineales en una región cercana a este. Con el objetivo de realizar control de actitud, se han empleado distintas variantes del controlador PID [6], [7], mientras que autores como [2], [4], [8], [9] utilizan técnicas de control óptimo, diseñando controladores LQR y LQG. En estos últimos, la sintonización de las ganancias (las matrices Q y R) se realiza de forma experimental, a través de ensayo y error. Un enfoque para llevar a cabo la sintonía de ganancias en controladores es formular dicho problema como uno de optimización, que puede ser abordado con técnicas heurísticas o numéricas. Para ello es necesario definir una función a

optimizar, y las ganancias resultantes producirán un mejor desempeño sólo en términos de dicha función. La utilización de métodos heurísticos en distintos sistemas se presenta en [10], [11], [12]. En [13] se realiza la comparación entre un control PD y un LQR, cuyas ganancias se ajustan mediante un algoritmo genético y el método empírico, respectivamente. En [13] se sugiere que el uso de un método iterativo para realizar la optimización de las ganancias resultaría en un mejor desempeño de los controladores, reduciendo el tiempo de subida y de asentamiento del sistema. En este trabajo se propone abordar la sintonización de un controlador LQR como un problema de optimización, con base en dos índices de desempeño diseñados. Para resolver este problema se utiliza el método de descenso de gradiente. Dado que el algoritmo converge hacia mínimos locales, se inicializa el vector de ganancias con distintos valores para buscar la solución con el mejor desempeño. Aunado a esto, se optimizan las funciones respecto a distintas entradas de referencia, con el propósito de obtener un controlador que sea capaz de satisfacer de forma adecuada distintos objetivos de control. El desempeño de los controladores se evalúa mediante la simulación del modelo no lineal del cuadcóptero, considerando el tiempo de subida, tiempo de asentamiento, y la energía del sistema.

II. MODELO DEL SISTEMA

A. Parámetros del vehículo

El cuadcóptero que se utiliza como base para este proyecto es el modelo *Rolling Spider*, de la línea de Minidrones fabricados por la compañía Parrot, que se muestra en la fig. 1. La documentación de MathWorks respecto a este cuadcóptero define los ejes coordinados de la siguiente manera:

- El origen se encuentra ubicado en el centro de masa del vehículo.
- El eje x es positivo en la dirección de la nariz del cuadcóptero.
- El eje y es positivo hacia la derecha del cuadcóptero.
- El eje z se define de acuerdo con la regla de la mano derecha; es positivo hacia abajo.

El paquete de soporte de Simulink para Minidrones de Parrot [14] define, para el modelo *Rolling Spider*, los parámetros que se muestran en la tabla I.



Fig. 1. Mímidrone Rolling Spider.

 TABLA I
 PARÁMETROS DEL CUADRÓPTERO.

Parámetro	Unidad
Masa	0.068 kg
Ancho	0.090 m
Largo	0.090 m
Altura	0.01587 m
Longitud de los brazos	0.0624 m
Radio de las hélices	0.033 m
Momento de inercia respecto al eje x	$0.0686 \times 10^{-3} \text{ kg m}^2$
Momento de inercia respecto al eje y	$0.092 \times 10^{-3} \text{ kg m}^2$
Momento de inercia respecto al eje z	$0.1366 \times 10^{-3} \text{ kg m}^2$
Factor de empuje (C_T)	0.0107
Factor de torque (C_Q)	7.8264×10^{-3}
Coefficiente de empuje (b) ¹	$4.72 \times 10^{-8} \text{ kg m}$
Coefficiente de arrastre (d) ²	$1.1393 \times 10^{-10} \text{ kg m}^2$

¹ Este coeficiente se relaciona con el factor de empuje por medio de la siguiente expresión [3]:

$$b = C_T \rho A r^2, \quad (1)$$

donde ρ es la densidad del medio, A es el área del círculo generado por la rotación de cada hélice, y r es el radio de las hélices.

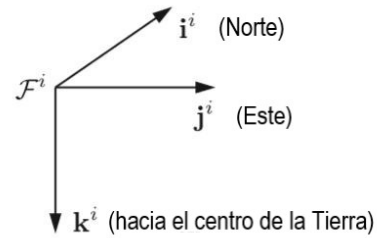
² Este coeficiente se relaciona con el factor de torque por medio de la siguiente expresión [3]:

$$d = C_Q \rho A r^3, \quad (2)$$

donde ρ es la densidad del medio, A es el área del círculo generado por la rotación de cada hélice, y r es el radio de las hélices.

B. Marcos de referencia

Para desarrollar las ecuaciones que describen el movimiento del vehículo, es necesario establecer los sistemas de referencia (también llamados marcos de referencia) en que se definirán las variables relacionadas con su dinámica. Se utilizan principalmente dos sistemas de referencia: un marco inercial fijo en un punto sobre la superficie terrestre y un marco con su origen en el centro de masa del vehículo, con sus ejes desplazados angularmente de acuerdo con la orientación del vehículo. Realizar la transformación de uno a otro requiere la definición de marcos intermedios como resultado de tres rotaciones subsecuentes. La secuencia de rotaciones utilizada para definir la orientación final de un marco coordenado es hasta cierto


 Fig. 2. Marco inercial F^i [1].

punto arbitraria. La rotación inicial podría realizarse en torno a cualquiera de los tres ejes. Sólo se tiene la limitación de que dos rotaciones sucesivas no deben ser realizadas en torno al mismo eje. Existen 12 posibles secuencias de rotación que se dividen en dos grupos:

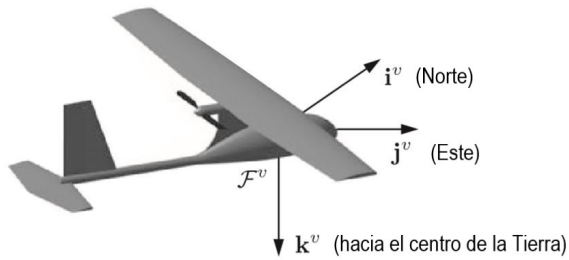
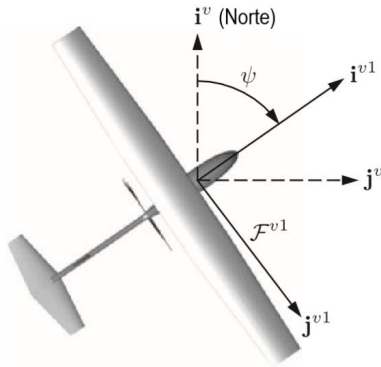
- Ángulos de Euler clásicos: la primera y la tercera rotación de la secuencia se realizan con respecto al mismo eje.
- Ángulos de Tait-Bryan: las rotaciones de la secuencia se realizan con respecto a ejes distintos.

Las dos secuencias de rotaciones más utilizadas pertenecen a los ángulos de Euler: ZXZ y ZYZ, llamadas respectivamente *convención X* y *convención Y*. Su desventaja radica en que cuando la segunda rotación tiene un valor muy pequeño, los ejes respecto a los que se realizan la primera y la tercera rotación se vuelven casi coincidentes, y los ángulos ϕ y ψ se vuelven indistinguibles. Los ángulos de Tait-Bryan no sufren de este problema, por lo que esta convención es utilizada en aplicaciones de ingeniería relacionadas con la orientación de vehículos móviles como satélites y aeronaves [15]. En este trabajo se emplea la convención ZYX; los marcos resultantes se definen a continuación.

1) *Marco inercial F^i* : Este marco se define con su origen en un punto determinado sobre la superficie terrestre. Se desprecia la rotación de la tierra, dado que esta no experimenta aceleraciones considerables, por lo que las fuerzas ficticias ejercidas sobre el vehículo son insignificantes, así como su curvatura. El error absoluto que existe para una distancia de un kilómetro, considerando la curvatura de la tierra y considerándola plana, es de alrededor de un micrómetro. El vehículo se desplaza en trayectorias mucho menores a un kilómetro, por lo que las consideraciones previas simplifican suficiente el estudio del movimiento sin comprometer de forma grave la exactitud.

El marco está formado por los vectores unitarios \mathbf{i}^i , con dirección al norte, \mathbf{j}^i , con dirección al este, y \mathbf{k}^i , que se encuentra dirigido hacia el centro de la tierra. Este sistema de coordenadas, que se muestra en la fig. 2, se conoce como marco NED (por sus siglas en inglés, *north-east-down*).

2) *Marco no inercial F^v* : El origen de este marco se encuentra ubicado en el centro de masa del VANT, mientras que sus ejes están alineados con los ejes del marco inercial F^i , como se observa en la fig. 3. Este marco no es inercial ya que


 Fig. 3. Marco no inercial F^v [1].

 Fig. 4. Transformación del marco F^v al marco F^{v1} [1].

el vehículo puede desplazarse con una aceleración respecto al marco inercial.

3) *Marco no inercial F^{v1}* : El marco F^{v1} se obtiene al rotar en sentido positivo, de acuerdo con la convención de la mano derecha, el marco F^v en torno al eje \mathbf{k}^v una cantidad correspondiente al ángulo de guiñada ψ , como se puede observar en la fig. 4. La transformación de F^v a F^{v1} está dada por:

$$\mathbf{p}^{v1} = R_v^{v1}(\psi)\mathbf{p}^v, \quad (3)$$

donde

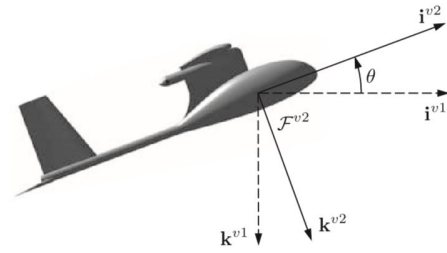
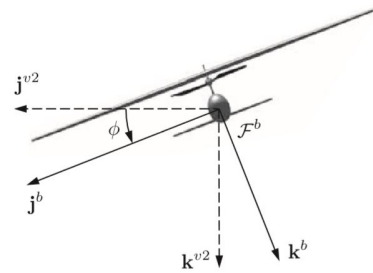
$$R_v^{v1}(\psi) = \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4)$$

4) *Marco no inercial F^{v2}* : El marco F^{v2} se obtiene al rotar en sentido positivo, de acuerdo con la convención de la mano derecha, el marco F^{v1} en torno al eje \mathbf{j}^{v1} una cantidad correspondiente al ángulo de cabeceo θ , como se puede observar en la fig. 5. La transformación de F^{v1} a F^{v2} está dada por:

$$\mathbf{p}^{v2} = R_{v1}^{v2}(\theta)\mathbf{p}^{v1}, \quad (5)$$

donde

$$R_{v1}^{v2}(\theta) = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix}. \quad (6)$$


 Fig. 5. Transformación del marco F^{v1} al marco F^{v2} [1].

 Fig. 6. Transformación del marco F^{v2} al marco F^b [1].

5) *Marco no inercial F^b* : El marco F^b se obtiene al rotar en sentido positivo, de acuerdo con la convención de la mano derecha, el marco F^{v2} en torno al eje \mathbf{i}^{v2} una cantidad correspondiente al ángulo de alabeo ϕ , como se puede observar en la fig. 6. La transformación de F^{v2} a F^b está dada por:

$$\mathbf{p}^b = R_{v2}^b(\phi)\mathbf{p}^{v2}, \quad (7)$$

donde

$$R_{v2}^b(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix}. \quad (8)$$

Debido a las propiedades de las matrices de rotación, podemos realizar la transformación del marco F^v al marco F^b si multiplicamos las matrices asociadas con los marcos intermedios:

$$R_v^b(\phi, \theta, \psi) = R_{v2}^b(\phi)R_{v1}^{v2}(\theta)R_v^{v1}(\psi) = \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{pmatrix}, \quad (9)$$

donde $c_\phi \triangleq \cos \phi$ y $s_\phi \triangleq \sin \phi$.

Para realizar la transformación del marco F^b al marco F^v , se realizan las transformaciones en sentido inverso, lo que equivale a invertir el orden de los productos matriciales. Debido a las propiedades de las matrices de rotación elemental, la inversión en el orden de las matrices produce una matriz resultante que es la transpuesta de la ec. (9):

$$R_b^v(\phi, \theta, \psi) = R_{v1}^v(\psi)R_{v2}^v(\theta)R_b^{v2}(\phi) \\ = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}. \quad (10)$$

C. Modelo cinemático

La cinemática describe el movimiento de los cuerpos sin considerar las fuerzas que lo producen. Para derivar el modelo cinemático del VANT se definen cuatro vectores. El vector $(x \ y \ z)^\top$ describe la posición del centro de masa respecto al marco inercial. La orientación del vehículo se describe mediante el vector $(\phi \ \theta \ \psi)^\top$, que se compone por los ángulos de alabeo, cabeceo, y guiñada, definidos en los marcos F^{v2} , F^{v1} , y F^v , respectivamente. Se definen también un par de vectores $(u \ v \ w)^\top$ y $(p \ q \ r)^\top$ que contiene las velocidades lineales y angulares del vehículo respecto al marco del cuerpo. Los vectores $\mathbf{v} = (\dot{x} \ \dot{y} \ \dot{z})^\top$ y $\mathbf{v}_B = (u \ v \ w)^\top$ se relacionan entre sí por medio de la matriz de la ec. (10):

$$\mathbf{v} = R_b^v \mathbf{v}_B, \quad (11)$$

La transformación entre $\boldsymbol{\omega} = (\dot{\phi} \ \dot{\theta} \ \dot{\psi})^\top$ y $\boldsymbol{\omega}_B = (p \ q \ r)^\top$ no puede realizarse utilizando la matriz de (10), debido a la definición de los elementos que componen a estos vectores. El vector $\boldsymbol{\omega}_B = (p \ q \ r)^\top$ contiene a los componentes del vector de velocidad angular, propiedad que es inherente al marco del cuerpo, mientras que los ángulos ϕ , θ , y ψ son magnitudes que se encuentran definidas en marcos distintos: el ángulo de alabeo ϕ es la rotación del marco F^{v2} al marco F^b , el ángulo de cabeceo θ es la rotación del marco F^{v1} al marco F^{v2} , y el ángulo de guiñada ψ es la rotación del marco F^v al marco F^{v1} . La matriz de transformación que se requiere para relacionar estos vectores no es el producto de matrices de rotación elemental, sino una combinación lineal de los vectores que contienen a cada derivada, donde los coeficientes son las matrices de rotación que relacionan cada marco intermedio con el marco del cuerpo [1]:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_{v2}^b(\phi) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_{v2}^b(\phi)R_{v1}^{v2}(\theta) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\ = \begin{pmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}. \quad (12)$$

Las derivadas de los desplazamientos angulares corresponden a rotaciones que provocan la transformación de un marco a otro, por lo que p es directamente la derivada de ϕ , la derivada de θ sólo requiere transformarse del marco F^{v2} al F^b y la derivada de ψ requiere una transformación desde el marco F^{v1} hasta el F^b . Debido a que esta matriz no se obtuvo como un producto de matrices elementales, la matriz que realiza la transformación en sentido contrario no es la transpuesta de la ec. (12), sino su inversa [1]. De acuerdo con lo anterior, se define la siguiente relación:

$$\boldsymbol{\omega} = T_b^v \boldsymbol{\omega}_B, \quad (13)$$

donde

$$T_b^v = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{pmatrix}. \quad (14)$$

Desarrollando los productos matriciales de la ec. (11) y (13) se obtiene el modelo cinemático del cuadróptero, que se muestra en la ec. (15).

$$\begin{cases} \dot{x} = w[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] - \\ \quad v[c(\phi)s(\psi) - c(\psi)s(\phi)s(\theta)] + u[c(\psi)c(\theta)] \\ \dot{y} = v[c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)] - \\ \quad w[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] + u[c(\theta)s(\psi)] \\ \dot{z} = w[c(\phi)c(\theta)] - u[s(\theta)] + v[c(\theta)s(\phi)] \\ \dot{\phi} = p + r[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\ \dot{\theta} = q[c(\phi)] - r[s(\phi)] \\ \dot{\psi} = r \frac{c(\phi)}{c(\theta)} + q \frac{s(\phi)}{c(\theta)} \end{cases} \quad (15)$$

D. Modelo dinámico

Considerando ahora las fuerzas que producen el movimiento del cuadróptero, se utiliza la ley de Newton aplicada a cuerpos rígidos para describir el movimiento lineal del cuerpo:

$$m(\boldsymbol{\omega}_B \times \mathbf{v}_B + \dot{\mathbf{v}}_B) = \mathbf{f}_B, \quad (16)$$

donde m es la masa del cuadróptero, y $\mathbf{f}_B = (f_x \ f_y \ f_z)^\top \in \mathbb{R}^3$ es la fuerza total, con sus componentes dirigidos a lo largo de los ejes del marco del cuerpo.

Para describir el movimiento rotacional del cuerpo empleamos la ecuación de Euler:

$$\mathbf{I} \cdot \dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times (\mathbf{I} \cdot \boldsymbol{\omega}_B) = \mathbf{m}_B, \quad (17)$$

donde $\mathbf{m}_B = (m_x \ m_y \ m_z)^\top \in \mathbb{R}^3$ es el torque total, con sus componentes definidos en torno a los ejes del marco del cuerpo. Debido a que los ejes principales del cuerpo coinciden con los ejes del marco de referencia, \mathbf{I} es la matriz de inercia diagonal:

$$\mathbf{I} = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \in \mathbb{R}^{3 \times 3}. \quad (18)$$

Desarrollando las expresiones anteriores, se obtiene el modelo que se muestra en la ec. (19). Los términos del lado izquierdo corresponden a las fuerzas y pares que actúan sobre el cuadróptero.

$$\begin{cases} f_x = m(\dot{u} + qw - rv) \\ f_y = m(\dot{v} - pw + ru) \\ f_z = m(\dot{w} + pv - qu) \\ m_x = \dot{p}I_x - qrI_y + qrI_z \\ m_y = \dot{q}I_y + prI_x - prI_z \\ m_z = \dot{r}I_x - pqI_x + pqI_y \end{cases} \quad (19)$$

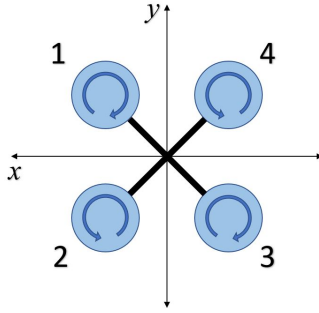


Fig. 7. Configuración en equis.

Las fuerzas que se ejercen sobre el vehículo, despreciando aquellas producidas por el viento, están dadas por:

$$\mathbf{f}_B = mgR_v^b \hat{\mathbf{e}}_z - f_t \hat{\mathbf{e}}_3, \quad (20)$$

donde $\hat{\mathbf{e}}_z$ es un vector unitario sobre el eje z del marco inercial, $\hat{\mathbf{e}}_3$ es un vector unitario sobre el eje z del marco del cuerpo, g es la aceleración gravitacional, y f_t es el empuje total generado por los rotores.

Los pares que se ejercen sobre el cuadricóptero, despreciando aquellos producidos por el viento, así como el efecto giroscópico [4], están dados por:

$$\mathbf{m}_B = \boldsymbol{\tau}_B, \quad (21)$$

donde $\boldsymbol{\tau}_B = (\tau_x \ \tau_y \ \tau_z)^\top \in \mathbb{R}^3$. Cada elemento de este vector corresponde al par ejercido respecto a cada eje.

Sustituyendo las ecs. (20) y (21) en la ec. (19), se obtiene la ec. (22). Este conjunto de ecuaciones representa el modelo dinámico del cuadricóptero.

$$\begin{cases} -mg[s(\theta)] = m(\dot{u} + qw - rv) \\ mg[c(\theta)s(\phi)] = m(\dot{v} - pw + ru) \\ mg[c(\theta)c(\phi)] - f_t = m(\dot{w} + pv - qu) \\ \tau_x = \dot{p}I_x - qrI_y + qrI_z \\ \tau_y = \dot{q}I_y + prI_x - prI_z \\ \tau_z = \dot{r}I_x - pqI_x + pqI_y \end{cases} \quad (22)$$

E. Entradas de control

Como puede observarse en la ec. (22), existen cuatro parámetros libres que determinan la dinámica del cuadricóptero. Estos son el empuje total y los pares en torno a cada eje del vehículo. Estas fuerzas o pares son el resultado de la diferencia de la velocidad de rotación de los rotores. Se considera que estas magnitudes son proporcionales a sumas y restas de los cuadrados de las velocidades de los rotores [3], [4], [5], [18]. La combinación de rotores que produce cada fuerza depende de la configuración del cuadricóptero.

En este trabajo se modela el cuadricóptero en una configuración en equis. Todos los rotores están involucrados en la generación de cada una de las fuerzas, a diferencia de una configuración en cruz. Considerando la numeración de los

rotores que se muestra en la fig. 7, se establecen las siguientes relaciones:

$$\begin{cases} f_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_x = \frac{\sqrt{2}}{2}bl(\Omega_2^2 + \Omega_3^2 - \Omega_1^2 - \Omega_4^2) \\ \tau_y = \frac{\sqrt{2}}{2}bl(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \\ \tau_z = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2), \end{cases} \quad (23)$$

donde l es la distancia entre cualquier rotor y el centro del vehículo, b es el factor de empuje, y d es el factor de arrastre [4]. Sustituyendo la ec. (23) en (22) se relaciona directamente la dinámica del sistema con las entradas de control, como se muestra en la ec. (24).

$$\begin{cases} -mg[s(\theta)] = m(\dot{u} + qw - rv) \\ mg[c(\theta)s(\phi)] = m(\dot{v} - pw + ru) \\ mg[c(\theta)c(\phi)] - b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) = m(\dot{w} + pv - qu) \\ \frac{\sqrt{2}}{2}bl(\Omega_2^2 + \Omega_3^2 - \Omega_1^2 - \Omega_4^2) = \dot{p}I_x - qrI_y + qrI_z \\ \frac{\sqrt{2}}{2}bl(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) = \dot{q}I_y + prI_x - prI_z \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) = \dot{r}I_x - pqI_x + pqI_y \end{cases} \quad (24)$$

F. Algoritmo de combinación de motores

Aunque se ha logrado relacionar directamente la dinámica del sistema con las entradas de control que podemos manipular (las velocidades de los rotores), estas aparecen numerosas veces en las ecuaciones, por lo que el diseño de un controlador se vuelve una labor compleja. Dado que hay una relación directa entre las velocidades de rotación y las fuerzas que producen, es posible estudiar el modelo y diseñar leyes de control considerando como entradas la fuerza de sustentación y los pares respecto a los ejes. Posteriormente se pueden convertir estas fuerzas a las respectivas velocidades que las generan y enviar la señal apropiada a cada motor. El algoritmo que realiza la conversión se conoce en la literatura como *motor mixing algorithm* (MMA) [16], [17], [18]. Expresando el sistema de ecuaciones de (23) de forma matricial, y despejando las constantes del lado derecho de cada igualdad, se obtiene el sistema que se muestra en (25).

$$\begin{pmatrix} \frac{f_t}{b} \\ \frac{2}{\sqrt{2}} \frac{\tau_x}{bl} \\ \frac{2}{\sqrt{2}} \frac{\tau_y}{bl} \\ \frac{\tau_z}{d} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix} \quad (25)$$

Dado que esta matriz transforma el vector de velocidades en el vector de fuerzas, se puede emplear su inversa, que se muestra en (26), para obtener las velocidades a partir de las fuerzas. La implementación de la matriz constituye el algoritmo de combinación más simple. Para asegurar que los actuadores efectúan de forma correcta las acciones de control, se añade lógica adicional que condiciona su operación. Como se ha mencionado antes, cada rotor tiene un sentido de giro definido, y las hélices acopladas a ellos están diseñadas con base en este, por lo que el algoritmo debe restringir

la velocidad de giro de los mismos. De acuerdo con la documentación de Simulink, el empuje máximo que puede producir cada rotor es de 0.3266 N. Calculando la velocidad asociada a esta cantidad, se determina un valor máximo de 2630.5 rad/s para cada rotor.

$$\begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix} = \begin{pmatrix} \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \end{pmatrix} \begin{pmatrix} \frac{f_t}{b} \\ \frac{\tau_x}{bl} \\ \frac{\tau_y}{bl} \\ \frac{\tau_z}{d} \end{pmatrix} \quad (26)$$

G. Modelo en el espacio de estados

Organizando el vector de estados de la forma:

$$\mathbf{x} = (x \ y \ z \ \phi \ \theta \ \psi \ u \ v \ w \ p \ q \ r)^T \in \mathbb{R}^{12}, \quad (27)$$

se define la dinámica del cuadcóptero en el espacio de estados a partir de las ecs. (15) y (22):

$$\begin{cases} \dot{x} = w[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] - \\ \quad v[c(\phi)s(\psi) - c(\psi)s(\phi)s(\theta)] + u[c(\psi)c(\theta)] \\ \dot{y} = v[c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)] - \\ \quad w[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] + u[c(\theta)s(\psi)] \\ \dot{z} = w[c(\phi)c(\theta)] - u[s(\theta)] + v[c(\theta)s(\phi)] \\ \dot{\phi} = p + r[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\ \dot{\theta} = q[c(\phi)] - r[s(\phi)] \\ \dot{\psi} = r \frac{c(\phi)}{c(\theta)} + q \frac{s(\phi)}{c(\theta)} \\ \dot{u} = rv - qw - g[s(\theta)] \\ \dot{v} = pq - ru + g[s(\phi)c(\theta)] \\ \dot{w} = qu - pv + g[c(\theta)c(\phi)] - \frac{f_t}{m} \\ \dot{p} = \frac{I_y - I_z}{I_x} rq + \frac{T_x}{I_x} \\ \dot{q} = \frac{I_z - I_x}{I_y} pr + \frac{T_y}{I_y} \\ \dot{r} = \frac{I_x - I_y}{I_z} pq + \frac{T_z}{I_z} \end{cases} \quad (28)$$

Sea \mathbf{u} el vector de control tal que $\mathbf{u} = (f_t \ \tau_x \ \tau_y \ \tau_z)^T \in \mathbb{R}^4$. El sistema de ecuaciones diferenciales de (28) puede expresarse como la función vectorial

$$\mathbf{f}(\mathbf{x}, \mathbf{u}). \quad (29)$$

H. Linealización del modelo

La linealización de (29) se desarrolla respecto a un punto $\bar{\mathbf{x}}$ que para una entrada determinada $\bar{\mathbf{u}}$ es la solución del sistema algebraico:

$$\mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = 0. \quad (30)$$

Debido a la cantidad de funciones trigonométricas que están presentes en \mathbf{f} , se realiza una aproximación previa a la linealización, que consiste en considerar que los argumentos de las funciones trigonométricas tienen valores pequeños, tal que $\cos \theta \approx 1$ y $\sin \theta \approx \theta$. A partir de lo anterior, el modelo se simplifica a:

$$\begin{cases} \dot{x} \approx w(\phi\psi + \theta) - v(\psi - \phi\theta) + u \\ \dot{y} \approx v(1 + \phi\psi\theta) - w(\phi - \psi\theta) + u\psi \\ \dot{z} \approx w - u\theta + v\phi \\ \dot{\phi} \approx p + r\theta + q\phi \\ \dot{\theta} \approx q - r\phi \\ \dot{\psi} \approx r + q\phi \\ \dot{u} \approx rv - qw - g\theta \\ \dot{v} \approx pq - ru + g\phi \\ \dot{w} \approx qu - pv + g - \frac{f_t}{m} \\ \dot{p} \approx \frac{I_y - I_z}{I_x} rq + \frac{T_x}{I_x} \\ \dot{q} \approx \frac{I_z - I_x}{I_y} pr + \frac{T_y}{I_y} \\ \dot{r} \approx \frac{I_x - I_y}{I_z} pq + \frac{T_z}{I_z} \end{cases} \quad (31)$$

El centro de masa del cuerpo puede ubicarse en cualquier punto del espacio, con la condición de que todos los rotores ejerzan la misma fuerza y la suma de estas sea igual al peso del cuadcóptero, además de tener una cantidad de movimiento nulo y evitar que el ángulo de guiñada ψ varíe, [4], [5], por lo que se trata de un punto de equilibrio forzado:

$$\bar{\mathbf{x}} = (\bar{x} \ \bar{y} \ \bar{z} \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)^T \in \mathbb{R}^{12}. \quad (32)$$

El vector de control que se requiere para satisfacer el equilibrio, es decir, que anula la aceleración debido a la gravedad y mantiene al cuadcóptero flotando en un punto, está dado por:

$$\bar{\mathbf{u}} = (mg \ 0 \ 0 \ 0)^T \in \mathbb{R}^4. \quad (33)$$

Con el punto de equilibrio $\bar{\mathbf{x}}$ y el vector de control $\bar{\mathbf{u}}$ se obtienen las matrices asociadas al sistema lineal:

$$A = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (34)$$

$$B = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \Big|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{pmatrix}. \quad (35)$$

De acuerdo con lo anterior, el modelo lineal es:

$$\dot{\mathbf{x}} = A(\mathbf{x} - \bar{\mathbf{x}}) + B(\mathbf{u} - \bar{\mathbf{u}}). \quad (36)$$

Utilizando el cambio de variable $\mathbf{x}_n = \mathbf{x} - \bar{\mathbf{x}}$ y $\mathbf{u}_n = \mathbf{u} - \bar{\mathbf{u}}$, se definen coordenadas relativas a $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$, estableciendo el punto de equilibrio como el origen. Escribiendo el modelo lineal en términos de este cambio de variable se obtiene la expresión:

$$\dot{\mathbf{x}}_n = A\mathbf{x}_n + B\mathbf{u}_n. \quad (37)$$

Desarrollando los productos matriciales, se obtiene la ec. (38).

$$\begin{cases} \dot{x}_n = u \\ \dot{y}_n = v \\ \dot{z}_n = w \\ \dot{\phi}_n = p \\ \dot{\theta}_n = q \\ \dot{\psi}_n = r \\ \dot{u}_n = -g\theta \\ \dot{v}_n = g\phi \\ \dot{w}_n = g - \frac{f_t}{m} \\ \dot{p}_n = \frac{T_x}{I_x} \\ \dot{q}_n = \frac{T_y}{I_y} \\ \dot{r}_n = \frac{T_z}{I_z} \end{cases} \quad (38)$$

III. DISEÑO DEL CONTROLADOR

A. Control LQR

De acuerdo con [19], un problema de control óptimo consiste en encontrar una ley de control \mathbf{u} para el sistema

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad (39)$$

tal que se cumpla algún objetivo de control mientras se optimiza un criterio de desempeño o función de coste

$$J = \psi(\mathbf{x}(T)) + \int_0^T l(\mathbf{x}(t), \mathbf{u}(t)) dt, \quad (40)$$

dada una condición inicial $\mathbf{x}(0) = \mathbf{x}_0$ y un conjunto de entradas de control permitidas $\mathbf{u}(t) = U$. Generalmente,

se asume que el estado $\mathbf{x}(t)$ es de dimensión n y que la entrada de control $\mathbf{u}(t)$ es de dimensión m . El conjunto U es un componente importante del problema de control óptimo. En algunos casos, corresponde al espacio de dimensión m , cuando no existen restricciones sobre $\mathbf{u}(t)$. En otros casos, el conjunto toma la forma de inequidades de los componentes de $\mathbf{u}(t)$. Estas restricciones reflejan el hecho de que en algunos sistemas, las entradas de control deben mantenerse acotadas.

En la función objetivo de (40), el término $\psi(\mathbf{x}(t))$ es la contribución al objetivo del estado final. Por ejemplo, este término se presenta si el objetivo deseado es controlar un objeto para que alcance una velocidad máxima en un tiempo determinado. El término integral representa una contribución que se acumula a lo largo del tiempo. Este término aparece, por ejemplo, si el objetivo es minimizar el gasto total de combustible de una máquina. Un problema específico puede tener ψ o l igual a cero, pero no ambas.

El controlador LQR (Regulador Cuadrático Lineal, por sus siglas en inglés) se utiliza para sistemas lineales de la forma

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \quad (41)$$

generando una acción de control que equivale al negativo del producto de la matriz K , llamada matriz de realimentación de estados, y el vector de estados:

$$\mathbf{u}(t) = -K\mathbf{x}(t). \quad (42)$$

La matriz K se diseña con el fin de minimizar la función de coste

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt, \quad (43)$$

donde \mathbf{Q} es una matriz hermítica definida positiva, o semidefinida positiva, y \mathbf{R} es una matriz hermítica definida positiva. Estas matrices representan la importancia relativa del error y del coste de la energía de las señales de control, respectivamente. La ley de control obtenida de la ecuación 42 es la ley de control óptimo. Si se determinan los elementos desconocidos de la matriz K para minimizar la función de coste, $\mathbf{u}(t) = -K\mathbf{x}(t)$ es óptima para cualquier estado inicial $\mathbf{x}(0)$ [20].

Para el modelo lineal de la ec. (37), se define la ley de control como:

$$\mathbf{u}_n = -K\mathbf{x}_n. \quad (44)$$

Deshaciendo el cambio de variable se obtiene la ley de control en términos del punto de equilibrio, como se muestra en la ec. (45):

$$\mathbf{u} = -K(\mathbf{x} - \bar{\mathbf{x}}) + \bar{\mathbf{u}}. \quad (45)$$

La matriz K se ha calculado empleando la función `lqr(A, B, Q, R)` de MATLAB. Los argumentos de la función corresponden a las matrices A y B del sistema linealizado, que se muestran en las ecs. (34) y (35), mientras que Q y R

se diseñan para penalizar el error de los estados o las acciones de control, respectivamente, y alcanzar un comportamiento deseado. Los valores de estas matrices fueron elegidos de forma empírica al analizar la simulación del vehículo. De tal manera, se define Q como una matriz identidad de dimensión 12×12 y R como la matriz diagonal de dimensión 4×4 que se muestra en la ec. (46).

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 110,000 & 0 & 0 \\ 0 & 0 & 110,000 & 0 \\ 0 & 0 & 0 & 70,000 \end{pmatrix} \quad (46)$$

La magnitud de las ganancias seleccionadas para esta matriz, correspondientes a los pares respecto a los ejes, se debe a que los momentos de inercia del vehículo tienen valores pequeños; un par elevado provocaría un movimiento excesivo, por lo que se han penalizado severamente dichas acciones de control.

Sustituyendo la ley de control de (44) en el sistema lineal se obtiene la expresión para el sistema en lazo cerrado:

$$\dot{\mathbf{x}}_n = (A - BK)\mathbf{x}_n. \quad (47)$$

B. Sintonización del controlador

Para realizar el ajuste del controlador LQR, es decir, la búsqueda de matrices Q y R adecuadas, se aborda este problema como uno de optimización. Sea el vector de error $\mathbf{e}(t) = (\bar{x} - x \quad \bar{y} - y \quad \bar{z} - z \quad \bar{\psi} - \psi)^\top$, se define como función objetivo la suma de la energía del sistema y la norma del vector de error, en cada instante de tiempo:

$$F(Q, R) = \int_0^T (E_p + E_c + \|\mathbf{e}(t)\|) dt. \quad (48)$$

Se considera también un segundo criterio de desempeño. Multiplicando el argumento de la integral anterior por el tiempo se obtiene la siguiente función:

$$F(Q, R) = \int_0^T t(E_p + E_c + \|\mathbf{e}(t)\|) dt. \quad (49)$$

Se calcula la energía del sistema desde el marco inercial, por lo que se utilizan las velocidades lineales y angulares respecto a este. Se decidió utilizar un método numérico porque garantiza producir al menos óptimos locales cuando son empleados en funciones suaves [21].

El método numérico aplicado es descenso de gradiente, un algoritmo iterativo de primer orden, que consiste en encontrar el mínimo de una función a partir de un punto inicial determinado, definiendo cada punto subsecuente al desplazarse en incrementos proporcionales al negativo del gradiente de la función en el punto anterior. Este método se basa en la observación de que si una función multivariable $F(\mathbf{x})$ es definida y diferenciable en la vecindad de un punto a , $F(\mathbf{x})$ disminuye más rápidamente si uno se desplaza desde a en la dirección del negativo del gradiente de $F(\mathbf{x})$ en a . Para valores suficientemente pequeños de α , se cumple la ec. (50) y $F(a_n) \geq F(a_{n+1})$ [22].

$$a_{n+1} = a_n - \alpha_n \nabla F(a_n) \quad (50)$$

El algoritmo empieza con una propuesta \mathbf{x}_0 como mínimo local de F , y considera la secuencia $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$ tal que se cumple

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha_n \nabla F(\mathbf{x}_n), \quad n \geq 0, \quad (51)$$

y $F(\mathbf{x}_n)$ en la secuencia

$$F(\mathbf{x}_0) \geq F(\mathbf{x}_1) \geq F(\mathbf{x}_2) \geq \dots \geq F(\mathbf{x}_n) \quad (52)$$

converge a un mínimo. Nótese que se permite al valor del tamaño de paso α cambiar en cada iteración.

Cuando se computa la longitud de paso α_k , se debe considerar la relación costo-beneficio. Es deseable elegir un α_k tal que se obtenga una reducción sustancial de F , pero al mismo tiempo es importante limitar el tiempo de operación del algoritmo. Una condición simple que puede imponerse sobre α_k es que resulte en una reducción de $F(\mathbf{x})$:

$$F(\mathbf{x}_k + \alpha_k \nabla F(\mathbf{x}_k)) < F(\mathbf{x}_k). \quad (53)$$

Para encontrar valores adecuados para α en cada iteración, se hace uso de un proceso de backtracking. La forma más básica de este se muestra en el algoritmo 1.

Algoritmo 1 Backtracking.

- 1: Elegir $\alpha_0 > 0$, $\rho \in (0, 1)$
 - 2: $\alpha = \alpha_0$
 - 3: **Mientras** $F(\mathbf{x}_k + \alpha_k \nabla F(\mathbf{x}_k)) > F(\mathbf{x}_k)$ **Hacer**
 - 4: $\alpha = \rho \alpha$
 - 5: **Fin Mientras**
 - 6: $\alpha_k = \alpha$
-

Una vez que se ha resuelto el problema de encontrar α en cada iteración, se puede formular el descenso de gradiente como se muestra en el algoritmo 2.

Algoritmo 2 Descenso de gradiente.

- 1: Elegir \mathbf{x}_0 , definir $\epsilon > 0$, $k = 1$, $\mathbf{x}_k = \mathbf{x}_0$
 - 2: **Mientras** $\|\nabla F(\mathbf{x}_k)\| \geq \epsilon$ **Hacer**
 - 3: $\mathbf{d}_k = -\nabla F(\mathbf{x}_k)$
 - 4: Elegir α utilizando el algoritmo 1.
 - 5: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{d}_k$
 - 6: **Fin Mientras**
 - 7: **Regresar** \mathbf{x}_k
-

El punto que se ha considerado para inicializar el algoritmo son las matrices Q y R obtenidas de forma empírica. El algoritmo se computará de acuerdo con el pseudocódigo, hasta que deje de cumplirse la condición de permanencia del ciclo *while*. Es decir, mientras la magnitud del vector gradiente, en la iteración actual de la función, sea mayor a una tolerancia seleccionada ϵ . Cuando la magnitud del vector gradiente se encuentra por debajo de este umbral, se

considera que se ha alcanzado un mínimo en la función. Con el propósito de encontrar matrices adecuadas para distintos puntos de consigna, se ejecuta el algoritmo de optimización considerando distintas entradas de referencia y condiciones iniciales, considerando también variaciones en el ángulo ψ . Posteriormente, se obtiene un promedio de las ganancias.

IV. SIMULACIONES

Se evaluó el desempeño de los controladores diseñados simulando el modelo no lineal del cuadricóptero, utilizando la versión 2019a de MATLAB, sobre el sistema operativo Windows 10 Home de 64 bits, en una computadora con las siguientes características:

- Procesador: Intel(R) Core(TM) i7-6700HQ @ 2.6 GHz
- Tarjeta gráfica: NVIDIA GeForce GTX 960M
- Memoria RAM: 12 GB

La evaluación de (48) requiere simular el sistema dinámico para cada nuevo vector $\mathbf{x}_i = [Q, R]^T$. Se ha empleado el método numérico de Runge-Kutta de orden cuatro para aproximar la solución del sistema de ecuaciones diferenciales, con un tamaño de paso igual a $\delta t = 2^{-8}$. Para computar el vector \mathbf{x}_{i+1} , debe estimarse el gradiente de (48). Debido a la complejidad de una solución analítica, se ha empleado el método de diferencias finitas para aproximar la derivación parcial. El tamaño de paso para este método es de 2^{-20} . El punto inicial de la función objetivo se ha definido por una matriz identidad de dimensión 12×12 para Q y la matriz (46) como R . El algoritmo de descenso de gradiente considera una condición de permanencia en el ciclo *while* de $\epsilon = 1 \times 10^{-12}$.

El controlador LQR sintonizado de forma empírica se denomina a partir de este punto como Controlador 1. El controlador cuyas ganancias se obtuvieron como resultado de la optimización del índice de desempeño (48), se denomina Controlador 2. El controlador cuyas ganancias fueron obtenidas como resultado de la optimización del índice de desempeño (49), es denominado como Controlador 3. La tabla II muestra las ganancias correspondientes de cada controlador.

El desempeño de los controladores se analiza de acuerdo con tres pruebas. La primera consiste en desplazar al vehículo en el eje x , desde el origen a una posición con coordenadas $(0.5 \text{ m}, 0, 0)$ en el espacio, a partir de condiciones iniciales nulas. Después de cinco segundos, la posición deseada se modifica a $(1 \text{ m}, 0, 0)$, considerando además una variación en el ángulo de guiñada de $\frac{\pi}{4}$ radianes. La segunda prueba consiste en desplazar al vehículo en el eje z , desde el origen a una posición con coordenadas $(0, 0, -0.5 \text{ m})$ en el espacio, a partir de condiciones iniciales nulas. Después de cinco segundos, la posición deseada se modifica a $(0, 0, -1 \text{ m})$, considerando además una variación en el ángulo de guiñada de $\frac{\pi}{4}$ radianes. Es importante recordar que, en la segunda prueba, el valor negativo de la coordenada en z se debe a la convención utilizada para el marco inercial; se desea probar el movimiento ascendente del vehículo. El tiempo de subida y de asentamiento se calcula de acuerdo con [19]. El primero corresponde al tiempo necesario para que la respuesta del sistema, ante un cambio en el punto de consigna, se eleve

TABLA II
COMPARATIVA DE LOS CONTROLADORES

Controlador	Ganancias	
	Matriz Q	Matriz R
Controlador 1	$Q_{1,1} = 1$ $Q_{2,2} = 1$ $Q_{3,3} = 1$ $Q_{4,4} = 1$ $Q_{5,5} = 1$ $Q_{6,6} = 1$ $Q_{7,7} = 1$ $Q_{8,8} = 1$ $Q_{9,9} = 1$ $Q_{10,10} = 1$ $Q_{11,11} = 1$ $Q_{12,12} = 1$	$R_{1,1} = 1$ $R_{2,2} = 1.1 \times 10^5$ $R_{3,3} = 1.1 \times 10^5$ $R_{4,4} = 7 \times 10^4$
Controlador 2	$Q_{1,1} = 1.8149$ $Q_{2,2} = 1.5863$ $Q_{3,3} = 1.1666$ $Q_{4,4} = 0.9654$ $Q_{5,5} = 0.9359$ $Q_{6,6} = 1.3705$ $Q_{7,7} = 0.6484$ $Q_{8,8} = 0.3933$ $Q_{9,9} = 0.7882$ $Q_{10,10} = 0.5640$ $Q_{11,11} = 1.74 \times 10^{-6}$ $Q_{12,12} = 0.0319$	$R_{1,1} = 1.0016$ $R_{2,2} = 1.1 \times 10^5$ $R_{3,3} = 1.1 \times 10^5$ $R_{4,4} = 7 \times 10^4$
Controlador 3	$Q_{1,1} = 1.8021$ $Q_{2,2} = 1.5971$ $Q_{3,3} = 1.3411$ $Q_{4,4} = 0.9625$ $Q_{5,5} = 0.9197$ $Q_{6,6} = 0.9631$ $Q_{7,7} = 0.5087$ $Q_{8,8} = 0.4602$ $Q_{9,9} = 0.2804$ $Q_{10,10} = 0.5258$ $Q_{11,11} = 9.44 \times 10^{-6}$ $Q_{12,12} = 0.1379$	$R_{1,1} = 0.9831$ $R_{2,2} = 1.1 \times 10^5$ $R_{3,3} = 1.1 \times 10^5$ $R_{4,4} = 7 \times 10^4$

desde el 10% del valor en estado estable hasta el 90%. El segundo se define como el tiempo mínimo, para que la respuesta se mantenga en un intervalo de $\pm 5\%$ del valor en estado estable. La tercer prueba consiste en que el vehículo siga una trayectoria simple formada por tres puntos. En esta prueba no se cambia de referencia de acuerdo con un tiempo definido, sino que se condiciona al vehículo a avanzar al siguiente punto únicamente cuando la magnitud del vector de estado es $\pm 1\%$ del estado deseado (posición deseada, con velocidades angulares y lineales igual a cero). Los puntos que componen la trayectoria son $P_1 = (0.5 \text{ m}, 0, 0)$, $P_2 = (0, 0.5 \text{ m}, 0)$, y $P_3 = (0, 0, -0.5 \text{ m})$.

En las tablas III y IV se muestra el tiempo de subida y tiempo de asentamiento de cada controlador, al realizar la primer y segunda prueba, respectivamente. En la tabla V se muestra la variación de la energía en cada una de las pruebas.

En las figuras 8 y 9 se muestra la posición del vehículo al realizar la primer prueba (desplazamiento en el eje x). A su vez, en las figuras 10 y 11, se muestra la posición del vehículo al realizar la segunda prueba (desplazamiento en el eje z).

TABLA III
RESPUESTA DEL SISTEMA EN LA PRUEBA 1

Controlador	Tiempo de subida (s)		Tiempo de asentamiento (s)	
	Posición 1	Posición 2	Posición 1	Posición 2
Controlador 1	2.2227	2.1875	3.4648	12.7070
Controlador 2	1.3555	1.1016	2.1172	11.4844
Controlador 3	1.2461	1.1484	1.9414	11.4648

TABLA IV
RESPUESTA DEL SISTEMA EN LA PRUEBA 2

Controlador	Tiempo de subida (s)		Tiempo de asentamiento (s)	
	Posición 1	Posición 2	Posición 1	Posición 2
Controlador 1	2.1992	1.9961	3.0664	12.6992
Controlador 2	1.8125	1.0508	2.5391	11.6680
Controlador 3	1.0352	0.957	1.4766	11.2461

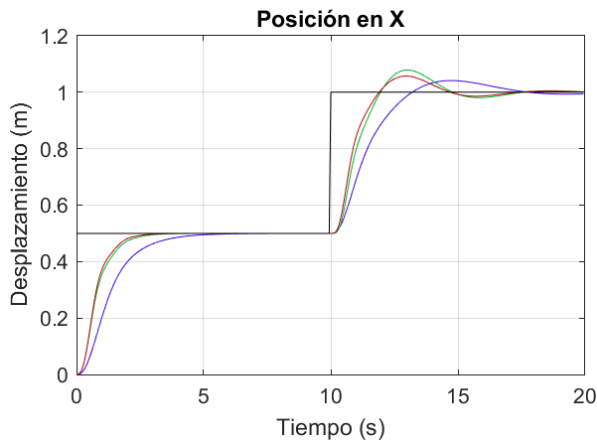


Fig. 8. Primer prueba. Controlador 1 en morado, Controlador 2 en verde, y Controlador 3 en rojo.

TABLA V
VARIACIÓN DE LA ENERGÍA

Controlador	Prueba 1	Prueba 2
Controlador 1	0.0060 J	9.2930 J
Controlador 2	0.0077 J	9.3840 J
Controlador 3	0.0086 J	9.6192 J

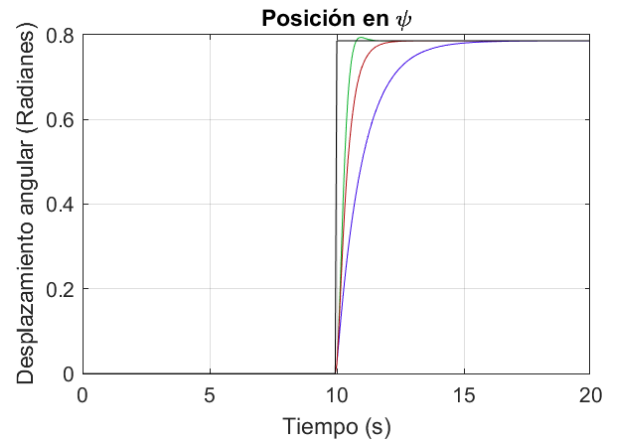


Fig. 9. Primer prueba. Controlador 1 en morado, Controlador 2 en verde, y Controlador 3 en rojo.

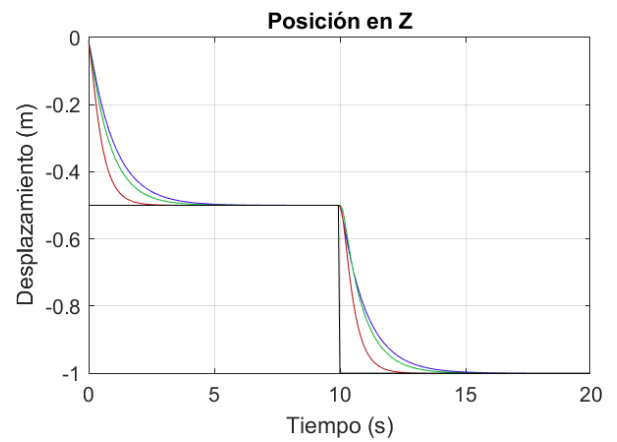


Fig. 10. Segunda prueba. Controlador 1 en morado, Controlador 2 en verde, y Controlador 3 en rojo.

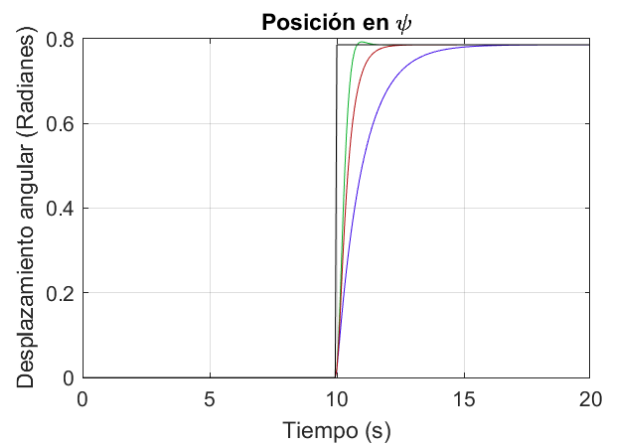


Fig. 11. Segunda prueba. Controlador 1 en morado, Controlador 2 en verde, y Controlador 3 en rojo.

En la tabla VI se muestra la variación de la energía y el tiempo que requirió cada controlador, para alcanzar el último

TABLA VI
COMPARATIVA DE CONTROLADORES EN LA TERCER PRUEBA

Controlador	Energía (J)	Tiempo (s)
Controlador 1	1.2921	14.6758
Controlador 2	0.9392	9.9023
Controlador 3	0.7272	8.5703

punto en la tercer prueba.

Las figuras 12, 13, y 14 muestran la posición del vehículo durante la realización de la tercer prueba, utilizando el controlador sintonizado empíricamente. Las figuras 15, 16, y 17 muestran la posición del vehículo durante la realización de la tercer prueba, utilizando el controlador sintonizado al optimizar el índice (48). Las figuras 18, 19, y 20 muestran la posición del vehículo durante la realización de la tercer prueba, utilizando el controlador sintonizado al optimizar el índice (49).

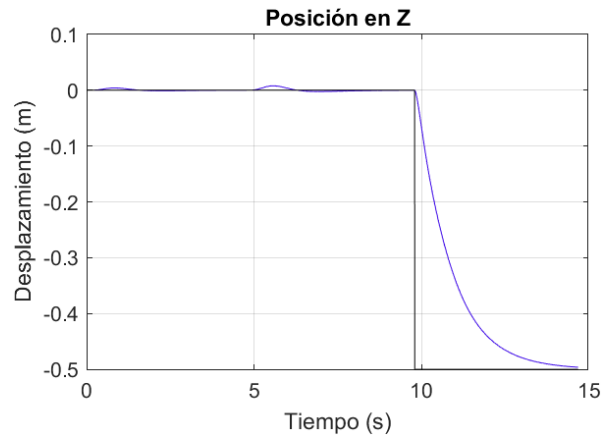


Fig. 14. Posición en el eje z durante la prueba de seguimiento de puntos, utilizando el Controlador 1.

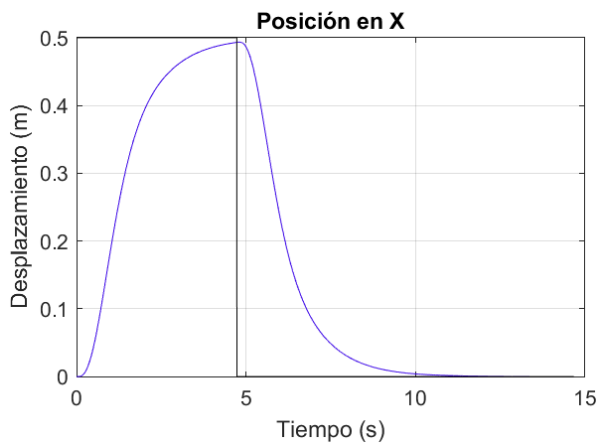


Fig. 12. Posición en el eje x durante la prueba de seguimiento de puntos, utilizando el Controlador 1.

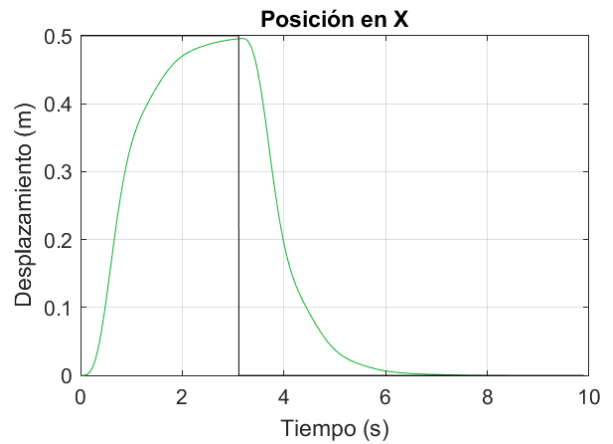


Fig. 15. Posición en el eje x durante la prueba de seguimiento de puntos, utilizando el Controlador 2.

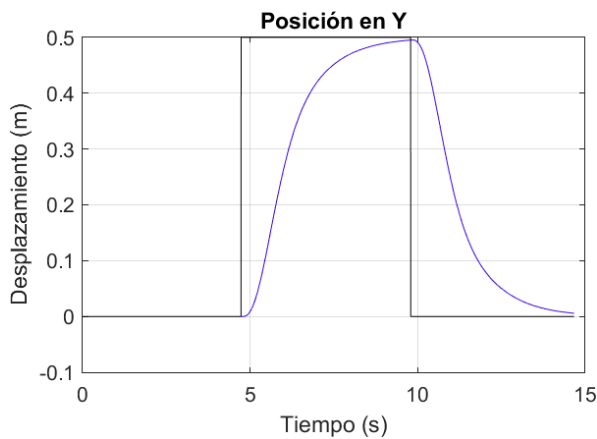


Fig. 13. Posición en el eje y durante la prueba de seguimiento de puntos, utilizando el Controlador 1.

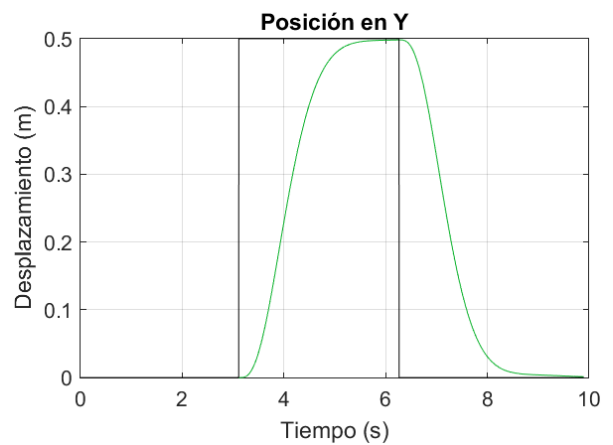


Fig. 16. Posición en el eje y durante la prueba de seguimiento de puntos, utilizando el Controlador 2.

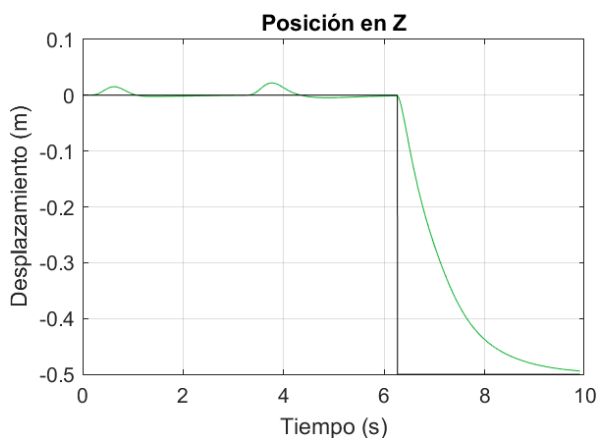


Fig. 17. Posición en el eje z durante la prueba de seguimiento de puntos, utilizando el Controlador 2.

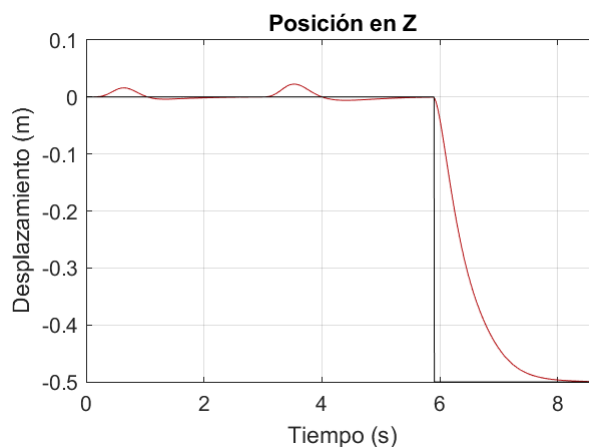


Fig. 20. Posición en el eje z durante la prueba de seguimiento de puntos, utilizando el Controlador 3.

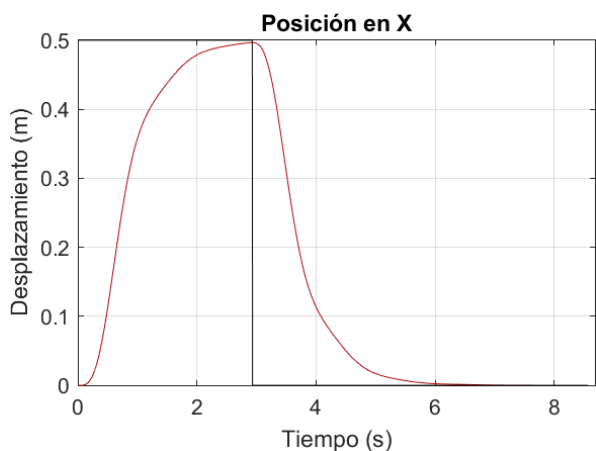


Fig. 18. Posición en el eje x durante la prueba de seguimiento de puntos, utilizando el Controlador 3.

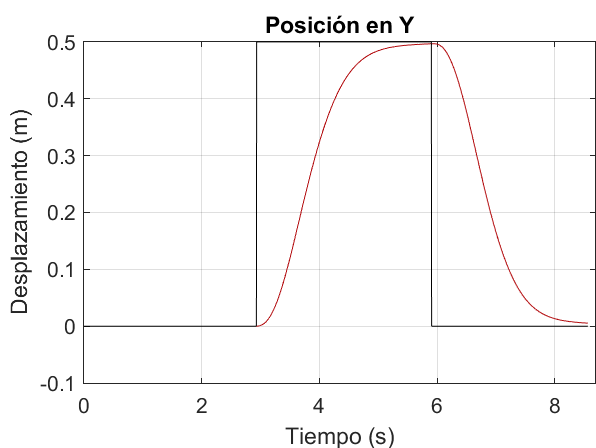


Fig. 19. Posición en el eje y durante la prueba de seguimiento de puntos, utilizando el Controlador 3.

V. CONCLUSIONES Y TRABAJO FUTURO

Utilizando la metodología propuesta, se obtienen dos conjuntos de ganancias que son óptimas respecto a los dos índices de desempeño correspondientes.

En la primer prueba, los Controladores 2 y 3, presentan, en todas las instancias, tiempos de subida y de asentamiento menores al Controlador 1. En promedio, los tiempos de subida y de asentamiento del Controlador 2, equivalen al 55.67% y 75.74% de los del Controlador 1, respectivamente. A su vez, los tiempos de subida y de asentamiento del Controlador 3, equivalen al 54.28% y 73.13% de los del Controlador 1, respectivamente. Sin embargo, los Controladores 2 y 3 presentan un incremento del 28.33% y 43.33% en la energía consumida, respecto al Controlador 1.

De acuerdo con los resultados de la segunda prueba, los tiempos de subida y de asentamiento del Controlador 2, equivalen, en promedio, al 67.53% y 87.34% de los del Controlador 1, respectivamente. A su vez, los tiempos de subida y de asentamiento del Controlador 3, equivalen al 47.51% y 68.36% de los del Controlador 1, respectivamente. En este caso, los Controladores 2 y 3 presentan un incremento del 0.98% y 3.51% en la energía consumida, respecto al Controlador 1.

En la prueba tres se comparan el tiempo y la energía que cada controlador necesita para realizar el seguimiento de la trayectoria. Debido a que la entrada de referencia sólo cambia cuando el vehículo alcanza, con un determinado grado de exactitud, el punto de consigna, los controladores que presenten una respuesta más rápida, es decir, tiempos de subida y de asentamiento menores, recorrerán la trayectoria en un tiempo menor. Esto a su vez, representa una reducción en la energía utilizada a lo largo de la trayectoria, ya que el simple hecho de mantener al vehículo en el aire consume energía. Así, el Controlador 2 realiza el seguimiento de la trayectoria en un tiempo equivalente al 67.47% del que necesita el Controlador 1, utilizando el 72.69% de la energía de este último. El Controlador 3 termina el seguimiento en un 58.4% del tiempo del Controlador 1 y consume el 56.28% de la energía.

En todas las pruebas, los controladores propuestos obtuvieron una reducción considerable en el tiempo de respuesta respecto al Controlador 1. Mientras que en la prueba 1 y 2, estos controladores presentan un consumo energético mayor, en la prueba tres se observa una reducción considerable en el uso de la energía. Estos resultados denotan la viabilidad del uso de los controladores diseñados, para aplicaciones donde se requiera realizar el seguimiento de una trayectoria definida por puntos, con el propósito de reducir el tiempo y el consumo energético del sistema.

Se deja como trabajo futuro la implementación del controlador diseñado en la planta física, así como la optimización por medio de distintos métodos y de acuerdo con distintos criterios de desempeño.

REFERENCIAS

- [1] Randal W. Beard and Timothy W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, Princeton, New Jersey, first edition, 2012.
- [2] Sevkuthan Kurak and Migdat Hodzic. Control and Estimation of a Quadcopter Dynamical Model. *Periodicals of Engineering and Natural Sciences (PEN)*, 6(1):63, 2018.
- [3] Zoran Benic, Petar Piljek, and Denis Kotarski. Mathematical Modelling of Unmanned Aerial Vehicles with Four Rotors. *Interdisciplinary Description of Complex Systems*, 14(1):88–100, 2016.
- [4] Francesco Sabatino. Quadrotor control: modeling, nonlinear control design, and simulation. Technical report, KTH Royal Institute of Technology, Stockholm, Sweden, 2007.
- [5] Gabriel de la Cal Mendoza. *Modelado, simulación, construcción y control de un Quadcopter*. Master thesis, Universitat Politècnica de Catalunya, 2014.
- [6] Lucas M Argentim, Paulo E Santos, and Renato A Aguiar. PID, LQR and LQR-PID on a Quadcopter Platform. *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 2013.
- [7] Faraz Ahmad, Pushpendra Kumar, and Pravin P Patil. Comparative Analysis of PID and Fuzzy-PID Controllers for Quadcopter Altitude Control. (July), 2019.
- [8] Emmanuel Okyere, Amar Bousbaine, Gwangtim T Poyi, Ajay K Joseph, and Jose M Andrade. LQR controller design for quad-rotor helicopters. (Pemd):1–5, 2018.
- [9] Péter Bauer, György Ritzinger, Alexandros Soumelidis, and József Bokor. LQ Servo control design with Kalman filter for a quadrotor UAV. 2:9–14, 2008.
- [10] Arturo Y. Jaen-cuellar, Luis Morales-velazquez, and Roque A. Osornorios. PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems. 2013.
- [11] Muhammed Arif Sen and Mete Kalyoncu. Optimal Tuning of a LQR Controller for an Inverted Pendulum Using the Bees Algorithm. (December), 2016.
- [12] Abdulla I Abdulla, Ibrahim K Mohammed, and Abdulhamed M Jasim. Roll Control System Design Using Auto Tuning LQR Technique. (April), 2019.
- [13] Maidul Islam, Mohamed Okasha, and Moumen Mohammad. Trajectory tracking in quadrotor platform by using PD controller and LQR control approach. *IOP Conference Series: Materials Science and Engineering*, 2017.
- [14] MathWorks. Parrot Minidrones Support from Simulink, 2019.
- [15] Herbert Goldstein, Charles Poole, and John Safko. *Classical Mechanics*. Addison Wesley, 3rd edition, 2002.
- [16] Corrado Santoro. Control Systems for Multi-rotors - Principles, Modeling and Software Design. *Dipartimento di Matematica e Informatica - Università di Catania*, page 60, 2018.
- [17] Walid Amanhoud. Model based control of quadcopters. Technical report, École Polytechnique Fédérale de Lausanne, Lausanne, Swiss, 2016.
- [18] Ernesto Alonso Paiva Peredo. *Modelado y Control de un Cuadricóptero*. Master thesis, Universidad de Piura, 2016.
- [19] David G. Luenberger. *Introduction to Dynamic Systems, Theory, Models, and Applications*. John Wiley & Sons, New York, 1979.
- [20] Katsuhiko Ogata. *Ingeniería de control moderna*. Pearson Educación S.A., Madrid, España, 5th edition, 2010.
- [21] J A Morales, M A Castro, D García, C Higuera, and J Sandoval. IDA-PBC Controller Tuning Using Steepest Descent. pages 1–14.
- [22] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer-Verlag New York, second edition, 2006.

Control IDA-PBC en voltaje para la regulación de posición de un sistema carro-péndulo

Jesús Sandoval*, Iván Velázquez*, César Higuera†, Abigail Chargoy*, Víctor Santibáñez‡

*Tecnológico Nacional de México, Instituto Tecnológico de La Paz,
Blvd. Forjadores, La Paz, B.C.S., 23080, MÉXICO.

E-mail: jsandoval@itlp.edu.mx, ivan_180283@hotmail.com.mx, abiggailc@gmail.com

†Tecnológico Nacional de México, Instituto Tecnológico de Ensenada,
Blvd. Tecnológico No. 150, Ex-ejido Chapultepec, Ensenada, MÉXICO
E-mail: cesarh@itlp.edu.mx

‡Tecnológico Nacional de México, Instituto Tecnológico de La Laguna,
Torreón, Coahuila, 27001, MÉXICO
E-mail: vsantiba@itlalaguna.edu.mx

Resumen—Se presenta un control IDA-PBC en voltaje para la regulación de posición de un sistema carro-péndulo. El diseño del controlador sigue un procedimiento establecido en la literatura que permite expresarlo en voltaje en lugar de un par. Dado que la ley de control en voltaje es calculada vía realimentación de posiciones y velocidades del carro-péndulo, en lugar de la corriente del motor, un ingrediente adicional en este trabajo es la verificación de que el voltaje medido en el motor corresponde efectivamente al suministrado por el controlador. Resultados experimentales sobre un sistema carro-péndulo son presentados para validar el desempeño del controlador propuesto.

I. INTRODUCCIÓN

El sistema carro-péndulo es un popular sistema mecánico subactuado disponible en numerosos laboratorios de control automático. Burdamente hablando, un sistema mecánico subactuado es aquel que tiene más grados de libertad que entradas de control. Principalmente, el sistema carro-péndulo se utiliza como plataforma de validación de nuevos algoritmos de control en tareas de investigación, así como también para la enseñanza de esquemas de control lineal en estudios de pre-grado [1], [2], [3]. Ejemplos de controladores no lineales reportados para la regulación de un sistema carro-péndulo son mostrados en [2] y [3], donde ambas propuestas logran llevar el péndulo a su posición vertical superior, mientras el carro es llevado a una posición deseada [2] o únicamente al origen [3].

Por otro lado, el sistema carro-péndulo ofrece el desafío de controlar la posición invertida del péndulo mediante una acción de control del único actuador localizado en el carro. Por lo general, el actuador es un motor de CD operando con un voltaje proporcionado por el controlador, incluso cuando la mayoría de las leyes de control publicadas en la literatura se dan en términos de un par en lugar de un voltaje (e.g., [3], [4], [5]), siendo una excepción el trabajo reportado en [6].

La contribución del presente trabajo es la introducción de un regulador de posición en voltaje para un sistema carro-péndulo accionado por una fuerza. El diseño del controlador propuesto

está basado en una extensión del método IDA-PBC (Interconnection and Damping Assignment Passivity-Based Control, [3]) publicado en [6]. A lo largo del documento, usamos la notación $(\cdot)_{2 \times 2}$ para indicar una matriz 2×2 , con $I_{2 \times 2}$ como la matriz identidad y $0_{2 \times 2}$ la matriz de ceros; mientras $0_2 \in \mathbb{R}^n$ es el vector de ceros 2×1 , $\nabla(\cdot) = \frac{\partial}{\partial(\cdot)}$, y $\det[A]$ denota el determinante de la matriz cuadrada A .

El resto del trabajo está organizado como sigue: En la sección II, se describe el sistema carro-péndulo y su modelo matemático completo (sistema mecánico más el motor de CD), así como el objetivo de control. La sección III introduce el diseño de un control IDA-PBC en voltaje, y resultados experimentales para ilustrar su desempeño son presentados en la sección IV. Finalmente, damos algunas conclusiones en la sección V.

II. SISTEMA CARRO-PÉNDULO

El sistema carro-péndulo consiste de un carro moviéndose horizontalmente con un péndulo acoplado al carro, el cual gira libremente (figura 1).

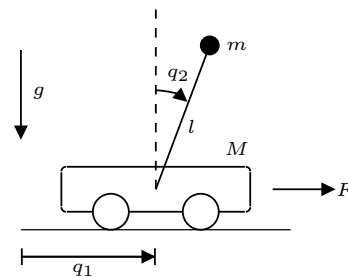


Figura 1. Diagrama esquemático de sistema carro-péndulo.

El desplazamiento del carro es controlado por un actuador que aplica una fuerza F sobre él, mientras la ausencia de actuación en el péndulo determina la naturaleza subactuado del mecanismo [2]. En la figura 1, el desplazamiento del carro y la posición articular del péndulo corresponden a q_1 y q_2 ,

respectivamente, mientras el resto de los parámetros del carro-péndulo están dados por: longitud l , momento de inercia I , la masa del péndulo m , y la masa del carro M , mientras la aceleración de la gravedad es g .

II-A. Modelo dinámico del carro-péndulo

Las ecuaciones de movimiento sin considerar fricción, pueden ser escritas en forma compacta como:

$$\mathcal{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathcal{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (1)$$

donde $\mathbf{q} = [q_1 \ q_2]^T$ es el vector de posiciones articulares, $\dot{\mathbf{q}}$ es el vector de velocidades, $\mathcal{M}(\mathbf{q})$ es la matriz de inercias, $\mathcal{C}(\mathbf{q}, \dot{\mathbf{q}})$ es la matriz de fuerzas centrífugas y de Coriolis, $\mathbf{g}(\mathbf{q})$ es el vector de pares gravitacionales y $\boldsymbol{\tau}$ es el vector de entradas de control, dadas por:

$$\mathcal{M}(\mathbf{q}) = \begin{bmatrix} (M + m) & ml \cos(q_2) \\ ml \cos(q_2) & (ml^2 + I) \end{bmatrix}, \quad (2)$$

$$\mathcal{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & -ml \sin(q_2) \dot{q}_2 \\ 0 & 0 \end{bmatrix}, \quad (3)$$

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} 0 \\ -mgl \sin(q_2) \end{bmatrix}, \quad (4)$$

$$\boldsymbol{\tau} = \begin{bmatrix} F \\ 0 \end{bmatrix}, \quad (5)$$

donde F es la fuerza aplicada al carro para su desplazamiento traslacional. Tomando en cuenta las definiciones (2)-(5) en (1), el modelo dinámico del carro-péndulo queda:

$$[M + m]\ddot{q}_1 + ml \cos(q_2)\ddot{q}_2 - ml \sin(q_2)\dot{q}_2^2 = F, \quad (6)$$

$$ml \cos(q_2)\ddot{q}_1 + [ml^2 + I]\ddot{q}_2 - mgl \sin(q_2) = 0. \quad (7)$$

II-B. Modelo dinámico del motor de CD

Para obtener el modelo dinámico del motor se considera el esquema mostrado en la figura 2.

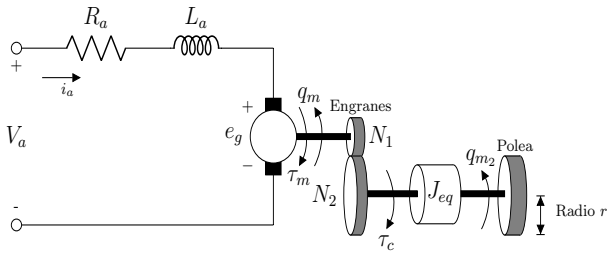


Figura 2. Circuito eléctrico del motor de CD y un sistema de tren de engranes acoplado al eje del motor

Se asume que la fuerza contraelectromotriz e_g es proporcional a la velocidad de rotación del rotor, $\omega_m = \dot{q}_m$, being q_m el desplazamiento angular del eje del motor, y la constante de fuerza contraelectromotriz del motor k_m , tal que e_g está dada por

$$e_g = k_m \omega_m \quad (8)$$

Por la ley de voltajes de Kirchhoff puede obtenerse

$$V_a - R_a i_a - L_a \frac{di_a}{dt} - k_m \omega_m = 0 \quad (9)$$

donde V_a es el voltaje aplicado en las terminales del motor, i_a es la corriente de armadura, R_a y L_a son la resistencia de armadura e inductancia del motor, respectivamente.

Por otro lado, el par del motor τ_m puede ser expresado como:

$$\tau_m = k_t i_a \quad (10)$$

donde k_t es la constante par-motor. De acuerdo a la segunda ley de Newton, el modelo dinámico del sistema mecánico del motor es expresado como:

$$J_m \frac{d\omega_m}{dt} + B_m \omega_m = \tau_m - \frac{\tau_c}{k_g} \quad (11)$$

donde J_m es el momento de inercia del motor, B_m el coeficiente de fricción viscosa en el eje del motor, y el par de carga está dado por τ_c , donde la relación de engranes $k_g = N_2/N_1$ (en general, $k_g \gg 1$) permite reflejar τ_c en el eje del motor. Considerando la relación del tren engranes, la velocidad del motor ω_m puede calcularse a través de la velocidad en el lado de la carga \dot{q}_{m_2} (usualmente medida con un sensor de velocidad), de la siguiente manera:

$$\omega_m = k_g \dot{q}_{m_2}. \quad (12)$$

De (9), despejando la corriente de armadura, y bajo la suposición que la inductancia del motor es despreciada ($L_a = 0$), resulta

$$i_a = \frac{V_a - k_m \omega_m}{R_a} \quad (13)$$

y sustituyendo (13) en (10), el par del motor queda

$$\tau_m = k_t \left(\frac{V_a - k_m \omega_m}{R_a} \right). \quad (14)$$

Enseguida, sustituyendo (14) en (11), se obtiene

$$J_m \frac{d\omega_m}{dt} + B_m \omega_m = k_t \left(\frac{V_a - k_m \omega_m}{R_a} \right) - \frac{\tau_c}{k_g}, \quad (15)$$

y despejando el par τ_c , queda

$$\tau_c = \frac{k_g k_t}{R_a} V_a - \left(\frac{k_g k_t k_m}{R_a} + k_g B_m \right) \omega_m - k_g J_m \frac{d\omega_m}{dt}. \quad (16)$$

II-C. Modelo completo del sistema carro-péndulo

La velocidad angular en el eje de la carga \dot{q}_{m_2} (en este caso una polea dentada), puede ser expresada en términos de la velocidad del carro \dot{q}_1 y el radio r de la polea, por la relación:

$$\dot{q}_{m_2} = \frac{\dot{q}_1}{r}. \quad (17)$$

Para expresar la velocidad angular en el eje del motor en términos del desplazamiento lineal del carro, se sustituye (17) en (12):

$$\omega_m = k_g \frac{\dot{q}_1}{r} \quad (18)$$

Por definición, el par en el eje de carga es

$$\tau_c = F r \quad (19)$$

donde la fuerza F es perpendicular al radio r de la polea, y es responsable del desplazamiento del carro. Si igualamos τ_c de (19) con (16), al sustituir (18) y su derivada temporal, es posible obtener F dado por:

$$F = K_v V_a - B_{eq} \dot{q}_1 - J_{eq} \ddot{q}_1 \quad (20)$$

donde las constantes están definidas como

$$K_v = \frac{k_g k_t}{r R_a}, B_{eq} = \frac{k_g^2 k_t k_m}{r^2 R_a} + \frac{k_g^2}{r^2} B_m, J_{eq} = \frac{k_g^2 J_m}{r^2}.$$

Dado que se asume que el par que genera la fuerza para mover el carro es originado por el motor, éste se puede expresar en términos de un voltaje aplicado en las terminales del motor. Con este fin, igualando (20) y (6), resulta

$$[M+m+J_{eq}] \ddot{q}_1 + ml \cos(q_2) \ddot{q}_2 - ml \sin(q_2) \dot{q}_2^2 + B_{eq} \dot{q}_1 = K_v V_a \quad (21)$$

Por tanto, las ecuaciones (7) y (21) representan la dinámica completa del sistema carro-péndulo en términos del voltaje aplicado al motor de CD como su único actuador.

II-D. Objetivo de control

El objetivo de control consiste en llevar al péndulo a su posición vertical superior, $q_2 = 0$, y al carro a una posición deseada constante q_{d1} . Formalmente, el objetivo de control puede ser establecido como

$$\lim_{t \rightarrow \infty} \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} = \begin{bmatrix} q_{d1} \\ 0 \end{bmatrix}. \quad (22)$$

III. DISEÑO DE UN IDA-PBC EN VOLTAJE

El método IDA-PBC en [3] permite diseñar una ley de control en términos de un par mecánico y en ocasiones debe ser convertido a un voltaje para su implementación en sistemas mecánicos accionados por motores eléctricos. Para lograr la conversión de par a voltaje, una estrategia natural es incorporar la dinámica de los actuadores al esquema de control, sin embargo, estos elementos adicionales no son considerados comúnmente en el diseño y análisis del sistema de control. En [6] se presentó una extensión del método IDA-PBC que incorpora la dinámica de motores eléctricos de corriente directa (CD) con escobillas e imanes permanentes.

III-A. El método IDA-PBC

El método IDA-PBC presentado en [3] aplicado al sistema carro-péndulo inicia definiendo su función de energía como la suma de la función de energía cinética más la función de energía potencial¹:

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^\top \mathcal{M}^{-1}(\mathbf{q}) \mathbf{p} + V(\mathbf{q}) \quad (23)$$

donde $\mathbf{q} \in \mathbb{R}^2$ y $\mathbf{p} \in \mathbb{R}^2$ son los vectores de posiciones y momentos generalizados, respectivamente, $\mathcal{M} = \mathcal{M}^\top > 0$ es la matriz de inercia, y V es la energía potencial, la cual se

¹Para simplificar las expresiones, los argumentos de todas las funciones serán omitidas, y serán explícitamente escritas sólo cuando la función sea definida por primera vez.

supone que es continuamente diferenciable. Las ecuaciones de movimiento, sin considerar fricción, pueden ser escritas con una estructura matemática tipo hamiltoniana, de la siguiente manera

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} 0_{2 \times 2} & I_{2 \times 2} \\ -I_{2 \times 2} & 0_{2 \times 2} \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{q}} H \\ \nabla_{\mathbf{p}} H \end{bmatrix} + \begin{bmatrix} \mathbf{0}_2 \\ G \end{bmatrix} u \quad (24)$$

donde $G = [1 \ 0]^\top$, y u es la entrada de control. El método IDA-PBC se caracteriza por asignar una función de energía deseada para el sistema en malla cerrada dada por:

$$H_d(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^\top M_d^{-1}(\mathbf{q}) \mathbf{p} + V_d(\mathbf{q}) \quad (25)$$

donde $M_d = M_d^\top > 0$, y V_d se supone que es continuamente diferenciable. El sistema en malla cerrada es propuesto con la siguiente estructura:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} 0_{2 \times 2} & \mathcal{M}^{-1} M_d \\ -M_d \mathcal{M}^{-1} & J_2(\mathbf{q}, \mathbf{p}) - G K_v G^\top \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{q}} H_d \\ \nabla_{\mathbf{p}} H_d \end{bmatrix} \quad (26)$$

donde $J = -J_2^\top$ y $K_v = K_v^\top > 0$ son matrices cuadradas de dimensión dos, y son *libres* para el usuario en el diseño de la ley de control. La principal dificultad en el diseño de la ley de control u es encontrar las soluciones M_d y V_d que resuelvan el siguiente conjunto de ecuaciones diferenciales parciales (PDEs):

$$G^\perp [\nabla_{\mathbf{q}} (\mathbf{p}^\top \mathcal{M}^{-1} \mathbf{p}) - M_d \mathcal{M}^{-1} \nabla_{\mathbf{q}} (\mathbf{p}^\top M_d^{-1} \mathbf{p}) + 2J_2 M_d^{-1} \mathbf{p}] = 0, \quad (27)$$

$$G^\perp [\nabla_{\mathbf{q}} V - M_d \mathcal{M}^{-1} \nabla_{\mathbf{q}} V_d] = 0, \quad (28)$$

tal que ambas soluciones M_d y V_d definen la ley de control en términos de un par, dada por:

$$u = [G^\top G]^{-1} G^\top [\nabla_{\mathbf{q}} H - M_d \mathcal{M}^{-1} \nabla_{\mathbf{q}} H_d + J_2 M_d^{-1} \mathbf{p} - K_v G^\top M_d^{-1} \mathbf{p}]. \quad (29)$$

Si M_d es definida positiva en una vecindad de \mathbf{q}^* , y

$$\mathbf{q}^* = \arg \min \{V_d\} \quad (30)$$

entonces $[\mathbf{q}^\top \ \mathbf{p}^\top]^\top = [\mathbf{q}^{*\top} \ \mathbf{0}_2^\top]^\top$ es un equilibrio estable del sistema en lazo cerrado (26) con una función de Lyapunov H_d . Finalmente, siguiendo los argumentos reportados en [3], bajo condiciones de detectabilidad se puede demostrar estabilidad asintótica y por tanto cumplir con el objetivo de control (22).

Para el sistema carro-péndulo, el modelo hamiltoniano (24) puede ser escrito con (2),

$$G = [1 \ 0]^\top, \quad (31)$$

y la función de energía potencial

$$V(q_2) = mgl \cos(q_2). \quad (32)$$

Por conveniencia de notación, y tomando en cuenta (7) y (21), los elementos de (2) y (32) se definen como:

$$\beta_1 = M + m + J_{eq}, \quad \beta_2 = ml, \quad \beta_3 = ml^2 + I, \quad \beta_4 = mgl \quad (33)$$

por tanto, la matriz (2) puede ser rescrita en forma compacta

$$\mathcal{M}(q) = \begin{bmatrix} \beta_1 & \beta_2 \cos(q_2) \\ \beta_2 \cos(q_2) & \beta_3 \end{bmatrix}. \quad (34)$$

Con el fin de cumplir con el objetivo de control (22), primero procedemos a diseñar un IDA-PBC en términos de un par. Enseguida, se sigue el procedimiento reportado en [7] y [8], donde

$$G^\perp = [0 \quad 1] \quad (35)$$

tal que $G^\perp G = 0$. Un paso importante para la solución de la PDE en (27) es la selección de la matriz

$$J_2 = \begin{bmatrix} 0 & \tilde{\mathbf{p}}^T \boldsymbol{\alpha}(q_2) \\ -\tilde{\mathbf{p}}^T \boldsymbol{\alpha}(q_2) & 0 \end{bmatrix} = \tilde{\mathbf{p}}^T \boldsymbol{\alpha} W \quad (36)$$

con $\tilde{\mathbf{p}} = M_d^{-1} \mathbf{p}$, $\boldsymbol{\alpha} = [\alpha_1(q_2) \quad \alpha_2(q_2)]^T$ y $W \in so(2)$. Más aún, utilizando la identidad

$$\frac{dM^{-1}}{dq_i} = -M^{-1} \frac{dM}{dq_i} M^{-1}$$

y siguiendo el procedimiento desarrollado en [8], donde definimos

$$\Lambda = \begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_3 & \lambda_4 \end{bmatrix} \triangleq M_d \mathcal{M}^{-1}$$

nos conduce al siguiente conjunto de ecuaciones:

$$\beta_2 \text{sen}(q_2) \Lambda \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \Lambda^T + \lambda_4 \frac{dM_d}{dq_2} - \begin{bmatrix} 2\alpha_1 & \alpha_2 \\ \alpha_2 & 0 \end{bmatrix} = 0 \quad (37)$$

Considerando de la definición de Λ , expresamos M_d como

$$M_d = \Lambda \mathcal{M} \triangleq \begin{bmatrix} d_1 & d_2 \\ d_3 & d_4 \end{bmatrix}$$

donde

$$\begin{aligned} d_1 &= \lambda_1 \beta_1 + \lambda_2 \beta_2 \cos(q_2), & d_2 &= \lambda_1 \beta_2 \cos(q_2) + \lambda_2 \beta_3, \\ d_3 &= \lambda_3 \beta_1 + \lambda_4 \beta_2 \cos(q_2), & d_4 &= \lambda_3 \beta_2 \cos(q_2) + \lambda_4 \beta_3. \end{aligned}$$

considerando los términos d_i anteriores en (37), resulta el siguiente sistema de ecuaciones algebraicas, las cuales no presentan obstáculo alguno en la solución de M_d , debido a que α_1 y α_2 son libres:

$$2\beta_2 \text{sen}(q_2) \lambda_1 \lambda_2 + \lambda_4 \frac{d}{dq_2} (\lambda_1 \beta_1 + \lambda_2 \beta_2 \cos(q_2)) - 2\alpha_1 = 0, \quad (38)$$

$$\beta_2 \text{sen}(q_2) [\lambda_1 \lambda_4 + \lambda_2 \lambda_3] + \lambda_4 \frac{d}{dq_2} (\lambda_1 \beta_2 \cos(q_2) + \lambda_2 \beta_3) - \alpha_2 = 0, \quad (39)$$

$$2\beta_2 \text{sen}(q_2) \lambda_3 \lambda_4 + \lambda_4 \frac{d}{dq_2} (\lambda_3 \beta_2 \cos(q_2) + \lambda_4 \beta_3) = 0. \quad (40)$$

De acuerdo a (28), resulta la siguiente PDE

$$\lambda_3 \nabla_{q_1} V_d + \lambda_4 \nabla_{q_2} V_d = -\beta_4 \text{sen}(q_2) \quad (41)$$

Con la asignación de $\lambda_3 = \cos(q_2)$ y $\lambda_4 = -k_4$ permite resolver (40), siendo k_4 una constante arbitraria. Una solución de la PDE en (41) es

$$V_d = -\frac{\beta_4 \cos(q_2)}{k_4} + \frac{k_p}{2} [-k_4(q_1 - q_{d1}) + \text{sen}(q_2)]^2 \quad (42)$$

con las constantes arbitrarias k_p estrictamente positiva y k_4 estrictamente negativa, que aseguran la positividad de V_d . Se procede con la asignación de los elementos de M_d , los cuales aseguran su simetría y positividad. Con este fin y realizando las manipulaciones algebraicas necesarias con (37)-(40), la matriz M_d queda de la siguiente manera:

$$M_d = \begin{bmatrix} k_1 & \cos(q_2)(\beta_1 + k_4 \beta_2) \\ \cos(q_2)(\beta_1 + k_4 \beta_2) & \beta_2 \cos(q_2)^2 + k_4 \beta_3 \end{bmatrix} \quad (43)$$

donde k_1 es una constante arbitraria estrictamente positiva. Por último, la ley de control (29) puede ser obtenida desarrollando cada uno de sus términos que son omitidos en el presente trabajo por razones de espacio.

III-B. Ley de control en voltaje

Una extensión del método IDA-PBC en la que se incorpora la dinámica de una clase de motores de CD con escobillas e imanes permanentes fue presentada en [6]. Enseguida, aplicamos esta extensión para rediseñar la ley de control (29) en términos del voltaje del único motor de CD utilizado para controlar el sistema carro-péndulo.

Tomando en cuenta (9), (7) y (21), el modelo dinámico completo del motor de CD puede ser escrito convenientemente de la siguiente manera:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} 0_{2 \times 2} & I_{2 \times 2} & 0_{2 \times 2} \\ -I_{2 \times 2} & 0_{2 \times 2} & A \\ 0_{2 \times 2} & -A^T & -B \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{q}} H \\ \nabla_{\mathbf{p}} H \\ \nabla_{\mathbf{z}} W_L \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ C \end{bmatrix} V_a \quad (44)$$

donde $\mathbf{z} = [i_a \quad 0]^T$, $A = \frac{k_g}{L_a} G G^T$, $B = \frac{R_a k_g}{k_t L_a^2} G G^T$, $C = \frac{1}{L_a} G G^T$ y el término $G u$ de (24) ha sido incluido en (44) por medio de la corriente de armadura como $G u = A \nabla_{\mathbf{z}} W_L$, esto es

$$G u = G G^T K_T \mathbf{z} \quad (45)$$

siendo $K_T = \text{diag}\{k_t, 0\}$, y la función semi-definida positiva

$$W_L(\mathbf{z}) = \frac{1}{2} \frac{k_t L_a}{k_g} i_a^2.$$

Las constantes k_g y k_t comúnmente son proporcionadas por el fabricante. Por otro lado, el sistema en malla cerrada se propone con la siguiente estructura matemática:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} 0_{2 \times 2} & \mathcal{M}^{-1} M_d & 0_{2 \times 2} \\ -M_d \mathcal{M}^{-1} & C_d & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & -G G^T \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{q}} H_d \\ \nabla_{\mathbf{p}} H_d \\ \nabla_{\mathbf{z}} W_d \end{bmatrix} \quad (46)$$

donde $C_d = J_2 - G K_v G^T$ con $J_2 = -J_2^T$ y $K_v = K_v^T > 0$ son matrices cuadradas de dimensión dos [3], y una función cuadrática semi-definida positiva:

$$W_d(\mathbf{z}) = \frac{1}{2} k_d [i_a - i_a^*]^2 \quad (47)$$

donde k_d es una constante arbitraria estrictamente positiva, e i_a^* es definida como:

$$i_a^* = \frac{1}{k_t} \nabla_{q_1} V(\mathbf{q}^*) \quad (48)$$

donde el vector \mathbf{q}^* está dado por

$$\mathbf{q}^* = [q_{d1} \ 0]^\top.$$

Igualando $\frac{d}{dt}[\mathbf{q}^\top \ \mathbf{p}^\top]^\top$ de (44) con (26) se obtiene

$$k_t G G^\top \mathbf{z} = \nabla \mathbf{q} H - M_d \mathcal{M}^{-1} \nabla \mathbf{q} H_d + D M_d^{-1} \mathbf{p} \quad (49)$$

donde $D = J_2 - G K_v G^\top$. Al multiplicar ambos lados de la igualdad de (49) por la matriz identidad $\begin{bmatrix} G^\top \\ G^\perp \end{bmatrix}$, considerando (31) y (35), esto es,

$$\begin{bmatrix} G^\top \\ G^\perp \end{bmatrix} k_t G G^\top \mathbf{z} = \begin{bmatrix} G^\top \\ G^\perp \end{bmatrix} [\nabla \mathbf{q} H - M_d \mathcal{M}^{-1} \nabla \mathbf{q} H_d + D M_d^{-1} \mathbf{p}] \quad (50)$$

se obtienen las *ecuaciones de igualación* (27) y (28) introducidas en [3]. Las soluciones M_d y V_d definen la corriente de armadura i_a de (50), que resulta del producto:

$$k_t G^\top \mathbf{z} = G^\top [\nabla \mathbf{q} H - M_d \mathcal{M}^{-1} \nabla \mathbf{q} H_d + D M_d^{-1} \mathbf{p}] \quad (51)$$

donde $G^\top G G^\top = G^\top$, o bien, la corriente i_a puede obtenerse

$$\begin{bmatrix} i_a \\ 0 \end{bmatrix} = \frac{1}{k_t} [\nabla \mathbf{q} H - M_d \mathcal{M}^{-1} \nabla \mathbf{q} H_d + D M_d^{-1} \mathbf{p}] \quad (52)$$

Note que i_a sólo depende de las posiciones y velocidades del sistema mecánico, esto es, $i_a = i_a(\mathbf{q}, \mathbf{p})$, tal que,

$$i_a(\mathbf{q}^*, \mathbf{0}_2) = \frac{1}{k_t} \nabla_{q_1} V(\mathbf{q}^*) \quad (53)$$

ya que por diseño $\nabla \mathbf{q} V_d(\mathbf{q}^*) = \mathbf{0}_2$, lo cual explica (48). Por otro lado, igualando $\frac{d\mathbf{z}}{dt}$ de (44) con (46), tomando en cuenta que $A = A^\top$ y después de algunas manipulaciones algebraicas, se obtiene la ley de control en voltaje:

$$V_a = -L_a k_d [i_a - i_a^*] + \frac{k_g}{\det[\mathcal{M}]} [a_3 p_1 - a_2 p_2] + R_a i_a \quad (54)$$

con i_a calculado de (52) e i_a^* definido por (48).

IV. RESULTADOS EXPERIMENTALES

Se llevaron a cabo experimentos sobre un sistema carro-péndulo construido en el Tecnológico Nacional de México, Instituto Tecnológico de La Paz (ver figura 3), para validar el desempeño del controlador (54), con i_a despejado de (52).

Para la ejecución del algoritmo de control se utilizó Matlab[®] junto con Simulink y una aplicación de Quarc[®], instalado en un equipo de cómputo que aloja la tarjeta de adquisición de datos MultiQ-PCI[®], y configurando un periodo de muestreo de 1 [ms]. Se estableció un retardo de tiempo de dos segundos, para después iniciar la ejecución del algoritmo, con el fin de colocar manualmente el péndulo cerca de su posición vertical superior y el carro en una posición cercana al origen. La posición deseada del carro fue $q_{d1} = 0$, mientras las ganancias del controlador (54) fueron $k_1 = 2500$, $k_p = 0,5$, $k_d = -1$, $k_v = 250$ y $k_d = 1$. Los parámetros de la unidad SRV02 de Quanser junto con sus valores numéricos son enlistados en la Tabla I.



Figura 3. Sistema carro-péndulo construido en el Tecnológico Nacional de México, Instituto Tecnológico de La Paz.

Tabla I
PARÁMETROS DE LA UNIDAD SRV02 [9]

Símbolo	Descripción	Valor [Unidad]
R_a	Resistencia de armadura	2,6 [Ω]
k_t	Constante de par del motor	$7,68 \times 10^{-3}$ [$\frac{\text{Nm}}{\text{A}}$]
k_m	Constante de fuerza contraelectromotriz	$7,68 \times 10^{-3}$ [$\frac{\text{V}}{\text{rad/s}}$]
k_g	Relación de engranes	70
J_m	Momento de inercia del rotor	$3,9 \times 10^{-7}$ [kg m^2]
B_m	Coefficiente de fricción viscosa	$1,5 \times 10^{-4}$ [$\frac{\text{Nm}}{\text{rad/s}}$]
m	Masa del péndulo	0,127 [kg]
l	Distancia del pivote al centro de masa	0,156 [m]
M	Masa del carro	0,258 [kg]

Para comprobar que el voltaje medido en el motor corresponde efectivamente al voltaje suministrado por el controlador (54), se construyó un simple circuito eléctrico, mostrado en la Figura 4, para medir, guardar y visualizar el voltaje del motor (M) por medio de una tarjeta Arduino Uno[®]. Esta tarjeta tiene la capacidad de leer voltajes analógicos de 0 – 5 [V], mientras que la fuente controlada de CD de la marca Quanser[®], denotada en la figura 4 por E_g , suministra el voltaje al motor de acuerdo a (54). La fuente se configuró para limitar el voltaje hasta ± 15 [V], con el fin de proteger el motor.

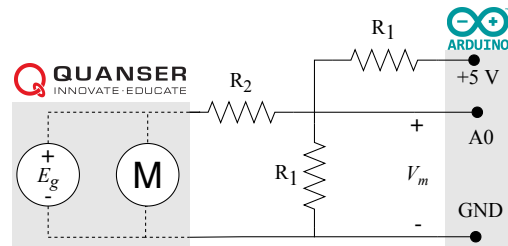


Figura 4. Diagrama eléctrico para medir el voltaje V_a dado por (54).

Después de un análisis se obtuvo la relación entre el voltaje medido V_m y el voltaje E_g de la fuente controlada de CD que

corresponde a la entrada de control (54), siendo:

$$V_m = \frac{1}{7}[E_g + 15] \text{ [V]} \quad (55)$$

Los valores de los resistores en el diagrama eléctrico son $R_1 = 10 \text{ M}[\Omega]$ y $R_2 = 30 \text{ M}[\Omega]$, y el pin A0 corresponde a una entrada analógica de la tarjeta Arduino Uno®.

Las gráficas mostradas en la Figura 5 corroboran el cumplimiento del objetivo de control (22), esto es, el desplazamiento del carro q_1 y la posición articular del péndulo q_2 , tienden a los valores deseados. Puede notarse un comportamiento oscilatorio del desplazamiento del carro alrededor de cero, el cual está asociado a varios factores como: proceso de discretización del algoritmo de control, velocidades estimadas a través del método numérico de Euler; así como a otros fenómenos no modelados como la fricción y las inercias de poleas y engranes. El desempeño del controlador y su evolución temporal se muestra en la figura 6, donde se han graficado tanto el voltaje calculado V_a de (54) como el voltaje medido V_m por (55). Al comparar ambos voltajes se observa una diferencia casi nula, esto es, $V_a \approx V_m$, y se puede concluir que efectivamente la ley de control (54) corresponde al voltaje que se le suministra al motor.

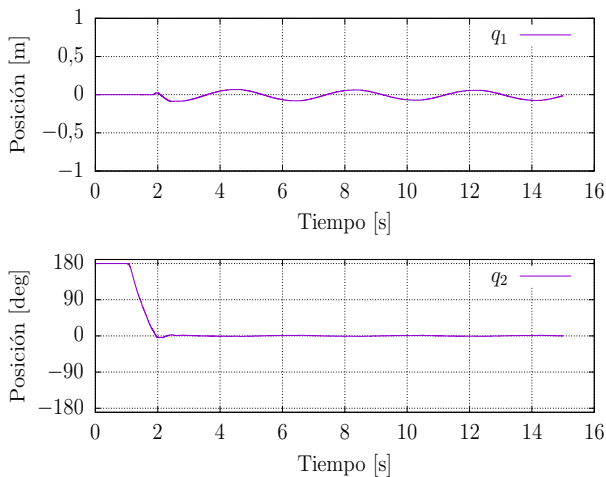


Figura 5. Evolución temporal de las posiciones del carro y el péndulo, respectivamente.

V. CONCLUSIONES

Se ha presentado un regulador de posición en voltaje para un sistema carro-péndulo. Para validar el sistema se mostraron resultados experimentales donde se logró llevar al péndulo a la posición vertical superior con el carro en una posición deseada, cumpliéndose así el objetivo de control (22). Además, se pudo concluir que la ley de control (54) corresponde al voltaje que se le aplica al motor, a pesar que el controlador depende explícitamente de las posiciones y velocidades del sistema carro-péndulo y de los parámetros del motor. Esto significa que no se requiere medir directamente la corriente de armadura.

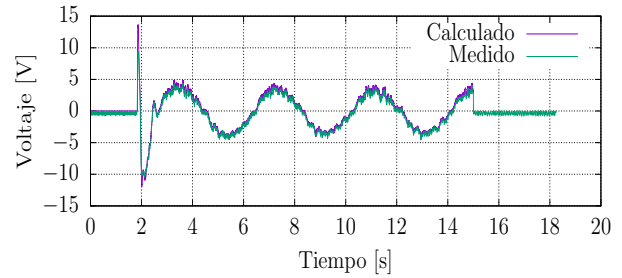


Figura 6. Evolución temporal de la ley de control en voltaje calculado de acuerdo a (54) y el voltaje medido (55) usando el circuito de la figura 4.

AGRADECIMIENTOS

Este trabajo fue parcialmente financiado por los proyectos Conacyt 166636, 166654, 134534, y por proyectos del TecNM.

REFERENCIAS

- [1] I. Siradjuddin, B. Setiawan, A. Fahmi, Z. Amalia, and E. Rohadi (2017), "State space control using LQR method for a cart-inverted pendulum linearised model", *International Journal Mechanical Mechatronics Engineering*, 17(1), 119–126.
- [2] R. Kelly, J. Sandoval y C. Monroy (2005), "Estabilización del sistema carro-péndulo: análisis del método por lagrangiano controlado", *Memorias del Congreso Nacional de Control Automático*, Cuernavaca, Morelos.
- [3] R. Ortega, M. Spong, F. Gómez-Estern and G. Blankenstein (2002), "Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment", *IEEE Transactions on Automatic Control*, 47(8), 1213–1233.
- [4] A. M. Bloch, N. Leonard, & J. Marsden (2000), Controlled Lagrangian and the stabilization of mechanical systems I: the first matching theorem. *IEEE Transactions on Automatic Control*, 45(12), 2253–2270.
- [5] G. Viola, R. Ortega, J. Banavar, J. A. Acosta & A. Astolfi (2007), Total energy shaping control of mechanical systems: simplifying the matching equations via coordinate changes. *IEEE Transactions on Automatic Control*, (52)6, 1093–1099.
- [6] J. Sandoval, R. Kelly, & V. Santibáñez (2010), Regulation of mechanisms with friction driven by brushed DC motors via IDA-PBC method. *Proceedings of IEEE Conference Decision and Control* (pp. 6225–6229), Atlanta, GA, USA.
- [7] J. A. Acosta, A. Fradkov & F. Gordillo (2005), Speed-gradient algorithms for underactuated nonlinear systems. *Proceedings of the 16th World Congress the International Federation of Automatic Control*, (pp. 842–847), Prague, Czech Republic.
- [8] J. Sandoval, R. Kelly & R. Ortega (2008), *Interconnection and damping assignment passivity based control of the pendubot*, (pp. 7700–7704), 17th IFAC, Seoul, Korea.
- [9] R. King, (2001), User Manual–SRV02 Rotary Servo Base Unit, QUANSER®, 3ed. edition.

Modelado de sistemas mediante regresión basada en un AG

Joel Artemio Morales Viscaya
Tecnológico Nacional de México
Instituto Tecnológico de La Paz
Baja California Sur, México
Email: iscviscaya@gmail.com

Carlos Antonio Olachea Hernández
Tecnológico Nacional de México
Instituto Tecnológico de La Paz
Baja California Sur, México
Email: antonio.mx.9605@gmail.com

Marco Antonio Castro Liera
Tecnológico Nacional de México
Instituto Tecnológico de La Paz
Baja California Sur, México
Email: mcastroliera@gmail.com

Resumen—En este trabajo se presenta una estrategia para el modelado de sistemas basada en un algoritmo genético, el algoritmo produce modelos para problemas de hasta tres variables independientes combinando linealmente funciones simples. Las funciones se escogen mediante una versión de un AG para una representación en que el cromosoma de cada individuo es una combinación, por lo que se presentan propuestas de operadores genéticos para dicha representación. Los coeficientes de las combinaciones lineales se producen mediante plantear el problema como uno de mínimos cuadrados. Se presenta también una interfaz gráfica para el algoritmo en Java así como pruebas de su desempeño en un par de conjuntos de datos.

I. INTRODUCCION

Con frecuencia es deseable describir en términos matemáticos el comportamiento de algunos sistemas o fenómenos de la vida real, ya sean físicos, sociológicos o incluso económicos. La descripción matemática de un sistema de fenómenos se llama modelo matemático y se construye con objetivos específicos. Por ejemplo, podemos desear controlar la dinámica de cierto sistema electromecánico al estudiar la relación entre sus variables de estado, o podemos analizar el decaimiento de una sustancia radiactiva ya sea en el fósil o en el estrato en que fué descubierto para datarlo [1].

El carácter empírico o teórico constituye la característica fundamental de un modelo. Un modelo teórico se basa en las leyes físicas que rigen los procesos, un modelo empírico se basa en las observaciones directas o los resultados de experimentos del fenómeno estudiado. Si bien en muchos casos los modelos teóricos son preferibles dado su mayor capacidad explicativa, modelos empíricos han demostrado poseer un mayor poder predictivo en sistemas complejos [2].

A la utilización de técnicas estadísticas para estimar las relaciones entre variables se le conoce como regresión, el principal problema de la regresión paramétrica es que presupone una forma explícita para la relación entre las variables en el modelo.

La estrategia que se propone es un algoritmo para construir modelos empíricos basado en las técnicas de regresión, pero sin presuponer una forma específica para el modelo, lo que algunos autores llaman regresión no paramétrica, con la particularidad de no producir un sistema donde las relaciones entre las variables sean complejas (casi uno de caja negra o gris), como ocurre con técnicas más avanzadas de modelación, como

son las redes neuronales artificiales o los sistemas basados en lógica difusa.

Las expresiones matemáticas que produce el algoritmo propuesto son combinaciones lineales y productos de funciones simples escogidas utilizando una adaptación de un algoritmo genético (AG), cuyos coeficientes se calculan de manera que minimicen la norma cuadrada del error.

La principal ventaja de aproximar información discreta o relaciones complicadas entre variables, con funciones analíticas sencillas, radica en su mayor facilidad de evaluación y manipulación, situación necesaria en ingeniería.

Los algoritmos genéticos, surgen en los años 1970, de la mano de John Herny Holland [3], son llamados así porque se inspiran en la evolución biológica y su base genético-molecular, actualmente constituyen una de las líneas más prometedoras de la inteligencia artificial.

Estos algoritmos hacen evolucionar una población de individuos o soluciones del problema que se pretende resolver, someténdola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.

II. REPRESENTACIÓN DE LOS INDIVIDUOS

Los cinco componentes necesarios para implementar un AG que resuelva un problema cualquiera, son [4]:

1. Una representación de soluciones potenciales al problema.
2. Una función de evaluación que juega el papel del ambiente, calificando a las soluciones producidas en términos de su "aptitud".
3. Una forma de crear una población inicial de soluciones potenciales (esto se efectúa normalmente de manera aleatoria, pero también pueden usarse métodos determinísticos).
4. Operadores genéticos que alteran la composición de los descendientes (normalmente se usan la cruza y la mutación).
5. Valores para los diversos parámetros utilizados por el algoritmo genético (tamaño de la población, probabilidad

de cruza y mutación, número máximo de generaciones, etc.)

Debido a que las funciones que se van a combinar linealmente para cada individuo se escogen de un banco de funciones candidatas, el cromosoma o la representación de las soluciones, es una combinación que toma n elementos de los N totales que hay en el banco, que se almacena como un arreglo de dimensión n , al que llamaremos $C = [c_1, c_2, \dots, c_n]$.

Sin embargo, en el algoritmo propuesto cada combinación no es directamente una solución al problema de regresión, sino la estructura de la función que se utilizará para construir el modelo (el cromosoma C determina cuáles funciones se utilizarán).

Para representar a los individuos dentro de la población de soluciones del algoritmo genético se utilizó una estructura de datos. Esta consta de las propiedades C Cromosoma, con la estructura de la función escogida para hacer regresión, $\{a\}$, con las constantes particulares que producen el mejor ajuste de dicha estructura y AP con la aptitud o calidad de la solución, como se aprecia en la figura (1).

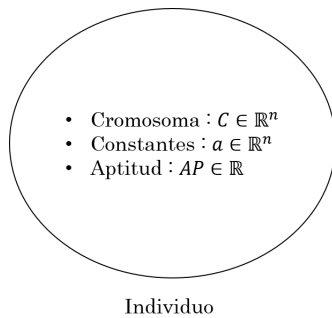


Figura 1. Representación gráfica de un individuo.

Las funciones que se pueden combinar linealmente para construir el modelo, es decir, el banco de funciones de la implementación del algoritmo, incluyen polinomios, exponenciales, gaussianas, sigmoideas así como funciones trigonométricas básicas y sus productos.

Cada función está asociada a un índice al que se hace referencia mediante los elementos que contiene la combinación (alelos, en terminología de AG); es importante remarcar que la representación tradicional de permutaciones no es adecuada para este problema, ya que el espacio vectorial de funciones que genera un conjunto de n funciones es el mismo sin importar en que orden se combinen linealmente estas.

El número de funciones N en el banco, que directamente es el número de funciones distintas que se pueden utilizar para la construcción del modelo, es un parámetro de diseño. El valor mínimo de N depende del número de variables independientes, once para una sola, 34 para dos variables y 79 para tres variables independientes. Al incrementar el parámetro se van agregando al banco de funciones seleccionables los productos, estiramientos y encogimientos de estas. De forma que no existe un valor máximo para N . Sin embargo, valores demasiado

grandes para N implican un considerable incremento en el tiempo de ejecución del algoritmo.

El número de funciones que se van a combinar linealmente para construir los individuos, que es equivalente a la longitud del cromosoma, es otro parámetro de diseño al que denotamos como n . Más funciones producen un mejor ajuste pero al mismo tiempo modelos más complejos de analizar analíticamente. Determinar dicho parámetro lleva a la conocida discusión del compromiso entre buen ajuste a los datos y la simplicidad en el modelo.

Para el caso $N = 31$, $n = 5$, un individuo cuyo cromosoma sea como el de la figura (2).

$$C = \begin{bmatrix} 1 & 8 & 15 & 19 & 24 \end{bmatrix}$$

Figura 2. Representación de los alelos de un cromosoma.

representa a la función $\phi(x) = a_1\phi_1(x) + a_2\phi_8(x) + a_3\phi_{15} + a_4\phi_{19} + a_5\phi_{24}$, en este caso

$$\phi_1 = e^x \quad \phi_8 = e^x \cos x \quad \phi_{15} = \frac{1}{1 + e^{-2x}}$$

$$\phi_{19} = e^{2x} \cos 2x \quad \phi_{24} = e^{3x} x^3$$

Es decir, el cromosoma representa la función

$$\phi(x) = a_1 e^x + a_2 e^x \cos x + a_3 \frac{1}{1 + e^{-2x}} + a_4 e^{2x} \cos 2x + a_5 e^{3x} x^3$$

III. ESTRUCTURA GENERAL DEL ALGORITMO PROPUESTO

El primer paso del algoritmo consiste en la carga y normalización de los datos con los que se llevará a cabo la regresión, debido a que es posible que exista una diferencia significativa en el orden de magnitud de los valores entre distintas variables independientes.

Para llevar a cabo la normalización de los datos, se calcula primero el valor máximo y mínimo de cada variable independiente. Posteriormente para cada valor x_i de la variable independiente x se produce \bar{x}_i utilizando (1).

$$\bar{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Posteriormente se genera una población inicial de individuos, cuyos cromosomas se construyen aleatoriamente mediante combinaciones de números enteros sin repetición, cuyo rango va desde cero hasta el tamaño del banco N , como se puede apreciar en el siguiente pseudocódigo.

Algorithm Inicialización de los individuos.

- 1: $nL \leftarrow$ longitud de C
- 2: **for** $i = 0$ **to** nL **do**
- 3: Generar valor aleatorio $r \in \{0 \dots N \mid r \notin C\}$.
- 4: $c_i \leftarrow r$
- 5: **end for**

Los pasos restantes, la parte principal del algoritmo genético que se muestra en la figura (3), se llevan a cabo iterativamente en un ciclo que se repite tantas veces como un parámetro de diseño $Gmax$.

En este ciclo, a cada individuo de la población se le calcula su aptitud y se le aplica una sucesión de operadores genéticos (cruza y mutación) con el objetivo de generar nuevos individuos que sean potencialmente mejores en términos de la aptitud.

Estos nuevos individuos junto con algunos de los mejores de la población actual, pasarán a la siguiente población, en un símil con la selección natural.

Además se utiliza un enfoque conocido como elitismo, que conserva al mejor individuo de la población en cada una de las iteraciones $Gmax$, esto con el propósito de que el mejor individuo de la población (que representa la solución del AG al problema), en la iteración $k + 1$ sea al menos tan bueno como el de la iteración k [5].

IV. EL CÁLCULO DE APTITUD

La evolución de los individuos en un algoritmo genético depende directamente del cálculo de aptitud de los mismos, este permite evaluar que tan buena es una solución para resolver el problema en particular y, mediante un algoritmo de selección, preservar dicha solución en posteriores iteraciones.

En este caso al tratarse de un problema de regresión, y poseer los individuos solamente la estructura de la función a utilizar, el cálculo no es demasiado simple y puede ser dividido en dos partes.

1. Encontrar los coeficientes que permitan un máximo ajuste de la función de regresión a los datos.
2. Calcular un índice de desempeño para determinar que tan bueno es el ajuste de una función de regresión.

Formalmente la primera parte del problema puede plantearse como el de, dado un conjunto de m pares ordenados (x, y) y un conjunto de funciones de la forma

$$\phi_i(x), i \in \{1, 2, \dots, n\} \quad (2)$$

encontrar los valores de a_1, a_2, \dots, a_n que hacen que la función de aproximación

$$\phi(x) = a_1\phi_1(x) + a_2\phi_2(x) + \dots + a_n\phi_n(x) \quad (3)$$

sea tal que para toda $i \in \{1, 2, \dots, m\}$

$$\phi(x_i) \approx y_i. \quad (4)$$

Primero es necesario definir el índice de desempeño, que utilizaremos para determinar cuando decimos que $\phi(x) \approx y$.

Esta cercanía entre $\phi(x)$ y y la definimos en el sentido de mínimos cuadrados como

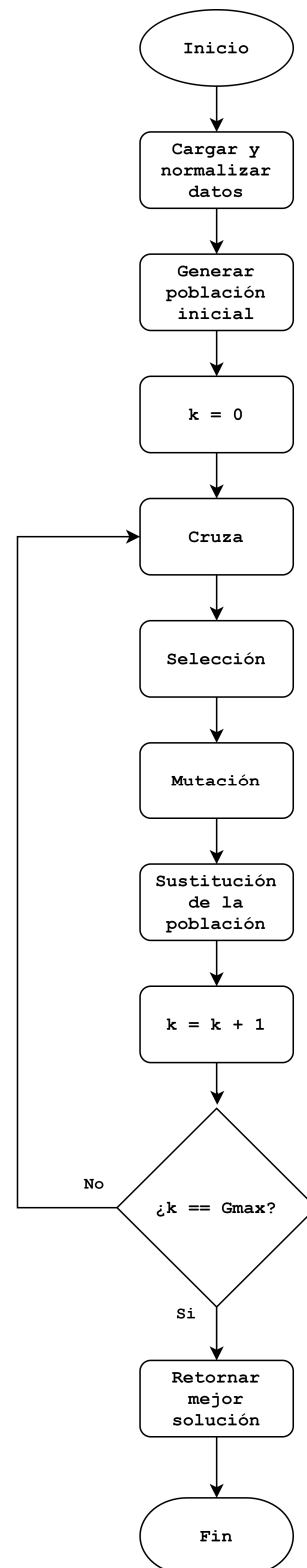


Figura 3. Diagrama del algoritmo genético.

$$\min_{(a_1, a_2, \dots, a_n)} E = \frac{1}{2} \sum_{k=1}^m [\phi(x_k) - y_k]^2 \quad (5)$$

esto debido principalmente a que es posible encontrar los coeficientes de máximo ajuste a_1, a_2, \dots, a_n a partir de este índice de desempeño y a que se utiliza con frecuencia en la literatura [6].

Sustituyendo (3) en (5) tenemos

$$E = \frac{1}{2} \sum_{k=1}^m [a_1 \phi_1(x_k) + a_2 \phi_2(x_k) + \dots + a_m \phi_m(x_k) - y_k]^2 \quad (6)$$

es decir

$$E = \frac{1}{2} \sum_{k=1}^m \left[\sum_{j=1}^n a_j \phi_j(x_k) - y_k \right]^2. \quad (7)$$

Para resolver el problema de minimización (5) de manera analítica es necesario encontrar el vector gradiente de E respecto a a_1, a_2, \dots, a_n .

Derivando E de (7) respecto a a_i se tiene

$$\frac{\partial E}{\partial a_i} = \sum_{k=1}^m \left[\sum_{j=1}^n a_j \phi_j(x_k) - y_k \right] \phi_i(x_k). \quad (8)$$

Para cada i , queremos los valores a_j que hacen que $\frac{\partial E}{\partial a_i} = 0$. Esto es

$$\sum_{j=1}^n \left[\sum_{k=1}^m \phi_j(x_k) \phi_i(x_k) \right] a_j = \sum_{k=1}^m \phi_i(x_k) y_k. \quad (9)$$

Para obtener directamente la expresión de la solución de (9) hay que notar que para cada i , se tiene que el error está dado por

$$e_i = \phi(x) - y_i = \sum_{j=1}^n a_j \phi_j(x_i) - y_i. \quad (10)$$

Si

$$\bar{\phi} = \begin{pmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_n(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_m) & \phi_2(x_m) & \dots & \phi_n(x_m) \end{pmatrix},$$

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \text{ y } \vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

entonces el vector de errores es

$$\vec{e} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{pmatrix}$$

Y el problema a resolver (5) es equivalente a minimizar

$$E = \|\vec{e}\|^2 = \vec{e}^T \vec{e} = (\bar{\phi} \vec{a} - \vec{y})^T (\bar{\phi} \vec{a} - \vec{y}) \quad (11)$$

El cálculo del gradiente de E produce (9) que puede ser escrita en términos de $\bar{\phi}$, \vec{a} y \vec{y} como

$$\bar{\phi}^T \bar{\phi} \vec{a} = \bar{\phi}^T \vec{y} \quad (12)$$

cuya solución por mínimos cuadrados produce directamente los coeficientes a_1, a_2, \dots, a_n de mejor ajuste de la función de regresión para cada individuo. El valor de aptitud de cada individuo resulta en el valor de la función de error E (7) con los coeficientes encontrados.

V. OPERADORES GENÉTICOS

En los sistemas biológicos, la cruce es un proceso complejo que ocurre entre parejas de cromosomas. Estos cromosomas se alinean, luego se fraccionan en ciertas partes y posteriormente intercambian fragmentos entre sí [4].

En el cómputo evolutivo tradicional se simula la cruce intercambiando segmentos de cadenas lineales de longitud fija (los cromosomas padre), produciendo de esta manera nuevos individuos (hijos) con propiedades similares a las de los padres.

La técnica de cruce desarrollada en esta estrategia, mostrada en el Algorithm Cruza, está orientada a problemas de combinatoria donde el orden de los alelos es irrelevante, es decir, solo importan los valores en el cromosoma y no el orden en que aparezcan en el individuo.

Se seleccionan aleatoriamente dos individuos en la población, a los que llamaremos padres. Los alelos de los padres se concatenan y se ordenan, resultando en una lista, con la cuál se generan dos nuevos individuos (hijos) a partir de tomar, en cada caso, los elementos impares o pares de dicha lista. Esto garantiza, que los alelos de los padres se transmiten y que además, los hijos no tendrán alelos repetidos (y las características comunes de los padres, las tendrán ambos hijos).

Los hijos generados mediante la cruce, corresponderán a la mitad de individuos de la nueva población y el resto serán elegidos mediante un proceso de selección que se será descrito más adelante.

En el siguiente algoritmo se aprecia la cruce que se explicó anteriormente.

Algorithm Cruza.

```

1:  $CP$  es la población actual.
2:  $NP$  es la población nueva.
3:  $PL \leftarrow$  longitud de  $CP$ .
4: for  $i \leftarrow 0$  to  $PL/2$  do
5:   Seleccionar dos padres  $F_1$  y  $F_2$  de  $CP$ .
6:   Concatenar las  $c$ 's de  $F_1$  y  $F_2$  en  $L$ .
7:   Ordenar  $L$ .
8:   Crear hijos  $S_1$  y  $S_2$ .
9:   Declarar  $LS \leftarrow$  longitud de  $L$ .
10:  Declarar indexador  $k \leftarrow 0$ .
11:  for  $j = 0$  to  $LS$  do
12:     $S_{1.c_j} \leftarrow L_k$ .
13:     $S_{2.c_j} \leftarrow L_{k+1}$ .
14:     $k \leftarrow k + 2$ .
15:  end for
16:  if  $S_1.C = F_1.C$  or  $S_1.C = F_2.C$  then
17:    Copiar aptitud y constantes de  $F$  similar, a  $S_1$ .
18:  else
19:    Evaluar  $S_1$ 
20:  end if
21:  if  $S_2.C = F_1.C$  or  $S_2.C = F_2.C$  then
22:    Copiar aptitud y constantes de  $F$  similar, a  $S_2$ .
23:  else
24:    Evaluar  $S_2$ 
25:  end if
26:   $NP_i \leftarrow S_1$ .
27:   $NP_{i+1} \leftarrow S_2$ .
28: end for
    
```

Para la selección, cuyo pseudocódigo se muestra en el Algorithm Selección, de los mejores individuos, se utilizó la estrategia conocida como selección por torneo, debido a que su complejidad es de orden menor a otras estrategias y además no produce una presión de selección muy alta [4].

Se considera un tamaño de torneo $t = 2$, donde se comparan las aptitudes AP de dos individuos seleccionados aleatoriamente de la población actual, y aquel con menor aptitud pasará a formar parte de la nueva población.

Algorithm Selección.

```

1: for  $i \leftarrow PL/2$  to  $PL - 1$  do
2:   Seleccionar dos individuos  $I_1$  y  $I_2$  de  $CP$ .
3:   if  $I_1.AP < I_2.AP$  then
4:      $NP_i \leftarrow I_1$ .
5:   else
6:      $NP_i \leftarrow I_2$ .
7:   end if
8: end for
    
```

Además se considera elitismo, lo cual garantiza la permanencia de la mejor solución en cada iteración, al colocarlo como el último elemento de la población siguiente NP en cada iteración.

Se denomina mutación a un operador que forma un nuevo

cromosoma a través de alteraciones (usualmente pequeñas) de los valores de los genes de uno de los cromosomas.

La mutación se considera un operador secundario en algoritmos genéticos tradicionales [4]. Es decir, su uso es menos frecuente que el de la cruza. El objetivo de la mutación es el de evitar un estancamiento en óptimos locales; valores altos para la probabilidad de mutación se asocian a una búsqueda más en anchura que en profundidad y mantienen una mayor diversidad en las soluciones.

Debido al carácter secundario de la mutación en un AG, este operador solo se aplica a algunos individuos de la población, estos se escogen a partir de generar un número aleatorio y en caso de superar un parámetro de diseño PM , el individuo se mutará.

La estrategia utilizada para llevar a cabo la mutación, se muestra en el Algorithm Mutación. Consiste en tomar de forma aleatoria algún gen del cromosoma (una posición entre 0 y n del arreglo) e insertar un alelo distinto (un número entero entre 0 y N) que no se encuentre en el cromosoma, lo que se conoce como una mutación por inserción.

Algorithm Mutación.

```

1: for  $i \leftarrow 0$  to  $PL$  do
2:   Generar valor aleatorio  $r_1 \in (0, 1)$ .
3:   if  $r_1 \leq PM$  then
4:     Generar valor aleatorio  $r_2 \in \{0 \dots N \mid r_2 \notin NP_i.C\}$ .
5:     Seleccionar  $c_k$  aleatoria de  $NP_i$ .
6:      $NP_i.c_k \leftarrow r_2$ .
7:   end if
8: end for
    
```

VI. IMPLEMENTACIÓN DEL ALGORITMO

El algoritmo desarrollado se implementó en Java 8 utilizando la IDE de NetBeans 8.2. Además se le dotó de una interfaz gráfica de usuario. Esta, inicialmente proporciona una pantalla en la cual se pueden ajustar los parámetros del algoritmo genético y la regresión como se ve en la figura (4).

Figura 4. Menú de configuración del software.

Una vez introducidos los parámetros del algoritmo, es posible seleccionar en el menú file, la opción cargar datos, desde la cual se selecciona un archivo csv con los valores que se utilizarán en la regresión. Además de cargarlos en memoria, los valores de las variables independientes se escalan entre 0

y 1 como se mencionó anteriormente y se muestran en forma de tabla en la parte derecha de la ventana, como se aprecia en la figura 5.

Datos del Archivo		
C1	C2	C3
0.6471154401432173	0.9922202777590811	3.7121
0.6936995899824827	0.8414874582127393	3.9542
0.7403029894704422	0.7129565749029639	4.1919
0.7860594044158695	0.6033463462789707	4.42
0.8299678530866812	0.5098845101074692	4.6328
0.8709311055073244	0.43023210159072045	4.8237
0.9076786848640012	0.36241474276996255	4.986
0.9389208646942193	0.3047679544995625	5.1133
0.9634256674815684	0.2558880774494627	5.1997
0.9800573639531078	0.21458599762177205	5.2407
0.9879497199176116	0.17984227412443068	5.2335
0.9865637452116499	0.15078189854389626	5.1777
0.9758224412404473	0.12663783626124608	5.0751
0.9560723016804943	0.10673419936730162	4.9303
0.9281988103717107	0.09046381054946043	4.7493
0.8933569462357311	0.07728119180633149	4.5398
0.8530289322219868	0.0667011621906621	4.3098
0.8087547402259909	0.05829042426690001	4.0671

Figura 5. Tabla de los datos cargados.

Con los datos cargados, se habilita la opción de generar el modelo, la cuál no sólo lleva a cabo el algoritmo mencionado anteriormente, sino que además lo presenta de manera explícita y la despliega en una gráfica (en caso de ser una o dos variables independientes) en la que también pueden ver los datos como puntos rojos.

Si se trata de una función de una variable independiente se muestra la curva de regresión $\phi = f(x)$, como en la figura (6). En caso de dos variables independientes se muestra la superficie $\phi = f(x_1, x_2)$, como en la figura (7). Para funciones de tres variables independientes no se despliega ninguna gráfica (ya que su gráfica vive en un espacio de cuatro dimensiones).

El programa muestra adicionalmente la magnitud del error, así como el tiempo de ejecución del algoritmo, como se puede ver en la figura 8.

En la parte inferior se muestra la ecuación interpretada que mejor se ajusta al set de datos. Se dan las opciones de exportar la ecuación en formato **png** o copiar la ecuación en formato **LATEX** para poder ser insertada en documentos compatibles.

Finalmente en la figura 10, se muestra la pantalla completa del software y la ubicación de los componentes anteriormente descritos.

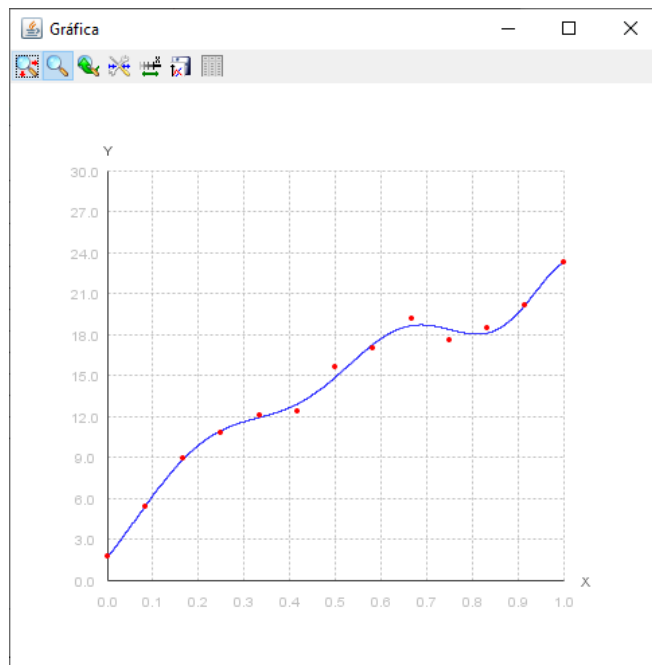


Figura 6. Ejemplo de una curva de regresión.

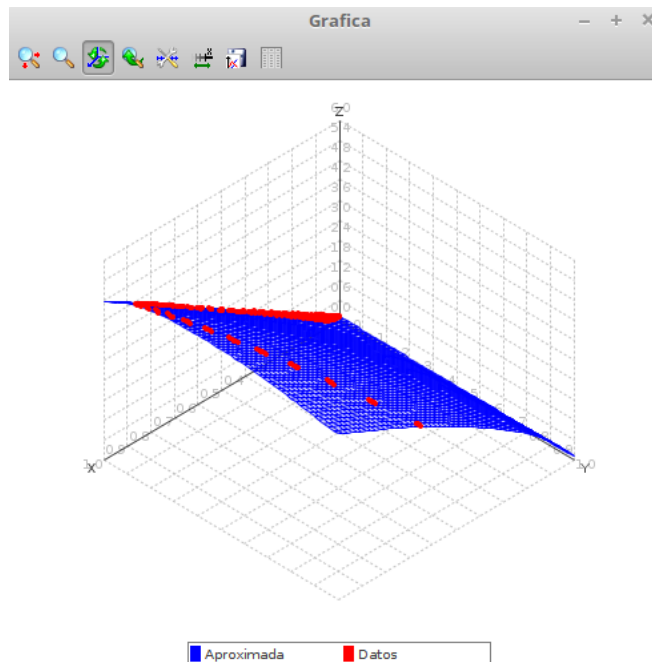


Figura 7. Ejemplo de una superficie de regresión.

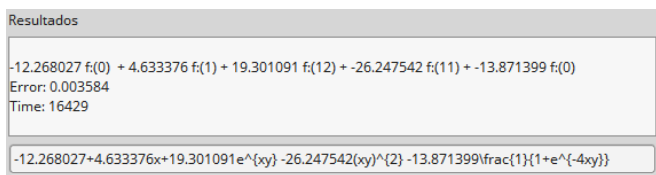


Figura 8. Resultados de la ejecución del algoritmo.

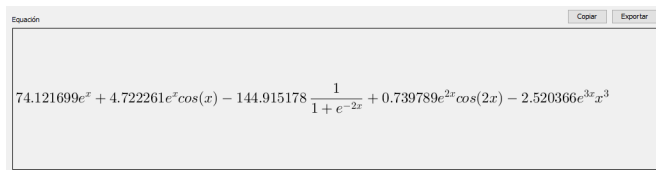


Figura 9. Ecuación obtenida del software.

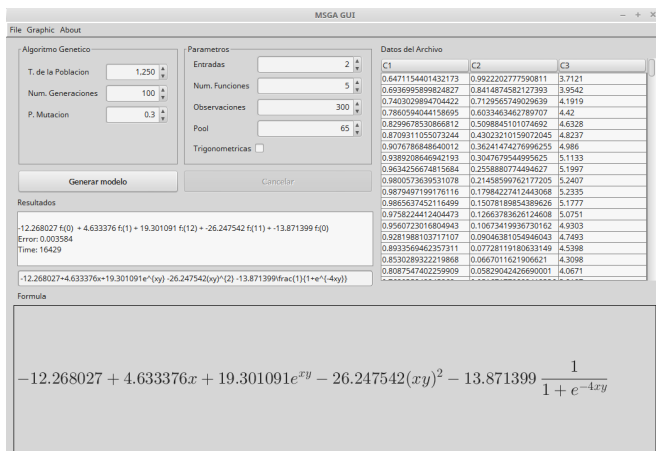


Figura 10. Pantalla completa del software.

VII. RESULTADOS Y TRABAJO FUTURO

La implementación desarrollada se probó en una computadora Aspire Acer modelo F5, con un procesador Intel Core i5-6200, 2.30 GHz y 8 GiB de RAM, bajo el sistema operativo Linux Mint 17 con dos conjuntos de datos. El primero con 490 registros del comportamiento de un bioreactor, en el cual se calcula la biomasa en futura a partir de la biomasa actual y el sustrato [7], es decir, un problema con dos variables independientes.

El segundo conjunto de datos, consta de 1,562 registros acerca del comportamiento de un levitador neumático presentado en [8], con tres variables independientes: posición anterior, velocidad anterior y una variable de control relacionada con la velocidad del dispositivo que produce la levitación, la variable de salida es la posición siguiente.

En el caso del modelo del bioreactor, se consiguió un ajuste considerablemente bueno en un tiempo razonablemente pequeño, como se puede apreciar en el Cuadro 1, utilizando $N = 130$, $GM_{ax} = 100$, $PL = 100$ y $PM = 0,3$. Al incrementar n el tiempo de ejecución es mayor pero la magnitud del error disminuye como era de esperar.

Modelo	Suma de errores al cuadrado	Tiempo
MSAG $n = 3$	0.1524	1.924 s
MSAG $n = 5$	0.0242	3.546 s
MSAG $n = 7$	0.0006	5.108 s

Cuadro I

Modelo	Suma de errores al cuadrado	Tiempo
MSAG $n = 3$	76.715	7.777 s
MSAG $n = 5$	69.124	18.662 s
MSAG $n = 7$	17.640	27.705 s

Cuadro II

En el caso del modelo del levitador neumático, utilizando los mismos parámetros, los resultados se pueden observar en el Cuadro 2. Se aprecia que el ajuste no es demasiado bueno en términos del error al cuadrado, sin embargo, este archivo de datos se construyó concatenando los resultados de tres diferentes pruebas en el levitador; cada una con diferentes objetivos de control.

Es bastante interesante notar que si se utiliza cada conjunto por separado es posible conseguir un error de 0 utilizando $n = 3$ para el primer set, con la función $w(x, y, z) = 2,380952 - 4,05e^{xy} - 99,7642(xy z)^2$. Para los dos conjuntos restantes el mismo comportamiento se consigue con $n = 5$, por lo que un modelo con una función definida a pedazos parece una mejor idea.

Otra causa del error cuadrado tan alto, además de los diferentes objetivos de control, es que los registros incluyen cálculos de velocidad por diferencias finitas y estimaciones de posición mediante visión artificial. Además el conjunto de datos no ha sido preprocesado y podría contar con la presencia de datos anómalos u *outlayers*.

A pesar de todas las limitaciones mencionadas, un modelo producido con este algoritmo se utilizó en [8] para llevar a cabo simulaciones del prototipo levitador que permitieron el ajuste adecuado de los parámetros utilizados en el cálculo de la entrada de control, lo que se conoce como sintonía de controladores.

Queda como trabajo futuro, comparar el modelo producido por esta estrategia contra modelos más sofisticados, así como probar el desempeño del mismo con parámetros distintos o con otras estrategias de cruce, selección o mutación.

REFERENCIAS

- [1] D. G. Zill. *Ecuaciones diferenciales con aplicaciones de modelado*. Ed. por Cengage Learning Editores. 9.^a ed. 2009.
- [2] D. Livignstone, D. Manallack y I Tetko. "Data modelling with neural networks: Advantages and limitations". En: *Journal of Computer-Aided Molecular Design* (1997), págs. 135-142. DOI: <https://doi.org/10.1023/A:1008074223811>.

- [3] J. H. Holland. *Adaptation in Natural and Artificial Systems*. Ed. por Michigan: MIT Press. 6.^a ed. 2001. 205 págs.
- [4] C. A. Coello Coello. *Introducción a la Computación Evolutiva*. Notas de curso. México, D.F.: CINVESTAV-IPN Departamento de Computación, mayo de 2013.
- [5] J. Koljonen y J. T. Alander. *Effects of population size and relative elitism on optimization speed and reliability of genetic algorithms*. In *Proceedings of the ninth Scandinavian conference on artificial intelligence*. Reporte de conferencia. Oct. de 2006, págs. 54-60.
- [6] Peña J. *Notas del curso de Métodos Numéricos*. Centro de Investigación en Matemáticas, 2013.
- [7] M. A. Castro Liera. “Un algoritmo Genético Distribuido con aplicación en la identificación difusa de un proceso fermentativo”. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas. Santa Clara, Cuba: Universidad Central “Marta Abreu” de las Villas, 2009.
- [8] Geraldo Sánchez O. P. “Diseño, construcción y control de un sistema de levitación neumática”. Tesis presentada en opción al grado de Ing. Electromecánico. La Paz, Baja California Sur: Instituto Tecnológico de La Paz, jun. de 2019.

Diseño de una Red Neuronal en FPGA para el Modelado de Sistemas Mecatrónicos.

René Cornejo León, Joel Artemio Morales Viscaya, Israel Santillán, Marco Antonio Castro Liera
Tecnológico Nacional de México, Instituto Tecnológico de La Paz, Blvd.Forjadores, La Paz, B.C.S., 23080, México.
cornejo_rene@hotmail.com, iscviscaya@gmail.com, israel.sm@lapaz.tecnm.mx, marco.cl@lapaz.tecnm.mx

Abstract—Este trabajo consiste en el diseño de una red neuronal para el modelado de sistemas dinámicos, en un FPGA. Para el diseño se utilizó la herramienta Simulink de Matlab así como Vivado mediante el software de System Generator. La red diseñada cuenta con una capa de entrada con dos neuronas (una para el estado del sistema previo al que se desea predecir y otra para la entrada externa o de control), una capa oculta de tamaño variable y 16 bits de punto fijo con signo (complemento a dos) para los buses de entrada. Dado que se trata de un problema de regresión, la capa de salida consta de una sola neurona, se utilizó como función de aprendizaje *backpropagation* (retropropagación) cuya implementación en un FPGA no resulta trivial. Se validó la red diseñada con un set de datos generados a partir de la solución de una ecuación diferencial.

I. INTRODUCCIÓN

Los sistemas mecatrónicos se representan mediante modelos matemáticos que no pueden ser exactos y generalmente son no lineales, lo que dificulta la definición y puesta a punto de la estrategia de control. Una alternativa son los métodos heurísticos, como el que aquí se presenta, que se configuran (entrenan) fuera de línea debido a la exigencia temporal y demanda de poder de cálculo de esta etapa. Aquí presentamos, una implementación en FPGA que es un sistema de gran poder de cómputo que puede ser parte del sistema mecatrónico de pequeño tamaño y poco requerimiento energético que, además, puede ser parte de un sistema autónomo.

El modelado de sistemas consiste en crear una abstracción matemática de un fenómeno a partir de definir sus variables dependientes, independientes y la relación matemática entre estas, lo que se conoce como identificación de sistemas. Los métodos usados en la identificación dependerán del conocimiento previo del sistema [1], por lo que se pueden clasificar en 3 categorías: modelos de caja blanca, de caja gris y de caja negra.

Cuando se conoce por completo el sistema a modelar se considera de caja blanca, esta clase de modelos son posibles de construir mediante el conocimiento físico del sistema. Cuando se desconocen algunos parámetros del sistema se considera de caja gris, estos modelos pueden ser construidos con los conocimientos disponibles mediante un modelo físico del sistema agregando un coeficiente que permita el ajuste de la salida del modelo para que se adapte a los datos observados. Si no se tiene conocimiento del comportamiento físico del sistema, se considerará como modelo de caja negra por lo que la estructura del modelo debe ser flexible, para adaptar la

salida de esta a los datos de entrada y que permita simular el comportamiento del sistema modelado[2].

Las redes neuronales artificiales (RNA) son un modelo matemático basado en el comportamiento de las neuronas biológicas, el cual busca generar la respuesta deseada a las entradas del sistema, mediante una función de aprendizaje[3]. Aunque las redes neuronales proporcionan modelos que pueden ser más exactos que otras alternativas, cuando el tamaño del set es muy grande o se utilizan muchas variables, el proceso de entrenamiento es muy costoso computacionalmente. Debido a que el algoritmo de entrenamiento requiere calcular el ajuste adecuado a cada uno de los pesos, la cantidad de pesos es proporcional al número de neuronas ocultas y entradas, esto se repite por cada juego de datos en el set de datos el cual es recorrido hasta cumplir con cierto número de ciclos, por lo que es requerido un gran número de cálculos para el ajuste de pesos. La implementación en paralelo utilizando FPGA ha demostrado ser útil reduciendo el tiempo de ejecución de algoritmos paralelizables como lo es el entrenamiento de una RNA. Sin embargo, no existe una implementación general de estos algoritmos en la literatura[4], [5].

Una matriz de compuertas lógicas programables o FPGA por sus siglas en inglés, se integra por diversos componentes los principales son: Los bloques CLB o bloques lógicos configurables, las interconexiones programables y los bloques entrada/salida. Esto se puede apreciar en la figura 1.

Las FPGA surgieron de la necesidad de probar los diseños de circuitos integrados antes de ser producidos en silicio, sin embargo se han empezado a usar las FPGA en aplicaciones donde se requiera que el hardware sea dinámico, aunque ofrece una respuesta más lenta que los dispositivos ASIC (circuitos integrados de aplicación específica) estos pueden ser usados en lugar del CPU con el fin de obtener una respuesta más rápida [6].

En este trabajo se presenta el diseño e implementación de una red neuronal para el modelado de sistemas, en un FPGA.

La implementación permite variar la representación en bits de los valores de entrada, el número de entradas, la cantidad de neuronas ocultas, el factor de aprendizaje y el número de épocas; consta de una sola capa oculta, la capa de salida es de una neurona, la función de aprendizaje que se utilizó es *backpropagation* y las funciones de activación implementadas son RELU e identidad, por lo que solo será útil en problemas

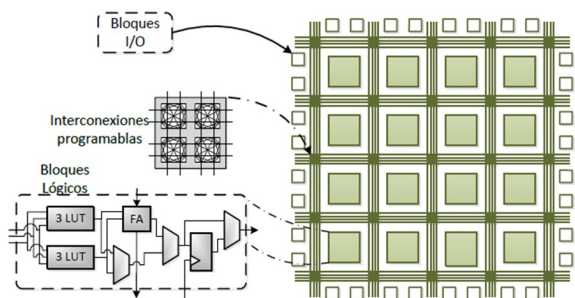


Fig. 1. Representación de la estructura básica de un FPGA[6].

de regresión.

II. ESTRUCTURA DE UNA RED NEURONAL

Las redes neuronales artificiales o RNA son un modelo computacional basado en el comportamiento de las neuronas biológicas, mediante un conjunto de perceptrones interconectados, se procesan las entradas de la red para producir la salida del sistema.

Como se puede apreciar en la figura 2, la red neuronal se compone por capas, la capa de entrada que representa las entradas del sistema, las capas ocultas que permiten abstraer características de las entradas y la capa de salida, cuya dimensión depende del número de salidas [7], [8].

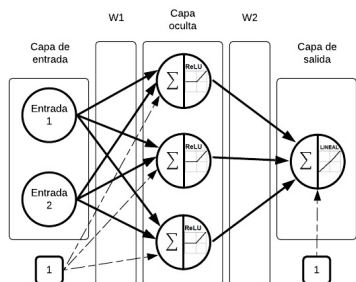


Fig. 2. Estructura básica de una RNA.

Como fue mencionado anteriormente, las RNA se componen por perceptrones, el cual es un modelo matemático que permite discriminar linealmente al vector de datos de entrada. Como se puede apreciar en la figura 3, se asigna un peso a cada una de sus interconexiones, lo que permite determinar qué tanto efecto tiene esa conexión en su respuesta o si tiene un efecto inhibitorio. A la sumatoria de los productos de cada peso por las entradas se le agrega un peso independiente conocido como Bias, el cual permite completar la ecuación del discriminador lineal que produce el perceptrón[7].

Para determinar el efecto de la salida de cada perceptrón en la siguiente capa, lo que se conoce como su activación, es necesario incluir una función (función de activación), la cual

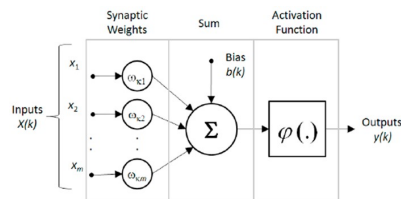


Fig. 3. Estructura de un perceptrón[8].

genera un umbral que el perceptrón tiene que sobrepasar para estar activo o inhibido, lo que a su vez permite introducir un comportamiento no lineal. De esta manera la red neuronal es capaz de resolver problemas no lineales, en la figura 4 se describen algunos ejemplos de funciones de activación, estas deben tener un rango de valores de salida finito y de preferencia ser derivables en todos sus puntos[3]. El algoritmo para producir la salida final de la red, se conoce como feedforward (hacia adelante).

	Función	Rango	Gráfica
Lineal	$y = x$	$\{-\infty, \infty\}$	
Escalon	$y = \text{sign}(x)$	$\{-1, 1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -1 \\ x, & \text{si } -1 \leq x \leq 1 \\ 1, & \text{si } x > 1 \end{cases}$	$\{-1, 1\}$	
ReLU	$y = \begin{cases} x, & \text{si } x \geq 0 \\ 0, & \text{si } x < 0 \end{cases}$	$\{0, 1\}$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$	$\{-1, 1\}$	

Fig. 4. Funciones de activación.

Para que las RNA funcionen correctamente es necesario encontrar los pesos adecuados para el problema a resolver, esto se logra por medio de algoritmos de aprendizaje, los cuales buscan minimizar una función de pérdida. Uno de los algoritmos más utilizados es el de backpropagation o retropropagación, el cual propaga el error hacia atrás desde la capa final para imputar el error asociado a cada neurona y emplea descenso del gradiente que mediante la derivada de la función, determina la dirección en la cual el error disminuye. Agregando un factor de aprendizaje que determina el incremento por ciclo de aprendizaje, se hace un cambio gradual de los pasos, ajustando la salida de la red a un set de entrenamiento[3], [9].

III. DISEÑO DE RED NEURONAL EN FPGA

Se decidió diseñar la red neuronal con una capa oculta y, ya que está pensado en el problema de modelar sistemas

dinámicos invariantes en el tiempo, dos entradas (el estado actual y la entrada de control). Una cantidad de neuronas de salida variable y 16 bits para el manejo de datos.

La red neuronal diseñada consta de los siguientes pasos:

A. Inicialización.

Al iniciar la red neuronal, el microcontrolador envía el set de entrenamiento y pesos generados de manera aleatoria al FPGA.

B. Descarga de pesos.

Los pesos recibidos por el FPGA, serán enviados a los bloques de feedforward y backforward para que estos sean grabados en su memoria individual.

C. Feedforward

En esta etapa se reciben los valores de entrada del set para procesarlos junto a los pesos recibidos y producir la salida de la red neuronal.

D. Backforward

Mediante los datos del set, se procesa la salida generada por el bloque de feedforward para obtener el error de la red, con el objetivo de ajustar los pesos a los valores de entrenamiento.

E. Escritura de pesos.

Al finalizar backforward, los pesos generados se guardan en un bloque de memoria, para su posterior envío a los bloques de feedforward y backforward.

F. Fin de ejecución

Por cada ciclo, se incrementa un contador, cuando este coincida con el tamaño del set, se envían los pesos generados al microcontrolador, para que este los respalde en un archivo binario.

IV. DISEÑO DE FEEDFORWARD EN FPGA

Para diseñar el feedforward, que se encarga de calcular la salida de la red neuronal, se utilizaron los bloques de entrada, producto punto, ReLU(rectified linear unit), ponderador, sumatoria, pesos, salida de neuronas y control. La interconexión de dichos bloque se lleva a cabo como se aprecia en la figura 5.

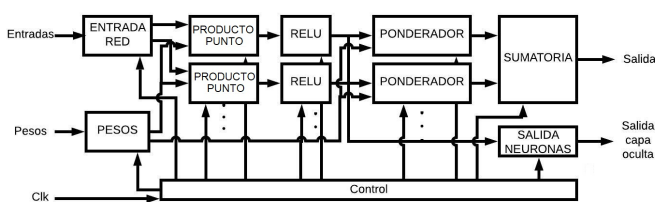


Fig. 5. Diagrama de bloques de feedforward.

El bloque de entrada recibe las entradas de la red y las envía secuencialmente al bloque ponderador, el bloque ponderador multiplica las valores enviados por el bloque de entrada por el peso correspondiente y acumula la suma de

sus resultados, el bloque ReLU envía el resultado de aplicar la función ReLU, el bloque ponderador multiplica su entrada por el peso correspondiente a la capa de salida, el bloque de salida suma las salidas de todos los bloques ponderador y el Bias, lo que da como resultado la salida de la red.

El bloque de salidas de neuronas recibe las salidas de los bloques ReLU para poder ejecutar el bloque backpropagation el cual necesita las salidas de la capa oculta, el bloque pesos el cual guarda los pesos asociados a cada conexión de la red neuronal. Este envía en grupos asociados a la misma capa y misma entrada a los bloques producto punto y ponderador. El bloque de control sincroniza la activación de cada bloque según la operación ejecutada.

A. Bloque entrada

Su función consiste en multiplexar las entradas de la red y enviarlas al bloque de producto punto como se muestra en la Tabla 1, en caso de calcular el Bias envía 1. En la figura 7 se muestra el diagrama digital del bloque.

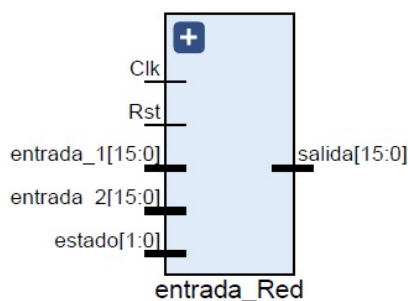


Fig. 6. Bloque de entrada.

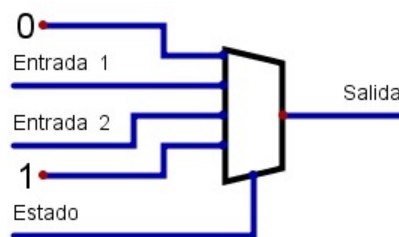


Fig. 7. Diagrama de bloque entrada.

estado	Salida
00	entrada_1
01	entrada_2
11	1
others	0

TABLE I

TABLA DE VERDAD DEL BLOQUE ENTRADA.

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.
- Entrada 1 y 2: Recibe la entrada de la red.

- Estado: Indica la entrada que sera enviada a la salida, en caso de ser "11" enviara el valor mas alto posible en complemento a dos.
- Salida: Envía la salida seleccionada en la entrada de estado.

B. Bloque producto punto

Envía el acumulado del producto de entrada enviada por el bloque de entrada y el peso asociado a la entrada. En las ecuaciones 1 y 2 se muestra la salida generada, donde entrada y peso son las entradas del bloque en 16 bits, el producto se guarda en 32 bits de los cuales se usan los 16 bits más significativos, la memoria es un registro en 18 bits que guarda el resultado de multiplicar la entrada y la salida, este valor se mantiene mientras no se inicialice el bloque, el bloque envía el valor guardado en la memoria en cada ciclo de reloj. En la figura 9 se muestra el diagrama digital del bloque.

$$Memoria = (Entrada * peso) + Memoria \quad (1)$$

$$Salida = Memoria \quad (2)$$

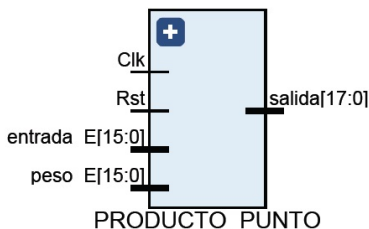


Fig. 8. Bloque de producto punto.

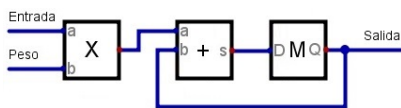


Fig. 9. Diagrama de bloque producto punto.

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.
- Entrada: Recibe la salida del bloque entrada.
- Peso: Recibe la salida del bloque de P memoria.
- Salida: Envía el valor acumulado.

C. Bloque función de activación

Realiza la función de activación ReLU, este bloque se considera la salida de la capa oculta. Se envía la entrada del bloque a la salida si esta es mayor a 0, en caso contrario se envía 0, se maneja una entrada de 18 bits, en donde se usan 2 bits para valor entero y 16 para el valor fraccional, cuando el valor es mayor a 1 se envía 1 en 16 bits. En la figura 11 se muestra el diagrama digital del bloque.

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.

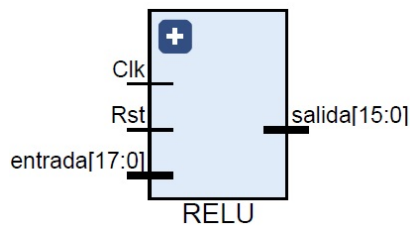


Fig. 10. Bloque de entrada.

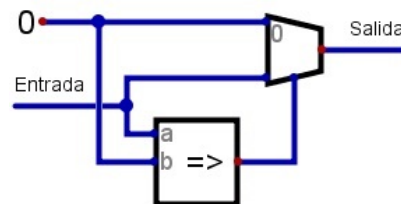


Fig. 11. Diagrama de bloque ReLU.

- Entrada: Recibe la salida del bloque producto punto.
- Salida: Envía el valor de entrada en caso de que sea mayor a 0, en caso de ser menor envía 0.

D. Bloque ponderador

Multiplica la salida del bloque de función de activación, por el peso asociado a la capa de salida, como se muestra en la ecuación 3. Las entradas se representan en 16 bits, el resultado se maneja en 32 bits donde se envían los 16 bits más significantes. En la figura 13 se muestra el diagrama digital del bloque.

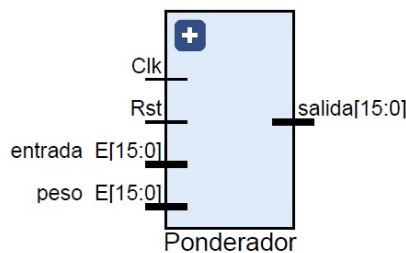


Fig. 12. Bloque de ponderador.

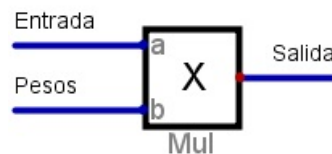


Fig. 13. Diagrama de bloque ponderador.

$$Salida = (Entrada * peso) \quad (3)$$

entrada	Salida
entrada > 0	entrada
entrada < 0	0

TABLE II
TABLA DE VERDAD DEL BLOQUE RELU.

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.
- Entrada: Recibe la salida del bloque RELU.
- Peso: Recibe la salida del bloque de pesos.
- Salida: Envía el valor acumulado.

E. Bloque sumatoria

Este bloque entrega la suma de todas las salidas del bloque ponderador, como resultado se obtiene la salida de la red neuronal. La suma se efectúa en un registro de 20 bits, 4 bits de valor entero y 16 fraccional, si el valor es mayor a 1 se envía 1 en 16 bits. En la figura 15 se muestra el diagrama digital del bloque.

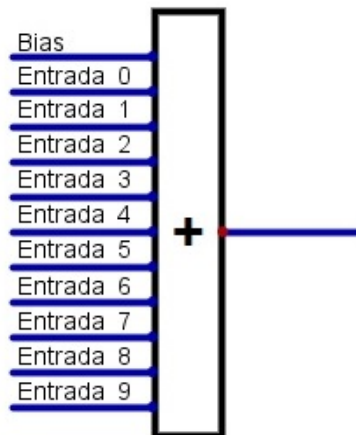


Fig. 15. Diagrama de bloque sumador.

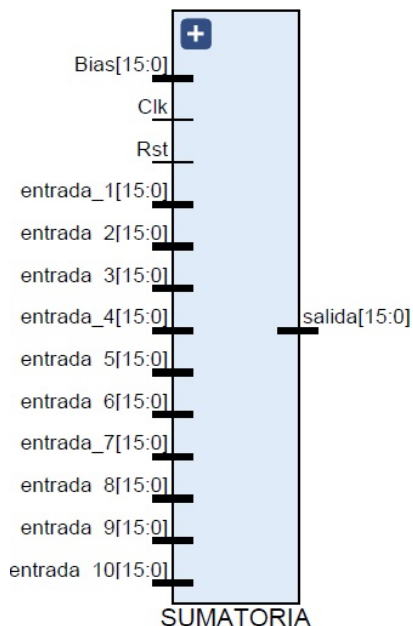


Fig. 14. Bloque de sumatoria.

$$salida = \sum Entradas \tag{4}$$

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.
- Bias: Recibe la salida del bloque entrada.
- Entradas 0 a 9: Recibe las salidas de todos los bloques de ponderador.
- Salida: La suma de todas las salidas de los bloques ponderador.

F. Bloque pesos

Este bloque guarda los pesos de la red neuronal, según la capa a la que pertenezca y la entrada asociada. Como se puede

apreciar en la figura 17, el envío de pesos marcado con la línea azul va junto a un valor que indica la posición del peso en la RNA. Cada peso de la red esta relacionado con una dirección de seis bits, dos donde se determina la capa y entrada asociada y cuatro donde se determina la posición en esa capa. Este bloque puede estar en dos estados, escritura y lectura. Si se encuentra en estado de escritura, el valor de entrada será guardado en la dirección establecida, de lo contrario, si se encuentra en estado de lectura, enviará el valor guardado.

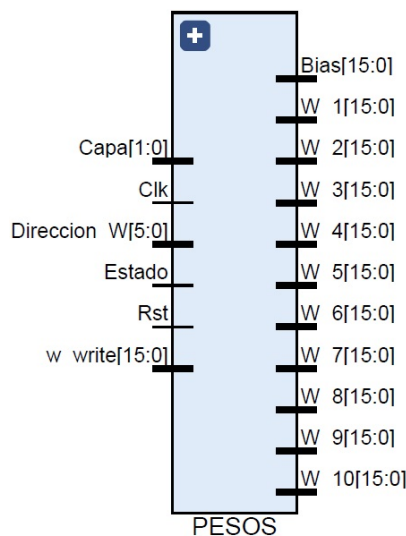


Fig. 16. Bloque P_MEMORIA.

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.
- Dirección_R: Recibe la posición del peso que debe enviar.
- Dirección_W: Recibe la posición del peso que debe escribir.
- Estado: Si es 1 indica que debe escribir el valor de la entrada w_write en la posición indicada en Direccion_W.

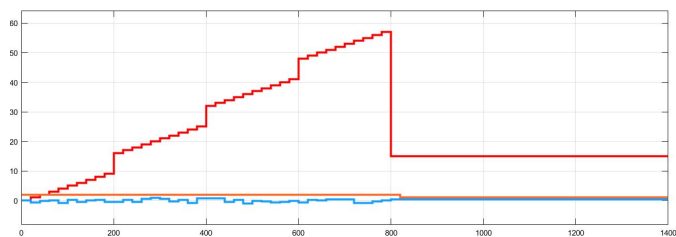


Fig. 17. Envío de pesos y ubicaciones al bloque pesos.

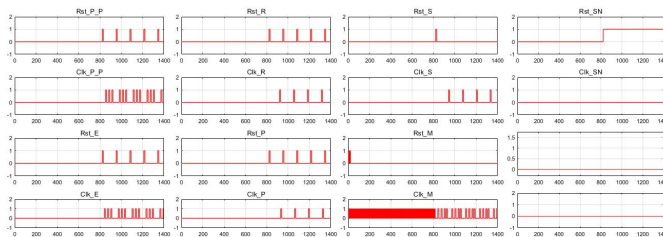


Fig. 19. Salidas del bloque control.

- W_write: Recibe el peso que debe escribir.
- W0 a 9: Envía el peso indicado en Dirección_R.

G. Bloque control

La función de este bloque es la de coordinar todos los bloques que compone la red, enviando pulsos en sus entradas de reloj. En la figura 19 se muestra la secuencia de activación de las salidas del bloque, en sus primeros 41 ciclos de reloj se realiza el envío de pesos, después se efectúa feedforward.

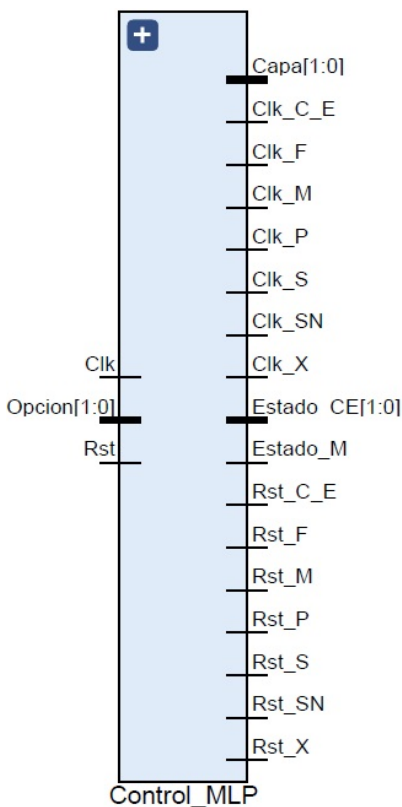


Fig. 18. Bloque de control.

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.
- Opción: Permite elegir la operación que el bloque debe ejecutar, 00 para feedforward, 01 para escritura de pesos y 10 envía las salidas de neuronas ocultas.

- Capa: Indica los pesos que debe enviar el bloque p_memoria.
- Clk_C_E: Señal de sincronía del bloque entrada_red.
- Rst_C_E: Inicialización de salida del bloque entrada_red.
- Clk_F: Señal de sincronía del bloque ReLU.
- Rst_F: Inicialización de salida del bloque ReLU.
- Clk_M: Señal de sincronía del bloque p_memoria.
- Rst_M: Inicialización de salida del bloque p_memoria.
- Clk_P: Señal de sincronía del bloque producto_punto.
- Rst_P: Inicialización de salida del bloque producto_punto.
- Clk_S: Señal de sincronía del bloque sumatoria.
- Rst_S: Inicialización de salida del bloque sumatoria.
- Clk_SN: Señal de sincronía del bloque salida_neuronas.
- Rst_SN: Inicialización de salida del bloque salida_neuronas.
- Clk_X: Señal de sincronía del bloque ponderador.
- Rst_X: Inicialización de salida del bloque ponderador.
- Estado_CE: Controla la entrada enviada por el bloque entrada_red.
- Estado_M: Controla la escritura de pesos del bloque p_memoria.

V. DISEÑO DE BACKFORWARD EN FPGA

Este bloque ejecuta la función de aprendizaje de backpropagation, con el fin de ajustar los pesos de la red a los valores de entrenamiento, se utilizaron los bloques error cuadrático, delta, pesos, p memoria y control. La interconexión de dichos bloques se lleva a cabo como se aprecia en la figura 20.

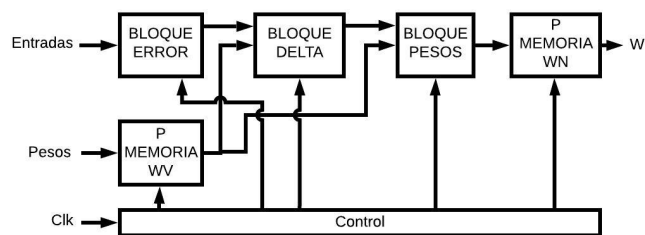


Fig. 20. Diseño de Backforward.

El bloque de error cuadrático calcula el error asociado a la salida de la red, el bloque delta calcula el error imputado a cada neurona, el bloque pesos actualiza los pesos pasados, el

bloque p memoria se utiliza para almacenar los pesos actuales y guardar los pesos actualizados, el bloque de control sincroniza la activación de todos los bloques según la operación ejecutada.

A. Bloque error cuadrático

Envía el error asociado a la salida de la red, el bloque le resta al valor esperado el valor obtenido y lo eleva al cuadrado para después dividirlo entre 2. En la figura 22 se muestra el diagrama digital del bloque.

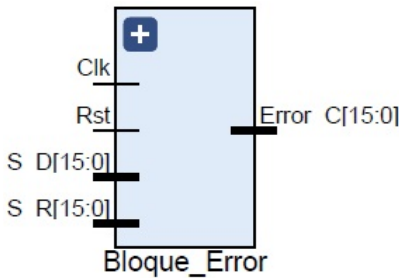


Fig. 21. Bloque de error.

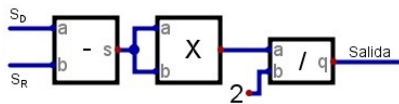


Fig. 22. Diagrama de bloque error cuadrático.

$$\frac{(SD - SR)^2}{2} \tag{5}$$

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.
- S_D: Recibe la salida ideal de la red.
- S_R: Recibe las salidas generada por el bloque feedforward.
- Error: Mediante la ecuación 5 se calcula el error de la red neuronal.

B. Bloque delta

Determina el valor delta correspondiente al peso actualizado, para obtener el valor delta primero se evalúa la entrada Selec, a la que se envía 0 si el peso pertenece a la capa oculta y 1 si pertenece a la capa de salida. Envía el valor en delta_s, en caso de que Selec sea 0 se evalúa si h es mayor a 0, si es mayor a 0 se envía el peso multiplicado por la entrada delta_s en caso contrario se envía 0. En la figura 24 se muestra el diagrama digital del bloque.

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.
- Delta_S: Recibe la salida del bloque Error.
- Selec: Si Selec es 0 y h es mayor a 0 envía el valor en la entrada DELTA_S, si Selec es 1 y h es mayor a 0 envía

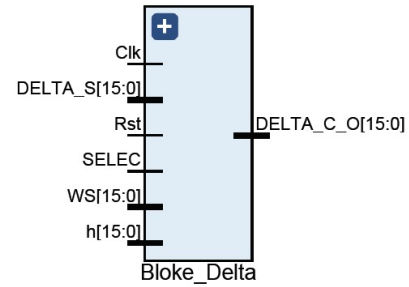


Fig. 23. Bloque de valor delta.

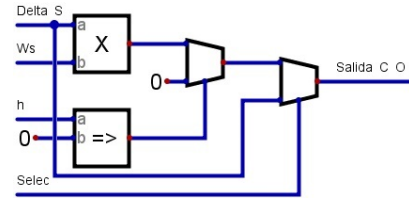


Fig. 24. Diagrama de bloque delta.

el valor en la entrada DELTA_S*WS, en caso de que h sea menor a 0 en ambos casos envía 0.

- WS: Recibe la salida del bloque p memoria.
- h: Envía el valor de entrada en caso de que sea mayor a 0, en caso de ser menor envía 0.
- DELTA_C_O: Envía el valor de entrada en caso de que sea mayor a 0, en caso de ser menor envía 0.

C. Bloque pesos

Calcula el valor del peso actualizado multiplicando las entradas delta, E y a, al resultado se suma Wa. En la figura 26 se muestra el diagrama digital del bloque.

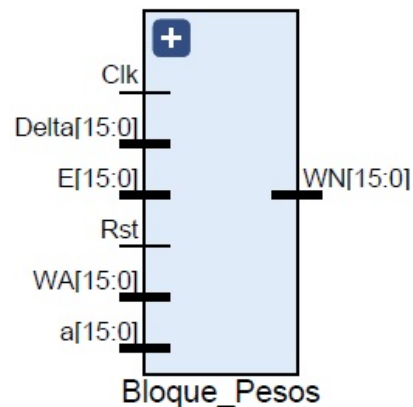


Fig. 25. Bloque de pesos.

$$Wa + a * E * Delta \tag{6}$$

- Clk: Señal de sincronía.
- Rst: Inicialización de salida de bloque y registros.

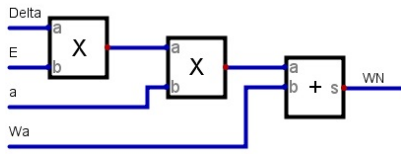


Fig. 26. Diagrama de bloque pesos.

- W_a = Peso que esta siendo modificado.
- a = Factor de aprendizaje.
- E = Entrada asociada al peso.
- Δ = Error imputado al peso multiplicado por la derivada de la función de activación de la neurona.
- WN = Envía el peso actualizado.

D. Bloque P memoria

Este bloque guarda los pesos de la red neuronal, según la capa a la que pertenezca y la entrada asociada. Este bloque puede estar en dos estados, escritura y lectura. Si se encuentra en estado de escritura, el valor de entrada será guardado en la dirección establecida, de lo contrario, si se encuentra en estado de lectura enviará el valor guardado. Cada peso de la red está relacionado con una dirección de seis bits, dos donde se determina la capa y entrada asociada y cuatro donde se determina la posición en esa capa.

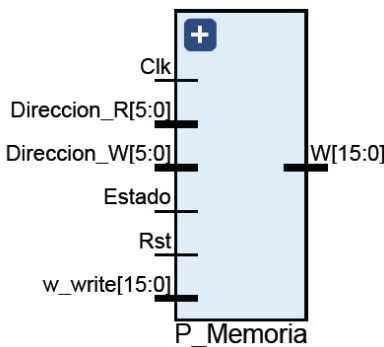


Fig. 27. Bloque P_MEMORIA.

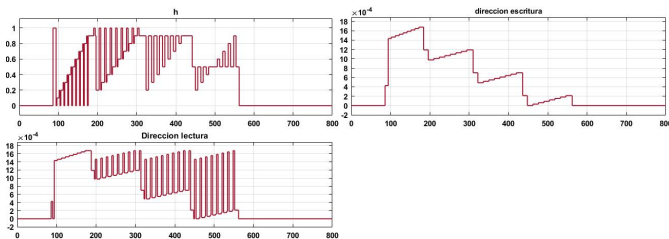


Fig. 28. Envío de direcciones de memoria y entradas a la red.

- Clk : Señal de sincronía.
- Rst : Inicialización de salida de bloque y registros.
- $Dirección_R$: Recibe la posición del peso que debe enviar.

- $Dirección_W$: Recibe la posición del peso que debe escribir.
- $Estado$: Si es 1 indica que debe escribir el valor de la entrada w_write en la posición indicada en $Dirección_W$.
- w_write : Recibe el peso que debe escribir.
- $salida$: Envía el peso indicado en $Dirección_R$.

E. Bloque control

La función de este bloque es coordinar la activación del resto de bloques mediante pulsos en sus entradas de reloj e inicialización. Este bloque ejecuta tres etapas: etapa de recepción de pesos actuales, backforward y envío de pesos actualizados.



Fig. 29. Bloque de control.

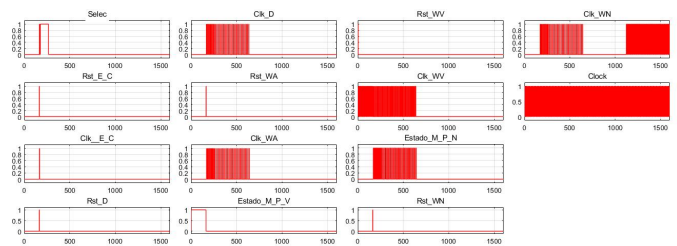


Fig. 30. Salidas del bloque de control.

- Clk : Señal de sincronía.
- Rst : Inicialización de salida de bloque y registros.
- $Opción$: Permite elegir la operación que el bloque debe ejecutar, 00 para feedforward, 01 para escritura de pesos y 10 envía las salidas de neuronas ocultas.
- S_N_O 0 al 9: Recibe las salidas de las neuronas ocultas.
- $Entrada1$ y 2 : Recibe las entradas a la red.

- Capa: Indica los pesos que debe enviar el bloque p_memoria.
- Clk_D: Señal de sincronía del bloque entrada_red.
- Rst_D: Inicialización de salida del bloque entrada_red.
- Clk_E_C: Señal de sincronía del bloque ReLU.
- Rst_E_C: Inicialización de salida del bloque ReLU.
- Clk_WA: Señal de sincronía del bloque p_memoria.
- Rst_WA: Inicialización de salida del bloque p_memoria.
- Clk_WN: Señal de sincronía del bloque producto_punto.
- Rst_WN: Inicialización de salida del bloque producto_punto.
- Clk_WV: Señal de sincronía del bloque sumatoria.
- Rst_WV: Inicialización de salida del bloque sumatoria.
- Selec: indica al bloque delta si el peso pertenece a la capa de salida.
- h: Envía la entrada necesaria para calcular delta.
- Clk_X: Señal de sincronía del bloque ponderador.
- Rst_X: Inicialización de salida del bloque ponderador.
- Estado_M_P_V: Controla la entrada enviada por el bloque entrada_red.
- Estado_M_P_N: Controla la escritura de pesos del bloque p_memoria.
- Estado_R: Controla la escritura de pesos del bloque p_memoria de los pesos viejos.
- Estado_W: Controla la escritura de pesos del bloque p_memoria de los pesos nuevos.

VI. RESULTADOS Y CONCLUSIONES

El diseño del código en VHDL fue emulado usando la herramienta de Matlab Simulink, para poder emular el código en VHDL se utilizó Vivado para compilar el código y System Generator para vincular Vivado a Simulink, el código fue emulado usando la opción de black box de Simulink como se aprecia en la figura 31.

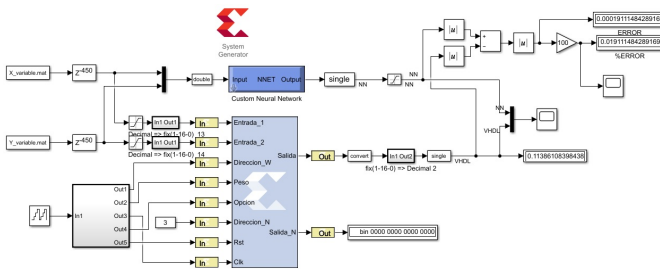


Fig. 31. Diagrama de simulink.

Para poder probar el funcionamiento de la red era necesario generar un juego de pesos que pudiera usar la red neuronal, por lo que se entrenó una red neuronal en Matlab con un set de datos generados a partir de un sistema simulado mediante ecuaciones diferenciales, los pesos generados se convirtieron a formato de 16 bits, y se guardaron en un bloque de subsistema que envía los pesos junto a su ubicación en la RNA, y el pulso de reloj del bloque.

Para poder obtener la precisión de la RNA en VHDL se emuló una red neuronal con la misma arquitectura y pesos

con un set de prueba, como se puede apreciar en la figura 32. Se comparó la respuesta de la RNA en Matlab y en VHDL, se obtuvo la diferencia en las respuestas y se graficó el error como se muestra en la figura 33, el error se mantiene cercano a 0 mientras no haya una variación súbita en la respuesta. En los picos donde llega a tener un error del 60% de una corta duración, el error promedio fue del 2.18%, esto puede deberse a la representación en un formato de 16 bits en complemento a dos, la resolución máxima en los números es de 0.000030518.

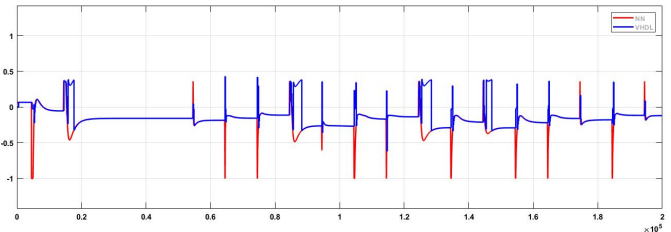


Fig. 32. Comparación de la respuesta de la RNA en CPU y VHDL.

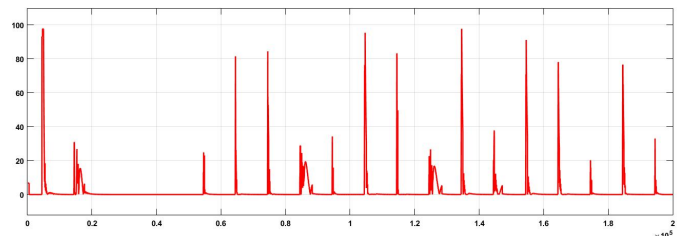


Fig. 33. Error de la RNA en VHDL.

Se implementaron los bloques en el software de Xilinx Vivado, se escogió la tarjeta Zybo z7-10 para la implementación, como resultado se obtuvo que el bloque feedforward requiere de 2,853 LUT, 1,383 FF, 20 DSP y 5 BUFG, que representan el 14% LUT, 4% FF, 25% DSP y 16% BUFG. Para el bloque backforward requiere de 1,056 LUT, 1,385 FF, 4 DSP y 4 BUFG, que representan el 6% LUT, 4% FF, 5% DSP y BUFG 13%. Como se puede observar, la implementación del algoritmo en el FPGA consume pocos recursos de este, lo que permitiría incrementar el número de neuronas en la capa oculta para resolver problemas más complejos o disponer de recursos para otros propósitos.

VII. TRABAJO FUTURO

Se planea llevar a cabo la implementación del código desarrollado en un FPGA de la marca Digilent modelo Zibo z7-10. Se obtendrán datos experimentales de un péndulo invertido sobre ruedas, con el fin de emular el comportamiento de este mediante la red implementada en FPGA y comparar el modelo obtenido con otros en la literatura.

REFERENCES

[1] I.Rivals and L.Personnaz, "Black-box modeling with state-space neural networks." *World scientific*, pp. 2-7, 1995.

- [2] J.Sjöberg, Q. Zhang, L.Ljung, and A.Benveniste, "Nonlinear black-box modeling in system identification: a unified overview," *Pergamon*, pp. 1–2, 1995.
- [3] L.Federico, R.García, and P.Britos, "Entrenamiento de redes neuronales basado en algoritmos evolutivos." *Universidad de buenos aires*, pp. 15–19, 2005.
- [4] J.Parra, D.Ramos, and A.Tigreros, "Implementación de redes neuronales utilizando dispositivos lógicos programables," *Vision de caso*, pp. 2–8.
- [5] P.Dondon, J.Carvalho, R.Gardere, P.Lahalle, G.Tsenov, and V.Mladenov, "Implementation of a feed-forward artificial neural network in vhdl on fpga." *Neurel*, pp. 1–3, 2014.
- [6] D.Francisco, "Algoritmos de aprendizaje neurocomputacionales para su implementación hardware." *Universidad de Málaga*, pp. 37–42, 2015.
- [7] M. Cabezón, J. E.Rosa, and F.Liberatore, "Predicción demanda eléctrica española. implementación de redes neuronales recurrentes en python," *Universidad Complutense de Madrid*, pp. 5–21, 2018.
- [8] H.Hultmann, D.Habineza, M.Rakotonbrade, C.Klein, and L.Coelho, "Nonlinear black-box sistem identification though neuronal networwks of a hysteretic piezoelectric robotic micromanipulator," *Elsevier*, pp. 1–6, 2015.
- [9] Y.Maeda and M.Wakamura, "Simultaneous perturbation learning rule for recurrent neuronal networks and its fpga implementation." *IEEE*, vol. 16, pp. 4–6, 2005.

Diseño de un Perfilador Hidrográfico Autónomo Costero

Josué Hiram Armenta Romero*, Marcos Israel Santillán Méndez*,
Leonardo Tenorio Fernández[‡] y Jesús Alberto Sandoval Galarza*

*División de Estudios de Posgrado e Investigación

Tecnológico Nacional de México / Instituto Tecnológico de La Paz

La Paz, Baja California Sur, México

Email: jarmenta8@hotmail.com, israel.sm@lapaz.tecnm.mx, jesus.sg@lapaz.tecnm.mx

[‡]Departamento de Oceanología

Instituto Politécnico Nacional / Centro Interdisciplinario de Ciencias Marinas / CONACyT

La Paz, Baja California Sur, México

Email: leonardo.tenof@gmail.com

Resumen—En este trabajo se presenta el diseño de un perfilador hidrográfico autónomo costero, capaz de descender y ascender de manera autónoma en un rango de profundidad de cero a cincuenta metros. Este flotador permitirá apoyar tareas de investigación del Departamento de Oceanología del Instituto Politécnico Nacional en el Centro Interdisciplinario de Ciencias Marinas, mediante la recopilación de datos hidrográficos de temperatura, salinidad y presión. La contribución del trabajo es una propuesta de perfilador hidrográfico autónomo diseñado específicamente para usarse en zonas costeras de bajas profundidades (hasta 50 metros). La propuesta incluye el diseño mecánico, la instrumentación requerida y el análisis de su modelo matemático para determinar sus parámetros.

Palabras Claves—Sistema, Móvil, Instrumentación, Oceanografía, Densidad, Temperatura, Salinidad, Presión, Flotabilidad, AUV, ROV, Cambio de Volumen, FPGA.

I. INTRODUCCIÓN

Desde hace siete décadas los instrumentos de flotabilidad neutra son utilizados de distintas formas, con el propósito de explorar y descubrir ciertos aspectos de la circulación de los océanos. Existen distintos tipos de dispositivos submarinos, los vehículos controlados remotamente (ROVs) y los vehículos submarinos autónomos (AUVs). El control y operación de ambos vehículos se basa en la patente No W02005019021 [1], la cual establece el control de boyancia para equipos submarinos.

El primer instrumento para estudiar las corrientes marinas fue creado por el oceanógrafo británico John Swallow en 1955, el cual estaba constituido por un tubo de aluminio, una batería y un circuito temporizador. En la Figura 1 podemos observar al oceanógrafo junto al flotador en el suelo. El equipo era capaz de permanecer estable a una profundidad dada debido a su compresibilidad menor que el agua salada, y a la rigidez del cuerpo. Este estudio se basó en el Principio de Arquímedes [2], el cual afirma que todo cuerpo sumergido en un fluido experimenta una fuerza vertical y hacia arriba igual al peso de la ausencia de fluido. A ésta fuerza vertical se la llama fuerza boyante o fuerza de flotabilidad.

Por otro lado el origen de los perfiladores hidrográficos autónomos Argo surge en los años 1990-1997 con un experimento denominado WOCE por sus siglas en inglés World Ocean Circulation Experiment; el cual pretendía recopilar un conjunto de observaciones sin precedentes. Dicho experimento requería recolectar datos de las corrientes oceánicas a mil metros de profundidad. Para lograrlo Russ Davis miembro de la Scripps Institution of Oceanography, en California, USA, junto a Doug Webb de Webb Research Corporation crearon un explorador autónomo de circulación lagrangiana (ALACE) en el año 1991. [3]

Éstos instrumentos usaban el principio de flotabilidad neutra, y periódicamente regresaban a la superficie para que su posición fuera determinada por un satélite. Fueron lanzados cerca de mil flotadores como parte del experimento y hoy en día los datos obtenidos pueden ser consultados en internet. En la actualidad Argo es un red global de perfiladores el cual colecta datos de alta fidelidad de temperatura y salinidad desde la superficie del mar hasta dos mil metros de profundidad en los océanos libres de hielo y corrientes de profundidades intermedias.

Los datos provienen de los perfiladores hidrográficos autónomos ubicados alrededor del mundo, y dichos flotadores realizan misiones en intervalos de nueve días, puesto que permanecen estables a una profundidad de mil metros obteniendo una densidad igual a la del ambiente en el que se encuentran. Posteriormente descienden hasta los dos mil metros para iniciar con los perfiles de temperatura y salinidad; resaltando el hecho que las lecturas se toman mientras el perfilador asciende a la superficie. Mientras transcurre el ascenso, el perfilador almacena la serie de mediciones de temperatura, salinidad y presión en su tarjeta de adquisición de datos alojada en el interior del perfilador.

Una vez se encuentre en la superficie los datos son transmitidos a los satélites que ofrezcan sus servicios bajo contrato. Es importante mencionar que se recolectan series de alrededor de doscientas mediciones de las variables de temperatura y salinidad. Por último, los datos son transmitidos de los satélites

a centros de recolección de datos para someterlos a un proceso de control de calidad, éste proceso puede durar desde un día hasta un año [4]. En la Tabla I se enlistan los símbolos utilizados en el presente trabajo.II.



Figura 1. Flotador creado por el oceanógrafo británico John C. Swallow (1955).

Tabla I
LISTA DE SÍMBOLOS

Símbolo	Descripción	Unidad
F_b	Fuerza Boyante	N
ρ	Densidad	$\frac{kg}{m^3}$
S	Salinidad	PSU
T	Temperatura	$^{\circ}C$
γ	Peso específico	$\frac{N}{m^3}$
V	Volumen	m^3
m	Masa	kg
p	Presión	bar

II. PARÁMETROS DE DISEÑO

Para lograr que un vehículo submarino sea capaz de descender y ascender de manera autónoma se requiere conocer su masa y su volumen. Posteriormente se modifica alguno de estos dos parámetros y en consecuencia se logra un cambio de densidad en él mismo, permitiendo que ésta sea distinta a la del agua de mar.

Para obtener la masa y el volumen, es necesario conocer en primer lugar la fuerza boyante que se ejercerá sobre el equipo una vez se encuentre completamente sumergido y flotando. A continuación se muestra la ecuación para calcular la fuerza de flotabilidad.

$$F_b = \gamma \cdot V \quad (1)$$

donde cada una de las variables está definida en la Tabla I, y $\gamma = \rho g$ siendo $g = 9.8$ [m/s^2].

II-A. Ecuación internacional de estado del agua del mar de 1980 (UNESCO)

La densidad es la cantidad de masa por unidad de volumen en una sustancia. En el agua de mar ésta aumenta con respecto a la disminución de la temperatura y al incremento de salinidad, mientras que disminuye debido al aumento de temperatura y al decremento de salinidad. [5]. “Se ha determinado una ecuación de estado del agua de mar diluida con agua pura o concentrada en evaporación, para ser utilizada con la escala de salinidades prácticas. Esta ecuación es más exacta que las ecuaciones utilizadas en la actualidad (Knudsen, Forch y Sørensen, 1902; Ekman, 1908; Cox, MacCartney y Culkin, 1970)” [6]. Dicha ecuación es válida para los intervalos de $S = 0$ a 42 PSU; $T = -2$ a 40 $^{\circ}C$; $p = 0$ a 1,000 bares. La densidad del agua de mar en función de la salinidad práctica y la temperatura está dada por:

$$\rho(S, T) = a + bs + cs^{\frac{3}{2}} + ds^2 \quad (2)$$

donde:

$$a = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5$$

$$b = b_0 + b_1t + b_2t^2 + b_3t^3 + b_4t^4$$

$$c = c_0 + c_1t + c_2t^2$$

$$d = d_0$$

y a su vez los coeficientes de los polinomios son las siguientes constantes:

$$a_0 = 999.8425$$

$$a_1 = 6.793952^{-2}$$

$$a_2 = -9.095290^{-3}$$

$$a_3 = 1.001685^{-4},$$

$$a_4 = -1.120083^{-6}$$

$$a_5 = 6.536332^{-9},$$

$$b_0 = 8.24493^{-1}$$

$$b_1 = -4.0899^{-3},$$

$$b_2 = 7.6438^{-5}$$

$$b_3 = -8.2467^{-7},$$

$$b_4 = 5.3875^{-9}$$

$$c_0 = 5.72466^{-3},$$

$$c_1 = 1.0227^{-4}$$

$$c_2 = -1.6546^{-6},$$

$$d_0 = 4.8314^{-4}$$

Estos valores constantes han sido propuestos por la UNESCO [3].

II-B. Cálculo de densidad del agua de mar

Para determinar la densidad del agua de mar en una columna de agua se utiliza el caso de estudio *Cuenca Alfonso*, el cual consiste en la toma de muestras de salinidad, temperatura y presión en un rango de profundidad de cero a cincuenta metros registrada mensualmente durante un año, información extraída de un instrumento oceanográfico no autónomo.

En la Figura 2 se puede apreciar cómo varía la temperatura en el mes de Junio del 2018, mientras en la Figura 3 se muestra un ejemplo de un perfil de salinidad creado a partir de los mismos datos del mes de Julio.

Para calcular la densidad, se insertaron los datos del estudio *Cuenca Alfonso* en (2), y para mayor facilidad se utilizó el software MATLAB para su cálculo numérico y visualización

gráfica. Esto con el fin de obtener la densidad durante el periodo de Mayo de 2018 a Septiembre de 2019.

En la Tabla II se enlistan las distintas densidades calculadas con MATLAB durante el periodo mencionado y en el rango de cero a cincuenta metros de profundidad. De la información obtenida, se registra como valor mínimo el 01 de Octubre de 2018 y como valor máximo el 19 de Junio de 2018, los cuales han sido resaltados en amarillo para una rápida y fácil identificación.

Por su parte en la Figura 4 se puede observar gráficamente la variación de la densidad durante el periodo seleccionado.

Tabla II
DENSIDAD DEL AGUA DE MAR DE LA CUENCA ALFONSO DURANTE EL PERIODO DE MAYO DE 2018 A SEPTIEMBRE DE 2019.

Fecha	Superficie ($\frac{kg}{m^3}$)	Fondo (50 metros)($\frac{kg}{m^3}$)
22 Mayo 2018	1,023.87	1,025.72
19 Junio 2018	1,023.45	1,025.75
25 Julio 2018	1,022.64.	1,025.08
21 Agosto 2018	1,021.98	1,024.85
01 Octubre 2018	1,021.75	1,024.28
08 Noviembre 2018	1,022.53	1,023.95
30 Enero 2019	1,024.67	1,024.73
29 Marzo 2019	1,024.28	1,025.53
30 Abril 2019	1,024.11	1,025.69
30 Mayo 2019	1,023.97	1,025.40
25 Junio 2019	1,023.75	1,025.71
31 Julio 2019	1,022.54	1,025.20
30 Agosto 2019	1,022.13	1,024.43
25 Septiembre 2019	1,022.11	1,024.44

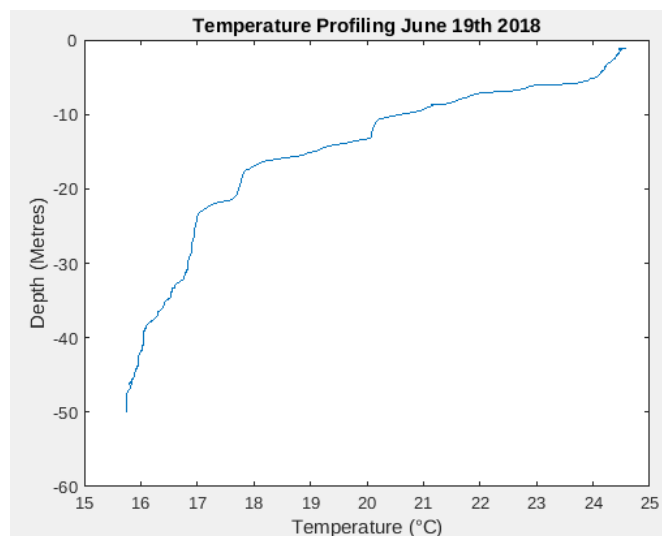


Figura 2. Perfil de temperatura del 19 de Junio de 2018 .

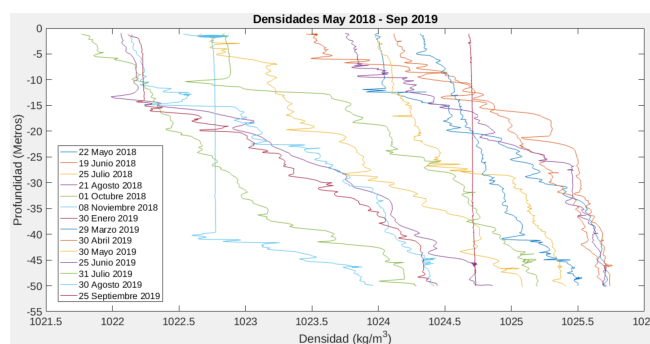


Figura 4. Cambio de densidades de Mayo de 2018 a Septiembre de 2019.

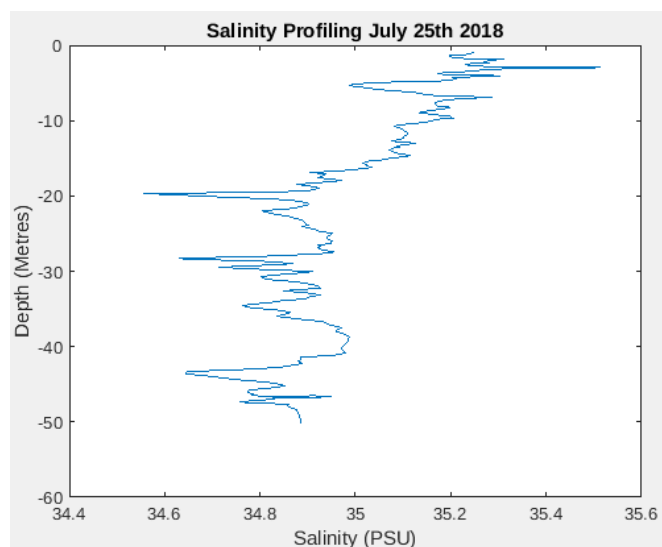


Figura 3. Perfil de salinidad del 25 de Julio de 2018 .

Para el cálculo del volumen del equipo se toma en cuenta un modelo vigente y comercial de un flotador perfilador hidrográfico con forma cilíndrica (ver Figura 5). En la se muestra el modelo.

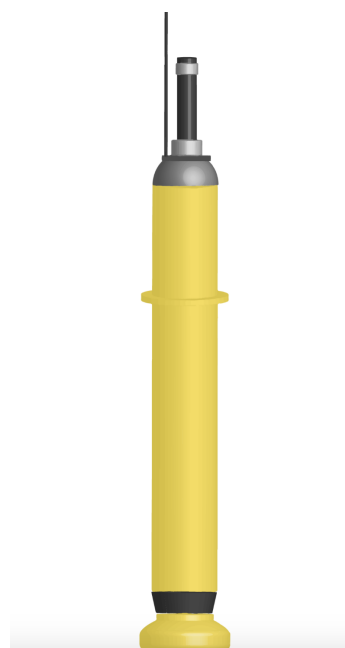


Figura 5. Perfilador hidrográfico comercial de la marca Argo.

Considerando una altura de 39 centímetros y un diámetro de 10.16 cm (4 pulgadas) se obtiene el volumen de la siguiente manera:

$$\begin{aligned} V &= \pi r^2 h \\ &= \pi(5.08)^2(39) \\ &= 3.1619 * 10^{-3} m^3 \end{aligned}$$

Enseguida, se procede a resolver (1) y determinar la fuerza de flotabilidad que se ejercerá en nuestro vehículo tanto en la superficie así como a una profundidad de cincuenta metros. De (1), en superficie la fuerza boyante está dada por:

$$F_b = (1,021.75 \cdot 9.8)(3.1619 * 10^{-3}) = 31.660N$$

mientras que la fuerza boyante a una profundidad de 50 metros es la siguiente:

$$F_b = (1,025.75 \cdot 9.8)(3.1619 * 10^{-3}) = 31.784N$$

Por otro lado, conociendo la F_b es posible determinar la masa que se requerirá para que el instrumento permanezca completamente sumergido y flotando a la profundidad deseada para esto se utiliza (3).

$$m = \frac{F_b}{g} \quad (3)$$

Resolviendo (3) con los valores obtenidos la masa en la superficie es:

$$m = \frac{31.660}{9.8} = 3.2307Kg$$

Mientras que para 50 metros de profundidad:

$$m = \frac{31.784}{9.8} = 3.2433Kg$$

Con esto tenemos todos los parámetros necesarios para poder diseñar el instrumento deseado.

II-C. Cambio de Densidad

En cuanto al cambio de densidad en el instrumento, es requerido que exista una variación ya sea en su masa o en su volumen. En la Tabla III se muestran las dos propuestas para modificar la boyancia del equipo, y de esta manera lograr el ascenso y descenso del mismo de 0 a 50 metros de profundidad.

Tabla III
PROPUESTAS CAMBIO DE DENSIDAD

Masa - Superficie	Masa - 50 metros	Diferencia
3,230.7 gr	3,243.3 gr	12.6 gr
Altura - Superficie	Altura - 50 metros	Diferencia
390 mm	391.5 mm	1.5 mm

III. DISEÑO Y CONSTRUCCIÓN

Para el diseño del instrumento propuesto se ha seleccionado un cuerpo cilíndrico, por la simetría geométrica, con las dimensiones mínimas requeridas para poder alojar a los sensores responsable de las mediciones y recolección de información de los parámetros oceanográficos de interés, y también se incluye un disco estabilizador para asegurarse que el vehículo sea

capaz de ascender y descender de manera que logre mantener su vertical en todo momento, y una membrana para efectuar su cambio de densidad. En la Figura 6 se muestra un simple esquema del vehículo propuesto.

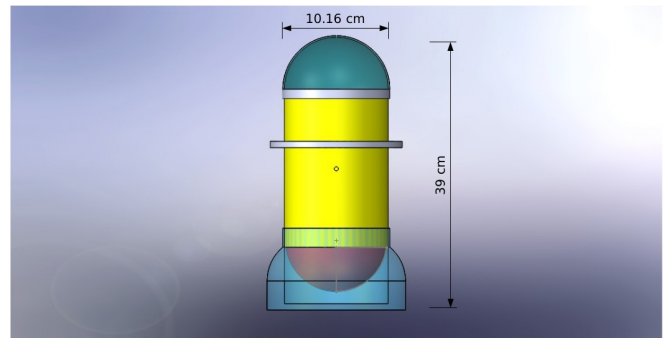


Figura 6. Diseño flotador perfilador propuesto.

Con relación a la elaboración del prototipo se propone utilizar como material principal el policloruro de vinilo cédulo 40 el cual se fabrica bajo la norma nacional NMX-E-145/1. Se decide utilizar este producto debido a que el fabricante recomienda para su operación una temperatura máxima de 60 °C y una presión máxima de 220 PSI para un diámetro nominal de 4 pulgadas. Éste material será utilizado para el cuerpo del equipo, la creación del disco estabilizador, y diversas piezas que se utilizarán en su construcción. Para el cambio de densidad se utilizará un motor diferencial que modifique la altura del dispositivo final con la intención de variar el volumen y por tanto lograr el ascenso o descenso según sea requerido. Además, la membrana con la que se efectuará esta acción será de hule de polímero de cloropreno debido a que sus propiedades logran que no se distorsione a la flexión, pero conservando una buena flexibilidad, así como su resistencia a un ambiente hostil, por ejemplo, las inclemencias del clima. En la Figura 7 se muestra la estructura interna que tendrá el equipo así como la tarjeta de adquisición de datos que comunica con los sensores y sistema de potencia.

Por otra parte en la Figura 8 se muestra la membrana flexible constituida por una brida de pvc y una capa de neopreno para asegurar un sellado perfecto.

Con respecto al prototipo final, éste es mostrado en la Figura 9.

III-A. Procesamiento de datos

Para desarrollar los perfiles de temperatura y salinidad se utilizará el poder de cálculo que nos ofrece el procesamiento en paralelo de un arreglo de compuertas lógicas programables en campo (FPGA). Los flotadores existentes en el mercado utilizan procesamiento secuencial, por ende no son capaces de realizar la adquisición de datos de más de una variable oceanográfica a la vez, por lo que requieren de otras técnicas como la interpolación. Con el modelo propuesto se obtendrán mediante el procesamiento paralelo, de ésta manera es posible asegurarse que el dato medido corresponde a la profundidad en la que se encuentra el equipo.



Figura 7. Estructura interna.



Figura 8. Membrana flexible de neopreno.

Un filtro es una herramienta (implementada en hardware o software) que se aplica a un conjunto de datos ruidosos para poder extraer información sobre un número de muestras de interés. En el área de las señales, el filtrado es un proceso mediante el cual se modifica el contenido espectral de una señal. Un filtro digital es un sistema lineal e invariante en el tiempo (LTI) se clasifican como: FIR (finite impulse response) que se caracterizan por ser sistemas no recursivos, e IIR (infinite impulse response) que se distinguen por tener retroalimentación en la señal de salida.

Los parámetros de diseño del filtro propuesto, el cual será pasa bajas, con una frecuencia de corte de 100 Hertz, una banda de transición de 300 Hz y una frecuencia de muestreo de 2000 Hz. Se sabe que un filtro FIR de orden “M” es posible describirlo mediante:

$$y(n) = b_0x(n) + b_1x(n^{-1}) + b_2x(n^{-2}) + \dots + b_Mx(n^{-M}) \quad (4)$$

donde la secuencia b_k representa el vector de coeficientes del filtro. [7]

Se diseñó un filtro FIR con la ayuda del software MATLAB y se verificó su desempeño con una tarjeta de adquisición de datos. Para esto último se procedió a enviar los datos por comunicación serial para posteriormente procesarlos según convenga. Para lograr un correcto diseño del filtro y que

además sea robusto, se agrega ruido blanco para corroborar si efectivamente se eliminan las frecuencias no deseadas. En la Figura 10 se observa en color rojo la señal original mientras que los datos con ruido blanco en color azul y por último los filtrados en color verde. Para concluir el diseño, se implementa dicho filtro dentro del hardware reconfigurable. Para este fin, es necesario convertir el vector de coeficientes a punto fijo, para esto los valores son elevados a la décimo quinta potencia y se redondean. Enseguida, se convierten a base 2 para poder ser manejados en el lenguaje de descripción de hardware. En la Tabla IV es posible observar el funcionamiento del filtro creado en hardware. En esta se observan los datos provenientes de la señal original, los datos con ruido blanco y la señal filtrada.

Tabla IV
FIR PASA BAJAS PARA PERFIL DE TEMPERATURA

Señal original	Señal con ruido	Señal filtrada
26.75	26.55	26.83
27.00	26.41	27.08
27.19	26.50	27.31
27.37	26.55	27.53
27.56	27.18	27.68
27.69	26.27	27.89

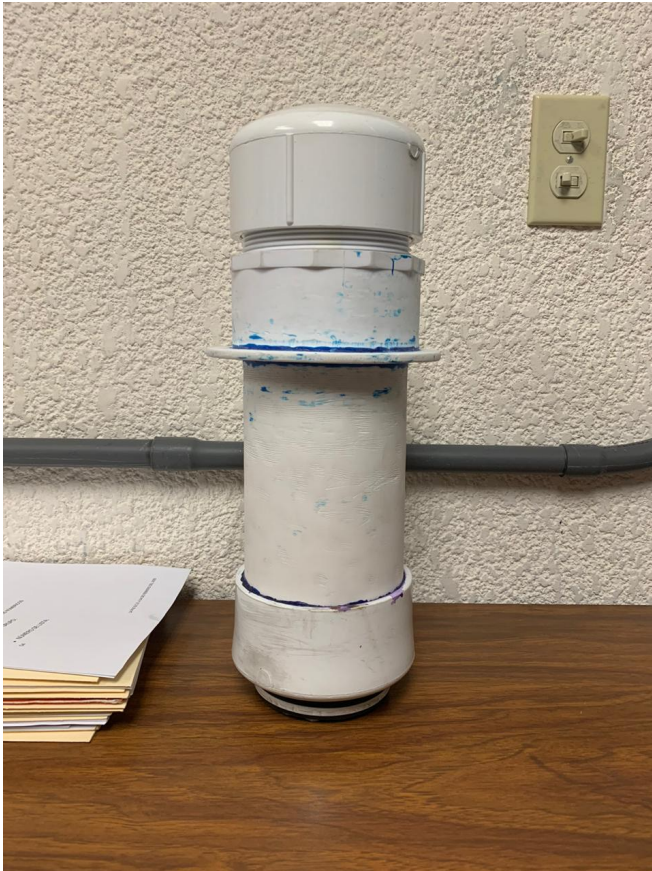


Figura 9. Prototipo de flotador perfilador propuesto.

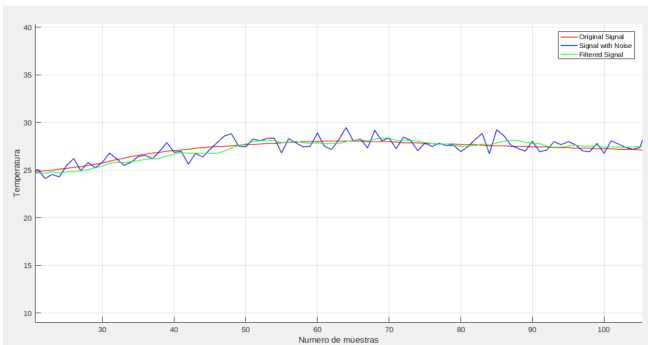


Figura 10. Datos de temperatura crudos, ruidosos y filtrados..

IV. CONCLUSIONES

Este documento presentó el diseño preliminar de un perfilador hidrográfico autónomo costero. Para su diseño se utilizó el caso de estudio Cuenca Alfonso, del cual se extrajeron los datos de temperatura, salinidad y profundidad para obtener perfiles de densidad reales de una zona en la cual pudiera operar el instrumento.

Con estos datos se analizó el perfil de densidad en la columna de agua de esta zona. Identificando sus valores mínimo y máximo que requiere el instrumento para funcionar

durante todo el año.

Fueron consideradas las dos estrategias para lograr el cambio de densidad en el equipo y se decidió por la estrategia del cambio de la altura del instrumento. Es importante mencionar que por autonomía se entiende la operación del vehículo sin la intervención de una persona para su ascenso y descenso a una profundidad específica.

REFERENCIAS

- [1] A. Jamieson, "Autonomous lander technology for biological research at mid-water, abyssal and hadal depths", Ch 5, PhD Thesis, University of Aberdeen, Scotland, 2004.
- [2] Bierman J, Kincaid E. Reconsidering Archimedes' principle. *The Physics Teacher*, Vol 41, September 2003, pp. 340-344.
- [3] Davis, R.E., D.C. Webb, L.A. Regier and J. Dufour, 1992: The Autonomous Lagrangian Circulation Explorer (ALACE). *J. Atmos. Ocean. Tech.*, 9, 264-285.
- [4] "These data were collected and made freely available by the International Argo Program and the national programs that contribute to it. (<http://www.argo.ucsd.edu>, <http://argo.jcommops.org>). The Argo Program is part of the Global Ocean Observing System."
- [5] J.Cifuentes, L. Torres-García, M Frías. *El océano y sus recursos, III : Las ciencias del mar : oceanografía física, matemáticas e ingeniería*. Marzo 2013, Capítulo 2. *Propiedades Físicas Del Agua De Mar*
- [6] 10° informe del Grupo Mixto de Expertos sobre Tablas y Patrones Oceanográficos: La Escala de salinidades prácticas de 1978 y la Ecuación internacional de estado del agua del mar de 1980.
- [7] Smith, Steven W. *Digital Signal Processing. A Practical Guide for Engineers and Scientist*. Amsterdam: Newnes, Elsevier Science. 2003. ISBN: 0-750674-44-X.

Implementación electrónica y análisis de la dinámica de un sistema regulador glucosa-insulina de orden fraccionario con retardos temporales

Berenice Fernández-Carreón*, Jesus Manuel Muñoz-Pacheco*, Luz del Carmen Gómez-Pavón*, Ernesto Zambrano-Serrano†, Andrés Anzo-Hernández‡

* Facultad de Ciencias de la Electrónica

Benemérita Universidad Autónoma de Puebla, Puebla, México

*Email: berenice.fernandez@alumno.buap.mx

†Facultad de Ingeniería Mecánica y Eléctrica

Universidad Autónoma de Nuevo León, Nuevo León, México

‡Facultad de Ciencias Físico Matemáticas

Benemérita Universidad Autónoma de Puebla, Puebla, México

Resumen—En este artículo se presenta el análisis dinámico de un sistema biológico que describe la interacción entre la glucosa e insulina. De esta manera es posible estudiar dos escenarios: libre de diabetes (dinámica regular) o padeciendo diabetes (dinámica caótica). Este modelo incorpora retardos temporales y las ecuaciones diferenciales que lo describen se basan en la teoría de cálculo de orden fraccionario. En consecuencia, el sistema de orden fraccionario presenta nuevos efectos que son de interés para su estudio y análisis, los cuales podrían ayudar a describir mejor la enfermedad de la diabetes mellitus. Para la validación del modelo, se ha realizado la implementación electrónica usando un procesador ARM. Los resultados experimentales concuerdan con la teoría relacionada demostrando la viabilidad del modelo propuesto.

Index Terms—orden fraccionario, *time-delay*, atractor caótico, glucosa, insulina, diabetes.

I. INTRODUCCIÓN

La diabetes técnicamente llamada diabetes mellitus, es un grupo de enfermedades caracterizadas por trastornos en los procesos metabólicos del cuerpo humano, en los que se interrumpe el mecanismo de control del nivel de azúcar en la sangre. En estos casos la insulina, el elemento principal de control no se secreta o las células del cuerpo ignoran su presencia [1]–[3].

Diversos procesos patológicos están involucrados en el desarrollo de diabetes mellitus, aunque la gran mayoría de los casos pueden incluirse en dos categorías. En la primera de ellas, diabetes mellitus tipo 1, donde la causa es una deficiencia absoluta en la secreción de insulina, a menudo con evidencia de destrucción autoinmune de las células pancreáticas. En la segunda categoría y mucho más prevalente esta la diabetes mellitus tipo 2, donde la causa es una combinación de resistencia a la acción de la insulina (generalmente asociada a obesidad) y una inadecuada respuesta secretora compensatoria [1]–[4].

Por un lado, la glucemia es la medida de concentración de glucosa libre en la sangre, un alto nivel de glucemia aparece cuando el organismo no cuenta con la suficiente cantidad de

insulina o cuando la cantidad de insulina es muy escasa, a este proceso derivado de la diabetes se le llama técnicamente hiperglucemia que se caracteriza por altos niveles de azúcar en la sangre. Por el otro lado, la hipoglucemia, también conocida como bajo nivel de glucosa o de azúcar en la sangre, ocurre cuando el nivel de glucosa en la sangre cae por debajo de lo normal. La hipoglucemia puede ser un efecto secundario de la insulina y otros tipos de medicinas para la diabetes que ayudan al cuerpo a crear más insulina.

A continuación se presentan algunos datos estadísticos sobre la población diagnosticada con esta enfermedad en México, donde datos recabados de la Encuesta Nacional de Salud (ENSA) 2000, la Encuesta Nacional de Salud y Nutrición (ENSANUT) 2006, 2012 y MC 2016, muestran que la prevalencia de diabetes por diagnóstico previo ha aumentado con una tendencia anual positiva de 2.7%. En 2016, la prevalencia de diabetes fue de 9.4% mayor respecto a la de 2012 y al menos en México hasta el 2016 había poco más de 6.4 millones de personas diagnosticadas con diabetes, cerca de 60 000 más que en 2012 [4]. Por lo tanto, se requiere seguir estudiando y comprendiendo la diabetes en la mayoría de los escenarios.

Los modelos matemáticos de sistemas biológicos son de gran importancia debido a que permiten analizar la interacción entre sus variables y así como manipular el modelo de tal forma que no se tenga que interactuar con el sistema real [5]–[8]. La elaboración de modelos matemáticos más cercanos a la realidad es de gran interés para la comunidad científica donde la meta es encontrar el que mejor describa el comportamiento del fenómeno bajo estudio. En los últimos años también se han desarrollado varios modelos que describen un comportamiento biológico los cuales suponen nuevas herramientas para combatir problemas médicos [9]–[12].

Una clase de modelos muy interesante son aquellos que incluyen retardos temporales (*time-delay*). En la naturaleza los retardos temporales realmente influyen en varios aspectos de nuestras vidas, por ejemplo, los reflejos humanos donde una

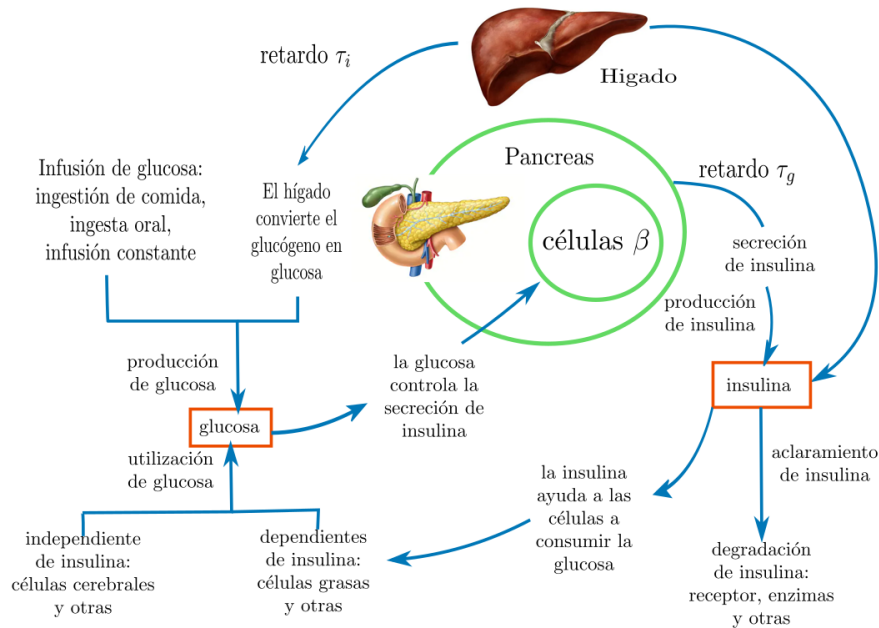


Figura 1: Sistema de control de insulina-glucosa estudiado por *Chuedoung et al.* [23].

pronta respuesta puede ser la diferencia entre recibir o no un golpe. Como el factor del tiempo existe y afecta a muchos sistemas la inclusión de estos a los modelos matemáticos resulta en una mejor manera de representar su comportamiento [13]–[15]. En la literatura podemos encontrar retardos temporales en modelos físicos, químicos, de ingeniería y en sistemas biológicos [2], [13].

Dentro de los modelos biológicos se ha demostrado que los modelos caóticos son más aproximados a los sistemas biológicos reales debido a su naturaleza caótica, ya que presentan comportamientos inesperados e impredecibles lo cuales se encuentran en muchos de los sistemas biológicos reales [7], [16]. Por lo anterior el estudio de modelos biológicos caóticos que incorporen retardos temporales es de gran importancia para el estudio de sistemas que describen trastornos o enfermedades en el ser humano ya que describen de mejor manera estos problemas de interés social. En los últimos años se han desarrollado modelos que describen varios padecimientos tales como el modelo propuesto por *S. Kajanchi et al* [13] el cual describe la presencia de cáncer mediante la interacción entre las células tumorales y el sistema inmune, también el propuesto por *Z. Mukandavire* [17] que describe el síndrome de inmunodeficiencia adquirida (SIDA), entre otros [18], [19].

Considerando la discusión anterior, un modelo biológico que describa la dinámica del sistema regulador de insulina-glucosa podría ayudar a hacer predicciones confiables y estudios hacia un tratamiento efectivo de la diabetes mellitus. Algunos trabajos anteriores de este tema son los descritos por *P Sadaghi et al.*, *Lenbury et al.*, *J. S. Bajaj et al.* y *Sarika et al.* [2], [20]–[22]. En el presente trabajo se analiza el modelo matemático de ecuaciones diferenciales con retardo propuesto

por *Chuedoung et al.* [23] que describe el comportamiento dinámico del sistema de retroalimentación de glucosa-insulina que involucra a las células β e incorpora dos retardos de tiempo uno para la generación de insulina y el otro para la generación de glucosa [2].

Se propone ampliar el modelo descrito por *Chuedoung et al.* a orden fraccionario debido a que la implementación del cálculo fraccionario, el cual es el estudio de derivadas e integrales de orden no entero, ha demostrado comportamientos nuevos en los sistemas que no son posibles de observar en sistemas de orden entero. [8], [24]–[29]. Por último, una vez descrito el modelo propuesto de orden fraccionario que incorpora retardos temporales se presenta una implementación de este usando una tarjeta ARM, para posteriormente poder visualizar sus correspondientes series de tiempo y fase en un osciloscopio.

El manuscrito se organiza como sigue. En la sección 2 se introduce el modelo regulador glucosa-insulina con retardos temporales. En la sección 3 se presenta el modelo propuesto considerando la teoría de cálculo de orden fraccionario. En la sección 4 se analiza la estabilidad del sistema regulador glucosa-insulina. En la sección 5 se implementa el sistema biológico en una tarjeta ARM y se discuten los resultados experimentales. Finalmente, en la sección 6 se dan las conclusiones.

II. MODELO MATEMÁTICO

El control primario de la secreción de insulina es un sistema de retroalimentación negativa dirigida entre las células pancreáticas y la concentración de glucosa en la sangre que fluye hacia ellas. Un nivel elevado de glucosa en sangre,

como durante la ingesta de una comida, estimula directamente las células para sintetizar y liberar insulina. El aumento del nivel de insulina, a su vez, reduce el nivel de glucosa en plasma a la normalidad, promoviendo el uso y almacenamiento de nutrientes. Por el contrario, una caída de la glucosa en sangre por debajo de lo normal, como durante el ayuno, inhibe directamente la secreción de insulina. La disminución de la tasa de secreción de insulina desplaza el metabolismo de la fase de absorción a la fase post-absorción. Por lo tanto, este simple sistema de retroalimentación negativa puede mantener un suministro relativamente constante de glucosa a los tejidos esencialmente sin requerir la participación de nervios u otras hormonas. El comportamiento del sistema se representa en la figura 1.

Además, el modelo matemático descrito por las ecuaciones diferenciales (1) representa el comportamiento dinámico del sistema de retroalimentación de glucosa-insulina que involucra a las células β , e incorpora dos retrasos de tiempo, uno siendo el retraso de producción de insulina desencadenado por la glucosa τ_g , y el otro es el retraso de glucosa hepática τ_i .

$$\begin{aligned} \frac{dx}{dt} &= r_1 y(t - \tau_g) z(t - \tau_g) - r_2 x + c_1 z(t - \tau_g), \\ \frac{dy}{dt} &= \frac{R_3 N}{z} - R_4 x(t - \tau_i) + C_2, \\ \frac{dz}{dt} &= R_5 (y - \hat{y})(T - z) + R_6 z(T - z) - R_7 z, \end{aligned} \quad (1)$$

donde $x(t)$ es la concentración de insulina, $y(t)$ es la concentración de glucosa, $z(t)$ es el número de células β y \hat{y} es la diferencia entre el nivel de glucosa en ayunas y su nivel basal, τ_g es el retardo en la secreción de insulina en respuesta al aumento del nivel de glucosa en sangre según la evidencia clínica informada por *Palumbo et al.* [30] y τ_i es el retardo en la caída de glucosa debido al aumento del nivel de insulina según la evidencia clínica informada por *Prager et al.* [31]. r_1, r_2, R_3-7, c_1 y C_2 son constantes de velocidad, N es el número normal de células y T es la densidad total de las células divisorias y no divisorias que se supone constante.

En un sujeto sano, existe un ajuste entre el retardo de tiempo τ_g en la producción de insulina en respuesta a la variación de glucosa en plasma, y el retardo correspondiente τ_i . Esto significa que si el retardo de la glucosa hepática τ_i aumenta, el sistema regulador de retroalimentación podrá mantener estabilidad al acortar el retardo de producción de insulina τ_g . Un mal funcionamiento en el ajuste de cualquiera de los dos retardos de tiempo puede conducir a una pérdida de estabilidad que podría resultar en que el nivel de glucosa en sangre se vuelva drásticamente alto (hiperglucemia) o extremadamente bajo (hipoglucemia). Según lo informado por *Bertram y Pernarowski* [32], típicamente se observa un retraso de 1 a 2 minutos en la actividad eléctrica de los islotes de Langerhans, encargados de la liberación de insulina.

Dado el valor de los parámetros de la tabla I y las condiciones iniciales $[x(0), y(0), z(0)] = [6.03, 1.79, 0.82]$ se obtuvieron los correspondientes diagramas de fase y tiempo de las figura 2 y 3 respectivamente ocupando los valores de $\tau_g = 0.56$ y $\tau_i = 0.05$. En ambas figuras se observa una

Tabla I: Tabla de parámetros para el sistema (1).

Parámetros	Valores
r_1	0.472
r_2	0.25
R_3	0.82
R_4	0.6
R_5	0.3
R_6	0.3
R_7	0.2
\hat{y}	1.42
N	1.27
T	1.5
c_1	0.1
C_2	0.8

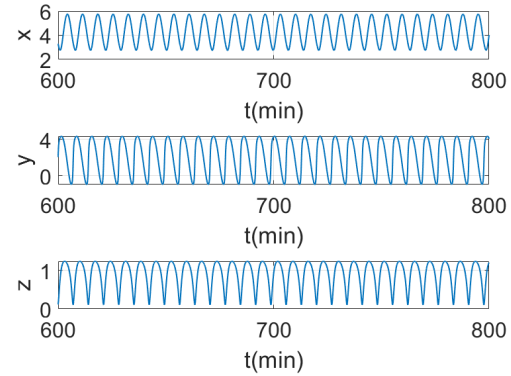


Figura 2: Series de tiempo del sistema de orden entero (1) con $\tau_g = 0.56$ y $\tau_i = 0.05$.

dinámica periódica (tiempo y fase) indicando que el sistema regulador glucosa-insulina funciona de manera normal.

III. MODELO DE ORDEN FRACCIONARIO PROPUESTO

Existen varias definiciones para el operador integrodiferencial de orden fraccionario entre las que destacan la de Grünwald-Letnikov, Riemann-Liouville y Caputo. Para el presente documento se ocupó un método de solución numérica derivado de la definición de Grünwald-Letnikov, debido a que ocupa un efecto de memoria, el cual se describe en la ecuación (2) [16], [33].

$${}_{(k-Lm/h)}D_{tk}^q f(t) \approx h^{-q} \sum_{j=0}^k (-1)^j \binom{q}{j} f(t_{k-j}). \quad (2)$$

Donde q es el orden fraccionario (no entero), Lm es el tamaño de memoria, $t_k = kh$, h es el tamaño de paso de integración y $(-1)^j \binom{q}{j}$ son los coeficientes binomiales $c_j^{(q)}$ ($j = 0, 1, \dots$). Los cuales se calculan con la siguiente expresión [16], [33]:

$$c_0^{(q)} = 1, \quad c_j^{(q)} = \left(1 - \frac{1+q}{j}\right) c_{j-1}^{(q)}. \quad (3)$$

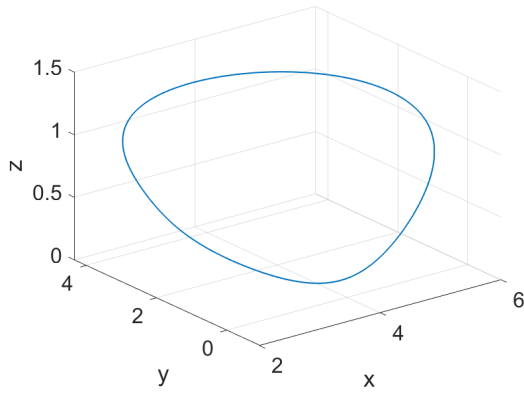


Figura 3: Diagrama de fase del sistema de orden entero (1) con $\tau_g = 0.56$ y $\tau_i = 0.05$.

Entonces la solución numérica general de la ecuación diferencial ${}_a D_t^q y(t) = f(y(t), t)$ puede ser expresada como

$$y(t_k) = f(y(t_k), t_k) h^q - \sum_{j=v}^k c_j^{(q)} y(t_k - j). \quad (4)$$

Para el término de memoria expresado en la suma, es usado un principio de memoria corta en el cual $v = 1$ para $k < (Lm/h)$ y $v = k - (Lm/h)$ para $k > (Lm/h)$, en caso de requerir memoria completa $v = 1$.

Como el presente caso trata un sistema de orden fraccionario con retardos, es necesario realizar consideraciones al modelo anterior. Lm debe ser múltiplo del retardo τ_g y τ_i , mientras que el valor de los retardos $\tau_g = h(t_{\tau_g})$ y $\tau_i = h(t_{\tau_i})$, donde t_{τ_g} y t_{τ_i} son el tamaño de memoria de τ_g y τ_i respectivamente y representan el número de pasos hacia atrás en la memoria [14], [26], [33].

Tomando en cuenta la definición anterior, el modelo de la ecuación (1) se representa en orden fraccionario de la siguiente forma:

$$\begin{aligned} D_t^q x(t) &= r_1 y(t - \tau_g) z(t - \tau_g) - r_2 x + c_1 z(t - \tau_g), \\ D_t^q y(t) &= \frac{R_3 N}{z} - R_4 x(t - \tau_i) + C_2, \\ D_t^q z(t) &= R_5 (y - \hat{y})(T - z) + R_6 z(T - z) - R_7 z. \end{aligned} \quad (5)$$

Del sistema de ecuaciones anterior se añade una variable al sistema original correspondiente al orden fraccionario q cuyo valor será propuesto a partir del análisis de estabilidad de la siguiente sección.

Dado que el modelo de ecuaciones (1) presenta un comportamiento periódico con $\tau_g = 0.56$ y $\tau_i = 0.05$ lo que se traduce en un buen funcionamiento del ajuste de glucosa-insulina, buscamos ahora un comportamiento inestable. Para esto se propone un nuevo valor para τ_g y τ_i en base a diagramas de bifurcación con $r = \sqrt{y(t_k)^2 + x(t_k)^2}$ que es la distancia por cada cruce de $z(t_k)$ a través de una superficie Poincaré y con $q = 1$ siendo el orden fraccionario [28].

Como se puede observar en la figura 4 en efecto el sistema muestra un comportamiento aparentemente estable para el

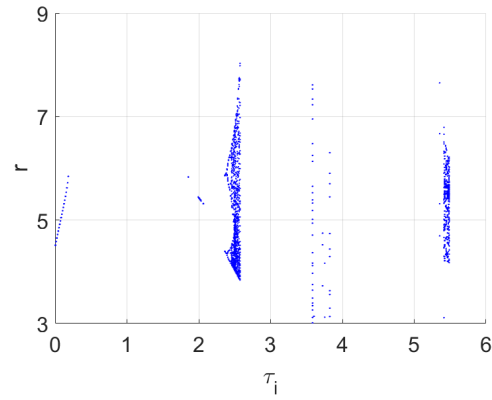


Figura 4: Diagramas de bifurcación variando τ_i con los parámetros descritos en la Tabla I y $\tau_g = 0.56$.

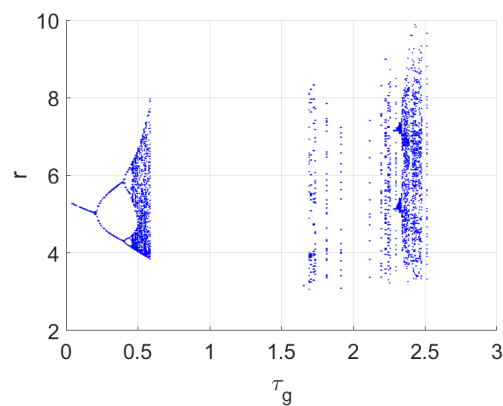


Figura 5: Diagramas de bifurcación variando τ_g con los parámetros descritos en la Tabla I y $\tau_i = 2.55$.

valor de $\tau_i = 0.05$, se observa que mientras el valor de τ_i aumenta el sistema presenta un comportamiento aparentemente caótico, por lo tanto se propone un nuevo valor para $\tau_i = 2.55$. Para corroborar el nuevo valor de τ_i proporciona el comportamiento deseado se obtuvo el diagrama de la figura 5 donde se puede apreciar que existe un comportamiento aparentemente caótico para $\tau_g = 0.56$.

De la misma forma se puede estimar un valor para q , realizando el diagrama de bifurcación correspondiente obteniendo la figura 6. Dado los resultados obtenidos de los diagramas de bifurcación anteriores se proponen los valores de $\tau_g = 0.56$, $\tau_i = 2.55$ y aparentemente $q > 0.935$ para el modelo de ecuaciones diferenciales de orden fraccionario propuesto (5).

IV. ANÁLISIS DE ESTABILIDAD

A continuación se muestra el análisis de estabilidad correspondiente al modelo de orden fraccionario propuesto. Se procede a calcular los puntos de equilibrio resolviendo $\dot{X} = F(\bar{X}) = 0$ con $\bar{X} = [x, y, z]^T \in \mathbb{R}^n$. Como resultado, se obtuvieron 4 puntos de equilibrio ($E_i^* = [x^*, y^*, z^*]$, con $i = 1, 2, 3, 4$) del mo-

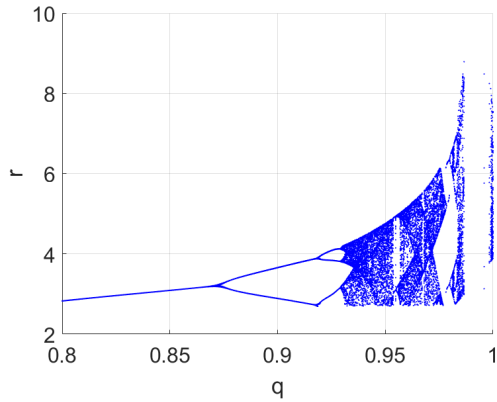


Figura 6: Diagramas de bifurcación variando el orden fraccional q con los parámetros descritos en la Tabla I, $\tau_g = 0.56$ y $\tau_i = 2.55$.

delo fraccionario propuesto (5), los cuales son: $E_1 = (-1.9691, 1.7726, -0.5256)$, $E_2 = (3.1871, 1.5910, 0.9363)$, $E_3 = (1.9696 + 0.8187i, -0.0340 + 0.6510i, 1.0272 - 1.3218i)$ y $E_4 = (1.9696 - 0.8187i, -0.0340 - 0.6510i, 1.0272 + 1.3218i)$.

Para obtener la estabilidad asintótica local de los puntos fijos biológicamente factibles (equilibrios reales) del modelo (5) alrededor de cada uno de los puntos de equilibrio, se calcula la matriz jacobiana J_E , y la ecuación característica del sistema dada por [14], [25], [26]:

$$|J_0 + e^{\lambda\tau} J_\tau - \lambda I|, \quad (6)$$

donde $J_E = J_0 + e^{\lambda\tau} J_\tau$, J_0 es la matriz jacobiana para un retardo $\tau = 0$, J_τ es la matriz jacobiana para el sistema con $\tau > 0$ y I la matriz identidad. De lo anterior se obtuvo la siguiente la matriz jacobiana J_E :

$$J_E = \begin{pmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & 0 & J_{23} \\ 0 & J_{32} & J_{33} \end{pmatrix} \quad (7)$$

Donde $J_{11} = -\frac{1}{4}$, $J_{12} = \frac{59z^* e^{-\lambda\tau_g}}{125}$, $J_{13} = e^{-\lambda\tau_g} \left(\frac{59y^*}{125} + \frac{1}{10} \right)$, $J_{21} = -\frac{3e^{-\lambda\tau_i}}{5}$, $J_{23} = -\frac{5207}{5000z^{*2}}$, $J_{32} = \frac{9}{20} - \frac{3z^*}{10}$ y $J_{33} = \frac{169}{250} - \frac{3z^*}{5} - \frac{3y^*}{10}$, estos resultados se obtuvieron con los valores de los parámetros indicados en la Tabla I donde x^* , y^* y z^* son los puntos de equilibrio.

Dados los puntos de equilibrio reales E_1 y E_2 se desarrolla la ecuación característica (6) obteniendo: $0.2096\lambda^2 - 0.4099e^{-0.56\lambda}e^{-2.55\lambda} - 2.176\lambda - \lambda^3 + 0.1488\lambda e^{-0.56\lambda}e^{-2.55\lambda} - 0.5728$ para E_1 y $-0.2917\lambda - 0.1826e^{-0.56\lambda}e^{-2.55\lambda} - 0.6131\lambda^2 - \lambda^3 - 0.2652\lambda e^{-0.56\lambda}e^{-2.55\lambda} - 0.05022$ para E_2 , respectivamente. Los valores propios respectivos son $\lambda_1 = 0.2179 + 1.1904i$, $\lambda_2 = -0.2972 + 0.4954i$, y $\lambda_3 = 0.2888 - 1.6859i$ para E_1 ; y $\lambda_1 = -0.6160$, $\lambda_{2,3} = 0.001471 \pm 1.4474i$ para E_2 .

Debido a que tenemos un modelo de orden fraccionario la ubicación de los valores propios del lado izquierdo o derecho

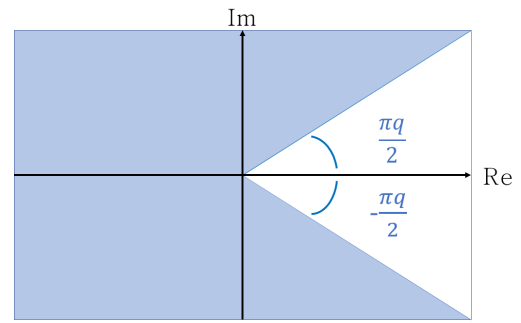


Figura 7: Estabilidad en el plano complejo para sistemas de orden fraccionario.

del plano complejo no es una condición con la cual se pueda demostrar la estabilidad puesto que para los sistemas de orden fraccionario existe un ángulo en el semiplano derecho del plano complejo a partir del cual el sistema es estable, como se muestra en la figura 7 [33].

Por lo tanto el sistema propuesto es estable si la siguiente condición se cumple

$$|arg(\lambda_i)| > \frac{\pi q}{2}, \quad (8)$$

donde λ_i $i = 1, 2, 3$ son los valores propios para cada equilibrio real del sistema. Para conocer el valor mínimo de q que origine caos, se despeja q de (8). Al resolver la ecuación anterior se obtiene que el sistema de orden fraccionario (5) posiblemente presente comportamiento caótico para $q > 0.8847$. resultado en lo que coincide con los resultados del diagrama de bifurcación de la figura 6. Por lo tanto se propone un orden fraccionario $q = 0.964$. Las series de tiempo y los diagramas de fase se muestran en las figuras 8 y 9 respectivamente, donde se aprecia el comportamiento esperado según el análisis realizado.

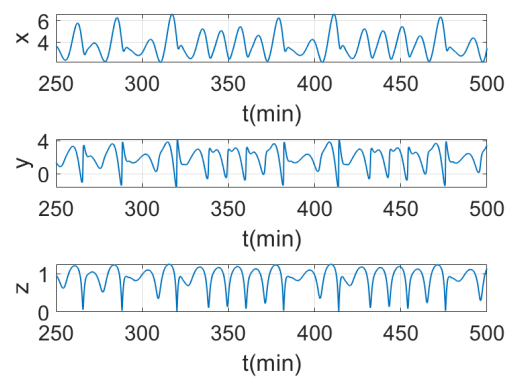


Figura 8: Series de tiempo del sistema de orden fraccionario propuesto con $q = 0.964$, $\tau_g = 0.56$, $\tau_i = 2.55$ y los parámetros descritos en la Tabla (I).

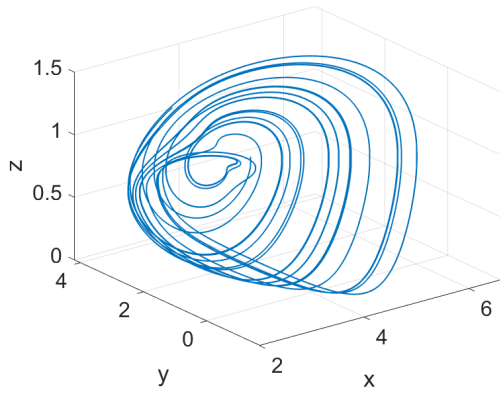


Figura 9: Diagrama de fase del atractor caótico del sistema de orden fraccionario propuesto con $q = 0.964$, $\tau_g = 0.56$, $\tau_i = 2.55$ y los parámetros descritos en la Tabla (I).

Para corroborar la presencia de caos se eliminó un transitorio del 50% del total de iteraciones para el resultado mostrado en las figuras 8 y 9 con la finalidad de observar si el comportamiento caótico persiste.

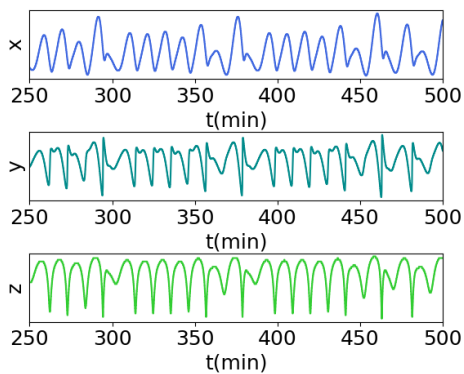


Figura 10: Series de tiempo del sistema discretizado.

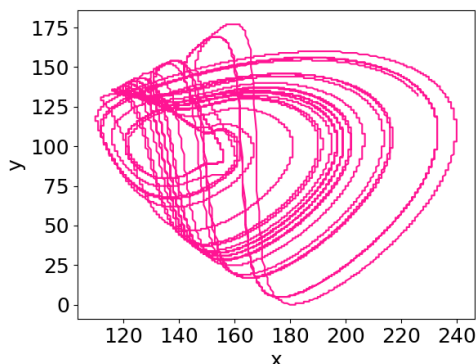


Figura 11: Diagrama de fase xy del sistema discretizado.

V. IMPLEMENTACIÓN

Se propuso el uso de la tarjeta ARM Raspberry Pi 3 para la implementación del sistema propuesto, debido a su bajo costo y compatibilidad con diversos esquemas de procesamiento de datos digitales, además que dentro de los programas que soporta se encuentra Python. Por lo que se transcribieron los programas de Matlab a Python para poder implementar los sistemas.

Es esta sección donde el uso de memoria corta dentro de los sistemas de orden fraccionario, adquiere más importancia, ya que el uso de memoria corta ayuda notablemente con la reducción del tiempo de cómputo.

Para la implementación del sistema se realizó una discretización de los valores de las variables del sistema a valores de 8 bits. Los resultados de simulación mediante Python se muestran en las figuras 10, 11, 12 y 13.

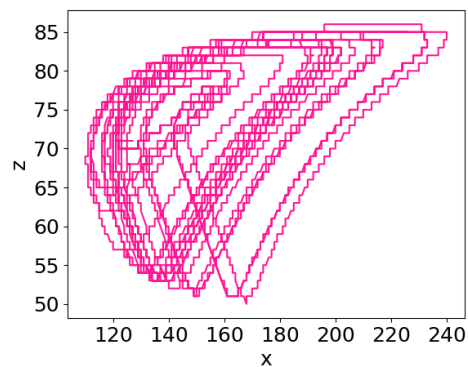


Figura 12: Diagrama de fase xz del sistema discretizado.

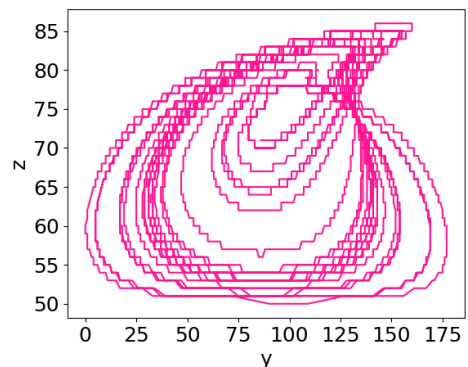


Figura 13: Diagrama de fase yz del sistema discretizado.

Una vez obtenidas las salidas digitales en Python, las cuales serán 2 salidas de 8 bits, por lo que se escogieron los pines GPIO(4,17,27,22,5,6,13,19) y GPIO(23,24,25,12,16,20,21,26) de la Raspberry Pi 3 para las salidas x , y y z . Posteriormente, los pines correspondientes a las salidas del sistema se conectan a un convertidor digital analógico DAC0800 [34].

Los resultados se obtuvieron utilizando un osciloscopio marca Agilent. En las figuras 14 y 15 se muestran las series

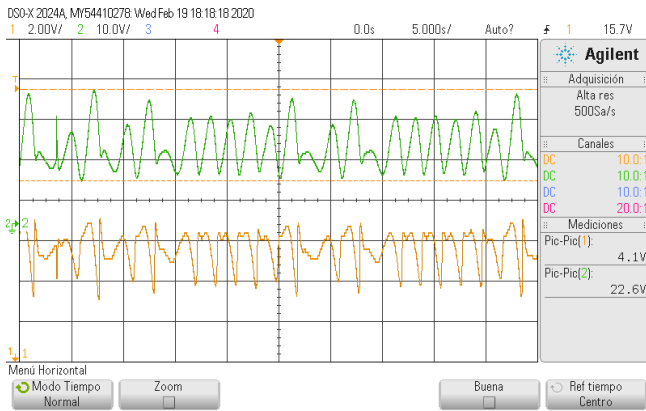


Figura 14: Series de tiempo experimentales de los estados x y y del sistema (5) con $q = 0.964$, $\tau_g = 0.56$, y $\tau_i = 2.55$.

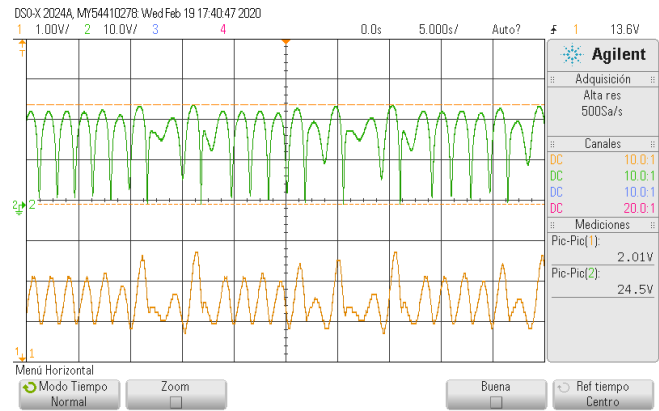


Figura 16: Series de tiempo experimentales de los estados x y z del sistema (5) con $q = 0.964$, $\tau_g = 0.56$, y $\tau_i = 2.55$.

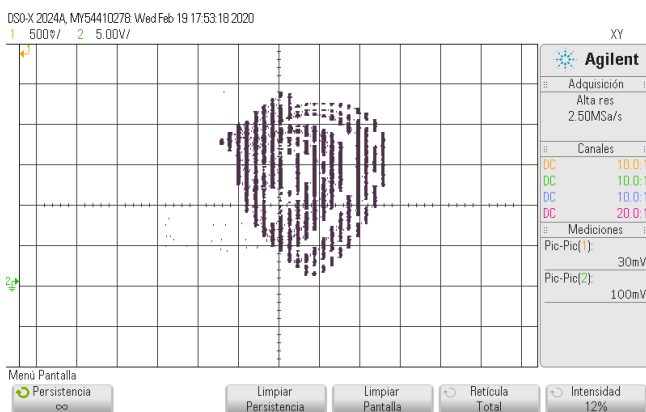


Figura 15: Diagrama de fase $x - y$ del atractor caótico del sistema regulador glucosa-insulina.

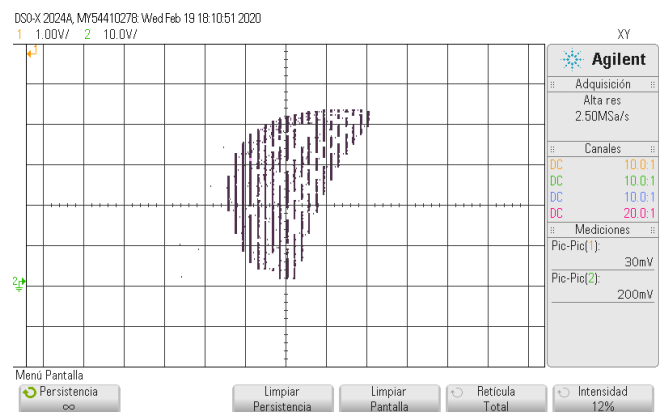


Figura 17: Diagrama de fase $x - z$ del atractor caótico del sistema regulador glucosa-insulina.

de tiempo y fase del sistema regulador de insulina-glucosa, en verde la concentración de insulina $x(t)$ y en amarillo la concentración de glucosa $y(t)$. Para las figuras 16 y 17 se muestran las series de tiempo la concentración de insulina $x(t)$ en amarillo y la concentración de células β $z(t)$ en verde. Por último en las figuras 18 y 19 se muestra en verde la concentración de células β $z(t)$ y en amarillo la concentración de glucosa $y(t)$.

VI. CONCLUSIÓN

El presente trabajo presentó un modelo de orden fraccionario con retardos temporales que describe el sistema de ajuste de glucosa e insulina en el cuerpo humano el cual se encarga de regular los niveles de azúcar en la sangre. Considerando el análisis de estabilidad para sistemas de orden fraccionario con retardos temporales, se determinó el valor mínimo para el orden fraccionario. Este valor determina el punto de bifurcación para observar comportamiento periódico y caótico, el cual indica un funcionamiento normal y no deseado, respectivamente, del sistema regulador glucosa-insulina. Además, se demostró que el sistema de orden fraccionario modifica los retardos previamente reportados. Los valores

modificados de los retardos $\tau_g = 0.56$ y $\tau_i = 2.55$ coinciden con los esperados vistos en pruebas realizadas en seres humanos, donde regularmente son de 1-2 minutos de retrasado a menores con desnutrición estos retardos fueron de hasta 5-10 minutos [32], [35]. Finalmente, los resultados teóricos fueron implementados en una tarjeta ARM.

El presente trabajo podría ser utilizado como referencia para el análisis de la diabetes al presentar un grado extra de libertad por el orden fraccionario, conduciendo a posible mejor entendimiento del sistema regulador glucosa-insulina.

AGRADECIMIENTOS

J.M. Muñoz-Pacheco agradece a CONACyT/MEXICO por el apoyo otorgado por el proyecto No. 258880 (Proyecto Apoyado por el Fondo Sectorial de Investigación para la Educación). B. Fernández Carreón agradece a CONACyT por el apoyo recibido a través de la beca de maestría. Los autores agradecen al Laboratorio Nacional de Supercómputo del Sureste de México perteneciente a la red de laboratorios nacionales CONACyT, por los recursos computacionales, el apoyo y la asistencia técnica (No. de Proyecto: 202001025C).

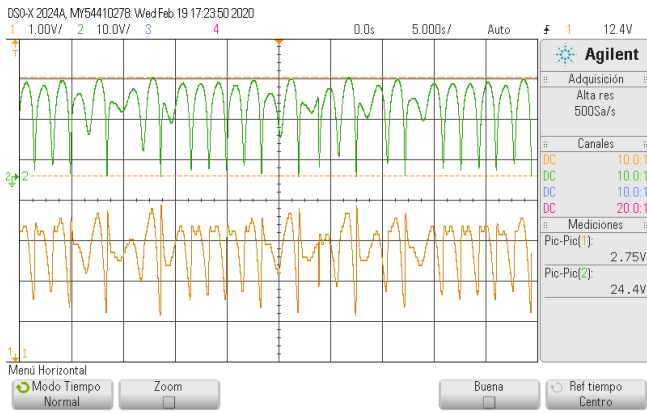


Figura 18: Series de tiempo experimentales de los estados z y y del sistema (5) con $q = 0.964$, $\tau_g = 0.56$, y $\tau_i = 2.55$.

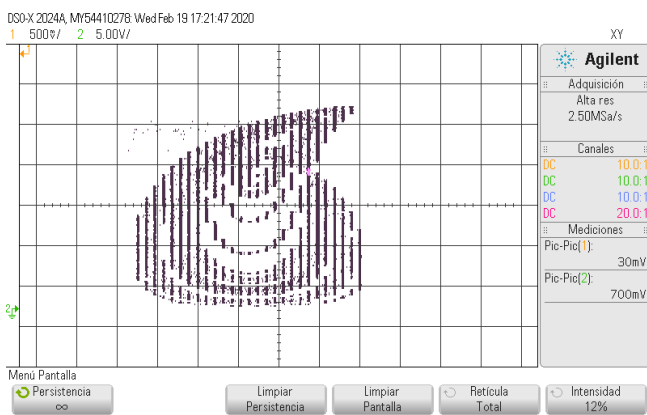


Figura 19: Diagrama de fase $z - y$ del atractor caótico del sistema regulador glucosa-insulina.

REFERENCIAS

- [1] A. D. Association, "Classification and diagnosis of diabetes: Standards of medical care in diabetes," *Diabetes Care*, vol. 43, pp. S14–S31, 2020.
- [2] P. Sadeghi, K. Rajagopal, B. Safarbali, S. Jafari, and P. Duraisamy, "A novel approach to numerical modeling of metabolic system: Investigation of chaotic behavior in diabetes mellitus," *Hindawi, Complexity*, p. 11, 09 2018.
- [3] J. A. Lozano, "Diabetes mellitus," *Offarm*, vol. 25 (10), pp. 66–78, 11 2006. [Online]. Available: <https://www.elsevier.es/es-revista-offarm-4-articulo-diabetes-mellitus-13095504>
- [4] R. Rosalba, B. Ana, A. Carlos, Z. Emiliano, V. Salvador, and B. Tonatiu, "Prevalencia de diabetes por diagnóstico médico previo en México," *Salud Pública de México*, vol. 60, pp. 1–9, 11 2018.
- [5] D. Laroze, "Sobre la importancia de sistemas no lineales en ciencias e ingeniería / on the importance of nonlinear systems in science and engineering," *Ingeniare. Revista chilena de ingeniería*, no. 3, p. 364, 2016.
- [6] D. Laroze, "On the importance of nonlinear systems in science and engineering," *Revista chilena de ingeniería*, vol. 24, no. 3, pp. 362–363, 2016.
- [7] J. M. G. Miranda, *Synchronization and control of chaos*. University of Barcelona, Spain: Imperial College Press, 1987.
- [8] C. Ionescu, A. Lopes, D. Copot, J. Machado, and J. Bates, "The role of fractional calculus in modeling biological phenomena: A review," *Communications in Nonlinear Science and Numerical Simulation*, vol. 51, pp. 141 – 159, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1007570417301119>
- [9] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE TRANSACTIONS ON NEURAL NETWORKS*, vol. 14, no. 6, pp. 1569–1572, 2003.
- [10] H. Kheiri and M. Jafari, "Stability analysis of a fractional order model for HIV/AIDS epidemic in a patchy environment," *Journal of computational and applied mathematics*, vol. 346, pp. 323–339, Jan 2019.
- [11] S. Panahi, Z. Aram, S. Jafari, J. Ma, and J. Sprott, "Modeling of epilepsy based on chaotic artificial neural network," *Chaos, Solitons and Fractals*, vol. 105, p. 150–156, Dec 2017.
- [12] L. L. Rubchinsky, C. Park, and R. M. Worth, "Intermittent neural synchronization in parkinson's disease," *Nonlinear Dynamics*, vol. 68, no. 3, pp. 329–346, May 2012. [Online]. Available: <https://doi.org/10.1007/s11071-011-0223-z>
- [13] S. Khajanchi, M. Perc, and D. Ghosh, "The influence of time delay in a chaotic cancer model," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 10, p. 103101, 2018. [Online]. Available: <https://doi.org/10.1063/1.5052496>
- [14] M. Lakshmanan and D. Senthilkumar, *Dynamics of Nonlinear Time-Delay Systems*, 01 2010.
- [15] J. G. I. Alquézar, "On time delay differential equations," Tesis Maestría, facultad de matemáticas, Universidad de Barcelona, Barcelona, 2015.
- [16] E. Z. Serrano, "Fractional order chaotic systems and their electronic design," Ph.D. dissertation, Instituto Potosino de Investigación científica y tecnológica, A.C., 2017.
- [17] Z. Mukandavire, W. Garira, and C. Chiyaka, "Asymptotic properties of an hiv/aids model with a time delay," *Journal of Mathematical Analysis and Applications*, vol. 330, no. 2, pp. 916 – 933, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0022247X0600850X>
- [18] D. Greenhalgh, Q. Khan, and F. Lewis, "Recurrent epidemic cycles in an infectious disease model with a time delay in loss of vaccine immunity," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 63, no. 5, pp. e779 – e788, 2005, invited Talks from the Fourth World Congress of Nonlinear Analysts (WCNA 2004). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0362546X04005966>
- [19] N. Yoshida and T. Hara, "Global stability of a delayed sir epidemic model with density dependent birth and death rates," *Journal of Computational and Applied Mathematics*, vol. 201, no. 2, pp. 339 – 347, 2007, special Issue: Dynamical Systems Theory and Its Applications to Biology and Environmental Sciences. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377042706000860>
- [20] W. Sarika, Y. Lenbury, K. Kumnungkit, and W. Kunphasuruang, "Modelling glucose-insulin feedback signal interchanges involving β -cells with delays," *ScienceAsia*, vol. 34, pp. 77–86, 2008. [Online]. Available: <http://www.scienceasia.org/2008.34.n1/077.php>
- [21] J. Bajaj, G. S. Rao, J. S. Rao, and R. Khardori, "A mathematical model for insulin kinetics and its application to protein-deficient (malnutrition-related) diabetes mellitus (pddm)," *Journal of Theoretical Biology*, vol. 126, no. 4, pp. 491 – 503, 1987. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0022519387801546>
- [22] Y. Lenbury, S. Ruktamatakul, and S. Amornsamarnkul, "Modeling insulin kinetics: responses to a single oral glucose administration or ambulatory-fed conditions," *Biosystems*, vol. 59, no. 1, pp. 15 – 25, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0303264700001362>
- [23] M. Chuedoung, W. Sarika, and Y. Lenbury, "Dynamical analysis of a nonlinear model for glucose–insulin system incorporating delays and β -cells compartment," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 12, pp. e1048 – e1058, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0362546X09001047>
- [24] F. A. Rihan, "Numerical modeling of fractional-order biological systems," *Hindawi Publishing Corporation, Abstract and Applied Analysis*, vol. 2013, p. 11, 2013.
- [25] O. Naifar, A. M. Nagy, A. B. Makhlof, M. Kharrat, and M. A. Hammami, "Finite-time stability of linear fractional-order time-delay systems," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 1, pp. 180–187, 2019. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.4388>
- [26] M. Lazarević, "Stability and stabilization of fractional order time delay systems," *Scientific Technical Review*, vol. 61, pp. 31–45, 02 2011.
- [27] V. F. Carreón, "Síntesis de la célula beta pancreática basada en amplificadores operacionales," Tesis Maestría, Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla, Puebla, 2017.

- [28] J. M. Muñoz-Pacheco, “Infinitely many hidden attractors in a new fractional-order chaotic system based on a fracmemristor,” *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2185–2196, Oct 2019. [Online]. Available: <https://doi.org/10.1140/epjst/e2019-900035-y>
- [29] E. Zambrano-Serrano, J. M. Muñoz-Pacheco, L. C. Gómez-Pavón, A. Luis-Ramos, and G. Chen, “Synchronization in a fractional-order model of pancreatic β -cells,” *The European Physical Journal Special Topics*, vol. 227, pp. 907–919, 2018.
- [30] P. Palumbo, S. Panunzi, and A. De Gaetano, “Qualitative behavior of a family of delay-differential models of the glucose-insulin system,” *Discrete and Continuous Dynamical Systems. Series B*, vol. 2, 03 2007.
- [31] R. Prager, P. Wallace, and J. M. Olefsky, “In vivo kinetics of insulin action on peripheral glucose disposal and hepatic glucose output in normal and obese subjects.” *The Journal of Clinical Investigation*, vol. 78, no. 2, pp. 472–481, 8 1986. [Online]. Available: <https://www.jci.org/articles/view/112599>
- [32] R. Bertram and M. Pernarowski, “Glucose diffusion in pancreatic islets of langerhans,” *Biophysical Journal*, vol. 74, no. 4, pp. 1722 – 1731, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S000634959877883X>
- [33] I. Petráš, *Fractional-Order Nonlinear Systems*, 2011.
- [34] N. S. Corporation, *DAC0800/DAC0802 8-bit Digital to Analog Converters*. [Online]. Available: <https://www.national.com>
- [35] E. Forrest, P. Robinson, and M. Hazel, “Insulin, growth hormone and carbohydrate tolerance in jamaican children rehabilitated from severe malnutrition,” *MedCarib*.

An embedded platform for Internet of Things Applications

Esther Machado Valenzuela*, Arnoldo Díaz-Ramírez*, Veronica Quintero Rosas*, Mario A. Camarillo† and Wei Wang‡

*Department of Computer Systems

Tecnológico Nacional de México/IT de Mexicali
Mexicali, BC, México

Email: esthermachadov@gmail.com, {adiaz, veronicaquintero}@itmexicali.edu.mx

†Department of Electrical and Electronics

Tecnológico Nacional de México/IT de Mexicali
Mexicali, BC, México

Email: mariocamarillo@itmexicali.edu.mx

‡Department of Computer Science

San Diego State University
San Diego, CA, USA

Email: wwang@sdsu.edu

Abstract—The Internet of Things paradigm allows the interconnection of everyday objects to monitor events of interest. The Internet of Things is been used in many application areas, such as healthcare, environmental monitoring, agro-industry, smart homes, smart cities and sports, just to mention a few. Scarcity of resources is common in Internet of Things applications. For instance, the devices used to monitor the environments are usually not connected to a permanent power supply. Instead, they use small size batteries. For that reason, the efficient use of limited resources is mandatory. These include memory, processor power or wireless communications. In this chapter, we introduce an embedded platform for IoT applications, which is flexible, scalable and energy-efficient. It is based in the use of an inexpensive microcontroller, an embedded operating system, and an efficient wireless network protocol. Details of the set-up of the embedded platform are provided, along with a example of its use.

I. INTRODUCTION

Kevin Ashton used the term Internet of Things (IoT) in 1999 for the first time ever [10]. IoT is a paradigm where everyday objects or things, equipped with sensors, are wirelessly connected to the Internet [15]. Several existing technologies and computational paradigms allowed the emergence of IoT, such as wireless sensor networks [9] and ubiquitous computing [20]. The IoT provides a pervasive presence of objects, and it allows the monitoring of events of interest without human interaction. IoT applications use different technologies, such as sensors, actuators, embedded systems, operating systems, wireless protocols, cloud services, artificial intelligence, among others.

IoT is used in many application areas, such as healthcare, environmental monitoring, agro-industry, smart homes, smart cities and sports, just to mention a few [16]. Recently, a growing interest in the use of IoT in industrial automatization has lead to explore the use of cyber-physical systems (CPS) along with IoT techniques [21]. In fact, IoT and CPS are the

most important enabling technologies of the fourth industrial revolution or Industry 4.0.

Scarcity of resources is common in IoT applications. For instance, the devices used to monitor the environments are usually not connected to a permanent power supply. Instead, they use small size batteries. For that reason, the efficient use of limited resources is mandatory. These include memory, processor power or wireless communications.

In this chapter, we introduce an embedded platform for IoT applications, which is flexible, scalable and energy-efficient. It is based in the use of an inexpensive microcontroller, an embedded operating system, and an efficient wireless network protocol. Details of the set-up of the embedded platform are provided, along with a example of its use.

The structure of the chapter is as follows. In Section 2, the technologies that conform the embedded platform are described. Section 3 explains the installation and set-up of the platform. In Section 4, an example of its use is discussed. Finally, the conclusions and future work are described in Section 5.

II. IOT ENABLING TECHNOLOGIES

A typical architecture of an IoT application is shown in Fig 1. The *perception layer* includes the physical objects (things), the sensors and the electronic devices used to collect and pre-process the information acquired from the sensors. The *network layer* includes the protocols used to send the collected information to a node with more computational resources. The *application layer*, where the information is processed and where decisions are taken, is generally conformed of one or several applications, which run on servers or in the cloud.

To implement the perception and network layer, different approaches have been used. One of them is through the use of complete platforms such as Arduino or Raspberry-pi,

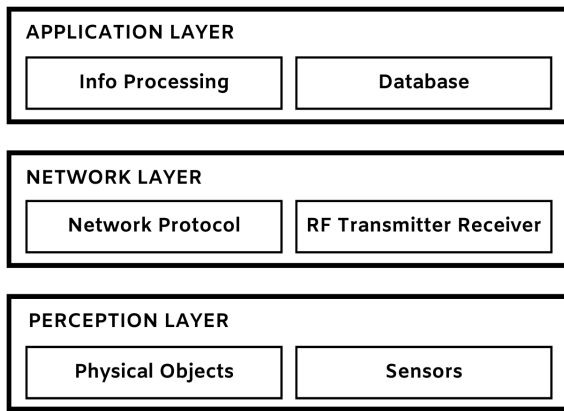


Fig. 1. IoT architecture

which are easy to use and configure. However, since these platforms are aimed at students without a background in electronics and programming, they are not energy-efficient and not practical to be used in most production IoT scenarios. Another approach consists in using inexpensive microcontrollers which run just one process repeatedly. This approach does not support the execution of concurrent activities (processes) and lacks flexibility. The third option is the use of an inexpensive microcontroller board with more computational resources than the previous approach, along with the use of an embedded operating system. This approach offers many advantages. On the one hand, it eases the connection of the platforms with the sensors and physical objects, while maintaining an energy-efficient architecture. On the other hand, the use of an operating system provides flexibility to the platform, allowing the execution of concurrent activities. Additionally, it allows the use of programming standards, which guarantees software portability, as well as the integration of libraries for many specific tasks, such as network communication protocols. This last approach is used in the proposed architecture, through the implementation of an embedded system which integrates both the perception and communication layer. The architecture is conformed of a microcontroller board or microcontroller unit (MCU), an embedded operating system, and a wireless communication platform.

A. Microcontroller board

An embedded system is a computational system embedded into a larger system. Embedded systems are generally implemented using microcontrollers. A microcontroller is a system-on-a-chip (SOC) and has CPU, RAM, timers, I/O ports, among other components. A microcontroller executes a single process to control larger systems such as robots, automobiles or Unmanned Aerial Vehicles, to mention a few. MCUs are available with on board memory and I/O ports, avoiding the building of a circuit to connect with external RAM, ROM and peripheral chips.

The STM32F407VET6 board from STMicroelectronics is a high performance MCU designed for medical, industrial

and consumer applications that require a high performance and high level of integration in a small unit, usually 10 x 10 mm [19]. This MCU uses an ARM 32-bit Cortex-M4 core with Float Point Unit, running a 168 MHz, 1 Mb Flash memory, and 192 Kb RAM memory. It includes 2 USB on-the-go (OTG) ports (one with high speed support), up to 15 communication interfaces, up to 15 timers, easily extendable memory range using the flexible static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories, and an analog true random generator, along other features. It is an inexpensive MCU (approximately 10 US dollars) which allows the implementation of robust IoT applications. For this reason, this MCU was chosen to implement the proposed embedded system.

B. Embedded operating system

An operating system (OS) is a software that manages the computer resources, such as CPU, memory, I/O devices and processes, and serves as an intermediary between the hardware and the user's applications. A general-purpose operating systems (GPOS) seeks for a fair distribution of the computational resources among the applications. In contrast, a real-time operating system (RTOS) assigns the computational resources based on the applications priority. For instance, a RTOS preempts a running process if a process with higher priority is activated.

An embedded operating system is a special kind of OS designed to run in an embedded system (e.g., a MCU). Due to the scarcity of computational resources of the embedded system, the embedded OS is specifically designed to be small and efficient, eliminating those functionalities which are not required, such as graphical user interface or hard disk support.

The widespread of the IoT paradigm has motivated the development of embedded hardware platforms and embedded OSs aimed to the development of IoT applications. Examples of embedded OS for IoT are Amazon FreeRTOS [1], Apache Mynewt [2], Contiki [4] or NuttX [3].

NuttX is a real-time embedded operating system developed by Gregory Nutt. Currently, NuttX is an effort undergoing incubation at the Apache Software Foundation (ASF). It is a compact and stable RTOS that emphasizes standards compliance, such as the Portable Operating System Interface (POSIX) and the American National Standards Institute (ANSI). Standard compliance is an important feature since it guarantees software portability and avoids the need of learning a new programming API. Also, NuttX has many important features, like POSIX/ANSI-like task controls, named message queues, counting semaphores, clocks/timers, signals, pthreads, robust mutexes, cancellation points, environment variables, filesystem, loadable kernel modules, lightweight embedded shared libraries and custom NuttX C library, just to mention a few. The use of NuttX on the selected embedded platform (i.e., the STM32F407 MCU) allows the execution of concurrent activities through the implementation of POSIX threads and semaphores. Consequently, the embedded platform is able to execute several tasks concurrently, as reading data from

different sensors, processing the sensed data, sending the processed data through the communication platform, and so on. This overcomes the limitation inherent to the microcontroller programming environment, where only a single task can be executed. Due to its stability, standard compliance and its important features, NuttX was selected for the proposed embedded platform.

C. Wireless communication platform

IoT applications include a communication layer to send the data acquired by the sensors to another devices and/or to end users. Due to the nature of the vast majority of IoT applications, the most common way of implementing the communication layer is through the use of low-power wireless technologies. Some wireless protocols are good candidates for IoT applications, which include Bluetooth Low Energy [13], ZigBee [18], Z-Wave [11], 6LoWPAN [17], WiFi-ah (HaLow) [14] or LoRaWAN [12].

ZigBee is a routing wireless protocol which is used along with IEEE 802.15.4 networks. The IEEE 802.15.4 or WPAN standard allows the implementation of self-organizing wireless networks with low-power consumption and short-range communication. Since the maximum distance allowed to communicate two WPAN nodes is approximately 10 meters (up to 100 meters with higher power consumption), ZigBee is usually used on top of the IEEE 802.15.4 to extend the communication range of the wireless network. This is accomplished by the use of intermediate ZigBee nodes as routers, allowing the implementation of an (in theory) unlimited sized network. ZigBee is developed by the ZigBee Alliance [8]. Due to its low-power consumption and availability, ZigBee and the IEEE 802.15.4 were selected as part of the proposed embedded platform. Particularly, the XBee XB24CZ7WIT-004 module from Digi was used [6], along with the XBee USB Adapter is a UART communication board [5]. The adapter is used to easily connect the XBee module to the MCU board.

III. INSTALLING NUTTX ON THE MCU

To install NuttX in the microcontroller it is required the use of a host computer. In this case, a personal computer running the Ubuntu Linux operating system was used. Some packages are required to configure, compile and install NuttX. The packages can be installed using the following command:

```
sudo apt-get install automake bison flex
build-essential gcc-arm-none-eabi gperf
git libncurses5-dev libtool libusb-dev
libusb-1.0.0-dev pkg-config minicom
```

To configure and compile NuttX, it is recommended to store all files at a specific folder. For example, a folder called `nuttxspace` can be created:

```
mkdir nuttxspace
```

Once the host computer's required packages are installed and the working folder is created, the next step is to download the OpenOCD source files from the git repository. OpenCd is required to enable the microcontroller programmer device,

as well as to perform debugging functions. This is done as follows:

```
git clone http://repo.or.cz/r/openocd.git
```

As a result of cloning the repository, several new folders will be created. One of them is `openocd`. In that folder, the `bootstrap` command must be executed to create the configuration files:

```
cd openocd
./bootstrap
```

Next, it is time to configure and enable the programmers supported by OpenOCD, such as the *STlink* programmer that was used in our example:

```
./configure --enable-internal-jimtbl
--enable-maintainer-mode
--disable-werror --disable-shared
--enable-stlink
--enable-jlink --enable-rlink
--enable-vslink --enable-ti-icdi
--enable-remote-bitbang
```

We can proceed to the compile and install OpenOCD:

```
make
sudo make install
```

After OpenOCD is installed, the next step is download NuttX from the git repository, which includes the OS source files, libraries, utilities, examples (apps), among others tools. This should be done in the working directory `nuttxspace`.

```
cd ..
git clone https://bitbucket.org/nuttx/nuttx
git clone https://bitbucket.org/nuttx/apps
git clone https://bitbucket.org/nuttx/tools
```

Now we have to configure the tools. This has to be done in the `kconfig-frontends` folder:

```
cd ~/nuttxspace/tools/kconfig-frontends/
./configure
```

And then we compile and install it.

```
make
sudo make install
```

It is required to create the required links:

```
sudo ln -s /usr/bin/ldconfig
```

We are ready to configure NuttX for the specific microcontroller to be used in which it will run. In our case, we use the STM32F407 microcontroller. For that reason, we have to be in the appropriate folder to proceed. To configure the STM32F407 microcontroller we need type the following:

```
cd ~/nuttxspace/nuttx
cd boards/arm/stm32/stm32f4discovery/configs/usbnsd
```

The `defconfig` file contains the system's configuration that is used when compiling the kernel. Additionally to the default configuration options, it is needed to add the following ones:


```

CONFIG_FAT_LCNAMES=y
CONFIG_FAT_LFN=y
CONFIG_FS_FAT=y
CONFIG_LIBC_FLOATINGPOINT=y
CONFIG_NSH_DISABLE_GET=y
CONFIG_NSH_DISABLE_IFUPDOWN=y
CONFIG_NSH_DISABLE_PUT=y
CONFIG_NSH_DISABLE_WGET=y
CONFIG_SCHED_HPWORK=y
CONFIG_SCHED_HPWORKPRIORITY=192
CONFIG_SENSORS=y
CONFIG_SENSORS_DHTXX=y
CONFIG_STM32_CCMECLUDE=y
CONFIG_STM32_DISABLE_IDLE_SLEEP_DURING_DEBUG=y
CONFIG_STM32_FREERUN=y
CONFIG_STM32_TIM1=y
CONFIG_TASK_NAME_SIZE=32
    
```

We can proceed now with the configuration of NuttX for the selected microcontroller. This is done in the `tools` folder.

```

cd ~/nuttxspace/nuttx/tools
./configure.sh stm32f4discovery/usbnsh
    
```

At this point, our system is ready to be compiled and to be installed in the microcontroller. It is important to remember that NuttX executes one process (application) only. Once it is developed and compiled along with NuttX, the OS can be installed in the microcontroller, where the process will run.

To communicate the host computer and the microcontroller, a programmer device is required. In our case, the *STlink V2* programmer was used. The connection between the programmer and the microcontroller is shown in Fig. 1. The programmer is connected to the host computer using the USB port. Once they are connected, into the `openocd/ontrib` folder, the following code will allow the access to the programmer:

```

sudo cp 60-openocd.rules /etc/udev/rules.d/
sudo udevadm trigger
    
```

IV. EXAMPLE OF AN IOT APPLICATION

To explain the use of the proposed platform, a simple IoT application was developed. The architecture of the application is shown in Fig. 2. The perception layer is composed of several sensors and a monitoring module. In this case, temperature and humidity sensors are used. The monitoring module is implemented using the STM32F407VET6 board and NuttX. The communication layer is implemented as a wireless communication module that uses the IEEE 802.15.4 and ZigBee protocol stack. The application layer is implemented as a process (server) running in a computer with the the Linux Ubuntu operating system and the IEEE 802.15.4 and ZigBee protocol stack.

Fig. 3 shows the details of the set up of the hardware used in the application. To read the temperature and humidity values of the environment, the DHT11 sensor was used. It is a digital sensor with a NTC thermistor to measure temperature. It outputs the temperature and humidity values in a serial fashion. The sensor measures temperature from 0°C to 50°C and humidity from 20% to 90%, with an accuracy of $\pm 1^\circ\text{C}$ and $\pm 1\%$.

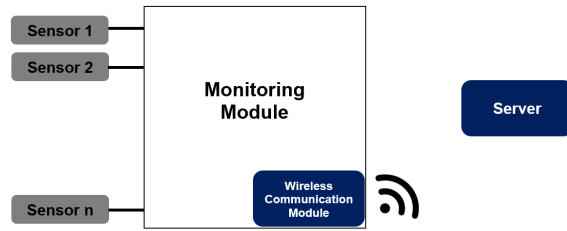


Fig. 2. Architecture of the application

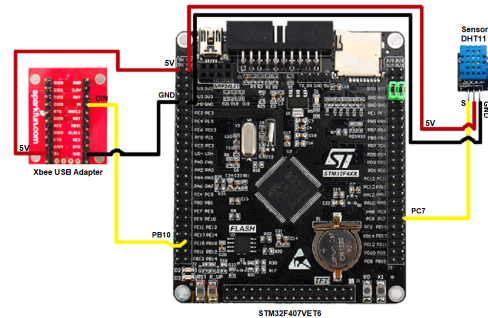


Fig. 3. Hardware set up

To use the DHT11 sensor with the STM32F407VET6 board, the files `stm32_dhtxx.c` and `stm32_gpio.c` must be copied onto the `nuttx/boards/arm/stm32/stm32f4discovery/src` folder. The first one adds board support for the sensor and it is located in the `nuttx/boards/arm/stm32/olimex-stm32-p407/src` folder. The second one is used to facilitate communication between the board and the sensor and can be found in the `nuttx/boards/arm/stm32/stm32f103-minimum/src`. Once this file is copied onto the correct folder, it has to be modified. Particularly, the reference to the `stm32f103-minimum.h` header file must be changed for a reference to the `stm32f4discovery.h` header file.

Additionally, some files must be modified in order to use the sensor. One of them is the `stm32f4discovery.h` header file. In the section labeled `/* STM32F4 Discovery GPIOs */`, the following lines must be added:

```

/* DHTxx pin configuration */
#define GPIO_DHTXX_PIN (GPIO_PORTC|GPIO_PIN7)
#define GPIO_DHTXX_PIN_OUTPUT (GPIO_OUTPUT|
    GPIO_FLOAT|GPIO_SPEED_100MHz|
    GPIO_DHTXX_PIN)
#define GPIO_DHTXX_PIN_INPUT (GPIO_INPUT|
    GPIO_FLOAT|GPIO_DHTXX_PIN)
    
```

Also, a new section must be included in the same file:

```

/* GPIO pins used by the GPIO Subsystem */
#define BOARD_NGPIOIN 1
#define BOARD_NGPIOUT 1
#define BOARD_NGPIPOINT 1
#define GPIO_IN1 (GPIO_INPUT|
    GPIO_CNF_INFLOAT|
    GPIO_PORTA|GPIO_PIN0)
#define GPIO_OUT1 (GPIO_OUTPUT|
    GPIO_CNF_OUTPPP)
    
```

```

GPIO_MODE_50MHz |
GPIO_OUTPUT_SET |
GPIO_PORTA | GPIO_PIN1)
#define GPIO_INT1 (GPIO_INTPUT |
GPIO_CNF_INFLOAT |
GPIO_PORTA | GPIO_PIN2)

Then we need to add the following lines in the section /*
Public Functions */ :

/* GPIO pins used by the GPIO Subsystem */
/*****
* Name: stm32_dhtxx_initialize
*
* Description:
* Called to initialize the DHTxx sensor
*****/
#ifdef CONFIG_SENSORS_DHTXX
int stm32_dhtxx_initialize (FAR const
char *devpath);
#endif

```

```

Another file that must be modified is stm32_bringup.c,
adding the following lines in the function int
stm32_bringup(void):

#ifdef CONFIG_SENSORS_DHTXX
ret = stm32_dhtxx_initialize ("/dev/dht0");
if (ret < 0) {
syslog (LOG_ERR,
"ERROR: stm32_dhtxx_initialize () failed:
%d\n", ret);
}
#endif

```

```

Finally, the Makefile file also must be modified, adding
the following lines:

ifeq ($(CONFIG_SENSORS_DHTXX),y)
CSRCS += stm32_dhtxx.c
endif

```

Once NuttX is configured and installed to work with the selected sensor, the next step is to configure the XBee module. The easiest way to do it is using the Next Generation Configuration Platform for XBee/RF Solutions (XCTU) software tool from Digi [7]. After the XCTU application is installed and running, the XBee modules must be connected to the host computer, via the USB port, to configure them. In our example, two XBee modules are required: one for the communication layer (embedded platform), called the client, and another for the application layer (the server), called the server. To configure each XBee module, the port used for connection should be selected in the XCTU application and eventually the module is automatically detected, as shown in Fig. 4.

The GUI shows the options that can be selected to configure the device, as it can be seen in Fig. 5. Among the options that must be set for the device that will operate as a server, we have the ID, that is a value that is defined by the user but has to be the same for all the devices of the network. Also, the coordinator option (CE) and the AP (API Enable) mode must be activated, as shown in Fig. 6. The most important options for the client device are shown in Fig. 7, which are the ID, that must be the same as the server device, the AP

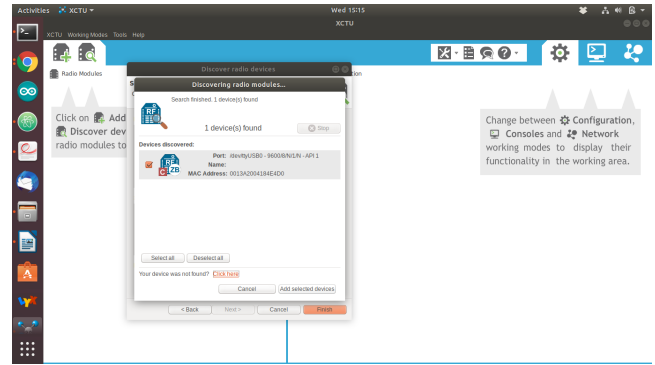


Fig. 4. Discovering the XBee module

option, that must be in “transparent mode [0]”, and the Pin used to receive data from the MCU board. Additionally, the client device can be configured to read the data from the MCU board (i.e., data acquired by the sensors), send it, and then to sleep for a defined period. Once it elapses, the cycle will be repeated infinitely. To defined this behavior, the “sleep mode” option must be set to “Cyclic Sleep”, while the “Cyclic Sleep Period” must be set for the amount of time desired. However, the periodic behavior can be also defined at application level.

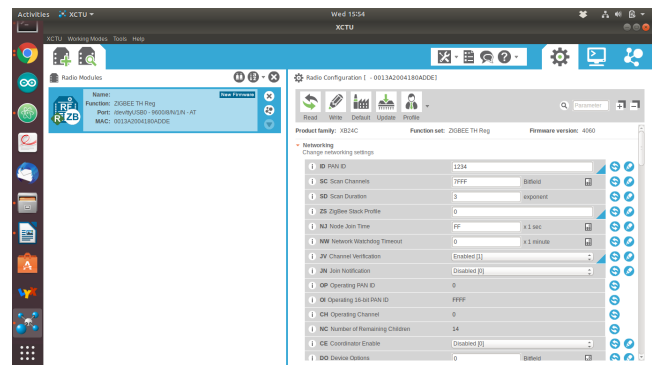


Fig. 5. XBee Configuration options

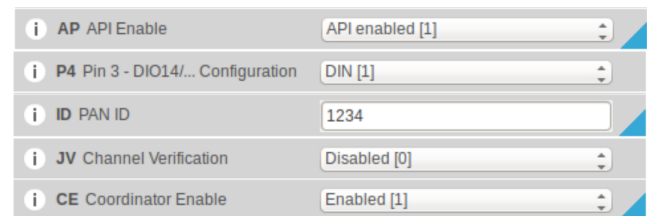


Fig. 6. Server configuration options

After the sensors and the XBee module are configured to work with the embedded system, the application can be developed. Algorithm 1 shows its pseudo-code. First, the system parameters and variables are initialized, such as data structures, semaphores, threads and functions prototypes. Afterwards, within an infinite loop, the application reads the temperature

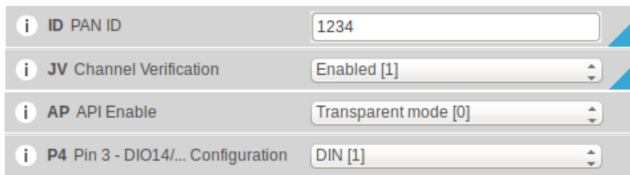


Fig. 7. Client configuration options

Algorithm 1 Pseudo-code of the monitoring (client) module

```

initialize system parameters

while (1)
{
    read temperature and humidity from sensor;
    store temperature and humidity in shared buffer;
    read temperature and humidity from shared buffer;
    send temperature and humidity values to server;
}
    
```

and the humidity values from the sensor(s). These values are stored in a shared buffer, which is protected by a semaphore to guarantee that no race condition will take place. In the POSIX standard, a semaphore is implemented as a mutex. Then, the application access the shared buffer to check if new values have been stored. In such a case, the new values are sent to the server.

To allow concurrent activities, such as reading from several sensors while sending information to the server, the POSIX threads (pthreads) API is used. The use of threads offers many advantages. For instance, the sensor's reading period may be different between sensors, depending on the context of the application. Also, the period to send the information to sensor may vary from sensor to sensor. Moreover, if the system includes actuators, they can be controlled by the embedded platform whenever is required, using threads and the real-time capabilities of NuttX. Concurrency is not limited by the sequential execution of code. Each thread is an independent execution unit.

The applications creates two POSIX threads. The first one, called the *producer thread*, periodically reads the temperature and humidity values from the sensor. Then it acquires exclusive access to the shared buffer and stores the values. The second thread, called the *consumer thread*, gets exclusive access to the shared buffer, checks if there are new values, to read and send them wirelessly to the server. The shared buffer, implemented as a C structure, is defined as follows:

```

struct TEMP {
    int tag;
    double temperature;
    double humidity;
    int read;
};
    
```

The variable `tag` contains a consecutive number. The variable `read` is used as a flag. If its value is 0, it means that the temperature and humidity values have been already read by the consumer thread. Conversely, if its value is set to

1, it means that the values are new and need to be sent to the server.

The code (C language) of the producer thread is shown in Algorithm 2. The thread reads the values from the sensor every 10 seconds. To do that, it reads the data from the serial port. Then, it gets exclusive access to the shared buffer using the `pthread_mutex_lock()` system call. Later, it stores the new values onto the shared buffer and releases the semaphore using the `pthread_mutex_unlock()` system call.

Algorithm 2 Producer thread code

```

void *producer_thread(){
    struct dhtxx_sensor_data_s data;
    int fd;
    int ret;
    int i = 1;
    // open the dht0 file for reading sensor values
    fd = open("/dev/dht0", O_RDWR);
    while (1) {
        ret = read(fd, &data,
                  sizeof(struct dhtxx_sensor_data_s));
        // Verify read is not empty
        if (ret < 0)
        {
            printf("Read error.\n");
            printf("Sensor reported error %d\n",
                  data.status);
        }
        else
        {
            /* Get exclusive access to buffer */
            pthread_mutex_lock(&mutex1);
            /* Update struct shared_temp */
            shared_temp.temperature = data.temp;
            shared_temp.humidity = data.hum;
            shared_temp.tag = i;
            shared_temp.read = 1;
            i++;
            pthread_mutex_unlock(&mutex1);
            // sleep ()
            sleep(10);
        }
    }
}
    
```

Algorithm 3 shows the code of the consumer thread. It also reads periodically the shared buffer for new values to send them wirelessly to the server. First, it gets exclusive access to the shared data structure to check for new values. Then, it opens the output port, which is connected to the XBee module. It stores the information to be sent in the string named buffer, and sends it through the output port using the `fprintf()` system call.

The code of the `main()` function is shown in Algorithm 4. First, an instance of the structure (shared buffer) and the mutex (semaphore) are declared globally. In the `main()` function, after the mutex is initialized, the threads area created using the `pthread_create()` system call. Before exiting, we wait for the threads to complete its execution and release the resources used by the mutex.

To compile and install the application into the MCU, a new folder must be created under the `apps/example` folder. This folder must contain four files. One of them contains the application C code. The remaining three files are

Algorithm 3 Consumer thread code

```
void *consumer_thread(){
    char buffer[100],temp[318], hum[318], tag[12];
    FILE *send_port;
    while (1) {
        /* Get exclusive access */
        pthread_mutex_lock(&mutex1);
        if (shared_temp.read == 1) {
            memset(buffer, '\0', sizeof(buffer));
            // Open the ttyS1 for sending the data
            send_port = fopen("/dev/ttyS1", "r+");
            sprintf(tag, 12, "%i", shared_temp.tag);
            sprintf(temp, 318, "%f",
                shared_temp.temperature);
            sprintf(hum, 318, "%f",
                shared_temp.humidity);
            strcat(buffer, "(");
            strcat(buffer, "\tag\":" );
            strcat(buffer, tag);
            strcat(buffer, ",\ntemperature\":" );
            strcat(buffer, temp);
            strcat(buffer, ",\nhumidity\":" );
            strcat(buffer, hum);
            strcat(buffer, ")");
            // Send the received data
            fprintf(send_port, "%s", buffer);
            // Close ttyS1
            fclose (send_port);
            shared_temp.read = 0;
        }
        pthread_mutex_unlock(&mutex1);
        sleep(10);
    }
    return (NULL);
}
```

required for the compilation process: Makefile, Kconfig and Make.defs. Once these files are created we need to configure the compilation process:

```
make menuconfig
```

Fig. 8 shows the GUI to configure the system before compilation. It is important to verify that the following options are correctly set:

- System type :
 - STM32 Peripheral support
 - USART2 : Set
 - USART3 : Set
- Device drivers :
 - Serial driver support
 - Serial console : No serial console
 - USART1 configuration : 9600 baud rate
 - USART2 configuration : 9600 baud rate
- Application support
 - Examples
 - Mutex example : Set

It is important to note that in our example, the application name is “Mutex example”. If the folder and the configuration files were set correctly (i.e., Makefile, Kconfig and Make.defs), the application name should appear in the NuttX configuration menu and must be selected (set), as shown in Fig 9.

Algorithm 4 Code of main() function

```
struct TEMP shared_temp;
pthread_mutex_t mutex1;

int main(int argc, FAR char *argv[])
{
    pthread_t producer_thread_id, consumer_thread_id;

    /* Initialize mutex */
    pthread_mutex_init(&mutex1, NULL);

    // create threads
    pthread_create(&producer_thread_id, NULL,
        producer_thread, NULL);
    pthread_create(&consumer_thread_id, NULL,
        consumer_thread, NULL);

    /* Wait for all threads to complete */
    pthread_join(producer_thread_id, NULL);
    pthread_join(consumer_thread_id, NULL);

    /* Clean up and exit */
    pthread_mutex_destroy(&mutex1);
    return (0);
}
```

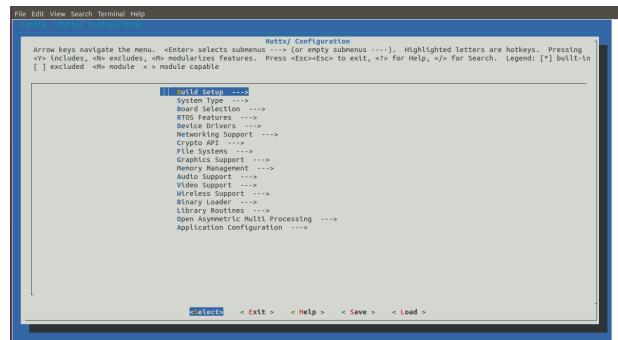


Fig. 8. NuttX configuration options

Then we can proceed to compile NuttX, including the new application:

```
make
```

To install and execute the application and the NuttX kernel into the MCU, we use the STLink V2 programmer. Fig. 10

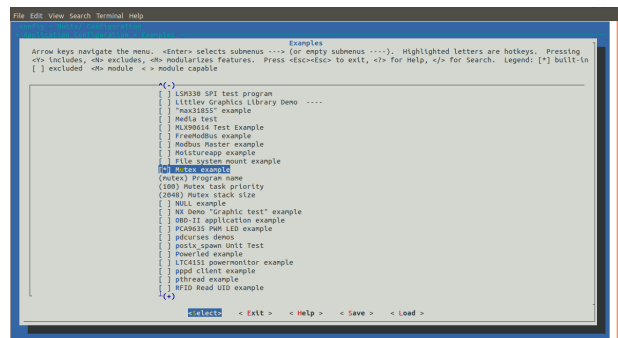


Fig. 9. Selecting the application

shows how to connect it to the MCU. Once connected, we execute the following command:

```
openocd -f interface/stlink-v2.cfg -f
target/stm32f4x.cfg -c init -c
"reset halt" -c "flash write_image
erase nuttx.bin 0x08000000"
```

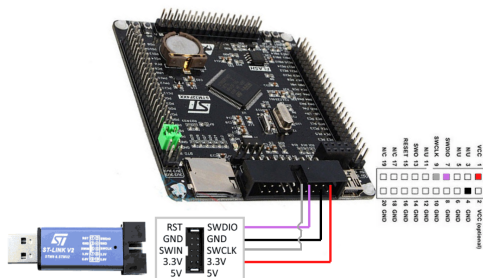


Fig. 10. STLink V2 programmer connection to the MCU

To test the application, we may connect the MCU to the host computer and execute the Minicom application:

```
sudo minicom -s
```

Then, under the *Serial port setup* configuration menu, we select the following options:

```
A - Serial device: /dev/ttyACM0
E - BPS/Par/ Bits: 9600 8N1
F - Hardware flow control: no
```

After this, we can see the application messages in the Nutshell of Minicom, as shown in Fig. 11. In Fig. 12, it can be observed the information received by the server application.

As it can be seen, the proposed platform allows the implementation of robust and flexible IoT applications. Additionally, the development of IoT-oriented software using programming standards, such as POSIX, promotes portability and facilitates software development.

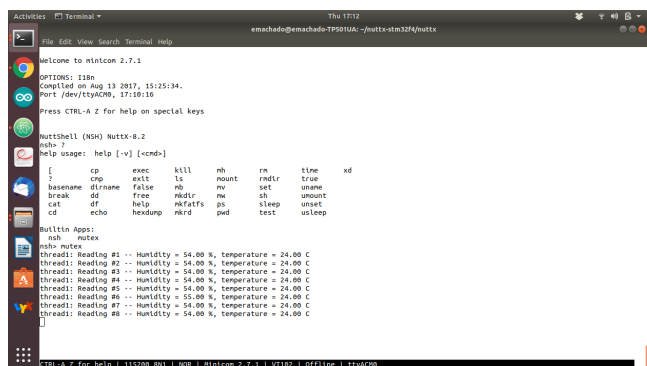


Fig. 11. Output of the consumer code

V. CONCLUSIONS AND FUTURE WORK

The emergence of the Internet of Things paradigm has motivated its application in a wide range of areas, such as healthcare, environment monitoring, smart homes and cities,

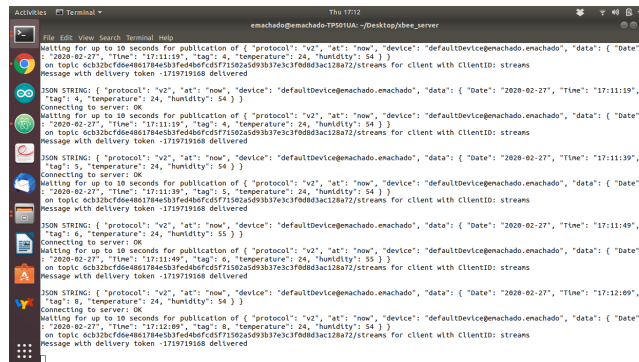


Fig. 12. Output of the server application

and in industrial applications. However, the limited resources of the IoT hardware platforms must be solved using low-power consumption, efficient and flexible infrastructure.

In this chapter, we introduced an embedded platform for IoT applications, which is flexible, scalable and energy-efficient. It is based in the use of an inexpensive microcontroller, an embedded operating system, and an efficient wireless network protocol. Details of the set-up of the embedded platform were provided, along with an example of its use. It was shown that the proposed platform is relatively easy to configure and use, allows concurrent execution of threads, and can be used in many IoT real-world applications.

As future work, we plan to use the proposed platform in agro-industry and industrial applications, along with different wireless technologies, such as LoRaWAN.

REFERENCES

- [1] Amazon FreeRTOS. url: <https://aws.amazon.com/freertos/>. last checked: March 2, 2020.
- [2] Apache Mynett. url: <https://mynewt.apache.org>. last checked: March 2, 2020.
- [3] Apache NuttX. url: <https://nuttx.apache.org>. last checked: March 2, 2020.
- [4] Contiki-OS. url: <http://www.contiki-os.org>. last checked: March 2, 2020.
- [5] XBee USB Adapter. url: <https://www.waveshare.com/product/xbee-usb-adapter.htm>. lastchecked: March 2, 2020.
- [6] XBee XB24CZ7WIT-004 datasheet. url: https://www.digi.com/pdf/ds_xbee_zigbee.pdf. last checked: March 2, 2020.
- [7] XCTU. url: <https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu>. lastchecked: March 2, 2020.
- [8] The ZigBee Alliance. url: <https://zigbeealliance.org>. last checked: March 2, 2020.
- [9] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422, 2002.
- [10] Kevin Ashton. That internet of things thing. *RFID Journal*, 2009.
- [11] Christopher W. Badenhop, Scott R. Graham, Benjamin W. Ramsey, Barry E. Mullins, and Logan O. Mailloux. The z-wave routing protocol and its security implications. *Computers and Security*, 68:112 – 129, 2017.
- [12] J. de Carvalho Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic, and A. L. L. Aquino. Lorawan — a low power wan protocol for internet of things: A review and opportunities. In *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pages 1–6, July 2017.
- [13] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9):11734–11753, Aug 2012.

- [14] E. Khorov, A. Krotov, A. Lyakhov, R. Yusupov, M. Condoluci, M. Dohler, and I. Akyildiz. Enabling the internet of things with wi-fi halow—performance evaluation of the restricted access window. *IEEE Access*, 7:127402–127415, 2019.
- [15] Shancang Li, Li Da Xu, and Shanshan Zhao. The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259, 2015.
- [16] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497 – 1516, 2012.
- [17] Geoff Mulligan. The 6lowpan architecture. In *Proceedings of the 4th Workshop on Embedded Networked Sensors*, EmNets '07, pages 78–82, New York, NY, USA, 2007. Association for Computing Machinery.
- [18] S. Safaric and K. Malaric. Zigbee wireless standard. In *Proceedings ELMAR 2006*, pages 259–262, June 2006.
- [19] STMicroelectronics. STM32F407 MCU datasheet. url: <https://www.st.com/en/microcontrollers-microprocessors/stm32f407-417.html>, Last checked: March 2,2020.
- [20] Mark Weiser. The computer for the 21 st century. *Scientific American*, 265(3):94–105, 1991.
- [21] M. Wollschlaeger, T. Sauter, and J. Jasperneite. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1):17–27, March 2017.

Monitorización de la integridad estructural para aerogeneradores *offshore* con estructura tipo *jacket* mediante métodos de aprendizaje automático.

Gabriela Aquino González
 Facultad de Ciencias de la Electrónica (FCE)
 Benemérita Universidad Autónoma de Puebla (BUAP)
 Puebla, México.
 gabriela.aquino@alumno.buap.mx

Yolanda Vidal Seguí
 Control, Modeling, Identification and Applications (CoDALab)
 Universitat Politècnica de Catalunya (UPC)
 Barcelona, España.
 yolanda.vidal@upc.edu

José Eligió Moisés Gutiérrez Arias
 Facultad de Ciencias de la Electrónica (FCE)
 Benemérita Universidad Autónoma de Puebla (BUAP)
 Puebla, México.
 arigutmses5@gmail.com

Francesc Pozo Montero
 Control, Modeling, Identification and Applications (CoDALab)
 Universitat Politècnica de Catalunya (UPC)
 Barcelona, España.
 francesc.pozo@upc.edu

Resumen—La energía eólica *offshore* tiene por objetivo aprovechar la fuerza del viento producida en alta mar. Para ello se colocan aerogeneradores, algunos de los cuales se encuentran en aguas profundas y utilizan plataformas con estructuras de soporte tipo *jacket* que hoy en día son la mejor solución para hacer frente a este entorno. Un problema al que se enfrentan las plataformas *offshore* son los daños estructurales, por lo cual la monitorización de la integridad estructural es imprescindible. En este trabajo, se establece una metodología para el diagnóstico y localización de un daño estructural en plataformas con estructuras tipo *jacket*. El enfoque propuesto se valida experimentalmente dentro del laboratorio CoDALab de la Universidad Politècnica de Catalunya en una plataforma a escala.

Palabras Clave—monitorización; integridad estructural; estructura tipo *jacket*; acelerómetros; aprendizaje supervisado; máquina de soporte vectorial, análisis de componentes principales; remodelación de datos

I. INTRODUCCIÓN

En los parques eólicos *offshore*, las turbinas eólicas también conocidas como aerogeneradores se erigen con diferentes tipos de soportes, dependiendo de la profundidad a la que se instalará el aerogenerador. A profundidades inferiores a 15 metros se utilizan monopilotes, que son estructuras bastante simples ancladas directamente al fondo marino. Para aerogeneradores colocados a profundidades menores o iguales a 30 metros, generalmente se usa el sistema de soportes por gravedad, que implica el uso de una gran plataforma de hormigón o acero con un diámetro y un peso aproximados de 15 metros y 1,000 toneladas respectivamente. Finalmente, a profundidades de más de 30 metros, las compañías de instalación usan soportes tipo *jacket*, éstos son estructuras que presentan tres o cuatro puntos de anclaje al fondo marino, lo que aumenta los niveles de seguridad al anclar las torres como se muestra en la Fig. 1. La energía eólica *offshore* tiene un potencial enorme, no obstante, solo puede explotarse disminuyendo los costos de

operación y mantenimiento. Por lo que la monitorización de la integridad estructural para proporcionar una alerta temprana de daños son esenciales para lograr este objetivo. Éste documento se enfoca en el problema de la detección de daños para soportes con estructuras tipo *jacket*.

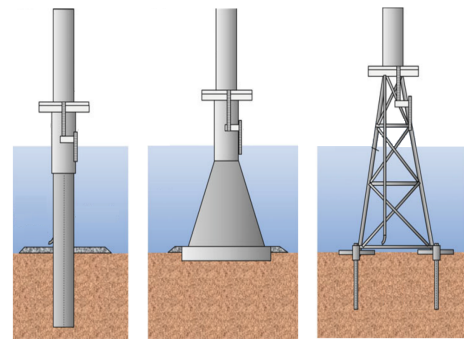


Fig. 1: Soportes para aerogeneradores *offshore* [1]. De izquierda a derecha monopilote, de gravedad y *jacket*.

En la literatura se pueden encontrar muchas metodologías para la detección de daños, entre las más prolíficas se encuentran las que utilizan métodos basados en vibración [2]. El interés en estos métodos ha estado creciendo en los últimos años debido a su simplicidad, facilidad de uso y alta efectividad [3]. Sin embargo, la mayoría de los estudios se centran principalmente en el caso de señales de respuesta de vibración y excitación de entrada medibles con solo unos pocos estudios recientes [4] centrados en solo en la respuesta a la vibración, cuya importancia se deriva del hecho de que en algunas aplicaciones la excitación no se puede imponer y a menudo, no se puede medir. Durante las décadas de 1970 y 1980,

la industria mostró interés en los métodos de identificación de daños basados en vibraciones para plataformas *offshore*, pero debido a muchos problemas prácticos estos esfuerzos se abandonaron a principios de la década de 1990 [5]. En los últimos años, algunos investigadores han discutido sobre la detección de daños en fundaciones en alta mar. Un trabajo a destacar es el realizado por Weijtjens et al. [6] relacionado con el monitoreo de un aerogenerador *offshore* real con monopilote basado en sus frecuencias de resonancia, donde los problemas clave son la variabilidad operativa y ambiental de las frecuencias de resonancia de la turbina que pueden ocultar cualquier cambio estructural. Finalmente, en [7] se utiliza un banco de pruebas experimental similar al indicado en este trabajo, la detección de daños se realiza (pero no la localización o clasificación) por medio del indicador de daño estimado de matriz de covarianza.

Este documento aporta un método de detección y localización de daños (este último se trata como un problema de clasificación) para un modelo a escala de un aerogenerador *offshore* con estructura tipo *jacket* utilizando solo datos de respuesta de aceleración. De acuerdo a [8], la excitación disponible es el viento, por lo que la excitación de entrada no es medible. En este trabajo la excitación de la vibración es dada por el viento (no se puede imponer y se supone que es desconocida).

La metodología aportada comprende los siguientes pasos:

- Simulación del viento como un ruido blanco gaussiano sobre el aerogenerador.
- Recopilación de datos. Se adquieren los datos procedentes de los acelerómetros colocados en la plataforma del aerogenerador y se crea una base de datos.
- Remodelación de los datos. Los datos en crudo se preprocesan usando una remodelación de la información (para aumentar la cantidad de información contenida en cada observación/muestra).
- Estandarización de los datos y el escalado de columnas. Esto se realiza con el fin de simplificar el cálculo de los componentes principales.
- Extracción de características. La técnica PCA (Análisis de componentes principales por sus siglas en inglés) se utiliza para extraer características, así como para reducir la dimensionalidad de los datos y el tiempo de cálculo.
- Clasificación. Se prueban los clasificadores vecino más cercano k (k -NN) y máquina de soporte vectorial (SVM) con *kernel* cuadrático. En este punto se incluye la evaluación del modelo de clasificación para sintonizar los parámetros requeridos por cada modelo de clasificación.

La fiabilidad del método propuesto se verifica de manera experimental utilizando un banco de pruebas explicado a detalle en secciones posteriores.

La estructura del documento se describe a continuación: la Sección II presenta el banco de pruebas de laboratorio utilizado para validar el enfoque propuesto. La Sección III detalla la metodología de detección y clasificación de daños. En la Sección IV se muestran los resultados obtenidos y finalmente las conclusiones son mostradas en la Sección V.

II. BANCO DE PRUEBAS DE LABORATORIO

La Fig.2 muestra la descripción del experimento, el cual se explica de la siguiente manera:

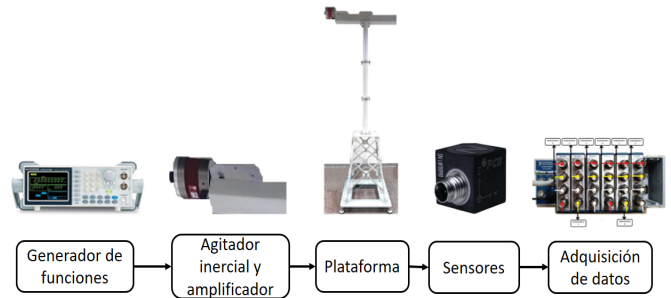


Fig. 2: Descripción general del banco de pruebas experimentales.

El experimento se inicia con una señal de ruido blanco dada por el generador de funciones. En él se simulon diferentes velocidades del viento multiplicando la amplitud de la señal de ruido blanco por los factores 0.5, 1, 2 y 3 de esta manera se crea la vibración necesaria para excitar la estructura y generar vibraciones (similares a las producidas por ráfagas de viento en las palas) a la plataforma en el laboratorio. La señal de ruido blanco emitida por el generador de funciones se amplifica y esta señal eléctrica se aplica al agitador. El agitador se coloca en la parte superior de la estructura, simulando así la masa de la góndola y los efectos ambientales del viento sobre toda la plataforma.

La plataforma a escala utilizada en este trabajo se muestra en la Fig. 3 (izquierda) y tiene tres componentes estructurales diferentes: góndola, torre y el soporte tipo *jacket*.

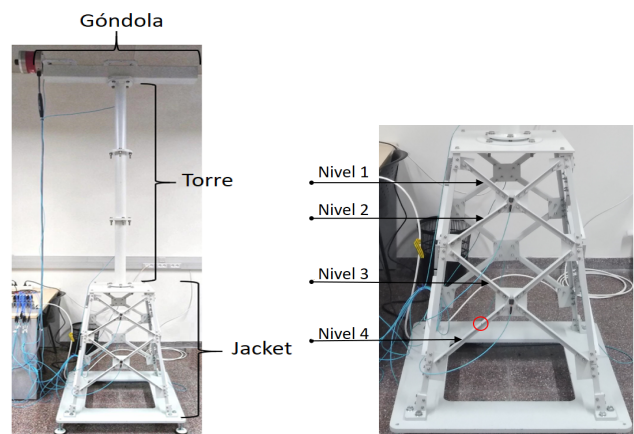


Fig. 3: Izquierda: Modelo a escala de la plataforma de un aerogenerador *offshore* con estructura tipo *jacket* utilizada en las pruebas experimentales). Derecha: Niveles de la estructura *jacket* donde se introduce el daño por grieta (círculo rojo).

La estructura tipo *jacket* tiene cuatro longitudes de barra diferentes, cada una en diferentes niveles. El daño (grieta de 5 mm) se introducirá uno a la vez en los cuatro niveles diferentes como se ilustra en la Fig.3 (derecha). Se ha demostrado que las grietas son uno de los principales tipos de daños encontrados en las plataformas de aerogeneradores *offshore* [9].

Finalmente, la plataforma es monitoreada por 8 acelerómetros triaxiales (PCB Piezotronics, modelo 356A17) que están conectados al sistema de adquisición de datos. Los acelerómetros han sido ubicados estratégicamente como se muestra en la Fig. 4. Al tener los acelerómetros una salida para cada componente espacial x , y y z , se obtienen datos de 24 señales de salida.

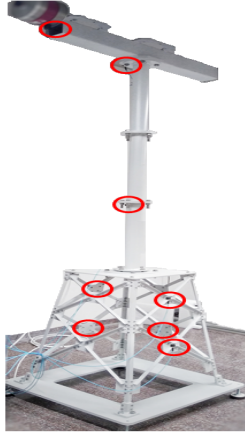


Fig. 4: Ubicación de los sensores en la plataforma.

III. METODOLOGÍA DE DETECCIÓN Y CLASIFICACIÓN DE DAÑOS

Para este trabajo fue necesario obtener un conjunto de datos sobre el cual se pudiera trabajar y que permitiera entrenar el algoritmo de aprendizaje supervisado.

III-A. Recopilación de datos

En el banco de pruebas de laboratorio se han considerado 5 estados estructurales diferentes en las barras de la estructura tipo *jacket*: estado saludable, daño en nivel 1, daño en nivel 2, daño en nivel 3 y daño en nivel 4. Para cada uno de estos casos se hicieron pruebas con las 4 amplitudes de ruido blanco: 0.5, 1, 2 y 3 obteniendo un total de 140 experimentos/pruebas cada uno de duración 60 segundos como se muestra en la Tabla I.

Estado estructural	Amplitud				Total de pruebas
	0.5	1	2	3	
1. Estado saludable	15	15	15	15	60
2. Daño en nivel 1	5	5	5	5	20
3. Daño en nivel 2	5	5	5	5	20
4. Daño en nivel 3	5	5	5	5	20
5. Daño en nivel 4	5	5	5	5	20
	35	35	35	35	140

Tabla I: Pruebas realizadas para cada caso y amplitud.

Para los sensores se ha considerado una frecuencia de muestreo de 275 Hz, factible en un entorno *offshore* [10]. De este modo, los datos en crudo se almacenan en una matriz \mathbf{Z} , la cual para cada prueba almacena 16,517 muestras de cada una de las mediciones de los 24 sensores como en (1).

$$\mathbf{Z}^{(p)} = \begin{pmatrix} z_{1,1}^{(p)} & z_{1,2}^{(p)} & \cdots & z_{1,24}^{(p)} \\ z_{2,1}^{(p)} & z_{2,2}^{(p)} & \cdots & z_{2,24}^{(p)} \\ z_{3,1}^{(p)} & z_{3,2}^{(p)} & \cdots & z_{3,24}^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ z_{16517,1}^{(p)} & z_{16517,2}^{(p)} & \cdots & z_{16517,24}^{(p)} \end{pmatrix} \quad (1)$$

donde $p = 1, 2, \dots, 140$ es el número de prueba. Al tener un total de 140 pruebas, se obtiene (2) una matriz \mathbf{Z} de datos en crudo con una dimensión total de $2,312,380 \times 24$.

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z}^{(1)} \\ \vdots \\ \mathbf{Z}^{(p)} \\ \vdots \\ \mathbf{Z}^{(140)} \end{pmatrix} \quad (2)$$

III-B. Remodelación de los datos

Con el objetivo de incrementar la cantidad de información contenida en cada muestra (fila), se hace la remodelación de los datos de la matriz \mathbf{Z} .

Dada la naturaleza tridimensional de la información recopilada en este documento (tiempo, sensores, experimentos), la remodelación se realizó en base al desarrollo propuesto por Westerhuis et al. [11]. Donde ahora nuestra información estará representada por una matriz bidimensional $\mathbf{X} = (x_{i,j}^{k,l}) \in \mathcal{M}_{(n_1+\dots+n_5) \times (24 \cdot 199)}$ de la siguiente manera:

En la matriz \mathbf{X} mostrada en (3), los dos subíndices $i = 1, \dots, n_j$ y $j = 1, \dots, J$ y los dos superíndices $k = 1, \dots, K$ y $l = 1, \dots, L$ están relacionados con la prueba experimental, el estado estructural, el sensor y el instante de tiempo, respectivamente. Cabe mencionar que J es la cantidad de diferentes estados estructurales; K es el número total de sensores y L es el número de marcas de tiempo por experimento.

De acuerdo a la información obtenida en la Sección III-A hemos considerado:

- $n_1 + n_2 + n_3 + n_4 + n_5 = 11,620$ muestras.
 - $n_1 = 4,980$
 - $n_2 = 1,660$
 - $n_3 = 1,660$
 - $n_4 = 1,660$
 - $n_5 = 1,660$
- $J = 5$ estados estructurales.
- $K = 24$ sensores.
- $L = 199$ instantes de tiempo

Por tanto al remodelar nuestros datos, finalmente obtenemos la matriz \mathbf{X} de datos en crudo con dimensiones $11,620 \times 4,776$.

De esta manera se ha recopilado la información relacionada con las mediciones del sensor y sus variaciones a lo largo del tiempo.

$$\begin{aligned}
 \mathbf{X} &= \begin{pmatrix} x_{i,j}^{k,l} \end{pmatrix} \\
 &= \begin{bmatrix} x_{1,1}^{1,1} & \cdots & x_{1,1}^{1,199} & x_{1,1}^{2,1} & \cdots & x_{1,1}^{2,199} & \cdots & x_{1,1}^{24,1} & \cdots & x_{1,1}^{24,199} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{4980,1}^{1,1} & \cdots & x_{4980,1}^{1,199} & x_{r_{11},1}^{2,1} & \cdots & x_{r_{11},1}^{2,199} & \cdots & x_{4980,1}^{24,1} & \cdots & x_{4980,1}^{24,199} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{1660,2}^{1,1} & \cdots & x_{1660,2}^{1,199} & x_{1660,2}^{2,1} & \cdots & x_{1660,2}^{2,199} & \cdots & x_{1660,2}^{24,1} & \cdots & x_{1660,2}^{24,199} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{1,5}^{1,1} & \cdots & x_{1,5}^{1,199} & x_{1,5}^{2,1} & \cdots & x_{1,5}^{2,199} & \cdots & x_{1,5}^{24,1} & \cdots & x_{1,5}^{24,199} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{1660,5}^{1,1} & \cdots & x_{1660,5}^{1,199} & x_{1660,5}^{2,1} & \cdots & x_{1660,5}^{2,199} & \cdots & x_{1660,5}^{24,1} & \cdots & x_{1660,5}^{24,199} \end{bmatrix} \quad (3) \\
 &= \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_5 \end{bmatrix} = \left[\mathbf{X}^1 \mid \mathbf{X}^2 \mid \cdots \mid \mathbf{X}^{24} \right].
 \end{aligned}$$

III-C. Estandarización

Los datos en crudo, o sea, los datos sin procesar se deben estandarizar ya que PCA es sensible a la magnitud de las variables además que simplificará el cálculo empírico de la matriz de covarianza necesario para la descomposición de PCA.

En este trabajo, se utiliza la estandarización en columnas, de esta manera cada columna de la nueva matriz estandarizada $\tilde{\mathbf{X}}$, tiene una media de cero y una desviación estándar de uno [12].

III-D. PCA (Análisis de Componentes Principales)

El método PCA genera un nuevo conjunto de variables conocidas como componentes principales (CP).

La matriz de transformación se calcula como una multiplicación de matriz a matriz.

$$\mathbf{T} = \tilde{\mathbf{X}}\mathbf{P},$$

donde \mathbf{T} es una matriz de $11,620 \times 4,776$ y \mathbf{P} es la matriz que contiene escritos como columnas las componentes principales de la matriz $\tilde{\mathbf{X}}$.

La reducción de dimensionalidad se realiza a través de un modelo PCA reducido mostrado en (4), donde \mathbf{P}_ℓ tiene las primeras CP ℓ y \mathbf{T}_ℓ es la proyección de la matriz escalada $\tilde{\mathbf{X}}$.

$$\mathbf{T}_\ell = \tilde{\mathbf{X}}\mathbf{P}_\ell. \quad (4)$$

Al aplicar la estandarización de columnas, la matriz de varianza es igual a $K \cdot L$. Esto significa que las primeras CP ℓ conservan una proporción de varianza-covarianza dada por:

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_\ell}{\lambda_1 + \lambda_2 + \cdots + \lambda_{4,776}},$$

donde $\lambda_1 + \lambda_2 + \cdots + \lambda_{4,776}$ son los valores propios asociados con las componentes principales de la matriz de varianza-covarianza, en orden decreciente.

En particular, en este documento se utilizan las primeras 443, 887 y 1770 componentes principales de la descomposición de PCA ya que representan 85%, 90% y 95% de la varianza, esto se puede observar en la Fig. 5.

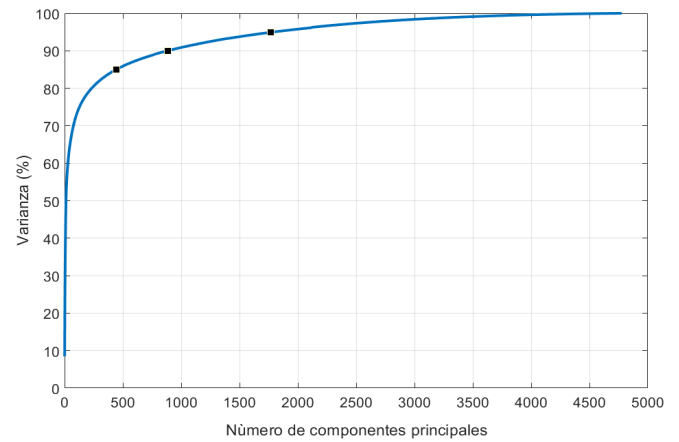


Fig. 5: Gráfica de relación de las CP con respecto al porcentaje de varianza.

III-E. Clasificación

Para poder diagnosticar y localizar un daño estructural, en este documento se prueban los algoritmos de clasificación de aprendizaje supervisado: k -NN y SVM. Después el rendimiento de cada uno se compara a través de diferentes métricas de evaluación.

El clasificador k -NN almacena los datos de entrenamiento en la memoria en lugar de construir un modelo y compara los nuevos datos de prueba con las instancias guardadas más cercanas para realizar la predicción.

La clasificación SVM es una técnica de clasificación binaria que debe adaptarse para hacer frente a problemas de clasificación múltiple. De manera específica se utilizó un clasificador SVM con *kernel* cuadrático representado en (5). Esta elección se realizó de acuerdo a lo observado en los diagramas de dispersión de la Fig. 6 donde se identifica una relación cuadrática.

$$K(x_i, x_j) = \left(1 + \frac{1}{\rho^2} x_i^\top x_j \right)^2 \quad (5)$$

donde ρ es el parámetro *kernel scale* y x_i y x_j denotan las diferentes muestras de nuestro conjunto de datos.

III-F. Validación cruzada κ -Fold

La validación cruzada es una técnica utilizada para evaluar los resultados y garantizar que sean independientes de la partición entre los datos de entrenamiento y prueba. En este documento se utiliza la validación cruzada de $\kappa = 5$.

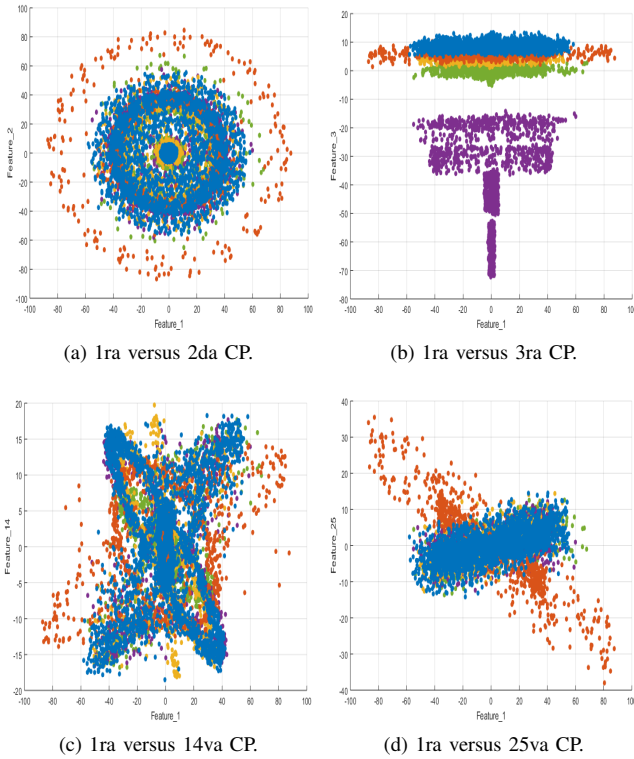


Fig. 6: Diagrama de dispersión de las muestras de acuerdo a los diferentes estados estructurales. Sanas (puntos azules), daño en nivel 1 (puntos naranjas), daño en nivel 2 (puntos amarillos), daño en nivel 3 (puntos morados) y daño en nivel 4 (puntos verdes).

III-G. Métricas para la evaluación de modelos de clasificación

En problemas de clasificación, para saber si el modelo que hemos entrenado tiene el mejor rendimiento de acuerdo con el problema presentado, es importante evaluar el modelo de clasificación a través de métricas. Las métricas que se han empleado son la exactitud, precisión, sensibilidad, F1-*score* y especificidad. Como se muestra en [13], éstas son fáciles de calcular y aplicables a problemas de clasificación binaria y múltiple.

De acuerdo a [14] y [15], cuando el problema de clasificación tiene varias clases el resultado es el promedio obtenido al sumar el resultado de cada clase y dividir el número total de clases. Las fórmulas para calcular estas métricas en un modelo de clasificación de varias clases se muestran en la Tabla II.

Donde TP_j es el número de casos positivos que fueron identificados correctamente de cada clase $j = 1, \dots, 5$; TN_j es la suma de los casos negativos que fueron clasificados correctamente; FP_j es la cantidad de casos negativos que fueron clasificados incorrectamente como positivos; FN_j es el total de casos positivos que fueron clasificados incorrectamente como negativos. Estos datos son obtenidos a partir de una matriz de confusión multiclase.

Métrica	Formula
Exactitud promedio (\overline{acc})	$\frac{1}{5} \sum_{j=1}^5 \frac{TP_j + TN_j}{TP_j + FP_j + FN_j + TN_j}$
Precisión promedio (\overline{ppv})	$\frac{1}{5} \sum_{j=1}^5 \frac{TP_j}{TP_j + FP_j}$
Sensibilidad promedio (\overline{tpr})	$\frac{1}{5} \sum_{j=1}^5 \frac{TP_j}{TP_j + FN_j}$
F1-Score promedio ($\overline{F1}$)	$2 \frac{\overline{ppv} \times \overline{tpr}}{\overline{ppv} + \overline{tpr}}$
Especificidad promedio (\overline{tnr})	$\frac{1}{5} \sum_{j=1}^5 \frac{TN_j}{TN_j + FP_j}$

Tabla II: Métricas para evaluar modelos de clasificación de varias clases, donde j se refiere a la clase individual y 5 es el número total de clases.

IV. RESULTADOS

Los mejores modelos obtenidos de los métodos de clasificación k -NN y SVM, se detallan en esta sección.

IV-A. Resultados para el método de clasificación k -NN

Primero, se prueba el clasificador k -NN. Los indicadores introducidos en la Sección III-G se emplean para ajustar el valor del parámetro k , es decir, el número de vecinos que se utilizarán y decidir la mejor variación que se adoptará al aplicar PCA.

Varianza	k	\overline{acc}	\overline{ppv}	\overline{tpr}	$\overline{F1}$	\overline{tnr}
85 % (443 CP)	1	0.857	0.860	0.846	0.853	0.950
	5	0.856	0.869	0.82	0.844	0.953
	10	0.867	0.891	0.827	0.858	0.956
	25	0.868	0.898	0.828	0.862	0.955
	50	0.874	0.912	0.831	0.87	0.958
	100	0.911	0.928	0.879	0.903	0.981
	150	0.933	0.942	0.908	0.925	0.988
	200	0.946	0.947	0.924	0.936	0.993
	250	0.940	0.938	0.917	0.927	0.991
	300	0.930	0.929	0.903	0.915	0.988
500	0.899	0.909	0.859	0.884	0.976	
90 % (887 CP)	1	0.857	0.856	0.845	0.85	0.961
	5	0.860	0.870	0.828	0.849	0.956
	10	0.872	0.887	0.838	0.862	0.961
	25	0.873	0.889	0.840	0.864	0.961
	50	0.884	0.913	0.846	0.878	0.964
	100	0.917	0.930	0.886	0.907	0.984
	150	0.938	0.945	0.914	0.929	0.990
	200	0.950	0.950	0.931	0.940	0.995
	250	0.944	0.941	0.922	0.932	0.993
	300	0.933	0.931	0.906	0.918	0.989
500	0.903	0.912	0.864	0.888	0.977	
95 % (1770 CP)	1	0.854	0.853	0.842	0.847	0.960
	5	0.858	0.865	0.829	0.847	0.956
	10	0.872	0.885	0.841	0.862	0.962
	25	0.873	0.888	0.841	0.864	0.962
	50	0.885	0.911	0.848	0.879	0.965
	100	0.918	0.930	0.888	0.908	0.985
	150	0.938	0.945	0.915	0.930	0.990
	200	0.950	0.950	0.930	0.940	0.995
	250	0.945	0.942	0.923	0.932	0.993
	300	0.934	0.931	0.908	0.919	0.989
500	0.904	0.913	0.866	0.889	0.978	

Tabla III: Indicadores de evaluación para el método k -NN usando diferentes porcentajes de varianza y diferente número de vecinos más cercanos (k). Los mejores clasificadores para cada variación se resaltan en negrita.

La Tabla III muestra los resultados obtenidos al trabajar con 85 %, 90 % y 95 % de la varianza, e ir variando el número

de vecinos k . Los clasificadores con el mejor rendimiento se resaltan en negrita. Se puede observar que los mejores clasificadores son 90 % y 95 % de la varianza y una $k = 200$. Se decide trabajar con 90 % de varianza ya que reducirá el requisito de memoria computacional, lo cual se comprueba en la Tabla IV.

Varianza	Exactitud	Velocidad de predicción (obs/seg)	Tiempo de entrenamiento (seg)
85 %	94.6 %	210	329
90 %	95.0 %	110	638
95 %	95.0 %	55	1251

Tabla IV: Resultados de la velocidad de predicción y el tiempo de entrenamiento para los mejores clasificadores de rendimiento de cada variación (85 %, 90 % y 95 %) utilizando el método de clasificación k -NN en una computadora Intel Core i7 de 3 GB y 16 GB de RAM.

Retomando la Tabla III, en particular para este clasificador con el 90 % de varianza, podemos observar que inicialmente, al aumentar el número de vecinos los valores de todas la métricas tienden a mejorar, sin embargo, cuando $k > 200$ el rendimiento de todas las métricas decrece.

En la matriz de confusión mostrada en la Tabla V, cada fila representa las clases verdaderas mientras que cada columna representa las clases predichas (por el clasificador). En particular, la primera fila (y la primera columna) están etiquetadas como 0 y corresponden al estado saludable. Las siguientes etiquetas (para filas y columnas) 1, 2, 3 y 4 corresponden al daño en los niveles correspondientes de la estructura tipo *jacket*. La exactitud que se obtiene es de 95 %, mientras que la precisión también es de 95 %, lo cual es el porcentaje que este modelo ha sido capaz de clasificar correctamente.

		Clase predicha				
		0	1	2	3	4
Clase verdadera	0	> 99 %			< 1 %	
	1	< 1 %	98 %	2 %		
	2	7 %		93 %		< 1 %
	3				75 %	25 %
	4	1 %		< 1 %		99 %

Tabla V: Matriz de confusión para el algoritmo k -NN con $k = 200$ vecinos. La etiqueta 0 corresponde al estado saludable y las etiquetas 1, 2, 3 y 4 corresponden al estado de daño ubicado en el nivel correspondiente. Un cuadrado en blanco significa 0 %.

IV-B. Resultados para el método de clasificación SVM

Dado que los diagramas de dispersión en la Fig. 6 revelan una relación cuadrática (particularmente la primera frente a la segunda componente principal), se probó el clasificador SVM cuadrático. En este caso los parámetros a sintonizar son C (*box constrain*) y ρ (*kernel scale*), que se ajustan usando las métricas detalladas en la Subsección III-G.

La Tabla VI resume los resultados obtenidos al trabajar con 85 %, 90 % y 95 % de la varianza. Dado que el problema que estamos tratando parece un problema separable, no se

Varianza	C	ρ	\bar{acc}	\bar{ppv}	\bar{tpr}	$\bar{f1}$	\bar{ttr}
85 % (443 CP)	1	5	0.987	0.993	0.983	0.988	0.995
		20	0.994	0.991	0.991	0.991	1.000
		30	0.995	0.994	0.994	0.994	1.000
		40	0.996	0.995	0.995	0.995	1.000
		50	0.997	0.996	0.996	0.996	1.000
		60	0.998	0.997	0.997	0.997	1.000
		70	0.998	0.998	0.998	0.998	1.000
		80	0.999	0.998	0.998	0.998	1.000
		90	0.999	0.999	0.999	0.999	1.000
		100	0.999	0.999	0.999	0.999	1.000
		150	0.998	0.998	0.998	0.998	1.000
		200	0.997	0.996	0.997	0.997	0.999
300	0.882	0.926	0.835	0.878	0.954		
90 % (887 CP)	1	5	0.987	0.993	0.982	0.988	0.995
		20	0.991	0.988	0.988	0.988	1.000
		30	0.995	0.993	0.993	0.993	1.000
		40	0.996	0.995	0.995	0.995	1.000
		50	0.997	0.997	0.997	0.997	1.000
		60	0.998	0.997	0.997	0.997	1.000
		70	0.998	0.997	0.997	0.997	1.000
		80	0.998	0.997	0.997	0.997	1.000
		90	0.998	0.998	0.998	0.998	1.000
		100	0.998	0.998	0.998	0.998	1.000
		150	0.998	0.998	0.998	0.998	0.999
		200	0.997	0.996	0.996	0.996	0.999
300	0.883	0.927	0.837	0.880	0.955		
95 % (1770 CP)	1	5	0.986	0.993	0.981	0.987	0.994
		20	0.990	0.987	0.986	0.986	1.000
		30	0.994	0.992	0.992	0.992	1.000
		40	0.996	0.994	0.994	0.994	1.000
		50	0.997	0.996	0.996	0.996	1.000
		60	0.997	0.997	0.997	0.997	1.000
		70	0.998	0.997	0.997	0.997	1.000
		80	0.998	0.997	0.998	0.998	1.000
		90	0.998	0.998	0.998	0.998	1.000
		100	0.998	0.997	0.998	0.998	1.000
		150	0.998	0.997	0.998	0.997	0.999
		200	0.996	0.996	0.996	0.996	0.999
300	0.998	0.997	0.997	0.997	1.000		

Tabla VI: Indicadores de evaluación para el modelo SVM usando diferentes porcentajes de varianza y diferentes valores para el parámetro ρ (*Kernel scale*). Como es un problema separable, tomamos el valor del parámetro C como 1. Los mejores valores para cada indicador están resaltados en negrita.

encontraron cambios en el desempeño de los indicadores al tomar diferentes valores de C . Por consiguiente se consideró $C = 1$. Los mejores casos son los que presentan 85 % de la varianza con $\rho = 90$ y $\rho = 100$.

De acuerdo con la Tabla VII, se observa que el clasificador con 85 % de la varianza y $\rho = 90$ tiene una mayor velocidad de predicción y un tiempo de entrenamiento más corto. De manera que este clasificador tiene el mejor rendimiento.

Varianza	Kernel	Exactitud	Velocidad de predicción (obs/seg)	Tiempo de entrenamiento (seg)
	scale ρ			
85 %	90	99.9 %	1700	76
	100	99.9 %	1600	79
90 %	90	99.8 %	320	256
	100	99.8 %	580	364
95 %	90	99.8 %	100	676

Tabla VII: Resultados de velocidad de predicción y tiempo de entrenamiento para los mejores casos de rendimiento de cada variación (85 %, 90 % y 95 %) utilizando el método de clasificación SVM.

Una vez determinado el mejor clasificador, observamos en

la Tabla VI el rendimiento de las métricas para el caso con 85 % de la varianza, el incrementar el parámetro ρ mejora el rendimiento general del método de clasificación hasta cierto punto. La degradación del rendimiento aparece para valores superiores a $\rho = 100$.

La matriz de confusión para el mejor clasificador (85 % varianza y $\rho = 90$) se representa en la Tabla VIII. De esta matriz de confusión y de acuerdo con las métricas de evaluación, obtenemos una exactitud promedio de 99.9 %, una precisión promedio de 99.9 % y una especificidad promedio de 100 %.

		Clase predicha				
		0	1	2	3	4
Clase verdadera	0	> 99 %	< 1 %	< 1 %		
	1	< 1 %	99 %	< 1 %		
	2		< 1 %	99 %		
	3				> 99 %	< 1 %
	4					100 %

Tabla VIII: Matriz de confusión del modelo SVM con una varianza del 85 %, $\rho = 90$ y $C = 1$. La exactitud obtenida es de 99.9 %. Esta matriz se encuentra etiquetada de la siguiente manera: etiqueta 0 corresponde a saludable, etiqueta 1 corresponde al daño en el nivel 1, etiqueta 2 daño en nivel 2, etiqueta 3 daño en nivel 3 y etiqueta 4 daño en nivel 4. Un cuadrado vacío significa 0 %.

V. CONCLUSIONES

El hecho de probar experimentalmente una metodología basada en la respuesta a la vibración para la monitorización de la integridad estructural, hace viable su aplicación a plataformas *offshore* de tamaño real, donde la excitación obviamente no puede imponerse y a menudo no es medible.

En cuanto a la contribución de este trabajo, se puede destacar la forma como se recopilan, organizan, escalan, transforman y reducen las dimensiones de los datos tridimensionales (procedentes de diferentes tiempos, sensores y experimentos) lo cual nos va a dar la oportunidad de que una muestra contenga mas información en comparación a la información que contienen las muestras a las que no se les realiza este tratamiento. En este documento, esta metodología se especializó para una aplicación específica, la cual se explica en III-A.

En este caso para realizar la detección y localización de daños fue necesario seleccionar un método de clasificación de aprendizaje automático supervisado. Se observó que la selección dependerá en gran parte del tipo de problema que al que nos enfrentemos y de como se encuentren agrupados los datos a tratar como en III-E.

De los dos métodos de clasificación de aprendizaje supervisado que fueron probados, el clasificador SVM cuadrático con $\rho = 90$, $C = 1$ y 443 componentes principales (85 % de varianza) tiene un rendimiento muy cercano al ideal, logrando en todos los indicadores un resultado igual o superior a 99.99 % con una velocidad de predicción muy rápida (1700 obs / seg) y un tiempo de entrenamiento corto (76 seg).

REFERENCIAS

- [1] Klijnstra, J.; Zhang, X.; van der Putten, S.; Röckmann, C. Technical risks of offshore structures. In *Aquaculture Perspective of Multi-Use Sites in the Open Ocean*; Springer, Cham, 2017; pp. 115–127.
- [2] Fritzen, C.P. *Vibration-Based Techniques for Structural Health Monitoring*. Structural Health Monitoring 2006, pp. 45–224.
- [3] Goyal, D.; Pabla, B. The vibration monitoring methods and signal processing techniques for structural health monitoring: a review. *Archives of Computational Methods in Engineering* 2016, 23, 585–594.
- [4] Vamvoudakis-Stefanou, K.J.; Sakellariou, J.S.; Fassois, S.D. Output-only statistical time series methods for structural health monitoring: A comparative study. *Proceedings of the 7th European Workshop on Structural Health Monitoring*.
- [5] Farrar, C.R.; Worden, K. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 2006, 365, 303–315.
- [6] Weijtjens, W.; Verbelen, T.; De Sitter, G.; Devriendt, C. Foundation structural health monitoring of an offshore wind turbine—a full-scale case study. *Structural Health Monitoring* 2016, 15, 389–402.
- [7] Zugasti Uriguén, E. Design and validation of a methodology for wind energy structures health monitoring. PhD thesis, Universitat Politècnica de Catalunya, 2014.
- [8] Pozo, F.; Vidal, Y. Wind turbine fault detection through principal component analysis and statistical hypothesis testing. *Energies* 2016, 9, 3.
- [9] Stutzmann, J.; Ziegler, L.; Muskulus, M. Fatigue crack detection for lifetime extension of monopile-based offshore wind turbines. *Energy Procedia* 2017, 137, 143–151.
- [10] Lian, J.; Cai, O.; Dong, X.; Jiang, Q.; Zhao, Y. Health monitoring and safety evaluation of the offshore wind turbine structure: a review and discussion of future development. *Sustainability* 2019, 11, 494.
- [11] Westerhuis, J.A.; Kourti, T.; MacGregor, J.F. Comparing alternative approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics: A Journal of the Chemometrics Society* 1999, 13, 397–413.
- [12] Vidal, Y.; Pozo, F.; Tutivén, C. Wind turbine multi-fault detection and classification based on SCADA data. *Energies* 2018, 11, 3018.
- [13] Hossain, M.; Sulaiman, M. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process (IJDKP)* 2015, 5, 1–11.
- [14] Krüger, F. Activity, Context, and Plan Recognition with Computational Causal Behaviour Models. PhD thesis, University of Rostock, Mecklenburg, Germany, 2016.
- [15] Hameed, N.; Hameed, F.; Shabut, A.; Khan, S.; Cirstea, S.; Hossain, A. An Intelligent Computer-Aided Scheme for Classifying Multiple Skin Lesions. *Computers* 2019, 8, 62.

A Genetic Algorithm Applied to the Nesting of Cutting Patterns in Finite Materials

Anabel Rodríguez Rodríguez
 Centro de Investigaciones en Óptica A.C.
 León, Guanajuato, México C.P. 37150.
 Email: anabelrr@cio.mx

Francisco Javier Cuevas de la Rosa
 Centro de Investigaciones en Óptica A.C.
 León, Guanajuato, México C.P. 37150.
 Telephone: (477) 441-4200
 Email: fjcuevas@cio.mx

Abstract—The present work presents the implementation of a Genetic Algorithm (GA) in which coding for a variable size chromosome was proposed, containing the necessary fields to carry out the combination of patterns in the material. An objective function was designed with the purpose of minimizing waste in the material, taking into account the patterns and overlapping pixels that are not outside the valid accommodation area, while maximizing the area occupied by the patterns within the valid accommodation area. The initial population is randomly generated, and each generation evolves by applying the selection, crossing and mutation operators. Each individual is evaluated by means of the aptitude function, and the best result is saved to be compared in each generation so that the best solution can be returned.

I. INTRODUCTION

The nesting of cutting patterns in finite materials is a key issue to deal with in a significant number of manufacturing processes in the metal, paper, textile, footwear and glass industries, in which the arrangement of cutting patterns is indispensable to maximize the usable area and minimize waste through the use of Vision, Computational Intelligence and Digital Image Processing techniques [1].

There are different strategies to approach the solution of this optimization problem, as are heuristic and metaheuristic methods [2], [3]: simulated annealing [4], neural networks, particle swarm optimization [5], [6], genetic algorithms [7]–[10], and linear programming methods, but there is still no global method, given the complexity of the issue [11].

The nesting of cutting pattern is a constantly evolving issue, and recently, many researchers have developed models to solve it [12]–[28].

Some of the most recent works propose a single-pass heuristic, placing the patterns in one position of the material until its capacity is exhausted, and generating partial solutions until the best one is found [26]. Another one uses an evolutionary framework to build hyperheuristics that will solve the accommodation problem; such framework considers minimizing two objectives: the number of patterns to be accommodated and the time needed for this work. Also, it integrates three multi-objective evolutionary algorithms, each of which will yield a solution to a given problem, and then, the best solution at each step of the algorithm is chosen [25].

Although much research has been done to solve the problem in question, and industrial processes have been automated,

many companies still carry out the nesting of cutting pattern process manually; hence, optimal use of materials depends on the experience of the operator in charge of the process, which affects cutting time and waste of material.

Considering current research and all the methods and techniques that have been applied to solve the problem so far, and despite the fact that optimal solutions have been reached, implementation of solutions is not always simple. It takes a lot of time to come to an adequate solution, and the storage space needed to process the images is either very large or very costly.

With this in mind, the solution proposed in this research work allows one to provide a simple coding that does not require much storage for the processing of the images. In addition, the fundamental tool used was a GA, which is a metaheuristic technique that works with sets of large solutions, and in it, the values of the initial population are assigned randomly, and generations evolve applying the selection, crossing and mutation operators in each iteration. Each individual is assigned an aptitude from the objective function that is guiding the search until the best solution is found.

II. GENETIC ALGORITHMS

Genetic Algorithms (GAs) are one of the most popular techniques among evolutionary algorithms that are inspired by the Darwin's Theory of Evolution [8], based on the principle of natural selection. GAs were implemented by John Holland, a researcher at the University of Michigan, in the late 1960s [10].

GAs are stochastic search strategies based on the natural selection mechanism, and they involve aspects of natural genetics, imitating biological evolution as a search strategy to solve problems, and providing solutions to complex problems with a large number of parameters [7].

GAs differ from conventional search strategies in that they work on a set of potential solutions, called population. In turn, the population is made up of solutions that are called individuals, and each one is constituted by a chain of data (genes) that represent each variable associated with the optimization problem. In each iteration, the GA operators are applied to the population: selection, crossing and mutation, to obtain successive improved populations. In each cycle, the objective function (fitness) of each individual is calculated,

and compared in each iteration in order to obtain the solution with the best adaptation (Figure 1).

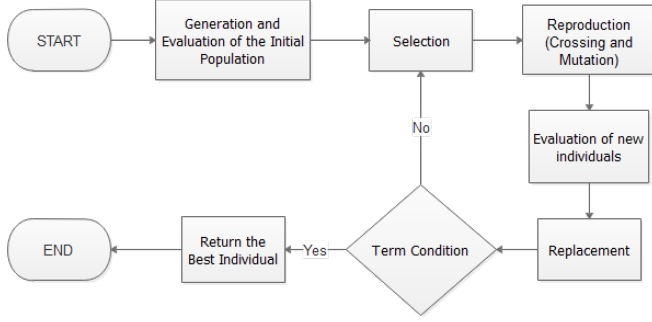


Fig. 1. Flow diagram of AG.

The procedure of a Simple Genetic Algorithm is shown in Algorithm 1.

Algorithm 1: Simple Genetic Algorithm

```

initialization;
Pop_Ini ← Initial Population
Pop_Aux ← Auxiliary Population
while Condition do
    for (0 to Size(Pop_Ini)/2) do
        select ← Selection(Pop_Ini);
        Crossing(select);
        Pop_Aux ← select;
    end
    Mutation(Pop_Aux);
    Pop_Ini ← Pop_Aux;
    Best_Solution = Better(Pop_Ini);
end
Result: Best_Solution
    
```

III. CHROMOSOME CODING

The algorithm generates the initial population randomly, obtaining a combination of patterns that will be accommodated. The size of the chromosome is variable because it is determined by the number of patterns involved in the arrangement. Each pattern is characterized by its identifier, its rotation angle, its the position in row and column in the accommodation area, and a flag that indicates whether or not the pattern will be taken into account to be accommodated (Figure 2).

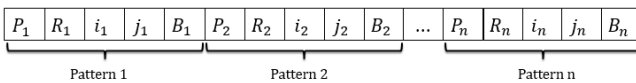


Fig. 2. Chromosome Coding.

Where n is the number of patterns that will be accommodated, P_k is the pattern identifier, R_k is the pattern rotation angle, i_k and j_k are the positions of the row and column in

the accommodation material respectively, B_k is the value that indicates whether or not the pattern is accommodated, and k lists the patterns from 1 to n .

IV. OBJECTIVE FUNCTION

Each chromosome is evaluated by the objective function, which is designed with the purpose of minimizing waste by avoiding overlaps between patterns, and by ensuring that there are no patterns outside the valid accommodation area, which leads to a better use of the area occupied by the patterns, and at the same time, minimize waste. Each individual is assigned a fitness value according to the objective function (Equation 1).

$$F = W_P \cdot (P_T - P_A) + W_w \cdot (A_T - A_{OC}) + W_{NO} \cdot Num_{Ov} + W_{NOA} \cdot Num_{OA} + W_{OP} \cdot Overlaps + W_{OA} \cdot Outside_Area \quad (1)$$

where W_P is the weight assigned to the number of patterns; $(P_T - P_A)$ is the number of patterns that are not accommodated, which refers to the difference between the number of patterns stored on the chromosome and the number of patterns that are accommodated, with a smaller difference meaning that there are more patterns in the arrangement; W_w is the weight assigned to waste; $(A_T - A_{OC})$ is the area of waste, which refers to the difference between the total area of material and the area occupied by the patterns; W_{NO} is the weight assigned to the number of overlapping patterns; Num_{Ov} is the number of overlapping patterns; W_{NOA} is the weight assigned to the number of patterns outside the accommodation area; Num_{OA} is the number of patterns outside the accommodation area; W_{OP} is the weight assigned to the number of overlapping pixels; $Overlaps$ is the number of overlapping pixels; W_{OA} is the weight assigned to the number of pixels outside the accommodation area; $Outside_Area$ is the number of pixels outside the accommodation area.

V. GENETIC OPERATORS

The genetic operators, selection, crossing and mutation, constitute the fundamental pillar of GAs to generate new populations and allow them to evolve with increasing iterations.

A. Selection Operator

Boltzman's selection method is one of the most used due to the satisfactory results it gives during the evolutionary process. It was proposed by Goldberg [29], and it uses a temperature variation function to control selection pressure. This method allows keeping selection pressure uniform at the beginning of the algorithm, while making it more elitist at the end; in other words, at the beginning, all individuals have virtually the same probability of selection, which benefits the least fit, which will have an equal opportunity to be selected as the best, thus allowing a scan over the entire search space. However, as the temperature drops, higher fitness will have a better probability of selection, allowing the evolutionary process to converge in the last iterations.

- *Boltzmann's Exponential Cooling*

An exponential function was designed in order to vary the temperature gradually in a soft and moderate fashion, as shown in Equation 2

$$T_i = T_{ini} \cdot \left(\sqrt[N]{\frac{T_{fin}}{T_{ini}}} \right)^{i \cdot \alpha}, \quad (2)$$

where T_i is the value of temperature in iteration i , T_{ini} and T_{fin} are the initial and final temperatures respectively, N is the number of iterations, and α is a factor that varies within the range $[0.5; 1]$ and that modifies the speed with which temperature decays. The behavior of this function is shown in Figure 3, taking the following values: $T_{ini} = 9000$, $T_{fin} = 2$, and $N = 3000$.

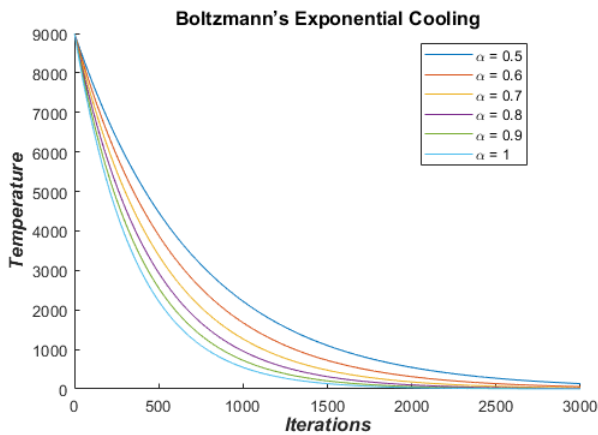


Fig. 3. Behavior of Boltzmann's exponential cooling.

B. Crossing Operator

Recombination is the main genetic operator. It represents sexual reproduction, and it operates on two or more chromosomes at the same time to generate its descendants, in which the characteristics of the parent chromosomes are combined.

- *Crossing at two points*

The crossing at two points takes place between two chromosomes selected at random; two crossing points are generated, and the genetic information of the parents is exchanged between both points, as shown in Figure 4.

This method is less destructive because it only changes part of the genes in the successors, allowing more information to be inherited from the parents to the new generation.

C. Mutation Operator

Mutation is a GA basic operator that provides a small portion of randomness in individuals in the population. As in the evolutionary process, characteristics that have not appeared in previous generations sometimes appear, which points to a change, often non-transcendental, between parents and children. Mutations occur with a very low probability, usually less than 1% of the total genes in a population.

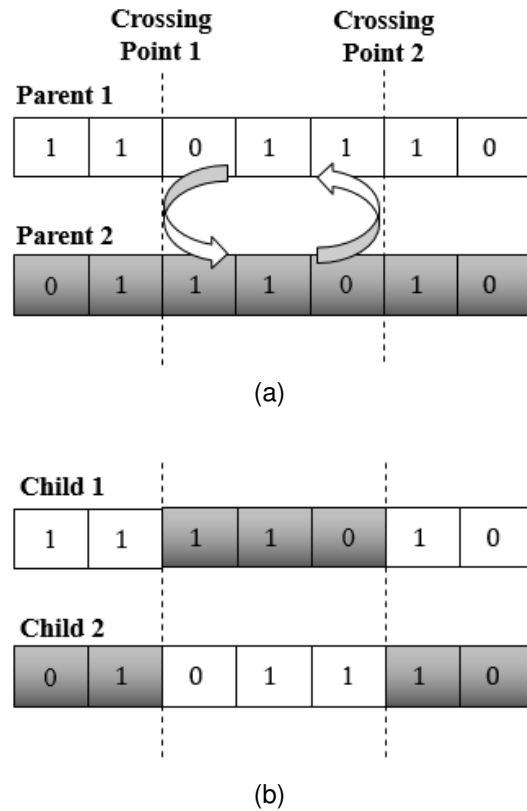


Fig. 4. Crossing at two points.
a) Parent's chromosomes before crossing,
b) Children's chromosomes after crossing

In GAs, mutation is used to maintain diversity in the population and to escape from local minimums, since allows reaching areas of the search space that were not covered by the individuals of the current population.

In this research, we implemented a logarithmic function that allows us to make a more effective mutation, depending on the probability of mutation, as shown in Equation 3.

$$t = 1 + \frac{\ln(1-u)}{\ln(1-P_m)}, \quad (3)$$

where t is the increase in the position that is going to mutate on the chromosome, u is a random number in the interval $[0; 1]$, and P_m is the mutation probability (0.001 in this work). An example of a single allele mutation is shown in Figure 5.

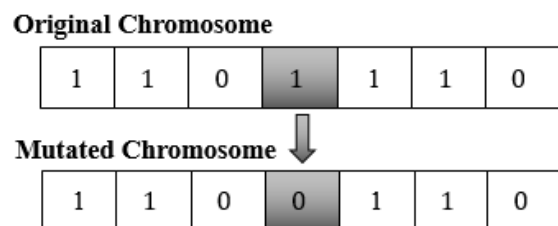


Fig. 5. Mutation of a single allele.

VI. IMPLEMENTATION

When the best solution is obtained at the end of the evolutionary process, it is possible that some patterns are overlapped or out of area, so a chromosome repair that can correct this problem is performed. For this purpose, a function has been designed that goes through the chromosome, and if any overlap is found, it changes the accommodation bit to 0, and so on, until all the patterns are gone, with the aim of removing the patterns that were badly accommodated. Finally, it tries to rearrange patterns in the accommodating areas that were not occupied. The pseudocode is shown in Algorithm 2.

Algorithm 2: Chromosome Repair Function

```

initialization;
Chromosome ← Best_Chromosome
Repaired ← Chromosome.clone()
for (n : 0 to Size(Chromosome)) do
    if (Repaired.Genes[n + 4] == 1) then
        overlapping, out_of_area ← false;
        if (Repaired.Overlapping > 0) then
            overlapping ← true;
        end
        if (Repaired.Out_of_Area > 0) then
            out_of_area ← true;
        end
    end
    if (overlapping || out_of_area) then
        Repaired.Genes[n + 4] == 0;
    end
end
Result: Repaired
    
```

A. Algorithm Generalization

In general, to solve the problem of nesting of cutting pattern, it is necessary to digitize the images, to carry out the evolutionary process (AG), and finally, to repair the best solution obtained by the AG. Figure 6 shows a flow chart of the whole process.

B. Initial Parameters

At the beginning of the algorithm, it is necessary to define some parameters such as population size, selection probability, crossing probability, mutation probability, maximum number of generations or stop criteria, which allow GA optimal running. Table I shows a description of each parameter.

VII. RESULTS

To obtain the results of this research, it was necessary to tune some parameters of the algorithm that allow adjusting the algorithm in order to achieve better convergence. The parameters are shown in Table II.

The limit parameters indicate that variables which do not reach the limit value will be penalized in the objective function. The weights in the objective function are those

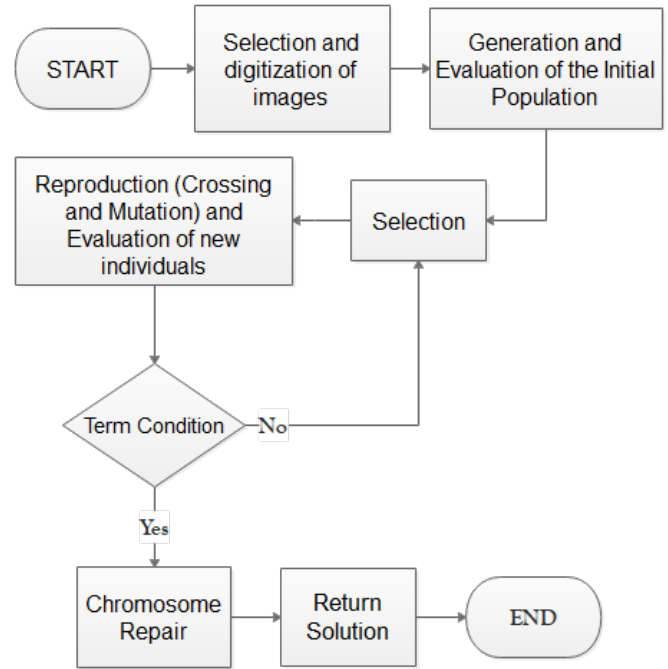


Fig. 6. Flow diagram for the solution of the nesting of cutting patterns.

 TABLE I
INITIAL PARAMETERS

Parameter	Meaning	Value
<i>Population_Size</i>	Number of individuals in the population	> 200
<i>Selection_Probability</i>	Percentage of selected individuals in the population	> 0.7
<i>Crossing_Probability</i>	Percentage of crossing individuals in the population	> 0.8
<i>Mutationd_Probability</i>	Percentage of mutated individuals in the population	< 0.5
<i>Stop_Criteria</i>	Number of GA iterations	> 100

that penalize individuals when they do not reach the limit corresponding to each variable.

It is recommended that the weights have a high value, since it allows differentiating between a good and a bad solution. As weights get larger, the value of the objective function will be higher, which means that it is a bad solution, and these solutions are less likely to be selected in the AG selection process.

Figure 7 shows one of the results obtained. The AG returns the best solution accommodating 384 patterns; once repair was performed, 317 patterns were accommodated, with 14% waste, after 100 iterations in 22 seconds.

Figure 8 shows another one of the results obtained. The AG returned a solution accommodating 435 patterns as the best one; after repair, 346 patterns were accommodated, with 12% waste, at the end of 200 iterations in 30 seconds.

Figure 9 shows GA result with irregular patterns. The best

TABLE II
 PARAMETER SENSITIVITY

Parameter	Meaning	Value
L_{OP}	Overlapping Patterns Limit	0
L_{OPi}	Overlapping Pixels Limit	0
L_{OAP}	Limit of Patterns Outside the Accommodation Area	0
L_{OAPi}	Limit of Pixels Outside the Accommodation Area	0
L_W	Limit of Material Waste	0
W_{NO}	Weight of Overlapping Patterns	> 500
W_{OP}	Weight of Overlapping Pixels	> 500
W_{NOA}	Weight of Patterns Outside the Accommodation Area	> 500
W_{OA}	Weight of Pixels Outside the Accommodation Area	> 500
W_W	Weight of Waste in Material	> 500

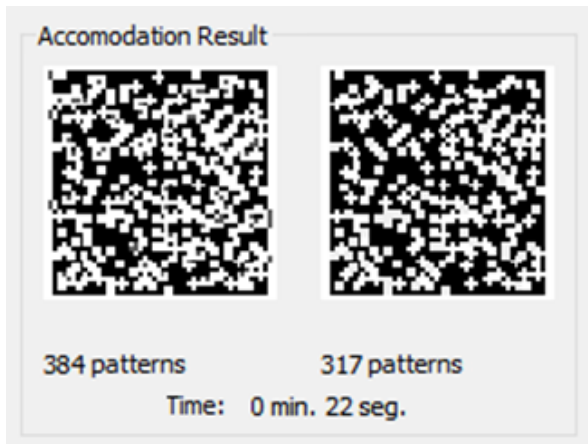


Fig. 7. GA result with regular patterns. Left image: best solution supplied by the AG. Right image: best solution after repair.

solution was obtained by arranging 73 patterns, with a waste of 20%, and performing 200 iterations in 42 seconds.

VIII. CONCLUSION

In this work, a GA is described to solve the problem of accommodating cutting patterns in finite materials; a solution in which it is not necessary to combine several metaheuristics to get a good result.

As an alternative, chromosome coding is proposed, which has proven to be robust and efficient according to the results obtained. The structure that defines the chromosome contains the necessary data to perform the accommodation in a simple way, since it allows performing the combinations of the cutting patterns in a random and diverse way at the beginning of the algorithm.

As a result of this research, it is demonstrated that the objective function is efficient, since it allows guiding the

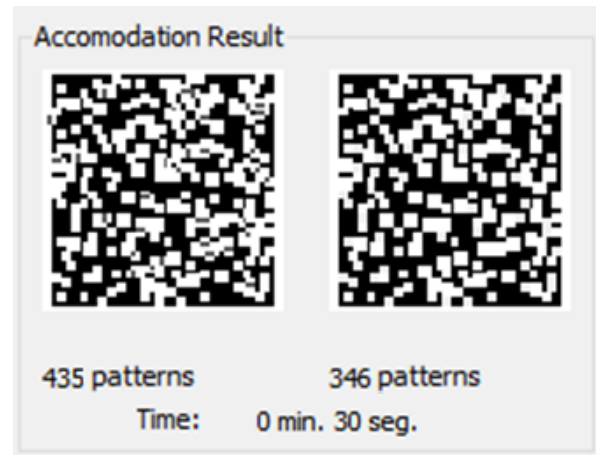


Fig. 8. GA result with regular patterns. Left image: best solution supplied by the AG. Right image: best solution after repair.



Fig. 9. GA result with irregular patterns. Best solution after repair.

search to find a solution that maximizes the use of the material while taking waste minimization into account.

ACKNOWLEDGMENT

The authors would like to thank Consejo Nacional de Ciencia y Tecnología (CONACYT) and Centro de Investigaciones en Óptica A.C. (CIO).

REFERENCES

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, New Jersey 07458: Pearson Education, Inc., 2008.
- [2] R. Martí, *Multi-Start Methods*. Boston, MA: Springer US, 2003, pp. 355–368. [Online]. Available: https://doi.org/10.1007/0-306-48056-5_12
- [3] C. R. Reeves, “Modern heuristic techniques for combinatorial problems,” C. R. Reeves, Ed. New York, NY, USA: John Wiley & Sons, Inc., 1993, ch. Genetic Algorithms, pp. 151–196. [Online]. Available: <http://dl.acm.org/citation.cfm?id=166648.166657>
- [4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, pp. 671–80, 1983.
- [5] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization,” *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, Jun 2007. [Online]. Available: <https://doi.org/10.1007/s11721-007-0002-0>
- [6] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Oct 1995, pp. 39–43.

- [7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [8] R. E. Leakey., *Charles Darwin, El origen de las especies. Versión Abreviada*, 1st ed. DF, México.: Martín Casillas Editores, S. A., 1980.
- [9] F. Cuevas, O. Gonzalez, Y. Susuki, D. Hernandez, M. Rocha, and N. Alcalá, "Genetic algorithms applied to optics and engineering," in *Fifth Symposium Optics in Industry*, E. Rosas, R. Cardoso, J. C. Bermudez, and O. Barbosa-García, Eds., vol. 6046, International Society for Optics and Photonics. SPIE, 2006, pp. 360 – 368. [Online]. Available: <https://doi.org/10.1117/12.674556>
- [10] J. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975. [Online]. Available: <https://books.google.com.mx/books?id=JE5RAAAAMAAJ>
- [11] E. Hopper and B. C. H. Turton, "A review of the application of meta-heuristic algorithms to 2d strip packing problems," *Artificial Intelligence Review*, vol. 16, no. 4, pp. 257–300, Dec 2001. [Online]. Available: <https://doi.org/10.1023/A:1012590107280>
- [12] G. C. Onwubolu and M. Mutingi, "A genetic algorithm approach for the cutting stock problem," *Journal of Intelligent Manufacturing*, vol. 14, no. 2, pp. 209–218, Apr 2003. [Online]. Available: <https://doi.org/10.1023/A:1022955531018>
- [13] P. Poshyanonda and C. H. Dagli, "Genetic neuro-nester," *Journal of Intelligent Manufacturing*, vol. 15, no. 2, pp. 201–218, Apr 2004. [Online]. Available: <https://doi.org/10.1023/B:JIMS.0000018033.05556.65>
- [14] M. Hma Jahromi, R. Tavakkoli-Moghaddam, A. Makui, and A. Shamsi, "Solving an one-dimensional cutting stock problem by simulated annealing and tabu search," *Journal of Industrial Engineering International*, vol. 8, pp. 1–8, 03 2012.
- [15] E. López-Camacho, H. Terashima-Marin, P. Ross, and G. Ochoa, "A unified hyper-heuristic framework for solving bin packing problems," *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6876–6889, Nov. 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.eswa.2014.04.043>
- [16] H. Terashima-Marin, P. Ross, C. J. Farías-Zárate, E. López-Camacho, and M. Valenzuela-Rendón, "Generalized hyper-heuristics for solving 2d regular and irregular packing problems," *Annals of Operations Research*, vol. 179, no. 1, pp. 369–392, Sep 2010. [Online]. Available: <https://doi.org/10.1007/s10479-008-0475-2>
- [17] M. E. Berberler, U. Nuriyev, and A. YÄldÄrÄm, "A software for the one-dimensional cutting stock problem," *Journal of King Saud University - Science*, vol. 23, no. 1, pp. 69 – 76, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1018364710000741>
- [18] A. M. D. Valle, T. A. de Queiroz, F. K. Miyazawa, and E. C. Xavier, "Heuristics for two-dimensional knapsack and cutting stock problems with items of irregular shape," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12589 – 12598, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417412007233>
- [19] S. C. Leung, D. Zhang, C. Zhou, and T. Wu, "A hybrid simulated annealing metaheuristic algorithm for the two-dimensional knapsack packing problem," *Computers Operations Research*, vol. 39, no. 1, pp. 64 – 73, 2012, special Issue on Knapsack Problems and Applications. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0305054810002510>
- [20] J. F. Gonçalves and M. G. Resende, "A biased random key genetic algorithm for 2d and 3d bin packing problems," *International Journal of Production Economics*, vol. 145, no. 2, pp. 500 – 510, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925527313001837>
- [21] Y.-P. Cui, Y. Cui, and T. Tang, "Sequential heuristic for the two-dimensional bin-packing problem," *European Journal of Operational Research*, vol. 240, no. 1, pp. 43–53, 2015. [Online]. Available: <https://ideas.repec.org/a/eee/ejores/v240y2015i1p43-53.html>
- [22] F. Brandão and J. P. Pedroso, "Bin packing and related problems: General arc-flow formulation with graph compression," *Computers OR*, vol. 69, pp. 56–67, 2016.
- [23] D. Zhang, L. Shi, S. C. Leung, and T. Wu, "A priority heuristic for the guillotine rectangular packing problem," *Information Processing Letters*, vol. 116, no. 1, pp. 15 – 21, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020019015001519>
- [24] H. I. Christensen, A. Khan, S. Pokutta, and P. Tetali, "Approximation and online algorithms for multidimensional bin packing: A survey," *Computer Science Review*, vol. 24, pp. 63 – 79, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1574013716301356>
- [25] J. C. Gomez and H. Terashima-Marin, "Evolutionary hyper-heuristics for tackling bi-objective 2d bin packing problems," *Genetic Programming and Evolvable Machines*, pp. 1–31, 02 2017.
- [26] I. Kierkosz and M. Luczak, "A one-pass heuristic for nesting problems," *Operations Research and Decisions*, vol. 1, pp. 37–60, 2019. [Online]. Available: <https://ideas.repec.org/a/wut/journal/v1y2019p37-60id1401.html>
- [27] J. Martinovic, G. Scheithauer, and J. Valério de Carvalho, "A comparative study of the arcflow model and the one-cut model for one-dimensional cutting stock problems," *European Journal of Operational Research*, vol. 266, no. 2, pp. 458–471, 2018. [Online]. Available: <https://ideas.repec.org/a/eee/ejores/v266y2018i2p458-471.html>
- [28] A. K. Sato, T. C. Martins, and M. S. G. Tsuzuki, "Massive parallelization accelerated solution for the 2d irregular nesting problem," *IFAC-PapersOnLine*, vol. 52, no. 10, pp. 119 – 124, 2019, 13th IFAC Workshop on Intelligent Manufacturing Systems IMS 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S240589631930864X>

Diversificación de un portafolio de inversión por medio de un Algoritmo Genético con codificación real

Juan Fernando García-Mejía
Centro Universitario UAEM
Atacomulco
Universidad Autónoma del
Estado de México
Atacomulco Méx
fgarciam@uaemex.mx

Elsa T. Rodríguez López
Centro Universitario UAEM
Atacomulco
Universidad Autónoma del
Estado de México
Atacomulco Méx
etrodriguezl@uaemex.mx

Mineli de Jesús Navarrete
Centro Universitario UAEM
Atacomulco
Universidad Autónoma del
Estado de México
Atacomulco Méx
mdejesusn@uaemex.mx

Yenit Martínez Garduño
Centro Universitario UAEM
Atacomulco
Universidad Autónoma del
Estado de México
Atacomulco Méx
ymartinezg@uaemex.mx

Carlos Eduardo Torres Reyes
Centro Universitario UAEM
Atacomulco
Universidad Autónoma del
Estado de México
Atacomulco Méx
ctorresr@uaemex.mx

Resumen—Los problemas de optimización en esencia consisten en encontrar el mínimo o máximo de una o varias funciones matemáticas, las cuales son abstracciones de una realidad compleja y se encuentran sujetas a ciertas restricciones. La naturaleza del dominio de estas funciones determinan si es un problema combinatorio (números enteros) o numérico, si el dominio es de números reales. Un caso de estudio de este tipo es el diseño de un portafolio de inversión a partir de la diversificación de los montos a invertir en los componentes de este instrumento bursátil, para esto se usó un Algoritmo Genético canónico con codificación real, con porcentajes de cruzamiento del 80% y de mutación de 1%. Como operadores de cruce y de mutación se usaron el operador BLX- α y el gaussiano respectivamente. Se estudió, por medio de técnicas estadísticas, el comportamiento del algoritmo para el problema propuesto, lo cual incluyó la repetitividad y la determinación de las mejores condiciones de operación de la presente propuesta.

Palabras clave—portafolio de inversión, algoritmo genético, codificación real

Abstract—Optimization problems essentially consist of finding the minimum or maximum of one or more mathematical functions, which are abstractions of a complex reality and are subject to certain restrictions. The domain nature of these functions determines whether it is a combinatorial (integer numbers) or numeric problem, if the domain is of real numbers. A case of study of this kind is the design of an investment portfolio from the diversification of the amounts to be invested in the components of this stock instrument, it was used a canonical genetic algorithm with real coding, with a crossover percentage of 80% and a mutation percentage of 1%. As crossover and mutation operators they were used the BLX and the Gaussian operators,

respectively. The behavior of the algorithm for the proposed problem was studied through statistical techniques, which included the repetitiveness and the determination of the best operating conditions of this proposal.

Keywords— investment portfolio, genetic algorithm, real coding

I. INTRODUCCIÓN

El concepto de optimización puede definirse de forma práctica como el proceso de elección de la decisión óptima de un problema, en otras palabras determinar los valores máximos y mínimos de una o varias funciones representativas de un problema, las cuales pueden estar sujetas a determinadas condiciones denominadas restricciones las cuales también se pueden representar por medio de expresiones matemáticas. Este tipo de problema se puede clasificar como numérico cuando el dominio del problema es representado por medio de números reales o combinatorio cuando dicha representación esta determinada por medio de números enteros. Lo anteriormente descrito se puede expresar por medio de expresiones matemáticas 2,3,4 [1]

$$\min(\max)f(x), x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n \quad (1)$$

$$g_j(x) \leq 0 \forall (j = 1, 2, \dots, J) \quad (2)$$

$$h_j(x) = 0 \quad (3)$$

Donde $f(x)$, $g_j(x)$ y $h_j(x)$ son funciones escalares del vector x . Los componentes de $x = [x_1, x_2, \dots, x_n]^T$ son denominados variables, $f(x)$ es la función objetivo, $g_j(x)$ y $h_j(x)$ son funciones que describen las condiciones de iniquidad e igualdad respectivamente. El vector óptimo x que resuelve la ecuación se denota por x^* con el correspondiente valor de optimización $f(x^*)$ [2].

Un problema que puede ser estudiado como un problema de optimización es la diversificación de portafolios de inversión, los cuales se definen como un conjunto de activos que un inversionista posee, tales como acciones, opciones, bonos, *commodities*, moneda extranjera entre otros instrumentos financieros, tiene como finalidad generar para su propietario ganancias económicas máximas con un margen de riesgo de pérdida mínimo. Esto se puede lograr con diferentes modelos, uno de ellos, el modelo de varianza mínima o de Markowitz [3] el cual se fundamenta en el supuesto teórico de que el diseño del portafolio de inversión puede estructurarse en función de la expectativa de rendimiento medio y la desviación estándar de los rendimientos del portafolio.

El punto de partida de Markowitz es que el valor de los activos financieros es el reflejo de las expectativas sobre el valor futuro, por lo tanto, se busca el portafolio que permita una maximización del rendimiento simultánea con una minimización del riesgo. Las expresiones matemáticas del modelo se muestran en las ecuaciones 4 y 5 [3]

$$G = \sum_i a_i * w_i \quad (4)$$

$$R = \sum_i \sum_j \sigma_{i,j} * w_i * w_j \quad (5)$$

Donde G es la ganancia del portafolio, R es el riesgo, i, j son índices, $\sigma_{i,j}$ se denomina matriz de covarianza de las a acciones y w representan los porcentajes a invertir en estas y están sujetos a la restricción $\sum w = 1$.

El problema de diversificación de un portafolio de inversión puede ser resuelto por medio de técnicas de softcomputing, en específico por medio de algoritmos evolutivos (Evolutionary Algorithms, EA, por sus siglas en inglés) que se definen como procesos computacionales que están soportados de manera teórica en las leyes de la herencia de Gregory Mendel y la teoría de la evolución de Charles Darwin. Los EA tienen tres características principales [4]:

Tienen una población. Un EA mantiene un conjunto de soluciones, llamado población, representada por números binarios, en trabajos recientes suele usarse la codificación por números reales, para optimizar o aprender el problema en forma paralela. Este es un principio básico del proceso evolutivo.

- Tienen una función objetivo. Todas las soluciones en una población se llaman individuo, tienen una representación llamada código y la evaluación de desempeño se denomina aptitud o fitness. Un EA privilegia a los individuos más aptos, que permitan la optimización y la convergencia de los algoritmos.

- Variación de motor. Es un conjunto de operaciones aplicadas a los individuos de una población con la finalidad de introducir variaciones en su código, esto es fundamental para buscar un espacio de soluciones.

La literatura especializada hace referencia algunos ejemplos del uso de softcomputing y algoritmos evolutivos en la diversificación de portafolios de inversión. En [5] se optimizó un portafolio de inversiones con instrumentos bursátiles del año 2013 pertenecientes a la bolsa de valores de Colombia haciendo uso del modelo de multiplicadores de Lagrange, donde se define el procedimiento para encontrar los máximos y mínimos de funciones de múltiples variables sujetas a restricciones que permiten transformar el problema restringido de n variables a uno sin restricciones como $n+k$ variables, donde k es igual al número de restricciones.

Otro acercamiento al diseño de portafolios de inversión se encuentra en [6] donde se usaron redes neuronales para predecir el comportamiento futuro de estos siguiendo las siguientes etapas: recolección de los datos, entrenamiento de la red, ejecución de la red neuronal, posteriormente se obtienen los resultados y por último se realiza la predicción. Esto consiste en predecir cómo será el comportamiento de las variables en un tiempo determinado.

En [7] se realiza la optimización de un portafolio de inversiones mediante el uso del modelo Media-Varianza de Markowitz tomando como referencia los activos financieros de la Bolsa Mexicana de Valores en el periodo del 1 de enero de 1999 al 25 de febrero del 2014, usando como técnica de inteligencia artificial un algoritmo evolutivo multiobjetivo con codificación binaria, lo cual crea cromosomas de extensión larga. En [8] usando un contexto de solución basado en Frenes de Pareto se optimizó un portafolio realizado con acciones de la Bolsa Colombiana de Valores usando como función objetivo el modelo de Markowitz.

En este trabajo se propone, a partir de lo revisado en la literatura especializada, la construcción de un portafolio de inversión constituido por activos de la Bolsa Mexicana de Valores, cuya diversificación es realizada por medio de un Algoritmo Genético con Codificación Real

II. ALGORITMOS GENÉTICOS

Uno de los primeros algoritmos evolutivos, empleados en la optimización numérica fue el Algoritmo Genético simple (Genetic Algorithm, GA, por sus siglas en inglés), fue desarrollado por John Holland en la década de los 60 en la Universidad de Michigan, basado en los principios de Charles Darwin y Gregory Mendel presentes en la naturaleza [9] La Figura 1 muestra el diagrama de flujo de un Algoritmo Genético simple o Canónico, para la solución de un problema de optimización mono objetivo:

$$C_n = \{g_n^1, g_n^2, g_n^3 \dots, x_n^t\} \quad (6)$$

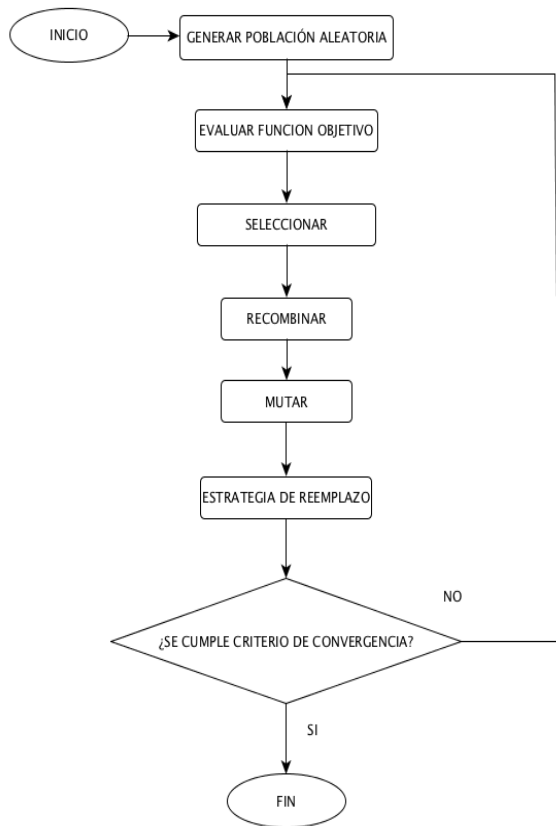


Fig. 1. Algoritmo Genético Canónico

A. Codificación Real

El conjunto de soluciones para un determinado problema de optimización resuelto con algoritmos genéticos recibe el nombre de población, la cual de manera inicial se puede generar de manera aleatoria, a partir de una distribución de probabilidad uniforme. Esto permite que el algoritmo converja de manera rápida evitando los óptimos locales [10] A cada uno de los elementos de una población de un GA se le denomina cromosoma y a la mínima expresión de un cromosoma se le denomina gen [11].

Posteriormente los elementos que conforman la población se representan con un determinado alfabeto, a este proceso se le conoce como codificación y determina los procedimientos de recombinación y mutación de los pasos subsiguientes[12] Para la propuesta que se desarrolla en esta propuesta se usa una codificación real.

La representación real usa números reales, esto involucra una serie de ventajas, dado que evita la pérdida de precisión dentro de la codificación, proporciona mayor libertad en el uso de operadores genéticos y emplea menos memoria representando más rápido el cálculo. Esta codificación permite trabajar con problemas de optimización cuyas soluciones se encuentren en el dominio discreto. Una representación de un cromosoma C con m genes se muestra en la ecuación 6 [12].

B. Selección

Después de la codificación de cromosomas sigue su evaluación en la función objetivo f de un problema determinado, calculando después la densidad de probabilidad p_n de cada uno de los C_n individuos partir de la ecuación 7 [13].

$$p_n = \frac{f(C_n)}{\sum f(C_n)} \quad (7)$$

El siguiente paso, posterior a la evaluación es la selección de cromosomas que se recombinarán con el propósito de generar nuevas soluciones de manera iterativa. Para esto se usó el operador de ruleta, el cual, es un método de selección que permite conservar el carácter estocástico de los GA y es el más común de los métodos [14]. Consiste en construir una ruleta formada por las densidades de probabilidad como lo muestra la Figura 2. De tal forma que las parejas que se seleccionan para realizar la recombinación son aquellas que abarquen la mayor superficie de la ruleta.

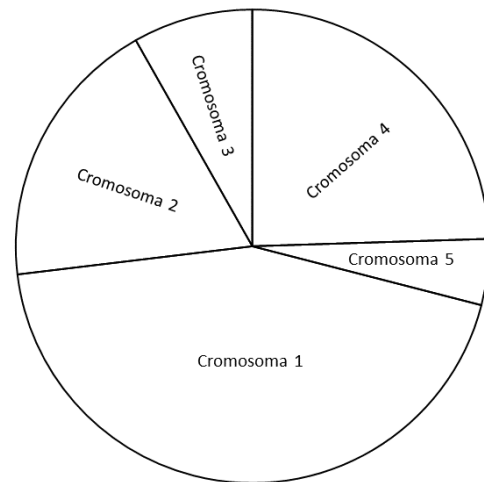


Fig. 2. Operador de ruleta

C. Cruzamiento

Si el problema de optimización tiene un vector de soluciones de la forma $\vec{x} = [x_1, x_2, x_3 \dots x_n]$, $\forall x \in \mathbb{R}$ las soluciones pertenecen a un problema del dominio continuo y se trata de un problema de optimización numérica, para esto existen diversas formas de recombinación, En esta propuesta se usó el operador BLX- α

El Operador de cruzamiento por mezcla (Blend Crossover, BLX- α , por sus siglas en inglés) es un operador de recombinación que, a partir de dos cromosomas progenitores C^{H1} y C^{H2} , genera de forma aleatoria un descendiente, a partir de la expresión 8 [15]

$$C^H = rand[(h_{min} - I * \alpha), (h_{max} + I * \alpha)] \quad (8)$$

Donde C^H es el cromosoma descendiente

$$h_{min} = \min(C_i^1, C_i^2),$$

$$h_{max} = \max(C_i^1, C_i^2)$$

Identify applicable funding agency here. If none, delete this text box.

$$I = h_{max} - h_{min}$$

$\alpha = rand[0,1]$ con distribución uniforme. Cabe destacar que se pueden producir los descendientes que sean necesarios.

D. Mutación

La mutación en la biología es un proceso que se define como cambios moleculares en las secuencias de ADN los cuales son permanentes y trascienden generacionalmente (Solari, 2007), es decir sus características son traspasadas a la siguiente generación. En los algoritmos genéticos se emula este proceso con la finalidad de introducir variabilidad genética, es decir, impedir que los cromosomas tiendan de manera prematura a un óptimo local lo cual se traduciría en una convergencia prematura. Para esta propuesta se usa el operador de mutación gaussiana que modifica a un cromosoma específico C elegido al azar por medio de una distribución de probabilidad gaussiana de media 0 y varianza definida como lo muestra la expresión 9 para cada gen g [16].

$$\sigma_k = \frac{T - t (g_k^{max} - g_k^{min})}{T} \quad (9)$$

Donde t es la generación actual, T es el máximo número de generaciones contempladas en el algoritmo de tal forma que el cromosoma mutado puede definirse como lo muestra 10

$$C' = C + N(0, \sigma_k) \quad (10)$$

III. METODOLOGÍA

Un algoritmo genético canónico es un algoritmo evolutivo que debe de estar constituido por una función objetivo que recibe el nombre de *fitness*, el cual para este caso es construida como una relación entre las ecuaciones de ganancia y riesgo, formado la expresión mostrada en la ecuación 11

$$\text{maximizar} \frac{\sum_{i=1}^N X_i * \bar{r}_i}{\sum_{i=1}^N \sum_{j=1}^N X_i * X_j * \sigma_{ij}} \quad (11)$$

Los cromosomas (potenciales soluciones a la ecuación 11) usados en esta propuesta están compuestos en forma de vector cuyos componentes representan los montos a invertir en el portafolio y tienen la forma que se describe en la ecuación 12

$$\vec{X} = (X_1, X_2 \dots \dots X_N)^T \quad (12)$$

Donde x_1 es el monto a invertir en la acción 1, x_2 es el monto a invertir en la acción 2 y x_n es el monto a invertir en la n -ésima acción del portafolio, que para esta propuesta se propone un portafolio de 10 acciones a partir de la selección realizada por un experto financiero en una ventana de 1 año. La tabla 1 muestran los valores estimados para el cálculo de la diversificación al momento de realizar el cálculo de portafolio óptimo

TABLA I MONTOS A INVERTIR

Acción	Valor de la acción
Aeromex	40.02
Bimbo	51.27
Cemex	9.57
Elektra	351.7
Herdez	42.99
Ienova	73.92
Kimber	40.75
Oma	85.73
Soriana	40.51
Walmex	42.26

Para propósitos de experimentación, se realiza, por simplicidad, un diseño de experimentos univariado, lo cual tiene implicaciones en los recursos computacionales empleados, esto se traduce a un conjunto experimental mostrado en la tabla II, donde el factor de mutación, y el porcentaje de mutación permanecen constantes, mientras se verifica las variaciones del calor de la función objetivo en términos de los cambios en el valor α del operador de cruzamiento

TABLA II MONTOS A INVERTIR

Test	cromosomas	Valor de alpha
1	100	0.1
2	100	0.2
3	100	0.3
4	100	0.4
5	100	0.5
6	100	0.6
7	100	0.7
8	100	0.8
9	100	0.9
10	100	Random [0,1]

El conjunto experimental será ejecutado 40 veces para cada uno de los test que lo componen con la finalidad de realizar un estudio estadístico que permita determinar el comportamiento del algoritmo. Esta propuesta se muestra en la figura 3. Cabe destacar que el algoritmo propuesto es escrito en Rstudio un software de licencia GNU.

En el análisis propuesto en la figura 3 tiene como propósito determinar si existen diferencias estadísticamente significativas entre los test que conforman el conjunto experimental descrito en la tabla I, para en caso afirmativo encontrar cual de ellos es el que ofrece la mejor solución al problema de diversificación de montos de un portafolio de inversión.

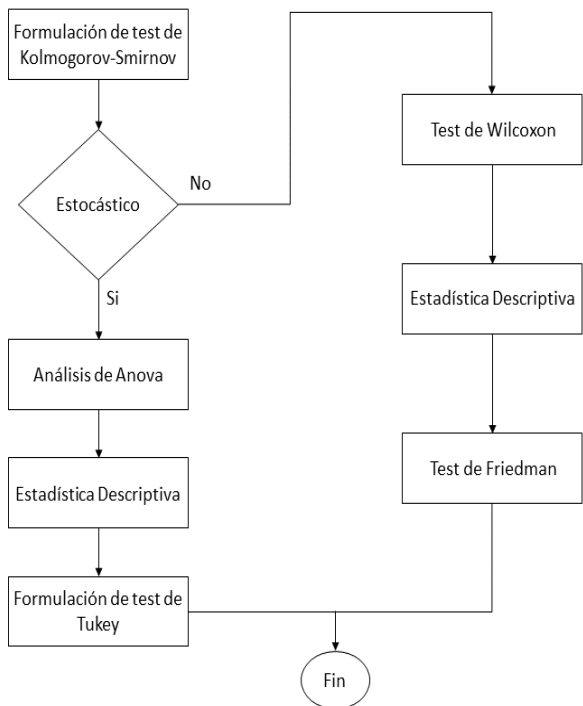


Fig. 3. Análisis Estadístico

IV. RESULTADOS

Como se mencionó en la sección anterior el primer paso de la metodología propuesta para el análisis estadístico fue realizar el test de Kolmogorov-Smirnov con la finalidad de determinar

la distribución de los resultados de los experimentos realizados. La tabla III muestra los resultados obtenidos

TABLA III RESULTADOS DE NORMALIDAD

Test	cromosomas	Valor de alpha	p-valor test de Kolmogorov
1	100	0.1	0.3132
2	100	0.2	0.7298
3	100	0.3	0.3573
4	100	0.4	0.9009
5	100	0.5	0.6677
6	100	0.6	0.9307
7	100	0.7	0.8787
8	100	0.8	0.8808
9	100	0.9	0.9436
10	100	Random [0,1]	0.5418

Los valores p del test de Kolmogorov que se obtienen en cada uno de los test es mayor a 0.05 (valor de significancia), por lo tanto es posible asumir que todos los datos presentan una distribución que se ajustan a una distribución normal, por lo tanto se asume que los datos son estocásticos, esto se puede observar por medio de la grafica de cajas que se muestra en la figura 4

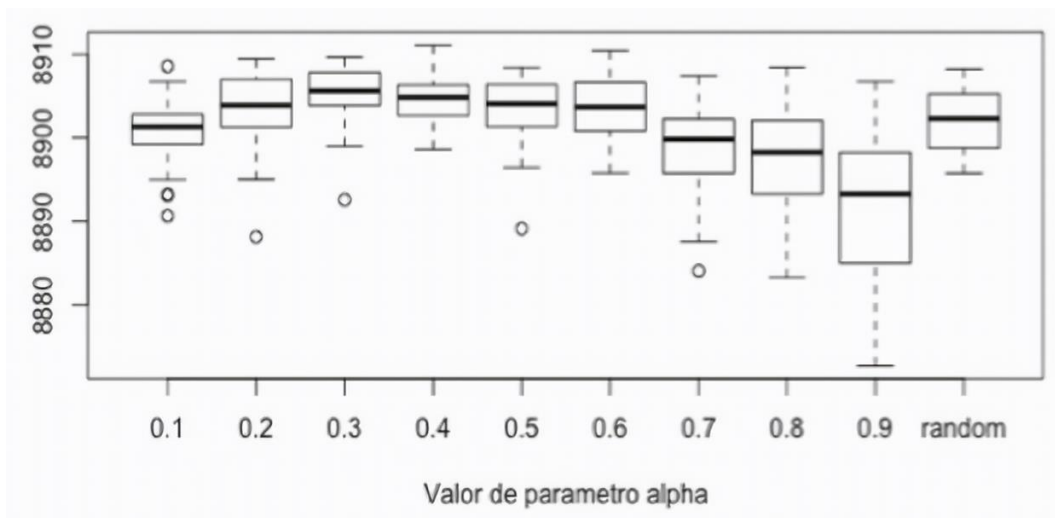


Fig. 4. Gráfico de cajas para grupos de valores de alpha

A partir de la presunción de normalidad mostrada en la tabla 3 y en la figura 4 se aplica un test de Anova, cuya finalidad es determinar si existen diferencias representativas entre los tests del conjunto experimental. El resultado del valor p de esta prueba es de $2e-16$ lo cual es mucho menor al valor de significancia de 0.05, por lo tanto se asume diferencias significativas, por lo tanto se pueden determinar los valores estadísticos mostrados en la Tabla IV

TABLA IV ESTADISTICOS DESCRIPTIVOS

Valor de α	Media	Desviación Estandar
0.1	8900.827164	3.707212797
0.2	8903.379498	4.521004873
0.3	8905.234046	3.447256221
0.4	8904.556455	3.165391002
0.5	8903.523556	3.716853885
0.6	8903.464895	3.649345998
0.7	8899.062237	4.957229757
0.8	8897.749476	6.029829737
0.9	8892.146462	8.411254507
random	8901.981701	3.587786303

Puede observarse en la Tabla IV que los valores de α que permiten obtener la media mas alta y la desviación estandar mas

baja (marcados en rojo) son 0.3 y 0.4 respectivamente, por lo tanto son de especial interes en la prueba *post hoc* que permite determinar en cuales parejas de cobinaciones existen diferencias significativas, esto se muestra en la Tabla V

TABLA V TEST DE TUKEY

Grupos	p valor
0.3-0.1	0.0018022
0.3-0.2	0.7731619
0.4-0.3	0.9997782
0.5-0.3	0.8463206
0.6-0.3	0.8182788
0.7-0.3	0.0000007
0.8-0.3	0.0000000
0.9-0.3	0.0000000
random-0.3	0.0734868

En la Tabla V los grupos con un p valor menor a 0.05, presentan diferencias estadisticamente significativas (indicados en color rojo), por lo tanto es posible afirmar, con base a esta tabla y a la tabla 4 que el test número 3 es la mejor opción de solución al problema de optimización y es estadisticamente igual a los test 6, 5,4, 2 y 10 pero diferente a 1,7,8,9. Por lo tanto en la figura 5 se muestra la gráfica de convergencia del algoritmo

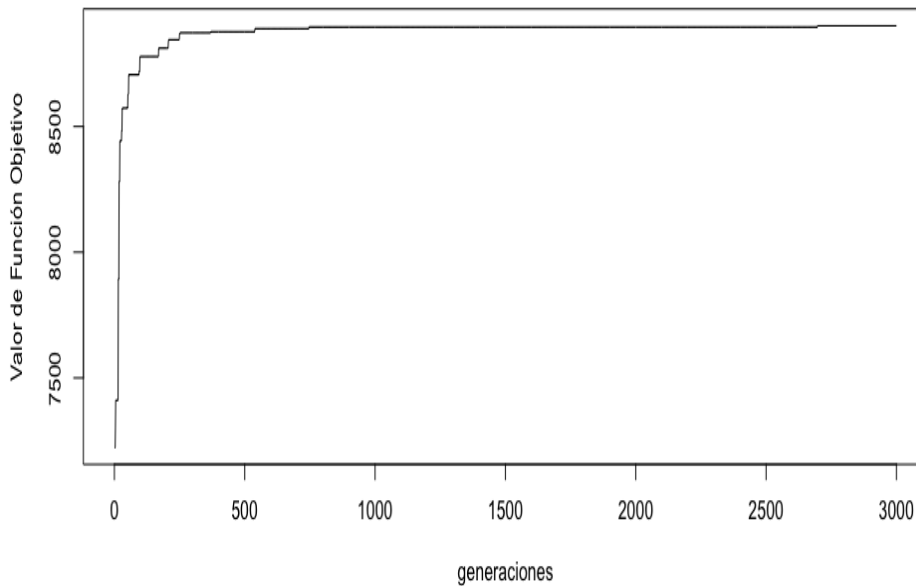


Fig. 5. Gráfico de convergencia del algoritmo propuesto

Se observa que el algoritmo converge al valor final cerca de la generación 2500, en promedio, mientras que la Tabla 6 muestra la diversificación promedio del portafolio, indicando los porcentajes del capital a invertir en cada uno de los activos, los cuales cumplen la restricción

V. CONCLUSIONES

La diversificación de los montos en un portafolio de inversión suele ser un problema de interés dado que, cuando se trata como un problema de optimización numérico la cantidad de combinaciones (que suele tender a infinito) implica una alta complejidad computacional, por lo tanto no es tratable por un especialista humano. El algoritmo propuesto demuestra los siguientes hechos:

- Un algoritmo genético canónico simple permite la diversificación de los montos de inversión de un portafolio
- La literatura especializada sugiere que en el operador de cruce BLX-alpha, α debe de tener un valor aleatorio entre cero y uno, pero los tests propuestos en el conjunto experimental demuestran que 0.3 es el mejor valor

REFERENCIAS

- [1] Duarte Muñoz, A, Metaheurísticas, Madrid, España, Universidad Rey Juan Carlos Servicio de Publicaciones, 2007
- [2] Peñuela Meneses, C. A., y M., Granada Echeverri, "Optimización multiobjetivo usando un algoritmo genético y un operador elitista basado en un ordenamiento no-dominado (nsga-ii)". *Scientia et Technica*, 1(35), pp.175-180, 2007 doi:10.22517/23447214.5393
- [3] Blanco Ramos, F., Ferrando Bolado, M., y M. F., Martínez Lobato, Teoría de la Inversión. Madrid, España, Pirámide 2015.
- [4] Burger, C. Propeller performance analysis and multidisciplinary optimization using a genetic algorithm. Dissertation, Auburn University, Alabama 2007
- [5] Cruz Trejos, E. A., Medina Varela, P. D., y H. D., Salazar Arias (2016), Optimización de portafolios de acciones utilizando los multiplicadores de Lagrange. *Scientia et Technica*, 18(1), 114-119. doi:10.22517/23447214.8301
- [6] Robledo Escobar, J. P., y C., García Gómez Portafolios de Inversión a través de Redes Neuronales y Algoritmo Genético (Tesis de maestría). Pereira, Colombia: Universidad Tecnológica de Pereira 2008
- [7] Castro Enciso, S. F. Creación de Portafolios de Inversión utilizando Algoritmo Evolutivos Multiobjetivo (Tesis de Maestría). Ciudad de México, México: Instituto Politécnico Nacional 2005.
- [8] Henaó Pérez, A. A. Creación de un Portafolio de Inversión constituido por valores primarios transados en la Bolsa de Valores de Colombia, empleando optimización multiobjetivo (Tesis de Maestría). Barranquilla, Colombia: Universidad del Norte 2007.
- [9] Yang, X.-S.. *Nature-Inspired Metaheuristic Algorithm*. Frome, Reino Unido: Luniver Press, 2010.
- [10] Mathias, K., Whitley, D., Kusuma, A., y C., Stork (1996). An Empirical Evaluation of Genetic Algorithms on Noisy Objective Functions. En S. K. Pal, & P. P. Wang, *Genetic Algorithms for Pattern Recognition* (págs. 65-86). Boca Raton, Estados Unidos: CRC Press.
- [11] Mishra, V. K., Application of Genetic Algorithms in Inventory Control. En M. Mittal, & N. H. Shah, *Optimal Inventory Control and Management Techniques*, pp. 32-46. Hershey, Estados Unidos: Business Science Reference, 2016.
- [12] Sumathi, S., Hamsapriya, T., y P., Surekha, *Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab*. Berlín, Alemania: Springer (2008).
- [13] Green, C. D. ,The generalisation and solving of Timetable Scheduling Problems. En L. D. Chambers, *Practical Handbook of Genetic Algorithms: Complex Coding Systems Volume III* pp 17-64, Estados Unidos: CRC Press, 1999.
- [14] Sivanandam, S. N., y S. N., Deepa *Introduction to Genetic Algorithms*. Berlín, Alemania: Springer (2008).
- [15] Gwiazda, T. D. *Genetic Algorithms Reference Volume I: Crossover for single-objective numerical optimization problems*. Lomianki, Polonia: Tomaszgwiazda, 2006
- [16] Roubos, H., y M., Setnes Compact Fuzzy Models and Classifiers through Model Reduction and Evolutionary Optimization. En L. Chambers, *The Practical Handbook of Genetic Algorithms Applications* (pág. 31). Chapman & Hall/CRC, 2001

Clasificación de un Disparo de Arma de Fuego Corta por Medio de los Coeficientes del Coseno de Fourier

Rogelio A. Ortega R., Daniel Martínez H., Rogelio Reyes R., Clara Cruz R., Héctor Becerril M.

Instituto Politécnico Nacional, ESIME Unidad Culhuacán

Av. Santa Ana No. 1000, Col. San Francisco Culhuacán, CP. 04430, México, Ciudad de México

Tel. (55) 5729-6000, Ext. 73264 Fax (55) 56562058

e-mail: alexortegare@gmail.com, daniell.142360@gmail.com, rreyesre@ipn.mx, ccruzra@ipn.mx, hbecerril@ipn.mx

Resumen – Actualmente el uso de armas de fuego cortas es cada vez más común, esto debido a que son armas pequeñas, fáciles de portar y fáciles de manipular; al momento de accionar un arma de este tipo (9mm, 380 y 40 Smith and Wesson) el disparo produce un sonido bastante similar entre ellos, dificultando la detección de cual arma en particular fue accionada. Basado en esta problemática se propone un algoritmo computacional el cual clasifique mediante la similitud del coseno la onda acústica de un disparo de arma de fuego corta utilizando los coeficientes del coseno de Fourier para poder distinguir las características del disparo de un arma en particular. Resultados experimentales demuestran una correcta clasificación del disparo de un arma de fuego corta (9mm, 380 y 40 S&W) con un 81% de acierto.

Palabras Clave – Arma Corta, Disparo de arma de fuego, Coeficientes del Coseno de Fourier, Similitud del coseno.

I. INTRODUCCIÓN

La explosión de un disparo de arma de fuego se ve afectada por muchos factores, incluido el tipo, calibre del arma de fuego, ubicación del tirador, velocidad del viento, temperatura, reflexión, reverberación del sonido, tipo y calidad de la bala. Estas características hacen que la clasificación de un disparo de arma de fuego corta sea más difícil de llevar a cabo.

En la actualidad se han desarrollado varios métodos para la detección y/o clasificación de disparos; por ejemplo, el método de la transformada wavelet de señales de audio [2]. Organizaciones como ShotSpotter Incorporated [3], ahora SST Inc., y los departamentos de policía de EE. UU han probado sistemas de detección de disparos en áreas con altas tasas de criminalidad. A lo largo de la costa del Pacífico de California comenzaron a trabajar con la tecnología para mejorar el tiempo de respuesta a incidentes. Pero los productos existentes son caros y, por lo tanto, no son accesibles para una gran parte de la población, debido a ello no son implementados en áreas más amplias. Por lo que los sistemas actuales de detección de disparos solo son utilizados por el ejército. Estos sistemas están desarrollados para la detección de explosiones de boca y/o golpes acústicos producidos por balas supersónicas. Un ejemplo típico de tal

sistema es Boomerang [4] o su predecesor Bullet Ears. Ambos sistemas utilizan una matriz de micrófonos para la detección y localización de disparos.

En este documento nos enfocaremos a las armas de fuego cortas que son aquellas que pueden ser disparadas por una sola mano debido a su ligereza y a sus cortas dimensiones, tales como las pistolas y los revólveres [1], comúnmente este tipo de armas son utilizadas para la defensa personal, sin embargo, en la actualidad son las más usadas para la realización de acciones ilícitas. El método de clasificación propuesto se basa en el cálculo de la autocorrelación de la señal del sonido del disparo, el cálculo de los coeficientes de coseno de Fourier de la autocorrelación y como método de clasificación se emplea la medida de similitud del coseno.

II. SISTEMA PROPUESTO

La Figura 1 muestra el diagrama a bloques del sistema de clasificación de disparos propuesto.

A. Firma de una Explosión de la Boca

Nuestro enfoque para la clasificación de un disparo se basa en la detección de la firma de la explosión de la boca (Muzzle Blast) [5]. La explosión de la boca tiene una firma única que se puede utilizar para clasificar los disparos de manera eficiente y económica. La explosión de la boca se propaga por el aire a la velocidad del sonido. A medida que la onda acústica de la explosión del hocico interactúa con la superficie del terreno circundante y los obstáculos del mismo, los reflejos de la onda se combinan, de modo que las señales posteriores recibidas en un micrófono son versiones de trayectoria múltiple de la explosión de la boca que dependen en gran medida del medio ambiente. Centrar el algoritmo únicamente en la explosión de la boca del cañón, que es la primera señal que llega al micrófono, elimina muchos de los factores medioambientales y hace que sea más fácil diseñar un algoritmo de clasificación de disparos que sea efectivo mientras sea computacionalmente económico.

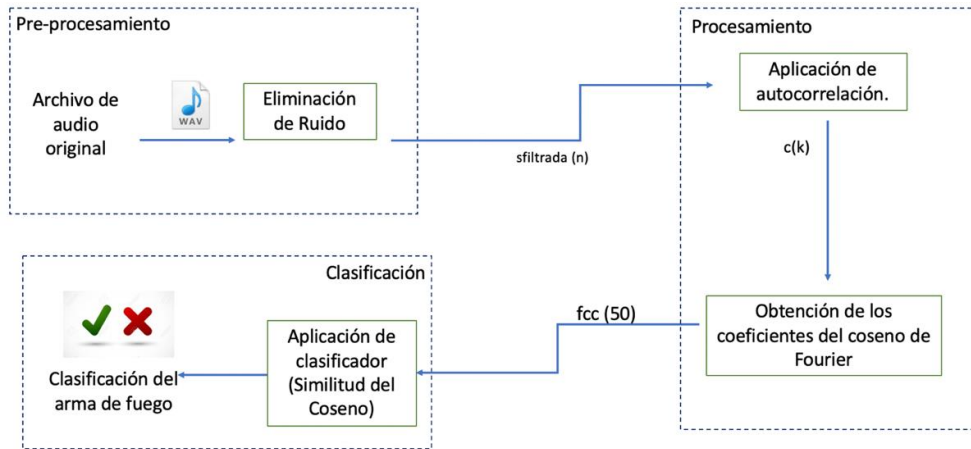


Figura 1. Diagrama a bloques del sistema de clasificación de disparos propuesto.

La explosión de la boca genera una onda de choque acústica con un máximo positivo de sobrepresión que aumenta muy rápido seguido de un mínimo de supresión, mientras la onda de choque se propaga el comportamiento no lineal del aire causa una perturbación en la presión que forman una onda de choque con forma "N" que tiene un inicio rápido y una rampa a un mínimo de presión [5]. Es importante aclarar que este patrón puede variar en amplitud y duración si se utilizan armas de diferentes calibres. La Figura 2 muestra la onda de choque "N" de un disparo de arma de fuego.

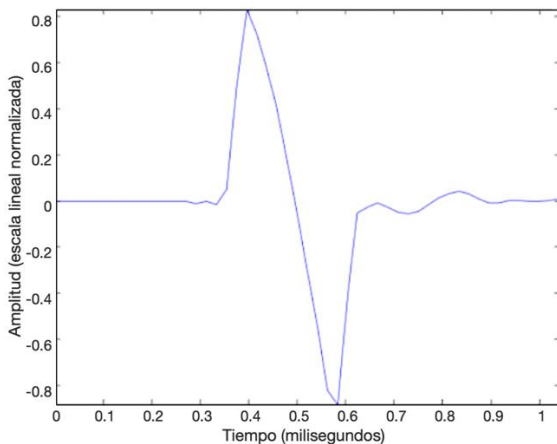


Figura 2. Onda de Choque "N" de un disparo [5].

B. Diseño del Filtro de la Onda de Choque

Se diseñó un filtro pasa bajas para eliminar el ruido dentro de las señales de audio digitalizadas de la onda de choque con forma de "N" que se produce en el momento de una detonación de un arma de fuego corta. El audio es una señal de dos canales, de los cuales solo se tomó uno de ellos para realizar el procesamiento de la señal digitalmente las señales empleadas están muestreadas a 44,100Hz con 16 bits por muestra.

Como audios de referencia para el diseño del filtro, se tomaron en consideración los audios de diferentes disparos de armas de fuego cortas (9mm, 380 y 40 S&W) a las cuales se les aplicó la transformada rápida de Fourier (FFT por sus siglas en inglés) la cual es empleada para transformar señales entre el dominio del tiempo y el dominio de la frecuencia utilizando (1).

$$X(k) = \sum_{j=1}^N x[j] W_N^{(j-1)(k-1)} \quad (1)$$

donde $W_N = e^{-\frac{2\pi i}{N}}$.

De esta manera al visualizar el espectro de frecuencias de las diferentes señales de disparos de arma de fuego, se encontró que la mayor parte de la energía se concentra en frecuencias inferiores a 1,500Hz como se muestra en la Figura 3.

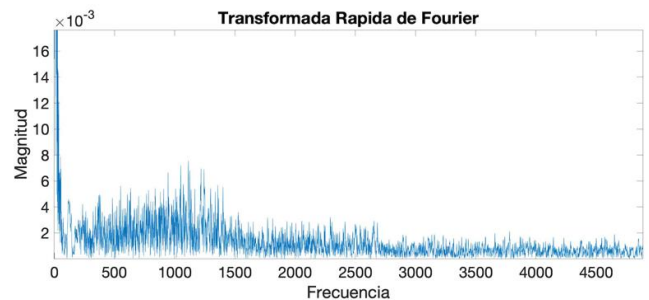


Figura 3. Espectro de frecuencias de un disparo de arma de fuego de 9mm.

Con base al resultado obtenido anteriormente se realizó un filtro FIR (Finite Impulse Response) pasa bajas debido a que este tipo de filtro ofrece una respuesta de fase más lineal y son más estables, ya que no poseen retroalimentación [6]. Dicho filtro cuenta con una frecuencia de corte de 1,500Hz y una banda de transición de 10Hz lo cual nos va a permitir eliminar ruidos no deseados y apreciar la representativa forma de "N" como se muestra en la Figura 4.

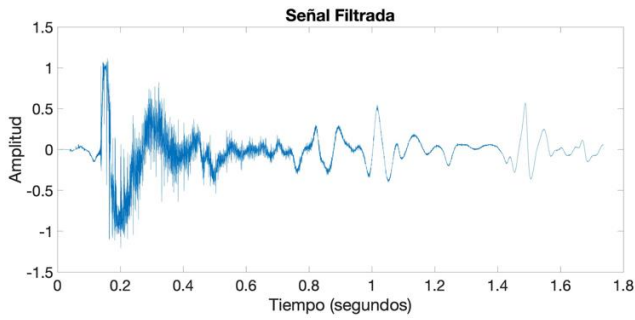


Figura 4. Señal Filtrada de un disparo de arma de fuego de 9mm.

C. Autocorrelación

La autocorrelación es de gran utilidad para encontrar patrones repetitivos de una señal o identificar la frecuencia fundamental de la misma [7, 8] la cual se obtiene mediante (2).

$$c(k) = \frac{1}{T} \sum_{i=0}^{T-k} (g(i)) * (g(i+k)) \quad (2)$$

Donde $k = 1, 2, 3, \dots, T-1$ y T es el número de muestras que tenemos.

De esta manera se obtienen resultados favorables cuando la señal encuentra un patrón repetitivo consigo misma como se muestra en la Figura 5.

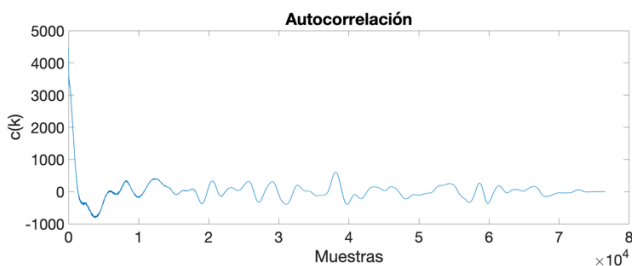


Figura 5. Autocorrelación de un disparo de arma de fuego de 9mm.

D. Coeficientes del Coseno de Fourier

Para obtener características individuales de los disparos se calculan los Coeficientes del Coseno de Fourier (FCC por sus siglas en inglés) de las funciones de autocorrelación anteriores utilizando (3).

$$c(k) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} a_n \cos(nk) \quad (3)$$

Donde los Coeficientes del Coseno de Fourier de la función de autocorrelación pueden ser calculados mediante (4).

$$a(n) = \frac{2}{T-1} \sum_{k=1}^{T-1} c(k) \cos\left(\frac{n\pi k}{T-1}\right) \quad (4)$$

De esta manera se obtienen las características individuales de los disparos, además de reducir la cantidad de elementos de nuestro vector como se ilustra en la Figura 6.

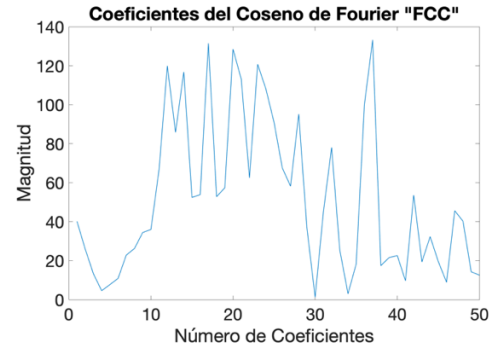


Figura 6. FCC de la función de Autocorrelación.

E. Similitud del Coseno

La similitud del coseno compara dos vectores distintos enfocándose en su orientación, es decir, en el caso de que los vectores sean paralelos entre ellos, se considera que los vectores son similares, sin embargo, si estos son ortogonales o perpendiculares entre sí, se consideraran como vectores diferentes, por lo que se mide el coseno del ángulo entre ellos [9], llegando a tener valores de entre -1 a 1, esto se puede expresar matemáticamente como en (5):

$$SC = \frac{\sum_{i=1}^N A_i B_i}{\sqrt{\sum_{i=1}^N (A_i)^2} \sqrt{\sum_{i=1}^N (B_i)^2}} \quad (5)$$

donde N es el número de muestras extraídas en FCC.

De esta manera observamos que los vectores que tengan la misma orientación tendrán una similitud de 1, dos vectores orientados a 90° entre sí tendrán una similitud de 0 y dos vectores opuestos tendrán una similitud de -1 independientemente de su magnitud.

III. PRUEBAS Y RESULTADOS

El sistema se desarrolló utilizando la plataforma de MathWorks Matlab® R2018a, en una computadora con procesador Intel (R) Core i7(R) CPU 2.60GHz, memoria RAM de 12.00GB y Sistema Operativo Windows 10 Home Single Language.

A continuación, se muestran los resultados obtenidos del sistema propuesto para 60 diferentes audios estéreo de disparos de armas de fuego cortas (9mm, 380 y 40 S&W) con frecuencia de muestreo de 44,100Hz y 16 bits por muestra. En la Figura 7 y 8 se muestran los audios originales y los audios filtrados de tres diferentes armas.

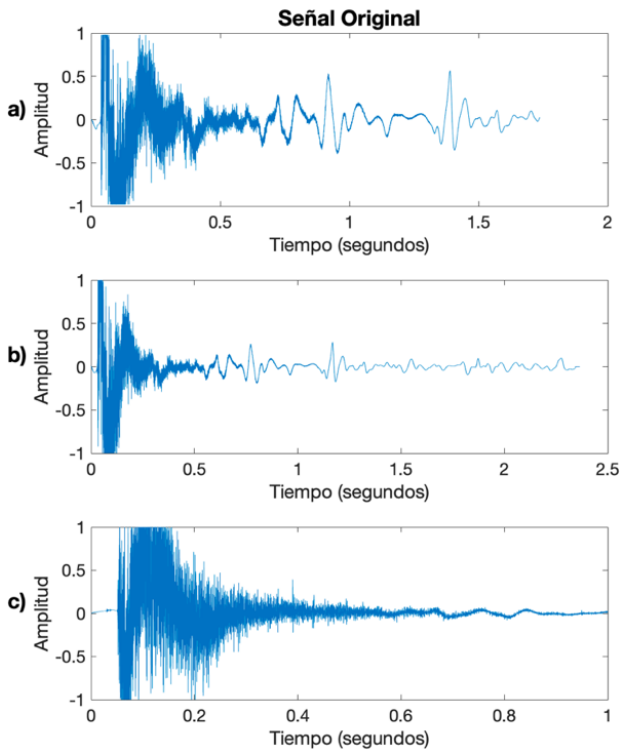


Figura 7. Señal original del disparo de un arma de fuego. (a) 9 mm, (b) 380, (c) 40S&W.

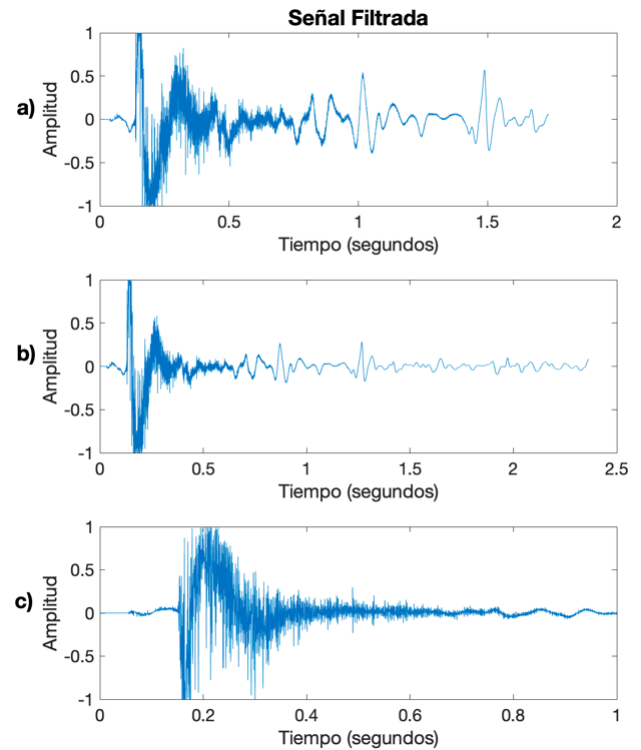


Figura 8. Señal filtrada del disparo de un arma de fuego. (a) 9 mm, (b) 380, (c) 40S&W.

La Figura 8 muestra de manera más clara la onda de choque “N” de una señal de disparo, igualmente se alcanza a apreciar la onda de choque con pequeñas diferencias, esto se debe a que al ser las tres armas de diferente calibre la onda “N” varia. Al comparar las tres ondas de las diferentes armas, podemos apreciar sus diferencias por lo que asumimos que cada arma tiene características diferentes sin embargo al ser estas clasificadas como armas de fuego cortas comparten varias de esas características.

La Figura 9 muestra la autocorrelación resultante del sistema propuesto empleando las señales filtradas. De igual manera, en la misma figura, podemos observar cómo es que las funciones de autocorrelación son similares, sin embargo, cada una tiene magnitudes diferentes lo cual hace que se diferencien entre ellas esto se debe al patrón de repetición que se tiene ya que gracias a ello se genera un valor máximo cuando se detecta la similitud consigo misma.

Posteriormente se obtienen las características de los coeficientes del coseno de Fourier de las funciones de autocorrelación anteriores, de esta manera disminuimos el número de muestras que tenemos y obtenemos características más específicas como se muestra en la Figura 10.

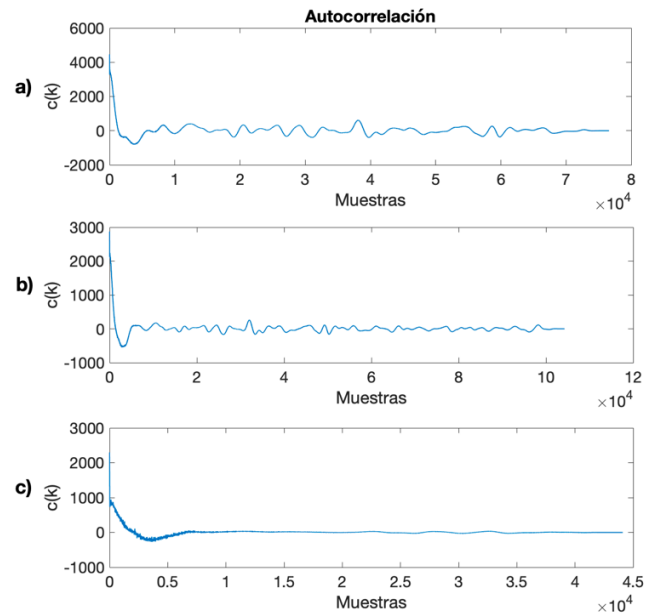


Figura 9. Autocorrelación de la señal filtrada del disparo de un arma de fuego. (a) 9 mm, (b) 380, (c) 40S&W.

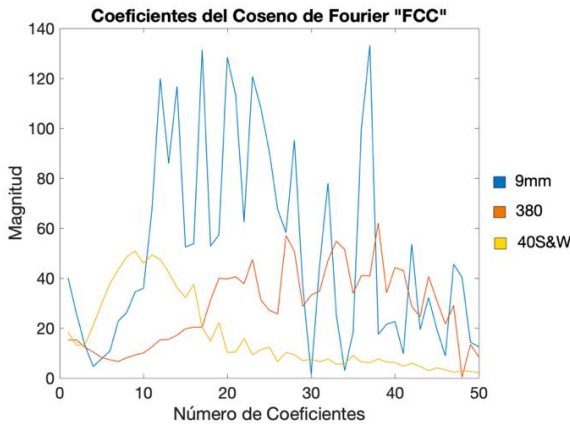


Figura 10. FCC de tres diferentes armas, 9 mm, 380 y 40S&W.

Partiendo de las características de los FCC, se emplea como método de clasificación la similitud del coseno con lo que obtendremos valores del rango de -1 a 1 esto con el objetivo de comparar las señales resultantes de los diferentes FCC estos resultados se muestran en la Tabla 1.

Tabla 1. Valor promedio de la Similitud del coseno para diferentes FCC.

	9mm	380	40S&W
9mm	0.975	0.751	0.819
380	0.751	0.913	0.652
40S&W	0.819	0.652	0.908

Tomando en consideración los resultados obtenidos en la Tabla 1, se propone emplear un umbral de clasificación de 0.9; de esta manera al realizar el análisis de una señal de onda acústica nueva, se podrá comparar la similitud de los coeficientes FCC de esta nueva señal con el valor promedio de los FCC empleados para el cálculo del umbral y así poder clasificar que arma en particular fue accionada.

Tabla 2. Porcentaje de acierto de clasificación.

	9mm	380	40S&W
Porcentaje de acierto	90.27%	80.10%	72.23%

Los resultados mostrados en la Tabla 2 demuestran la correcta clasificación de diferentes disparos de armas de fuego cortas.

IV. CONCLUSIÓN

El presente artículo propone y desarrolla un algoritmo computacional que clasifica la onda acústica que genera un disparo de arma de fuego corta mediante el método de los coeficientes del coseno de Fourier los cuales se utilizaron para obtener las características individuales de los disparos de un arma de fuego, con ayuda de estos se empleó el método de clasificación de la similitud del coseno. También se pudo observar que las características obtenidas dependen de

muchos factores, como la reflexión, la reverberación del sonido, la dirección del arma de fuego, la posición de disparo, la orientación o ubicación de la grabadora de sonido y el ruido ambiental presente en el momento del disparo, en conjunto todos estos factores podrían llegar a generar una incorrecta clasificación. Sin embargo, se demostró que el método propuesto es capaz de clasificar de manera correcta con un porcentaje promedio de acierto del 81%.

AGRADECIMIENTOS

Agradecemos al Instituto Politécnico Nacional, a la COFAA del IPN y al Programa Institucional de Formación de Investigadores del IPN por el apoyo otorgado para el desarrollo de este trabajo.

REFERENCIAS

- [1] Secretaría de la Defensa Nacional, 01 de mayo de 2019 <https://www.gob.mx/sedena/acciones-y-programas/armas-de-fuego-19734>
- [2] Chacón-Rodríguez, Alfonso, "Evaluation of Gunshot Detection Algorithms" in Circuits and Systems I: Regular Papers, IEEE Transactions on circuits and systems, 2011
- [3] ShotSpotter History. URL <http://shotspotter.com/company/history>
- [4] Boomerang datasheet, disponible en: http://www.bbn.com/products_and_services/boomerang [17-03-2015]
- [5] R. C. Maher. "Summary of gunshot acoustics", Universidad del Estado de Montana, 2006. URL http://www.montana.edu/rmaher/publications/maher_aac_04_06.pdf.
- [6] Paz Martin, "Uso de la placa Discovery para el cálculo e implementación de filtros FIR e IIR", Buenos Aires, Argentina, EST. 2016.
- [7] Proakis, J. G. & Manolakis, D. G. "Tratamiento digital de señales", Pearson Prentice-Hall. pp. 458-473, 2007.
- [8] Albertí, E. B. "Procesado digital de señales-2: Fundamentos para comunicaciones y control (Vol. 169)". Univ. Politèc. de Catalunya, 2010.
- [9] Broumi, S., & Smarandache, F. "Cosine similarity measure of interval valued neutrosophic sets", Neutrosophic Sets and Systems, Vol. 5, 2014.

RED NEURONAL PARA LA DETECCIÓN DE DATOS ANÓMALOS EN DISPOSITIVOS IoT

Jesus A. Lechuga O., Fernando Sánchez C., Clara Cruz R., Rogelio Reyes R.

Instituto Politécnico Nacional, ESIME Unidad Culhuacán

Av. Santa Ana No. 1000, Col. San Francisco Culhuacán, CP. 04430, México, Ciudad de México

Tel. (55) 5729-6000, Ext. 73264 Fax (55) 56562058

e-mail: jlechugao1500@alumno.ipn.mx, fsanchezc1600@alumno.ipn.mx, ccruzra@ipn.mx, rreyesre@ipn.mx

Resumen- Asegurar el Internet de las cosas (IoT) se ha convertido en un reto significativo para la ejecución de aplicaciones y servicios. Particularmente, la funcionalidad de los dispositivos IoT se encuentra determinada en gran medida de la fiabilidad de su transmisión de datos. Los ataques cibernéticos, como la inyección de datos y las amenazas man-in-the-middle generan desafíos en el tema de seguridad. Para asegurar los dispositivos IoT contra tales ataques se requiere tener en cuenta el estricto poder computacional y la necesidad de operaciones que demanden los mínimos recursos. Se propone un método de aprendizaje automático para el análisis y detección de anomalías en los datos involucrados en la comunicación de los dispositivos IoT. Este método permite que el servidor de IoT, que recolecta muestras de estos, autentique la confiabilidad de los datos. En adición a ello, el método propuesto es capaz de comunicar los resultados obtenidos durante el análisis con una precisión del 99.5%.

Keywords- IoT, Redes Neuronales, Datos Anómalos, Seguridad informática, ESP8266.

I. INTRODUCCIÓN

En la actualidad existen miles de dispositivos electrónicos como pueden ser cámaras de seguridad, refrigeradores, microondas, automóviles, equipo industrial entre otros objetos de última generación los cuales se encuentran interactuando constantemente entre ellos a través de internet. Esta red de dispositivos es llamada Internet de las cosas (IoT) [1], la cual se espera que con el pasar del tiempo alcance la cifra de 20 mil millones de dispositivos activos [2]. Las redes de sensores inalámbricas (WSN), han recibido una gran atención en los últimos años desde puntos de vista tanto académicos como industriales, debido a los avances de la tecnología. Las WSN se encuentran diseñadas y constituidas por sensores autónomos con el propósito de hacer posible la comunicación entre sí con un consumo mínimo de energía además, su propósito es la recolección de datos para su posterior uso.

II. ANTECEDENTES

Janice Cañedo et al., [3] presenta un método de aprendizaje automático con el objetivo de obtener una predicción sobre la veracidad del dato entrante en el sistema. La ventaja de este sistema propuesto es el bajo costo computacional del mismo debido a que el vector de entrada de la Red Neuronal Artificial

(RNA) cuenta con pocos datos. El método propuesto utiliza únicamente sensores de temperatura, lo que implica una desventaja debido a la independencia entre sí mismos. También implementan el uso de dos microcontroladores a la vez: Atmel 328P y ESP8266 en distintas placas, lo que incrementa el costo del hardware. El sistema propuesto por el autor implementa sensores de temperatura (DHT11); recolecta el ID del dispositivo en cuestión, el dato proporcionado (temperatura y humedad) y el tiempo de retardo del envío a la base de datos con fines de entrenamiento de la Red Neuronal Artificial (RNA) Perceptrón Multicapa.

Elike Hodo, et al., [4] expone un análisis de amenazas de IoT y utiliza una RNA para combatir estas amenazas. Un Perceptrón Multicapa, un tipo de RNA supervisada, la cual se entrena utilizando rastreos de paquetes de Internet, luego se evalúa su capacidad para frustrar los ataques de denegación de servicio distribuido (DDoS / DoS). El documento se centra en la clasificación de patrones normales y de amenaza en una red IoT. El procedimiento ANN se valida contra una red IoT simulada. Los resultados experimentales demostraron una precisión del 99.4% y pudieron detectar con éxito varios ataques DDoS / DoS.

Ameeth Kanawaday, et al., [5] Propone un enfoque industrial de un sistema IoT y explora el uso del pronóstico de la Media Móvil Integrada Autorregresiva (ARIMA) en los datos de series de tiempo recopilados de varios sensores de una máquina de corte longitudinal, para predecir las posibles fallas y defectos de calidad, mejorando así el proceso general de fabricación. El uso de Machine Learning demuestra, por lo tanto, un componente vital en IoT que tiene casos de uso en la gestión y control de calidad, reduciendo el costo de mantenimiento y mejorando el proceso de fabricación en general.

III. SISTEMA PROPUESTO

Con el fin de determinar la orientación del sistema propuesto, a continuación, se describe de manera general la metodología para la recolección y creación de un banco de datos para realizar las pruebas y más adelante se discute el método de aprendizaje automático utilizado. La Figura 1 muestra el diagrama a bloques general del sistema propuesto, el cual contiene dos bloques principales uno asociado a la fase de entrenamiento y el otro asociado a la fase de prueba.

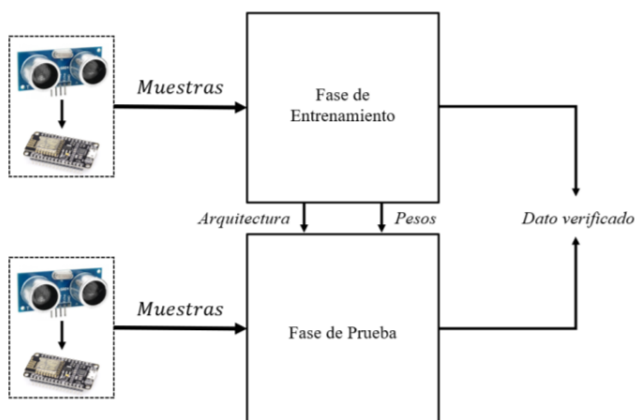


Fig. 1. Diagrama a bloques general del sistema propuesto.

A. Bloque de Fase de Entrenamiento

Los datos de entrada del bloque de Entrenamiento provienen de dispositivos NodeMCU, los cuales están basados en el conocido WiFi ESP8266, se conectaron distintos sensores en cada módulo, que en conjunto formaron un sistema de seguridad de domótica. Los sensores utilizados fueron: sensor Magnético MC-38, un Sensor Ultrasónico HC-SR04, Sensor de Movimiento HR-SR501 y finalmente una Fotorresistencia LDR (GL55). Se agregaron dispositivos NodeMCU sin sensores a la red con el fin de enviar los datos falsos que servirán de entrenamiento para la RNA. Los módulos fueron conectados a un punto de acceso conectado mediante un cable Ethernet a un ordenador. En la Figura 2 se muestra a detalle el procesamiento de los datos durante la fase de entrenamiento.

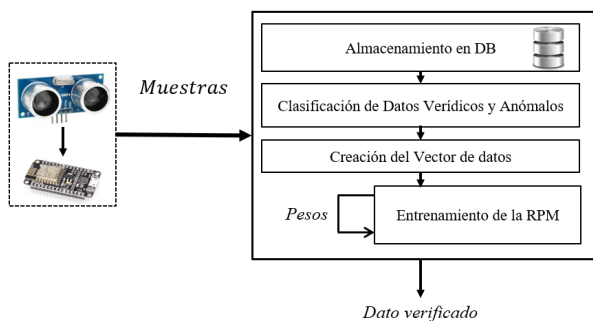


Fig. 2. Diagrama a bloques de la Fase de Entrenamiento del sistema propuesto.

Se construyó una base de datos con el sistema para almacenar las muestras tomadas por los dispositivos IoT. Cada lectura de los sensores fue almacenada en la base de datos aproximadamente cada cinco segundos y se almacenaron los siguientes datos; Hora: tiempo del registro en la tabla, Puerta: indica si el sensor está activado o no. Distancia: distancia que existe entre la posición del sensor Ultrasónico y la pared (o la persona, en caso de ser activada). Movimiento: indica si se detecta movimiento. Luz: indica si la luz se encuentra encendida o apagada. Falso: con el fin de conseguir entrenar la Red Neuronal (debido a que es de tipo Supervisada [6]) se ha reservado un campo

binario el cual ha sido nombrado VERACIDAD para indicar si el vector a verificar es veraz o no lo es. Los parámetros que contiene cada registro se ilustran en la Tabla 1.

TABLA I. PARÁMETROS ALMACENADOS EN LA BASE DE DATOS

Sensores					
Hora	Puerta	Distancia	Movimiento	Luz	Veracidad
10:58:00	1	80	0	0	0

Se utilizó una RNA tipo Perceptrón multicapa debido a que es uno de los tipos de redes más comunes y su arquitectura es una de las más poderosas. Se basa en otra red más simple llamada Perceptrón en la cual el número de capas ocultas solo es una como se muestra en la Figura 3. La RNA Perceptrón Multicapa están formadas por una capa de entrada, capas ocultas y una de salida. Cada una de las neuronas en la capa oculta o de salida recibe características de entrada de una capa previa (retro propagación). Debido a esto no existen conexiones entre las neuronas de la capa oculta [7]. La capa de entrada recibe tantas características como las variables independientes que se necesiten representar. La capa de salida corresponde a la clasificación de la red neuronal.

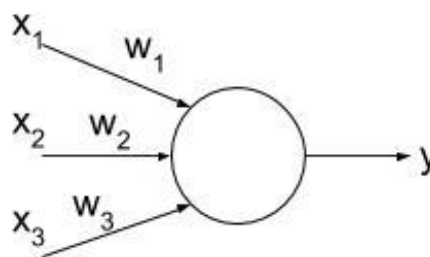


Fig. 3. Red Neuronal Artificial Perceptrón.

Frank Rosenblatt [8], propone el algoritmo de aprendizaje la RNA Perceptrón en donde automáticamente aprenderá los mejores coeficientes de la matriz de pesos los cuales se multiplicarán con las características de entrada para decidir si una neurona es activada o no. Por lo tanto, este algoritmo es usado para categorizar una o más clases. El valor de una neurona puede ser definida como (1), de este modo se puede obtener el valor de cada una de las neuronas, en donde se calcula a partir de los pesos w y los valores de entrada x .

$$y = f(w_1x_1 + \dots + w_nx_n) \quad (1)$$

Donde y es la neurona para evaluar.

Dependiendo del valor que se obtenga de esta operación pasara por una función de activación (2) en donde se predice la clase a la que pertenece, la función que se ocupa en las redes perceptrón es una variante de una la función escalón.

$$\phi = \begin{cases} 1 & \text{si } z \geq 0 \\ -1 & \text{si } z < 0 \end{cases} \quad (2)$$

Donde ϕ es la función de activación.

Las conexiones de la RNA Perceptrón multicapa también conocidas con el nombre de feedforward. Usualmente, las

neuronas de una capa se encuentran conectadas a la siguiente capa. Es cuando se dice que se encuentra conectada totalmente. El Perceptrón multicapa define una relación entre las variables de entrada y las de salida. Esta relación se obtiene propagando hacia adelante los valores de las variables de entrada. Para este proceso cada una de las neuronas de la red, procesa la información recibida por las entradas y produce una respuesta que se propaga mediante las conexiones correspondientes, hacia las neuronas de la siguiente capa como se puede observar en la Figura 4.

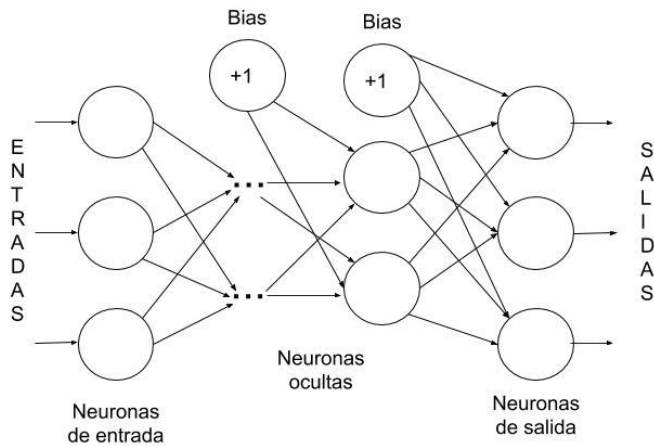


Fig. 4. Red Neuronal Artificial Perceptrón Multicapa.

Esta RNA se entrena mediante datasets (conjunto de datos), que han sido previamente clasificados. Esta red hace el uso de el algoritmo backpropagation [9], el cual es utilizado para entrenar la red a su vez que reduce los errores de clasificación cometidos por la misma y ajusta los pesos. Este algoritmo se fundamenta en el método del gradiente, donde los pesos se modifican mediante procesos iterativos, así mismo se calcula el error (3) para ajustar los pesos de cada capa.

$$E = \frac{1}{N} \sum_{n=1}^N e(n) \quad (3)$$

Donde E es una función de error, N es el número de características y $e(n)$ es el error cometido por la red para el patrón n, dado por (4).

$$e(n) = \frac{1}{n_c} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2 \quad (4)$$

Siendo $Y(n) = (y_1(n), \dots, y_{n_c}(n))$ y

$S(n) = (s_1(n), \dots, s_{n_c}(n))$ los vectores de salida de la red y salidas deseadas para el patrón n.

Para obtener mejores resultados se utilizan métodos del gradiente estocástico (5), los cuales consisten en una sucesiva minimización de los errores para cada característica, $e(n)$, en lugar de minimizar el error total E, donde cada peso w de la red se modifica para cada característica de entrada n.

$$\Delta w(n) = w(n-1) - \eta \frac{\partial e(n)}{\partial w} \quad (5)$$

Donde $e(n)$ es el error para las características y η es la tasa de aprendizaje.

B. Bloque de la Fase de Prueba.

Una vez que se tiene la red entrenada, se pone a prueba con el conjunto de datos reservados para verificar su capacidad de generalización, como se muestra en la Figura 5.

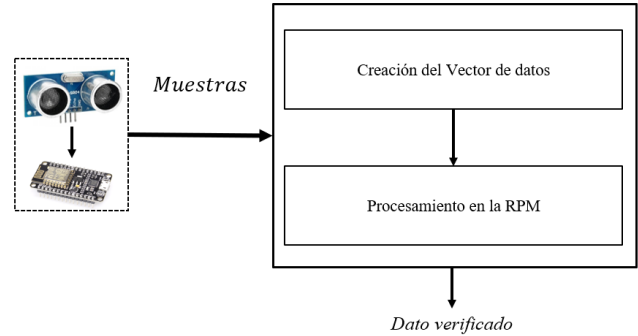


Fig. 5. Diagrama a bloques general de la Fase de Prueba del sistema propuesto.

IV. EXPERIMENTACIÓN Y RESULTADOS

Se realizaron las conexiones entre los sensores y los módulos ESP8266 como se muestra en las Figuras 6 y 7.

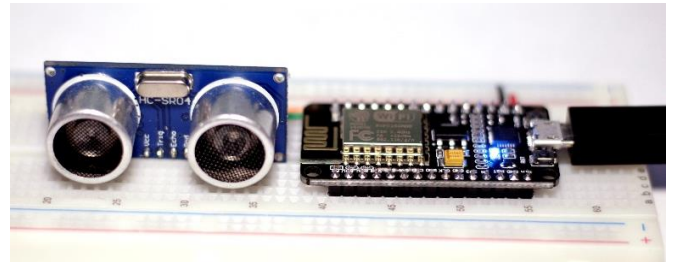


Fig. 6. Prototipo de la conexión entre un Módulo NodeMCU y un Sensor Ultrasónico HC-SR04.

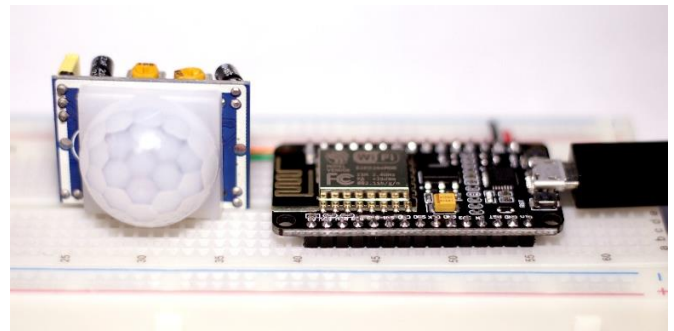


Fig. 7. Prototipo de la conexión entre un Módulo NodeMCU y un Sensor de Movimiento HR-SR501

A continuación, se recolectaron aproximadamente 4,000 muestras de datos de los módulos y se almacenaron las mismas en una base de datos MySQL dentro del ordenador conectado mediante un cable Ethernet al Punto de Acceso. Se separan los datos en dos conjuntos aleatorios con la metodología de 80-20 (entrenamiento y pruebas, respectivamente).

Cuando se espera que un dato sea verídico se le asigna el valor de 1. Al verificar el uso de nuestra red, se modifican los sensores para enviar datos anómalos a la base de datos cada 5 segundos. Los datos inválidos se introducen nuevamente a la red neuronal con la intención de verificar el funcionamiento de la red. Se usan valores de 1 y 0, para predecir, tanto datos verídicos y datos anómalos respectivamente.

La recolección de datos tuvo una duración mayor a las 8 horas, con el fin de completar un día normal en una habitación, en donde se encuentra una persona efectuando movimientos de entrada-salida de la misma y en tiempos aleatorios o de la persona que se encuentra en la habitación; las características empleadas en los patrones de entrenamiento de la red neuronal que se usaron son: un sensor de apertura de puerta, sensor de distancia, sensor de movimiento y un sensor de luz. Posteriormente, se hizo la inyección de datos anómalos, de una forma similar, modificando los sensores, para simular el efecto de intrusión de un agente externo.

Se utiliza la división de los datos en un conjunto para entrenar la red neuronal y otro para pruebas, usando la metodología de ochenta – veinte. Para los datos anómalos se tomó en consideración los siguientes criterios: 1) La interconexión de los sensores, 2) el sensor de inicio es la apertura de la puerta, 3) si un sensor se encuentra activo, o no se activó el sensor de puerta se cataloga como anómalo.

La arquitectura propuesta para el entrenamiento de la RNA consta de 4 neuronas de entrada las cuales tienen los valores tomados por los sensores descritos anteriormente, consta de 2 capas ocultas, cada una de estas con 5 y 3 neuronas respectivamente, con una neurona en la capa de salida donde se evalúa si los datos son verídicos o no, como se puede observar en la Figura 8. La arquitectura empleada es totalmente conectada, se utiliza un bias, el cual se encuentra conectado tanto a las capas ocultas como a la capa de salida, el objetivo del bias es hacer un desplazamiento en el hiperplano de separación, lo que permite incrementar la velocidad de la convergencia de la red.

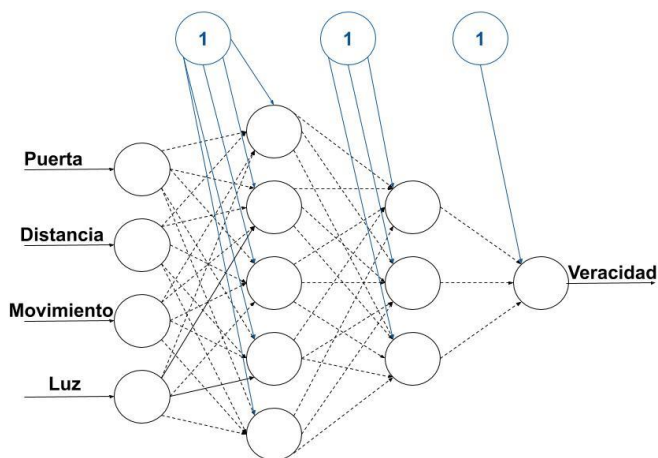


Fig. 8. Arquitectura propuesta para la RNA Perceptrón Multicapa.

Se usa un factor de convergencia de 0.001, con un error mínimo de 0.0001, se emplea una función de activación, unidad

lineal rectificada (ReLU), además del optimizador Adam [10], este algoritmo se basa en el descenso de gradiente estocástico, el cual ayuda a mejorar el valor de los pesos de RNA en un conjunto de miles de datos, lo que da como resultado un aproximado de 313 épocas para que la RNA convergiera como se puede observar en la Figura 9.

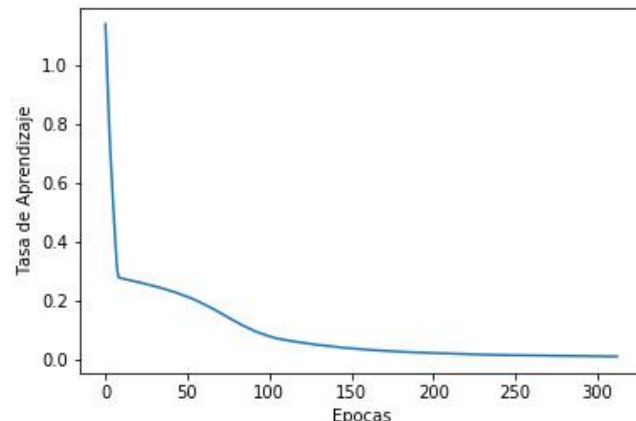


Fig. 9. Grafica del error de entrenamiento para RNA Perceptrón Multicapa propuesta.

En la tabla 2, se puede apreciar la forma en la que la red neuronal toma las decisiones con los datos de prueba. Cada columna muestra el valor de los sensores, como la veracidad de cada uno de ellos, obteniendo un 99% de eficacia para la clasificación y cerca de 1% arrojando falsos negativos como se muestra en la tabla 3.

TABLA 2. EJEMPLO DE LAS CARACTERÍSTICAS DE LOS DATOS EMPLEADOS PARA LA CLASIFICACIÓN

ID	Sensores				Veracidad
	Puerta	Distancia	Movimiento	Luz	
6	0	1000	0	0	1
95	1	0	0	1	0
294	0	0	0	0	0
965	0	47	1	0	1

Es importante señalar que para una correcta clasificación se deben tomar una gran variedad de patrones de entrenamiento (aprox. 4000), lo que permitirá a la RNA en conjunto con los dispositivos IoT tener una efectividad para la detección de intrusos bastante alta; este modelo de sistema es escalable lo que permite agregar una mayor diversidad de sensores, pero tomando en consideración que se deberán incrementar los datos recolectados como patrones de entrenamiento.

TABLA 3. CLASIFICACIÓN DE DATOS ANÓMALOS POR LA RNA PERCEPTRÓN MULTICAPA

Matriz de Confusión		
Observaciones	Positivos	Negativos
Positivos	177	0
Negativos	1	622

IV. CONCLUSIONES

Los problemas de seguridad en dispositivos IoT no siempre se pueden resolver mediante una actualización de firmware debido a que en la mayoría de las veces no se dispone de uno, teniendo que esperar a una nueva versión de hardware. Existen diferentes formas de obtener el control de estos dispositivos, lo cual compromete no solo su integridad si no la de toda nuestra red al acceder a ella, con el fin de obtener información crítica o provocar comportamientos anómalos. Por lo que utilizar la RNA dota al sistema de una capa de seguridad extra al mantener monitoreados los dispositivos IoT, y ello permite encontrar posibles conexiones apócrifas. El desarrollo de este proyecto permitió ampliar la visión acerca de las nuevas tecnologías y su seguridad asociada, las cuales prometen conectar todos nuestros dispositivos a la red y de la misma manera concientizarnos sobre comprometer nuestros datos en la misma, debido a que el usuario no siempre posee todo el control sobre los dispositivos inteligentes.

AGRADECIMIENTOS

Agradecemos al Instituto Politécnico Nacional, a la COFAA del IPN y al Programa Institucional de Formación de Investigadores del IPN por el apoyo otorgado para el desarrollo de este trabajo.

REFERENCIAS

- [1] K. Ly and Y. Jin, "Security Challenges in CPS and IoT: From End-Node to the System," 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Pittsburgh, PA, 2016, pp. 63-68.
- [2] S. Naik and V. Maral, "Cyber security — IoT," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2017, pp. 764-767.
- [3] J. Cañedo and A. Skjellum, "Using machine learning to secure IoT systems," 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, 2016, pp. 219-222.
- [4] E. Hodo et al., "Threat analysis of IoT networks using artificial neural network intrusion detection system," 2016 International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, 2016, pp. 1-6.
- [5] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using IoT sensor data," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2017, pp. 87-90.
- [6] S. Haykin, "Introduction" in "Neural Networks and Learning Machines", A. Dworkinm, New Jersey, NJ, USA: Ed. Pearson, 2016, pp. 34-37.
- [7] J.G Hilerá José and V. J Martínez, "Perceptron multinivel" in "Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones, ESP, Ed. Madrid. RA-MA. 1995. Pp 113-115.
- [8] F. Rosenblatt, "The perceptron, a perceiving and recognizing automaton Project Para". Cornell Aeronautical Laboratory, 1957.
- [9] S. Haykin, "Introduction" in "Neural Networks and Learning Machines", A. Dworkinm, New Jersey, NJ, USA: Ed. Pearson, 2016, pp 129-134.
- [10] Diederik P. Kingma and J. Lei Ba, "ADAM: A Method for Stochastic Optimization", 2015 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 2015.

Marca de Agua Invisible en Imágenes Digitales Utilizando IWT-DCT y Mapas Caóticos

Maria de Jesus Torres Mendoza, Clara Cruz Ramos, Rogelio Reyes Reyes, Volodymyr Ponomaryov

Instituto Politécnico Nacional, ESIME Unidad Culhuacán

Av. Santa Ana No. 1000, Col. San Francisco Culhuacán, CP. 04430, Ciudad de México.

Tel. (55) 5729-6000 Ext. 73264 Fax (55) 56562058

email: mariatm235@outlook.com, ccruzra@ipn.mx, rreyesre@ipn.mx, vponomar@ipn.mx

Resumen—Actualmente la tecnología permite la distribución de imágenes digitales de forma rápida y sencilla, por lo que es importante la protección de los derechos de autor en archivos multimedia. Las técnicas de marca de agua son utilizadas comúnmente para la protección de los derechos de autor, sin embargo, todavía es un desafío lograr la imperceptibilidad de la marca de agua y su robustez ante diversos ataques. En el presente trabajo se propone un sistema de marca de agua no ciego basado en la transformada entera de wavelet (IWT) y la transformada coseno discreta (DCT) con el objetivo de mejorar la robustez e imperceptibilidad. La imagen de marca de agua es cifrada por el mapa caótico de Arnold antes de ser insertada revolviendo los píxeles para ofrecer seguridad adicional. Los resultados demuestran una buena calidad en la imagen portadora (38 dB), así como de la imagen de marca de agua recuperada (35 dB) y robustez ante ataques como compresión JPEG, ruido sal y pimienta, re-escalamiento y recorte.

Palabras clave—Marca de agua, Mapa Caótico de Arnold, Transformada entera de wavelet (IWT), Transformada coseno discreta (DCT), YCbCr.

I. INTRODUCCIÓN

Con el crecimiento de la tecnología es común la distribución de datos digitales como video, imágenes y audio. Por lo que es importante la protección de los derechos de autor con el fin de preservar los datos digitales.

Una marca de agua es un proceso de ocultamiento de información, donde la información es incrustada en datos digitales (imagen, video, audio) que puede detectarse o extraerse posteriormente, es utilizada para la protección de derechos de autor, la seguridad de los datos, identificación de la propiedad y autenticación de contenido [1]. Las marcas de agua pueden ser visibles o invisibles. La marca de agua visible es perceptible para el ojo humano, en cambio la marca de agua invisible consiste en incrustar una imagen secreta en una imagen portadora sin que la marca de agua sea perceptible por el ojo humano [2]. La marca de agua se puede realizar en el dominio espacial o el dominio de la transformada. En el dominio espacial, los datos son incrustados directamente en los píxeles, la ventaja es que es simple de implementar y tiene baja complejidad computacional, sin embargo, este esquema no es robusto ya que es sensible ante ataques [3]. En el dominio de la transformada, los datos no son incrustados directamente en los píxeles, por lo que tiene una mayor robustez ante diversos ataques. Entre las transformadas más utilizadas se encuentran la

transformada rápida de Fourier (FFT), la transformada coseno discreta (DCT), la transformada discreta de wavelet (DWT), la transformada entera de wavelet (IWT), entre otras.

Las técnicas de marca de agua tienen el desafío de aumentar la robustez e imperceptibilidad ante una variedad de ataques, tales como compresión JPEG, ruido, filtrado, recorte, entre otros. Estas técnicas pueden ser clasificadas de acuerdo con tres características: robustez, imperceptibilidad y capacidad de inserción, las cuales están relacionadas entre sí ya que al momento de modificar una característica las otras se pueden ver afectadas [4]. El PSNR (Peak Signal to Noise Ratio) es una métrica utilizada para evaluar la calidad de la imagen.

Winarno, et al. [5], proponen el uso de la DCT basada en sub-bloques de 8*8 píxeles, recopilando los coeficientes DC para ser transformados con la DWT, e introduce la marca de agua binaria en la sub-banda LL (Low-Low). Sin, embargo el PSNR obtenido es bajo, utilizando una imagen portadora en escala de grises. Susanto, et al. [6], proponen un método de marca de agua combinando la Transformada coseno discreta (DCT) y la transformada Wavelet de Haar (HWT) basada en dos niveles utilizando bloques de 4*4 píxeles, para imágenes en escala de grises, obteniendo un PSNR alto. Sin embargo, la robustez ante ataques como ruido de sal y pimienta, recorte, es bajo. D. Bansal, et al. [7], proponen un sistema de marca de agua para imágenes a color, utilizando una marca de agua binaria y el espacio de color RGB en la portadora. El canal de color rojo de la imagen portadora se utiliza para aplicar la DWT para insertar la marca de agua. Posteriormente, la DCT se implementa en el bloque de 8*8 píxeles de la imagen. En [8], M. Jamali et al, proponen un sistema para imágenes a color utilizando el espacio de color YUV concentrando la información de la marca de agua binaria en el canal Y. La inserción de la marca de agua se realiza en los coeficientes DCT de las sub-bandas de wavelet.

En el presente trabajo, se utiliza la transformada entera de wavelet. La transformada entera de wavelet (IWT) asigna un conjunto de datos enteros con otro conjunto de datos enteros a diferencia de la DWT la cual convierte los valores de punto flotante a entero por proceso de truncamiento. El truncamiento resulta en la pérdida de datos. La sub-banda LL de la transformada entera de wavelet parece ser una copia cercana con una escala menor de la imagen original, mientras que, en el caso de transformada discreta de wavelet, la sub-banda LL resultante está distorsionada [9]. La ventaja de la IWT es que

consume menos tiempo que la DWT, ya que la IWT exige menos longitud de datos evitando problemas con la precisión de punto flotante de la DWT.

La mayoría de las técnicas utilizan imágenes binarias como marca de agua e imágenes en escala de grises como portadoras, además de que la robustez ante algunos ataques es baja. En este artículo se propone un sistema de marca de agua no ciego para imágenes a color, en este método se realiza la inserción en el dominio de la transformada combinando la IWT y DCT para obtener una alta imperceptibilidad y robustez para ataques como ruido de sal y pimienta, recorte, re-escalamiento, y compresión JPEG. La transformación de la IWT se aplicó en el segundo nivel. Además, proporcionando seguridad adicional revolviendo los píxeles de la marca de agua mediante el mapa caótico de Arnold antes de ser insertada, y revolviendo la sub-banda LL2 de la portadora después de aplicar la transformada DCT, lo que hace que los píxeles se distribuyan uniformemente en todo el espacio.

II. MÉTODO PROPUESTO

El sistema propuesto cuenta con dos etapas: la inserción de la marca de agua y la extracción de la marca de agua. El esquema de extracción utiliza un método no ciego ya que necesita de la imagen portadora y la imagen marcada.

A. Etapa de Inserción de la Marca de Agua

Esta etapa consiste en insertar los bits de la marca de agua en la imagen portadora RGB. Antes de insertar la marca de agua, los bits de la imagen de marca se revuelven utilizando el mapa caótico de Arnold. El método de inserción está basado en el dominio de la transformada combinando IWT en el segundo nivel y la DCT. La inserción se realiza en N píxeles por bloque, donde N equivale a 4. Los pasos para la inserción de la marca de agua son los siguientes:

1. La imagen portadora es transformada al espacio YCbCr.
2. Al canal de luminancia Y se le aplica la transformada IWT obteniendo las sub-bandas LL1, HL1, LH1, HH1.
3. A la sub-banda LL1 se le aplica nuevamente IWT obteniendo las sub-bandas LL2, HL2, LH2, y HH2.
4. Se selecciona la sub-banda LL2 de tamaño $m_2 \times n_2$, y se divide en sub-bloques de 8×8 píxeles aplicando DCT.

Para obtener el número de bloques se utiliza (1), donde B es el número de bloques.

$$B = \left(\frac{m_2}{8} * \frac{n_2}{8} \right) \quad (1)$$

5. Se revuelven los píxeles de la sub-banda LL2-DCT, utilizando el mapa caótico de Arnold [10].
6. Se revuelven los píxeles de marca de agua utilizando el mapa caótico de Arnold y se convierte en un vector V_w .
7. Se insertan los bits de marca de agua con la ecuación (2), en los N coeficientes del bloque, desde las frecuencias bajas a las altas en forma de zig-zag como se muestra en la Fig. 1, sin tomar el coeficiente DC.

$$AC' = AC + \alpha * V_w \quad (2)$$

Donde AC son los coeficientes seleccionados DCT, α es el factor de escala, V_w indica el vector de la marca de agua.

8. Se aplica el mapa caótico de Arnold inverso a los píxeles de la sub-banda LL2-DCT' que contiene la marca de agua.
9. Se aplica la IDCT a cada bloque de la sub-banda LL2'.
10. Se aplica la transformada inversa de IWT dos veces para obtener el canal de luminancia con la marca Y'.

DC	AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC	AC

Fig. 1. Inserción en los coeficientes de la DCT.

11. Finalmente, se transforma el canal de luminancia marcado Y' junto con los canales de crominancia (obtenidos en el paso 1) del espacio de color YCbCr al espacio de color RGB, para obtener la imagen marcada.

El diagrama a bloques del proceso de inserción de la marca de agua se muestra en la Fig. 2.

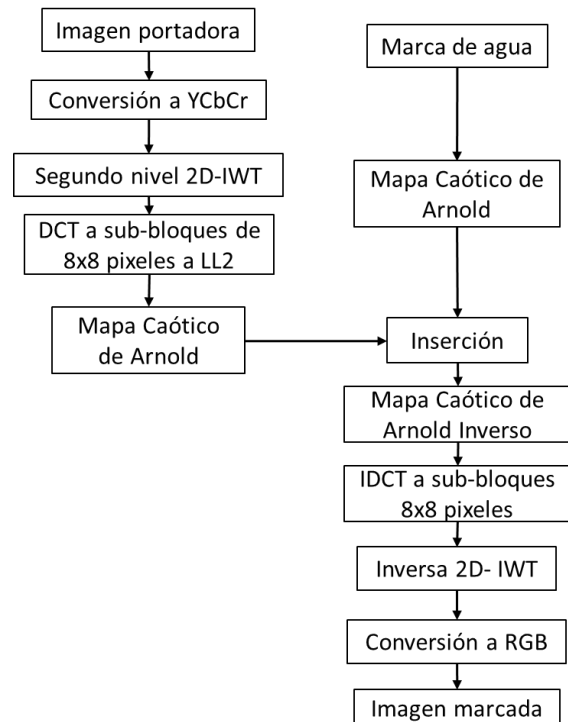


Fig. 2. Diagrama del proceso de inserción de la marca de agua.

B. Etapa de Extracción de la marca de agua

Esta etapa consiste en extraer los bits de la imagen marcada RGB. Para la extracción se necesita de la imagen portadora y la imagen marcada ya que es un método no ciego. Los pasos para la extracción de la marca de agua son los siguientes:

1. La imagen portadora es transformada al espacio YCbCr.
 2. Al canal de luminancia Y se le aplica la transformada IWT obteniendo las sub-bandas LL1, HL1, LH1, HH1.
 3. A la sub-banda LL1 se le aplica nuevamente IWT obteniendo las sub-bandas LL2, HL2, LH2, y HH2.
 4. Se selecciona la sub-banda LL2 de tamaño $m_2 \times n_2$, y se divide en sub-bloques de 8×8 pixeles aplicando DCT.
- Para obtener el número de bloques se utiliza (1), donde B es el número de bloques.
5. Se revuelven los pixeles de la sub-banda LL2-DCT, utilizando el mapa caótico de Arnold.
 6. La imagen marcada es transformada al espacio YCbCr.
 7. Al canal de luminancia Y' se le aplica la transformada IWT obteniendo las sub-bandas LL1', HL1', LH1', HH1'.
 8. A la sub-banda LL1' se le aplica nuevamente IWT obteniendo las sub-bandas LL2', HL2', LH2', y HH2'.
 9. Se selecciona la sub-banda LL2' de tamaño $m_2 \times n_2$, y se divide en sub-bloques de 8×8 pixeles aplicando DCT. Para obtener el número de bloques se utiliza (1), donde B es el número de bloques.
 10. Se revuelven los pixeles de la sub-banda LL2-DCT', utilizando el mapa caótico de Arnold.
 11. Se extraen los bits de marca de agua con la ecuación (3), en los N coeficientes del bloque, desde las frecuencias bajas a las altas como se muestra en la figura 1, sin tomar el coeficiente DC.

$$V_w = (AC' - AC)/\alpha \quad (3)$$

Donde AC' son los coeficientes DCT de la imagen marcada, AC son los coeficientes DCT de la imagen original, α es el factor de escala, V_w indica el vector de la marca de agua.

12. El vector de la marca de agua se transforma a matriz de tamaño $r \times r$ con la ecuación (4).

$$r = \sqrt{B * N} \quad (4)$$

13. Se aplica el mapa caótico de Arnold inverso a los pixeles de la marca de agua obtenidos en el paso anterior.

El diagrama del proceso de extracción de la marca de agua se muestra en la Fig. 3.

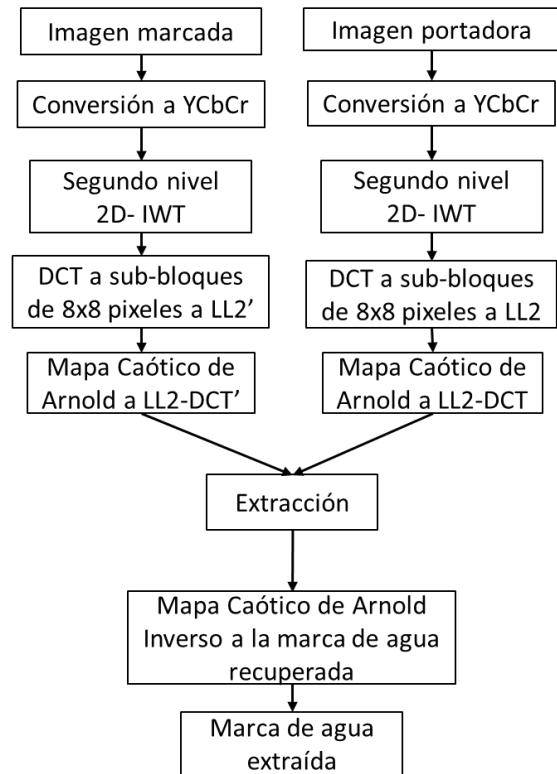


Fig. 3. Diagrama a bloques del proceso de extracción de la marca de agua.

III. PRUEBAS Y RESULTADOS

Para evaluar el sistema propuesto se utilizaron imágenes a color de 1024×1024 pixeles como portadoras, las cuales son comúnmente utilizadas en el área de procesamiento digital de imágenes, para el caso de las imágenes de marca de agua se utilizaron imágenes a escala de grises con una resolución de 64×64 pixeles. El sistema se desarrolló utilizando la plataforma de MathWorks Matlab® R2017b. Las imágenes portadoras se muestran en la Fig. 4, y la imagen utilizada como marca de agua se muestra en la Fig. 5.

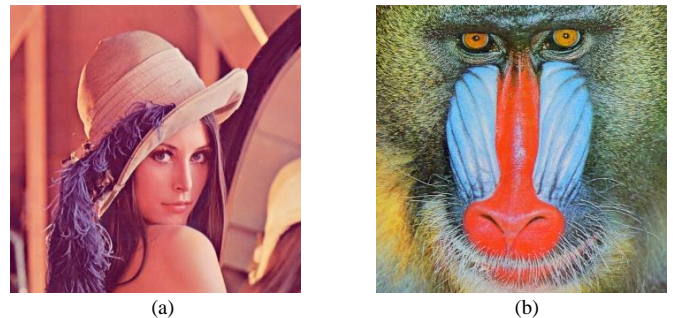


Fig. 4. Imágenes portadoras: (a) Lena; (b) Baboon.



Fig. 5. Marca de agua utilizada.

Se utilizaron métricas para medir la calidad de las imágenes y la resistencia que tiene la marca de agua ante diversos ataques, las métricas utilizadas son la Relación señal a ruido pico (PSNR), el Índice de similitud estructural (SSIM) y el coeficiente de correlación (CC). Estas métricas permiten evaluar la imperceptibilidad y robustez del sistema. La tabla I, muestra los resultados obtenidos del PSNR y SSIM de las imágenes marcadas.

TABLA I. PSNR Y SSIM OBTENIDOS DE LAS IMÁGENES MARCADAS

Imágenes portadoras	Imágenes Marcadas	
	PSNR	SSIM
<i>Lena</i>	39.44 dB	0.9976
<i>Baboon</i>	38.23 dB	0.9935

Las Figs. 6 y 7 muestran las imágenes de Lena y Baboon marcadas, la marca de agua cifrada con el mapa caótico de Arnold, y la marca de agua recuperada.



Fig. 6. Resultados de la imagen Lena como portadora: (a) Imagen marcada; (b) Imagen de marca de agua cifrada; (c) Marca de agua recuperada.

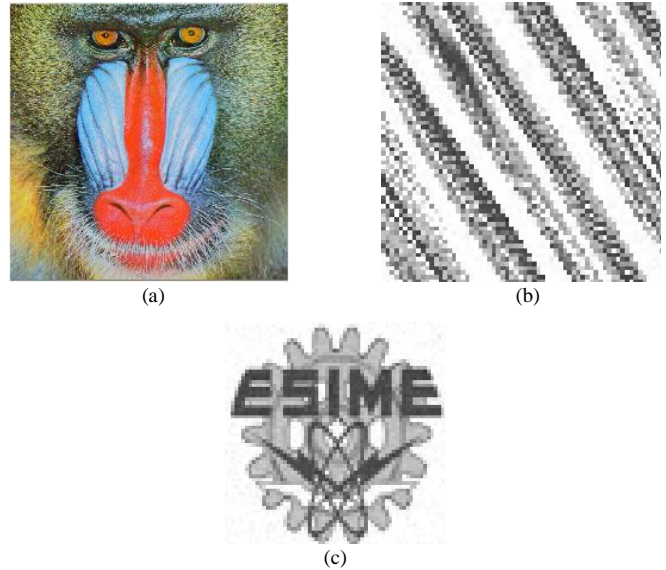


Fig. 7. Resultados de la imagen Baboon como portadora: (a) Imagen marcada; (b) Imagen de marca de agua cifrada; (c) Marca de agua recuperada.

El sistema propuesto muestra buena calidad en las imágenes marcadas ya que el PSNR es mayor a 35 dB.

Para evaluar la robustez del sistema se probaron ataques de compresión, ruido de sal y pimienta, recorte y re-escalamiento a las imágenes portadoras protegidas, los resultados de la extracción se obtendrán con el coeficiente de correlación, comparando la marca de agua original con la marca de agua extraída. La Fig. 8 muestra los coeficientes de correlación para diferentes factores de calidad de compresión JPEG para la marca de agua recuperada de las imágenes de Lena y Baboon. La tabla II muestra los ataques que se hicieron a las imágenes portadoras y el coeficiente de correlación de la marca de agua recuperada.

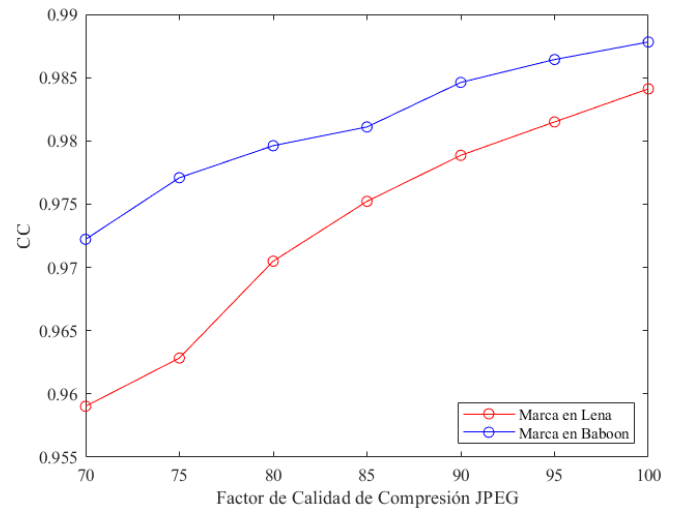










Fig. 8. Coeficiente de Correlación de la marca de agua obtenido para el ataque de compresión JPEG.

TABLA II. RESULTADOS DEL COEFICIENTE DE CORRELACIÓN ENTRE LA MARCA DE AGUA ORIGINAL Y LA MARCA DE AGUA EXTRAÍDA

Ataque	Lena	Baboon
Ruido de Sal y pimienta (dens=0.001)	 CC = 0.9792	 CC = 0.9835
Recorte (25%)	 CC = 0.8209	 CC = 0.8267
Re-escalamiento [512x512]	 CC = 0.9257	 CC = 0.9052
Re-escalamiento [2048x2048]	 CC = 0.9749	 CC = 0.9810

En [7] el promedio del PSNR obtenido en las distintas imágenes marcadas es bajo (35 dB), el cual es menor al PSNR promedio obtenido por el sistema aquí propuesto (38 dB), debido a que en [7] se trabaja en el canal rojo (RGB) por lo que la robustez de su sistema es menor.

Para el ataque de compresión JPEG con factor de calidad 90 en [8] se obtuvo un coeficiente de correlación promedio de 0.9005 de la imagen de marca de agua extraída. En cambio, en el sistema propuesto se obtuvo un coeficiente de correlación promedio de 0.9651 después del ataque de compresión JPEG con factor de calidad 90. Por lo que el sistema propuesto en este trabajo obtuvo mejores resultados para el ataque de compresión JPEG en comparación con [8].

IV. CONCLUSIONES

En el presente trabajo se propone un método para la inserción de marca de agua en imágenes a color no ciego, basado en la Transformada Entera de Wavelet y la Transformada Coseno Discreta, la combinación de estas dos transformadas permitió al sistema tener una buena robustez e imperceptibilidad. Se proporcionó seguridad adicional revolviendo los píxeles de la marca de agua con el mapa caótico de Arnold, también se revolviaron los píxeles de la sub-banda LL2-DCT de la imagen portadora antes de insertar la marca de agua en esta sub-banda lo que permite que los píxeles modificados se distribuyan a lo largo y ancho de LL2, mejorando la calidad de la imagen protegida. El sistema obtuvo valores de PSNR superiores a 35dB en las imágenes marcadas de Lena y Baboon, lo que indica una buena calidad, además se demostró que la marca de agua permanece invisible para el ojo humano. La marca de agua fue resistente ante los ataques de ruido de sal y pimienta, recorte, re-escalamiento, y en especial a los ataques de compresión JPEG.

AGRADECIMIENTOS

Agradecemos al Instituto Politécnico Nacional, al Consejo Nacional de Ciencia y Tecnología de México, a la Comisión de Operación y Fomento de Actividades Académicas (COFAA) del IPN y a la Beca de Estímulo Institucional de Formación de Investigadores (BEIFI) por el apoyo otorgado para el desarrollo de este trabajo.

REFERENCIAS

- [1] Chun-Shien Lu. *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*. USA: Idea Group Publishing, 2005.
- [2] S. Dhiman and O. Singh, "Analysis of Visible and Invisible Image Watermarking," *International Journal of Computer Applications*, vol.147, no.3, pp. 36-38, August 2016.
- [3] M. Saqib and S. Naaz, "Spatial and Frequency Domain Digital Image Watermarking Techniques for Copyright Protection," *International Journal of Engineering Science and Technology*, vol. 9, pp. 691-699, June 2017.
- [4] M. D. Gupta. *Watermarking: Volume 2*. Croatia: IntechOpen, 2012, pp. 46.
- [5] A. Winamo, D. R. I. M. Setiadi, A. A. Arrasyid, C. A. Sari and E. H. Rachmawanto, "Image watermarking using low wavelet subband based on 8x8 sub-block DCT," *International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, pp. 11-15, October 2017.
- [6] A. Susanto, D. R. I. M. Setiadi, E. H. Rachmawanto and C. A. Sari, "A Robust Non-Blind Image Watermarking Method Using 2-Level HWT-DCT," *International Seminar on Application for Technology of Information and Communication*, Semarang, pp. 304-308, September 2018.
- [7] D. Bansal and M. Mathuria, "Color image dual watermarking using DCT and DWT combine approach," *International Conference on Trends in Electronics and Informatics (ICEI)*, Tirunelveli, 2017, pp. 630-634.
- [8] M. Jamali, M. Bagheri, N. Karimi and S. Samavi, "Robustness and Imperceptibility Enhancement in Watermarked Images by Color Transformation," *2020 25th International Computer Conference, Computer Society of Iran (CSICC)*, Tehran, Iran, 2020, pp. 1-5.
- [9] R. Punidha and M. Sivaram, "Integer Wavelet Transform Based Approach for High Robustness of Audio Signal Transmission," *International Journal of Pure and Applied Mathematics*, vol.116, pp. 295-304, 2017.
- [10] D. R. Alshibani and R. Shaker, "Implementation of Gray Image Encryption using Multi-Level of Permutation and Substitution," *International Journal of Applied Information Systems*, vol. 10, no. 1, pp. 25-30, November 2015.

Sistema Esteganográfico Reversible en Imágenes Estereoscópicas por Modificación del Histograma

Jocelyn Guadalupe Arenas Hernández, Diego Denilson Cañada Meza
 Clara Cruz Ramos, Rogelio Reyes Reyes, José Antonio Loaiza Brito
Instituto Politécnico Nacional, ESIME Unidad Culhuacán, Ingeniería en Computación
 Tel. (55) 5729-6000 Ext. 73264 Fax (55) 56562058

Av. Santa Ana No.1000, Col San Francisco Culhuacán, CP. 04430, Ciudad de México, México
 e-mail: jocelynarenashdez@gmail.com, diego.denil.cm@gmail.com, ccruzra@ipn.mx, reyesre@ipn.mx, jloaiza@ipn.mx

Resumen-- Hoy en día, en la mayoría de las aplicaciones es deseable que la comunicación se realice en secreto, tal es el caso de las comunicaciones corporativas, legales y médicas, por mencionar algunas. La esteganografía es una ciencia que estudia las diferentes técnicas de ocultamiento de información en objetos conocidos como medios portadores, los cuales permiten transportar información oculta dentro de ellos. Sin embargo, en la mayoría de las técnicas esteganográficas tradicionales no es posible recuperar el medio portador, debido a que sufren alteraciones en la etapa de inserción y/o extracción de la información. En este trabajo se propone un sistema de ocultamiento de datos reversible empleando el método de modificación del histograma aplicado a imágenes estereoscópicas, logrando así recuperar el mensaje original y el medio portador sin alteraciones. Los resultados experimentales demuestran una alta imperceptibilidad de la información oculta en las imágenes modificadas y la recuperación total del medio portador, obteniendo valores promedio de PSNR de 60 dB e infinito respectivamente.

Palabras clave – Esteganografía, esteganografía reversible, imagen estereoscópica, histograma.

I. INTRODUCCIÓN

La esteganografía se usa comúnmente para garantizar la entrega exitosa de los datos secretos cifrados, mientras que el desplazamiento de datos del histograma a menudo se implementa para permitir el ocultamiento de datos reversible [1]. El ocultamiento de datos reversible es un campo emergente, donde la información está incrustada dentro del medio portador. El ocultamiento reversible de información permite que, después de que se haya extraído totalmente y de forma correcta la información insertada en el medio portador sea posible volver a recuperar el medio de transporte original sin ninguna alteración [2].

Las técnicas esteganográficas presentan dos etapas fundamentales: la etapa de *inserción* del mensaje en el medio portador y, la de *detección y/o extracción* del mensaje oculto. El medio portador se procesa mediante alguna técnica esteganográfica para ocultar la información. En la etapa de extracción del mensaje, se escanea el medio portador en el orden que se insertó la información durante la primera etapa. Cuando se detecta el mensaje, se comienza a extraer y una vez extraído todo el mensaje, se procesa para que este pueda ser visualizado con una representación adecuada [3].

Una imagen estereoscópica es una imagen generada a partir de la unión de dos imágenes bidimensionales con diferentes ángulos entre ellas, lo que genera un efecto de profundidad, dando como resultado una imagen tridimensional [4]. Por su composición, este tipo de imagen duplica su cantidad de píxeles incrementando su capacidad de ocultamiento de información, la cual es de gran utilidad para los sistemas esteganográficos. En los últimos años, la relevancia de las imágenes estereoscópicas ha aumentado en sectores como: simulación 3D, fotogrametría, educación, tecnología aeroespacial, entre otros. De acuerdo con el informe Fortune Business Insight publicado en septiembre de 2019 [5], se estima que la tasa de crecimiento anual tendrá un valor del 42.20% en un pronóstico al año 2026 en el campo de realidad virtual.

Noriega L. R. [6], propone un método para ocultar información en imágenes estereoscópicas utilizando la técnica de esteganografía LSB y un modelo binocular de distorsión apenas perceptible (BJND) que incorpora la capacidad máxima de carga útil evitando artefactos visuales y una generación 3D inexacta. Los resultados presentan una capacidad de carga útil de hasta 1.87Mb sin degradar la calidad visual.

Priya G. [7], propone un esquema de mapa de ubicación utilizando solo un bit que permite mejorar la calidad de la imagen, donde la recuperación de datos secretos incrustados no es posible a menos que el punto máximo y el punto cero del histograma se transmitan al receptor. El sistema realiza el corrimiento de histograma a partir del punto pico para insertar el mapa de ubicación junto con la carga útil.

Thodi M. y Rodríguez J. [8], proponen una técnica de desplazamiento de histograma como alternativa a la incrustación del mapa de ubicación, mejorando el rendimiento de distorsión a bajas capacidades de inclusión y mitiga el problema de control de capacidad. Los resultados experimentales muestran que la expansión del error de predicción duplica la capacidad máxima de incrustación en comparación con la expansión de diferencia.

II. SISTEMA PROPUESTO

En la Figura 1 se muestra la propuesta del diagrama a bloques general del sistema esteganográfico reversible, el cual consta de dos bloques principales: el primero hace referencia a la etapa de *inserción* de la información a ocultar y el segundo a la etapa de *extracción* del mensaje incrustado a partir de una llave de seguridad y la estegoimagen recibida.

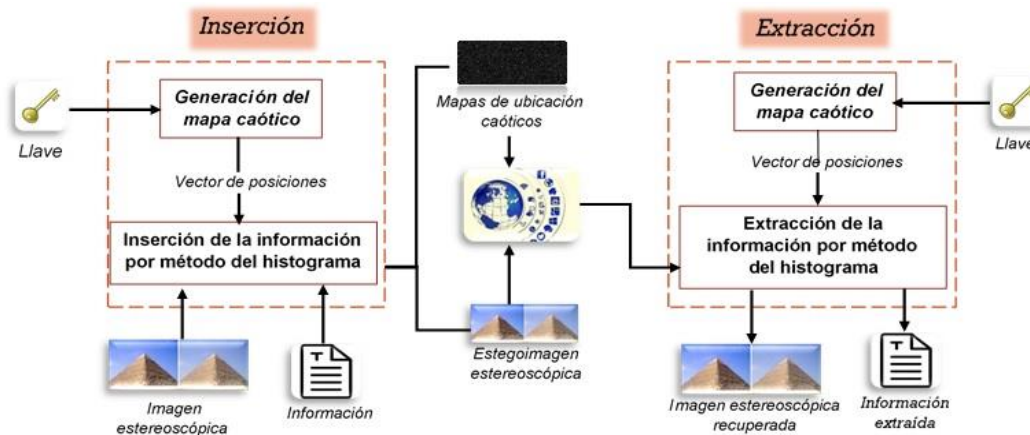


Fig. 1. Diagrama a bloques general del sistema propuesto

El enfoque reversible que ofrece el método de modificación del histograma permite que la información insertada y la imagen portadora se recuperen correctamente después de su procesamiento en las etapas de inserción y extracción de la información. De igual manera, el ocultamiento de información a través de este método ayuda a generar un mapa de ubicación caótico (*MUC*) que se envía a través de un medio de comunicación hacia el receptor.

A. Etapa de inserción

La figura 2 muestra una descripción detallada del proceso que se lleva a cabo durante la etapa de inserción, en la que se ingresa la información a insertar, la llave de seguridad *K* y la imagen estereoscópica *I_{LR}* como parámetros de entrada.

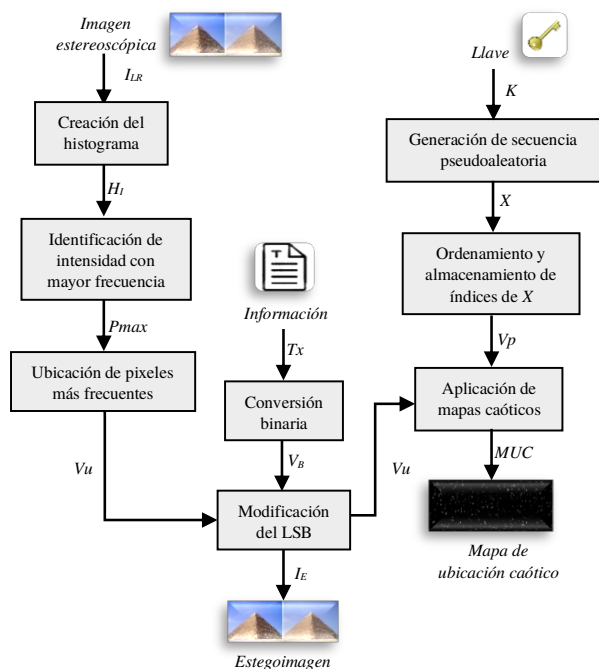


Fig. 2. Diagrama a bloques de la etapa de inserción

1) Creación del histograma

En el bloque de creación del histograma se ingresa como parámetro de entrada la imagen estereoscópica *I_{LR}* y se genera su histograma *H_I* como se muestra a continuación:

$$H_{LR}^R = imhist(I_{LR}(:, :, 1)) \quad (1)$$

donde *I_{LR}* representa la imagen estereoscópica izquierda (*L*) y derecha (*R*), *imhist()* representa el cálculo del histograma de la imagen, *H_{LR}^R* representa el histograma obtenido para el canal rojo. En la figura 3 se muestra la gráfica de los histogramas de la imagen estereoscópica.

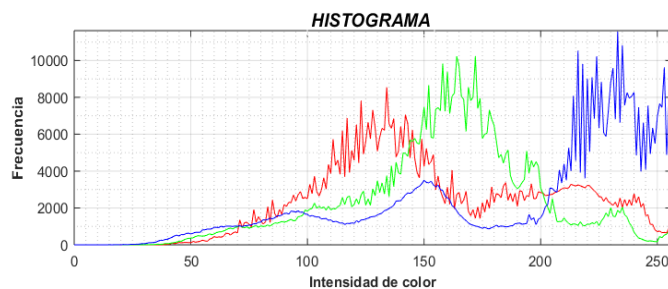


Fig. 3. Histograma de cada canal de la imagen estereoscópica

2) Identificación de la intensidad con mayor frecuencia

La identificación de la intensidad con mayor frecuencia se obtiene empleando (2):

$$P_{m\acute{a}x} = \max(H_{LR}^R) \quad (2)$$

donde *P_{máx}* es el valor de la intensidad con el mayor número de apariciones en el canal rojo de la imagen estereoscópica. Para los canales verde y azul, se utilizará el mismo valor de *P_{máx}* obtenido del canal rojo.

3) Ubicación de los píxeles más frecuentes

La figura 4 ejemplifica como se ubican las intensidades de mayor frecuencia en cada uno de los canales.

134		134	134		134	134	134		134	134		134			134		
134		134	134		134		134	134		134	134	134	134	134	134	134	134
134			134									134		134	134	134	
	134				134		134			134					134		
	134				134	134	134	134	134	134	134				134		

Fig. 4. Ubicación de las intensidades más frecuentes con $P_{m\acute{a}x}=134$ para el canal (a) rojo, (b) verde y (c) azul.

Las posiciones de las zonas donde se ubicó el valor 134, se almacenan en el vector de ubicaciones Vu , el cual será utilizado posteriormente en la etapa *modificación del LSB*.

4) *Conversión binaria de la información*

La información Tx que se ingresa al sistema está compuesta por letras, números y caracteres. En el bloque de conversión binaria, Tx se transforma al sistema de ceros y unos con base al código ASCII.

La figura 5 es un ejemplo de la conversión de un texto a binario.

H	O	L	A
01001000	01001111	01001100	01000001

Fig. 5. Conversión de la palabra 'HOLA' a binario

V_B almacena la secuencia binaria generada por los caracteres de Tx , posteriormente se envía al bloque de modificación LSB.

5) *Modificación del LSB*

En la figura 6 se muestran los valores de intensidad donde se modifican los LSB de las intensidades.

134 ₀		134 ₀	134 ₀		135 ₁	135 ₁	134 ₀		134 ₀	134 ₀		134			134		
135 ₁		134 ₀	134 ₀		135 ₁		135 ₁	135 ₁		135 ₁	134 ₀	134 ₀	134	134	134	134	134
134 ₀			135 ₁									135 ₁		134	134	134	
	134 ₀				134 ₀		134 ₀			134 ₀			134			134	
	135 ₁				134 ₀	135 ₁	134 ₀	135 ₁	134 ₀	134 ₀	134 ₀		134			134	

Fig. 6. (a)-(c) valores P_{max} modificados para cada canal R, G y B respectivamente

La intensidad cambia a 135 cuando se inserta un 1 y conserva su mismo valor cuando se ingresa un 0. La estegoimagen I_E es la imagen resultante con información oculta.

6) *Generación de secuencia pseudoaleatoria*

La llave de seguridad K es el parámetro de entrada con el que se genera la secuencia pseudoaleatoria X , compuesta por números decimales aleatorios, como se muestra en (3):

$$X = rand_K(1, MN) \tag{3}$$

donde $rand_K$ representa el método de generación de números pseudoaleatorios basado en la llave K , MN es el tamaño de la secuencia que se determina al multiplicar el largo por el ancho de la imagen estereoscópica.

La figura 7 es un ejemplo de una secuencia pseudoaleatoria.

	1	2	3	4	5	...
X	0.37	0.49	0.24	0.78	0.11	...

Fig. 7. Generación de secuencia pseudoaleatoria X

7) *Ordenamiento y almacenamiento de índices de X*

La secuencia X entra a un proceso de ordenamiento ascendente de los valores que la componen, lo cual se realiza como en (4):

$$[X_o, Vp] = sort(X, 'ascend') \tag{4}$$

donde $sort$ es un método de ordenamiento ascendente, X_o es el vector X ordenado y, finalmente Vp es el vector de posiciones que contiene los nuevos índices de la secuencia X .

En la figura 8 se muestra el ejemplo de ordenamiento de la secuencia pseudoaleatoria X .

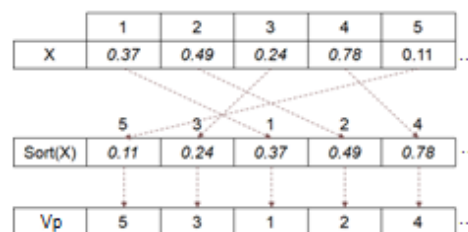


Fig. 8. Generación y mezcla caótica de la secuencia pseudoaleatoria

8) *Aplicación de mapas caóticos*

Se implementan los mapas caóticos de una dimensión para el cifrado de imágenes basada en la mezcla total; lo que ofrece mejores rendimientos y rangos caóticos más grandes [9]. Para la aplicación de mapas caóticos se reciben los Vu de la imagen y el Vp que se obtuvieron en los bloques anteriores.

En la figura 9 se representa como quedan los MUC.

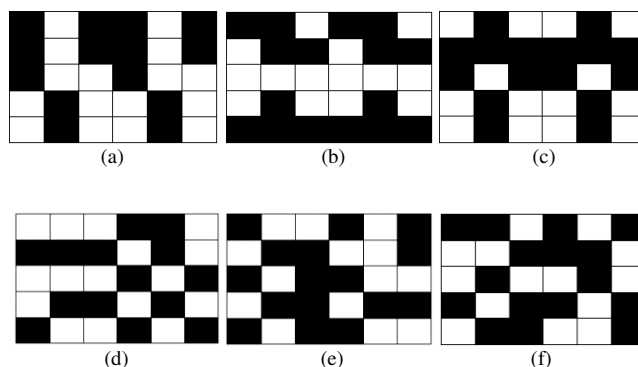


Fig. 9. (a)-(c) posiciones de los pixeles donde se modifica el LSB; (d)-(f) posiciones de los pixeles después de aplicar la mezcla caótica (MUC)

Los mapas caóticos reordenan cada uno de los pixeles del Vu recibido en función al Vp y se consigue así el MUC de cada canal. Se observa en la figura 9 como al generar los MUC de manera aleatoria, se vuelve impredecible la existencia de algún orden.

B. Etapa de extracción

La extracción de la información es la etapa final del sistema esteganográfico. En la figura 10 se visualiza el esquema de la etapa de *extracción* de información, donde de modo general se recibe la llave de seguridad K , la estegoimagen I_E y el MUC obtenidos durante la etapa de inserción.

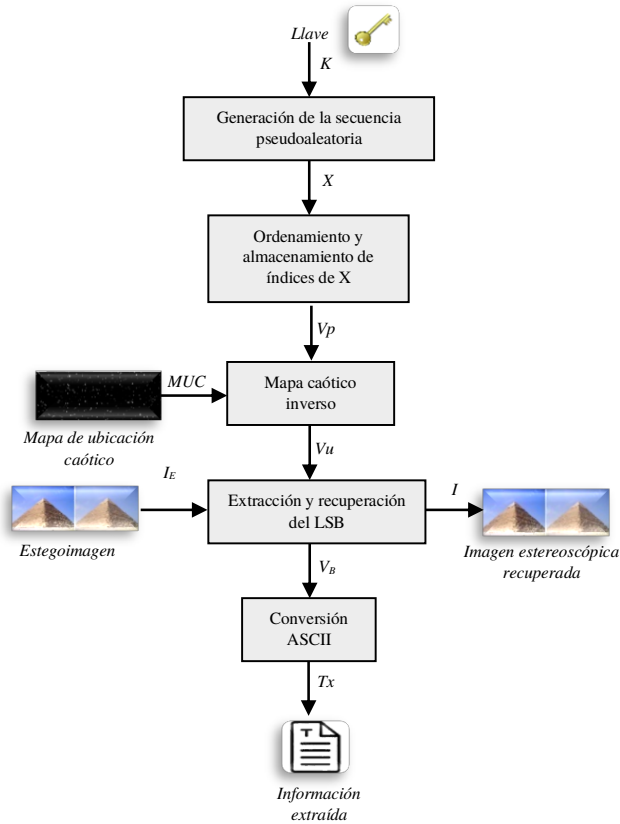


Fig. 10. Diagrama a bloques de la etapa de extracción

En la etapa de extracción, en los bloques *generación de la secuencia pseudoaleatoria* y, *el ordenamiento y almacenamiento de índices de X*, se introduce exactamente la misma llave de seguridad K que se usó en la etapa de inserción.

El proceso de creación del vector de posiciones V_p es idéntico al empleado en la etapa de inserción. El V_p obtenido en esta etapa y el MUC que envía el emisor, se ingresan al bloque *Mapa caótico inverso* para reconstruir los V_u recuperando así el orden en el que la información fue incrustada.

En la figura 11 se muestra como resultado los V_u reconstruidos en función de los MUC y de los V_p para cada canal de la imagen estereoscópica.

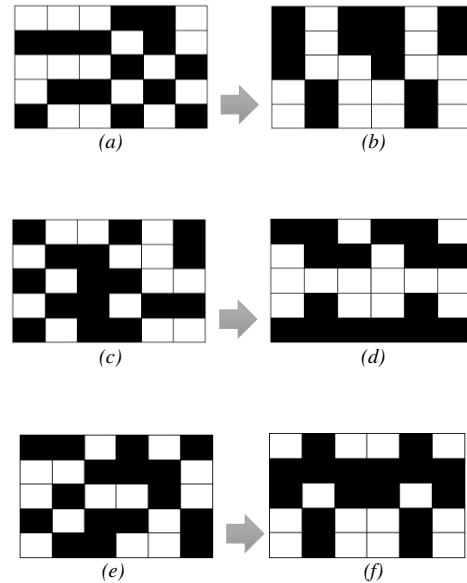


Fig. 11. Reconstrucción de los V_u : (a) MUC y (b) V_u del canal rojo; (c) MUC y (d) V_u del canal verde; (e) MUC y (f) V_u del canal azul

Los V_u reconstruidos se transfieren al bloque *extracción y recuperación del LSB* el cual también recibe la estegoimagen I_E . Finalmente, a la salida se tendrá la imagen estereoscópica original recuperada y la información extraída en binario. Dicha información en binario tiene que pasar por el bloque de conversión con el código ASCII para tener una representación en texto plano.

III. PRUEBAS Y RESULTADOS

El sistema propuesto se desarrolló con el software *MathWorks Matlab® R2019*, en una computadora con procesador Intel (R), Core (TM) i3 CPU 1.9GHz, RAM de 6.00GB y S. O. de 64 bits, Windows 10 Pro, Microsoft.

Las imágenes estereoscópicas utilizadas en la implementación del sistema computacional propuesto se descargaron de la base de datos *Middlebury* [10].

En las figuras 12, 13 y 14 se observan las pruebas experimentales y resultados obtenidos con el sistema esteganográfico reversible propuesto.

En las figuras se aprecia que a simple vista es imperceptible algún cambio entre las imágenes estereoscópicas “art”, “cones” y “dolls” (figs. 12a, 13a y 14a) respecto sus estegoimágenes (figs. 12b, 13b y 14b).

Debajo de las imágenes estereoscópicas y de las estegoimágenes, se visualizan los V_u y los MUC correspondientes a cada imagen. De las imágenes estereoscópicas (figs. 12a, 13a y 14a) se obtienen sus vectores de ubicación V_u (figs. 12c, 13c y 14c) y sus mapas de ubicación caóticos MUC (figs. 12d, 13d y 14d).

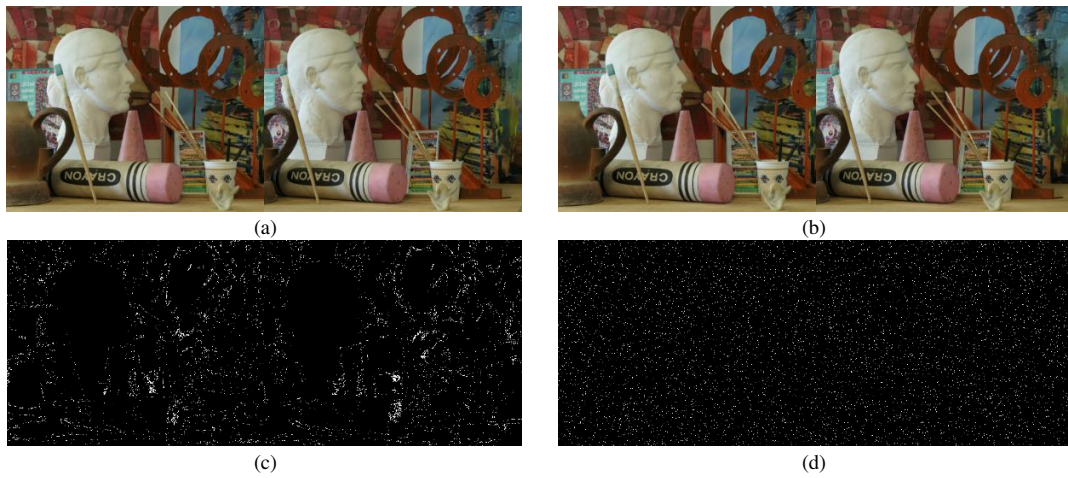


Fig. 12. Pruebas con “art” (a) imagen estereoscópica, (b) estegoimagen, (c) posiciones originales de los pixeles modificados y (d) *MUC*

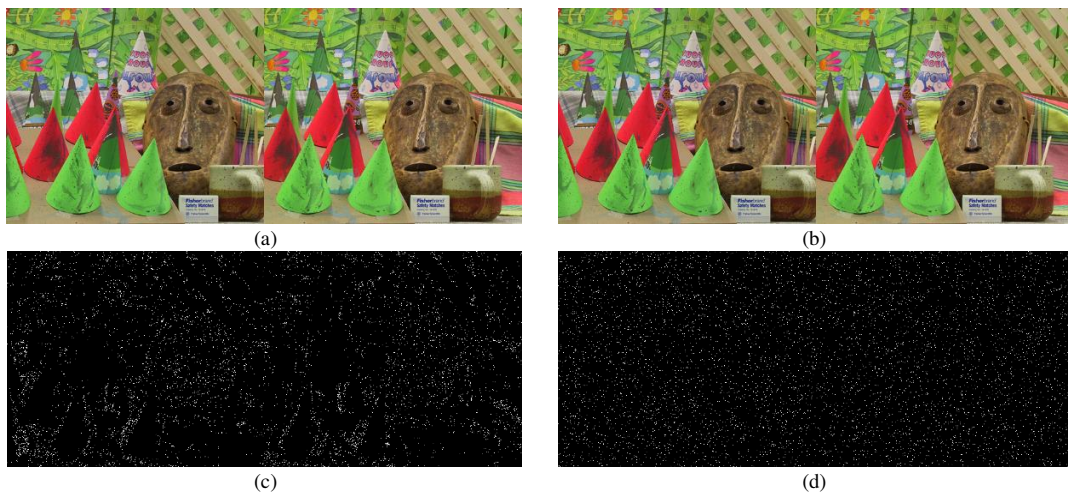


Fig. 13. Pruebas con “cones” (a) imagen estereoscópica, (b) estegoimagen, (c) posiciones originales de los pixeles modificados y (d) *MUC*

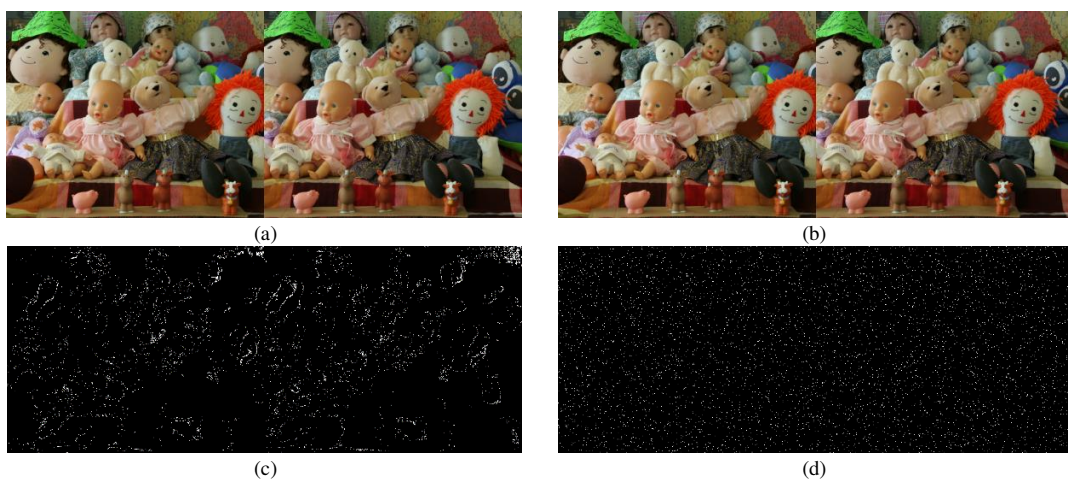


Fig. 14. Pruebas con “dolls” (a) imagen estereoscópica, (b) estegoimagen, (c) posiciones originales de los pixeles modificados y (d) *MUC*

En la tabla 1 se muestran los resultados para la evaluación objetiva de la imperceptibilidad en las imágenes modificadas mediante la evaluación del error cuadrático medio (MSE), la relación señal a ruido pico (PSNR) y el índice de similitud estructural (SSIM).

Tabla 1. Resultados del cálculo del MSE, PSNR y SSIM de la estegoimagen.

<i>Imagen estereoscópica</i>	<i>MSE</i>	<i>PSNR</i>	<i>SSIM</i>
<i>Art</i>	0.0085	68.86 dB	0.9999
<i>Cones</i>	0.0085	68.84 dB	0.9999
<i>Dolls</i>	0.0077	69.28 dB	0.9999

Los resultados obtenidos comprueban que las imágenes sufrieron pequeños cambios en su contenido al insertar la información. En el caso de la prueba con la imagen “art” se tiene un MSE con valor de 0.0085, el PSNR de 68.86 dB y un SSIM de 0.9999, estos valores indican que la imagen tuvo una alta imperceptibilidad.

Recuperación del contenido

En la etapa de extracción, las imágenes estereoscópicas recuperadas son totalmente idénticas a las imágenes originales. En la tabla 2 se muestran los resultados obtenidos para el MSE, PSNR y SSIM de las imágenes recuperadas después de la extracción de la información.

Tabla 2. Resultados del cálculo del MSE, PSNR y SSIM de la imagen estereoscópica recuperada.

<i>Imagen estereoscópica</i>	<i>MSE</i>	<i>PSNR</i>	<i>SSIM</i>
<i>Art</i>	0	Infinito	1
<i>Cones</i>	0	Infinito	1
<i>Dolls</i>	0	Infinito	1

El valor 0 obtenido en el MSE indica que la imagen recuperada es completamente idéntica a la imagen original, por lo tanto, es justificado que el valor del PSNR resulte infinito. Finalmente, el índice con valor de 1 para el SSIM indica una igualdad total entre la imagen estereoscópica original y la imagen recuperada.

IV. CONCLUSIONES

En este artículo se propone un sistema esteganográfico reversible, en el cual las imágenes estereoscópicas utilizadas como medio portador de la información permitieron una alta capacidad de inserción y una alta imperceptibilidad en la modificación de las imágenes estereoscópicas; con la ventaja de ser un sistema reversible, lo que implica que la imagen estereoscópica recuperada después de extraer la información queda idéntica a la forma original. En la implementación del sistema no hubo pérdidas en la información recuperada ni en calidad de las imágenes estereoscópicas portadoras.

En comparación con otros sistemas esteganográficos analizados que logran insertar y extraer información, la mayoría de estos presentan limitantes en la capacidad de inserción, o solo pueden aplicarse en imágenes a escala de grises; el sistema propuesto mejoró las limitantes antes mencionadas debido al empleo de imágenes estereoscópicas como medio portador en conjunto con el método reversible de modificación del histograma.

AGRADECIMIENTOS

Agradecemos al Instituto Politécnico Nacional, a la COFAA del IPN y al Programa Institucional de Formación de Investigadores del IPN por el apoyo otorgado para el desarrollo de este trabajo.

REFERENCIAS

- [1] T. Chiung Chen, C. Yu Hsuan y C. Yung Chen, «A Histogram Shifting-based Reversible Data hiding Scheme Using Multi-pattern Strategy,» de *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2015.
- [2] Z. J. Guzmán Zavaleta, «Algoritmo Reversible para Ocultamiento de Datos y su Arquitectura Hardware,» Tesis Maestría, Puebla, 2008.
- [3] J. Neil F. y J. Sushil, «Exploring Steganography: Seeting the usen,» IEEE, pp. 26-34, 1998.
- [4] C. Baviera, «¿Qué es la visión estereoscópica?,» [En línea]. Available: <https://www.clinicabaviera.com/blog/bye-bye-gafasconoce-tus-ojosque-es-la-vision-estereoscopical/>. [Último acceso: septiembre 2019].
- [5] Fortune Business Insights, «Information and Technology,» septiembre 2019. [En línea]. Available: <https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-market-101378>. [Último acceso: marzo 2020].
- [6] C. Cruz Ramos, R. Reyes Reyes, V. Ponomaryo y L. R. Noriega Galeana, «LSB-Steganography Framework for Stereoscopic Images based on BJND,» de 14th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE), México, 2017.
- [7] J. Gupta, P. Gupta y S. Gupta, «Reversible Data Hiding Technique Using Histogram Shifting,» de 2nd International Conference on Computing for Sustainable Global Development (INDIACom), India, 2015.
- [8] D. M. Thodi y J. J. Rodríguez, «Expansion Embedding Techniques for Reversible Watermarking,» IEEE TRANSACTIONS ON IMAGE PROCESSING, vol. 16, nº 3, pp. 721-730, Marzo 2007.
- [9] C. Pak y L. Huang, «new color image encryption using combination of the 1D chaotic,» Elsevier, nº 138, pp. 129-137, 2017
- [10] «Middlebury Stereo Vision,» [En línea]. Available: <http://middlebury.edu/>. [Último acceso: Septiembre 2019].

Esteganografía en Imágenes a Color mediante DWT y Método de Tinkerbell

Edgar Cruz Pérez, Rogelio Reyes Reyes, Clara Cruz Ramos, Volodymyr Ponomaryov
 Instituto Politécnico Nacional, ESIME Unidad Culhuacán
 Av. Santa Ana No. 1000, Col. San Francisco Culhuacán, CP. 04430, Ciudad de México.
 Tel. (55) 5729-6000 Ext. 73264 Fax (55) 56562058
 email: Edg_const@hotmail.com, ccruzra@ipn.mx, reyesre@ipn.mx, vponomar@ipn.mx

Resumen—La esteganografía es un conjunto de técnicas que permiten ocultar información (texto, imágenes, etc.) dentro de un elemento portador de manera que no se pueda percibir que existe información oculta en el elemento portador. El principal requisito de la esteganografía es que el mensaje secreto dentro del objeto alterado sea indetectable. En el caso de esteganografía en imágenes digitales existen diversos métodos que trabajan en el dominio espacial o frecuencial. En el presente trabajo se propone un método para poder ocultar múltiples imágenes digitales a color en una sola imagen digital a color, utilizando la Transformación Discreta de Wavelet – Discrete Wavelet Transform (DWT) en diferentes niveles. La imagen portadora se divide en sus planos rojo (R), verde (G) y azul (B) y se le realiza una descomposición DWT a cada canal hasta un tercer nivel de la imagen portadora y de las imágenes a ocultar, posteriormente para el nivel $N=3$ se buscarán los valores más cercanos a cada una de las posiciones de las imágenes a ocultar y se sustituirán en un canal. El método propuesto demuestra alta imperceptibilidad obteniendo un PSNR promedio de 41dB y un SSIM de 0.99.

keywords—Esteganografía, imágenes digitales, RGB, DWT.

I. INTRODUCCIÓN

En la actualidad, con el constante avance de la tecnología y comunicaciones se ha generado la necesidad de una constante comunicación, compartiendo información a uno o más destinatarios (comúnmente imágenes, audio, video y documentos), asegurando que la información no se vea afectada o apoderada por terceros. Sin embargo, uno de los medios de comunicación más utilizados es el Internet, debido a su crecimiento y fácil acceso, la posibilidad que la información pueda ser interceptada se ha vuelto más tangible, debido a ello, es necesario emplear diferentes métodos para mantener la comunicación segura. Una de las técnicas empleadas comúnmente para el envío de información confidencial es la esteganografía [1], cuyo principal objetivo es ocultar la mayor cantidad de información del mensaje en un elemento portador sin que pueda ser percibido por personal no autorizado.

Existen diferentes métodos de esteganografía que involucran imágenes digitales [2,3]. Estos métodos utilizan una imagen digital como objeto portador para ocultar información, generando una nueva imagen, usualmente lleva por nombre imagen estego. Por lo general, los métodos de esteganografía en imágenes digitales se clasifican en dos secciones: esteganografía con incrustación de datos, y esteganografía sin

incrustación de datos.

La esteganografía de una imagen digital se puede establecer en dos categorías, esteganografía en el dominio espacial y esteganografía en el dominio frecuencial. Los métodos utilizados en el dominio frecuencial modifican directamente los píxeles de la imagen digital para poder ocultar los datos, por otro lado, los métodos en el dominio frecuencial ocultan información en componentes frecuenciales que son invisibles en la imagen.

Para el dominio espacial, los valores pueden ser utilizados rápida y fácilmente en el Bit Menos Significativo (LSB – Least Significant Bit), sin que la imagen sufra una modificación perceptible, estos métodos pueden ser por el método de sustitución LSB o basado en la coincidencia LSB [5]. Las modificaciones que se dan por el método de incrustación en el dominio espacial comúnmente no pueden ser detectadas fácilmente por el ojo humano, pero la información incrustada es sensible a los ataques comunes en imágenes. Para solventar el problema de modificación de información en las imágenes portadoras, se han propuesto métodos de esteganografía en el dominio de la frecuencia, utilizando la Transformada Discreta de Fourier (DFT) [6], Transformada Discreta de Wavelet (DWT) [7,8] y la Transformada Discreta de Coseno (DCT) [9].

Para poder ocultar información sin incrustar datos, no se requiere de una modificación en los objetos portadores, los cuales podrán resistir varias tecnologías de esteganálisis. Una de las técnicas más comunes para ocultar información secreta sin incrustar información, es mediante la búsqueda de imágenes adecuadas que ya contengan información. Para la información secreta, el problema reside en la modificación de la imagen donde se oculta la información sin incrustar, debido al cambio de escala, cambio de luminancia o adición de ruido, los datos distorsionan la información y puede no recuperarse completamente.

En el presente artículo, se propone un método que hace uso de los dos métodos, incrustando información y también realizando una búsqueda para encontrar coincidencias entre las imágenes. Se hace uso de la DWT, y en el tercer nivel se incrustará información de la imagen secreta, posteriormente, para las sub-bandas de niveles inferiores, se comparan los datos más cercanos y se guardan para generar una nueva imagen.

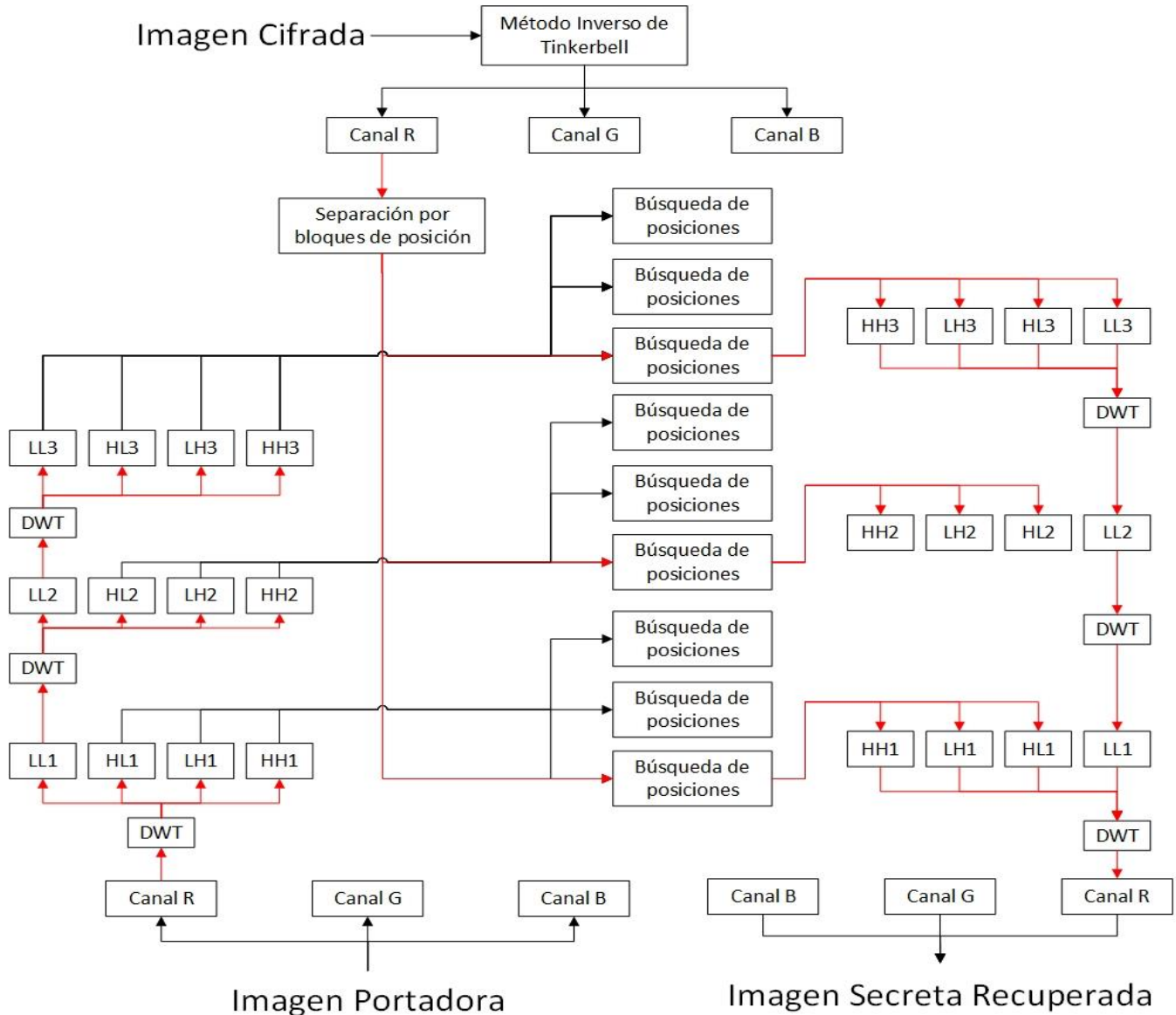


Figura 1. Diagrama a bloques de ocultamiento de información en el canal rojo del método propuesto.

II. MÉTODO PROPUESTO

El sistema propuesto cuenta con dos etapas principales: ocultamiento de información y extracción de información, ver la figura 1.

A. *Ocultamiento de la información.*

El diagrama a bloques para el ocultamiento de la información se muestra en la figura 1, donde se puede observar que se tiene una imagen portadora y una imagen a ocultar, cabe mencionar que las tres imágenes secretas pasarán por el mismo proceso de ocultamiento. Como se muestra en la figura 1, tanto para la imagen secreta como para la imagen portadora se descompondrán en sus planos RGB, y a cada uno de esos planos se les realizará la Transformación Discreta de Wavelet (DWT),

descomponiéndose en cuatro sub-bandas no superpuestas empleando filtros Haar. Estas sub-bandas son coeficientes de aproximación (LL), detalles horizontales (HL), detalles verticales (LH) y detalles diagonales (HH), la sub-banda LL se le vuelve a realizar la DWT y obteniendo nuevamente cuatro sub-bandas hasta alcanzar el valor final (tercer nivel).

Una vez alcanzado el nivel final en las sub-bandas LL, HL, LH y HH de los tres planos RGB de la imagen a ocultar, se realizará una búsqueda de los valores más cercanos con respecto a la imagen portadora en un solo canal y se sustituirán, se almacenarán las posiciones donde se realizó la sustitución. Cada sustitución de sub-banda de los tres canales de la imagen a ocultar se realizará en la misma sub-banda del canal de la imagen portadora, es decir, los valores de la sub-banda LL de los tres planos de la imagen a ocultar sustituirán valores en la sub-banda LL de un canal de la imagen portadora, del mismo

modo se realizará la sustitución en las demás sub-bandas del nivel final. Las tres imágenes sustituirán valores en las sub-bandas del nivel final en cada canal diferente de la imagen portadora. Para las sub-bandas HL, LH y HH de los niveles inferiores, se realizará una búsqueda de los elementos similares de las sub-bandas de las imágenes a ocultar en la imagen portadora en los mismos niveles, no se realizará sustitución, sino que se guardarán las posiciones.

Una vez encontradas todas las posiciones, se procede a realizar la Transformación Discreta de Wavelet Inversa (IDWT) para generar una imagen nueva, que contiene la información de las cuatro sub-bandas de las imágenes a ocultar en el nivel final. Las posiciones obtenidas, se utilizarán para generar otra imagen digital, quedando finalmente tres imágenes digitales con las posiciones de las imágenes ocultas.

A las imágenes llave, se le realizará un cifrado con el método Tinkerbell, de esta manera no se podrá obtener la información para generar la imagen secreta de la imagen portadora.

B. Extracción de información.

Para la extracción de la información, se toma en cuenta que el receptor debe de tener cuatro imágenes, una de ellas sería la imagen portadora modificada, y las otras tres imágenes llave permitirán recuperar los valores de las imágenes ocultas y poder reconstruirlas. Se debe realizar el proceso inverso del método Tinkerbell para recuperar la imagen de las posiciones de la imagen secreta. Se realizará la descomposición de la imagen portadora con la DWT en el mismo nivel al que se sustituyeron los valores, con la imagen llave se recuperarán los valores de las sub-bandas de la imagen a ocultar en los planos RGB. Una vez obtenido los valores de la sub-banda de cada nivel, se realizará la IDWT para recuperar la sub-banda LL del nivel inferior, y se realiza el mismo proceso IDWT hasta recuperar la imagen a ocultar. Este proceso se realiza para cada canal y se recuperarán las tres imágenes digitales.

C. Método de Tinkerbell

En el método propuesto, se hace uso del método de Tinkerbell, el cual es una función caótica la cual se obtiene a partir de (1) y (2):

$$x_{n+1} = x_n^2 - y_n^2 + ax_n + by_n \quad (1)$$

$$y_{n+1} = 2x_n y_n + cx_n + dy_n \quad (2)$$

Los valores generalmente utilizados para a, b, c, d son:

$$a = 0.9, b = -0.6013, c = 2.0, d = 0.50$$

$$a = 0.3, b = 0.6000, c = 2.0, d = 0.27$$

Para las ecuaciones (1) y (2), tanto x como y representan dos secuencias de Tinkerbell, y a, b, c, d son constantes, para este método se utilizaron los valores de $a = 0.3, b = 0.6000, c = 2.0, d = 0.27$. En la figura 2 se muestra el comportamiento del método de Tinkerbell con los valores previamente mencionados.

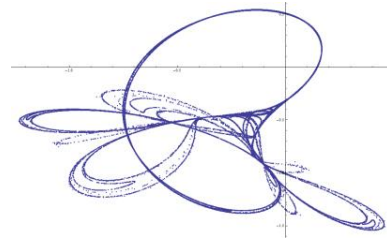


Figura 2. Atractor de Tinkerbell

III. RESULTADOS EXPERIMENTALES

Para evaluar el sistema propuesto, se utilizaron las imágenes de prueba de 512x512 píxeles, las imágenes utilizadas para las pruebas se obtuvieron de la base de datos Signal and Image Processing Institute [10]. Las imágenes seleccionadas para realizar el procedimiento fueron Airplane, Lena, Baboon y Peppers (mostrados en la figura 3(a)-(d) respectivamente).



Figura 3. Imágenes originales obtenidas de la base de datos Signal and Image Processing Institute.

A continuación, se muestran los resultados obtenidos con imágenes escaladas a 256x256 píxeles. En la figura 4(a) se puede observar la imagen portadora, a la cual se le realizó el proceso de sustitución en el tercer nivel DWT y en las figuras 4(b)-(d) se muestran las imágenes a ocultar. En la figura 5(a) se muestran los resultados obtenidos de la imagen portadora modificada, y en las figuras 5(b)-(d) se muestran los resultados de las imágenes secretas recuperadas.

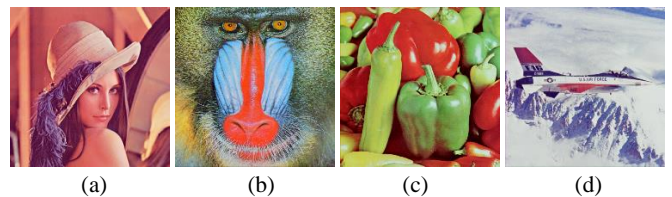


Figura 4. (a) Imagen portadora original, (b)-(d) imágenes a ocultar.

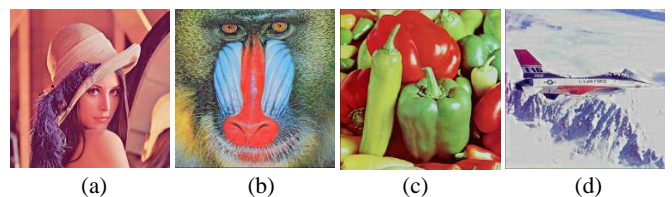


Figura 5. (a) Imagen portadora modificada, (b)-(d) imágenes recuperadas.

Con las posiciones almacenadas de los diferentes niveles de la DWT, se crea una nueva imagen, la cual estará compuesta por las posiciones X y Y de los píxeles sustituidos y similares de las imágenes secretas con respecto a la imagen portadora. En

la figura 6(a) se muestra la imagen creada a partir de las posiciones de la imagen secreta Baboon (figura 4(b)), mostrando similitudes con la imagen secreta, y la figura 6(b) se muestra la imagen de la figura 6(a) después de aplicarse el método de Tinkerbelle, la cual ya no muestra similitud alguna con la imagen secreta. Dichas imágenes creadas son el doble de ancho y mismo alto que la imagen secreta.

En la figura 7 se muestra el proceso por el cual pasa la imagen secreta, la figura 7(a) es la imagen secreta original, la figura 7(b) es la imagen creada con las posiciones y la figura 7(c) es la imagen secreta recuperada.

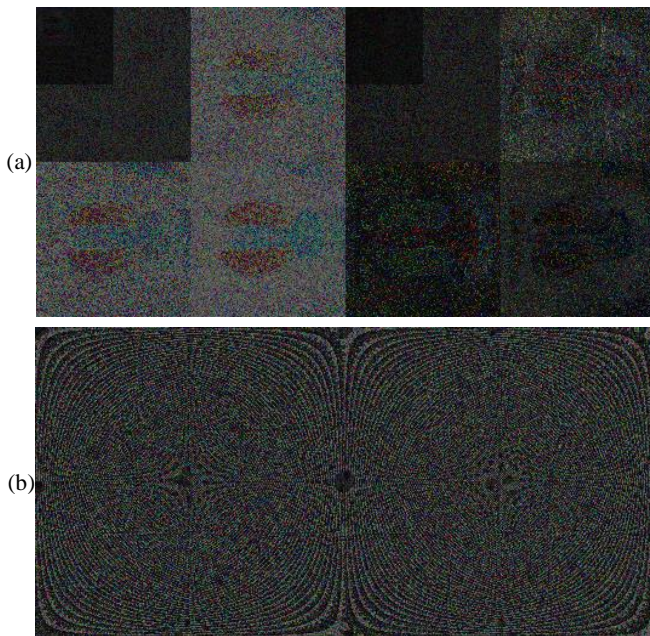


Figura 6. (a) Imagen resultante después del proceso de ocultamiento de la imagen secreta Baboon, y (b) imagen resultante después de pasar por el método de Tinkerbelle.

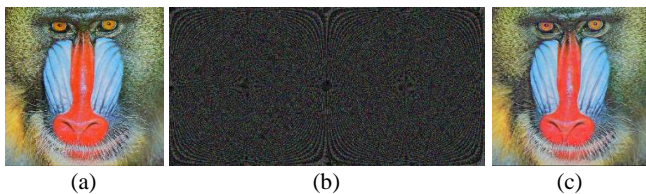


Figura 7. (a) Imagen secreta original, (b) imagen resultante después del proceso de ocultamiento de imágenes, y (c) imagen secreta obtenida del proceso de recuperación.

A. Calidad de las imágenes marcadas.

Se hará uso de los parámetros PSNR y SSIM para medir la calidad de las imágenes portadoras modificadas. Como se muestra en la figura 1, se realiza el proceso de búsqueda y sustitución de información (en el tercer nivel de la DWT) y búsqueda (en los niveles inferiores de la DWT) en un solo canal de la imagen portadora, cabe mencionar que este proceso también se realiza en los otros dos canales, de esta manera se pueden ocultar tres imágenes. En la figura 8 se muestran los resultados obtenidos de la imagen portadora original y la imagen portadora modificada.



Figura 8. (a) Imagen portadora original, (b) imagen portadora modificada.

En la tabla 1 se muestran los valores de PSNR y SSIM de la imagen portadora original con respecto a la imagen modificada. Se puede observar que mientras más grande sea la imagen portadora, incrementarán los valores de PSNR y SSIM, sin embargo, también incrementa el tiempo de procesamiento para encontrar las posiciones en la imagen portadora.

TABLA 1. VALORES PSNR Y SSIM DE LA IMAGEN PORTADORA ORIGINAL EN COMPARACIÓN CON LA IMAGEN PORTADORA MODIFICADA.

Imagen portadora	Tamaño	PSNR (dB)	SSIM
Lena	128x128	39.34	0.9960
	256x256	46.73	0.9995
	512x512	54.89	0.9999
Baboon	128x128	38.09	0.9879
	256x256	45.86	0.9957
	512x512	56.93	0.9999
Peppers	128x128	40.07	0.9957
	256x256	48.18	0.9991
	512x512	55.19	1.0000
Airplane	128x128	38.98	0.9953
	256x256	46.53	0.9989
	512x512	59.47	1.0000

B. Calidad de imágenes recuperadas.

Las imágenes utilizadas para ocultar se re-escalaron a diferentes tamaños (al igual que las imágenes portadoras). Cabe mencionar que las imágenes secretas no necesariamente deben ser del mismo tamaño que la imagen portadora, puede ser de diferentes tamaños. En las tablas 2 se muestran los valores PSNR y SSIM obtenidos para las imágenes secretas recuperadas.

TABLA 2. VALORES PSNR Y SSIM DE LAS IMÁGENES SECRETAS RECUPERADAS.

Imagen	Tamaño	PSNR (dB)	SSIM
Secreta 1	128x128	29.81	0.9812
	256x256	30.82	0.9866
	512x512	35.25	0.9851
	1024x1024	43.35	0.9983
Secreta 2	128x128	38.50	0.9978
	256x256	41.02	0.9979
	512x512	37.46	0.9853

	1024x1024	43.16	0.9991
Secreta 3	128x128	24.26	0.9424
	256x256	30.65	0.9493
	512x512	40.91	1.0000
	1024x1024	36.62	0.9985

C. Robustez de ocultamiento.

Para medir la robustez del sistema propuesto, las imágenes portadoras modificadas fueron atacadas por dos tipos de ruido, Gaussiano y Salt & Pepper. Como se mencionó anteriormente, la información recuperada es sensible a diferentes tipos de ataques, lo cual se muestra en las tablas 3 y 4, donde en la imagen portadora es alterada con ruido Gaussiano con densidades de $\sigma = 0.1$ y $\sigma = 0.01$ respectivamente. En las tablas 5 y 6 se muestran los resultados obtenidos para las imágenes portadoras contaminadas con el ruido Sal y Pimienta con densidades de 0.1 y 0.01 respectivamente.

TABLA 3. IMÁGENES RECUPERADAS CON PORTADORA MODIFICADA CON RUIDO GAUSSIANO CON DENSIDAD $\sigma = 0.1$.

Imágenes	PSNR (dB)	SSIM
Secreta 1	17.78	0.8245
Secreta 2	17.89	0.8097
Secreta 3	17.02	0.7977

TABLA 4. IMÁGENES RECUPERADAS CON PORTADORA MODIFICADA CON RUIDO GAUSSIANO CON DENSIDAD $\sigma = 0.01$.

Imágenes	PSNR (dB)	SSIM
Secreta 1	20.42	0.8478
Secreta 2	20.68	0.8357
Secreta 3	19.57	0.8213

TABLA 5. IMÁGENES RECUPERADAS CON PORTADORA MODIFICADA CON RUIDO SALT & PEPPER CON DENSIDAD=0.1.

Imágenes	PSNR (dB)	SSIM
Secreta 1	16.05	0.6465
Secreta 2	15.98	0.5904
Secreta 3	15.63	0.6378

TABLA 6. IMÁGENES RECUPERADAS CON PORTADORA MODIFICADA CON RUIDO SALT & PEPPER CON DENSIDAD=0.01.

Imágenes	PSNR (dB)	SSIM
Secreta 1	25.20	0.9196
Secreta 2	26.11	0.9168
Secreta 3	26.31	0.8930

Tanto el ruido gaussiano como el ruido Salt & Pepper afecta la calidad de las imágenes secretas recuperadas, en las figuras 9(a), 9(d), 9(g), 10(a), 10(d) y 10(g) se muestran las imágenes secretas recuperadas sin que la portadora haya sido atacada por ningún tipo de ruido. Las figuras 9(b), 9(e) y 9(h) son las imágenes secretas recuperadas de imagen portadora contaminada con ruido Gaussiano con densidad de $\sigma = 0.1$, donde se muestra que son más ruidosas en comparación con las figuras 9(c), 9(f) e 9(i), que son las imágenes secretas

recuperadas de imagen portadora con ruido Gaussiano con densidad de $\sigma = 0.01$. Para las figuras 10(b), 10(e) y 10(h) son las imágenes secretas recuperadas de imagen portadora contaminada con ruido Sal y Pimienta con densidad = 0.1, las cuales muestran un mayor grado de distorsión y alteración en comparación con las figuras 10(c), 10(f) e 10(i), que son las imágenes secretas recuperadas de imagen portadora con ruido Sal y Pimienta con densidad de $\sigma = 0.01$.

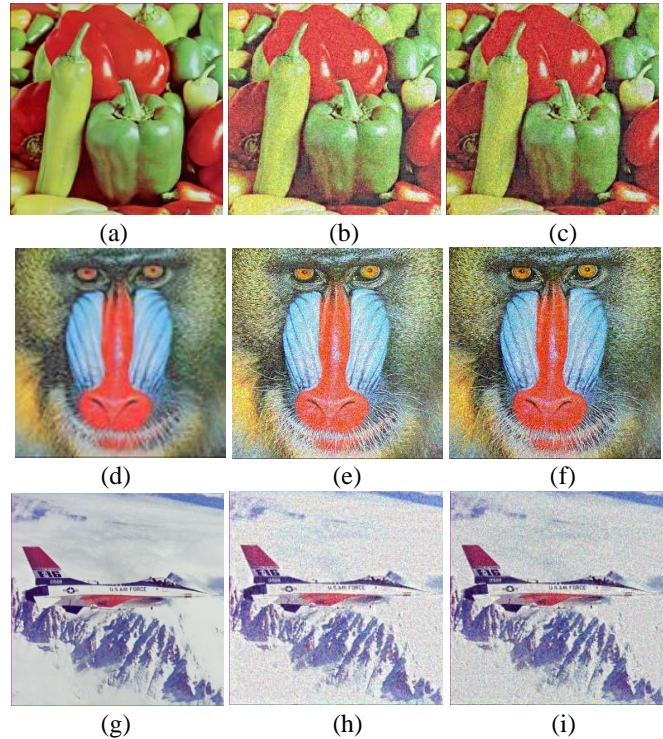
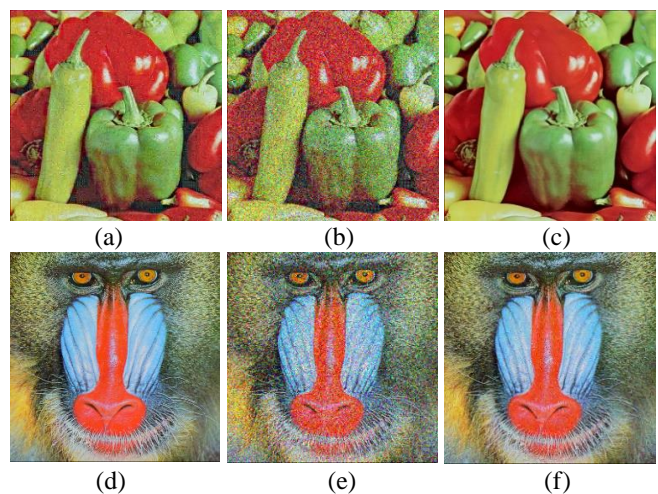


Figura 9. (a), (d) y (g) Imágenes secretas recuperadas sin ruido adicionado, (b), (e) y (h) imágenes secretas recuperadas de portadora contaminada con ruido Gaussiano con densidad $\sigma = 0.1$, (c), (f) e (i) imágenes secretas recuperadas de portadora contaminada con ruido Gaussiano con densidad $\sigma = 0.01$.



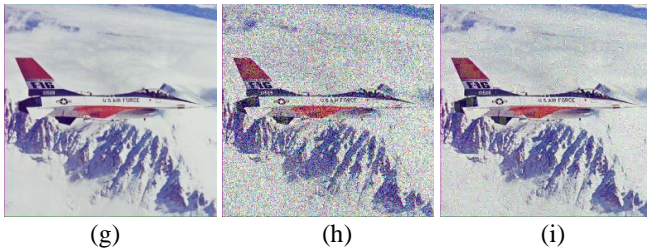


Figura 10. (a) (d) y (g) Imágenes secretas recuperadas sin ruido adicionado, (b), (e) y (h) imágenes secretas recuperadas de portadora contaminada con ruido Sal y Pimienta con densidad=0.1, (c), (f) e (i) imágenes secretas recuperadas de portadora contaminada con ruido Sal y Pimienta con densidad=0.01.

IV. CONCLUSIONES

En el presente trabajo se propone un método de esteganografía en imágenes digitales, el cual sustituye datos en el tercer nivel de la DWT y busca similitudes en los niveles inferiores. Se pudo observar que las imágenes portadoras modificadas tienen una alta calidad en PSNR y SSIM (47.49 dB y 0.9973 respectivamente) para diferentes resoluciones de imagen (para 128x128, 256x256 y 512x512), sin embargo, mientras más grande sea la imagen portadora más tiempo requerirá el sistema propuesto para poder encontrar las similitudes entre las imágenes. El método propuesto demostró ser lo suficientemente robusto para poder recuperar las imágenes secretas aún en presencia de ruido Gaussiano o multiplicativo con densidades altas ($\sigma = 0.1$ y $\sigma = 0.01$ para el ruido Gaussiano y, *Densidad* = 0.1 y *Densidad* = 0.01 para el ruido Sal y Pimienta), a pesar de tener una calidad relativamente baja con respecto a los parámetros PSNR y SSIM (17.65 dB y 0.8106 respectivamente con ruido Gaussiano con densidad $\sigma = 0.1$ y, 15.88 dB y 0.6249 respectivamente con ruido Sal y Pimienta con *densidad* = 0.1) las imágenes secretas recuperadas visualmente son reconocibles. Cabe mencionar que el método propuesto permite insertar imágenes secretas de mayor resolución que la imagen portadora debido a que los datos se insertan en diferentes niveles de la DWT, con la desventaja de un mayor tiempo de procesamiento.

AGRADECIMIENTOS

Agradezco al Instituto Politécnico Nacional, al Consejo Nacional de Ciencia y Tecnología de México, a la Comisión de Operación y Fomento de Actividades Académicas (COFAA) y al Programa Institucional de Formación de Investigadores del IPN por el apoyo otorgado para el desarrollo de este trabajo.

REFERENCIAS

- [1] A. Singh, M. Chauhan and S. Shukla, "Comparison of LSB and Proposed ModifiedDWT Algorithm for Image Steganography," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida (UP), India, 2018, pp. 889-893.
- [2] A. G. Ranade, S. Dargad and K. Mistry, "RGB Model Based Image Enhancement Technique for Steganography," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018, pp. 1-4.
- [3] V. K. Shalini and T. Abirami, "High secure digital image steganography for secret communication," 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM), Chennai, 2017, pp. 920-922.
- [4] S. Kamila, R. Roy and S. Changder, "A DWT based steganography scheme with image block partitioning," 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, 2015, pp. 471-476.
- [5] S. K. Yadav and M. Dixit, "An improved image steganography based on 2-DWT-FFT-SVD on YCBCR color space," 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, 2017, pp. 567-572.
- [6] McKeon, R.T.: Strange Fourier steganography in movies. In: Proceedings of the IEEE International Conference on Electro/information Technology (EIT), pp. 178-182 (2007).
- [7] A. K. Al-Asmari, A. Salama, A. M. Iliyasu and M. A. Al-Qodah, "A DWT Ordering Scheme for Hiding Data in Images Using Pixel Value Difference," 2012 Eighth International Conference on Computational Intelligence and Security, Guangzhou, 2012, pp. 553-557.
- [8] Y. Han, Y. Shang and W. He, "DWT-Domain Dual Watermarking Algorithm of Color Image Based on Visual Cryptography," 2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Beijing, 2013, pp. 373-378.
- [9] Y. M. Behbahani, P. Ghayour and A. H. Farzaneh, "Eigenvalue Steganography based on eigen characteristics of quantized DCT matrices," ICIMU 2011: Proceedings of the 5th international Conference on Information Technology & Multimedia, Kuala Lumpur, 2011, pp. 1-4.
- [10] <http://sipi.usc.edu/database/>

Generación de modelos digitales de elevación mediante Interferometría de Radar de Apertura Sintética

Zenia Pérez Falls

Departamento de Oceanología
Centro Interdisciplinario de Ciencias Marinas. CICIMAR
La Paz., BCS, México
zperezfalls@gmail.com

Guillermo Martínez Flores

Departamento de Oceanología
Centro Interdisciplinario de Ciencias Marinas. CICIMAR
La Paz, BCS, México
gmflores@ieec.org

Resumen—La generación de modelos digitales de elevación (MDE) a partir de imágenes de radar contribuye al conocimiento de la estructura del terreno, con aplicaciones en hidrología superficial, planificación ambiental, modelación de escorrentías, entre otras. La técnica de Interferometría de Radar de Apertura Sintética (InSAR) se basa en la combinación de dos imágenes de Radar de Apertura Sintética (SAR) de la misma escena adquirida desde puntos ligeramente diferentes. Para aplicar esta técnica se utilizaron imágenes del satélite Sentinel-1 del 3 y 27 de marzo de 2017 obtenidas de la Agencia Espacial Europea. El tratamiento de las imágenes de radar se realizó con el software SNAP y el programa SNAPHU. La comparación entre el MDE generado y otros dos MDE (INEGI, SRTM) se realizó en R. Como resultado se obtuvo un MDE de 15 m de resolución espacial con dimensiones de 6609 x 7337 píxeles, un rango de elevación de 2, 100 m, mínimo de -128 m, media de 432. Este modelo se comparó con los MDE de INEGI (15 m) y SRTM (30 m), obteniendo un coeficiente de determinación de 0.96 y error residual máximo de 400 m.

Palabras claves — SAR; InSAR; software; SNAP; SIG, Sentinel-1

I. INTRODUCCIÓN

Los modelos digitales de elevación (MDE) son estructuras numéricas de datos que representa la distribución espacial de la altitud de la superficie del terreno [1]. Estos modelos son componentes elementales de la cartografía y se utilizan en diversos campos, como gestión de recursos naturales, proyectos de ingeniería e infraestructura, análisis de desastres y riesgos, cartografía topográfica, análisis de deslizamientos e inundaciones, entre otras. Los MDE se pueden generar a partir de múltiples técnicas como fotogrametría, sistemas de posicionamiento global (GPS), altimetría de láser y técnicas de interferometría de radar (InSAR) [2]. En las últimas décadas, diversos autores han trabajado en la generación de MDE con la técnica InSAR provenientes de sistemas aereotransportados y satélites (Radar de Apertura Sintética (SAR)), debido a que esta técnica es una de las más eficientes en la construcción de MDE a diferentes resoluciones espaciales [3, 4].

SAR es un sistema de radar que se utiliza para la teledetección. Usa la amplitud y las diferencias de fases entre

la señal de retorno y la señal creada [3]. Los sistemas SAR expresan la distribución espacial de la amplitud y la fase de la señal retornada por el terreno y los objetos presentes en la escena en el área barrida por el satélite. La amplitud está relacionada con las propiedades dieléctricas (reflectividad) del terreno y la fase de la señal con la distancia entre el sensor (satélite) y el suelo para cada píxel [4].

Los sistemas de radar emiten energía electromagnética desde una antena, esta se propaga por el espacio hasta ser interceptada por el objetivo que recibe y redirige en distintas direcciones, parte de la energía reflejada o redirigida (eco) es detectada por la antena del radar que procesa la señal para adquirir información del objeto [5]. Estos sistemas activos operan en frecuencias de microondas (1 mm a 100 cm). Pueden operar en el día o la noche, minimizar las condiciones atmosféricas (nubosidad), detectar la rugosidad de la superficie, así como el contenido de humedad del suelo [6].

La tecnología de imagen SAR es reconocida como una herramienta muy prometedora para la captura de información cartográfica porque utiliza sensores activos y las nubes son básicamente transparentes en esta parte del espectro electromagnético (1 mm a 100 cm).

Actualmente, InSAR es una de las aplicaciones de las imágenes de radar que presenta más perspectivas en cuanto a su aplicación para la extracción de topografía [7, 8]. Posee la capacidad de medir patrones de interferencia causados por la diferencia de fase entre dos o más imágenes, posibilitando la determinación de diferencias angulares y distancias relativas con gran precisión [9].

La interferometría detecta la radiación reflejada de cada pase en la misma órbita. Es una técnica geodésica basada en la combinación de dos imágenes SAR de la misma escena adquiridas desde puntos ligeramente diferentes. Actualmente, esta técnica ha tomado gran importancia como una herramienta poderosa para realizar el mapeo de la superficie terrestre relacionada con su alta resolución espacial, precisión y la capacidad de generar MDE [9, 3].

Las imágenes SAR del Satélite Sentinel-1 de la Agencia Espacial Europea (ESA), son de libre acceso y están disponibles desde su fecha de operación en 2014 hasta la actualidad. La misión Sentinel-1 opera en cuatro modos de imágenes exclusivos: con resoluciones espaciales y coberturas distintas: *Extra Wide swath* (EW) con resolución espacial de 25 x 100 m y franja de cobertura de 400 km; *Strip Mode* (SM) con resolución espacial de 5 x 5 m y franja de cobertura de 80 km; *Wave Mode* (WV) con resolución espacial de 5 x 20 m y franja de cobertura de 20 km; y finalmente *Interferometric Wide swath* (IW) con resolución espacial de 5 x 20 m y cobertura de 250 km [10]. Este último modo es el utilizado para realizar interferometría.

La técnica InSAR mediante Sentinel-1 es aplicable para la generación de MDE en cualquier región donde haya imágenes disponibles. El objetivo de este trabajo es evaluar MDE obtenidos a partir de imágenes SAR del satélite Sentinel-1 mediante la técnica InSAR y compararlo con MDE de referencia SRTM (*Shuttle Radar Topography Mission*) y CEM (Continuo de Elevaciones Mexicanos 3.0) obtenido de INEGI (Instituto Nacional de Estadística y Geografía).

II. DATOS Y MÉTODOS

A. Datos

Para implementar la técnica InSAR se obtuvieron las imágenes del servidor *Alaska Satellite Facility* del sur de la península de Baja California, correspondientes a las fechas del 3 y 27 de marzo de 2017 (*path 56, frame 516*) y órbita descendente (Fig.1). Se seleccionó un par interferométrico generado a partir de imágenes SAR del satélite Sentinel-1, de banda C (5,4 Hz) en el modo de adquisición *Interferometric Wide swath* (IW), con un nivel de procesamiento *Single Look Complex* (SLC), 250 km de cobertura y una resolución espacial de 5 x 20 m.

Para la generación de MDE se deben de tener en cuenta una serie de criterios para reducir la descorrelación temporal que surge entre la adquisición de las imágenes.

Las imágenes se seleccionaron según los siguientes criterios: 1) que el rango de la distancia entre la ubicación del sensor en las dos tomas de las imágenes (línea base perpendicular) sea de 130 a 300 m [11]. Este primer criterio es debido a que los interferogramas generados con líneas base perpendiculares muy pequeñas (< 30 m) son casi inútiles debido a su alta sensibilidad al ruido de fase y a los efectos atmosféricos; así como los que presentan valores superiores 450 m, generalmente son casi imposibles de desenvolver la fase. Por lo tanto, la línea base perpendicular óptima se encuentra entre 150 y 300 m. 2) que el tiempo entre las dos pasadas de las imágenes (línea base temporal) sea entre 6 y 12 días [10], debido a que mientras mayor sea el intervalo de tiempo entre las dos imágenes, se va a obtener más pérdida de coherencia. Además, de ser más afectados por factores como el clima, la época del año, los cambios en la vegetación, así como las distorsiones atmosféricas.

B. Métodos

Para la generación de MDE se utilizó el software SNAP (*Sentinel Application Platform*) desarrollado por la ESA. Además, el programa SNAPHU [12] que describe un algoritmo de flujo de red estadístico para el desenvolvimiento de fase, y permite procesar la fase interferométrica a partir de unos datos de entrada (interferograma). Este busca estimar de manera congruente los valores que se ajusten estadísticamente como máximos probables [12]. Se utilizó también, el Sistema de Información Geográfica (QGIS) para la salida cartográfica del modelo para la comparación de los MDE (Fig. 2).

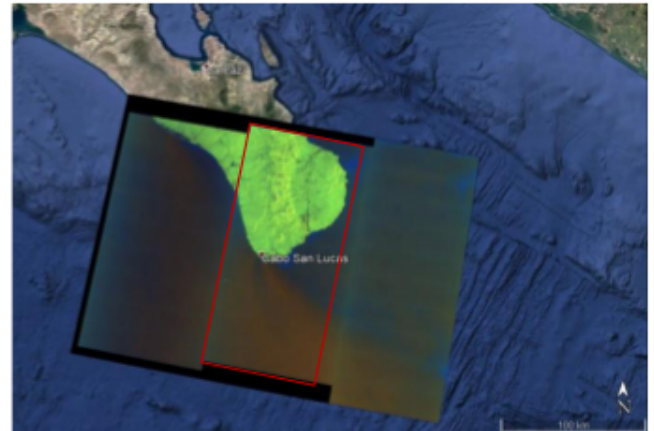


Fig. 1. Ubicación de las imágenes Sentinel-1 A

En la Fig. 2 se muestra la secuencia de procesamiento para generar modelos digitales de elevación con la técnica de InSAR.

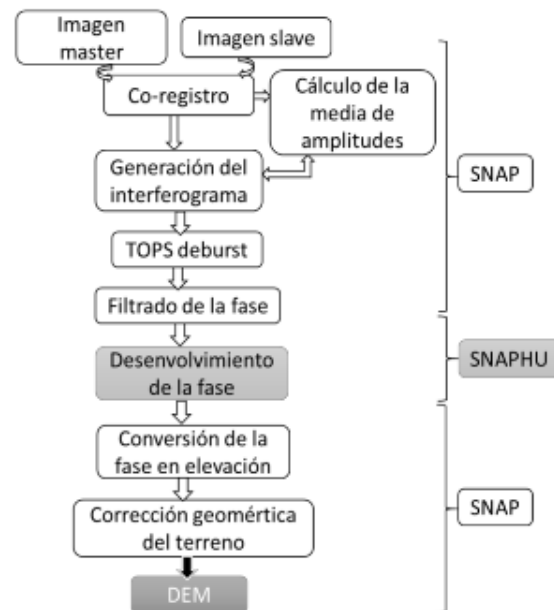


Figura 2. Esquema metodológico para procesamiento de imágenes de radar mediante la técnica InSAR utilizando SNAP y SNAPHU

A continuación, se describen las etapas del procesamiento representado en la Fig. 2 para implementar InSAR.

Co-registro

Consiste en la superposición de las dos imágenes. Se determina una como referencia (*master*) y la segunda que es adquirida con fecha posterior (*slave*) cuya cobertura será un subconjunto de la imagen *master* [5].

Generación del interferograma

Suponiendo que las fases de reflectividad del terreno y de retraso atmosférico son las mismas en ambas imágenes, y que el ruido en ambas puede ser omitido, la fase interferométrica (ϕ) será calculada restando las fases de las dos imágenes SAR (1):

$$\phi = \phi_M - \phi_S = 4\pi (R_M - R_S) / \lambda = 4\pi \Delta R / \lambda \quad (1)$$

Donde ϕ_M y ϕ_S son las fases de las imágenes, R_M y R_S son las distancias entre el sensor y el punto-dispersor en el terreno y λ es la longitud de onda usada [5].

En esta etapa se utiliza un MDE de referencia como el SRTM para la interpolación (binomial) de datos entre el par de imágenes interferométricas [9].

Las imágenes SAR contienen tres bandas o *subswath* las cuales deben de ser procesadas por separados. En la Fig. 3 se observan las tres subfranjas de la imagen Sentinel-1A, resaltando la subfranja IW2 que fue la procesada.

Para la elaboración del interferograma se obtuvo la diferencia de fase entre las imágenes *master* y *slave* en cada píxel.

El interferograma calculado (Fig. 4) contiene la variación de fase de diversos factores contribuyentes (2), por lo que en la variación de la fase (ϕ) influyen la fase de la tierra plana (curvatura de la tierra) (ϕ_{flat}); la fase topográfica (ϕ_{DEM}) (superficie topográfica de la tierra); condiciones atmosféricas (ϕ_{atm}) como cambio de humedad, temperatura y presión entre las dos adquisiciones de imágenes; y otros ruidos (ϕ_{noise}) como cambios de los dispersores, diferentes ángulos de visión y volumen de dispersión [13].

$$\phi = \phi_{DEM} + \phi_{flat} + \phi_{atm} + \phi_{disp} + \phi_{noise} \quad (2)$$

Como la interferometría SAR intenta estimar la contribución de la superficie terrestre (ϕ_{flat} y ϕ_{DEM}) que se considera igual para ambas imágenes y derivar las alturas de la topografía, la fase topográfica dependerá de la fase interferométrica (ϕ) y la curvatura de la Tierra (ϕ_{flat}) (3):

$$\phi_{DEM} = \phi - \phi_{flat} \quad (3)$$

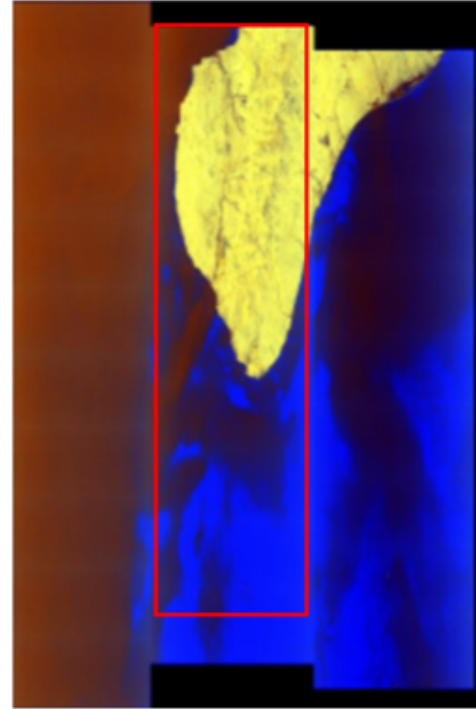


Fig. 3. Imagen Sentinel-1A del 3 de marzo de 2017

Como parte de este procedimiento se crea una banda de coherencia que permite verificar la viabilidad del proceso interferométrico y la calidad del mismo [10]. Esta banda de coherencia es el parámetro usado para evaluar la calidad de la fase y sus valores oscilan entre 0 y 1 donde valores cercanos a uno indican alta coherencia, es decir, confiabilidad en el interferograma (Fig. 4).

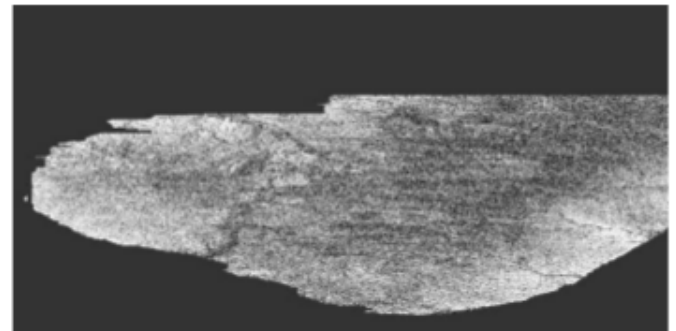


Fig. 4. Resultados de coherencia

TOPS deburst

Debido a que los productos del Sentinel-1 se crean a manera de franjas, que a su vez están compuestas por una serie de ráfagas, es necesario realizar el proceso *TOPS deburst* que consiste en eliminar estas ráfagas (Fig. 5) [9].

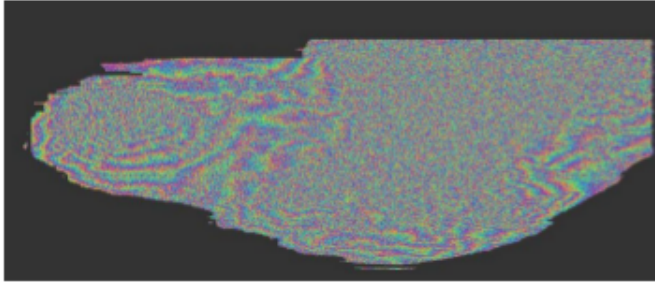


Fig. 5. Interferograma envuelto con TOPS deburst

Filtro Goldstein

Posteriormente, se realizó el filtro Goldstein que es una técnica de pre-procesamiento que reduce el ruido de la fase interferométrica, lo que facilita su desenvolvimiento en términos de precisión [8]. Al realizar este filtrado las franjas interferométricas se acentúan y se vuelven más nítidas (Fig. 6).

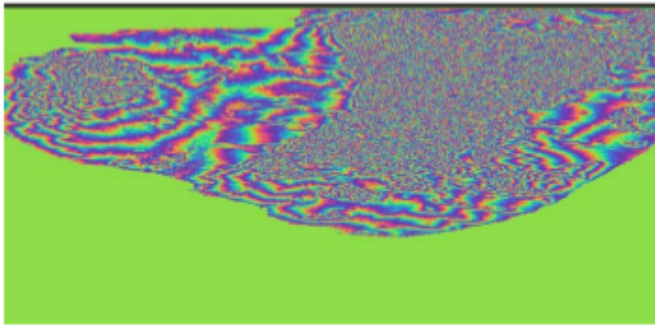


Fig. 6. Filtro Goldstein realizado al interferograma envuelto

Desenvolvimiento de fase

Partiendo del interferograma envuelto (Fig. 6) que contiene la información de la fase y la amplitud, se pasa al desenvolvimiento de fase mediante el algoritmo SNAPHU. El enfoque de la interferometría de radar en este paso es trabajar con la señal de fase relativa bidimensional, que es el módulo 2π de la señal de fase absoluta (es desconocida). En ese sentido, el problema directo es la fase envuelta dada en un intervalo de $(-\pi, \pi)$ y por otra parte el problema inverso, que es el desenvolvimiento de la fase siendo más complejo debido a su no linealidad y no singularidad [9].

Conversión de la fase a elevación

Para la generación del MDE se convierte la fase interferométrica en valores de altura, calculando la fase de un número de alturas discretas y comparándolas con la fase real del interferograma para determinar la altura. En este paso se utiliza como referencia un MDE externo (SRTM de 30 m).

Geocodificación de MDE

Se geocodifica el modelo obtenido corrigiendo las distorsiones geométricas SAR y se utiliza un MDE de referencia (SRTM de 30 m) y creando un producto proyectado en el mapa. La geocodificación convierte una imagen de rango inclinado o geometría de rango de terreno en un sistema de coordenadas de mapa. Implica, además, el uso del DEM para corregir las distorsiones geométricas inherentes, como el escorzo, la escala y la sombra producidas por la topografía [10]. También el hecho de la topografía de las escenas y la inclinación del sensor que hace que las distancias puedan ser distorsionadas o susceptibles a errores en las imágenes de radar.

Una vez generado el MDE se realizará una comparación con los MDE de referencia (SRTM y INEGI) para evaluar los valores de elevación obtenidos por el Sentinel-1.

III. RESULTADOS Y DISCUSIÓN

A. MDE generado por Sentinel-1

Se obtuvo un MDE con dimensiones de 6609 x 7337 píxeles, resolución espacial de 15 m, valor máximo de 2, 100 m, media de 432 y una mediana de 299.

Para la comparación entre MDE se seleccionó una región con valores de coherencia cercanos a 0.5. Se consideró este umbral debido a que es estimado como un buen indicador de la calidad de la fase interferométrica para la generación del MDE [14]. Esta región se muestra en el recuadro de la Fig. 7 (coordenadas 593975, 2562609 Este y 646100, 2603724 Norte).

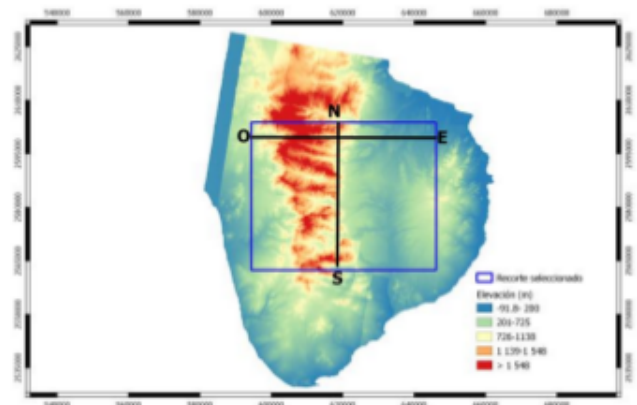


Fig. 7. MDE generado a partir de InSAR. Recorte seleccionado para comparación de los MDE. Perfiles trazados

La primera comparación se realizó con las estadísticas básicas correspondientes a los tres MDE. Se observa en la Tabla I que las mayores diferencias se aprecian en los valores mínimos de elevación, siendo el MDE del Sentinel-1 el único que presenta valores negativos, mientras que las variaciones en los máximos no sobrepasan los 20 metros entre los tres MDE.

TABLA I. Estadísticas básicas para los MDE

Estadísticas básicas	Sentinel-1	SRTM	INEGI
Número de píxeles	9524975	9524975	9524975
Mínimo	-128	43	0
Máximo	2090	2082	2079
Media	565	529	528
Mediana	385.8	352	350
Coefficiente de variación	1.37	1.44	1.45

Los histogramas (Fig. 8), muestran que la distribución en los tres MDE es similar en los valores máximos, sin embargo, el modelo generado es el que presenta mayor frecuencia de píxeles en las menores elevaciones y mayor distribución en los máximos de elevación.

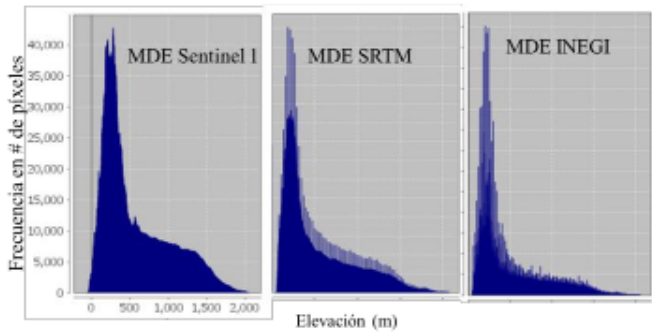


Fig. 8. Histogramas de frecuencias

Además, con la finalidad de comparar las diferencias en la elevación de los MDE se trazaron dos perfiles (Fig. 7). Los resultados de los perfiles muestran que el comportamiento de los MDE es muy similar en los dos con un coeficiente de correlación R^2 de 0.96 para ambas comparaciones. Sin embargo, siguen sobresaliendo los valores mínimos en el MDE generado.

Se ha demostrado que los métodos de interpolación que se utilizan para generar el MDE tiene una influencia muy importante sobre su calidad, además, de las distorsiones de la atmósfera son factores que podrían haber influido en los resultados negativos obtenidos.

Este presenta una irregularidad en la estimación de las alturas (ruido de fase) para puntos de elevaciones cercanos entre sí, mostrando una mayor variación que los modelos del SRTM y INEGI (Fig 9 y 10).

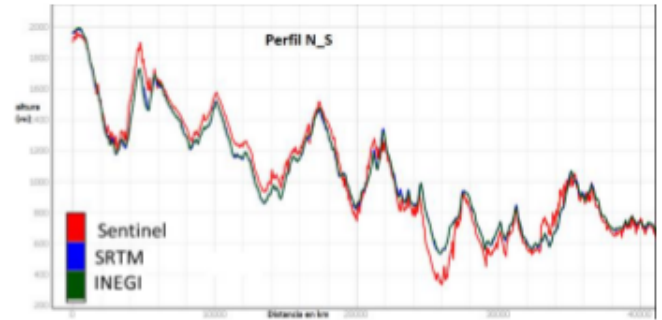


Fig. 9. Perfil Norte-Sur

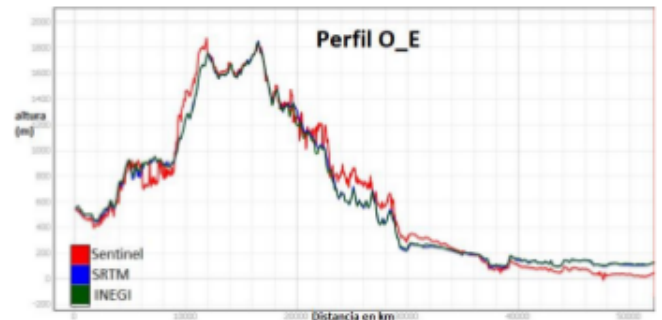


Fig. 10. Perfil Oeste-Este

B. Comprobación mediante puntos aleatorios

Se seleccionaron 100 puntos aleatorios en la región previamente delimitada para los MDE (Fig. 7) con la finalidad de determinar la correlación entre las elevaciones correspondientes (Fig. 11).

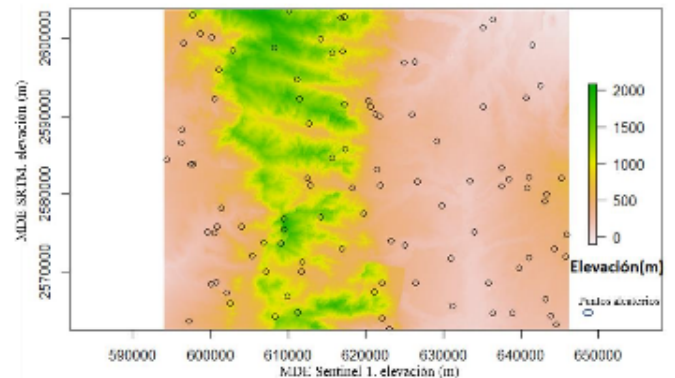


Fig. 11. Puntos aleatorios comunes tomados en los MDE comparados (Sentinel-1, SRTM, INEGI)

Con los diagramas de dispersión obtenidos de los 100 puntos aleatorios generados, se calculó la línea de tendencia y el coeficiente de determinación R^2 con un valor de 0.96 (Fig. 12) para las comparaciones entre MDE Sentinel-1 y SRTM; MDE Sentinel-1 y INEGI. Se muestra de esta forma que existe una fuerte relación entre los MDE del SRTM y INEGI con el del Sentinel-1.

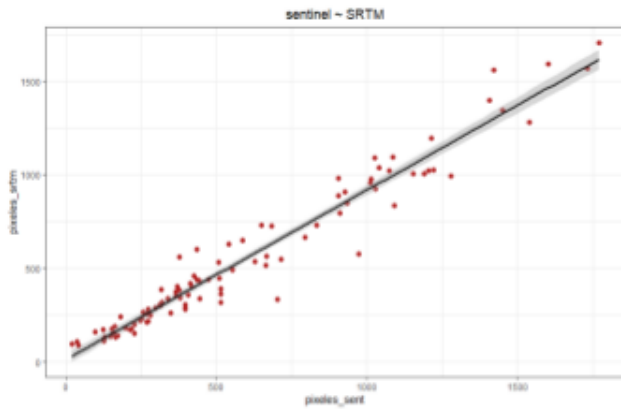


Fig. 12. Recta de regresión lineal de puntos aleatorios en MDE Sentinel-1, SRTM, INEGI

Adicionalmente, se obtuvieron los errores residuales de la línea de regresión lineal entre los puntos aleatorios de los MDE Sentinel-1 y SRTM; MDE Sentinel-1 y INEGI. El comportamiento en ambas comparaciones es similar con residuales máximos de 400 metros y mínimos de -200 m (ruido en la fase durante el procesamiento de datos) (Fig. 13).

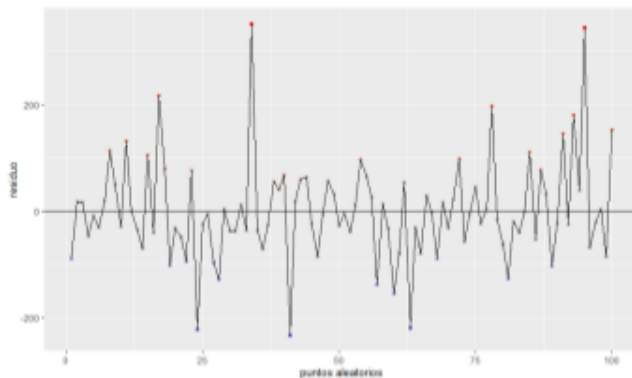


Fig. 13. Residuales de regresión lineal entre DEM Sentinel-1 con SRTM y INEGI

La comparación realizada en este trabajo del MDE generado mediante InSAR con los tomados de referencia (SRTM y INEGI) mostraron resultados similares entre los mismos. A pesar de observarse los exagerados valores negativos que mostró el MDE generado con InSAR en el resto de los parámetros tuvieron valores muy similares. Otro aspecto importante que se tuvo en cuenta fue el tamaño de la línea base perpendicular, las imágenes utilizadas en este trabajo presentaban una línea de 134 m. A pesar de no encontrarse entre los valores de base perpendicular óptima propuesto por [1] (entre 150 y 300 m), se obtuvieron resultados acertados.

Otros autores como [14, 15] también evalúan los modelos mediante puntos aleatorios seleccionados a partir de los MDE para validar el modelo creado, diferenciando el comportamiento de este en zonas urbanas y en llanas. En estos trabajos se evalúan los MDE generados mediante InSAR con el método *Root Mean Square* (RMS). En los resultados de

[14] obtuvieron valores máximos del error en las zonas montañosas de 321 m. Mientras que en [15] se seleccionaron 30 puntos aleatorios y el RMS también resultó mayor en áreas montañosas con 21.7 m.

IV. CONCLUSIONES

Los MDE son utilizados en numerosas áreas y son procesados mediante diversos software, por lo que la utilización de datos gratuitos como lo son las imágenes SAR del satélite Sentinel-1 permite la generación de MDE con alta resolución y de una amplia cobertura (250 km) [3].

Los puntos aleatorios determinados, sirvieron para evaluar los valores de elevación del MDE generado, esto mostró las similitudes entre los modelos, así como en los perfiles trazados sobre los mismos.

Para un futuro trabajo, se propone establecer puntos de control en el área de estudio para así tener puntos in situ y obtener una evaluación más exacta del modelo. Además, realizar otras combinaciones de imágenes para solucionar las partes donde se obtuvo menos coherencia. Las imágenes InSAR son una posibilidad confiable para obtener MDE con resoluciones medias a través del acceso de datos gratuitos y libre acceso.

AGRADECIMIENTOS

Los autores agradecen al Centro Interdisciplinario de Ciencias Marinas del Instituto Politécnico Nacional por el apoyo para la realización de este trabajo (a través del Proyecto SIP- 20195187 "Evaluación de subsidencia costera en los estados de Tabasco y Campeche mediante interferometría diferencial de radar de apertura sintética").

REFERENCIAS

- [1] A.M. Felicísimo, "Modelos Digitales del Terreno. Introducción y aplicaciones en las ciencias ambientales", 1994.
- [2] X.Wang, D. Holland, H. Gudmundsson, "Accurate coastal DEM generation by merging ASTER GDEM and ICESat/GLAS data over Mertz Glacier, Antarctica". *Remote Sensing of Environment*, 206, 2018, pp 218-230.
- [3] P.Vidal, W. Pérez, A. Fernández-Sarría, "Evaluación de Modelos Digitales de Elevación (MDEs) obtenidos a partir de imágenes Sentinel 1 en la región metropolitana de Chile", *Teledetección: hacia una visión global del cambio climático*, 2019, pp 373-376.
- [4] M. Crosetto, "Calibration and validation of SAR interferometry for DEM generation. *ISPRS Journal of Photogrammetry and Remote Sensing*", 57(3), 2002, pp 213-227.
- [5] O. Sarychikhina, E. Glowacha, F. Suárez, R. Mellors, R. Hernández, "Aplicación de DInSAR a los estudios de subsidencia en el Valle de Mexicali", *Boletín de la Sociedad Geológica Mexicana*. Vol 63, N° 1, 2011, pp 1-13.
- [6] A.K. Gabriel y R.M. Goldstein, "Crossed orbit interferometry: theory experimental results from SIR_B", *Int.J. Remote Sensing*, vol 9, 1998.
- [7] C.A. Guerrero y P.A. Hernández, "Determinación de un modelo digital de elevación a partir de imágenes de radar sentinel 1 usando interferometría SAR", 2017, pp 45-110.
- [8] C.Patry y F. Rocca, "Limits to the resolution of elevation maps from stereo images", *Int.J. Remote Sensing*, vol 11, 1990.
- [9] R. Golstein & C.L.Warner, "Filtrado de fase de interferograma radar para aplicaciones geofísicas". *Geophysical Research Letters* (Vol 25), 1998, pp 4035-4038.

- [10] European Space Agency, "Sentinel-1 Document Library", Obtenido de ESA Sentinel Online: <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/document-library>, 2017.
- [11] A.Ferretti, A. Monti-Guamieri, C. Prati, F. Rocca, "InSAR Principles: Guidelines for SAR interferometry processing and interpretation", Vol.19, Paises bajos: ESA Publications, 2007.
- [12] C.W.Chen, H.A. Zebker, H.A., "Network approaches to two-dimensional phase unwrapping: intractability and two new algorithms", Journal Optical Society of America, 17, 200, pp 401-414.
- [13] K. Nikoňakopoulos, "Preliminary results of using sentinel 1 SAR data for DSM generation", European Journal of Geography Vol 6, Number 3, 2015, pp 52 -68.
- [14] J. M Quiñonero y F. Alonso, "Creación de Modelos Digitales de Elevaciones a partir de diferentes métodos de interpolación para la determinación de redes de drenaje". I Jomada de SIG libre. Universidad de Girona. SIGTE, pp. 4, 2007.
- [15] H. B. Makineci, H. Karabörk, "Evaluation digital elevation model generated by synthetic aperture radar data", The International.

Monitoreo de sargazo en el Mar Caribe

Guillermo Martínez-Flores¹, Dulce C. López-Sillas², José M. Echevarría-Rubio¹, Rubén A. Morales-Pérez³

¹Instituto Politécnico Nacional, Centro Interdisciplinario de Ciencias Marinas, Departamento de Oceanología, La Paz, Baja California Sur, México, gmflores@ipn.mx, jechegarriar@disroot.org

²Secretaría de Comunicaciones y Transportes, Oficina de Servicios a la Marina Mercante, La Paz, Baja California Sur, México, caro_lopez02@hotmail.com

³Instituto Mexicano de Tecnología del Agua, Coordinación de Hidráulica, Jiutepec, Morelos, México, rrmorales@tlaloc.imta.mx

Resumen— Los eventos recurrentes de arribo de sargazo al caribe mexicano han causado serios problemas ambientales y económicos, motivando la necesidad de su detección a largo plazo y a diversas escalas espaciales. El objetivo de este trabajo (en proceso) es implementar un enfoque metodológico para realizar esta detección a partir de imágenes de diversa resolución espacio-temporal (MODIS, Landsat-8 y Sentinel-2) y software de código abierto. Se presenta la segmentación de regiones identificadas como sargazo, así como su cobertura en 11 cuadrantes de 10×10 grados de la malla sinusoidal MODIS (MOD09A1) correspondientes al Mar Caribe. El proceso se realiza con una cobertura temporal de enero de 2018 a septiembre de 2019, con un total de 880 imágenes que corresponden a escenas con la reflectancia para 7 bandas, en composiciones de 8 días (46 escenas anuales por celda). Se presentan los resultados de los algoritmos para detectar la presencia de sargazo y cuantificar su cobertura usando el índice de algas flotantes (en escenas MODIS). El monitoreo continuo de sargazo es esencial para elaborar sistemas de alerta temprana para mitigar su impacto en la zona costera del caribe mexicano.

Palabras clave—Sargassum; sensores remotos; procesamiento de imágenes; reflectancia; índice de algas flotantes.

I. INTRODUCCIÓN

El sargazo flotante se compone de algas del género *Sargassum fluitans* y *Sargassum natans*, macroalgas pardas holopelágicas. Son un componente importante de la flora marina de zonas tropicales y subtropicales y se propagan por la fragmentación vegetativa. Estas algas se alinean en filamentos en la superficie del océano por efecto de la circulación marina, brindando un hábitat de mar abierto que concentra a tortugas, aves, invertebrados y más de 100 especies de peces [1]. Por otra parte, la gran cantidad de sargazo acumulado en las playas y costas, causan un impacto adverso en los ecosistemas locales, provocando cambios en la composición física y química de las aguas, modificando procesos fisiológicos y ecológicos en los ecosistemas de arrecifes de coral cercanos a estas áreas. Los efectos negativos también repercuten en la economía de la sociedad, en actividades como la pesquería y el turismo, haciendo necesaria la inversión de fondos destinados a la eliminación de las grandes cantidades de esta alga en zonas turísticas [2, 3, 4].

A partir de 2014 muchas islas y países del Caribe comenzaron a ser afectados. En el caso del Caribe Mexicano, la afluencia masiva de sargazo pelágico a la costa inició a principios de verano de 2014 y terminó en diciembre de 2015. Este comportamiento sin precedentes hace necesario realizar un monitoreo de la distribución y cobertura de sargazo en esta región, con el objetivo de realizar un seguimiento periódico de su trayectoria mediante el procesamiento de imágenes satelitales.

El análisis espectral de la reflectancia superficial constituye el método esencial para el estudio de los componentes suspendidos o disueltos en aguas oceánicas. El uso de los datos provistos por el sensor MODIS (*Moderate Resolution Imaging Spectroradiometer*) a bordo de los satélites Aqua y Terra, por su cobertura global y libre acceso, es el punto de partida para desarrollar un método para la detección de sargazo [5, 6]. Sin embargo, la detección precisa de estas relativamente pequeñas superficies mediante el uso de sensores remotos, aun enfrenta retos debido a la resolución de las imágenes, cobertura de nubes y limitaciones de los algoritmos. Se han propuesto varios índices espectrales para superar este tipo de inconvenientes, entre los que destacan el EVI (*Enhanced Vegetation Index*), NDVI (*Normalized difference vegetation index*) y el FAI (*Floating Algae Index*).

Para realizar esta detección y monitoreo requiere que el algoritmo sea estable ante cambios ambientales y condiciones de visibilidad. El que brinda más fiabilidad para la detección de sargazo es el FAI, debido a que es el menos influenciado por los efectos atmosféricos y brinda resultados con un fondo relativamente homogéneo, con el que se ha logrado detectar cobertura de algas aun en condiciones adversas, con reflejo considerable del Sol. FAI es un índice más fiable que NDVI y EVI, por lo que es recomendable para la detección de sargazo en regiones extensas [6, 7].

Localizar a tiempo la trayectoria y tamaño de estas manchas de algas flotantes es importante para comprender su impacto en la ecología y ayudar a la toma de decisiones para su control.

I. DATOS Y MÉTODOS

Se utilizó una serie de 880 imágenes MYD09A1, producto que proporciona una estimación de la reflectancia espectral de las bandas 1 a 7 obtenidas por el sensor MODIS a bordo del satélite Aqua [8] (Tabla 1).

Tabla 1. Bandas espectrales incluidas en el producto MYD09A1 (NIR=Near Infrared, SWIR=Short-wave Infrared).

Banda	Región espectral	Ancho de banda (nm)
1	Red *	620 - 670
2	NIR *	841 - 876
3	Blue	459 - 479
4	Green	545 - 565
5	NIR	1230 - 1250
6	SWIR*	1628 - 1652
7	SWIR	2105 - 2155

*Bandas utilizadas para la detección de sargazo. Las bandas de la región espectral roja, verde y azul se utilizaron para la representación de escenas en color verdadero.

Estas imágenes tienen cobertura global y se distribuyen en 460 cuadrantes identificados por un número consecutivo de este a oeste (0-35) y otro de norte a sur (0-17). Cada cuadrante (*tile*), consta de 2400×2400 píxeles, con resolución espacial de 500 m, proyectadas en un sistema de coordenadas sinusoidal (Fig. 1).

El valor de cada píxel en estas imágenes se establece seleccionando un valor de las adquisiciones realizadas por el sensor en un período de 8 días, con la finalidad de mejorar la calidad de la imagen, reduciendo los píxeles con cobertura de nubes y con reflejo del sol (*sun glint*) excesivo [8].

Las imágenes utilizadas en este trabajo corresponden a los 11 cuadrantes que cubren el Mar Caribe (Fig. 2). Para cada cuadrante se procesaron 46 imágenes por año (composiciones de 8 días), por lo tanto, para el periodo de enero de 2018 a septiembre de 2019 se requirieron 80 escenas por cuadrante. Los procedimientos para el tratamiento de estas imágenes se desarrollaron en lenguaje de programación en R con el entorno de desarrollo RStudio, utilizando los paquetes MODIS, rgdal, gdalUtils, raster, maptools, tmap, rasterVis, RStoolbox, entre otros. El paquete MODIS se utilizó para realizar la descarga automática de imágenes en formato HDF, desde los servidores LAADS DAAC (Level 1 and Atmosphere Archive and Distribution System Distributed Active Archive Center) de la NASA. El resto de los paquetes de R se usaron para acceder a los archivos HDF, realizar las operaciones entre las capas ráster para calcular el FAI, reproyectar los datos geográficos, así como para generar los mapas y gráficas.

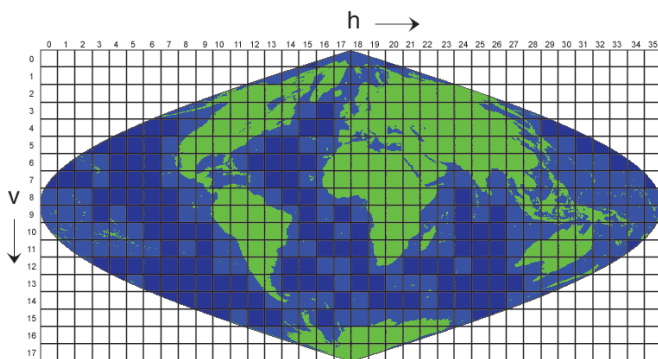


Fig. 1. Malla sinusoidal MODIS. La superficie terrestre es cubierta con 460 cuadrantes de $10 \times 10^\circ$ aproximadamente. Imagen tomada de [8].

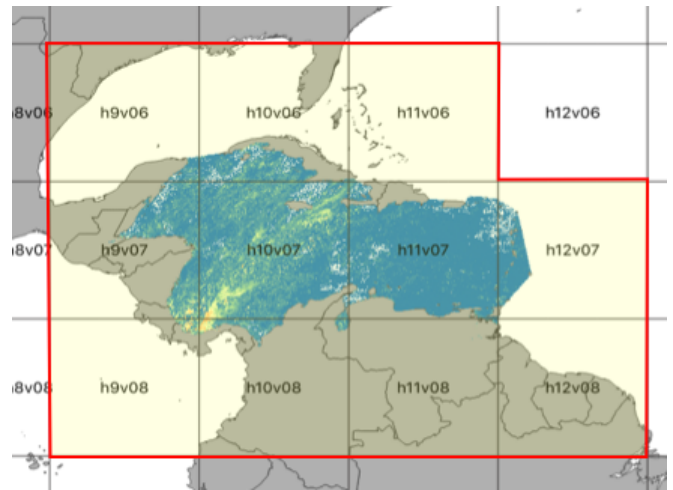


Fig. 2. Cobertura espacial de las 11 escenas utilizadas para generar los 80 mosaicos imágenes MYD09A1 para el Mar Caribe.

La Fig. 3 muestra un esquema conceptual de las fases de procesamiento, que incluye: a) la obtención de imágenes, b) cálculo del FAI, c) generación de mosaico en proyección sinusoidal, d) reproyección a coordenadas geográficas, e) extracción de la región de interés (Mar Caribe), f) almacenamiento de imágenes FAI en una *stack*, g) segmentación de imágenes para generar imágenes binarias a partir de un valor umbral, y h) generación de mapas (FAI, RGB) y gráficas.

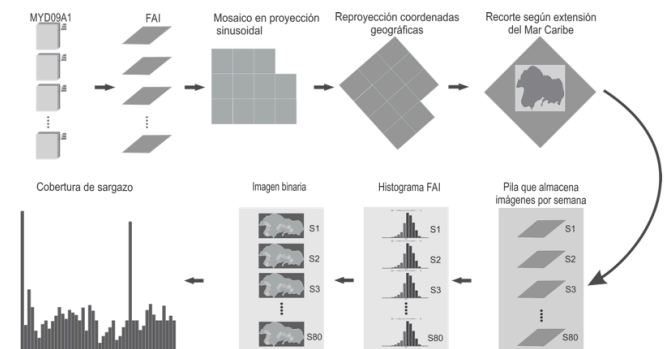


Fig. 3. Esquema general de procesamiento, desde la obtención de imágenes hasta la generación de mapas y gráficas de la variación en la cobertura (km^2) de sargazo.

A continuación, se describe la implementación del algoritmo para calcular el índice de algas flotantes [5, 6].

A. Índice de algas flotantes

El índice de algas flotantes (FAI) introduce la reflectancia corregida de Rayleigh (R_{rc}), eliminando los efectos de la dispersión molecular (Rayleigh):

$$R_{rc} = \pi L_t^* / (F_0 \cos \theta_0) - R_r$$

Donde L_t^* es el sensor de resplandor calibrado después de un ajuste para la absorción del ozono y otros gases, F_0 es la irradiación solar extraterrestre en el momento de adquisición de la información, θ_0 es el ángulo de cenit solar y R_r es la reflectancia Rayleigh estimada con 6S (Second Simulation of the Satellite Signal in the Solar Spectrum). Se usa R_{rc} en las

longitudes de onda de 645, 555 y 469 nm para los canales R (rojo), G (verde), B (azul) respectivamente en la imagen compuesta (RGB). El espectro de las algas flotantes tiene un pico distinguible en los 859 nm, este contraste se mantiene incluso cuando las condiciones de la atmósfera son más turbias y existe brillo (reflejo) solar [5, 7]. El FAI se define como [5]:

$$FAI = R_{rc,NIR} - R'_{rc,NIR}$$

$$R'_{rc,NIR} = R_{rc,RED} + (R_{rc,SWIR} - R_{rc,RED}) \times \frac{(\lambda_{NIR} - \lambda_{RED})}{(\lambda_{SWIR} - \lambda_{RED})}$$

Donde $R'_{rc,NIR}$ es la reflectancia en la banda NIR obtenida de la interpolación lineal entre las bandas roja y SWIR. Para implementar la fórmula con los datos MODIS, se utilizaron los valores centrales de las bandas: $\lambda_{RED} = 645$ nm, $\lambda_{NIR} = 859$ nm, $\lambda_{SWIR} = 1240$ nm [5].

III. RESULTADOS Y DISCUSIÓN

La composición de imágenes en color verdadero (RGB) se realizó para cada uno de los 11 cuadrantes que cubren la región de interés, conformando los 880 mosaicos para el periodo de estudio. La Fig. 4 muestra una composición (RGB) de estos cuadrantes para las escenas correspondientes a la primera semana de 2018. Los polígonos que representan los países están en coordenadas geográficas WGS84 y la malla (cian) que delimita la cobertura de los cuadrantes está en coordenadas sinusoidales.

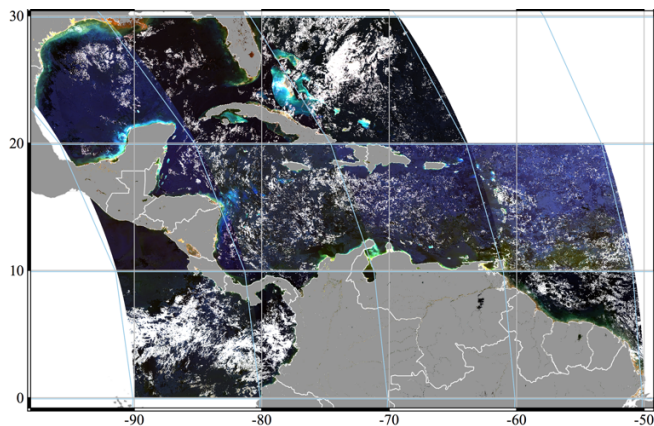


Fig. 4. Mosaico de los 11 cuadrantes MYD09A1 que cubren la región de interés. Composición en color verdadero (RGB) para la primera semana de 2018. Las líneas cian representan la malla en sistema de coordenadas sinusoidal y la gris en coordenadas geográficas WGS84.

Una vez conformados estos mosaicos con las bandas espectrales requeridas para el tratamiento (R, G, B, NIR, SWIR), se calculó el FAI para cada uno (80 mosaicos de composiciones de 8 días para el periodo de enero de 2018 a septiembre de 2019). La secuencia del FAI consistió de 80 imágenes, a partir de las cuales se obtuvieron mapas de la distribución espacial del sargazo en el Mar Caribe. La Fig. 5 muestra el mapa correspondiente al primer mosaico de la secuencia. Los gráficos que se muestran arriba y a la derecha del mapa, corresponden al valor acumulado del FAI por columnas y por filas, respectivamente.

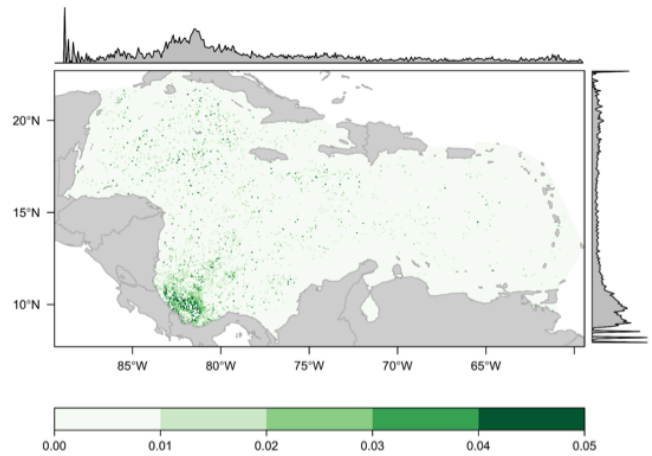


Fig. 5. Distribución espacial del FAI para la primera semana de 2018. Los gráficos en la parte superior y a la derecha del mapa, corresponden al FAI acumulado por columnas y por filas, respectivamente.

Las 80 imágenes que representan la distribución del FAI (Fig. 5) fueron segmentadas en regiones con mayor probabilidad de indicar la presencia de sargazo, generando imágenes binarias. Estas imágenes binarias permitieron estimar la cobertura de sargazo en km^2 , al contabilizar los píxeles con nivel digital igual a 1 y considerando el tamaño del píxel (0.25 km^2).

La gráfica de la Fig. 6, muestra las variaciones en la cobertura de sargazo obtenida para cada mosaico del FAI segmentado. Los valores máximos se presentaron en las composiciones de 8 días, que iniciaron el 01/01/2018 y el 13/06/2019, con más de $13,000 \text{ km}^2$ (Fig. 6). Sin embargo, consideramos que estos valores podrían estar sobreestimados debido al efecto de algunas condiciones que dificultan la estimación del FAI. Estos efectos deben atenuarse en las imágenes, por ejemplo, con el correcto enmascaramiento de nubes, la eliminación más eficaz de áreas con reflejo del Sol (*sun glint*) y aerosol marino (mejorar la corrección atmosférica). No obstante, la tendencia general es congruente (excepto en el primer valor de la serie) con los mayores arribos reportados en verano para esta región.

Adicionalmente, se extrajo la fecha, ubicación e índice de algas flotantes (fecha, latitud, longitud, FAI) de las coberturas de sargazo más conspicuas (en las imágenes binarias), generando un archivo *shapefile* con esa estructura. Estos datos se utilizarán en una etapa posterior, como criterio para la selección espaciotemporal de imágenes con mayor resolución espacial (Landsat-8 y Sentinel-2), y así posibilitar una detección más precisa de la cobertura del sargazo en estas áreas identificadas con MODIS. Esto permitirá también la detección de sargazo en zonas cercanas a la costa, misma que se complica con imágenes de mediana resolución (500 m), debido a la variedad de ecosistemas en estas regiones someras, lo que plantea un reto mayor para las técnicas descritas hasta ahora.

IV. CONCLUSIONES

Aunque la tendencia del arribo de sargazo al Mar Caribe en general concuerda con los resultados obtenidos (excepto en dos mosaicos en los que la cobertura es sobreestimada), es necesario implementar un método más eficiente para su detección a múltiples escalas espaciales y temporales. También es necesario el correcto enmascaramiento de los efectos de las condiciones ambientales que afectan la calidad de los píxeles, con la finalidad de automatizar este procedimiento. Para dar continuidad a este estudio, se propone la implementación de algoritmos de aprendizaje automático aplicado a imágenes multi-resolución, así como la incorporación de datos de dirección e intensidad de corrientes marinas (HYCOM) para predecir la trayectoria y tiempo de arribo del sargazo a zonas sensibles. También es necesario incorporar el estudio de la dinámica de otros parámetros ambientales que pudieran explicar las causas y origen de estos afloramientos masivos.

AGRADECIMIENTOS

Los autores agradecen al Centro Interdisciplinario de Ciencias Marinas del Instituto Politécnico Nacional, por el apoyo para la realización de este trabajo a través del Proyecto SIP-20195187.

REFERENCIAS

- [1] Lapointe, Brian & Herren, Laura & Debortoli, David & Vogel, Margaret. (2015). Evidence of sewage-driven eutrophication and harmful algal blooms in Florida's Indian River Lagoon. *Harmful Algae*. 43. 10.1016/j.hal.2015.01.004.
- [2] Gower, J., Young, E., & King, S. (2013). Satellite images suggest a new Sargassum source region in 2011. *Remote Sens. Lett.* 4, 764–773.
- [3] Witherington, B., Hiram, S., & Hardy, R. (2012). Young sea turtles of the pelagic Sargassum-dominated drift community: Habitat use, population density, and threats. *Marine Ecology Progress Series*, 463, 1–22.
- [4] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2014). TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation.
- [5] Hu, C. (2009). A novel ocean color index to detect floating algae in the global oceans. *Remote Sensing of Environment*, 113, 2118-2129.
- [6] Hu, C., Feng, L., Hardy, R. F., & Hochberg, E. J. (2015). Spectral and spatial requirements of remote measurements of pelagic Sargassum macroalgae. *Remote Sensing of Environment*, 167, 229-246.
- [7] Wang, M., & Hu, C. (2016). Mapping and quantifying Sargassum distribution and coverage in the Central West Atlantic using MODIS observations. *Remote Sensing of Environment*, 183, 350-367.
- [8] Vermote, E. (2015). MYD09A1 MODIS/Aqua Surface Reflectance 8-Day L3 Global 500m SIN Grid V006 [Data set]. NASA EOSDIS Land Processes DAAC. Accessed 2020-02-28 from <https://doi.org/10.5067/MODIS/MYD09A1.006>

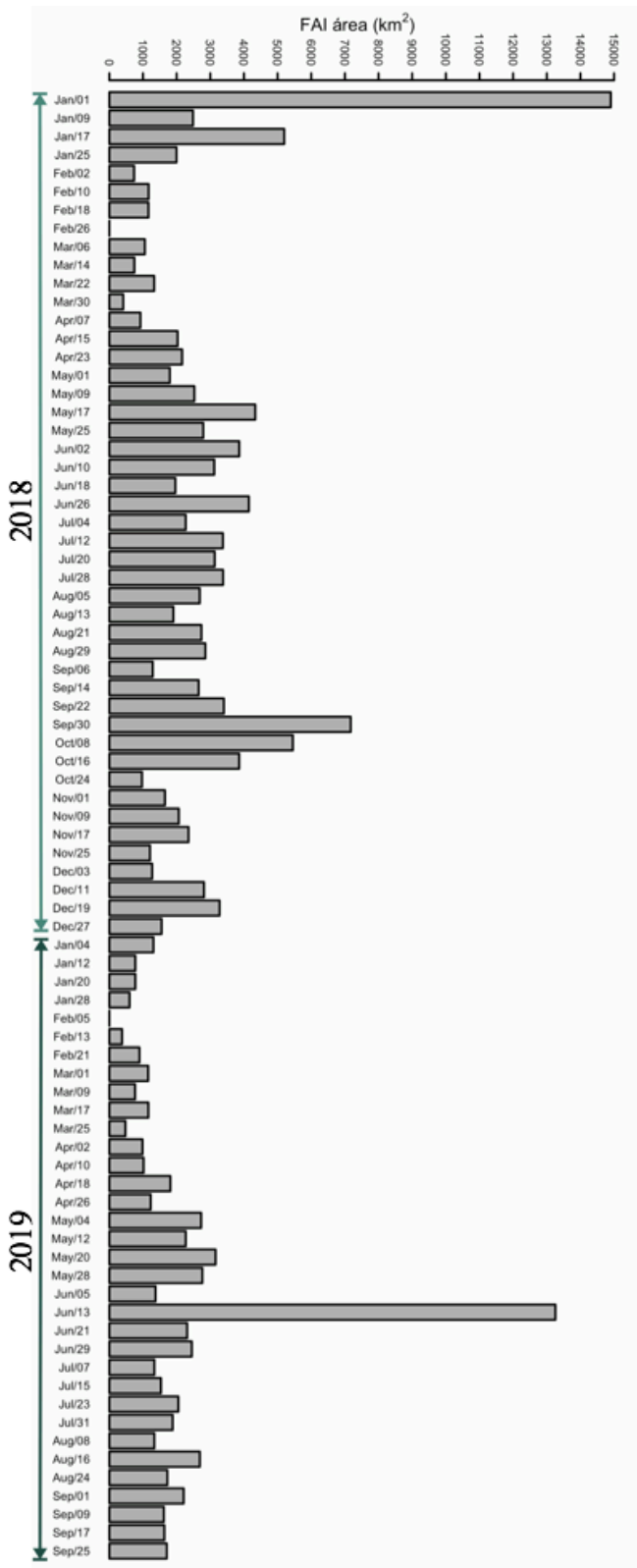


Fig. 6. Variación en la cobertura de sargazo (km²) en el Mar Caribe de enero de 2018 a septiembre de 2019.

SCI-HI: Pruebas de un radio-interferómetro de dos elementos implementado en Roach2 con ancho de banda 40-120MHz

Luis Javier Ulloa-Velazquez

Facultad de Ciencias de la Electrónica
Benemérita Universidad Autónoma de Puebla (BUAP)
Puebla, México
ulloa.200921740@gmail.com

Omar López-Cruz

Coordinación de Astrofísica
Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Tonantzintla, Puebla, México
omarlx@inaoep.mx

Germán Ardul-Muñoz

Facultad de Ciencias de la Electrónica (BUAP)
Benemérita Universidad Autónoma de Puebla
Puebla, México
germanardul.munoz@correo.buap.mx

Olga Félix-Beltrán

Facultad de Ciencias de la Electrónica (BUAP)
Benemérita Universidad Autónoma de Puebla
Puebla, México
olga.felix@correo.buap.mx

Richard Torrealba-Meléndez

Facultad de Ciencias de la Electrónica (BUAP)
Benemérita Universidad Autónoma de Puebla
Puebla, México
richard.torrealba@correo.buap.mx

Resumen—En este trabajo se presentan las pruebas de integración de un radio-interferómetro de dos antenas para el experimento “Sonda Cosmológica de la Isla” para la detección del Hidrógeno Neutro (SCI-HI), con un ancho de banda de 40-120MHz. El sistema propuesto está basado en la plataforma de la tarjeta Roach2 con tecnología FPGA. El diseño electrónico del interferómetro, así como su implementación en la tarjeta Roach2 son presentados. Una interfaz de comunicación y visualización Roach2-PC es creada usando el lenguaje de programación Python; esta tiene el propósito de cargar el archivo de programación a dicha tarjeta, ajustar las señales y crear archivos log para su posterior análisis, se muestra una segunda interfaz encargada de tomar los archivos log y realizar un análisis estadístico de los espectros almacenados en un tiempo dado de muestreo.

Index Terms—SCI-HI, Interferencia, antenas, Roach2, espectro de radio, front-end, Simulink-MATLAB, Python.

I. INTRODUCCIÓN

El proyecto SCI-HI es un proyecto mexicano con el objetivo de explorar la formación de las primeras estrellas en el Universo, SCI-HI busca determinar inflexiones en el espectro integrado a distintas épocas (40-120 MHz) de la línea de 21 cm del hidrógeno neutro. Estas observaciones nos ayudarían a detectar la formación de las primeras estrellas, galaxias y fin de la época oscura del Universo. El segundo prototipo a ser probado en la Reserva de la biosfera Isla Guadalupe (lat: 28°58'24" N, lon: 118°18'4" W) a unos 260 km de la península de Baja California, el primer prototipo constaba de solo una antena de banda ancha (Hibiscus) [1]. En la Fig.

1 se observa la evolución temporal de las fluctuaciones en el fondo de 21cm [2]. SCI-HI busca la absorción que se observa alrededor de $z = 20$ (80MHz), esto corresponde aproximadamente a unos 180 millones de años después del Big Bang, este espacio-tiempo ha permanecido inexplorado [1].

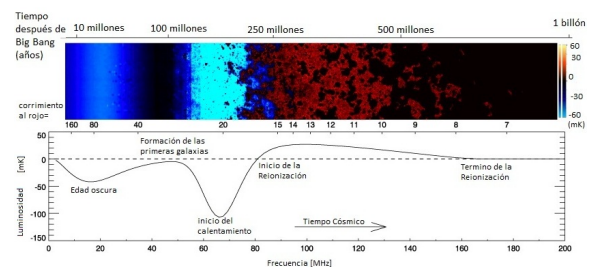


Figura 1. Características relevantes del tiempo cósmico, frecuencia y corrimiento al rojo [2].

Para mejorar la detección en cuanto a la eliminación de efectos sistemáticos [3], se propone un radio-interferómetro de 2 antenas con un nuevo diseño de antena (Mango Peel) representado en la Fig. 2. El diseño del radio-interferómetro está basado en los tutoriales del grupo Casper¹ (Collaboration for Astronomy Signal processing and Electronics Research).

¹<https://casper-toolflow.readthedocs.io/projects/tutorials>

Asimismo, el retardo instrumental compensatorio es realizado mediante un bloque que utiliza BRAM² para su almacenamiento y es un retardo programable en tiempo de ejecución, por lo que el retardo está dado en ciclos de reloj, partiendo de la frecuencia de ejecución de la FPGA, que es de 100MHz. Entonces, cada unidad de retraso de la secuencia de señal que entra al bloque será retardada 10ns, esto nos ayuda a alinear una señal de datos específicos con otra señal que se encuentra retrasada por diferencia de longitud de recorrido, desde la antena hasta el convertidor, o por un ángulo de medición $Am \neq 0$ medido a partir del cenit.

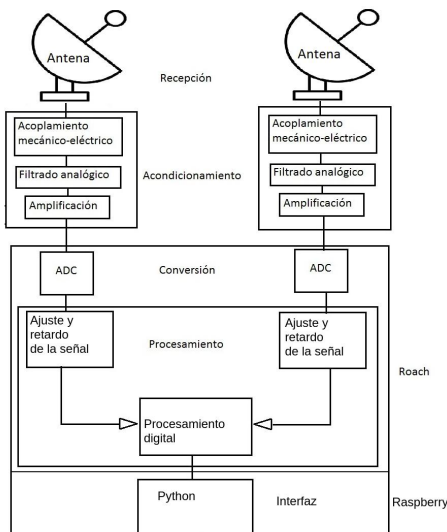


Figura 2. Esquema general del interferómetro implementado. El procesamiento de las señales captadas incluyen los ajustes necesarios a la señal e interferirlas para obtener el espectro de interés. Se realiza la interfaz y de comunicación con lenguaje de programación Python.

El procesamiento de las señales cósmicas se realiza con equipos dedicados CASPER³, en específico con la tarjeta Roach2⁴ (*Reconfigurable Open Architecture Computing Hardware 2*). Roach2 es un hardware de procesamiento independiente de FPGA basado en Virtex 6; cuenta con un PC embebido y varios bloques de memoria; para la distribución de tareas de la FPGA a todos los dispositivos periféricos, el ADC con el que se trabaja tiene una frecuencia de muestreo $F_s = 400\text{MHz}$, el código de programación se realiza mediante Simulink-MATLAB a través de bloques prefabricados por CASPER y Xilinx. Para poder llevar la señal desde la antena hasta las entradas del ADC, que contiene la Roach2, es necesaria una etapa de electrónica compuesta por filtros pasivos en el rango de interés y amplificadores activos con ganancias considerables.

II. INTERFERÓMETRO

Se considera un radio-interferómetro al instrumento con la capacidad de hacer interferir las señales captadas por dos

²<https://casper-toolflow.readthedocs.io/en/latest/blockdocumentation.html>

³https://casper.berkeley.edu/wiki/Main_Page

⁴<https://casper.ssl.berkeley.edu/wiki/ROACH2>

antenas diferentes denotadas por $v_{c1}(t)$ y $v_{c2}(t)$, con dichas antenas separadas por una línea de base b como se observa en la Fig. 3, tomando en cuenta que los receptores son acondicionados para proporcionar cualquier amplificación y filtrado en las frecuencias de interés.

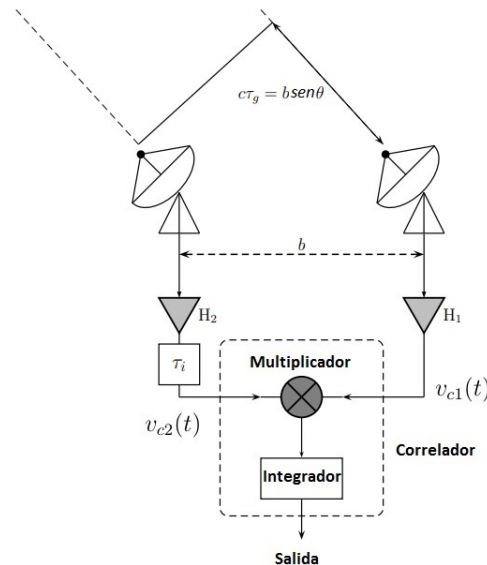


Figura 3. Esquema de un interferómetro de dos elementos simplificado; H_1 y H_2 representan la cadena de receptores integrados a su electrónica, τ_g es el retardo de tiempo geométrico y τ_i es el retardo de tiempo instrumental compensatorio [4].

La energía que se propaga en una línea de transmisión viaja a una velocidad menor que la velocidad de la luz c , y está relacionada con el factor de velocidad V_f [6]:

$$V_f = \frac{V_p}{c}, \quad (1)$$

donde V_f es el factor de velocidad y V_p es la velocidad de propagación en la línea.

El factor de velocidad para una línea de transmisión depende casi por completo del dieléctrico que contiene dicha línea, dada por

$$V_f = \frac{1}{\sqrt{\epsilon_r}}, \quad (2)$$

con ϵ_r como la permitividad relativa del dieléctrico. Comúnmente los fabricantes incluyen la velocidad de propagación de los cables para RF, en este caso, para las pruebas se utiliza un cable con $V_f = 0.76$, por lo que se tiene una velocidad de propagación dada como :

$$V_p = 300 \times 10^6 \text{ m/s} \times 0.76 = 228 \times 10^6 \text{ m/s}. \quad (3)$$

Al saber el tiempo de retraso mínimo que nos genera el bloque y la velocidad de propagación en la línea de transmisión, podemos conocer la longitud que recorre la señal por unidad de retraso ($t = 1\text{ns}$) o el retardo temporal necesario a una señal para poner ambas señales en fase a través de la relación:

$$D_l = V_p n_r b_r, \quad (4)$$

donde D_l representa la diferencia de longitud del recorrido de las señales (en metros), n_r es el número de retardos aplicado al bloque (adimensional), b_r es la base de tiempo de retardo (1×10^{-9} s) equivalente un ciclo de reloj al que trabaja la FPGA, misma que se establece a $1/4$ de la frecuencia de muestreo del ADC. Por otro lado, se realiza una pequeña calibración en amplitud, puesto que dada una señal de prueba a ambas entradas del ADC no cuantifican el mismo valor, por lo que se les agrega un offset programable para poner el punto de referencia de cada señal en cero.

El ancho de banda de un ADC está definido básicamente por la frecuencia de muestreo, que debe ser por lo menos dos veces la frecuencia máxima a procesar [7], por lo que basta una frecuencia de muestreo de $2B$ muestras/s (B es el ancho de banda). Se garantiza una reconstrucción perfecta para el límite de banda, entonces el ancho de banda del ADC es $B = \frac{F_s}{2} = 200\text{MHz}$. Dado que el ADC tiene una resolución de 14 bits, puede codificar entradas analógicas desde 0 hasta 16384 niveles diferentes. Cabe agregar que la entrada de voltaje que admite dicho ADC es de $2.2V_{pp}$, al tomar como referencia central 0V el rango de voltaje admitido es de -1.1 a 1.1V . A partir de estos datos podemos conocer la resolución de voltaje llamada Q , dada por el límite de voltaje superior menos el límite de voltaje inferior dividido entre el número de niveles, por lo que se pueden cuantificar pasos de $134\mu\text{V}$.

II-A. Integración de la electrónica y propuesta de diseño para procesamiento en Roach2

La propuesta de la electrónica que se contempla al final del proyecto se encuentra descrita en la Fig. 4, mismo que se pretende sea portable en las etapas finales. Esto facilitaría su traslado y eliminaría el ruido eléctrico inducido por la corriente de línea (120V AC) requerida por las fuentes de los diferentes dispositivos, como los amplificadores, señal de reloj e inclusive la misma Roach2.

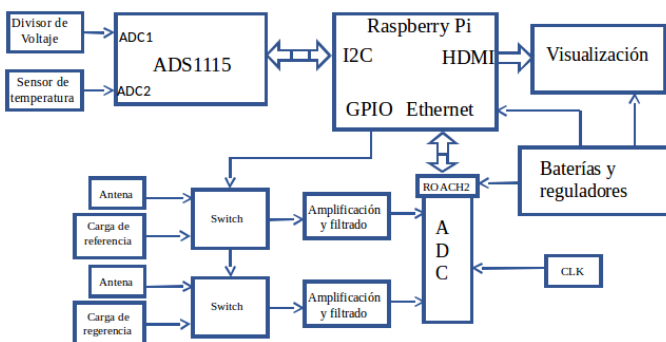


Figura 4. Esquema general de la electrónica para la operación del radio-interferómetro. Se presenta el suministro de energía eléctrica, el monitoreo de variaciones de temperatura en el entorno y caídas de voltaje en el suministro eléctrico que pueda hacer variar los resultados; el control de dos switches para la conmutación entre las señales de las antenas y una carga de referencia, la etapa de front-end que incluye detectar, filtrar y amplificar la señal; una señal de reloj que es ingresada al ADC, lo que hereda una fracción de la frecuencia de reloj a la FPGA, mini PC Raspberry Pi, que se encarga de realizar la extracción de información de la Roach2, así como visualizar y almacenar toda la información recabada.

En la Fig. 5 se observa el diagrama que describe la estructura de programación que se implementa en la Roach2. Las etapas de ADC, offset y retardo son con el fin de acondicionar la señal para realizar su interferencia, donde son calibradas las señales en amplitud y desfasadas mediante un retardo programable. La interferencia se realiza través de un detector de ley cuadrada (potencia lineal), considerando que la tensión de la señal de una antena está representada por

$$V \sin(2\pi\nu_0 t),$$

mientras que para la otra se considera un retardo temporal dado como:

$$\tau = (D/c) \sin \theta.$$

D es la distancia entre la separación de las antenas, c la velocidad de la luz y θ es la posición angular de la fuente, por lo que la señal será :

$$V \sin(2\pi\nu_0(t - \tau)).$$

Así, la salida del receptor que implementa el bloque interferencia se describe como [8]:

$$V_o = [V \sin(2\pi\nu_0 t) + V \sin(2\pi\nu_0(t - \tau))]^2 \quad (5)$$

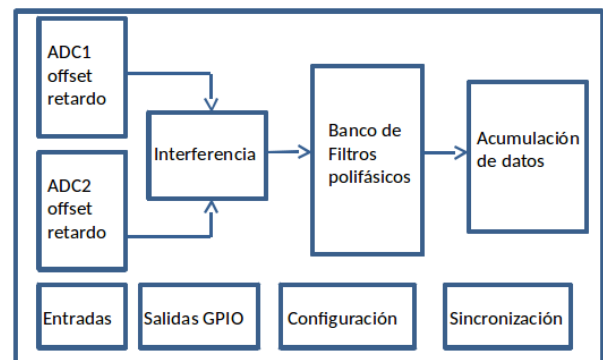


Figura 5. Diagrama que describe la programación de la Roach2 diseñada en Simulink-MATLAB.

La estructura de filtros polifásicos prototipo incluye la multiplicación de la función entrante por una función ventana; la señal es dividida en toques paralelos para aplicarles filtros de respuesta al impulso finito, seguida de una transformada rápida de Fourier. Por último, se implementa un acumulador de canales, donde los datos son tomados de la salida de la transformada rápida de Fourier y almacenada en registros par e impar hasta que se almacena el número de registros definidos por el usuario.

II-B. Interfaz

La comunicación entre la PC y la Roach2 se realiza mediante una interfaz desarrollada en Python con el uso de funciones de la librería `corr`⁵ que incluye funciones dedicadas a la comunicación PC-Roach2, en la Fig. 6 se muestra la pantalla principal de la comunicación PC-Roach2.

⁵<https://github.com/ska-sa/corr>

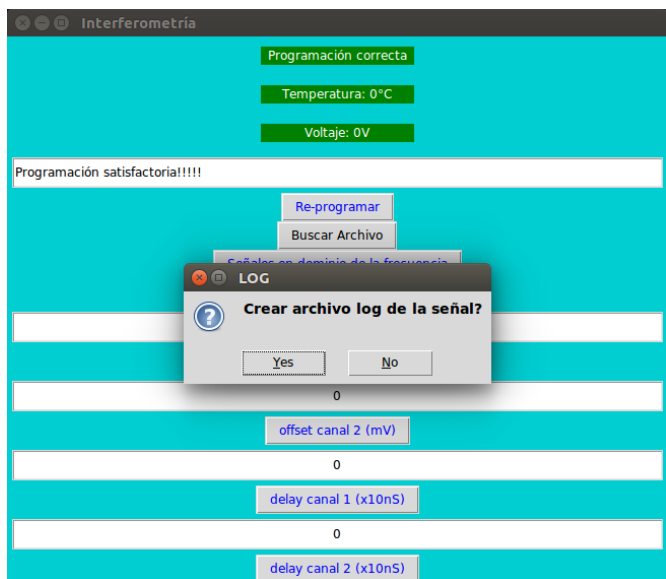


Figura 6. Ventana principal de la interfaz PC-Roach2. En una etapa posterior se pretende obtener variables asociadas a la Roach2 como voltaje y temperatura.

La anterior interfaz está hecha para cargar el firmware diseñado en Simulink previamente compilado, botones para el ajuste de las señales que incluyen retardo programable y offset para mantener dichas señales en fase. Mediante un hilo diferente se crea un log que almacena los datos del espectro, con el fin de realizar un análisis fuera de línea, por lo que fue creada una segunda interfaz que realiza un análisis estadístico, dicha interfaz mostrada en la Fig. 7, donde se carga el log de los espectros almacenados y se realiza el análisis de todas esas muestras. Al final podemos obtener una gráfica de un ancho de banda específico dentro de los 40 a 120MHz.

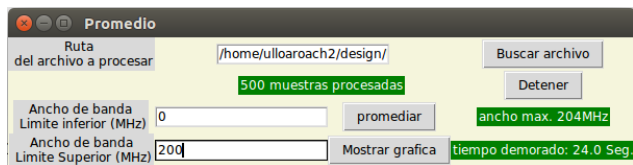


Figura 7. Interfaz para analizar los datos almacenados en un archivo log.

III. PRUEBAS DE INTEGRACIÓN

Para realizar mediciones de integración del radio-interferómetro se tiene un ambiente de pruebas en laboratorio, mostrado en la Fig. 8. Dicho ambiente es realizado dentro de una jaula de Faraday con el objetivo de eliminar las señales provenientes de estaciones de radio cercanas que se encuentran dentro del ancho de banda de nuestro interés. El analizador de espectros es utilizado para comparar los detalles de un espectro obtenido por la Roach2 contra el mismo analizador; el generador de señales 1 se utiliza para simular la recepción de una señal de muy baja potencia (antes de integrar la antena), mientras que el generador de señales 2 es utilizado como generador de señal de reloj hacia el ADC.

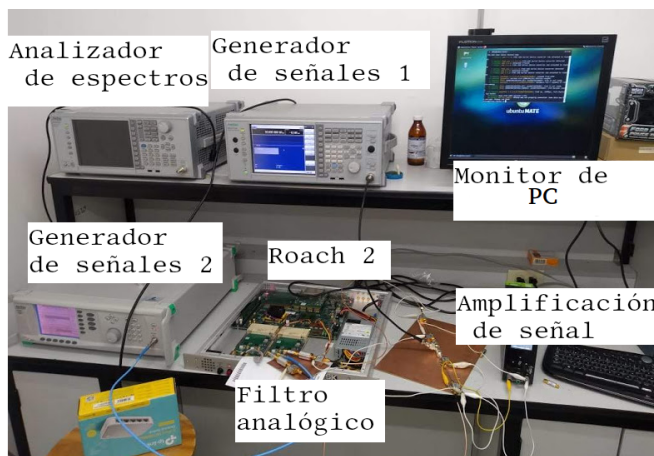


Figura 8. Ambiente donde se prueba el software, hardware y la integración del radio-interferómetro.

Se ajusta el generador de señales que simula nuestra señal recibida con una potencia de -100dBm , a una frecuencia de 80MHz . Dicha señal es dividida mediante un conector tipo T para ingresar la misma señal a las dos entradas, simulando la recepción en fase; mismo conector presenta atenuaciones en la práctica de 38dBm , por lo tanto se tiene que la potencia de la señal de prueba es de $\approx -138\text{dBm}$. En la Fig. 9 se observa el espectro con ancho de banda de $40\text{-}120\text{MHz}$, la forma de la gráfica incluye la respuesta del filtrado, amplificado y atenuación del sistema electrónico. A partir de esta gráfica, apenas es posible observar la señal inducida al realizar un zoom en la frecuencia deseada, que se observa en la Fig. 10. Dentro de las variaciones en diferentes frecuencias podemos observar que sobresale un pico en los 80MHz .

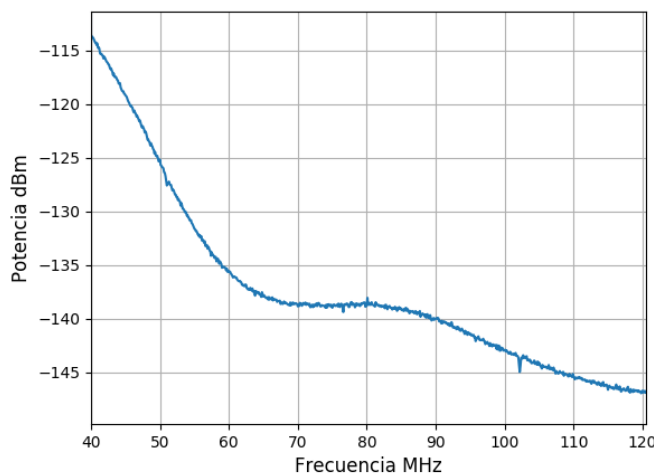


Figura 9. Espectro de un ancho de banda de $40\text{-}120\text{MHz}$. La forma de la respuesta incluye los efectos debido a la electrónica de "front-end", principalmente a la de los filtros pasivos, difícilmente es posible indicar qué frecuencia es ruido y cuál es inducida.

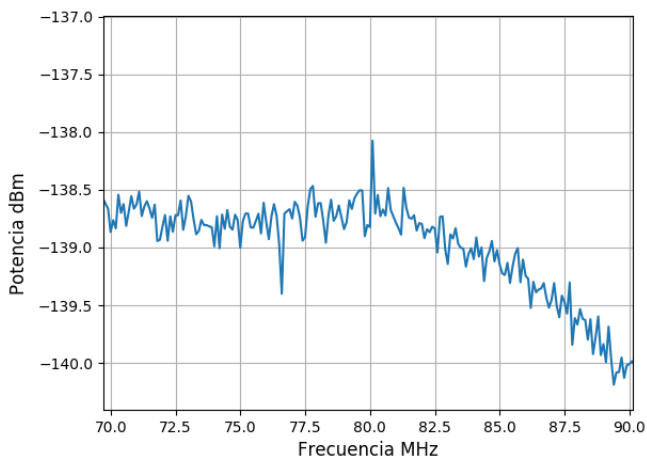


Figura 10. Espectro de un ancho de banda acotado, donde se puede apreciar una señal marcada en los 80MHz en un instante de tiempo.

Al tener el sistema realizando mediciones de los espectros y almacenarlos durante 10 min, al obtener el promedio de esas muestras, se puede observar que los niveles de ruido se reducen y se remarca la señal que fue inducida, mismo que se aprecia en la Fig. 11. Por otro lado, en la Fig. 12, podemos apreciar la comparación de la misma fracción de ancho de banda, donde es más apreciable que existen señales en ciertas frecuencias que son remarcadas después de hacer el análisis estadístico en un tiempo de muestreo y difícilmente se observan? en tiempo real.

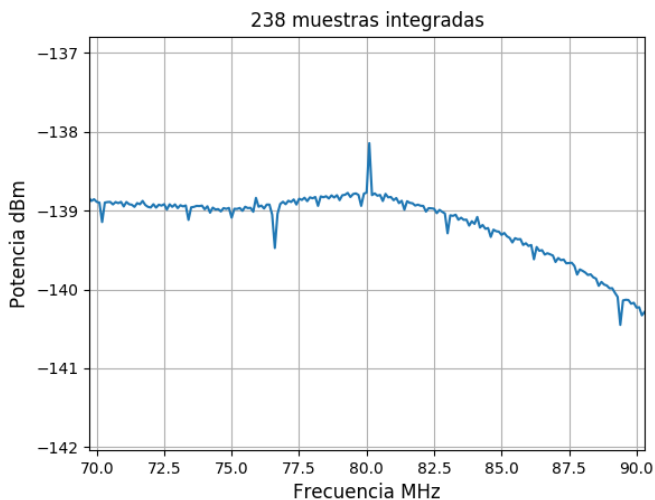


Figura 11. Espectro de una fracción del ancho de banda, con un promedio de observación de 10 min.

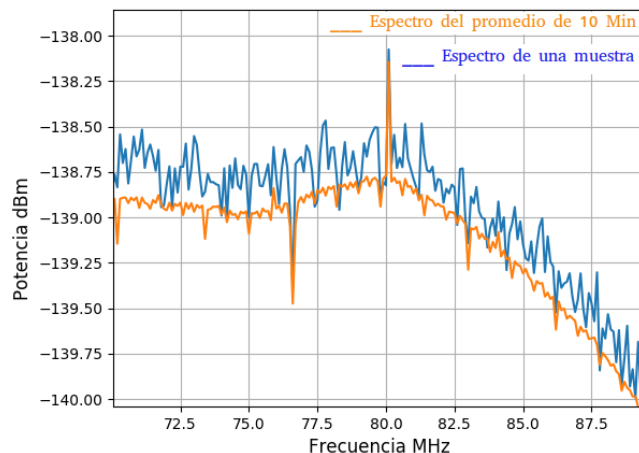


Figura 12. Comparación del espectro tomado en un instante de tiempo (azul) contra el espectro que resulta de promediar 238 muestras equivalente a 10 min de muestreo (naranja).

IV. CONCLUSIONES

Hasta este punto se ha logrado realizar la integración del radio-interferómetro de dos elementos junto con su interfaz para la comunicación, visualización y almacenamiento de datos, además del programa que realiza el análisis estadístico de los datos almacenados. Se logró observar el espectro de una señal inducida de potencia por debajo del ruido mediante un generador de señales, misma que se remarcan al realizar un análisis estadístico de los datos, al mismo tiempo que las componentes espectrales del ruido desaparecen, es posible indicar que para realizar una observación más precisa se debe saber diferenciar la señal esperada del ruido debido a los componentes del sistema, inserciones por fenómenos naturales e inclusive interferencias por origen humano, mientras más prolongado sea el tiempo de recolección de datos mas fiable serán los resultados promediados, en mediciones, la estrategia de orientación de la antena o un mal retardo de la señal para su análisis puede ser también los culpables de un mal resultado del procesamiento de la señal, el tiempo que dura el muestreo de la señal es conocido como tiempo de integración, por lo que entre más tiempo de observación se tenga, más marcadas serán las señales del continuo.

REFERENCIAS

- [1] Voytek, T. C., Natarajan, A., Jáuregui García, J. M., Peterson, J. B., & López-Cruz, O. 2014, *Astrophysical Journal Letters*, 782, L9
- [2] 21 cm cosmology in the 21st century, Jonathan R Pritchard and Abraham Loeb, 2012.
- [3] Liu, A., Pritchard, J. R., Tegmark, M., et al. 2013, *Physical Review D*, 87, 043002
- [4] Juan- Pierre Jansen van Rensburg, The design of a two-element correlation interferometer operating at L-band, Stellenbosch University, 2012.
- [5] <https://casper.ssl.berkeley.edu/wiki/ROACH2>
- [6] Blake & Roy, *Sistemas electrónicos de comunicaciones*, Thomson, 2014.
- [7] Irrázaval M. Pablo, *Análisis de Señales*, Segunda Edición, Departamento de Ingeniería Eléctrica, Pontificia Universidad Católica de Chile, 2008.
- [8] *Interferometry and Synthesis in Radio Astronomy*, A. R. Thompson, J. M. Moran, and G. W. Swenson, Jr, 2012. Wiley, New York, 2nd edition, 2001.

Diseño de un Gateway Lora/WiFi como punto de acceso a una red de de largo alcance para mensajería de texto

Jonathan Alejandro Reyes Gracia
Tecnológico Nacional de México
Campus La Paz
La Paz, B.C. S., México
M20310002@lapaz.tecnm.mx

Ruben Ganiz Mero
Tecnológico Nacional de México
Campus La Paz
La Paz, B.C.S., México
L15310248@lapaz.tecnm.mx

Juan Pablo Morales Álvarez
Tecnológico Nacional de México
Campus La Paz
La Paz, B.C.S., México
juan.ma@lapaz.tecnm.mx

Daniel Armando Hiraes-Valles
Tecnológico Nacional de México
Campus La Paz
La Paz, B.C.S., México
daniel.hv@lapaz.tecnm.mx

Jorge Alejandro Rubio Montiel
Tecnológico Nacional de México
Campus La Paz
La Paz, B.C.S., México
gorgue4714@gmail.com

Resumen—Se presentan las actividades requeridas para la elaboración de un dispositivo *gateway* funcional que, en colaboración con otros, permite crear una red de tipo *mesh* para mensajería a través de dispositivos móviles haciendo uso de las tecnologías *WiFi* y *LoRa* para lograr comunicación sin depender de redes celulares o Internet, proponiendo así, una alternativa de comunicación eficiente y segura. Se describe el proceso de transmisión de datos a través de la red, las características y alcances de la misma, se muestra el diseño por computadora de una carcasa para el circuito integrado, se incluye el proceso de creación de una aplicación móvil para el protocolo *wifi* vía *sockets* y el diseño de nodos repetidores para la red. El nombre corto de los *gateways* fue definido como *Hermesh* para efectos prácticos, comerciales y de este proyecto en general.

I. INTRODUCCIÓN

Las redes de comunicación convencionales para la sociedad actual son las redes celulares y el Internet, sin embargo, dichas redes no cubren en su totalidad las zonas rurales y/o marginadas. Se cuenta con 83.1 millones de usuarios de telefonía celular, lo cual es el 73.5 % del total de las personas con una edad de entre seis años o más. Dejando con ello a 29.9 millones de personas desconectadas, un 26.5 % de este total [1]. El número de usuarios en México conectados a Internet, es de 74.3 millones de personas [1]. El Consejo Nacional de Evaluación de la Política de Desarrollo Social (CONEVAL) dentro de los indicadores de carencia social, señala que en el 2016 23.7 millones de personas se encontraban con carencia de acceso a los servicios básicos en la vivienda y para el 2018 la cifra aumentó a 24.7 millones de personas [2]. La Organización de las Naciones Unidas para la Alimentación y la Agricultura en su informe México rural del Siglo XXI de 2018 [3], establece que la población rural representa el 23 % de la población de México, la Encuesta Nacional de la

Dinámica Demográfica (ENADID) 2018 indica que residen 124.9 millones de personas en México, y 28.7 millones viven en alguna población rural [4]. Con lo anterior se deduce que durante el 2018 de las 29.9 millones de personas sin teléfono celular, 28.7 millones viven en zonas rurales y 24.7 millones carecen de servicios básicos [?].

El *gateway LoRa/WiFi* que se diseñó, ofrece interconectar una red de comunicación por mensajería de textos en la que se comuniquen en una red independiente dispositivos móviles, ya que no requiere de cobertura móvil o de Internet, tal idea está sustentada en el uso y combinación del protocolo *LoRa* para la comunicación de largo alcance y el protocolo *WiFi* como señal de corto alcance; el canal de comunicación generado es consistente y seguro para comunicaciones de emergencia y convencionales. A diferencia de otros dispositivos de radio-frecuencia, *LoRa* posee cifrado de seguridad asimétrico que impide la lectura de los mensajes por terceras personas y su fuente de alimentación incluye una batería recargable a través de un panel solar mejorando la autonomía del dispositivo. La idea consiste en colocar *gateways* en puntos estratégicos que permitan interconectar una red y cubrir las zonas objetivo para las y los usuarios que adquieran un dispositivo *gateway* y deseen hacer uso de la red. Los *gateway LoRa/WiFi* tienen la capacidad de comunicarse entre sí hasta 15 km en línea de vista.

Se exponen las limitaciones del dispositivo en su etapa actual de prototipo y la escalabilidad del mismo.

II. JUSTIFICACIÓN

El desarrollo tecnológico de los últimos años ha tenido impacto para la sociedad en diferentes aspectos, siendo uno de los mas importantes la comunicación, la gran mayoría de la población mundial se encuentra en constante comunicación

entre sí, gracias a la infraestructura conformada por sistemas de postes, cableados y antenas, que por lo general se ubican exclusivamente en zonas urbanizadas donde se concentra la mayoría de la población y por razones de costos o complejidad de las instalaciones normalmente se excluyen aquellas zonas que se encuentran muy retiradas o con baja densidad de residentes. Las áreas rurales tienen un gran potencial de crecimiento económico vinculado a la producción alimentaria en los países en desarrollo y a los servicios turísticos que se pueden ofrecer. Contar con un canal de comunicación independiente y de único pago permitirá a los habitantes de las comunidades rurales mejorar su calidad de vida e impulsar el mercado local.

Otros beneficios de esta red de comunicación radican en la portabilidad de los dispositivos, permitiendo que la topología de la red sea adaptable y que los nodos se reubiquen en otras zonas donde se necesiten o se agreguen más para expandir la cobertura sin ninguna instalación o programación específica. El protocolo *WiFi* en este *gateway*, es una señal de corto alcance que proporciona el importante beneficio de interconectar hasta 250 usuarios en radio con línea de vista hasta 100 metros, a diferencia de protocolos como *Bluetooth*. El protocolo *LoRa* es para la comunicación de largo alcance, la configuración de este *gateway* permitiría interconectar hasta 100 nodos *LoRa* por *gateway* o Repetidor *LoRa* [9], sin embargo esa cifra no es pequeña en una Red que se configure en tipo *mesh* y donde sus dispositivos se encuentran entre sí en un rango de entre 5 y 15km. *LoRa*, incluye un cifrado asimétrico de seguridad para que los mensajes en caso de ser interceptados por receptores no deseados no puedan ser leídos. Este Gateway es autónomo y confiable para comunicaciones de emergencia o convencionales.

III. REDES INALÁMBRICAS

Las redes inalámbricas son redes que utilizan ondas de radio para conectar los dispositivos funcionando de manera similar a las redes cableadas [5], sin embargo, las redes inalámbricas deben convertir las señales que contienen información en una forma adecuada para la transmisión a través del espacio. La infraestructura inalámbrica en comparación con la alternativa cableada tradicional permite a los dispositivos remotos conectarse sin dificultad y a largas distancias [5].

La Fig. 1 muestra un diagrama que clasifica las redes inalámbricas según su alcance y tasa de transferencia de datos.

III-A. Red de amplia cobertura y baja potencia.

Existen múltiples propuestas de redes *Low-Power Wide Area Network* (LPWAN por sus siglas en inglés), casi todas ellas presentan características comunes:

- Operar en la banda libre *ISM* por debajo de 1 GHz. Este tipo de señales presentan un menor factor de atenuación al atravesar obstáculos respecto a la banda libre *ISM* de 2.4 GHz.
- Poseer esquemas de modulación que posibilitan la creación de enlaces de varios kilómetros en zonas urbanas y

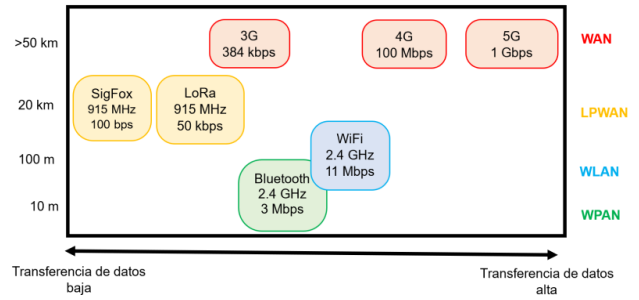


Figura 1. Clasificación de redes inalámbricas según alcance y tasa de transferencia. [6]

de decenas de kilómetros en zonas rurales con muy bajo coste energético.

Si bien hay varias redes o propuestas de redes *LPWAN* que cumplen los requisitos iniciales para el *Internet of Things* (IoT), las empresas que en la actualidad tienen soluciones completas en el mercado son: *Sigfox* con su protocolo del mismo nombre y *LoRa Alliance*, con su protocolo *LoRa* (Long Range) [7].

III-B. LoRa

La capa física *LoRa*, implementa modulación de espectro expandido tipo *Chirp*, patentada por *Semtech*, permite comunicaciones de largo alcance de bajo consumo de potencia y amplia cobertura, siendo muy resistente a interferencias por factores de ruido. Opera en las bandas *ISM* de: 868MHz (Europa) a +14dBm como máximo. En 915MHz (América del Norte y del Sur, Australia y Nueva Zelanda) a un máximo de 20dBm. En China puede operar entre los 470 - 510 MHz, a 14dBm como máximo. La carga útil de cada transmisión puede oscilar entre 2-255 bytes y la velocidad de datos puede alcanzar hasta 50 Kbps cuando se emplea la agregación de canales [8]. Cómo *LoRa* utiliza *Chirp Spread Spectrum* (CSS) como modulación, modula los mensajes enviados mediante una señal que varía continuamente en frecuencia lo que aumenta significativamente el rango de comunicación y la resistencia a interferencias [8].

III-B1. Formato de trama física: Aunque la modulación *LoRa* se puede usar para transmitir tramas arbitrarias, se específica y se implementa un formato de trama física en los transmisores y receptores *LoRa*. El ancho de banda (BW) y el factor de dispersión (SF) son constantes para una trama. El *SF* deberá ser el mismo para que un emisor dado pueda ser escuchado por determinado receptor. Diferentes *SF* son ortogonales entre ellos, una característica que da potencial a *LoRa*, ya que permite reutilizar un mismo canal para más nodos de manera simultánea variando el *SF*. Dado que hay 2^{SF} chirps en un símbolo, un símbolo puede codificar *SF* bits de información [8]. A mayor *SF*, menor velocidad de transferencia pero mayor inmunidad al ruido. Una trama *LoRa* inicia con un preámbulo. A su vez el preámbulo comienza con una secuencia de chirps constantes que cubren toda la banda de frecuencia. Los últimos dos chirps codifican la palabra de

sincronización. La palabra de sincronización esta formada por un byte que se usa para diferenciar las redes LoRa que usan las mismas bandas de frecuencia [8]. Un dispositivo configurado con una palabra de sincronización dada dejaría de escuchar una transmisión si la palabra de sincronización descodificada no coincide con su configuración. La palabra de sincronización es seguida por 2.25 chirps invertidos, para una duración de 2.25 símbolos. La duración total de este preámbulo se puede configurar entre 10.25 y 65,539.25 símbolos [8]. La estructura del preámbulo se puede ver en la Fig. 2.

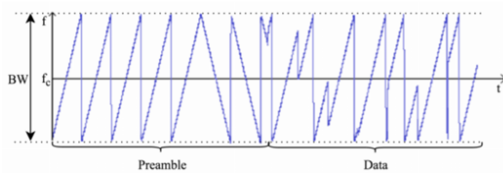


Figura 2. Variación de frecuencia a lo largo del tiempo de una señal de muestra emitida por un transmisor *LoRa*. Estructura de preámbulo de sincronización. f_c es la frecuencia central del canal, y BW es el ancho de banda [8].

IV. METODOLOGÍA

El desarrollo del *gateway* LoRa/WiFi requiere que los dispositivos de los usuarios cuenten con capacidad para conectarse a redes WiFi convencionales, dado que el *gateway* permitirá que dispositivos móviles se interconecten con el mediante WiFi, y retransmitirá los mensajes de los usuarios vía LoRa.

Dichas características son:

- Compatibilidad con WiFi (2.4GHz) por parte de los dispositivos móviles
- Portabilidad del *gateway* para cobertura adaptable
- Autonomía en el consumo de energía
- Escalabilidad tecnológica

IV-A. Placa de desarrollo a utilizar

Se realizó un análisis acerca de distintas tecnologías factibles para desarrollar este proyecto, entre otras, *Sigfox* ya que también es un protocolo LPWAN. Sin embargo, se decidió por *LoRa*, dado que esta es una tecnología libre de usar y no requiere de un servicio en la nube, aunque puede tenerlo si se utiliza *LoRaWAN*. Dado que se desea usar un protocolo sin Internet se determinó utilizar *LoRa* y el chip de desarrollo *LoPy4* fabricado por la empresa *Pycom*, esto por sus ventajas en el procesamiento y que cuenta con los módulos de comunicación requeridos ya integrados en dicha placa: los módulos WiFi, LoRa y el propio microcontrolador ESP32, lo cual evita el uso de componentes extras que compliquen el diseño de circuito y el tamaño final del dispositivo. La Fig. 3 muestra los componentes de *LoPy4*.

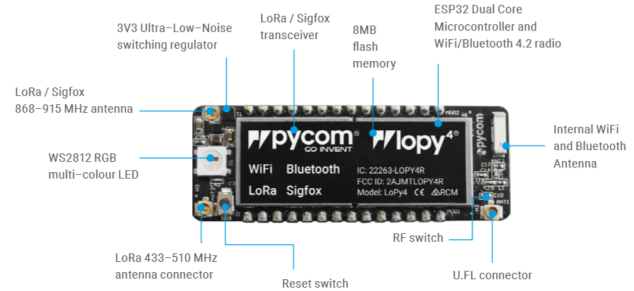


Figura 3. Componentes de *LoPy4* [9]

IV-B. Construcción del canal de comunicación LoRa/WiFi

En la Fig. 4 se pueden observar los componentes que conforman el enlace de comunicación WiFi/LoRa, así como una representación gráfica de la interacción entre ellos.

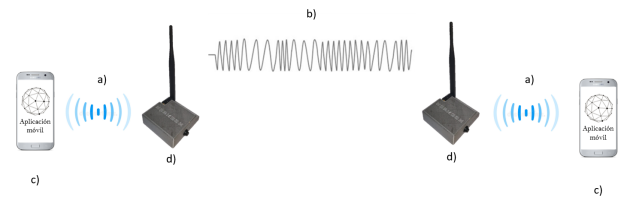


Figura 4. Diagrama de operación; donde (a) representa la comunicación WiFi entre dispositivos móviles y *gateway* Hermes; (b) el protocolo LoRa entre dos *LoPy4*; (c) el dispositivo móvil y (d) el dispositivo *LoPy4*.

El microcontrolador se encarga de recibir los mensajes emitidos del dispositivo móvil y transmitirlos por protocolo LoRa hasta su par, que hace la tarea inversa. El mensaje sale en forma de onda del emisor, codificando con CSS la información transmitida. Además de contener una llave de seguridad, para que el mensaje solo pueda ser abierto por el dispositivo que espera su recepción. La comunicación entre dispositivos, dependiendo de condiciones ambientales puede llegar hasta 15km en línea de vista y de hasta 500m dentro de edificios.

Para efectuar la comunicación entre los usuarios se diseñó una aplicación móvil, esta se encarga de ser la interfaz entre el dispositivo móvil y el punto de acceso inalámbrico LoRa como se observa en la Fig. 5. La aplicación permite escribir y mostrar mensajes de la conversación. El *gateway* se conecta a un dispositivo móvil, siempre que se encuentren en cobertura de WiFi.

IV-C. Construcción del prototipo HERMESH

Se construyó el circuito electrónico, montando los módulos *LoPy4* sobre una placa fenólica perforada como se muestra en la Fig. 6, con fuente de alimentación de 4.2V. Dicho procedimiento fue duplicado para realizar ensayos de comunicación en un enlace punto-a-punto.

Considerando los aspectos de funcionalidad sin olvidar la portabilidad y estética, se desarrolló un diseño a través del

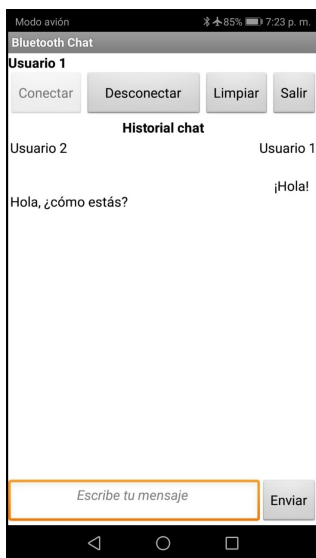


Figura 5. Interfaz de aplicación móvil.

Con el prototipo impreso en 3D se realiza la interconexión de los componentes electrónicos, posteriormente se montan en la carcasa. La Fig. 7 muestra el diseño del dispositivo terminado al cual se le asignó el nombre de "HERMESH".

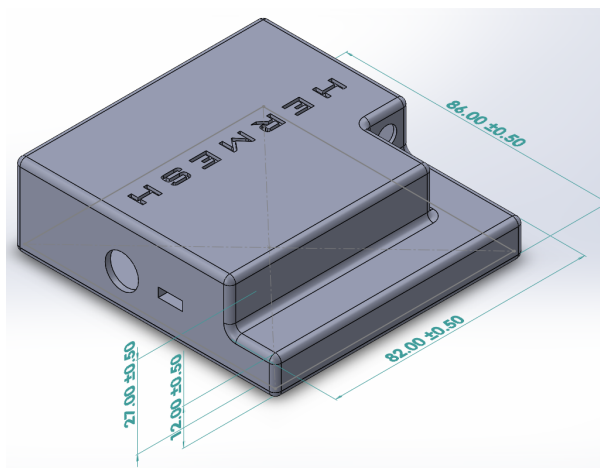


Figura 7. Diseño de carcasa en SolidWorks en vista isométrica.

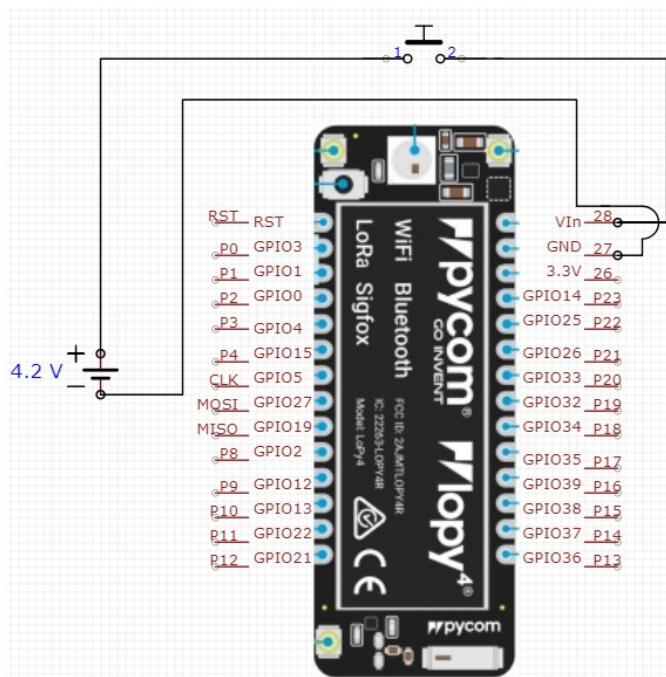


Figura 6. Diagrama eléctrico del dispositivo.

programa de dibujo asistido por computadora, *SolidWorks*. El diseño (Fig. 7) cuenta con características tales como:

- Botón de acción Encendido/Apagado.
- Puerto micro USB para recarga de batería.
- Antena externa.
- Pestaña para guardar antena externa.
- Dimensiones de 86 X 82.5X 27.5 mm

V. PRUEBAS Y RESULTADOS

Para poner a prueba la distancia que ofrece *LoRa* como plataforma de comunicación, se realizaron tres pruebas en distintos escenarios con el fin de analizar la efectividad del enlace y su comportamiento enviando y recibiendo mensajes. A continuación, se muestran los resultados de las pruebas realizadas.

La configuración de los equipos fue la siguiente:

- Modo *LoRa*.
- Frecuencia de trabajo 915MHz.
- Factor de dispersión SF = 12.
- Potencia de transmisión 20dBm.

V-A. Prueba 1

Para determinar el tiempo de vida de la batería del dispositivo se realizaron cálculos del consumo eléctrico midiendo los valores de corriente en sus tres estados de trabajo: espera, recepción y transmisión; donde se obtuvieron los valores mostrados en la Tabla I.

La corriente promedio demandada por el dispositivo se calculó mediante 500 mensajes recibidos y 500 transmitidos en un tiempo de uso de 14 horas. El tiempo que tarda en recibir y transmitir mensajes es en promedio de 1 segundo en cada estado. El porcentaje de tiempo se muestra en la Tabla I.

La corriente promedio se calcula con la ecuación (1).

$$I_P = I_E * t_E + I_R * t_R + I_T * t_T \quad (1)$$

donde:

- I_P = Corriente promedio.
- I_E = Corriente en modo espera.

Tabla I
DEMANDAS DE CORRIENTE Y DURACIÓN PROMEDIO DE LOS ESTADOS DE TRABAJO.

Estado de trabajo	Corriente	Tiempo
Espera	60 mA	13.7222 h (98.01 %)
Receptor	140 mA	0.1388 h (0.99 %)
Transmisor	220 mA	0.1388 h (0.99 %)

- t_E = Tiempo promedio en modo espera.
- I_R = Corriente en modo receptor.
- t_R = Tiempo promedio en modo receptor.
- I_T = Corriente en modo transmisor.
- t_T = Tiempo promedio en modo transmisor.

Se aplica la fórmula (1) con los valores de la Tabla I.

$$I_P = 62.4mA$$

Para obtener el tiempo de vida de la batería se tiene la fórmula (2).

$$tiempo\ de\ vida = \frac{C(mAh)}{I(mA)} \tag{2}$$

donde:

- C = Capacidad de la batería.
- I = Corriente demandada.

Con una batería 18650 de 3000mAh y la corriente promedio, sustituimos en la fórmula (2).

$$tiempo\ de\ vida = \frac{3000mAh}{62.4mA} = 48.07h$$

El tiempo de vida esperado para la batería, en el caso descrito en esta prueba será de 48 horas y 4.2 minutos.

V-B. Prueba 2

Con el fin de analizar el comportamiento del enlace en zona urbana se realizó una prueba de alcance en un área altamente transitada de la ciudad de La Paz Baja California Sur, entre los puntos A y B como se muestra en la Fig. 8, con direcciones de Boulevard 5 de febrero e/ Boulevard Forjadores y calle Aquiles Serdán e/ calle Miguel L. de Legaspy, respectivamente.

La distancia obtenida entre los puntos fue de 1.4 km, destacando que no hay visibilidad entre estos. Para estimar las pérdidas de comunicación debido a los obstáculos se enviaron 100 mensajes desde cada punto hacia el otro, haciendo registro de los mensajes entregados y de aquellos que llegaban con algún error.

Tabla II
RESULTADOS DE LA PRUEBA 2.

Punto de recepción	Punto A	Punto B
Mensajes entregados	100 %	100 %
Mensajes sin error	85 %	93 %
Mensajes con error	15 %	7 %

Debe destacarse que todos los mensajes enviados fueron entregados a su destino. En esta prueba se presentaron casos,



Figura 8. Imagen mostrando la trayectoria de la prueba 2[Google Maps, 2019].

en donde una cadena de texto se particionaba y se empalmaba con la cadena de texto anterior o con la siguiente. A estos mensajes se les denominó mensajes con error. La Fig. 9 representa una conversación donde se aprecia la manera en la que los mensajes fueron enviados y como fueron recibidos.

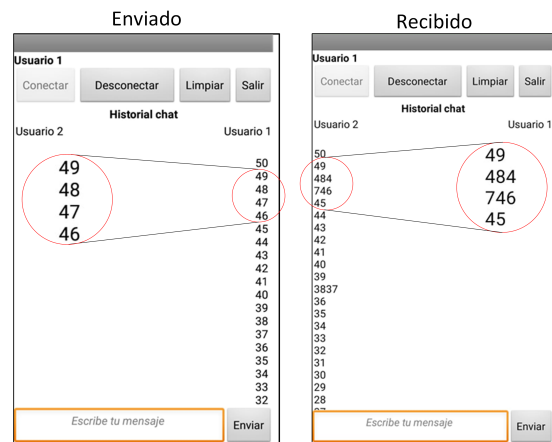


Figura 9. Imagen que muestra mensajes con error en una conversación.

V-C. Prueba 3

La tercera prueba fue desarrollada en la bahía de La Paz, Baja California Sur, aprovechando así la costa de la ciudad como se muestra en la Fig. 10 en donde los puntos C y D son playa frente a la preparatoria CETMAR y la playa El Coromuel, respectivamente. Aquí buscamos un alcance mayor ubicando los puntos en línea de vista sin ningún tipo de obstáculo.

La distancia total obtenida es de 7.62 km entre los puntos C y D. De nuevo se enviaron, de igual manera, 100 mensajes desde cada punto hacia el otro. Los resultados se muestran en la Tabla III.

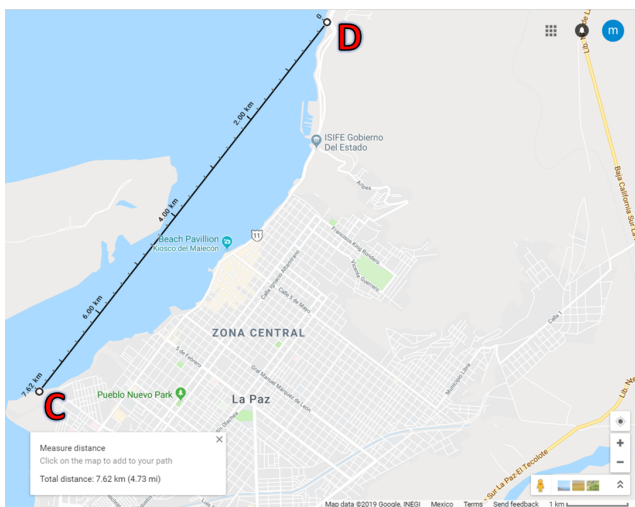


Figura 10. Imagen mostrando la trayectoria de la prueba 3 [Google Maps, 2019].

Tabla III
RESULTADOS DE LA PRUEBA 3.

Punto de recepción	Punto A	Punto B
Mensajes entregados	85 %	95 %
Mensajes sin error	80 %	88 %
Mensajes con error	5 %	7 %

Esta prueba mostró comunicación exitosa entre los puntos C y D, donde gran parte de los mensajes enviados fueron entregados, con porcentajes bajos de mensajes con error.

V-D. Prueba 4

En la cuarta prueba de distancia se emitieron 100 mensajes desde el punto A localizado sobre techo del Laboratorio de Electrónica del Tecnológico Nacional de México Campus La Paz hasta el punto B ubicado en calle México esquina calle Colima. La distancia total es de 1.71 km en línea recta como se muestra en la fig. 11

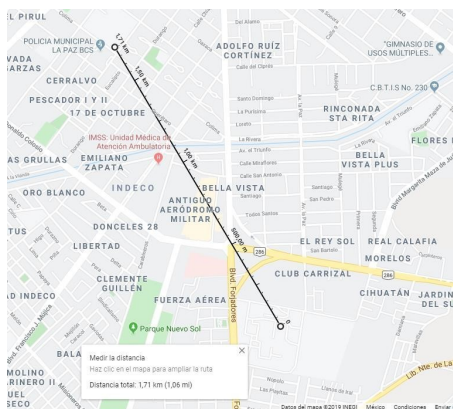


Figura 11. Imagen mostrando la trayectoria de la prueba 4 [Google Maps, 2019].

La transmisión de cada mensaje fue clasificado según su condición de entrega en el punto 2 en las categorías: Entregado o No entregado. Los resultados se muestran en la Tabla IV.

Tabla IV
RESULTADOS DE LA PRUEBA 4.

Categoría	Cantidad
Mensajes Entregados	91
Mensajes No Entregados	9

Basados en los resultados obtenemos un 91 % de éxito de entrega a 1.71 km distancia sin línea de vista. Los aspectos que consideramos que pudieron haber dificultado o impedido la correcta entrega de los mensajes son principalmente: Otras frecuencias de transmisión que chocaran con nuestros paquetes y los edificios que se encontraban en la trayectoria que a pesar de contar con 2 pisos de altura en el punto A, definitivamente no pudimos evitar.

V-E. Prueba 5

Esta quinta prueba consistió en volver a hacer la misma cantidad de transmisiones pero a una mayor distancia, la recta que se consideró apropiada parte del punto A en el Muelle Fiscal del Malecón de La Paz, Baja California Sur hasta el punto B en el muelle de la playa El Coromuel con el fin de obtener menos interferencia tanto física como de otras frecuencias y evaluar el porcentaje de eficiencia de la transmisión de mensajes. La distancia total de la prueba fue de 4.25 km en línea recta a través del mar como se puede apreciar en la Fig. 12

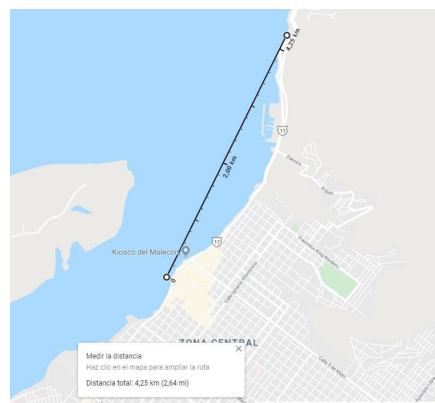


Figura 12. Imagen mostrando la trayectoria de la prueba 4 [Google Maps, 2019].

Los resultados que se obtuvieron tuvieron una mejora a pesar de incrementar la distancia entre emisor y receptor, respecto a la prueba 4, como se muestran en la Tabla V.

La conclusión de estos resultados es que en una distancia de 4.25km se obtiene 99 % de éxito de entrega en línea de vista. Es decir, *LoRa* ve reducida su eficiencia de recepción con obstáculos físicos.

Tabla V
RESULTADOS DE LA PRUEBA 4.

Categoría	Cantidad
Mensajes Entregados	99
Mensajes No Entregados	1

VI. CONCLUSIONES

De acuerdo a los resultados obtenidos en las pruebas realizadas, el *gateway LoRa/WiFi*: HERMESH, es capaz de permitir la interconexión de dispositivos móviles en una red de largo alcance sin hacer uso de la red celular o de Internet.

Dado que estas pruebas se desarrollaron en diferentes escenarios, comunicación con y sin obstáculos, donde se estableció la red de manera exitosa con rangos de aproximadamente 1.4 km en zona urbana y 7.62 km en línea de vista. Analizando estos valores se llega a la conclusión de que el enlace es susceptible a interferencias por objetos sólidos como lo son casas habitaciones, edificios, arboles, arbustos y vehículos en tránsito, entre otros.

Se valida una eficiencia de entrega de mensajes superior al 90%, también se hace énfasis en la portabilidad de los nodos hace que la topología de red sea flexible, que los mensajes se puedan entregar por diferentes rutas. Uno de los usos alternativos que puede tener esta red es como medio de comunicación de emergencia ante desastres naturales como los huracanes que se presentan cada año en el estado de Baja California Sur, dado que no depende de infraestructura fija la cual es susceptible a daños por dichos desastres.

Para dar seguimiento al proyecto se proponen las siguientes recomendaciones:

- Diseñar y utilizar repetidores Hermesh para ampliar la cobertura de la red de mensajería.
- Realizar las mejoras que permitan a la red enviar y recibir paquetes de datos de tipo multimedia.
- La red de comunicación para mensajería puede usarse para interconectar sensores y actuadores a largas distancias sin Internet, se recomienda hacer las pruebas.
- Desarrollar mejorar a la aplicación móvil para controlar dispositivos del tipo *IoT* para control y automatización de procesos.
- Realizar mejoras en la carcasa para que el estado de funcionamiento sea visible a través de LEDs.

REFERENCIAS

- [1] INEGI "Encuesta Nacional sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares ENDUTIH", 2018. INEGI, 2018
- [2] CONEVAL "Diez años de medición de pobreza multidimensional en México: avances y desafíos en política social." 2018 CONEVAL: 2018.
- [3] Organización de las Naciones Unidas para la Alimentación y la Agricultura, 2018. "México rural del Siglo XXI" FAO: 2018
- [4] INEGI "Encuesta Nacional de la Dinámica Demográfica ENADID", 2018. INEGI, 2018
- [5] Salazar, J., 2017. "Redes inalámbricas." Republica checa: TechPedia.
- [6] Fernández Martínez, R., Ordieres Meré, J., Martínez de Pisón Ascacibar, F., González Marcos, A., Alba Elías, F., Lostado Lorza, R. and Pernía Espinoza "Redes Inalámbricas de sensores: Teoría y aplicación práctica", 2009. Universidad de La Rioja, 2009

- [7] Monfort, I. O., 2017. "Estudio de la arquitectura y el nivel de desarrollo de la red LoRaWAN y de los dispositivos LoRa." Universidad Oberta de Catalunya.
- [8] Augustin, A.; Yi, J.; Clausen, T.; Townsley, W.M. "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things.", 2016. Sensors 2016, 16, 1466
- [9] "Pycom", 2017 LoPy4, Datasheet

Emulación del movimiento del cuello humano con un robot de 3 grados de libertad

Eduardo Pastor Torres
Facultad de Ciencias de la Electrónica
Benemérita Universidad Autónoma de
Puebla
Puebla, Pue., México
eduardo.pastor@alumno.buap.mx

María Aurora Diozcora Vargas Treviño
Facultad de Ciencias de la Electrónica
Benemérita Universidad Autónoma de
Puebla
Puebla, Pue., México
auroravargast@hotmail.com

Sergio Vergara Limón
Facultad de Ciencias de la Electrónica
Benemérita Universidad Autónoma de
Puebla
Puebla, Pue., México
vergara2@hotmail.com

Felix Francisco Gonzalez Canales
Facultad de Ciencias de la Electrónica
Benemérita Universidad Autónoma de
Puebla
Puebla, Pue., México
felixfcoglz@gmail.com

David Eduardo Pinto Avendaño
Facultad de Cs. de la Computación
Benemérita Universidad Autónoma de
Puebla
Puebla, Pue., México
davideduardopinto@gmail.com

Resumen — *Los sistemas autónomos son elementos que cada día cubren mayores áreas de trabajo, se están volviendo tan sofisticados que estos comienzan a interactuar con las personas y desempeñan tareas similares a ellas. En particular, los robots humanoides se han vuelto un caso de estudio y dada su complejidad, es necesario analizar de forma paralela los elementos que los componen, ya sean mecanismos, inteligencia artificial, reconocimiento facial, entre otros. En este trabajo se muestra el diseño de un robot esférico de 3 grados de libertad, el cual emula el movimiento de un cuello humano. Por consiguiente, se analiza la dinámica y se propone un esquema de control proporcional derivativo. Además, se analiza la respuesta de cada articulación en el dominio del tiempo y se concluye con las acciones de control como trabajo a futuro.*

Palabras clave—*Robot humanoide; Cuello; 3 grados de libertad; control de posición.*

Abstract — *Autonomous systems are elements that are covering several working areas every day, these are being so sophisticated that they are reaching interaction with people and they are developing similar activities like they do. Human robots are a study case and because of its complex development, it's necessary analyze its components in a parallel way such that there are different areas of study, from mechanical development, artificial intelligence, facial recognition, and others. In this job we show a 3 degree of freedom (DOF) spherical robot which emulates the movement of a human neck, then, we study the dynamic model and finally we propose a control scheme, its average is studied in time domain, thus, we conclude the job purposing new control actions for future job.*

Keywords—*Humanoid robot, neck, 3 degree of freedom, position control.*

I. INTRODUCCIÓN

En los últimos años la sociedad se ha interesado enormemente en el desarrollo de robots humanoides, esto debido a que la necesidad de otorgar apoyo a los seres humano siempre ha existido. Pero encontrar quien pueda otorgar dicha ayuda se dificulta. Sin embargo, los robots pueden ser una alternativa

capaz de desempeñar dicha tarea. Por ello, una barrera a romper es lograr una comunicación natural entre los seres humanos y las máquinas. Se sabe que la comunicación entre los seres humanos no depende únicamente de la forma oral, 60% de la comunicación se logra implementando gestos y expresiones faciales [1], evidentemente hoy día es crucial determinar un buen aspecto físico de los robots humanoides y éste debe ser capaz de desarrollar habilidades como la expresión de emociones o gesticulación muscular.

En la actualidad existen diversas empresas que están trabajando en el desarrollo de robots humanoides, entre ellos se encuentra la empresa Boston Dynamics, uno de sus últimos desarrollos es “Atlas”, un robot que es capaz de saltar objetos y es capaz de ascender a través de escalones, desarrollar actividades de parkour, entre otros [2]. El instituto tecnológico de Italia (ITT), en colaboración de diversas instituciones europeas, ha desarrollado ICUB, un robot con dimensiones similares a un niño de 3 años que es capaz de realizar movimientos de caminata, sujetar objetos, entre otros [3]. Otros sistemas interesantes dan pauta a los desarrolladores de Hanson Robotics y su robot humanoide Han: un robot diseñado con piel de goma, capaz de responder a gestos, expresiones mediante un sistema mecánico y de reconocimiento facial [4]. Cumple tareas simples como mantener un contacto visual y establecer conversación con otras personas a través de reconocimiento de voz.

Para lograr el control de la cabeza de un robot humanoide se tiene como pieza clave la interacción con el cuello [5], éste último proporciona dinámica a la cabeza y, consecuentemente, facilita la comunicación con los seres humanos.

El laboratorio de Ingeniería del Lenguaje y del Conocimiento (LKE por sus siglas en inglés), ha adquirido recientemente a

“Arthur”, un robot humanoide capaz de realizar expresiones faciales con ayuda de 27 servomotores (figura 1), sin embargo, la base de la cabeza, la cual emula el comportamiento del cuello, falla y genera sobrecalentamiento en los motores, este tipo de problemas provoca que los movimientos del cuello queden en determinado momento totalmente desactivados y, por lo tanto, el sistema global deja de tener el desempeño deseado. Dada esta problemática, en este trabajo se propone el diseño mecánico de un robot esférico de 3 grados de libertad el cual emula al cuello humano. Además, se desarrolla su sistema de control a través de una FPGA que emula un procesador y retroalimenta el sistema con motores de corriente directa con acoplamiento de codificadores de posición (encoders).



Figura 1: Arthur, el robot humanoide del posgrado en ingeniería del lenguaje y conocimiento.

II. MODELO DINÁMICO

Actualmente, una de las formas para desarrollar los modelos dinámicos de un robot es utilizar el formalismo Lagrangiano. En este formalismo, las ecuaciones de movimiento están dadas por las ecuaciones de Euler-Lagrange, cuya forma es:

$$\tau_i = \frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}_i} \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) \right] - \frac{\partial}{\partial q_i} \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}), \quad (1)$$

donde $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ es la función Lagrangiana, la cual se deduce a partir de la cinemática del sistema, misma que es obtenida por la convención Denavit-Hartenberg (D-H) [6], mientras que τ es el par aplicado en el sistema mecánico.

En robótica, las ecuaciones de movimiento de Euler-Lagrange, dadas por la expresión (1) se escriben como:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}), \quad (2)$$

donde $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ representa a la matriz de inercia, $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ como la matriz de fuerza centrípeta y coriolis, y $g(\mathbf{q}) \in \mathbb{R}^{n \times 1}$ es un vector de fuerzas gravitacionales; las dimensiones de las matrices y vector se definen por el robot de n grados de libertad. Finalmente, el modelo dinámico depende de sus posiciones \mathbf{q} , velocidades $\dot{\mathbf{q}}$ y aceleraciones $\ddot{\mathbf{q}}$ [7].

El robot implementado para emular el cuello se ilustra en la figura 2, el sistema es un robot esférico de 3 grados de libertad, el primer grado de libertad permite emular el movimiento de la cabeza hacia adelante y hacia atrás (denominado flexo-extensión en el área de la salud, flexo como movimiento hacia adelante y extensión como movimiento hacia atrás), el segundo grado de libertad otorga el movimiento hacia los hombros (bilateral) y el tercer grado de libertad da genera el giro hacia la izquierda y derecha (rotacional), movimientos principales del cuello humano [8].

A. Cinemática del robot

Los parámetros resultantes del análisis del robot de tres grados de libertad son obtenidos con la convención Denavit-Hartenberg [6], la cual es basada en las articulaciones y eslabones involucrados, con ello, es posible deducir la cinemática del robot. Para nuestro caso de estudio, el vector $[x_0 \ y_0 \ z_0]$ representa la posición casa, por lo que las posiciones cartesianas de nuestro robot en función del ángulo \mathbf{q} tienen la forma [9]:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} l_1 c_1 \\ l_1 s_1 \\ 0 \end{bmatrix}, \quad (3)$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} c_1(l_1 + \beta_1 c_2 + l_2 c_2) \\ s_1(l_1 + \beta_1 c_2 + l_2 c_2) \\ s_1(\beta_1 + l_2) \end{bmatrix}, \quad (4)$$

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} l_3(c_3 s_3 + c_1 s_2 s_3) + \beta_1 c_1 c_2 + \beta_2 c_1 c_2 \\ \beta_1 c_2 s_1 - l_3(c_1 c_3 - s_1 s_2 s_3) + \beta_2 c_2 s_1 \\ \beta_1 s_2 + \beta_2 s_2 - l_3 s_2 c_3 \end{bmatrix}, \quad (5)$$

donde $s_i = \text{sen}(q_i)$ y $c_i = \text{cos}(q_i)$, con $i=1,2,3$; l_i y β_i son valores longitudinales utilizados para el análisis de la cinemática, dichos valores se especifican en el apéndice.

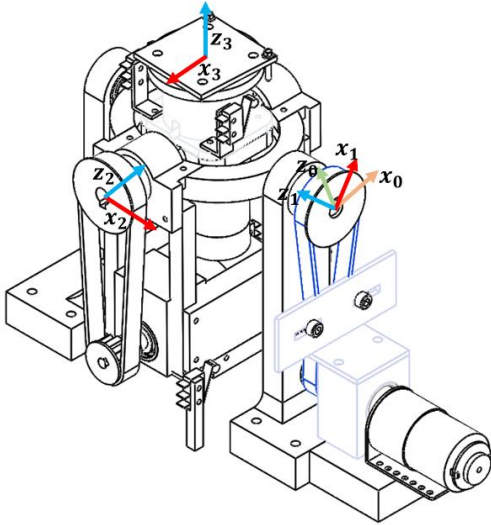


Figura 2: Robot esférico de 3 grados de libertad.

B. Dinámica del Robot

La dinámica del robot de tres grados de libertad mostrada en la figura 2 se obtiene a partir de la expresión dada en (2). Respectivamente, para este cuello robótico la matriz de inercia y de coriolis están dada por:

$$M(\mathbf{q}) = \begin{bmatrix} m_{1,1}(q_2, q_3) & m_{1,2}(q_2, q_3) & m_{1,3}(q_2, q_3) \\ m_{1,2}(q_2, q_3) & m_{2,2}(q_2, q_3) & m_{2,3}(q_3) \\ m_{1,3}(q_2, q_3) & m_{2,3}(q_3) & m_{3,3}(q_2, q_3) \end{bmatrix} \quad (6)$$

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} c_{1,1}(\mathbf{q}, \dot{\mathbf{q}}) & c_{1,2}(\mathbf{q}, \dot{\mathbf{q}}) & c_{1,3}(\mathbf{q}, \dot{\mathbf{q}}) \\ c_{2,1}(\mathbf{q}, \dot{\mathbf{q}}) & c_{2,2}(\mathbf{q}, \dot{\mathbf{q}}) & c_{2,3}(\mathbf{q}, \dot{\mathbf{q}}) \\ c_{3,1}(\mathbf{q}, \dot{\mathbf{q}}) & c_{3,2}(\mathbf{q}, \dot{\mathbf{q}}) & c_{3,3}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} \quad (7)$$

donde $M(\mathbf{q}), C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathfrak{R}^{3 \times 3}$, las formas explícitas de los elementos de las matrices están dadas en el apéndice A. El vector $\mathbf{g}(\mathbf{q}) \in \mathfrak{R}^{3 \times 1}$ representa las fuerzas o pares gravitacionales, el cual, en este caso tiene la forma:

$$\mathbf{g}(\mathbf{q}) = g \begin{bmatrix} g_1(\mathbf{q}) \\ g_2(\mathbf{q}) \\ g_3(\mathbf{q}) \end{bmatrix} \quad (8)$$

donde:

$$\begin{aligned} g_1(\mathbf{q}) &= l_3 m_3 c_1 c_3 + (l_{c1} m_1 - l_1) s_1 \\ &\quad + [(-b_1 - l_2) m_3 + (l_{c2}) m_2] c_2 s_1 \\ &\quad - l_3 m_3 s_1 \\ g_2(\mathbf{q}) &= [l_{c2} m_2 - b_1 m_3 - l_2 m_3] c_1 s_2 \\ &\quad + l_3 m_3 s_3 c_1 c_2 \\ g_3(\mathbf{q}) &= l_3 m_3 s_2 c_1 c_3 - l_3 m_3 s_1 s_3 \end{aligned} \quad (9)$$

III. SISTEMA DE CONTROL

A. Hardware implementado.

El Sistema de control del robot esférico de tres grados de libertad es ejecutado con una FPGA Altera Cyclone V, la figura 3 describe los recursos disponibles, entre ellos se encuentra la emulación de un microprocesador de 32 bits, dicho elemento es capaz de adquirir datos de un decodificador de cuadratura; actuar motores de transmisión directa a través de moduladores de ancho de pulso (PWM); comunicación con una computadora a través de wifi y usar ésta para transferir el algoritmo de control, ganancias, respuesta del sistema, entre otros.

El microprocesador implementado dentro de la FPGA ha sido desarrollado en el laboratorio de robótica y control de la Facultad de Ciencias de la Electrónica-BUAP [10]. Este microcontrolador es utilizado para ejecutar el controlador digital del sistema gracias a su capacidad de desarrollar operaciones tipo flotante, ejecutar diversas operaciones, incluyendo funciones trigonométricas, lectura y escritura en la memoria RAM y memoria ROM, poseer puertos de entrada y salida de propósito general. Además, de generar, adquirir y transferir datos a módulos de propósito específico desarrollados dentro de la FPGA. Dado que tenemos un sistema robusto, es posible lograr control en tiempo real.

Una de las señales de salidas otorgadas por el sistema de control son los pulsos modulados que actuarán en los motores, estos requieren una etapa de potencia para amplificarlas, esto se logra con la implementación de un puente H. Los motores dan respuesta al cambio de posición de cada articulación y es posible medir la posición resultante gracias a los codificadores de posición de cuadratura que tienen acoplados. Los codificadores retroalimentan al microprocesador a través de los módulos de decodificación, así es posible cerrar el lazo de control.

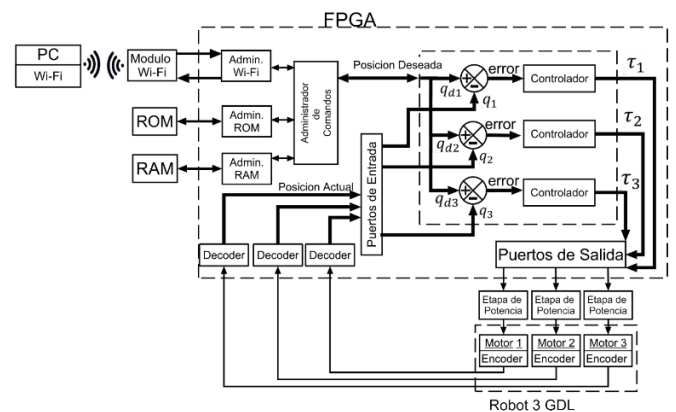


Figura 3: Descripción a bloques del hardware implementado como sistema de control.

B. Lazo de control implementado

Para lograr el control del robot esférico se implementa un controlador proporcional derivativo con compensación gravitacional con la forma de:

$$\boldsymbol{\tau} = k_p \tilde{\mathbf{q}} - k_v \dot{\mathbf{q}} + k_{gv} \mathbf{g}(\mathbf{q}), \quad (10)$$

donde

- \mathbf{q} es el vector de posición actual ($^\circ$).
- $\tilde{\mathbf{q}}$ es el vector de error de posición.
- $\dot{\mathbf{q}}$ es el vector de velocidad ($^\circ/s$).
- $\boldsymbol{\tau}$ es el vector de par aplicado (N m).
- k_p es la matriz de ganancia del controlador proporcional.
- k_v es la matriz de ganancia del controlador derivativo.
- k_{gv} es la matriz de “ganancia” utilizada en el controlador para la compensación gravitacional obtenida de la masa, campo gravitacional y la magnitud del centro de masa

El lazo de control es descrito en la figura 4, donde \mathbf{q}_d es el vector de la posición deseada ($^\circ$). La parte proporcional depende del error de posición ($\tilde{\mathbf{q}}$), la parte derivativa en función de la velocidad adquirida ($\dot{\mathbf{q}}$), además de la compensación gravitacional en función de la posición (\mathbf{q}). Como se mencionó, el sistema se rige de acuerdo con (10), la posición deseada es indicada a través de la interfaz de usuario en la computadora.

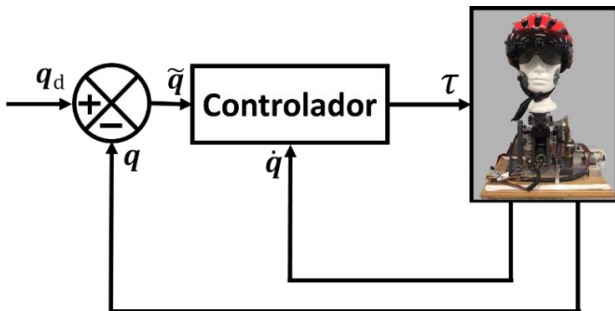


Figura 4: Diagrama a bloques del control a implementar.

IV. RESULTADOS

La tabla 1 describe los rangos en los que funciona el emulador del cuello del robot humanoide, el primer grado de libertad es el encargado de emular el movimiento flexo-extensión, la segunda articulación permite el movimiento de inclinación lateral, mientras que el tercer grado de libertad se encarga de emular el movimiento rotacional. La implementación del sistema se logró bajo el diseño propuesto, el cual abarca el modelo dinámico. Debido a que no se tiene acceso a la cabeza real de Arthur, se utiliza una cabeza de unicel con incrustaciones de acero en puntos aleatorios con el fin de

aproximarnos a las características físicas de la cabeza del robot Arthur.

Tipo de movimiento	Rango de movimiento ($^\circ$)
Flexo-extensión	$[-35^\circ, 25^\circ]$
Inclinación lateral (por lado)	15°
Rotacional	60°

Tabla 1: Rangos de operación de robot esférico de 3 grados de libertad.

Los pares calculados se obtienen asumiendo que el centro de masa se encuentra en el centro de la cabeza, es decir, se aplicó el modelo de partícula de la mecánica clásica. En acuerdo con los pares calculados en cada motor y los pares estimados, las ganancias estimadas del controlador preliminar son se muestran en la tabla 2. Estas ganancias fungen como ganancias proporcionales k_p , las ganancias derivativas se toman como $k_v = 2\sqrt{k_p}$, según sugiere la teoría de controladores proporcionales-derivativos [11]. Las ganancias previamente mencionadas son sintonizadas después de una serie de pruebas hasta conseguir una respuesta del sistema favorable.

Las ganancias del controlador serán limitadas con la función tangente hiperbólica y la ley de control implementada en el procesador tiene la forma:

$$\boldsymbol{\tau} = k_p \tanh(k_s \tilde{\mathbf{q}}) - k_v \tanh(k_s \dot{\mathbf{q}}) - k_{gv} \mathbf{g}(\mathbf{q}). \quad (11)$$

donde $k_{gv} = mgl$ y es la ganancia de compensación gravitacional.

Después de ejecutar el lazo de control de control repetidas veces, se logró sintonizar el robot con un error de posición mínimo en el estado estacionario, y los parámetros de ganancia se muestran en la tabla 3. Las ganancias k_p y k_{gv} son dadas en un rango de 0 a 1000, lo cual significa que el par otorgado por cada uno de los motores es menor al 50% del total que pueden suministrar.

Número de motor	Ganancia estimada
M_1	59.75%
M_2	35.31%
M_3	50.73%

Tabla 2: Relación de ganancias necesarias para el robot de 3 grados de libertad.

Ganancias propuestas de controlador			
Parámetro	q_1	q_2	q_3
k_p	400	353	360
k_s	0.4	0.35	0.3
k_v	10	8	6
k_{gv}	320	250	200

Tabla 3: Ganancias implementadas en robot rotacional.

Se ha implementado un procesador emulado en una FPGA, para los diversos procesos que debe ejecutar el procesador, es necesaria la transferencia del código a ejecutarse en lenguaje máquina. Para esto, existen diversas formas de hacerlo, desde el uso de sistemas de comunicación con protocolo serial [12] o protocolo LAN [13], ambos son comunicación vía cable. En cambio, nuestra propuesta de transferencia de datos al sistema embebido es manejando la comunicación con protocolo wifi [14]. Dado que es una comunicación inalámbrica, esta comunicación nos da la ventaja de operar libre de cables, además de otorgar un aislamiento físico entre el equipo utilizado como interfaz de usuario y el robot. Este protocolo de comunicación se desarrolló utilizando el software LABVIEW.

En la figura 5 se muestra la interfaz de usuario (hecha en LABVIEW) que permite comunicarse con el robot, dicha interfaz es utilizada para transferir en lenguaje máquina el algoritmo de control. El algoritmo de control también permite la lectura y escritura de direcciones de memoria RAM, transferencia de datos de la respuesta del controlador, entre otros. La interfaz está conformada por un botón denominado “descargar”, el cual permite la transferencia del lenguaje máquina previamente descrito; un indicador de grabado, es decir, cómo se comporta la transferencia de información de la computadora al robot; dos ventanas de depuración de transferencia de datos y una ventana de diálogo utilizada para indicar posibles errores durante la ejecución del software.

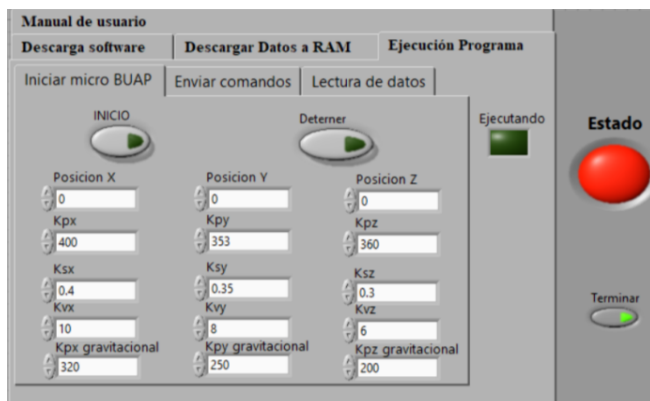


Figura 5: Interfaz de usuario, transferencia de software base.

La figura 6 y figura 7 representan el resultado del movimiento del primer grado de libertad, la posición inicial $q_i = 0^\circ$ y en el caso del movimiento flexo se proporciona la posición final $q_d = 8.5^\circ$, el cambio de posición inicia en $t = 6.9\text{ s}$, alcanzando su posición estacionaria en $t = 8.4\text{ s}$. En la gráfica se observa un sobretiro de $q_{st} = 9.84^\circ$, mientras que su posición en el estado estacionario es de $q_{est} = 8.46^\circ$. Para el movimiento de extensión: $q_d = -8.5^\circ$, cambio de posición inicia en $t = 2\text{ s}$, la posición estacionaria se alcanza en $t = 2.22\text{ s}$, con sobretiro de $q_{st} = -9.38^\circ$, y su posición en el estado estacionario es de $q_{est} = -8.29^\circ$.

Para el segundo grado de libertad tenemos la respuesta cuando su giro es positivo (emulando el movimiento de acercar la

cabeza al hombro derecho) en la figura 8, se ha definido como posición final $q_d = 8.5^\circ$, en $t_i = 2.9\text{ s}$ inicia la acción de control y logramos su posición estacionaria $q_{est} = 8.4^\circ$ cuando $t = 3.7\text{ s}$, el sobretiro alcanzado es $q_{st} = 10.81^\circ$. La respuesta del giro negativo (emulando acercamiento de la cabeza al hombro izquierdo) se aprecia en la figura 10, tenemos como posición deseada $q_d = -10^\circ$, su acción de control inicia en $t = 0.95\text{ s}$ y su posición estacionaria es alcanzada cuando $t = 1.9\text{ s}$ con un sobretiro de $q_{st} = -11.43^\circ$, su posición en el estado estacionario es de $q_{est} = -9.7^\circ$.

Finalmente hacemos una prueba en el tercer grado de libertad, el cual se encarga del movimiento rotacional, este permite emular el movimiento de giro de la cabeza hacia la derecha o hacia la izquierda. La figura 10 describe la trayectoria que sigue la cabeza es en sentido horario, se pretende una posición deseada $q_d = 24^\circ$, su acción de control comienza en $t_i = 2\text{ s}$, se alcanza un sobretiro de $q_{st} = 27.3^\circ$ y el dispositivo logra su estabilidad en $t_f = 3.4\text{ s}$ con posición en el estado estacionario de $q_{est} = 23.6^\circ$.

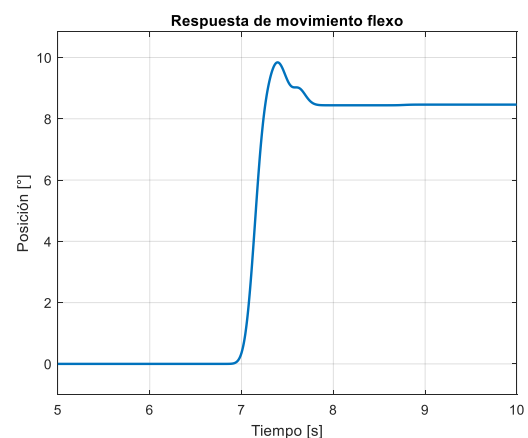


Figura 6: Respuesta de movimiento flexo de robot esférico.

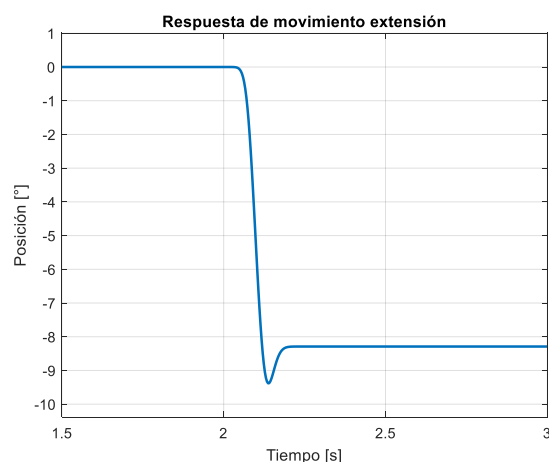


Figura 7: Respuesta de movimiento extensión de robot esférico.

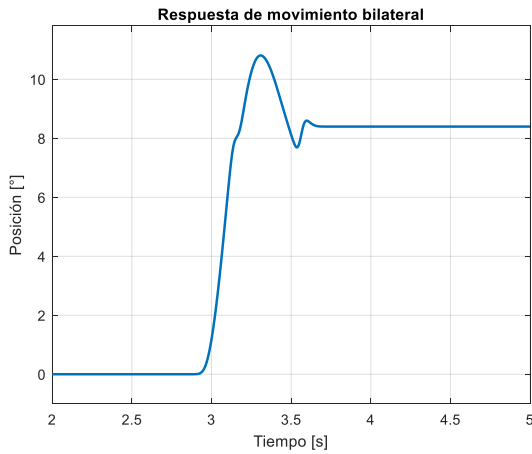


Figura 8: Respuesta de movimiento bilateral de robot esférico.

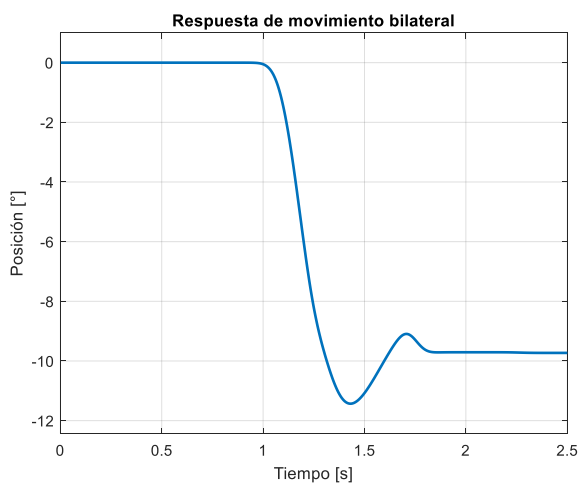


Figura 9: Respuesta de movimiento bilateral de robot esférico.

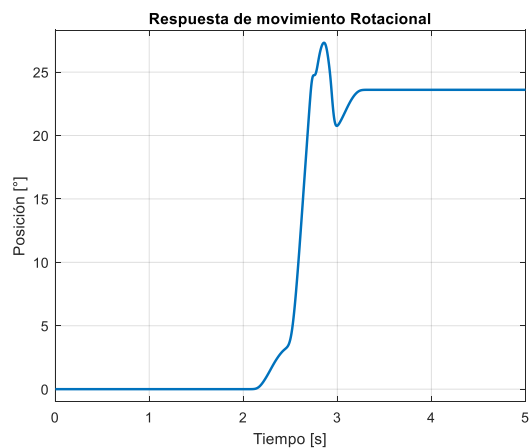


Figura 10: Respuesta de movimiento rotacional de robot esférico.

V. CONCLUSIONES

En este trabajo se diseñó y construyó un robot esférico de tres grados de libertad con el fin de emular el movimiento de un cuello para un robot humanoide. A partir del diseño del robot, se obtuvo el modelo dinámico, el cual describe adecuadamente el movimiento de un cuello humano. Se propuso un sistema de control para el robot basado en un lazo de control proporcional-derivativo con compensación gravitacional, sin embargo, la respuesta obtenida con este controlador genera un cambio de posición rápida en cada articulación, lo cual no emula correctamente el movimiento del cuello humano. Por consiguiente, al análisis anteriormente descrito se le debe compensar el sobretiro existente en todos los cambios de posición. Las posibles causas de sobretiro en el sistema pueden deberse en primer lugar a que no se ha considerado el centro de masa del sistema con el que contamos, este genera variaciones en la compensación gravitacional; por otro lado, el modelo dinámico no contempla otros fenómenos físicos existentes como la fricción. Así, como trabajo a futuro se puede añadir un control de trayectoria, el cual permita variar la velocidad de movimiento de cada una de las articulaciones, por lo cual se espera sea posible aproximar el comportamiento del robot al del cuello humano. De ser necesario, se contemplará el análisis del centro de masa de la cabeza que carga este robot.

VI. REFERENCIAS

- [1] F. J. L. C., «Desarrollo de un sistema mecatrónico para robot humanoide que permite emular el movimiento del cuello de los seres humanos,» *Benemérita Universidad Autónoma de Puebla, Facultad de ciencias de la electrónica.*, 2017.
- [2] «Bostom Dynamics,» 2019. [En línea]. Available: <https://www.bostondynamics.com/atlas>. [Último acceso: 14 Enero 2020].
- [3] I. I. d. Tecnología, «ICub,» 2009. [En línea]. Available: <http://www.icub.org>.
- [4] «La Nación,» 15 Abril 2015. [En línea]. Available: <http://www.lanacion.com.ar/1786374-desarrollan-un-expresivo-robot-que-responde-a-gestos-humanos>. [Último acceso: Noviembre 2015].
- [5] L. C. V. D. M. B. Rob Reilink, «Mechatronic design of the Twete humanoid head,» *Springer*, 2011.
- [6] M. J. Enos, *Dynamics and Control of Mechanical Systems: The Falling Cat and Related Problems*, Waterloo, ON, Canada: University of Waterloo, 1993.
- [7] F. R. Cortés, *Robótica: control de robots manipuladores*, Primera ed., Alfaomega, 2011.
- [8] M. Frank H. Netter, *Atlas de anatomía humana*, sexta ed., Elsevier Masson, 2014.
- [9] L. R. Valencia, «Sistemas dinámicos,» *Departamento de Física, Universidad de Chile*, 2008.

- [10] G. Z. Arce, «DISEÑO E IMPLEMENTACIÓN DE UNA TARJETA CON FPGA,» *MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA*, 2018.
- [11] K. Ogata, *Ingeniería de control moderno*, Pearson, 2010.
- [12] N. F. Figueroa-Mosquera, «Android Application USB-Serial datalogger for embedded systems ; Aplicativo Android datalogger serial-USB para sistemas embebidos,» *Interfases, Universidad de Lima*, vol. 9, n° 209-220, 2016.
- [13] A. Choudhary, «FPGA Based Solution For Ethernet Controller As Alternative For TCP/UDP Software Stack,» *IEEE Xplore Digital Library, 6th Edition of International Conference on Wireless Networks & Embedded Systems (WECON)*, 2018.
- [14] A. Soetedjo, Y. I. Nakhoda y C. Saleh, «An Embedded Platform for Testbed Implementation of Multi-Agent System in Building Energy Management System,» *Energies (19961073)*, vol. 12, p. 3665, 2019.

VII. APÉNDICE A

$$\begin{aligned}
 c_{1,1}(\mathbf{q}, \dot{\mathbf{q}}) &= \dot{q}_2[2l_3^2 m_3 - 2l_{c2}^2 m_2 - 2l_3^2 m_3 c_3^2] c_2 s_2 & (12) \\
 &\quad + 2\dot{q}_3 m_3 (-l_3^2 c_2^2 s_3) \\
 c_{1,2}(\mathbf{q}, \dot{\mathbf{q}}) &= -\dot{q}_2 [l_3^2 \sin(q_2) s_3 + b_1 l_3 c_2 m_3 c_3] \\
 &\quad + \dot{q}_1 [4b_1 l_3 m_3 c_2^2 s_3 - 2b_1 l_3 m_3 s_3] \\
 c_{1,3}(\mathbf{q}, \dot{\mathbf{q}}) &= \dot{q}_3 [(b_1 l_3 m_3 c_2 c_3)] \\
 &\quad + 2\dot{q}_2 l_3^2 c_2 c_3^2 m_3 + \dot{q}_1 [b_1 l_3 c_2 s_2 c_3] \\
 c_{2,1}(\mathbf{q}, \dot{\mathbf{q}}) &= \dot{q}_1 [l_1^2 m_3 + l_{c2}^2 m_2 - l_3^2 m_3] c_2 s_2 + b_1 l_3 m_3 s_3 \\
 &\quad - 2b_1 l_3 m_3 c_2^2 s_3 + l_3^2 m_3 s_2 c_2 c_3^2 \\
 c_{2,2}(\mathbf{q}, \dot{\mathbf{q}}) &= 2m_3 \dot{q}_3 l_3^2 c_3 s_3 \\
 c_{2,3}(\mathbf{q}, \dot{\mathbf{q}}) &= \dot{q}_3 b_1 l_3 m_3 s_3 \\
 &\quad + 2m_3 \dot{q}_1 b_1 l_3 s_2 s_3 - l_3^2 c_2 + l_3^2 c_2 c_3^2 \\
 c_{3,1}(\mathbf{q}, \dot{\mathbf{q}}) &= \dot{q}_1 [l_3^2 m_3 c_2^2 c_3 s_3 - b_1 l_3 m_3 s_2 c_2 c_3] \\
 c_{3,2}(\mathbf{q}, \dot{\mathbf{q}}) &= \dot{q}_2 [-l_3^2 m_3 s_3 c_3] \\
 &\quad + 2m_3 l_3 \dot{q}_1 [l_3 c_2 [1 - c_3^2] - b_1 s_2 s_3] \\
 c_{3,3}(\mathbf{q}, \dot{\mathbf{q}}) &= 0
 \end{aligned}$$

$$\begin{aligned}
 m_{1,1} &= m_1 l_1^2 + m_2 [l_1^2 + (\beta_1^2 + l_2^2 + 2\beta_1 l_2) \cos(q_2)^2 + 2l_1 (\beta_1 + l_2) \cos(q_2)] & (13) \\
 &\quad + m_3 [l_3^2 + (\beta_1^2 + \beta_2^2 + [l_3^2] [\cos(q_3)^2 - 1] + 2\beta_1 \beta_2) \cos(q_2)^2 \\
 &\quad + 2l_3 (\beta_1 + \beta_2) \cos(q_2) \sin(q_2) \sin(q_3)] \\
 m_{1,2} &= m_3 [l_3^2 \cos(q_2) \cos(q_3) \sin(q_3) - l_3 (\beta_1 + \beta_2) \cos(q_3) \sin(q_2)] \\
 m_{1,3} &= m_3 [l_3^2 \sin(q_2) + l_3 (\beta_1 + \beta_2) \cos(q_2) \sin(q_3)] \\
 m_{2,1} &= m_{1,2} \\
 m_{2,2} &= m_2 [\beta_1^2 + l_2^2 + 2\beta_1 l_2] \\
 &\quad + m_3 [\beta_1^2 + \beta_2^2 + l_3^2 \frac{1 + \cos(2q_3)}{2} + 2\beta_1 \beta_2] \\
 m_{2,3} &= m_3 l_3 (\beta_1 + \beta_2) \cos(q_3) \\
 m_{3,1} &= m_{1,3} \\
 m_{3,2} &= m_{2,3} \\
 m_{3,3} &= m_3 l_3^2
 \end{aligned}$$

$$\begin{aligned}
 \beta_1 &= 0.125m & (14) \\
 \beta_2 &= 0.108m \\
 l_1 &= 0.083m \\
 l_2 &= 0.03m \\
 l_3 &= 0.05m
 \end{aligned}$$

Diseño en Arduino de un Sistema de Seguimiento de Trayectorias Caóticas Esféricas para Robot Móvil

J. C. Núñez-Pérez
Instituto Politécnico Nacional,
IPN-CITEDI
Tijuana, Baja California, México
jnunez@ipn.mx

J. L. Flores Vacio
Universidad Autonoma de Baja
California, UABC
Tijuana, Baja California, México
jose.flores52@uabc.edu.mx

Y. Sandoval-Ibarra
Tecnologico Nacional de Mexico,
Instituto Tecnológico de Tijuana
Tijuana, Baja California, México
jumasaniiba@gmail.com

J. R. Cárdenas-Valdez
Tecnologico Nacional de Mexico,
Instituto Tecnológico de Tijuana
Tijuana, Baja California, México
jumasaniiba@gmail.com

M. J. Garcia-Ortega
Tecnologico Nacional de Mexico,
Instituto Tecnológico de Tijuana
Tijuana, Baja California, México
mdejgaro@yahoo.com

A. Calvillo-Téllez
Instituto Politécnico Nacional,
IPN-CITEDI
Tijuana, Baja California, México
calvillo@citedi.mx

Abstract—Este artículo muestra el procedimiento de diseño completo de un sistema completo de seguimiento de trayectorias pseudoaleatorias, basado en un sistema caótico esférico de cuatro alas y en una tarjeta Arduino MEGA. El artículo describe el proceso de análisis del sistema, su creación a base de los osciladores caóticos esféricos de cuatro alas, el algoritmo de control de posición de un robot diferencial y se muestran las simulaciones computacionales de las trayectorias pseudoaleatorias generadas. Además se desarrolló una interfaz gráfica en el software Matlab® para agregar flexibilidad al proceso de diseño y la obtención de los resultados. El diseño de este sistema se coloca al nivel del estado del arte actual en el diseño de aplicaciones para los sistemas caóticos.

Palabras clave— Arduino, caos, esférico de cuatro alas, seguimiento de trayectorias, robot móvil.

I. INTRODUCTION

En la literatura el caos es un concepto que hace referencia a un comportamiento que puede estar presente en varios fenómenos naturales, o fenómenos que involucran a la sociedad humana, como por ejemplo el clima, la formación de las costas hasta la oferta y la demanda. Un fenómeno presenta caos cuando el comportamiento de este presenta irregularidades, donde pareciera que no tiene ningún orden específico en su comportamiento y que no se rige por nada en específico. Sin embargo, la ciencia moderna describe que ese no es el caso en la realidad, ya que si se conocen las condiciones iniciales que generaron el comportamiento caótico que parece impredecible, se puede conocer cuál será su comportamiento en cualquier momento. Por ende en realidad la ciencia sugiere que el caos es un fenómeno determinista, donde si por alguna razón las condiciones iniciales para generar el caos se vuelven a repetir, se observará que se tendrá el mismo comportamiento que la

primera vez que este se generó, y así sucesivamente hasta que se decida cambiar mínimamente alguna de las condiciones iniciales que generan el caos [1-2].

La descripción breve de un sistema caótico consiste en un modelo con comportamiento no lineal donde sus condiciones iniciales traen consigo un comportamiento dinámico, y por tal motivo tiende a evolucionar [3]. El comportamiento de este tipo de sistemas se encontró de manera accidental por Lorenz quien en el campo de la meteorología describió que el comportamiento de un sistema caótico dependerá de las condiciones iniciales que lo describen [4].

El oscilador caótico de Chua es la razón principal por la cual se incrementaron en gran medida las investigaciones sobre los sistemas caóticos. Lo anterior debido a la complejidad que este puede desarrollar y de fácil construcción, el cual consta de dos capacitores, una resistencia y un último elemento el cual se le terminó por llamar diodo de Chua [5]. Además se demostró que solo se ocupa que el modelado matemático sea mediante un sistema de tres ecuaciones, y que una de ellas sea no lineal, por ende las otras dos pueden presentar linealidad [6].

La aplicación del caos no solo se limita a la encriptación de información. Además se puede utilizar en crear ambientaciones para comprobar que tan eficiente sería un robot dentro de ellas, incluso para hacer pruebas a sistemas inteligentes. También se pueden analizar las posibilidades de crear nuevas técnicas basadas en caos para desarrollar controladores que puedan ajustarse rápidamente a sistemas caóticos.

A los robots móviles se les clasifica de esta manera debido a que su principal función es desplazarse de manera efectiva, ya sea en forma área, terrestre o el mar. Su desarrollo principal es

para exploración, pero dentro de la industria también pueden ser utilizados en las actividades de traslado de objetos y limpieza. Así que dedicar tiempo al desarrollo de uno de estos robots beneficia al área industrial. La función principal del robot es acceder y realizar tareas que sean difíciles para las personas ya sea porque sean muy desgastantes, o que en su defecto sean peligrosas para la salud de la persona, por tal motivo la investigación de los robots móviles es importante.

Este artículo está organizado de la siguiente manera: en la sección II se presenta la teoría fundamental del sistema caótico esférico de cuatro alas, el seguimiento o planeación de trayectorias y el algoritmo para controlar la posición de un robot. En la sección III se presenta la simulación en Matlab. La sección IV presenta y discute los resultados, los cuales fueron comparados con los artículos reportados en el estado del arte. Finalmente, las conclusiones derivadas de este trabajo son descritas en la Sección V.

II. MARCO TEÓRICO

A. Sistema Caótico Esférico de Cuatro Alas

El oscilador caótico esférico de cuatro alas o *Four-Wing* es un sistema de gran interés debido a que permite realizar varias configuraciones con ciertas modificaciones a su conjunto de ecuaciones, el cual está representado por (1) [7]:

$$\begin{aligned} \frac{dx}{dt} &= a * (y - x) + e * y * z \\ \frac{dy}{dt} &= c * x + d * y - x * z \\ \frac{dz}{dt} &= -b * z + x * y \end{aligned} \quad (1)$$

donde los coeficientes del modelo son $a = 1$, $b = 43$, $c = -1$, $d = 16$, $e = 4$. La simulación tridimensional del sistema de ecuaciones (1), del oscilador caótico esférico de cuatro alas en el software Matlab, utilizando el método de Runge-Kutta de 4to orden, se muestra en la figura 1:

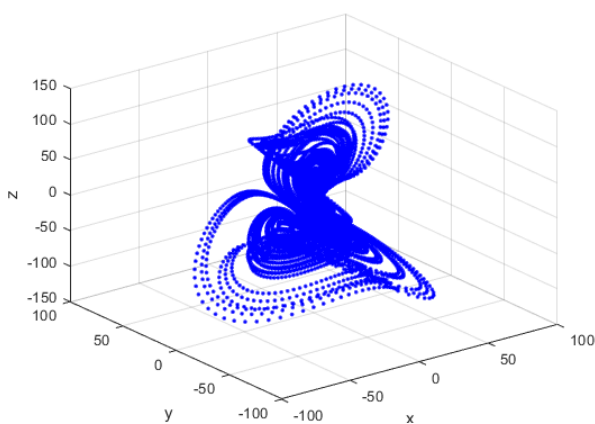


Fig. 1. Comportamiento tridimensional $x-y-z$ del sistema caótico esférico de cuatro alas.

La figura 2 muestra las distintas perspectivas bidimensionales del sistema caótico esférico de cuatro alas.

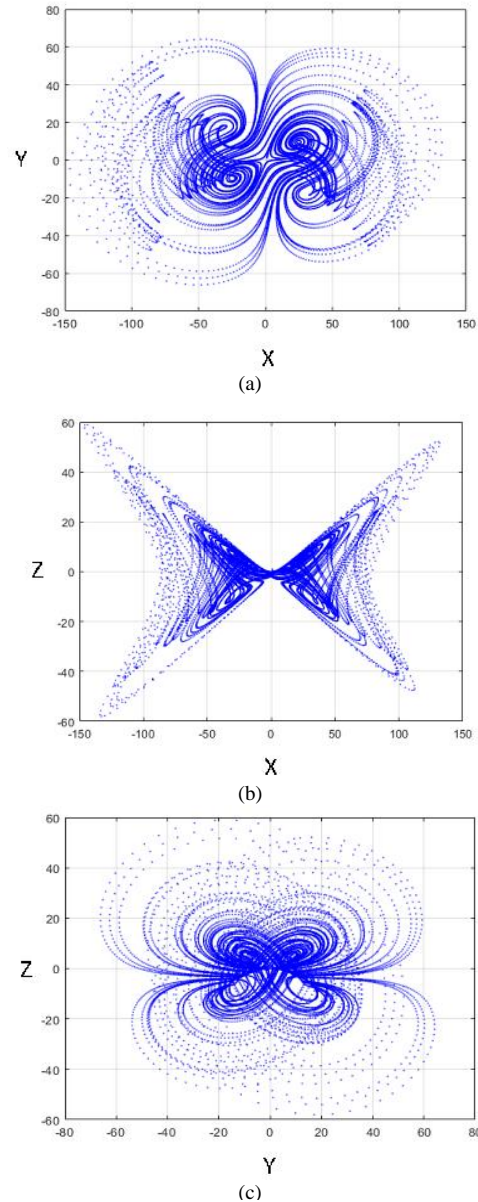


Fig. 2. Comportamiento del sistema caótico esférico de cuatro alas, visto desde tres perspectivas distintas de dos dimensiones a) $x-y$, b) $x-z$, c) $y-z$.

B. Seguimiento de Trayectorias

Los robots móviles tienen un enfoque muy específico en la exploración, por ello generan mucho interés en la investigación de diseño de caminos o trayectorias por las cuales estos se moverán. A este concepto se le conoce como sistema de seguimiento de trayectorias (del inglés, *path planning*), y se desarrolla para que no haya una persona detrás del control directo del robot. Por ende, la investigación es para que los robots realicen actividades de manera autónoma. La efectividad de un algoritmo para el seguimiento de trayectorias se encuentra

en la capacidad de responder a las cuestiones de navegación, donde este considera su posición actual, los puntos a los cuales se está dirigiendo y cómo llegará a todos esos puntos. Un sistema de seguimiento de trayectorias busca la eficiencia de navegación, localización, mapeo y planificación de movimiento, respectivamente:

- La localización ayuda a que el robot determine su ubicación en el entorno, para esto se pueden utilizar numerosos métodos como la utilización de una cámara, GPS, y sensores de ultrasonido.
- El mapeo parece similar a la localización, pero la diferencia es el entorno donde se hace el desplazamiento ubicando los obstáculos que pueden estar dentro, creando así un tipo de memoria para este robot.
- La planificación toma en cuenta la ruta diseñada de antemano donde se conoce la meta final, y las direcciones de la ruta pueden ser del tipo absoluto o relativo [8].

C. Algoritmo para Controlar Posición

Terminando de generar las trayectorias lo siguiente es generar un algoritmo que sirva para seguirlas, el robot móvil a trabajar es el diferencial, el cual se mueve mediante dos ruedas que tienen el mismo radio y que estas separadas por una distancia simétrica para ambas partes. Otro punto a destacar es que no considerarán elementos como la fricción, y el peso, por lo tanto se considera que el área donde se moverá es ideal. El diagrama cinemático para la resolución se ve en la figura 3:

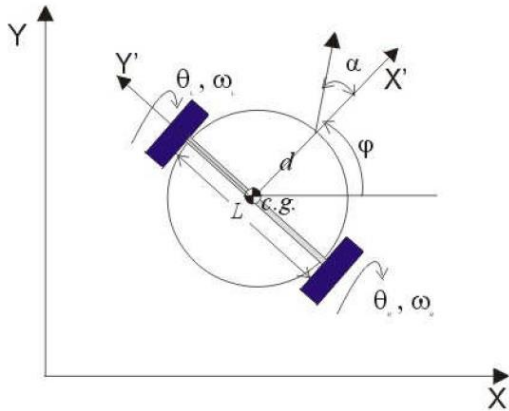


Fig. 3. Visualización de la ruta definida basada en comportamiento senoidal

Llevando a cabo el análisis se encuentra que el movimiento que puede tener este tipo de robot está dado por una velocidad lineal y otra velocidad angular. Donde la velocidad lineal implica que exista un cambio en las posiciones de los ejes "x" y "y". La velocidad angular influye en la dirección u orientación que este mismo tendrá, la velocidad lineal se origina cuando las dos ruedas giran a la misma velocidad, mientras que la angular se origina al no haber una igualdad. Tomando en cuenta que las ruedas tienen el mismo radio y que la distancia de separación entre ellas a un punto medio es simétrica, se obtiene el siguiente modelado para las velocidades:

$$v = \frac{(w_R + w_L)}{2} * R \quad (2)$$

$$v = \frac{(w_R - w_L)}{L} * R \quad (3)$$

Estableciendo que el robot móvil diferencial se encuentra en un punto en el plano de dos dimensiones y se desea mover a otro punto dentro de ese mismo plano, se deduce que si se plasma de manera simple en el diagrama se obtiene la figura 4:

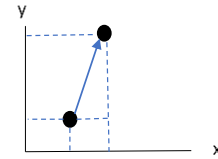


Fig. 4. Simplificación del diagrama que se muestra en la figura 3.

El cual termina por ser un triángulo, el cual retomando trigonometría, y teniendo en cuenta las velocidades, entonces se encuentra que el modelo que determina las posiciones es el siguiente:

$$\begin{aligned} x &= v * \sin \varphi \\ y &= v * \cos \varphi \\ \varphi &= w \end{aligned} \quad (4)$$

Estableciendo las ecuaciones (4) se puede hacer una modificación en estas para generar un tipo de memoria al robot móvil, debido a que cuando se mueve al tratar de alcanzar el punto deseado debe de saber en cuál punto en el plano se encuentra. Por lo tanto las ecuaciones (4) se modifican obteniendo el siguiente modelo:

$$\begin{aligned} x_{i+1} &= x_i + v * \sin \varphi \\ y_{i+1} &= y_i + v * \cos \varphi \\ \varphi_{i+1} &= \varphi_i + w \end{aligned} \quad (5)$$

En este punto se puede conocer la posición final resultante si se sabe en qué posiciones actuales se encuentra el robot y que velocidades se aplican. Sin embargo, en este caso no se conocen las velocidades, sino que se conocen los puntos deseados, por ende para lograr el seguimiento de trayectorias se generan velocidades. Para el algoritmo de seguimiento se divide la ecuación en matrices, una de posiciones actuales, otra de las velocidades deseadas y otra de las ecuaciones trigonométricas. Por lo anterior, el modelado de las ecuaciones (5), se convierte en el siguiente modelo:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ \varphi_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \varphi_i \end{bmatrix} + \begin{bmatrix} \sin \varphi & 0 \\ \cos \varphi & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} v \\ w \end{bmatrix} \quad (6)$$

Teniendo en cuenta estas modificaciones se puede proceder a realizar los despejes necesarios para dejar solo la matriz de

velocidad que se desea encontrar, dando como resultado el siguiente modelo:

$$\begin{bmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \\ \varphi_{i+1} - \varphi_i \end{bmatrix} * \begin{bmatrix} \sin \varphi & 0 \\ \cos \varphi & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} v \\ w \end{bmatrix} \quad (7)$$

Antes de llegar al cálculo de las velocidades se tiene que conocer el ángulo que hay entre la posición actual con la deseada, de tal modo que se conozca la orientación que el robot móvil necesita para poder avanzar. Para eso se toma en consideración un algoritmo que se basa en los cuatro cuadrantes del plano cartesiano, y utilizando la tangente [9] para este procedimiento. Tomando en cuenta todo lo descrito anteriormente entonces se tienen dos ecuaciones que determinan los catetos (8 - 9):

$$x\Delta = x_{i+1} - x_i \quad (8)$$

$$y\Delta = y_{i+1} - y_i \quad (9)$$

Así se tendrán cuatro casos que dan la orientación (10 - 13). Se elige el caso dependiendo de valor de "x" e "y", donde si los dos son valores positivos es el caso de la ecuación (10), y si "x" es negativa es el caso de la ecuación (11), si los dos son negativos es el caso de la ecuación (12) y si "x" es positiva e "y" negativa es el caso de la ecuación (13),

$$\varphi_d = \tan^{-1} \left(\frac{y\Delta}{x\Delta} \right) \quad (10)$$

$$\varphi_d = \frac{\pi}{2} + \tan^{-1} \left(\frac{x\Delta}{y\Delta} \right) \quad (11)$$

$$\varphi_d = \pi + \tan^{-1} \left(\frac{y\Delta}{x\Delta} \right) \quad (12)$$

$$\varphi_d = \frac{3 * \pi}{2} + \tan^{-1} \left(\frac{x\Delta}{y\Delta} \right) \quad (13)$$

Conociendo el ángulo deseado, lo siguiente son los algoritmos de control. Tomando la condición dependiente que la orientación deseada no es igual a la que posee el robot móvil se realiza un ajuste utilizando la ecuación (5) con ciertos despejes y dando un tiempo para aplicar esta velocidad angular, entonces se tiene una nueva ecuación:

$$w = \frac{\varphi_d - \varphi_i}{t} \quad (14)$$

El tiempo aplicado en estas iteraciones es de 0.1 segundos. Cuando la orientación del punto deseado y la orientación del robot móvil sean iguales se procede a aplicar el algoritmo para cálculo de las velocidades de la ecuación (7).

D. Generación de trayectoria definida

Esta trayectoria fue realizada con el fin de que, al momento de incorporarle la trayectoria caótica, surja una nueva y determinada pseudoaleatoria, llamada así porque la suma de ambas hará que el robot móvil pareciera que tiene un comportamiento aleatorio. Lo cual realmente no es cierto debido

a que la trayectoria obtenida consiste de dos rutas deterministas: la caótica que tiene comportamiento determinista, y la definida que se comporta con un patrón; la suma por ende se denomina pseudoaleatoria. El diseño de esta trayectoria se determinó que fuera una con el comportamiento que tiene una función senoidal, la cual está representada por la siguiente ecuación.

$$y(t) = \text{amplitud} \times \sin(2\pi \times \text{frecuencia} * \text{tiempo}) \quad (15)$$

Donde ciclo = $f * t$. El algoritmo de esta trayectoria consiste en su inicio en $t = 0$, con frecuencia = 0.2 Hz, Amplitud = 2.5. Al final de un ciclo "x" e "y" se intercambiarán dándole a "x" el comportamiento de "y", mientras que a "y" el comportamiento que tenía "x", y así sucesivamente. Después de cada ciclo se volverán a intercambiar su comportamiento. Además, una vez terminado el ciclo completo, donde ambas variables van a su punto de origen, habrá un cambio en la amplitud (+0.05) y otro en la duración del ciclo (+1). Con este cambio se recalcula la frecuencia, de tal manera que se tendrá una trayectoria incremental automática. La figura 5 muestra la trayectoria correspondiente generada con el comportamiento de la función senoidal, y alternando las variables "x" e "y".

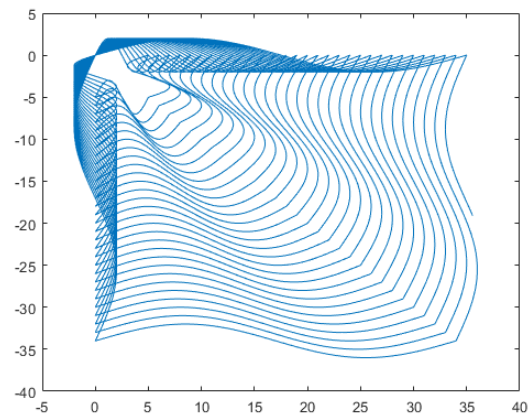


Fig. 5. Visualización de la ruta definida basada en comportamiento senoidal

III. SIMULACIÓN

La simulación del robot móvil se desarrolló en el programa de Matlab, se utilizó un código tipo librería. Este código contiene todos los elementos para observar cómo sería el movimiento del robot en un entorno definido por parámetros, tales como: el intervalo de pasos, salto para redibujar su nueva posición, las dimensiones del plano por donde se mueve, y el tiempo que dura en esa posición.

En la figura 6 se observan los resultados de simulación de (a) las trayectorias caóticas, (b) la trayectoria definida, y por último en (c) la trayectoria pseudoaleatoria generada de la suma de las trayectorias.

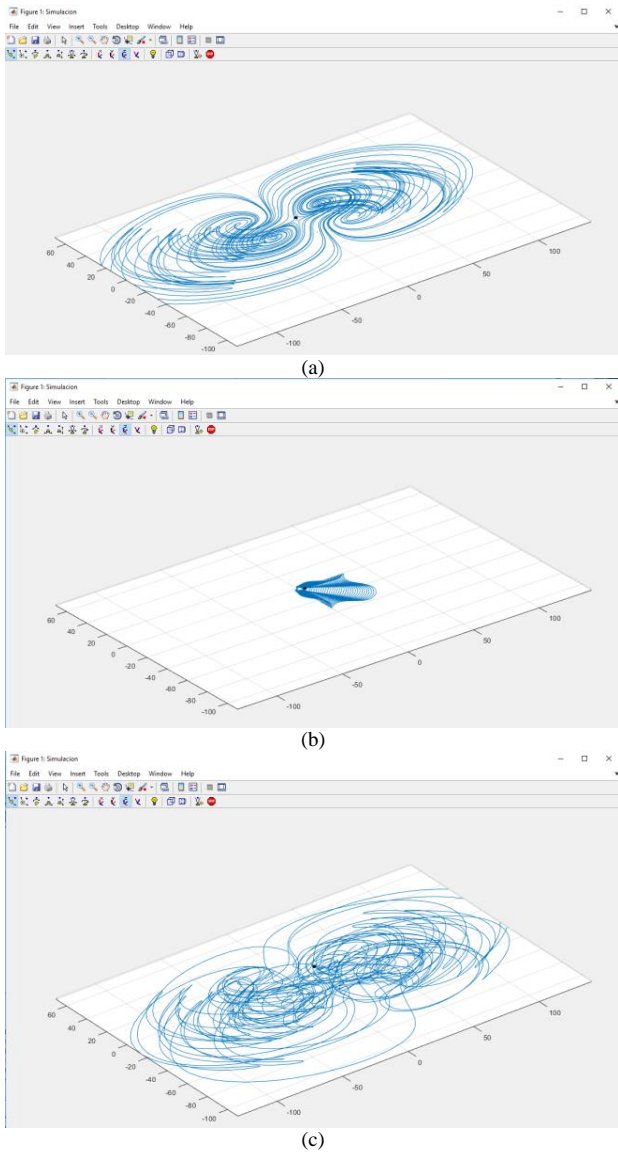


Fig. 6. Trayectorias generadas: (a) caótica de cuatro alas, (b) definida con comportamiento senoidal, (c) pseudoaleatoria

IV. RESULTADOS DE IMPLEMENTACIÓN

La placa de prototipado de Arduino que se utiliza es la tipo MEGA, la cual funciona con el lenguaje de programación C o C++. Por consiguiente se traduce lo escrito en Matlab a este lenguaje, debiéndose tomar en cuenta las diferencias entre cada lenguaje en la declaración de variables y vectores.

A. Generación de oscilador caótico esférico de cuatro alas

Se declara el sistema de ecuaciones (1) como funciones dentro del código para el Arduino, donde a los coeficientes y variables se le agrega un prefijo (por ejemplo, "val") debido a que el nombre de las variables no se debe de repetir, del mismo modo para el valor resultante de las funciones se le agrega el prefijo (por ejemplo, "d"). El método numérico empleado de

Runge-Kutta de 4to orden se declara usando la función "loop", para que este vaya iterando por cada vez que el bucle vuelva a leer el código. Al final de este mismo las variables de "x", "y", "z" son remplazadas con los nuevos valores obtenidos. Utilizando un osciloscopio se obtienen los resultados de implementación del oscilador caótico esférico que se muestran en la figura 7, donde se representan los ejes $x - y$, $x - z$, $y - z$.

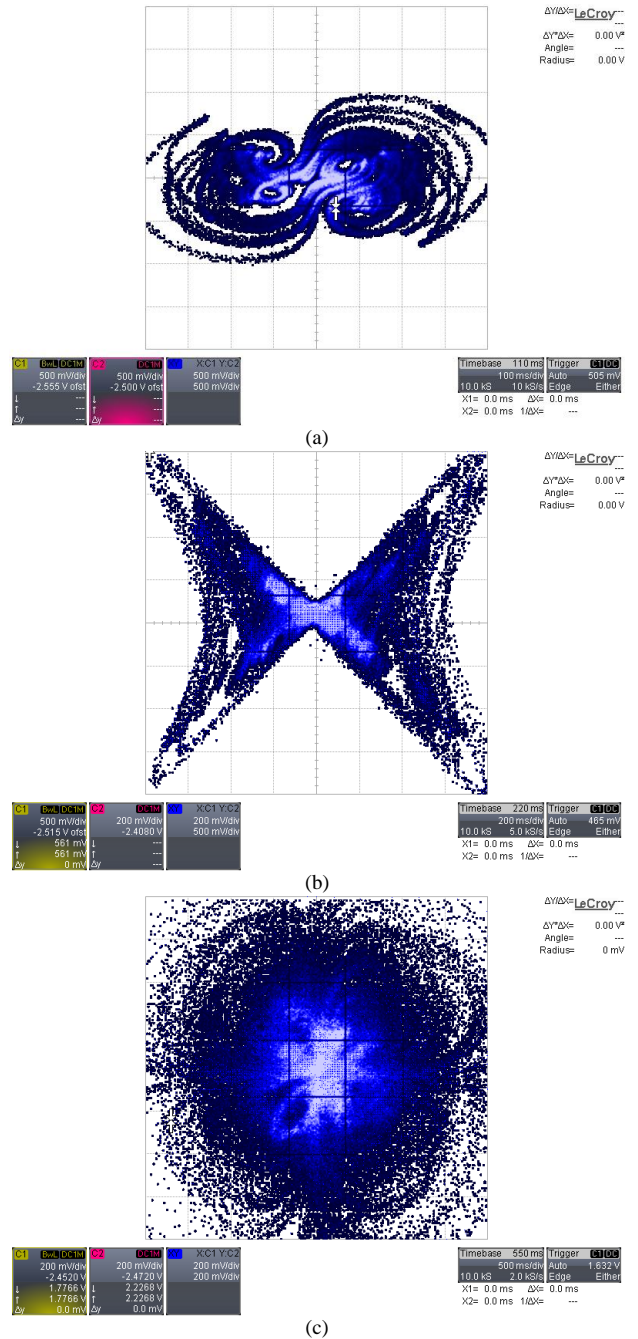


Fig. 7. Implementación del sistema caótico de cuatro alas en Arduino, visto desde tres perspectivas distintas de dos dimensiones a) $x - y$, b) $x - z$, c) $y - z$.

Para generar el algoritmo de trayectoria definida no se llevará a cabo una comprobación visualizada en un osciloscopio debido a que esa parte ya se demostró en el apartado 4A.

B. Implementación de señal pseudoaleatoria

Por cuestión de practicidad no se aplicó el algoritmo a un robot móvil diferencial, el cual se tenía como objetivo utilizar un Arduino Robot [10]. Se decidió comprobar solo la funcionalidad del sistema haciendo uso de Matlab-Simulink.

Los datos se envían al programa Matlab-Simulink mediante comunicación serial. La configuración de la comunicación serial se realiza a través de los bloques “*Query Instrument*”, “*demux*” y “*XY Graph*”. La figura 8 muestra el resultado de la trayectoria pseudoaleatoria después de haber ejecutado el programa.

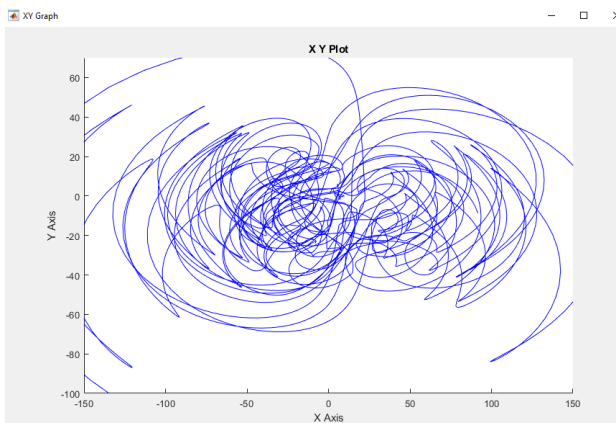


Fig. 8. Resultado de la trayectoria caótica implementada en Arduino

En la figura 8 se puede demostrar la funcionalidad de la propuesta de este trabajo de investigación usando Matlab®. La aplicación generada es capaz de cambiar el algoritmo caótico, y la trayectoria definida, esto con el fin de poder manejar de manera más eficiente las funciones de creación y visualización del seguimiento de trayectorias para robots móviles.

En comparación con el estado del arte actual [1-10], este trabajo reporta por primera vez el diseño de una interfaz gráfica de usuario que permite la configuración de las trayectorias pseudoaleatorias usando un sistema caótico esférico de cuatro alas y además con trayectorias definidas reprogramables. Esta interfaz permitirá el diseño metodológico de nuevos osciladores caóticos para generar nuevos valores y trayectorias pseudoaleatorias para diversas aplicaciones.

V. CONCLUSIONES

Se diseñó un sistema que permite la generación de trayectorias pseudoaleatorias para un robot móvil. Estas trayectorias corresponden a una combinación entre un sistema caótico de cuatro alas y una trayectoria senoidal definida. Se simuló también el control de posición de un robot móvil, y se llevó el diseño a una tarjeta Arduino MEGA, la cual tiene ciertas limitantes por hacer uso de un microcontrolador y un PWM de solo 8 bits. La generación de trayectorias pseudoaleatorias basado en las descripciones de un oscilador caótico esférico de cuatro alas y una trayectoria senoidal definida pudo ser realizada, de una manera general y compacta a través de una interfaz gráfica generada en Matlab®. La implementación hecha a partir de una tarjeta Arduino MEGA permite generar distintas trayectorias empleando una metodología simple de entender e implementar. El sistema es capaz de intercambiar el tipo de oscilador caótico y la trayectoria definida sin tener dependencia, y con una variedad de aplicaciones futuras.

AGRADECIMIENTOS

Los autores desean agradecer al Instituto Politécnico Nacional por el apoyo financiero a través del proyecto de investigación “SIP-2020156.

REFERENCIAS

- [1] Cheng, D., Hu, X. and Shen, T., “Analysis and design of nonlinear control systems.” 1st ed. New York: Springer, Press 2011.
- [2] D.P. Feldman, “Chaos and Fractals An Elementary Introduction”, 1st ed: Oxford, Press 2012.
- [3] H.O. peitgen, H. Jurgens, D. Saupe (2004),” Chaos and Fractals: New Frontiers Of Science”, 1ST Ed, United States: Springer.
- [4] M. Ababneh, “A new four-dimensional chaotic attractor”, Ain Shams Engineering Journal, vol 9, pp.1849-1854, 2018.
- [5] J. Pospisil, J.Brzobohaty, “ Elementary canonical state models of Chua’s circuit family”, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol 43, Issue: 8, Aug 1996.
- [6] P. J. Obeso, “Diseño e Implementación en un FPGA de Oscilador Caótico para Aplicaciones en Sistemas de Seguridad”, Tesis de Maestría en CITEDI-IPN, Tijuana, México, Junio 2015.
- [7] R.Wang, H. Sun, J. Wang, L. Wang, and Y. Wang. “Applications of modularized circuit designs in a new hyper-chaotic system circuit implementation”. Chinese Physics B, Vol 24: Issue 2, p.020501, 2015.
- [8] A. Koubaa. Robot Path Planning and Cooperation. 1st ed. Springer, 2019.
- [9] R. Pallas-Areny, (2001), “Sensors and Signal Condition” 2nd Ed, New York: A Wiley-Interscience Publication.
- [10] "Arduino - Robot", Arduino.cc, 2019. [Online]. Available: <https://www.arduino.cc/en/Guide/Robot>. [Accessed: 1- March- 2020].

ATTITUDE ESTIMATION OF A MOBILE ROBOT BY DATA FUSION AND A LOWER HEADING REFERENCE SYSTEM

B. Benito Salmerón-Quiroz
SEPI, ESIME Azcapotzalco
Instituto Politécnico Nacional
Ciudad de México
bsalmeron@ipn.mx
Gerardo Villegas Medina
SEPI, ESIME Azcapotzalco
Instituto Politécnico Nacional
Ciudad de México
gvillegasm@ipn.mx

Jose Eduardo Hernandez
Hernandez
SEPI, ESIME Azcapotzalco
Instituto Politécnico Nacional
Ciudad de México
lalohdz53@live.com.mx

Nestor Javier Arroyo de Luna
SEPI, ESIME Azcapotzalco
Instituto Politécnico Nacional
Ciudad de México
nestorarroyo38@gmail.com

Maribel Angelica Mendoza
Nuñez
ESIME Culhuacan
Instituto Politécnico Nacional
Ciudad de México
mmendozan@ipn.mx

Abstract— Generally, the attitude estimation and the measurement of the angular velocity are a requirement for the attitude control. As a result, the computational cost and the complexity of the control loop are relatively high. In the present paper, a technique for attitude stabilization is proposed; the technique proposed is designed with attitude estimation based in data fusion of an Attitude and Heading Reference System AHRS system. With this approach, only the measurements of at least two noncollinear directional sensors are needed. Since the control laws are highly simple and a model based in an observer for angular velocity reconstruction is not needed, the proposed new strategy is very suitable for embedded implementations. The global convergence of the estimation techniques is proved. Simulations with some robustness tests are performed.

Keywords—*Robotic, Control, Attitude, Quaternion, Non-Linear Observer*

I. INTRODUCTION

Autonomous robotic systems have been suggested for a number of applications such as unmanned aerial vehicles, unmanned underwater vehicles and robots manipulators in medical applications and Mobile Robot.

A fundamental requirement for an autonomous vehicle is its ability to localize itself with respect to its environment. Navigation on a flat and horizontal ground only requires estimations of position and heading. However, in many cases, the environment is not so well structured, and the angular orientation of the vehicle may change along its path. In this case, a real time estimation of the attitude may be necessary. Indoor mobile robots generally need an accurate navigation system. If

they move on inclined planes, the orientation can be used to correct the position and heading estimations provided by odometry, or to interpret external sensor data, proximity, in order to build an accurate model of the environment, recent use of MEMs has been introduced.

The attitude estimation of an autonomous vehicle is a subject that has attracted a strong interest the last years. This is due to the fact that can be applied to multiple applications, such as spacecrafts, satellites, tactical missiles, where the attitude is essential for control or monitoring purposes. In the last decade the application of Micro-Electro-Mechanical-Systems (MEMS) has gained a strong of interest. In addition to traditional attitude estimation in aerospace and automobile communities, the reduced cost of MEMS inertial sensors has spurred new applications in robotics [1], virtual reality and biomechanics [2]. Furthermore, this increasingly interest has motivated the development of low cost, lightweight and low-power consumption Attitude and Heading Reference Systems (AHRS).

An AHRS is composed of inertial and magnetic sensors, namely, three rate gyros, three accelerometer and three magnetometers, orthogonally mounted such that the sensor frame axes coincide with the body frame in question. This attitude estimation problem is described as following: Rate gyros provide continuous attitude information with good short-term stability when their measurements are integrated. The attitude (orientation) of a rigid body can be parameterized by several methods: for instance, Euler's angles, Cardan angles and unit quaternion. The unit quaternion is a four parameter representation with one constraint. Therefore, it yields the

lowest dimensionality possible for a globally non-singular representation of the attitude. For more details on attitude representations, the reader can refer to the survey written by [3]. Several approaches have been applied to the attitude estimation problem. These estimators fall into three main families.

The first one deals with a constraint least-square minimization problem proposed firstly by Wahba [4][5]. These techniques have been adapted to sequentially estimate time-varying attitude [3][6].

The second approach is within the framework of the Extended Kalman Filter [6] (EKF). Its major feature concerns the ability to fuse signals acquired from different sensor types. An excellent survey of these methods is given in [3].

The third approach issues from nonlinear theory, and non linear observers are applied to the attitude determination problem [7], [8],[9]. In this approach, the convergence of the error to zero is proved in a Lyapunov sense.

In this paper, an attitude estimator using quaternion representation is studied. Two approaches are jointly used, namely a constraint least-square minimization technique and a prediction technique. The prediction is performed in order to produce a pseudo-estimate of the acceleration and of the attitude quaternion. This predictor is driven by a quaternion pseudo-measurement error which is obtained from the quaternion propagated through the kinematic equation and the one obtained via the constraint minimization problem. Actually, this latter problem is divided in three steps. First, the body accelerations are estimated from the previously computed quaternion. Using these updated accelerometer measurements together with the magnetometer measurements; a pseudo-measure quaternion is estimated via an optimization technique.

The present paper is organized as follows. In section 2 a quaternion-based formulation of the orientation of rigid body is given. The problem statement is formulated in section 3. The attitude's estimation and prediction is presented in section 4. Simulation results are given in section 5-A. The paper ends with some concluding remarks given in section 6

II. MATHEMATICAL BACKGROUND

As mentioned in the introduction, the attitude of a rigid body can be represented by a unit quaternion, consisting of a unit vector \vec{e} known as the Euler axis, and a rotation angle β about this axis. The quaternion q is then defined as follows:

$$q = \begin{pmatrix} \cos \frac{\beta}{2} \\ \vec{e} \sin \frac{\beta}{2} \end{pmatrix} = \begin{pmatrix} q_0 \\ \vec{q} \end{pmatrix} \in \mathbb{H} \quad (1)$$

Where

$$\mathbb{H} = \{q \mid q_0^2 + \vec{q}^T \vec{q} = 1, q_0 \in \mathbb{R}, \vec{q} \in \mathbb{R}^3\} \quad (2)$$

$\vec{q} = [q_1 \ q_2 \ q_3]^T$ and q_0 are known as the vector and scalar parts of the quaternion respectively. In attitude control applications, the unit quaternion represents the rotation from an inertial coordinate system $N(x_n, y_n, z_n)$ located at some point in the space (for instance, the earth NED frame), to the body coordinate system $B(x_b, y_b, z_b)$ located on the center of mass of a rigid body.

If \vec{r} is a vector expressed in N , then its coordinates in B are expressed by:

$$\vec{b} = \bar{q} \otimes r \otimes q \quad (3)$$

Where $b = [0 \ \vec{b}^T]^T$ and $r = [0 \ \vec{r}^T]^T$ are the quaternions associated to vectors \vec{b} and \vec{r} respectively. \otimes denotes the quaternion multiplication and \bar{q} is the conjugate quaternion multiplication of q , defined as:

$$\bar{q} = [q_0 \ -\vec{q}^T]^T \quad (4)$$

The rotation matrix $C(q)$ corresponding to the attitude quaternion q , is computed as:

$$C(q) = (q_0^2 - \vec{q}^T \vec{q}) I_3 + 2(\vec{q} \vec{q}^T - q_0 [\vec{q}^x]) \quad (5)$$

Where I_3 is the identity matrix and $[\xi^x]$ is a skew symmetric tensor associated with the axis vector ξ :

$$[\xi^x] = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix}^x = \begin{pmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{pmatrix} \quad (6)$$

Thus, the coordinate of vector \vec{r} expressed in the B frame is given by:

$$\vec{b} = C(q) \vec{r} \quad (7)$$

The quaternion attitude error used to quantify the mismatch between two attitudes q_1 and q_2 is computed by:

$$\bar{q}_2 q_1 = q_e \otimes q_1 \quad (8)$$

III. PROBLEM STATEMENT

In the case of the attitude estimation, one seeks to estimate the attitude and accelerations of a rigid body. From now on it is assumed that the system is equipped with a tri axis accelerometer, three magnetometer and three rate gyros mounted orthogonally.

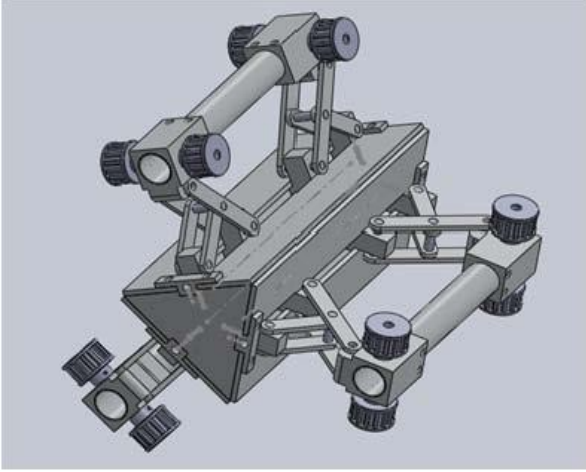


Fig. 1. Prototype of the mobile robot

In this section we describe the body's kinematic of the model. In the present work, our typical capture configuration relies primarily on the Robot of figure 1 equipped with six 1 inch diameter wheels driven by 3 DC gear head motors. The mechanical model is based on single pinion architecture suitable for light vehicles and consists of following elements: a steering rack, a steering column coupled to the steering rack through a pinion gear, and the assist motor. The combination of this information jointly to knowledge a priori of the robot makes possible to obtain information on the mobile robots respect to the base.

The controller was able to set the power level to each motor independently, but there was no feedback loop based on tachometers, or current sensing. In order to estimate the attitude robot position with respect to an inertial frame, a module containing three rate gyros, three accelerometer and three magnetometer assembled in tri axis, are positioned near the centre of robot. Thus, the attitude for the articulation is estimated. The combination of this information jointly to knowledge a priori of the dynamic of the movement of the robot makes possible to obtain information of the attitude of the robot respect to the base.

The equation describing the relation between the quaternion and the body's kinematic is given in introducing the angular variation

$$\vec{w} = [w_x \ w_y \ w_z]^T \text{ From this, it follows.}$$

$$\dot{q} = \frac{1}{2}\Omega(\vec{w})q(t) = \frac{1}{2}\Xi(q)\vec{w}(t) \quad (9)$$

Where $\Omega(\vec{w})$ y $\Xi(q)$ are defined as:

$$\Omega(\vec{w}) = \begin{bmatrix} -[\vec{w} \times] & \dots & \vec{w} \\ \dots & \dots & \dots \\ -\vec{w}^T & \dots & 0 \end{bmatrix} \quad (10)$$

$$\Xi(q) = \begin{bmatrix} q_0 I_{3 \times 3} + [\vec{q} \times] \\ \dots \\ -\vec{q}^T \end{bmatrix} \quad (11)$$

The quaternion must be:

$$q^T q = \vec{q}^T \vec{q} + q_0^2 = 1 \quad (12)$$

In the other hand, the matrix $\Xi(q)$ has the relation:

$$\Xi^T(q)\Xi(q) = q^T q I_{3 \times 3} \quad (13)$$

$$\Xi(q)\Xi^T(q) = q^T q I_{4 \times 4} - q^T q$$

$$\Xi^T(q)(q) = 0_{3 \times 1}$$

Generally $\Xi^T(q)\lambda = -\Xi^T(\lambda)q$, for any $\lambda \in \mathbb{H}$.

$$C(q) = (q_0^2 - \vec{q}^T \vec{q}) I_{3 \times 3} + 2\vec{q} \vec{q}^T - 2q_0 [\vec{q} \times] \quad (14)$$

That is denoted like the orientation matrix 3-D of dimension 3x3.

3.1. Modeling sensors

1) Rate Gyros: The angular velocity $\vec{\omega} = [\omega_1 \omega_2 \omega_3]^T$ is measured by the rate gyros, which are supposed to be orthogonally mounted. The output signal of a rate gyro is influenced by various factors, such as bias drift and noise. Since an integration step is required in order to obtain the current attitude quaternion (9), even the smallest variation of the rate gyro measurement will produce a wrong estimation of the attitude. The bias is denoted by \vec{V} , belonging to space \mathbb{R}^3 . The rate gyro measurements are modeled by [7]:

$$\vec{\omega}_G = \vec{\omega} + \vec{v} + \vec{\eta}_G \quad (15)$$

$$\dot{\vec{v}} = -T^{-1}\vec{v} + \vec{\eta}_v \quad (16)$$

Where $\vec{\eta}_G$ and $\vec{\eta}_v$ supposed by Gaussian white noises and $T = \tau I_3$ is a diagonal matrix of time constants. In this case, the constant τ which has been set to 100 s. The bias vector \vec{v} will

be estimated online, using the observer presented in the following section.

2) Accelerometers: Since the 3-axis accelerometer is fixed to the body, the measurements are expressed in the body frame B. Thus, the accelerometer output can be written as:

$$\vec{b}_A = C(q)(\vec{a} - \vec{g}) + \vec{\eta}_A \quad (17)$$

where $\vec{g} = [0 \ 0 \ g]^T$ and $\vec{a} \in \mathbb{R}^3$ are the gravity vector and the inertial accelerations of the body respectively. Both are expressed in frame N. $g = 9.81 \text{ m/sec}^2$ denotes the gravitational constant and $\vec{\eta}_A \in \mathbb{R}^3$ is the vector of noises that are supposed to be white Gaussian.

3) Magnetometers: The magnetic field vector \vec{h}_M is expressed in the N frame it is supposed to be $\vec{h}_M = [h_{Mx} \ 0 \ h_{Mz}]^T$. Since the measurements take place in the body frame B, they are given by:

$$\vec{b}_M = C(q)\vec{h}_M + \vec{\eta}_M \quad (18)$$

Where $\vec{\eta}_M \in \mathbb{R}^3$ denotes the perturbing magnetic field. This perturbation vector is supposed to be modeled by Gaussian white noises.

IV. ATTITUDE'S ESTIMATION AND PREDICTION

The attitude estimator uses quaternion representation. Two approaches are jointly used, namely a estimation with a constraint least-square minimization technique and a prediction of the estate at the instant k. The prediction is performed in order to produce a pseudo-estimate of the accelerations and the attitude quaternion. This prediction is driven by a estate which is obtained from the quaternion propagated through the kinematic equation and the one obtained via the constraint minimization problem.

Actually, this latter problem is divided in three steps. First, the body accelerations are estimated from the previously computed quaternion. Then, the influence of the body accelerations is predicted from the accelerometer measurements. Using these updated accelerometer measurements together with the magnetometer measurements, a pseudo-measure estate is estimated via an optimization technique. In this way, the quaternion that is obtained by the estimation with a constraint least-square is insensitive to the body accelerations. Thus, no assumptions of the weakness (or not) of the accelerations are done, and no switching procedure from one model to another one is necessary.

Therefore, the main advantage of the approach presented in this paper compared to others approaches, is that the estimated attitude remains valid even in the presence of high accelerations over long time periods.

In this paper a critter that takes in account the evolution of the attitude state via determination of $\mathbf{x} =$

$[q_0, q_1, q_2, q_3, a_x, a_y, a_z]^T$ in the function $f(\mathbf{x})$ is proposed. The minimum error is chosen, but it takes in account the prediction of the state $\hat{\mathbf{x}}$ and the coefficients of weight for the state μ and the measures estimated (MesEstimated = MS) at the instant k.

$$f(\mathbf{x}) = \frac{1}{2} \left[\mu \left(\sum_{j=1}^n (\text{MesEstimated} - v_{\text{mes}}(j))^2 \right) + (1 - \mu) \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 \right] \quad (19)$$

with $q^T q - 1 = 0$

The process of Estimation and Prediction needs the determination of his gradient; this one is obtained by equation (19).

$$H_q = \left[\frac{\partial}{\partial q} \left(\frac{\partial f}{\partial q} \frac{\partial q}{\partial \mathbf{x}} \right) \right] \quad (20)$$

$$H_q = \left[\frac{\partial^2 f}{\partial q^2} \frac{\partial q}{\partial \mathbf{x}} \right]^T$$

Similarly, is the obtention for the gradient of the state for the case of acceleration.

Finally, the total Gradient is obtained by the fusion between the calcule show for the quaternion case a the gradient omitted for the acceleration case.

$$F(\mathbf{x}) = \begin{bmatrix} H_q & \frac{\partial g_q}{\partial a} \\ \frac{\partial g_a}{\partial \mathbf{x}} & H_a \end{bmatrix} \quad (21)$$

For the prediction's process of $\hat{\mathbf{x}}$, several tehcnique have been validated, for purpose of simplicity, the prediction via spline is chosen. Cubic spline is a spline constructed of piecewise third-order polynomials wich pass through a set of n control points.

Suppose we are n+1 data point $(\hat{\mathbf{x}}_k, MS_k)$ such that.

$a = x_0 \dots x_n$, Then the coefficients of the vector μ exists cubic polynomials with coefficients μ_{ij} $0 \leq i \leq 3$ such that the following hold.

- 1) $\mu(\hat{\mathbf{x}}) = \mu_j(\hat{\mathbf{x}}) = \sum_{j=0}^3 [\hat{\mathbf{x}} - x_j]^i \quad \forall \hat{\mathbf{x}} \in [\hat{\mathbf{x}} - x_{j+1}] \ 0 \leq k \leq n - 1$
- 2) $\mu(x_j) = y_k \quad 0 \leq k \leq n - 1$
- 3) $\mu_j(x_{j+1}) = \mu_{j+1}(\mu_{j+1}) \quad 0 \leq k \leq n - 2$
- 4) $\mu'_j(x_{j+1}) = \mu'_{j+1}(x_{j+1}) \quad 0 \leq k \leq n - 2$
- 5) $\mu''_j(x_{j+1}) = \mu''_{j+1}(x_{j+1}) \quad 0 \leq k \leq n - 2$

So, we see that the cubic spline not only interpolates the data $(\hat{\mathbf{x}}_k, MS_k)$ but matches the first and second derivatives at the knots. Notice, from the above definition, one is free to specify constrains on the endpoints. The end point constrains $\mu''(a) = 0 \ \mu''(b) = 0$ is chose.

The estimation of the torque is part of another work that is in process and only we present his basic model. Since the driver torque is not measured in line, we introduce an estimator for $\Gamma_{Mot} = \Gamma$, Essentially, the estimated value of the driver torque is

$$\Gamma_{iest} = G^{(-1)}(\Gamma_{LZ}(z) - H(z) \cdot \Gamma_{Mot}(z)) \quad (22)$$

Where Γ_{LZ} is the torque in the steering column part and Γ_{Mot} is the assist motor torque. In order that $G^{(-1)}$ can be physically realizable (numerator degree of the transfer function is always less or equal than denominator degree), it is necessary to introduce a correction transfer function $G_c(z)$ to maintain the properness. With this correction, the inverse transfer function becomes

$$\Gamma_{iest} = G^{(-1)} * G_c(z) * (\Gamma_{LZ}(z) - H(z) \cdot \Gamma_{Mot}(z)) \quad (23)$$

V. RESULTS

5.1. Modeling sensors

In this section, some simulation results are presented in order to show the performance of the proposed control laws. A rigid body with low moment of inertia is taken as the experimental system. In fact, the low moment of inertia makes the system vulnerable to high angular accelerations which proves the importance to apply the control.

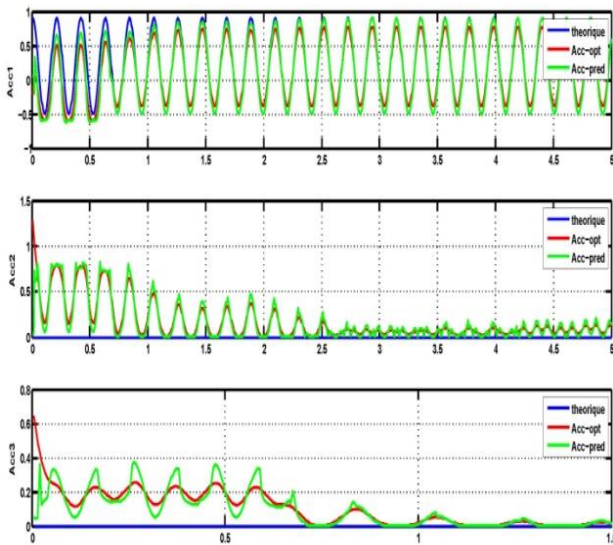


Fig. 2. Simulation Study / Estimation and Prediction of the Acceleration

The proposed technique is compared to the existing methods (namely, the Multiplicative Extended Kalman Filter (MEKF) and the Additive Kalman Filter (AEKF)). Initial conditions are set to extreme error values in order to assess the effectiveness of attitude estimation. These results are depicted in figure 2 and

figure 3. The proposed method performances are similar to those of the Extended Kalman Filter (Multiplicative and Additive). However, for extreme errors the convergence rate for our estimation-prediction is higher.

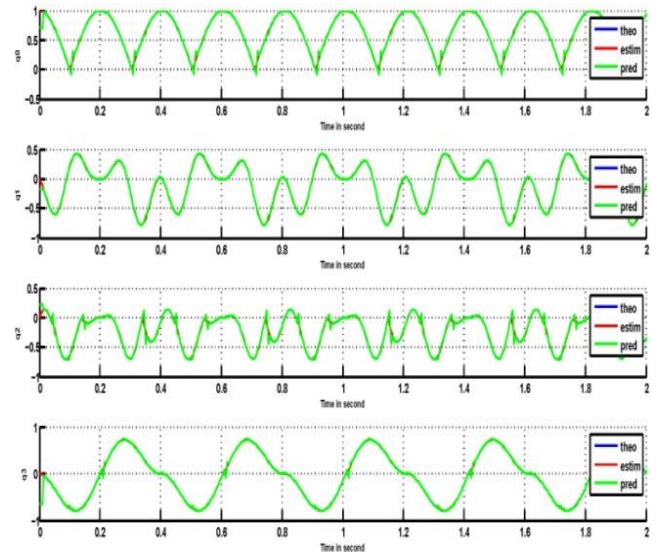


Fig. 3 Estimation and Prediction of the Quaternion

5.2. Experimental results

The estimation methodology proposed in this work is implemented and evaluated in real time, in order to assess its effectiveness. For this purpose an embedded system was designed, and developed.

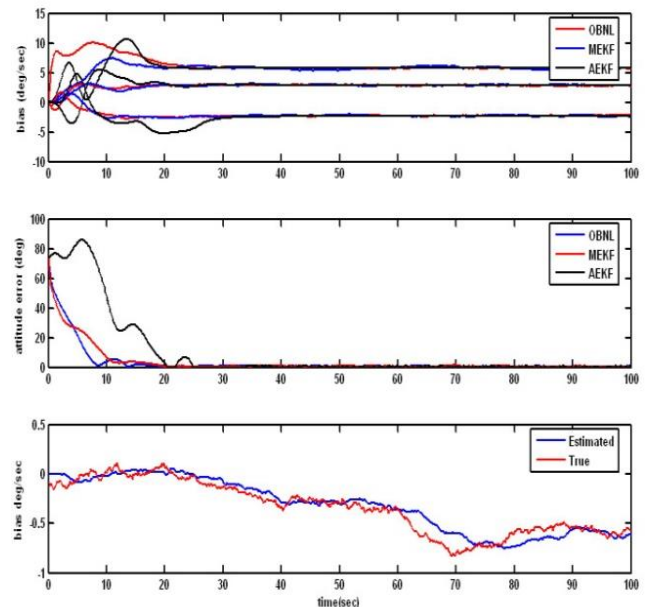


Fig.5. Constant bias estimated, error estimation, variant bias estimated

Special attention was paid to the low power consumption requirements and weight, leading to the selection of the Digital Signal Controller dsPIC which was used with a clock speed of 4MHz. It contains extensive Digital Signal Processor (DSP) functionality with high performance 16 bit microcontroller (MCU) architecture but without floating point unit.

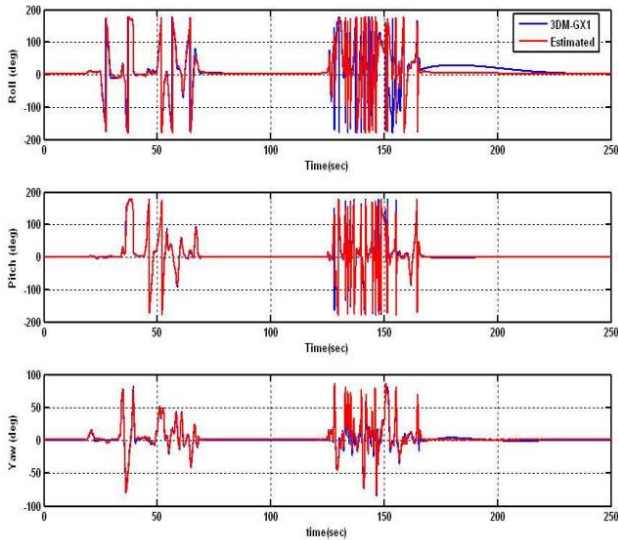


Fig 6. Experimental Data

The sensor suite is based on a sensor board equipped with a tri-axis accelerometer (ADXL135), a dual axis gyro (LPR530AL) and single axis gyro (LY 530ALH) and a tri-axis magnetometer (Micromag3). All sensors outputs are analog except the Micromag 3 which is digital and uses the Serial Peripheral Interface (SPI) bus system as underlying physical communication layer. The total system supply voltage is 3.3 V. The dimension and weight are 60x40x15mm and 60g, respectively. For purpose of validation A Commercial AHRS (Sensors, 2013) is used to acquire the data instead of the MEMS sensors presented in section (III-A) (Robot showed in virtual reality figure 1). This AHRS also provides the Euler angles. The methodology of estimation and prediction are implemented in real-time using the LabView environment. Remember that the

attitude estimate is computed using a unit quaternion formulation. For comparison purpose, the estimate quaternion is converted into Euler angles.

As can be shown, after large angular velocity change over a long period, the AHRS has a low convergence rate (approx. 1 min) compared to the one achieved with our proposed methodology. On the other hand, this system doesn't provide the acceleration of the body so for validation we have done slowly movement and abrupt movement to appreciate the effect of the acceleration in our method. In figures 5 and 6 the global convergence of the estimation techniques is proved.

ACKNOWLEDGMENT

The authors would like to thank to the Instituto Politécnico Nacional, SEPI ESIME Unidad Azcapotzalco, ESIME Unidad Culhuacan the SIP and the CONACYT.

REFERENCES

- [1] M., Gavrilă B.D. (1999). The visual analysis of human movement: a survey. *Comput. Vis. Image Underst.* pp. 88–98.
- [2] Choo K., Fleet D. J. (2003). People tracking using hybrid monte carlo filtering. *Eighth International Conference on Computer Vision (ICCV'01)* 2, 321.
- [3] Choukroun, D. (2003). Novel methods for attitude determination using vector observations. Tesis de doctorado. Israel Institute of Technology, Haifa, Israel.
- [4] Lefferts, E.J., F.L. Markley y M.D. Shuster (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics* 5(5), 417–429.
- [5] Salcedo, S. (1991). A globally convergent velocity observer for rigid body motion. *IEEE Transactions on Automatic Control* 36(12), 1493–1497.
- [6] Sensors, MicroStrain Microminature (2013). <http://www.microstrain.com>.
- [7] Shuster, M.D. (1993). A survey of attitude representations. *Journal of the Astronautical Sciences* 41(4), 439–517.
- [8] Surya, P.N. Singh y J. Waldron Kenneth (2005). Attitude estimation for dynamic legged locomotion using range and inertial sensors. En: *IEEE International Conference on Robotics and Automation, ICRA'05*.
- [9] Thienel, J. y R.M. Sanner (2003). A coupled nonlinear spacecraft attitude controller and observer with an unknown constant gyro bias and gyro noise. *IEEE Transactions on Automatic Control* 48(11), 2011–2014

Aplicación de las Ecuaciones de Poincaré-Chetaev-Rumyantsev al Modelado Dinámico de Mecanismos Robóticos

Alfredo Delgado-Spíndola
TecNM / I. T. de la Laguna,
Torreón, Coah., 27000
Email: frdesp20@gmail.com

Ricardo Campa
TecNM / I. T. de la Laguna,
Torreón, Coah., 27000
Email: recampac@correo.itlalaguna.edu.mx

Eusebio Bugarin
TecNM / I. T. de Ensenada,
Ensenada, B.C., 22780
Email: ebugarin@ite.edu.mx

Resumen—La obtención de las ecuaciones que describen la dinámica de un mecanismo robótico es de gran importancia, ya sea para fines de su diseño, simulación o control de movimiento. Aunque en la literatura existen diversos métodos para determinar estas ecuaciones, la búsqueda de una metodología general que se pueda aplicar por igual a diferentes tipos de sistemas robóticos sigue siendo un tema que atrae la atención de la comunidad científica actual. En el presente trabajo se estudian las ecuaciones de Poincaré-Chetaev-Rumyantsev (P-C-R) que se utilizan para obtener la dinámica de mecanismos con y sin restricciones cinemáticas en su modelo. Se destaca que la notación tensorial empleada originalmente por los autores del método se ha cambiado a notación matricial. Finalmente se obtiene el modelo dinámico de un sistema con restricciones no holonómicas para ejemplificar la aplicación de las ecuaciones de P-C-R.

Palabras clave—Mecanismo robótico, modelado dinámico, restricciones cinemáticas, cuasi-velocidades, ecuaciones de P-C-R.

I. INTRODUCCIÓN

En robótica se denomina mecanismo a un sistema de cuerpos rígidos conectados por medio de articulaciones [1], y se refiere específicamente a la parte mecánica de un robot. El modelado de mecanismos robóticos es importante para el control automático, ya que en la medida con que se cuente con un modelo que describa mejor la realidad las conclusiones tomadas de la teoría de control serán más efectivas. Además, la simulación del controlador en conjunto con la planta, utilizando algún sistema de cómputo que utiliza sus modelos matemáticos, es un paso previo a la implementación en el sistema real que rara vez se omite.

Supóngase que para describir la configuración de un mecanismo robótico con n grados de libertad (gdl) se emplean $m \geq n$ variables generalizadas. Si $m = n$ entonces esas m variables de configuración son suficientes para describir el modelo del sistema y este sería un modelo mínimo, sin restricciones cinemáticas; además, en caso de que tal modelo requiera derivadas temporales hasta de orden l de las m variables de configuración (como es el caso del modelo cinemático de velocidad donde $l = 1$, o el modelo dinámico

donde $l = 2$), entonces se podrían emplear ml variables de estado (y, por tanto, ml ecuaciones diferenciales de primer orden o *ecuaciones de estado*) para describir tal sistema.

Pero en caso de que $m > n$, el modelo obtenido ya no será mínimo (en el sentido de que no será descrito por un número mínimo de variables), y necesariamente se presentan $r > 0$ restricciones cinemáticas (relaciones entre las ml variables de estado), las cuales pueden ser holonómicas o no holonómicas dependiendo de si pueden ser integradas respecto al tiempo o no, respectivamente, para obtener restricciones en función únicamente de las coordenadas generalizadas. Además, las ml ecuaciones de estado que se utilicen no serían independientes, pues deberían de satisfacer las r restricciones, de modo que ahora se llegaría a un modelo descrito por un sistema de ecuaciones algebro-diferenciales.

Si el sistema solo tiene r_h restricciones holonómicas, entonces $r = r_h = \bar{r}$, donde

$$\bar{r} = (m - n)(l + 1). \quad (1)$$

Si el sistema tiene r_h restricciones holonómicas y r_{nh} restricciones no holonómicas entonces $r = r_h + r_{nh} < \bar{r}$.

Existen diferentes metodologías para obtener el modelo dinámico de un mecanismo robótico, siendo las más comunes las de Newton-Euler, Euler-Lagrange y Hamilton. La aplicación de tales metodologías al modelado de mecanismos sin restricciones es directa y lleva a un modelo mínimo; pero en el caso de mecanismos con restricciones se debe emplear además algún procedimiento para llegar a un modelo dinámico mínimo que satisfaga las restricciones propias del sistema. En la literatura se pueden encontrar dos procedimientos generales para esto:

- (A) Primero aplicar las ecuaciones de movimiento de la formulación seleccionada para obtener un modelo dinámico no mínimo y luego usar algún método matemático para reducir la dimensión del sistema, eliminando las restricciones.
- (B) Primero usar algún método matemático para reducir la dimensión del sistema, eliminando las restricciones, y luego aplicar las ecuaciones de movimiento de la formulación seleccionada para obtener el modelo dinámico mínimo.

Este trabajo fue financiado parcialmente por el CONACyT y el Tecnológico Nacional de México (TecNM).

En [2], [3] y [4] se explica cómo aplicar el procedimiento (A) utilizando las formulaciones de Euler-Lagrange, de Hamilton y de Newton-Euler, respectivamente. El procedimiento (B) ha sido explotado más recientemente, tal como se menciona a continuación.

Aunque el primer trabajo sobre modelado de sistemas con restricciones ya tiene casi tres siglos, no fue sino hasta finales del siglo XVIII y principios del siglo XIX cuando el tema atrajo a un número mayor de investigadores, los cuales propusieron sus propias ecuaciones para obtener el modelo de sistemas con restricciones; tal es el caso de las ecuaciones de Volterra, Maggi, Hamel, Poicaré-Chetaev y Kane entre otros (ver [5], [6] y [7]).

Durante la década de 1990, V. V. Rumyantsev extendió las ecuaciones de Poincaré-Chetaev (un método general para la obtención de ecuaciones de sistemas restringidos) a una clase mayor de sistemas no holonómicos (ver [8]–[10]). Las ecuaciones resultantes son conocidas como las ecuaciones de Poincaré-Chetaev-Rumyantsev (P-C-R) y, como el mismo Rumyantsev menciona, tales ecuaciones son una formulación más general para obtener el modelo dinámico de mecanismos robóticos, en el sentido de que se aplica por igual a sistemas sin restricciones y con restricciones (holonómicas o no holonómicas); además se puede demostrar que otras formulaciones, como las de Euler-Lagrange, Maggi, Kane, Hamel, etc., se pueden obtener a partir de las ecuaciones de P-C-R [5].

El principal inconveniente de los trabajos publicados por Rumyantsev para cierta comunidad dentro de la robótica, es el uso de nomenclatura tensorial. En ese sentido, cabe destacar que en este documento se presentan las ecuaciones de P-C-R utilizando notación matricial para el caso cuando se tienen restricciones independientes del tiempo (aunque en el trabajo publicado por Rumyantsev también pueden considerarse restricciones que dependen del tiempo).

Lo que resta de este documento se encuentra organizado de la siguiente manera: en la sección II se realiza una revisión de los conceptos básicos para el modelado de mecanismos robóticos. La sección III explica las ecuaciones de movimiento de P-C-R y en la sección IV se aplican las mismas para modelar un sistema sin restricciones y otro con restricciones no holonómicas. Por último, en la sección V se presentan las conclusiones del trabajo.

II. CONCEPTOS BÁSICOS

A. Coordenadas generalizadas y grados de libertad

Se le llama *configuración* de un sistema mecánico a la ubicación en el espacio físico de todos los elementos del sistema en un instante de tiempo dado. Para describir la configuración de un sistema mecánico se pueden utilizar una serie de variables que determinen de manera única la ubicación de todos los puntos del mismo. Dichas variables se conocen en la literatura como *coordenadas generalizadas*.

Si para describir un sistema se utilizan m coordenadas generalizadas, las cuales se denotan como $\rho_1, \rho_2, \dots, \rho_m$, entonces

es posible definir un *vector de coordenadas generalizadas* ρ , tal que

$$\rho = [\rho_1 \quad \rho_2 \quad \dots \quad \rho_m]^T \in \mathbb{R}^m$$

donde \mathbb{R}^m representa el espacio de configuración del sistema y, por lo tanto, cada elemento del mismo corresponde a una configuración específica.

Para un mismo sistema es posible emplear diferentes conjuntos de coordenadas generalizadas (incluso con un número diferente de coordenadas). Y, en general, dados dos conjuntos de coordenadas generalizadas que describen el mismo sistema, debe ser posible expresar un conjunto en términos del otro (o, lo que es lo mismo, definir una transformación de coordenadas generalizadas).

Además, para describir la configuración de un sistema, es común emplear conjuntos de coordenadas generalizadas en las que cada coordenada es independiente de las otras. De hecho, algunos autores afirman que las coordenadas generalizadas deben ser independientes por definición. Sin embargo, en este trabajo se tomará el enfoque de [11], quien habla de *coordenadas generalizadas independientes* (o *mínimas*) y *coordenadas generalizadas dependientes* (*redundantes*, o *no mínimas*).

Entonces el número de grados de libertad de un mecanismo robótico es el número de variables *independientes* que determinan su configuración [12]. Y como las coordenadas generalizadas son justamente las que describen tal configuración, el número de grados de libertad es también el número *mínimo* de coordenadas generalizadas que se requieren en el sistema. Si un sistema tiene n gdl, entonces se pueden emplear n coordenadas generalizadas independientes, denotadas como q_1, q_2, \dots, q_n , y es posible definir un *vector de coordenadas generalizadas independientes*, tal que

$$q = [q_1 \quad q_2 \quad \dots \quad q_n]^T \in \mathbb{R}^n.$$

B. Restricciones cinemáticas

Sea n el número de grados de libertad de un sistema y m el número de coordenadas generalizadas empleadas para describir su configuración. Si $m > n$ entonces deben existir $m - n$ restricciones de configuración.

Las restricciones de configuración se conocen también como *restricciones holonómicas* y un sistema que posee este tipo de restricciones se denomina *sistema holonómico*. En general, una restricción holonómica puede expresarse mediante una ecuación de la forma $\alpha(\rho) = 0$, de manera que las $m - n$ restricciones holonómicas se pueden agrupar en un *vector de restricciones holonómicas* de la forma

$$\alpha(\rho) = [\alpha_1(\rho) \quad \alpha_2(\rho) \quad \dots \quad \alpha_{m-n}(\rho)]^T = \mathbf{0} \in \mathbb{R}^{m-n}. \quad (2)$$

Si se deriva (2) con respecto al tiempo pueden obtenerse también restricciones que involucran a las velocidades y aceleraciones generalizadas:

$$\dot{\alpha}(\rho, \dot{\rho}) = \mathbf{0} \in \mathbb{R}^{m-n}, \quad (3)$$

$$\ddot{\alpha}(\rho, \dot{\rho}, \ddot{\rho}) = \mathbf{0} \in \mathbb{R}^{m-n}. \quad (4)$$

Debe notarse aquí que las ecuaciones (2)-(4) representan el conjunto de $r = \bar{r}$ restricciones holonómicas dado por (1), ya que $l = 2$.

Existen sistemas que tienen restricciones en el espacio de las aceleraciones y pueden tenerlas o no en el espacio de las velocidades, pero no tienen restricciones en el espacio de configuración. Tales restricciones se conocen como *restricciones no holonómicas* y los sistemas que las poseen son *sistemas no holonómicos*.

En general existen dos tipos de restricciones no holonómicas, las cuales son explicadas a continuación. Una *restricción no holonómica de primer orden* es aquella que se puede expresar en la forma

$$\beta(\rho, \dot{\rho}) = \mathbf{0}, \quad (5)$$

pero no es posible integrarla para obtener una restricción que dependa sólo de ρ (o, en otras palabras, no es posible escribir $\beta(\rho, \dot{\rho})$ como la derivada temporal de una función $\alpha(\rho)$).

Una *restricción no holonómica de segundo orden* es una ecuación de la forma

$$\gamma(\rho, \dot{\rho}, \ddot{\rho}) = \mathbf{0}$$

que no es integrable (no es posible escribir $\gamma(\rho, \dot{\rho}, \ddot{\rho})$ como la derivada temporal de una función $\beta(\rho, \dot{\rho})$). Tales sistemas tienen un total de $r < \bar{r}$ restricciones.

C. Cuasi-velocidades

Las velocidades de los cuerpos de un sistema para cierta configuración y cierto instante de tiempo pueden ser especificados ya sea por las velocidades generalizadas o, de forma más general, por las llamadas *características cinemáticas* o *cuasi-velocidades* (que son una combinación lineal de las velocidades generalizadas). Un ejemplo de cuasi-velocidades son las expresiones para calcular la velocidad angular $\omega = [\omega_x \omega_y \omega_z]^T \in \mathbb{R}^3$ a partir de una convención dada de los ángulos de Euler (ϕ, ϑ, ψ) y sus derivadas temporales $(\dot{\phi}, \dot{\vartheta}, \dot{\psi})$, (si se emplea la convención ZYZ de ángulos de Euler [13]); entonces las componentes de la velocidad angular son:

$$\begin{aligned} \omega_x &= -\dot{\vartheta} \sin(\phi) + \dot{\psi} \sin(\vartheta) \sin(\phi), \\ \omega_y &= \dot{\vartheta} \cos(\phi) + \dot{\psi} \sin(\vartheta) \cos(\phi), \\ \omega_z &= \dot{\phi} + \dot{\psi} \cos(\vartheta). \end{aligned}$$

En este caso las velocidades generalizadas serían $\dot{\phi}, \dot{\vartheta}$ y $\dot{\psi}$, mientras que ω_x, ω_y y ω_z serían las cuasi-velocidades, ya que no son la derivada con respecto al tiempo de ninguna de las coordenadas generalizadas. En forma general las cuasi-velocidades pueden ser escritas como

$$\nu_i = f_i(\rho, \dot{\rho}). \quad (6)$$

Si las cuasi-velocidades (6) son integrables, las coordenadas generalizadas que determinan la configuración del sistema pueden ser encontradas simplemente integrando ν_i con res-

pecto al tiempo. Por otro lado, si las cuasi-velocidades son no integrables, se utilizan las variables

$$\pi_i = \int_0^t \nu_i dt,$$

que tienen una función similar a las coordenadas generalizadas, pero ya no son función de la configuración del sistema. Las variables π_i son conocidas como cuasi-coordenadas [14].

III. ECUACIONES DE POINCARÉ-CHETAEV-RUMYANTSEV

Las ecuaciones de P-C-R son una extensión a las ecuaciones propuestas por Poincaré y Chetaev. Las primeras ecuaciones fueron propuestas por Poincaré, quien propuso un método para obtener las ecuaciones de sistemas holonómicos utilizando coeficientes de estructura, los cuales eran constantes. Posteriormente, Chetaev extendió el trabajo de Poincaré al caso en el que los coeficientes de estructura son variables y además las restricciones podrían ser ciertos tipos de restricciones no holonómicas; a estas ecuaciones se les conoce como ecuaciones de Poincaré-Chetaev. En sus trabajos, Rumyantsev extiende las ecuaciones de Poincaré-Chetaev al caso en el que las restricciones no holonómicas pueden ser no lineales, con lo cual sus ecuaciones se convierten en una herramienta general para el modelado de sistemas mecánicos.

Supóngase un sistema mecánico de n grados de libertad bajo r restricciones dadas por ecuaciones diferenciales, ya sea la derivada de las restricciones holonómicas (2) o restricciones no holonómicas de primer orden (5), las cuales pueden ser agrupadas en un vector de la forma

$$\nu_\beta = f_\beta(\rho, \dot{\rho}) = \mathbf{0} \in \mathbb{R}^r; \quad (7)$$

además, se seleccionan n velocidades arbitrarias (aunque en general se toman la velocidades independientes), las cuales pueden ser agrupadas en un vector de velocidades dado por

$$\nu_q = f_q(\rho, \dot{\rho}) \in \mathbb{R}^n. \quad (8)$$

Agrupando lo vectores (7) y (8) es posible formar el vector de cuasi-velocidades el cual está dado por

$$\nu = f(\rho, \dot{\rho}) = \begin{bmatrix} \nu_q \\ \nu_\beta \end{bmatrix} \in \mathbb{R}^m. \quad (9)$$

En muchos sistemas mecánicos es posible escribir (9) en la forma

$$\nu = S(\rho)\dot{\rho}, \quad (10)$$

donde $S(\rho) \in \mathbb{R}^{m \times m}$, y si las velocidades arbitrarias (8) se seleccionan adecuadamente es posible asegurar que $S(\rho)$ sea invertible; y en tal caso se puede obtener

$$\dot{\rho} = T(\rho)\nu, \quad (11)$$

donde $T(\rho) = S(\rho)^{-1}$.

Ahora, es necesario encontrar la energía cinética $\mathcal{K}_i(\rho, \dot{\rho})$ y la energía potencial $\mathcal{U}_i(\rho)$ de cada i -ésimo cuerpo ($i =$

$1, 2, \dots, N$) en función de las coordenadas y velocidades generalizadas, las cuales están definidas como [15]:

$$\begin{aligned}\mathcal{K}_i(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) &= \frac{1}{2} [m_i \mathbf{v}_i^T \mathbf{v}_i + \boldsymbol{\omega}_i^T I_i \boldsymbol{\omega}_i], \\ \mathcal{U}_i(\boldsymbol{\rho}) &= -m_i \mathbf{p}_i^T \mathbf{g}_0,\end{aligned}$$

donde para cada i -ésimo cuerpo \mathbf{v}_i y $\boldsymbol{\omega}_i$ son los vectores de velocidad lineal y angular, respectivamente; \mathbf{p}_i es el vector de posición, \mathbf{g}_0 es el vector de gravedad, m_i es la masa e I_i es la matriz de momentos de inercia. Entonces, la función lagrangiana $\mathcal{L}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}})$ está dada por la energía cinética total menos la energía potencial total; es decir,

$$\mathcal{L}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) = \mathcal{K}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) - \mathcal{U}(\boldsymbol{\rho})$$

donde

$$\mathcal{K}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}) = \sum_{i=1}^N \mathcal{K}_i(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}), \quad \mathcal{U}(\boldsymbol{\rho}) = \sum_{i=1}^N \mathcal{U}_i(\boldsymbol{\rho})$$

Una vez encontrado el lagrangiano se utiliza (11) para hacer un cambio de variables y con esto se define un nuevo lagrangiano en función de las cuasi-velocidades dado por

$$\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\nu}) = \mathcal{K}(\boldsymbol{\rho}, \boldsymbol{\nu}) - \mathcal{U}(\boldsymbol{\rho}).$$

Además del calculo del lagrangiano, para la obtención de las ecuaciones de P-C-R, es necesario obtener los coeficientes de estructura; dichas variables se obtienen utilizando

$$R_{(j,k)i} = \sum_{t=1}^m \sum_{s=1}^m \left(\frac{\partial S_{(j,t)}}{\partial \rho_s} - \frac{\partial S_{(j,s)}}{\partial \rho_t} \right) T_{(t,k)} T_{(s,i)}, \quad (12)$$

$$k, i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m,$$

donde $S_{(a,b)}$ y $T_{(a,b)}$ son los elementos el renglón a y la columna b de las matrices $S(\boldsymbol{\rho})$ y $T(\boldsymbol{\rho})$, respectivamente. Además los coeficientes cumplen con $R_{(j,k)i} = -R_{(j,i)k}$ [10]. Dichos coeficientes son los elementos $R_{(\cdot, \cdot)i}$ de una serie de matrices $R_i(\boldsymbol{\rho}) \in \mathbb{R}^{m \times n}$, las cuales permiten formar el vector de términos de no holonomía dado por

$$\boldsymbol{\nu}_{nh} = \begin{bmatrix} \boldsymbol{\nu}_q^T R_1^T \frac{\partial \mathcal{L}}{\partial \boldsymbol{\nu}} \\ \boldsymbol{\nu}_q^T R_2^T \frac{\partial \mathcal{L}}{\partial \boldsymbol{\nu}} \\ \vdots \\ \boldsymbol{\nu}_q^T R_n^T \frac{\partial \mathcal{L}}{\partial \boldsymbol{\nu}} \end{bmatrix} \in \mathbb{R}^n.$$

Por último las ecuaciones de P-C-R, utilizando cuasi-velocidades, están dadas por

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\nu})}{\partial \boldsymbol{\nu}_q} \right) - A(\boldsymbol{\rho})^T \frac{\partial \mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\nu})}{\partial \boldsymbol{\rho}} = \boldsymbol{\nu}_{nh} + A(\boldsymbol{\rho})^T \boldsymbol{\tau}_\rho \in \mathbb{R}^n \quad (13)$$

donde la matriz $A(\boldsymbol{\rho})$ se define como

$$A(\boldsymbol{\rho}) = \begin{bmatrix} \frac{\partial \dot{\boldsymbol{\rho}}}{\partial \boldsymbol{\nu}_q} \end{bmatrix} \in \mathbb{R}^{m \times n}. \quad (14)$$

Finalmente, una vez desarrolladas las ecuaciones de P-C-R, se sustituyen las ecuaciones de restricción (7); es decir, $\boldsymbol{\nu}_\beta = \mathbf{0}$ en (13).

Las ecuaciones de P-C-R (13) escritas en forma compacta son

$$M(\boldsymbol{\rho}) \dot{\boldsymbol{\nu}}_q + C(\boldsymbol{\rho}, \boldsymbol{\nu}) \boldsymbol{\nu}_q + \mathbf{g}(\boldsymbol{\rho}) - \boldsymbol{\nu}_{nh} = \boldsymbol{\tau}_{\boldsymbol{\nu}_q} \in \mathbb{R}^n$$

donde $M(\boldsymbol{\rho}) \in \mathbb{R}^{n \times n}$ es una matriz simétrica y definida positiva conocida como la matriz de inercias, $C(\boldsymbol{\rho}, \boldsymbol{\nu}) \in \mathbb{R}^{n \times n}$ es la matriz de términos generados por las fuerzas centrífugas y de Coriolis, $\mathbf{g}(\boldsymbol{\rho}) \in \mathbb{R}^n$ es el vector de fuerzas debidas a la gravedad, $\boldsymbol{\tau}_{\boldsymbol{\nu}_q}$ es el vector de fuerzas aplicadas a las q velocidades arbitrarias y $\boldsymbol{\nu}_q$ es el vector de velocidades arbitrarias.

IV. APLICACIÓN DE LAS ECUACIONES DE P-C-R

Con el fin de ejemplificar el uso de las ecuaciones de P-C-R, éstas se aplican a dos sistemas:

- Un sistema sin restricciones.
- Un sistema con restricciones no holonómicas de primer orden.

A. Sistema sin restricciones (con coordenadas mínimas).

En el caso de un robot cuyo modelo es descrito usando un número mínimo de coordenadas (por ejemplo un manipulador serial) se tienen el mismo número de coordenadas generalizadas m que grados de libertad n . Por lo tanto, el vector de coordenadas generalizadas es igual al vector de coordenadas independientes; es decir

$$\boldsymbol{\rho} = \mathbf{q} \in \mathbb{R}^{m=n},$$

por lo que no se tiene ningún tipo de restricción. Debido a esto el vector de cuasi-velocidades (9) para este caso está formado únicamente por las velocidades arbitrarias (8) y tomando $\boldsymbol{\nu}_q = \boldsymbol{\nu} = \dot{\boldsymbol{\rho}} = \dot{\mathbf{q}} \in \mathbb{R}^n$ se obtiene

$$S(\boldsymbol{\rho}) = T(\boldsymbol{\rho}) = I \in \mathbb{R}^{n \times n}.$$

Además, otra consecuencia de no tener restricciones es que todos los coeficientes de (12) se desvanecen; i.e., $R_i(\boldsymbol{\rho}) = 0$ [8].

Por otro lado, debido a que $\dot{\mathbf{q}} = \boldsymbol{\nu}_q = \dot{\boldsymbol{\rho}}$ la relación (14) se convierte en una matriz identidad, es decir

$$A(\boldsymbol{\rho}) = \begin{bmatrix} \frac{\partial \dot{\boldsymbol{\rho}}}{\partial \boldsymbol{\nu}_q} \end{bmatrix} = \frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{q}}} = I \in \mathbb{R}^{n \times n}.$$

Por último, como en este caso todas las cuasi-velocidades son iguales a las velocidades generalizadas, el lagrangiano en función de las cuasi-velocidades es igual al lagrangiano en función de las velocidades generalizadas; es decir

$$\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\nu}) = \mathcal{L}(\boldsymbol{\rho}, \dot{\boldsymbol{\rho}}).$$

Por lo que las ecuaciones de P-C-R para sistemas sin restricciones quedan como

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \boldsymbol{\tau} \in \mathbb{R}^n$$

donde $\boldsymbol{\tau}$ son las fuerzas aplicadas a cada una de las coordenadas generalizadas mínimas \mathbf{q} . Dichas ecuaciones son las ya conocidas ecuaciones de Euler-Lagrange para sistemas sin restricciones [16].

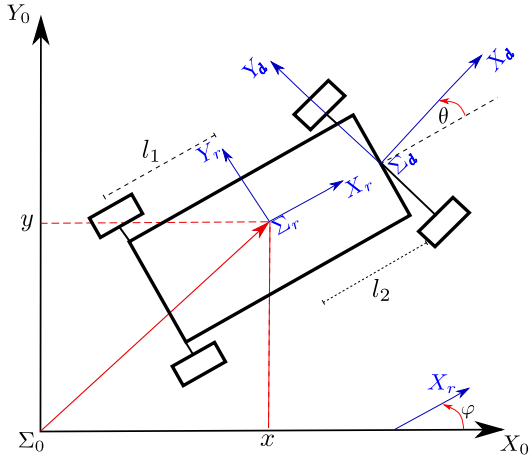


Fig. 1. Vista de planta del robot móvil

B. Sistema con restricciones no holonómicas.

Ahora se analiza un mecanismo no holonómico dado por un robot móvil con ruedas (RMR) en configuración Ackerman o tipo carro (ver Figura 1). Para simplificar el análisis se considera que tal mecanismo está formado por dos cuerpos rígidos: el chasis junto con las dos ruedas traseras sobre un eje fijo (cuerpo 1) y las dos ruedas delanteras (cuerpo 2). Se supone que estas ruedas delanteras se encuentran en un eje común y su orientación se puede ajustar para cambiar la dirección de avance del vehículo. Cabe mencionar que en [5] se analiza este mismo sistema, sólo que ahí se utiliza otra formulación para obtener el modelo dinámico.

Nótese de la figura 1 que se ha utilizado un marco fijo (inercial) $\Sigma_0(X_0, Y_0, Z_0)$, un marco unido al chasis del RMR $\Sigma_r(X_r, Y_r, Z_r)$, y un marco $\Sigma_d(X_d, Y_d, Z_d)$ que se considera unido a las ruedas delanteras, con origen en el punto medio de la distancia entre las mismas. Como el movimiento se realiza sólo en un plano (2D) la posición del RMR queda determinada por las coordenadas x y y del origen del marco Σ_r con respecto a Σ_0 ; por otra parte la orientación relativa de Σ_r respecto a Σ_0 queda determinada por el ángulo φ que es el ángulo entre el eje X_0 y X_r . Por último, θ es el ángulo de la orientación relativa entre X_r y X_d . Por lo tanto, el vector de coordenadas generalizadas no mínimas es (con $m = 4$)

$$\rho = [\varphi \quad \theta \quad x \quad y]^T \in \mathbb{R}^4$$

y el vector de velocidades generalizadas no mínimas está dado por

$$\dot{\rho} = [\dot{\varphi} \quad \dot{\theta} \quad \dot{x} \quad \dot{y}]^T \in \mathbb{R}^4,$$

El sistema se encuentra sujeto a $r = 2$ restricciones no holonómicas, las cuales son

$$\begin{aligned} -\dot{x}\sin(\varphi) + \dot{y}\cos(\varphi) - l_2\dot{\varphi} &= 0, \\ -\dot{x}\sin(\varphi + \theta) + \dot{y}\cos(\varphi + \theta) + l_1\dot{\varphi}\cos(\theta) &= 0. \end{aligned}$$

Ahora, el vector de velocidades arbitrarias (8) se elige como (usando $n = m - r = 2$):

$$\nu_q = [\dot{\varphi} \quad \dot{\theta}]^T \in \mathbb{R}^2.$$

A partir de estas definiciones se forma el vector de cuasi-velocidades (9); el cual está dado por:

$$\nu = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \end{bmatrix} = \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ -\dot{x}S_\varphi + \dot{y}C_\varphi - l_2\dot{\varphi} \\ -\dot{x}S_{\varphi+\theta} + \dot{y}C_{\varphi+\theta} + l_1\dot{\varphi}C_\theta \end{bmatrix} \in \mathbb{R}^4 \quad (15)$$

donde los símbolos S_\cdot y C_\cdot corresponden a $\sin(\cdot)$ y $\cos(\cdot)$, respectivamente. Note que el vector de cuasi-velocidades se puede reescribir como

$$\nu = S(\rho)\dot{\rho} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -l_2 & 0 & -S_\varphi & C_\varphi \\ l_1C_\theta & 0 & -S_{\varphi+\theta} & C_{\varphi+\theta} \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{x} \\ \dot{y} \end{bmatrix}, \quad (16)$$

de modo que

$$\dot{\rho} = T(\rho)\nu = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ t_{31} & 0 & t_{33} & t_{34} \\ t_{41} & 0 & t_{43} & t_{44} \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \end{bmatrix}, \quad (17)$$

donde los elementos de $T(\rho) = S(\rho)^{-1}$ son (para $\theta \neq k\pi$, con $k \in \mathbb{Z}$):

$$\begin{aligned} t_{31} &= (l_1C_\varphi C_\theta + l_2C_{\varphi+\theta})/S_\theta, \\ t_{33} &= C_{\varphi+\theta}/S_\theta, \\ t_{34} &= -C_\varphi/S_\theta, \\ t_{41} &= (l_1S_\varphi C_\theta + l_2S_{\varphi+\theta})/S_\theta, \\ t_{43} &= S_{\varphi+\theta}/S_\theta, \\ t_{44} &= -S_\varphi/S_\theta, \end{aligned}$$

La energía cinética del sistema, en función de las velocidades generalizadas no mínimas, está dada por

$$\begin{aligned} \mathcal{K}(\rho, \dot{\rho}) &= \frac{m_t}{2}(\dot{x}^2 + \dot{y}^2) + \frac{I_t}{2}\dot{\varphi}^2 + \frac{I_2}{2}\dot{\theta}^2 + I_2\dot{\varphi}\dot{\theta} \\ &\quad + m_2l_1\dot{\varphi}(-\dot{x}S_\varphi + \dot{y}C_\varphi), \end{aligned} \quad (18)$$

donde I_i y m_i son las masas e inercias del i -ésimo cuerpo (con $i = 1, 2$), $m_t = m_1 + m_2$ e $I_t = I_1 + I_2 + m_2l_1^2$. Por otro lado, debido a que el movimiento se realiza en un plano horizontal, la energía potencial es igual a cero $\mathcal{U}(\rho) = 0$. Por lo que el lagrangiano sólo depende de la energía cinética, es decir

$$\mathcal{L}(\rho, \dot{\rho}) = \mathcal{K}(\rho, \dot{\rho}).$$

Tomando de (17) las velocidades generalizadas en función de las cuasi-velocidades $\dot{\rho} = \mathbf{f}(\rho, \nu)$ y sustituyéndolas en (18) se obtiene el nuevo lagrangiano, el cual queda dado por

$$\begin{aligned} \mathcal{L}(\rho, \nu) &= \frac{I_2\nu_2^2}{2} + \frac{I_t\nu_1^2}{2} + \frac{m_t}{2S_\theta^2}\tilde{k}_1 + \frac{m_t}{2S_\theta^2}\tilde{k}_2 \\ &\quad + I_2\nu_1\nu_2 + \frac{l_1m_2\nu_1}{S_\theta}\tilde{k}_3, \end{aligned}$$

donde \tilde{k}_i , con $i = 1, 2, 3$, son variables auxiliares dadas por

$$\begin{aligned}\tilde{k}_1 &= (\nu_3 C_\varphi S_\theta - \nu_4 S_\varphi + C_\theta S_\varphi (\nu_3 + l_1 \nu_1) + 2l_2 \nu_1 C_\varphi S_\varphi)^2, \\ \tilde{k}_2 &= (\nu_3 C_{\varphi+\theta} - \nu_4 C_\varphi + l_2 \nu_1 C_{\varphi+\theta} + l_1 \nu_1 C_\varphi C_\theta)^2, \\ \tilde{k}_3 &= \nu_3 S_\theta + \frac{l_2 \nu_1 S_\varphi}{2} + \frac{l_2 \nu_1 S_\theta}{2} - \frac{l_2 \nu_1 S_{2\varphi+\theta}}{2} + \frac{l_2 \nu_1 S_{3\varphi}}{2}.\end{aligned}$$

Una vez obtenido el lagrangiano es necesario calcular las matrices $R(\rho)_i$ utilizando los elementos de las matrices (17) y (16) en la ecuación (12) y con las cuales se obtiene

$$R(\rho)_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & r_r \end{bmatrix}, \quad R(\rho)_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -r_r & 0 \end{bmatrix},$$

donde

$$\begin{aligned}r_r &= l_1 S_\theta + \frac{S_{\varphi+\theta} (l_2 S_{2\varphi} + l_1 C_\theta S_\varphi)}{S_\theta} \\ &+ \frac{C_{\varphi+\theta} (l_2 C_{\varphi+\theta} + l_1 C_\varphi C_\theta)}{S_\theta}.\end{aligned}$$

Por último, la matriz (14) resulta

$$A(\rho) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \frac{l_2 C_{\varphi+\theta} + l_1 C_\varphi C_\theta}{S_\theta} & 0 \\ \frac{l_2 S_{2\varphi} + l_1 C_\theta S_\varphi}{S_\theta} & 0 \end{bmatrix}.$$

Utilizando las ecuaciones de P-C-R (13), agrupando los términos multiplicados por $\dot{\nu}$, ν y posteriormente sustituyendo las restricciones del sistemas, es decir

$$\nu_3 = 0 \quad \nu_4 = 0,$$

es posible encontrar el modelo dinámico en forma compacta que está dado por

$$M(\rho)\dot{\nu}_q + C(\rho, \nu)\nu_q + g(\rho) - \nu_{nh} = \tau_{\nu_q} \in \mathbb{R}^2,$$

con $\nu_q = [\nu_1 \quad \nu_2]^T$,

$$\begin{aligned}M(\rho) &= \begin{bmatrix} m_{11} & I_2 \\ I_2 & I_2 \end{bmatrix}, \quad C(\rho, \nu) = \begin{bmatrix} -\nu_2 c & -\nu_1 c \\ \nu_1 c & 0 \end{bmatrix}, \\ g(\rho) &= \mathbf{0}, \quad \nu_{nh} = \begin{bmatrix} \nu_1 \nu_2 c \\ \nu_1^2 c \end{bmatrix},\end{aligned}$$

donde

$$\begin{aligned}m_{11} &= (m_t (l_1^2 + l_2^2) + I_t S_\theta^2 + 2l_1 l_2 m_t - l_1^2 m_t S_\theta^2 \\ &+ 2l_1 l_2 m_2 S_\theta^2 - 2l_1 l_2 m_t S_\theta^2) / S_\theta^2, \\ c &= m_t C_\theta (l_1 + l_2)^2 / S_\theta^3,\end{aligned}$$

Por último, el vector de fuerzas generalizadas está dado por

$$\tau_{\nu_q} = [\tau_\varphi \quad \tau_\theta]^T$$

donde τ_φ es el par aplicado al giro del chasis del robot y τ_θ es el par aplicado al giro del eje delantero del RMR.

V. CONCLUSIONES

En el presente trabajo se describen las ecuaciones de movimiento de Poincaré-Chetaev-Rumyantsev utilizando casi-velocidades, las cuales son una formulación general para la obtención de las ecuaciones de la dinámica de mecanismos robóticos con y sin restricciones. Se destaca que se empleó notación matricial en lugar de la notación tensorial del trabajo original de Rumyantsev. Al final se muestra la relación entre las ecuaciones de P-C-R y de Euler-Lagrange para sistemas sin restricciones y se aplican las ecuaciones de P-C-R a un sistema con restricciones no holonómicas para ejemplificar su uso.

REFERENCIAS

- [1] K. Waldron and J. Schmiedeler, "Kinematics", in Springer Handbook of Robotics, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 9–33.
- [2] F. H. Ghorbel, O. Chételat, R. Gunawardana, and R. Longchamp, "Modeling and set point control of closed-chain mechanisms: Theory and experiment", IEEE Transactions on Control Systems Technology, vol. 8, pp. 801–815, 2000.
- [3] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach. Springer, 2009.
- [4] J. Angeles and S. K. Lee, "The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement", Journal of Applied Mechanics, vol. 55, p. 243–244, 1988.
- [5] S. K. Soltakhanov, M. Yushkov, and S. Zegzhda, Mechanics of Non-holonomic Systems: A New Class of Control Systems. Springer, 2009.
- [6] T. R. Kane, "Dynamics of nonholonomic systems", Journal of Applied Mechanics, vol. 28, pp. 574–578, 1961.
- [7] N. Chetaev, "On the equations of Poincaré", Journal of Applied Mathematics and Mechanics, vol. 5, pp. 253–262, 1941.
- [8] V. Rumyantsev, "On the Poincaré-Chetaev equations", Journal of Applied Mathematics and Mechanics, vol. 58, no. 3, pp. 373–386, 1994.
- [9] V. Rumyantsev, "The general equations of analytical dynamics", Journal of Applied Mathematics and Mechanics, vol. 60, no. 6, pp. 899–909, 1996.
- [10] V. Rumyantsev, "On the Poincaré and Chetaev equations," Journal of Applied Mathematics and Mechanics, vol. 62, no. 4, pp. 495–502, 1998.
- [11] H. Baruh, Analytical Dynamics. McGraw-Hill, 1999.
- [12] N. Chetaev, Theoretical Mechanics Springer, 1989.
- [13] J. J. Craig, Introduction to Robotics, Mechanics and Control. Pearson Prentice-Hall, 2005.
- [14] J. I. Neimark and N. A. Fufaev, Dynamics of Nonholonomic Systems. American Mathematical Society, 1967.
- [15] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, Robotics: Modeling, Planning and Control. Springer, 2009.
- [16] R. Kelly and V. Santibáñez, Control de Movimiento de Robots Manipuladores. Pearson-Prentice Hall, 2003.

Seguimiento de Trayectoria de un Robot Móvil Omnidireccional en Plano Inclinado

Armando Sáenz, Tadeo Espinoza
 Facultad de Ingeniería, Ciencias y Arquitectura
 Universidad Juárez del Estado de Durango
 Gómez Palacio, Durango
 Email: jsaenz@ujed.mx, atespinoza@ujed.mx

Eusebio Bugarin
 Tecnológico Nacional de México
 Instituto Tecnológico de Ensenada
 Ensenada, Baja California
 Email: ebugarin@ite.edu.mx

Resumen—El artículo analiza el seguimiento de trayectoria para un robot móvil omnidireccional que se desplaza sobre un plano inclinado. El estudio es para robots móviles que estén equipados con cuatro ruedas mecanum. La trayectoria a realizar es sobre un plano inclinado con una pendiente constante. Lo anterior implica la existencia del vector de gravedad en el modelo dinámico. El modelado se realizó por el método de Euler-Lagrange. La ley de control utilizada es de aceleración resuelta en complemento con un PI de velocidad. SE demuestra estabilidad por el método directo de Lyapunov.

Palabras Clave: Robot Móvil con Ruedas Omnidireccional, Ley de control de aceleración resulta, Sistemas lineales, Sistemas no lineales, Euler-Lagrange, Cinemática, Dinámica

I. INTRODUCCIÓN

El primer Robot Móvil con Ruedas (RMR) fue construido por Mac Barret en el año de 1954 [1], el RMR fue nombrado como *Guide-O-Matic*. El RMR se desplaza por una nave industrial siguiendo un cable e iba deteniéndose en distintos puntos. En cada punto los trabajadores podían tomar o dejar herramientas. Al día de hoy, el uso de los RMR se ha ampliado a exploración, asistencia personal, transporte de carga y transporte de personal por mencionar algunos.

Un RMR se puede definir como una máquina electromecánica con la capacidad de desplazarse libremente sobre un terreno regular o irregular. La capacidad de desplazamiento va a depender del tipo de ruedas con las que este equipado. La característica más importante que posee un RMR es la capacidad de realizar movimientos autónomos (sin interacción humana), utilizando sus motores (en su mayoría de corriente directa) que son accionados por una computadora a bordo [2]. Las ruedas se pueden clasificar en dos tipos: las primeras son las convencionales (e.g. fijas, orientables y de castor) y las segundas son las ruedas especiales (e.g. universales, suecas y esféricas) [3]. Cada una de las ruedas tienen restricciones de movimiento diferentes que deben de ser consideradas para realizar un adecuado movimiento del RMR.

Un RMR puede ser considerado como un sistema robótico que puede ser dividido en 3 subsistemas más pequeños [4]. Los subsistemas son: 1) *Etapas de potencia*, es donde se recibe la primera señal de control y es la encargada de entregar la corriente necesaria para realizar el movimiento deseado. 2) *Actuadores*, la corriente entregada por la etapa de potencia es

inyectada a los actuadores que convierten la energía eléctrica en mecánica ocasionando que las ruedas comiencen a girar. 3) *Configuración cinemática*, este es el último subsistema y el cual traduce el giro de las ruedas en un desplazamiento o cambio de orientación del robot, pudiéndose generar uno o ambos dependiendo de la velocidad a la que giren las ruedas y la disposición que tengan las mismas. Las disposiciones o configuraciones más importantes son: unicycle, triciclo, Ackerman o tipo coche, diferencial, omnidireccional y sked steer.

En el área de control automático se diseñan leyes de control para calcular el voltaje de armadura, par de los motores o la velocidad a la que deben de girar las ruedas para completar una tarea. Algunas de las tareas que para un RMR son: la navegación, punto a punto, persecución pura, seguimiento de línea o de carretera. Estas son algunas de las tareas que puede realizar un RMR más no son todas. Para una literatura más amplia respecto a la variedad de tareas que puede realizar un RMR se recomiendan las siguientes referencias [1], [3], [5], [6].

Algunas investigaciones sobre el diseño de leyes de control para RMR son mencionadas a continuación. En [7] se presenta un estado del arte de los robots móviles omnidireccionales hasta el año de 1998. Posteriormente, en [8] se propone una ley de control que cancela las fuerzas internas para un mejor desempeño del robot. Al mismo tiempo, [9] se presenta una ley de control cinemática para seguimiento de trayectorias asegurando acotamiento de las velocidades. Años más tarde, en [10] se reporta una ley de control basada en modos deslizantes considerando que los actuadores (motores de corriente directa) están en modo corriente y después en [11] la teoría es extendida para utilizar los actuadores en modo voltaje. En [12], donde se diseña una ley de control por lógica difusa para el caso de seguimiento de postura utilizando las velocidades de las ruedas como señal de control. En [13] se aborda el problema de seguimiento de trayectoria diseñando una ley de control por modos deslizantes; en [14] se estudia el desempeño de cuatro esquemas de control predictivo para seguimiento de trayectoria. En [15] se propone una ley de control para cada subsistema del robot (i.e. cinemática y dinámica con actuadores) demostrando estabilidad de cada uno por separado. En [16] se presenta un control servo-visual

basado en imagen de un RMR omnidireccional incluyendo dinámica de actuadores.

La aportación de este artículo es el estudio de un seguimiento de trayectoria para un RMR omnidireccional en un plano inclinado. El modo tradicional de un RMR se hace sobre un plano perpendicular a la gravedad lo que tiene como consecuencia una energía potencial constante por no hace un cambio de altura. Sin embargo, en un plano inclinado la gravedad se puede descomponer habiendo un componente absorbida por el suelo y la otra teniendo un efecto sobre el RMR. Se implementa una ley de control por aceleración resuelta en complemento con un PI para lograr el objetivo de control. Se realiza una prueba de estabilidad por el método directo de Lyapunov [17].

Las secciones del artículo se mencionan a continuación. La sección II esta conformada por la subsecciones de modelado cinemático y modelado dinámico. En la cinemática se presentan los modelos de velocidad de postura y de configuración [2]. En la dinámica se encuentran el calculo de las energías cinéticas y potencial del RMR, posteriormente se aplican las ecuaciones de movimiento de Lagrange y obtener el modelo dinámico. La sección III muestra la ley de control implementada y su prueba de estabilidad. En la sección IV son mostrados la trayectoria a realizar y los resultados de simulación. Por último, la sección de conclusiones es la número V.

Nota: Las funciones trigonométricas $\sin(\epsilon)$ y $\cos(\epsilon)$ serán representadas como s_ϵ y c_ϵ , respectivamente. La representación $I_{n \times n}$ se refiere a la matriz identidad de dimensión n .

II. FORMULACIÓN DEL PROBLEMA

La cinemática de un RMR son las expresiones que sirven para conocer el vector de velocidad al que puede acceder un RMR y la velocidad que deben de tener las ruedas para lograr dicha velocidad. En el caso de un robot móvil omnidireccional el vector de velocidad no esta restringido. La dinámica relaciona el par aplicado a las ruedas, la orientación y la velocidad RMR para obtener el vector de aceleración. La metodología implementada es tomada de [2].

II-A. Cinemática

Considere un RMR equipado con cuatro ruedas mecanum como el mostrado en la Figura 1, el cual es ubicado en un plano inclinado. Existen tres marcos: el primero es el marco inercial llamado Σ_W , el segundo es el marco Σ_B que es coincidente con Σ_W y esta rotado un ángulo β respecto a y_W consiguiendo que el plano $x_B - z_B$ este alineado con el plano inclinado, y el tercer marco es el Σ_R que esta fijado al RMR y cuyo origen coincide con el centro de masa.

Sea $P^B \triangleq [x \ y]^T$ con $x, y \in \mathbb{R}$ el vector que describe el origen de Σ_R respecto a Σ_B y $\theta \in \mathbb{R}$ es el ángulo formado entre x_B y x_R visto desde z_B . La postura del RMR (i.e.

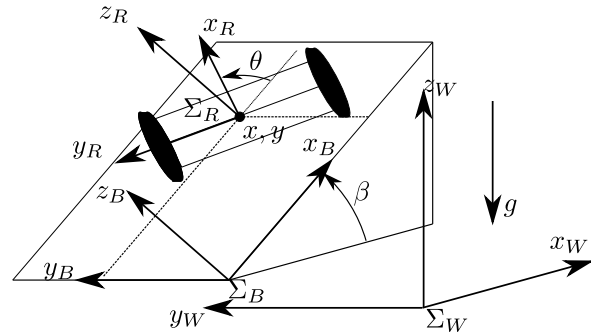


Figura 1. RMR en un plano inclinado

posición y orientación) es representada por $\xi^B \in \mathbb{R}^3$ y la velocidad por $\dot{\xi}^B \in \mathbb{R}^3$ que son definidas como:

$$\xi \triangleq \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad \dot{\xi} \triangleq \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}. \quad (1)$$

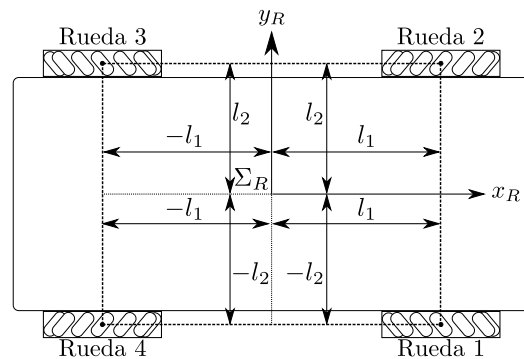


Figura 2. Posición de las ruedas en el RMR

Sean $\eta_1, \eta_2 \in \mathbb{R}$ las componentes de velocidad lineal x_R e y_R , respectivamente. La velocidad angular es representada por $\eta_3 \in \mathbb{R}$ medida desde z_R . La matriz $R_R^B(\theta) : \mathbb{R} \rightarrow \mathbb{R}^{3 \times 3}$ es la matriz de rotación desde el marco Σ_R a Σ_B y se define como:

$$R_R^B(\theta) = \begin{bmatrix} c_\theta & -s_\theta & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

Siendo $\eta \triangleq [\eta_1 \ \eta_2 \ \eta_3]^T$ la velocidad del carro vista desde Σ_R es multiplicada por la matriz de rotación $R_R^B(\theta)$ para obtener $\dot{\xi}$:

$$\dot{\xi} = R_R^B(\theta)\eta. \quad (3)$$

Se define $\varphi_i \in \mathbb{R}$ con $i = 1, 2, 3, 4$ al ángulo de giro de la rueda i , $\dot{\varphi}_i$ la velocidad rotacional de la rueda i , r es el radio de las ruedas, l_1 y l_2 es la distancia a cada una de las ruedas a los largo de x_R y y_R (véase Figura 2), respectivamente. El modelo cinemático de que relaciona η con $\dot{\varphi}$ es

$$\dot{\varphi} \triangleq \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \\ \dot{\varphi}_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & 1 & l_1 + l_2 \\ 1 & -1 & -(l_1 + l_2) \\ 1 & 1 & -(l_1 + l_2) \\ 1 & -1 & l_1 + l_2 \end{bmatrix} \eta = E\eta. \quad (4)$$

Al juntar (3) con (4) se obtiene

$$\dot{\mathbf{q}} \triangleq \begin{bmatrix} \dot{\xi} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} R_R^B(\theta) \\ E \end{bmatrix} \boldsymbol{\eta}. \quad (5)$$

El modelo (5) es conocido como *modelo cinemático de configuración* [2].

II-B. Dinámica

La formulación a utilizar para obtener el modelo dinámico es Euler-Lagrange es la descrita en [2].

El plano de movimiento del RMR tiene un ángulo β respecto al eje x_W (véase Figura 1). La orientación del plano coincide con la del marco Σ_B y la dinámica es obtenida respecto a este marco. El RMR esta compuesto por 5 cuerpos que son el cuerpo principal (R) y las cuatro ruedas (R_1, R_2, R_3 y R_4). Se define \mathbf{q} de la siguiente manera:

$$\mathbf{q} \triangleq \begin{bmatrix} \xi \\ \varphi \end{bmatrix}. \quad (6)$$

Los parámetros que intervienen son : $m \in \mathbb{R}$ en [kg] $\mathcal{I}_R = \text{diag} \{I_{xR}, I_{yR}, I_{zR}\}$ es el tensor de inercia del cuerpo principal en [kg-m²], m_r es la masa de una la rueda en [kg] y $\mathcal{I}_r = \text{diag} \{I_{xr}, I_{yr}, I_{zr}\}$ es el tensor de inercia de las ruedas en [kg-m²]. Se considera que $I_{xr} = I_{zr}$.

Definiendo $\mathcal{U}_i \in \mathbb{R}$ con $i = R, R_1, R_2, R_3, R_4$ como la energía potencial del cuerpo i . Las expresiones para las energías potenciales son:

$$\mathcal{U}_R = mgs_\beta x, \quad (7)$$

$$\mathcal{U}_{R_1} = m_r g s_\beta (x + l_1 c_\theta + l_2 s_\theta), \quad (8)$$

$$\mathcal{U}_{R_2} = m_r g s_\beta (x + l_1 c_\theta - l_2 s_\theta), \quad (9)$$

$$\mathcal{U}_{R_3} = m_r g s_\beta (x - l_1 c_\theta - l_2 s_\theta), \quad (10)$$

$$\mathcal{U}_{R_4} = m_r g s_\beta (x - l_1 c_\theta + l_2 s_\theta). \quad (11)$$

Las energías cinéticas son

$$\mathcal{K}_R = \frac{1}{2} (m(\dot{x}^2 + \dot{y}^2) + I_{zR} \dot{\theta}^2), \quad (12)$$

$$\begin{aligned} \mathcal{K}_{R_1} = & \frac{m_r}{2} \dot{y}^2 + \left(m_r \dot{\theta} (l_1 c_\theta + l_2 s_\theta) \right) \dot{y} \\ & + \frac{m_r}{2} \dot{x}^2 + \left(m_r \dot{\theta} (l_2 c_\theta - l_1 s_\theta) \right) \dot{x} \\ & + \frac{I_{yr}}{2} \dot{\varphi}_1^2 + \left(\frac{I_{xr}}{2} + m_r \left(\frac{l_1^2}{2} + \frac{l_2^2}{2} \right) \right) \dot{\theta}^2, \end{aligned} \quad (13)$$

$$\begin{aligned} \mathcal{K}_{R_2} = & \frac{m_r}{2} \dot{y}^2 + \left(m_r \dot{\theta} (l_1 c_\theta - l_2 s_\theta) \right) \dot{y} \\ & + \frac{m_r}{2} \dot{x}^2 - \left(m_r \dot{\theta} (l_2 c_\theta + l_1 s_\theta) \right) \dot{x} \\ & + \frac{I_{yr}}{2} \dot{\varphi}_2^2 + \left(\frac{I_{xr}}{2} + m_r \left(\frac{l_1^2}{2} + \frac{l_2^2}{2} \right) \right) \dot{\theta}^2, \end{aligned} \quad (14)$$

$$\begin{aligned} \mathcal{K}_{R_3} = & \frac{m_r}{2} \dot{y}^2 - \left(m_r \dot{\theta} (l_1 c_\theta + l_2 s_\theta) \right) \dot{y} \\ & + \frac{m_r}{2} \dot{x}^2 - \left(m_r \dot{\theta} (l_2 c_\theta - l_1 s_\theta) \right) \dot{x} \\ & + \frac{I_{yr}}{2} \dot{\varphi}_3^2 + \left(\frac{I_{xr}}{2} + m_r \left(\frac{l_1^2}{2} + \frac{l_2^2}{2} \right) \right) \dot{\theta}^2, \end{aligned} \quad (15)$$

$$\mathcal{K}_{R_4} = \frac{m_r}{2} \dot{y}^2 - \left(m_r \dot{\theta} (l_1 c_\theta - l_2 s_\theta) \right) \dot{y} \quad (16)$$

$$\begin{aligned} & + \frac{m_r}{2} \dot{x}^2 + \left(m_r \dot{\theta} (l_2 \cos(\theta) + l_1 s_\theta) \right) \dot{x} \\ & + \frac{I_{yr}}{2} \dot{\varphi}_4^2 + \left(\frac{I_{xr}}{2} + m_r \left(\frac{l_1^2}{2} + \frac{l_2^2}{2} \right) \right) \dot{\theta}^2. \end{aligned}$$

La diferencia de energías o Lagrangiano es representando como \mathcal{L} y cuya expresión es

$$\begin{aligned} \mathcal{L} &= \mathcal{K}_R + \mathcal{K}_{R_1} + \mathcal{K}_{R_2} + \mathcal{K}_{R_3} + \mathcal{K}_{R_4} \\ &\quad - \mathcal{U}_R - \mathcal{U}_{R_1} - \mathcal{U}_{R_2} - \mathcal{U}_{R_3} - \mathcal{U}_{R_4} \\ &= 2I_{xr} \dot{\theta}^2 + \frac{1}{2} \left(I_{zR} \dot{\theta}^2 + (m + 4m_r) (\dot{x}^2 + \dot{y}^2) \right) \\ &\quad + \frac{I_{yr}}{2} (\dot{\varphi}_1^2 + \dot{\varphi}_2^2 + \dot{\varphi}_3^2 + \dot{\varphi}_4^2) + 2m_r \dot{\theta}^2 (l_1^2 + l_2^2) \\ &\quad - gxs_\beta (m + 4m_r). \end{aligned} \quad (17)$$

La ecuación de movimiento de Lagrange para un robot móvil omnidireccional es

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\xi}} \right) - \frac{\partial \mathcal{L}}{\partial \xi} + R_R^B(\theta) E^\top \left(\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\varphi}} \right) - \frac{\partial \mathcal{L}}{\partial \varphi} \right) = R_R^B(\theta) E^\top \boldsymbol{\tau}_\varphi$$

Resolviendo las derivadas se llega a

$$M_R \ddot{\xi} + R_R^B(\theta) E^\top M_\varphi \ddot{\varphi} + \mathbf{g}_R = R_R^B(\theta) E^\top \boldsymbol{\tau}_\varphi. \quad (19)$$

con

$$\begin{aligned} M_R &= \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{11} & 0 \\ 0 & 0 & m_{33} \end{bmatrix}, \\ m_{11} &= m + 4m_r, \\ m_{33} &= 4m_r l_1^2 + 4m_r l_2^2 + 4I_{xr} + I_{zR}, \\ M_\varphi &= I_{yr} I_{4 \times 4}, \\ \mathbf{g}_R &= \begin{bmatrix} gs_\beta (m + 4m_r) \\ 0 \\ 0 \end{bmatrix}. \end{aligned}$$

Despejando η de (3) y derivando respecto al tiempo se obtiene

$$\dot{\boldsymbol{\eta}} = \dot{R}_B^R(\theta) \dot{\xi} + R_B^R(\theta) \ddot{\xi}. \quad (20)$$

con

$$R_B^R(\theta) = R_R^B(\theta)^\top, \quad (21)$$

$$\dot{R}_B^R(\theta) = \dot{\theta} \begin{bmatrix} -s_\theta & c_\theta & 0 \\ -c_\theta & -s_\theta & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (22)$$

Derivando (4) respecto al tiempo y sustituyendo (20) en el resultado se llega a

$$\ddot{\varphi} = E \dot{R}_B^R(\theta) \dot{\xi} + E R_B^R(\theta) \ddot{\xi}. \quad (23)$$

Sustituyendo (23) en (19) el modelo dinámico es reescrito como

$$\bar{M} \ddot{\xi} + \bar{C}(\dot{\theta}) \dot{\xi} + \mathbf{g}_R = R_R^B(\theta) E^\top \boldsymbol{\tau}_\varphi. \quad (24)$$

$$\bar{M} = \begin{bmatrix} \bar{m}_{11} & 0 & 0 \\ 0 & \bar{m}_{11} & 0 \\ 0 & 0 & \bar{m}_{33} \end{bmatrix},$$

$$\bar{m}_{11} = m + 4m_r + \frac{4I_{yr}}{r^2},$$

$$\bar{m}_{33} = 4 \left(I_{xr} + m_r (l_1^2 + l_2^2) + \frac{I_{yr} (l_1 + l_2)^2}{r^2} \right) + I_{zR},$$

$$\bar{C}(\dot{\theta}) = \begin{bmatrix} 0 & -\frac{4I_{yr}\dot{\theta}}{r^2} & 0 \\ \frac{4I_{yr}\dot{\theta}}{r^2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

El modelo (24) es la dinámica que se busca estabilizar. En la siguiente sección se describe el diseño del control.

III. CONTROL

Sea $\xi^* = [x^* \ y^* \ \theta^*]^\top$ la postura deseada, e es el error definido como:

$$e = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x^* - x \\ y^* - y \\ \theta^* - \theta \end{bmatrix} = \xi^* - \xi. \quad (25)$$

El objetivo de control es llevar el error asintóticamente a cero, i.e.

$$\lim_{t \rightarrow \infty} e = 0. \quad (26)$$

La primera y segunda derivada del error respecto al tiempo

$$\dot{e} = \dot{\xi}^* - \dot{\xi}, \quad \ddot{e} = \ddot{\xi}^* - \ddot{\xi}. \quad (27)$$

Sea $u \in \mathbb{R}^3$ la señal de control (i.e. $\tau_\varphi = u \lim_{t \rightarrow \infty} e = 0$), el modelo dinámico expresado en espacio de estados del error es

$$\frac{de}{dt} = \dot{e}, \quad (28)$$

$$\frac{d\dot{e}}{dt} = -\bar{M}^{-1} \left(R_R^B(\theta) E^\top u - \bar{C}(\dot{\theta}) \dot{\xi} - g_R \right).$$

En lo subsiguiente se presenta el diseño de una ley de control basada en un PD. A partir del error de las dos posiciones y la orientación, se calcula el par de cada una de las cuatro ruedas del robot.

III-A. Ley de control

La ley de control a utilizar es conocida como *ley de control por aceleración resuelta*, la cual elimina todas las no linealidades y se complementa con una *ley de control PI*. Las ganancias de la ley de control son $k_{p1}, k_{p2}, k_{p3}, k_{i1}, k_{i2}, k_{i3} \in \mathbb{R}$. La ley de control es la siguiente:

$$u = (E^\top)^\dagger R_B^R(\theta) \left[M \left(\ddot{\xi}^* + K_p \dot{e} + K_i e \right) + \bar{C}(\dot{\theta}) \dot{\xi} + g_R \right] \quad (29)$$

con

$$K_p = \text{diag} \{k_{p1}, k_{p2}, k_{p3}\}, \quad (30)$$

$$K_i = \text{diag} \{k_{i1}, k_{i2}, k_{i3}\}. \quad (31)$$

Sustituyendo (29) en (28) se obtiene la siguiente dinámica de lazo cerrado

$$\frac{de}{dt} = \dot{e}, \quad (32)$$

$$\frac{d\dot{e}}{dt} = -K_p \dot{e} - K_i e.$$

El único punto de equilibrio de (32) es el origen (i.e. $e = 0, \dot{e} = 0$). En el siguiente apartado se analiza la estabilidad del origen.

III-A1. Prueba de estabilidad: La función candidata de Lyapunov (FCL) a utilizar es

$$V(e, \dot{e}) = \frac{1}{2} \dot{e}^\top \dot{e} + \alpha \dot{e}^\top e + \frac{1}{2} e^\top K_i e, \quad (33)$$

donde $\alpha > 0$ seleccionada adecuadamente para que $V(e, \dot{e})$ sea definida positiva. La derivada temporal de (33) es:

$$\dot{V}(e, \dot{e}) = - \begin{bmatrix} e & \dot{e} \end{bmatrix} P \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, \quad (34)$$

con

$$P = \begin{bmatrix} \alpha K_i & \frac{\alpha}{2} (K_p - I_{3 \times 3}) \\ \frac{\alpha}{2} (K_p - I_{3 \times 3}) & K_p \end{bmatrix}.$$

La FCL es definida negativa si $k_{pj} > \alpha$ y $k_{ij} > 0$ con $j = 1, 2, 3$. De lo anterior se concluye estabilidad asintótica global del origen.

IV. SIMULACIÓN

La simulación es realizada en MATLAB/SIMULINK 2017. La trayectoria a realizar es

$$\xi^* = \begin{bmatrix} \frac{1}{2} \cos \left(\frac{1}{10} \pi t \right) + 1 \\ \frac{1}{2} \sin \left(\frac{1}{10} \pi t \right) + 1 \\ \frac{1}{10} \pi t \end{bmatrix}, \quad (35)$$

$$\dot{\xi}^* = \begin{bmatrix} -\frac{1}{20} \pi \sin \left(\frac{1}{10} \pi t \right) \\ \frac{1}{20} \pi \cos \left(\frac{1}{10} \pi t \right) \\ \frac{1}{10} \pi \end{bmatrix}, \quad (36)$$

$$\ddot{\xi}^* = \begin{bmatrix} -\frac{1}{200} \pi^2 \cos \left(\frac{1}{10} \pi t \right) \\ -\frac{1}{200} \pi^2 \sin \left(\frac{1}{10} \pi t \right) \\ 0 \end{bmatrix}. \quad (37)$$

Los parámetros utilizados para el modelo son mostrados en la Tabla I. La duración de la simulación es de 150 segundos.

Tabla I
PARÁMETROS DEL ROBOT

Descripción	Símbolo	Valor
Masa del cuerpo principal	m	2.8 Kg
Masa de las ruedas	m_r	0.38 Kg
Inercia del cuerpo principal	I_{zR}	0.060848 Kg-m ²
Inercia de la rueda en el eje de rotación	I_{xr}	0.000324 Kg-m ²
Inercia de la rueda en el eje del motor	I_{yr}	0.000469 Kg-m ²
Distancia a la rueda a lo largo de x_R	l_1	0.1524 m
Distancia a la rueda a lo largo de y_R	l_2	0.1505 m
Radio de las ruedas	r	0.05m
Angulo del plano inclinado	β	10 grados

Los valores de las ganancias utilizadas son $k_{p1} = 0,25$, $k_{p2} = 0,46$, $k_{p3} = 0,6$, $k_{i1} = 17$, $k_{i2} = 17$, $k_{i3} = 3$. El método de sintonización utilizado es eurístico. Los resultados de la simulación se muestran en las Figuras 3 - 9.

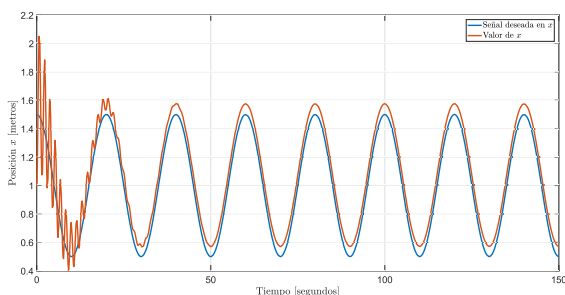


Figura 3. Resultado en simulación de posición en x

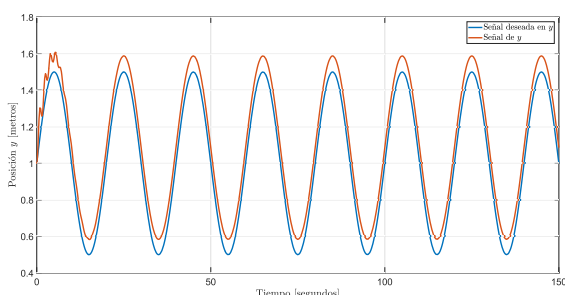


Figura 4. Resultado en simulación de posición en y

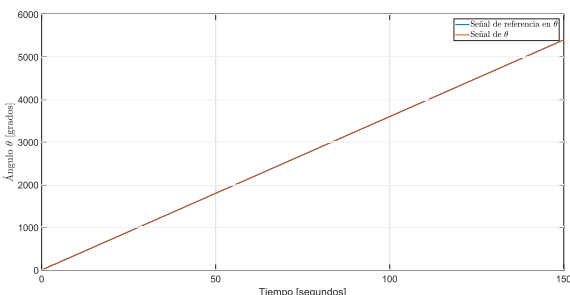


Figura 5. Resultado en simulación de la orientación θ

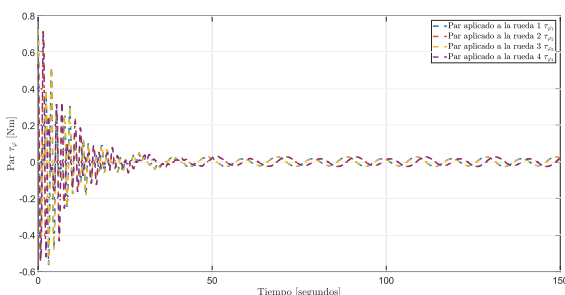


Figura 6. Par aplicado a las ruedas

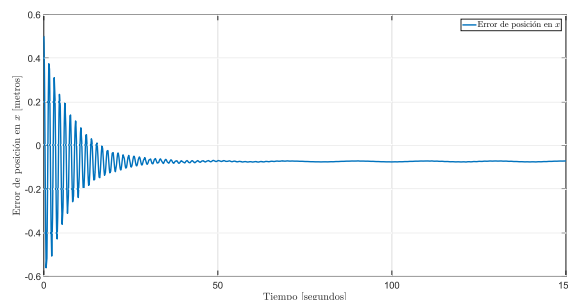


Figura 7. Error de posición en x

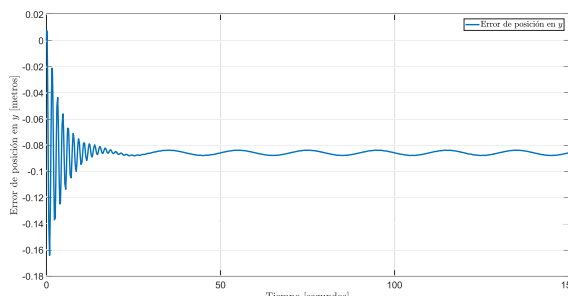


Figura 8. Error de posición en y

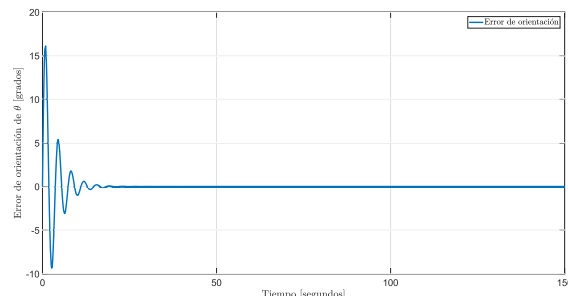


Figura 9. Error de orientación

El tiempo de estabilización para el error de estado estacionario (véanse Figuras 3, 4, 7 y 8) es de 40 segundos con 0.07m. El comportamiento de θ es presentado en la Figura 5 con un compartamiento más rápido que la posición logrando un error de estado estacionario de alrededor de los 25 segundos y error por debajo de los 0.001 grados (véase Figura 9). Los pares de los motores están en la Figura 6. El mayor par aplicado a cada uno de los motores es de 0.73 Nm.

V. CONCLUSIONES

Este artículo tiene como objetivo estudiar el seguimiento de trayectoria de un robot móvil omnidireccional en el plano inclinado. El hecho de hacerlo en plano inclinado fuerza la aparición del vector de gravedad en el modelo dinámico. Los resultados de simulación muestran que el objetivo de control en la orientación es satisfactoriamente alcanzado. En cuanto a la posición muestra un error en estado estacionario, esto es debido a que la trayectoria utilizada para la posición es más demandante que la de orientación. Como trabajo a futuro

se recomienda la implementación de un control adaptable que pueda compensar el vector de gravedad ya que este es constante y podría ser fácilmente estimado con una regresión lineal.

AGRADECIMIENTOS

Este proyecto ha sido parcialmente financiado por PFCE 2019 y proyectos TecNM.

REFERENCIAS

- [1] S. Nof, ed., *Handbook of Industrial Robotics*. John Wiley & Sons, 1998.
- [2] G. Campion, G. Bastin, and B. D'Andréa-Novel, "Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, 1996.
- [3] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Elsevier, 2014.
- [4] R. S. Ortigoza, M. M. Aranda, G. S. Ortigoza, V. M. H. Guzmán, M. A. M. Vilchis, G. S. González, J. C. H. Lozada, and M. O. Carbajal, "Wheeled Mobile Robots : A Review," *IEEE Latin America Transactions*, vol. 10, no. 6, pp. 2209–2217, 2012.
- [5] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, "Control of Wheeled Mobile Systems," in *Wheeled Mobile Robotics*, pp. 61–159, Elsevier, 2017.
- [6] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, vol. 26. JOHN WILEY & SONS, INC., 2006.
- [7] K. Watanabe, "Control of an omnidirectional mobile robot," in *Second International conference on Knowledge-Based Intelligent Electronic System*, no. April, pp. 503–508, 1998.
- [8] D. Zhao, X. Deng, and J. Yi, "Motion and internal force control for omnidirectional wheeled mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 3, pp. 382–387, 2009.
- [9] G. Indiveri, "Swedish wheeled omnidirectional mobile robots: Kinematics analysis and control," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 164–171, 2009.
- [10] H. Sira-Ramírez, C. López-Urbe, and M. Velasco-Villa, "LINEAR OBSERVER-BASED ACTIVE DISTURBANCE REJECTION CONTROL OF THE OMNIDIRECTIONAL MOBILE ROBOT," *Asian Journal of Control*, vol. 15, no. 1, pp. 1–13, 2013.
- [11] C. Ren and S. Ma, "Generalized proportional integral observer based control of an omnidirectional mobile robot," *Mechatronics*, vol. 26, pp. 36–44, 2015.
- [12] N. Hacene and B. Mendil, "Fuzzy Behavior-based Control of Three Wheeled Omnidirectional Mobile Robot," *International Journal of Automation and Computing*, vol. 16, no. 2, pp. 163–185, 2019.
- [13] T. N. Kigezi, S. Alexandru, E. Mugabi, and P. I. Musasizi, "Sliding mode control for tracking of nonholonomic wheeled mobile robots," in *2015 Australian Control Conference, AUCC 2015*, pp. 21–26, 2015.
- [14] K. R. Sharma, F. Dusek, and D. Honc, "Comparitive study of predictive controllers for trajectory tracking of non-holonomic mobile robot," in *2017 21st International Conference on Process Control (PC)*, pp. 197–203, IEEE, jun 2017.
- [15] J. R. García-Sánchez, R. Silva-Ortigoza, S. Tavera-Mosqueda, C. Márquez-Sánchez, V. M. Hernández-Guzmán, M. Antonio-Cruz, G. Silva-Ortigoza, and H. Taud, "Tracking control for mobile robots considering the dynamics of all their subsystems: Experimental implementation," *Complexity*, vol. 2017, 2017.
- [16] A. Saenz, V. Santibañez, and E. Bugarin, "Image Based Visual Servoing for Omnidirectional Wheeled Mobile Robots in Voltage Mode," *International Journal of Engineering Research & Science*, vol. 4, no. 4, pp. 48–53, 2018.
- [17] R. Kelly and V. Santibañez, *Control de movimiento de robots manipuladores*. Pearson Educación, 2003.

Determinación de los Parámetros Geométricos de un Mecanismo Plano de Cinco Barras Mediante una Técnica de Calibración de Cámara

Mauricio Arredondo Soto¹
Departamento de Ingeniería Mecánica
DICIS, Universidad de Guanajuato
Salamanca GTO., México
m.arredondosoto@ugto.mx

J. Elías Gonzalez Chavez
Departamento de Ingeniería Mecánica
DICIS, Universidad de Guanajuato
Salamanca GTO., México
je.gonzalezchavez@ugto.mx

Mario A. García Murillo*
Departamento de Ingeniería Mecánica (*correspondencia)
DICIS, Universidad de Guanajuato
Salamanca GTO., México
garcia.mario@ugto.mx

Axel Herrera Cabrera
Departamento de Ingeniería Mecánica
DICIS, Universidad de Guanajuato
Salamanca GTO., México
a.herreracabrera@ugto.mx

Resumen—En este trabajo se presenta un método novedoso para determinar los parámetros geométricos de un mecanismo plano de cinco barras destinado a tareas de posicionamiento en el plano. El método utiliza técnicas de fotogrametría por lo que se requiere del proceso de calibración de cámara y está basado en el concepto de espacio vértice. Una metodología para la caracterización del sistema de medición es incluida en este trabajo. La aplicabilidad y ventajas de usar la metodología propuesta se muestran a través de la estimación de las dimensiones geométricas de un mecanismo plano de cinco barras .

Palabras clave—Calibración de cámara, parámetro geométrico, mecanismo plano, espacio vértice.

I. INTRODUCCIÓN

En las últimas décadas ha aumentado drásticamente el interés por el desarrollo e implementación de robots para asistir y automatizar procesos tales como tareas de maquinado, operaciones de tomar y posicionar, dispositivos de rehabilitación y cirugía, medición de coordenadas, entre otros. Muchas de estas tareas demandan gran precisión, sin embargo, un número elevado de juntas cinemáticas pasivas puede conducir a errores de manufactura y ensamble, y por tanto disminuir su precisión. No obstante, este problema se puede reducir con una calibración apropiada de los parámetros geométricos del robot, sin embargo, esta es una tarea complicada desde los puntos de vista teórico y experimental [1].

Existen varios métodos para determinar los parámetros geométricos de robots y mecanismos. Su clasificación está basada generalmente en la posición de los instrumentos de medición y los elementos auxiliares. Entonces, se pueden definir tres grupos: medición externa, auto-calibración y calibración mediante movimientos restringidos del robot [1]. El primer grupo incluye estrategias basadas en la medición de la pose (posición y orientación) del efector final u otros elementos representativos mediante instrumentos externos, tales como

máquinas de medición por coordenadas [2], sistemas de visión e instrumentos ópticos [3]–[6], sensores láser [7], sistemas de visión utilizando patrones [8], [9] o esferas [1]. El segundo grupo está relacionado con auto-calibración, donde el método utilizado calibra el robot automáticamente, incluso durante su operación. Estos incluyen el uso de sensores redundantes [10], eslabones redundantes [11], entre otros [12].

Finalmente, la idea principal del tercer grupo se basa en que normalmente el número de sensores del robot es igual a su número de grados de libertad [13]. Por lo tanto, si uno o más GdL del mecanismo están fijos o restringidos, la movilidad del sistema resultante será menor que el número de sensores por lo que el algoritmo de calibración puede realizarse de forma similar a los métodos que utilizan sensores extra o eslabones adicionales, como se demuestra en [14]–[16].

En este trabajo se presenta una metodología novedosa para determinar los parámetros geométricos de mecanismos. Esta utiliza la popular técnica de calibración de cámara propuesta por Zhang [17] que está basada en observar un patrón plano en diversas posiciones y orientaciones. El método tiene la ventaja de requerir solo una cámara y patrones planos, además de que puede ser realizada sin la necesidad de posicionar el robot mediante un sistema de control. Más aún, el método puede ser utilizado para determinar las dimensiones de eslabones particulares sin desensamblar el mecanismo.

Los experimentos son realizados para un mecanismo plano de cinco barras con 2-GDL. El resto de este trabajo está organizado como sigue: Sección 2 presenta una breve descripción del mecanismo estudiado. Sección 3 analiza el proceso de calibración y el método de Zhang. Sección 4 introduce el sistema de medición propuesto. Secciones 5 y 6 describen la metodología y su aplicaciones para calcular los parámetros geométricos del mecanismo descrito en la Sección 1. Finalmente, en la sección 7 se discuten las conclusiones del trabajo.

II. DESCRIPCIÓN DEL MECANISMO

El robot estudiado es el mecanismo plano de cinco barras o 5-R, donde R indica revoluta, que se muestra en la Figura 1. Este dispositivo se emplea para posicionar el palpador de un sistema digitalizador de geometría con el cual se pueden obtener las coordenadas de los puntos pertenecientes a contornos de piezas por donde pasa el palpador.

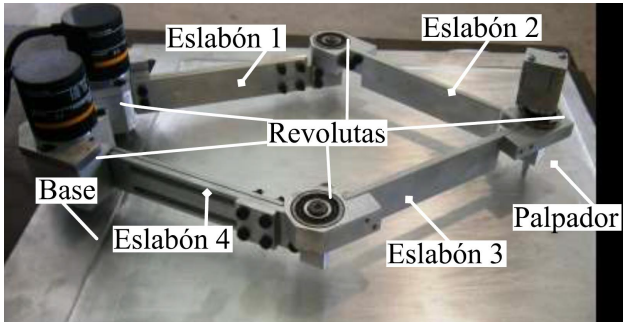


Fig. 1. Mecanismo plano de cinco barras.

Los parámetros geométricos a considerar en este estudio son las longitudes de los cinco eslabones, $L_i, i = 0..4$ y las direcciones de los ejes de revoluta, n_A, n_B, n_C, n_D y n_E , que además son nominalmente paralelos entre sí y normales al plano de movimiento del mecanismo. Un esquema del mecanismo plano 5-R con sus parámetros geométricos se muestra en la Figura 2.

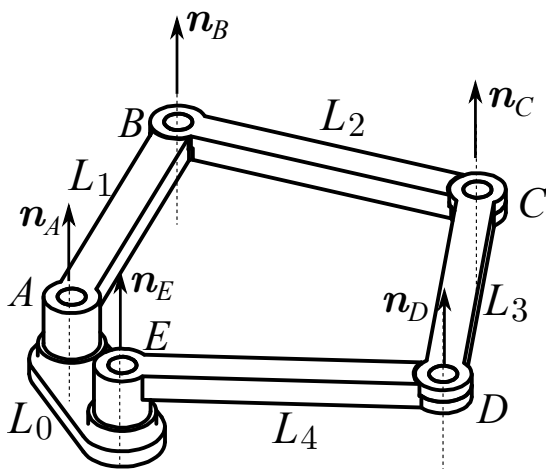


Fig. 2. Esquema del mecanismo y sus parámetros geométricos.

Las longitudes de los eslabones, mostradas en la Tabla 1, se obtuvieron utilizando una máquina de medición por coordenadas de la marca MITUTOYO modelo BH-V504 como se muestra en la Figura 3.

III. DESCRIPCIÓN DEL SISTEMA DE MEDICIÓN

Una calibración de cámara consiste en calcular las características geométricas y ópticas de la cámara, las cuales son

TABLA I
LONGITUDES DE LOS ESLABONES DEL MECANISMO, (MM).

Parámetro	Magnitud
L_0	68.4203
L_1	175.408
L_2	175.481
L_3	174.074
L_4	176.917

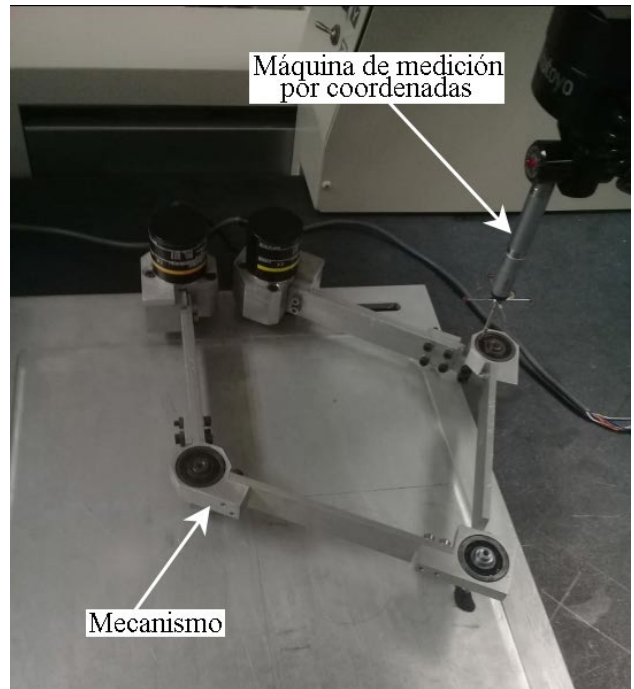


Fig. 3. Obtención de las dimensiones del mecanismo mediante una máquina de medición por coordenadas.

conocidas como parámetros extrínsecos e intrínsecos, vea la Figura 4. En palabras sencillas, el proceso consiste en extraer información 3D de imágenes 2D.

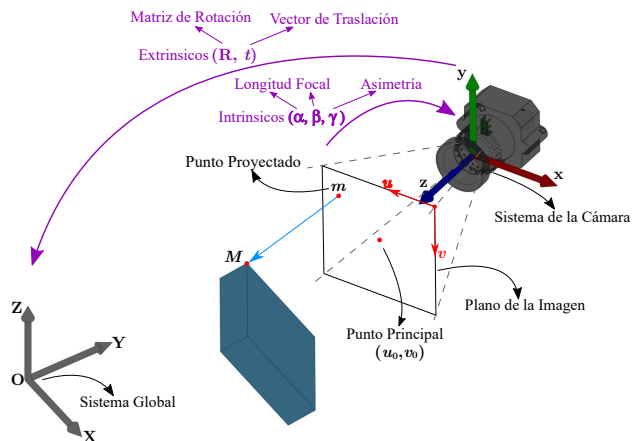


Fig. 4. Parámetros intrínsecos y extrínsecos del modelo de cámara.

La idea principal detrás del proceso de calibración es describir la matriz que relaciona los sistemas de referencia de la cámara y el mundo real calculando los parámetros intrínsecos (coordenadas del punto principal, $[u_0, v_0]^T$, factores de escala α y β de la imagen, u y v ejes de la imagen, y un parámetro que describe la asimetría de los dos ejes de la imagen, γ), y los parámetros extrínsecos (posición, \mathbf{t} y orientación de la cámara relativa a un sistema de referencia en la imagen descrita por la matriz \mathbf{R}) de la cámara, vea la Figura. 4.

Normalmente un sistema de visión requiere más de una cámara para extraer información 3D. Sin embargo, el algoritmo propuesto por Zhang [17] permite realizar este proceso utilizando solo una cámara. En la Figura 4 se muestran los elementos principales considerados en el modelo geométrico de la cámara. El punto M , determinado por el vector $\mathbf{M} = [x, y, z]^T$ en el mundo real, tiene un correspondiente punto m cuyas coordenadas en el plano de la imagen son (u, v) . Entonces, las coordenadas de M tienen dos transformaciones: a) proyección que transforma un punto 3D en un punto en 2D de la imagen; b) transformación del sistema de la cámara (en unidades métricas) a el sistema de la imagen (en pixeles). Estas dos transformaciones son representadas matemáticamente como sigue:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

La primer matriz del lado derecho de la ecuación (1) es llamada la matriz de parámetros intrínsecos de la cámara. Por otro lado, la segunda matriz contiene los parámetros extrínsecos, donde \mathbf{R} es la matriz de rotación asociada con el sistema de la cámara y un sistema de referencia en el mundo real, finalmente, \mathbf{t} es el vector de traslación entre ambos sistemas de referencia.

El algoritmo de Zhang está basado en el cálculo de la matriz de homografía o de proyección entre el plano del patrón, el cual es un arreglo de puntos coplanares, y su imagen en la cámara para n imágenes diferentes. El proceso consiste en utilizar la matriz de homografía para re proyectar los puntos 2D en el espacio 3D. Entonces, el algoritmo vuelve a tomar el arreglo de puntos y compara su posición con los del arreglo del paso previo. Las diferencias resultantes son minimizadas mediante un proceso de optimización no-lineal que utiliza el algoritmo de Levenberg-Marquard.

Esta metodología proporciona los mejores parámetros intrínsecos para componer la matriz de homografía final, la cual puede ser usada para extraer los parámetros extrínsecos 3D más precisos de un patrón plano para alguna posición espacial. Este proceso está ilustrado en la Figura 5.

IV. CARACTERIZACIÓN DEL SISTEMA DE MEDICIÓN

Para asegurar que las mediciones son precisas, es necesario caracterizar el sistema utilizado. Específicamente, se busca determinar el error relativo, repetibilidad y reproducibilidad de las mediciones. Estos parámetros se calcularon basados en

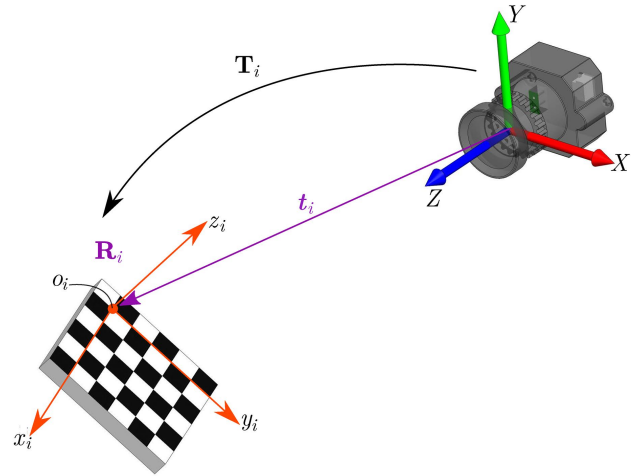


Fig. 5. Medición de la pose, $\mathbf{T}_i = [\mathbf{R}, \mathbf{t}]$, de un patrón plano relativo al sistema de la cámara.

la norma ISO 5725 [18]. El sistema de medición se muestra en la Figura 6. El arreglo experimental está compuesto de los siguientes dispositivos:

- Patrones de calibración con tamaño de cuadro de 10 mm.
- Cámara Genie Nano C2020 de la marca Teledyne.
- Estación de traslación micrométrica de tres ejes.
- Equipo de cómputo, PC.



Fig. 6. Configuración del experimento para la caracterización del sistema de medición.

La discusión anterior sirvió para formular el siguiente procedimiento.

- 1) Realizar la calibración de cámara utilizando el método presentado en [17], el cual fue descrito en la sección previa. Los parámetros intrínsecos obtenidos se muestran en la Tabla II.
- 2) El patrón se posiciona en la estación micrométrica de 3-ejes. Cada eje es desplazado 2.50 mm con saltos de 0.50 mm a una distancia de 65 cm de la cámara, realizando

un barrido completo del espacio tridimensional. Una imagen es tomada para cada posición.

- 3) El procedimiento es ejecutado por tres diferentes operadores, tres veces cada uno.

Los resultados de la caracterización del sistema de medición se presentan en la Tabla III.

TABLA II
PARÁMETROS INTRÍNECOS DE LA CÁMARA

Parámetro	Valor
Longitud focal	[1359.0117 1364.7086] \pm [19.5053 19.6984]
Punto principal	[363.8465 224.9075] \pm [3.01340 2.0200]
Distorsión	[-0.3709 -0.5355 -0.0007 0.0031 0.0000] \pm [0.0155 0.2217 0.0002 0.0008 0.0000]
Error en pixel	[0.0658 0.1014]

TABLA III
CARACTERÍSTICAS DEL SISTEMA DE MEDICIÓN

	Traslación	Rotación
Error	0.003744 mm	$1.43 \times 10^{-4}^\circ$
Desviación Stdr.	0.001975	2.78×10^{-5}
Repetibilidad	0.008673	0.001327
Reproducibilidad	0.008333	0.001198

V. DESCRIPCIÓN DEL MÉTODO PROPUESTO

La técnica novedosa de identificación de parámetros propuesta en este trabajo está inspirada en el método para obtener los espacios de trabajo con orientación constante de robots [19]. Este método busca determinar los espacios vértice que generan los pares cinemáticos del mecanismo, para después calcular sus dimensiones en términos de las coordenadas de los centros de estos espacios.

La idea básica es obtener las coordenadas de varios puntos que yacen en la superficie de cada espacio vértice. Para este fin, el sistema de medición descrito en las secciones 3 y 4 es utilizado e implementado en Matlab© [20]. Para después reconstruir estos espacios tridimensionales, ajustando los puntos a una geometría mediante el método de mínimos cuadrados.

En primer lugar, se sujetan patrones planos a cada uno de los cuatro eslabones móviles como se muestra en la Figura 7. Después, se instala la cámara de manera que esta pueda observar todos los patrones al mismo tiempo, para un rango de movimiento completo. El sistema de medición completo para realizar el experimento se muestra en la Figura 8.

La tarea es obtener la dirección y posición de cada uno de los ejes de revoluta, relativos al sistema de la cámara, T_c , para posteriormente encontrar su intersección con el plano de trabajo y de esta forma determinar las dimensiones de trabajo reales de los eslabones del mecanismo. En la Figura 9 se muestra un esquema del mecanismo junto con los cuatro patrones, sus correspondientes sistemas de referencia, $T_i, i = 1...4$, y el sistema coordinado de la cámara, T_c .

El experimento se realizó capturando 30 imágenes de 30 posiciones distintas del mecanismo plano. Luego de extraer los parámetros extrínsecos de los sistemas de referencia de los

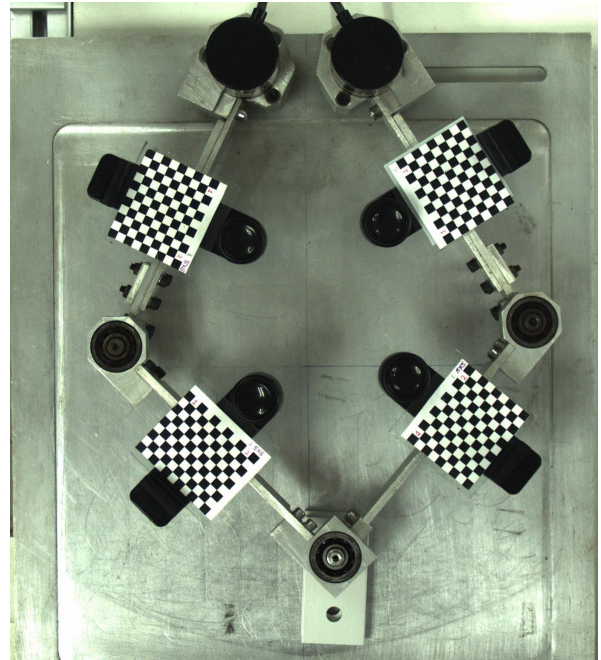


Fig. 7. Mecanismo con patrones sujetos a sus cuatro eslabones móviles.

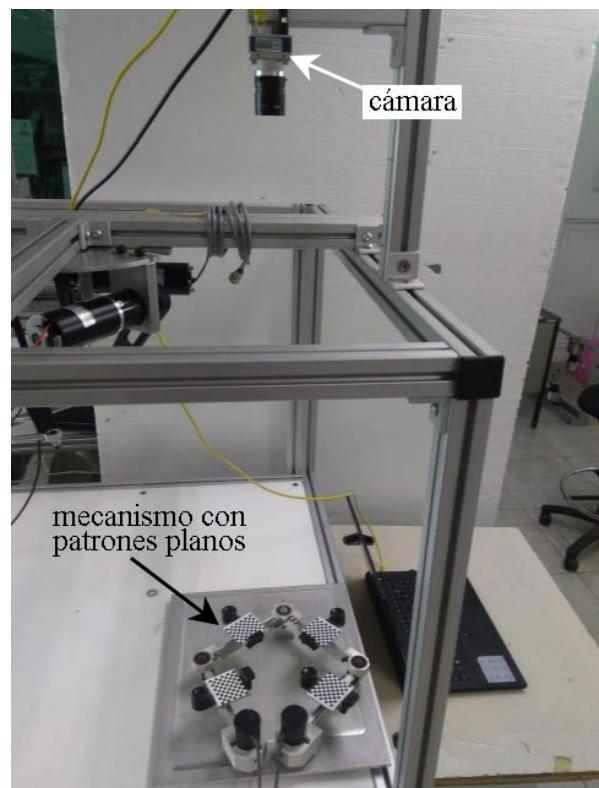


Fig. 8. Configuración completa del experimento.

patrones para cada posición, se obtienen las nubes de puntos mostradas en la Figura 10.

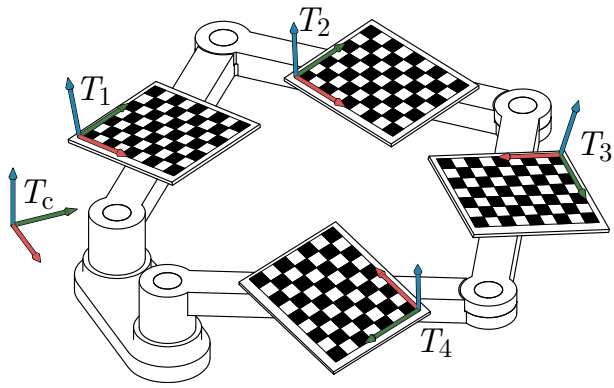


Fig. 9. Mecanismo con los cuatro patrones sujetos, sus correspondientes sistemas de referencia, T_i , $i = 1...4$, y el sistema coordinado de la cámara, T_c .

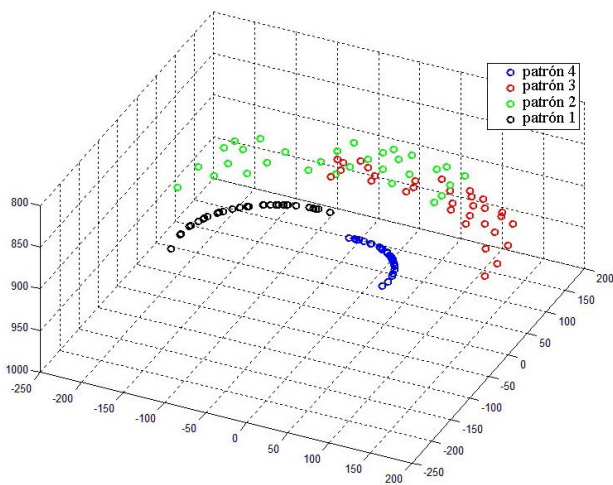


Fig. 10. Nubes de puntos generadas por los orígenes de los sistemas de referencia correspondientes a los patrones durante el experimento.

A. Análisis para los ejes n_A y n_E

Los espacios vértice de las juntas de revoluta A y E , son circunferencias que yacen en planos perpendiculares a los ejes n_A y n_E respectivamente; y cuyos centros, ${}^c r_A$ y ${}^c r_E$, definen la posición de estos mismos ejes, el superíndice "c" indica que estas coordenadas están relativas al sistema de la cámara, T_c . En la Figura 11 se muestran los parámetros necesarios para analizar los ejes de las revolutas A y E .

El origen del sistema coordinado del patrón 1 para la i -ésima imagen, representado por el punto ${}^c O_{1,i}({}^c x_{1,i}, {}^c y_{1,i}, {}^c z_{1,i})$, yace en el perímetro de una circunferencia, por lo que es necesario determinar primero el plano de dicha circunferencia. El plano que mejor se ajusta a un conjunto de puntos es aquel que minimiza las distancias ortogonales de los puntos al plano [21], es decir, minimiza la función:

$$E(a, b, c, d) = \sum_{i=1}^n \frac{|ax_i + by_i + cz_i + d|^2}{(a^2 + b^2 + c^2)} \quad (2)$$

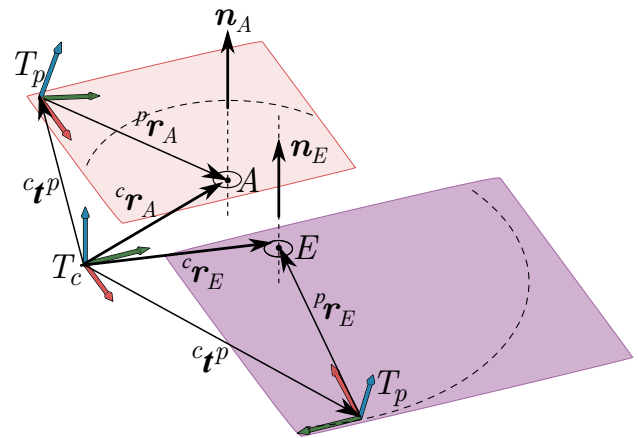


Fig. 11. Parámetros necesarios para analizar los ejes n_A y n_E .

Este ajuste desemboca en un problema de descomposición de valor singular en el cual, el plano que mejor se ajusta contiene al centroide de los datos, $c_0 = (x_0, y_0, z_0)$ y el vector normal es el vector singular correspondiente al valor singular más pequeño de la matriz M definida como:

$$M = \begin{bmatrix} x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ x_2 - x_0 & y_2 - y_0 & z_2 - z_0 \\ \vdots & \vdots & \vdots \\ x_n - x_0 & y_n - y_0 & z_n - z_0 \end{bmatrix} \quad (3)$$

Una vez obtenido el plano, los puntos se proyectan sobre este y se escriben relativos a un nuevo sistema coordinado que yace en el plano, T_p . Entonces, el problema se reduce a encontrar los parámetros de la circunferencia que mejor se ajusta a un conjunto de puntos en el plano. La ecuación de una circunferencia puede escribirse como:

$$({}^p x_{1,i} - h)^2 + ({}^p y_{1,i} - k)^2 = r_i^2, \quad (4)$$

donde el superíndice "p" indica que los puntos están relativos al sistema del plano, T_p , r_i es el radio de la circunferencia y los parámetros h y k son las coordenadas del centro de la circunferencia. El proceso de ajustar una circunferencia a un conjunto de puntos consiste en minimizar las diferencias entre los radios, $r_i - r$, así la función objetivo es:

$$d_i(\mathbf{u}) = \sqrt{({}^p x_{1,i} - h)^2 + ({}^p y_{1,i} - k)^2} - r \quad (5)$$

La ecuación (5) indica que se debe encontrar el centro y el radio de la circunferencia que mejor se ajuste al conjunto de n puntos en el problema de mínimos cuadrados no-lineal descrito como:

$$E(h, k, r) = \sum_{i=1}^n d_i(\mathbf{u})^2, \quad (6)$$

donde $\mathbf{u} = [h, k, r]^T$.

El problema de mínimos cuadrados no-lineal puede ser resuelto con el algoritmo de Gauss-Newton. Este es un procedimiento iterativo donde se requiere una aproximación

inicial \mathbf{u}_0 . El método consiste en realizar varias iteraciones de acuerdo al siguiente esquema matemático:

$$\mathbf{u}_{s+1} = \mathbf{u}_s - (\mathbf{J}(\mathbf{u}_s)^T \mathbf{J}(\mathbf{u}_s))^{-1} \mathbf{J}(\mathbf{u}_s)^T \mathbf{d}(\mathbf{u}_s) \quad (7)$$

donde $\mathbf{J}(\mathbf{u})$ es la matriz jacobiana definida por las derivadas parciales de la función $d_i(\mathbf{u})$ con respecto a cada variable objetivo $(h, k, r)^T$, y el vector $\mathbf{d}(\mathbf{u})$ contiene las funciones $d_i(\mathbf{u})$ para todos los datos.

Debido a que el centro de la circunferencia ${}^p\mathbf{r}_A = [h, k, 0]^T$ está relativo al sistema del plano, T_p , es necesario transformarlo al sistema de la cámara, T_c , vea Figura 11, lo cual se consigue mediante la siguiente ecuación:

$${}^c\mathbf{r}_A = {}^c\mathbf{t}^p + [{}^c\mathbf{R}^p] {}^p\mathbf{r}_A \quad (8)$$

donde ${}^c\mathbf{t}^p$ es el vector del origen del sistema en el plano relativo al de la cámara y ${}^c\mathbf{R}^p$ es la matriz de orientación del sistema del plano relativo al de la cámara. Nótese que se puede realizar un análisis similar para el eje de la revoluta E .

B. Análisis para los ejes \mathbf{n}_B , \mathbf{n}_C y \mathbf{n}_D

Los espacios vértice de las juntas de revoluta B , C y D también generan circunferencias que yacen en planos perpendiculares a los ejes \mathbf{n}_B , \mathbf{n}_C y \mathbf{n}_D respectivamente. Sin embargo, estas circunferencias no son relativas al sistema coordenado de la cámara T_c , como se observa en la Figura 10, sino que son relativas al sistema de referencia de cualquier patrón adyacente. Por ejemplo, el origen del sistema coordenado del patrón 3 relativo al sistema de referencia del patrón 4, T_4 para la i -ésima imagen, representado por el punto ${}^4O_{3,i}$ (${}^4x_{3,i}$, ${}^4y_{3,i}$, ${}^4z_{3,i}$), yace en el perímetro de una circunferencia con centro en ${}^4\mathbf{r}_D$ como se observa en la Figura 12.

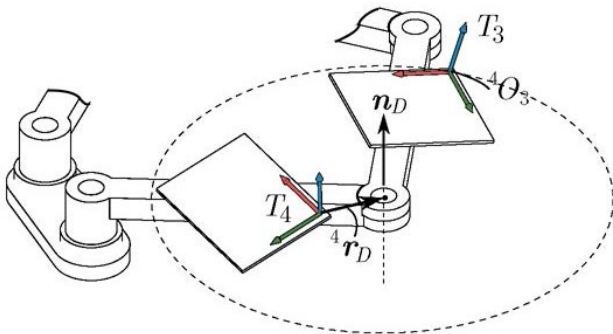


Fig. 12. Circunferencia producida por el movimiento del punto 4O_3 relativo a T_4 .

La pose de un sistema coordenado, correspondiente a un patrón, relativo a otro se puede obtener mediante la siguiente transformación:

$${}^i\mathbf{T}^j = [{}^i\mathbf{T}^j]^{-1} \mathbf{T}^j \quad (9)$$

donde ${}^i\mathbf{T}^j = \begin{bmatrix} {}^i\mathbf{R}^j & {}^i\mathbf{t}^j \\ \mathbf{0} & 1 \end{bmatrix}$, ${}^i\mathbf{R}^j$ es la matriz de orientación del sistema T_j relativo al sistema T_i y ${}^i\mathbf{t}^j$ es el vector de posición del origen del sistema T_j relativo al T_i .

Entonces, es posible aplicar el mismo procedimiento descrito en el análisis para los ejes \mathbf{n}_A y \mathbf{n}_E , para obtener las posiciones de los ejes de revoluta ${}^1\mathbf{r}_B$, ${}^2\mathbf{r}_C$ y ${}^4\mathbf{r}_D$. Nótese que las direcciones de los ejes \mathbf{n}_B , \mathbf{n}_C y \mathbf{n}_D pueden ser obtenidas directamente relativas al sistema de la cámara T_c , calculando el vector normal del plano que mejor se ajusta a las nubes de puntos de los patrones 2 y 3, mostradas en la Figura 10. Por lo tanto, es necesario determinar los vectores de posición de los ejes de revoluta ${}^1\mathbf{r}_B$, ${}^2\mathbf{r}_C$ y ${}^4\mathbf{r}_D$ relativos al sistema de la cámara T_c , mediante las siguientes transformaciones:

$${}^c\mathbf{r}_{Bi} = {}^c\mathbf{t}_i^1 + [{}^c\mathbf{R}_i^1] {}^1\mathbf{r}_B \quad (10)$$

$${}^c\mathbf{r}_{Ci} = {}^c\mathbf{t}_i^2 + [{}^c\mathbf{R}_i^2] {}^2\mathbf{r}_C \quad (11)$$

$${}^c\mathbf{r}_{Di} = {}^c\mathbf{t}_i^4 + [{}^c\mathbf{R}_i^4] {}^4\mathbf{r}_D \quad (12)$$

donde i indica la i -ésima posición de los sistemas de referencia T_1 , T_2 y T_4 respectivamente, relativos al sistema de la cámara.

VI. DIMENSIONES DE TRABAJO REALES DE LOS ESLABONES DEL MECANISMO

Una vez conocida la posición y orientación de todos los ejes de revoluta relativos al sistema de referencia de la cámara, ${}^c\mathbf{r}_A$, ${}^c\mathbf{r}_{Bi}$, ${}^c\mathbf{r}_{Ci}$, ${}^c\mathbf{r}_{Di}$, ${}^c\mathbf{r}_E$, \mathbf{n}_A , \mathbf{n}_B , \mathbf{n}_C , \mathbf{n}_D y \mathbf{n}_E , es posible proyectarlos para encontrar su intersección con el plano de trabajo como se ilustra en la Figura 13, donde se proyectan los ejes de las revolutas A y E . El plano de trabajo se obtiene de ajustar un plano a un conjunto de puntos obtenidos de deslizar un patrón plano por la superficie de la base del mecanismo y capturar imágenes durante el proceso.

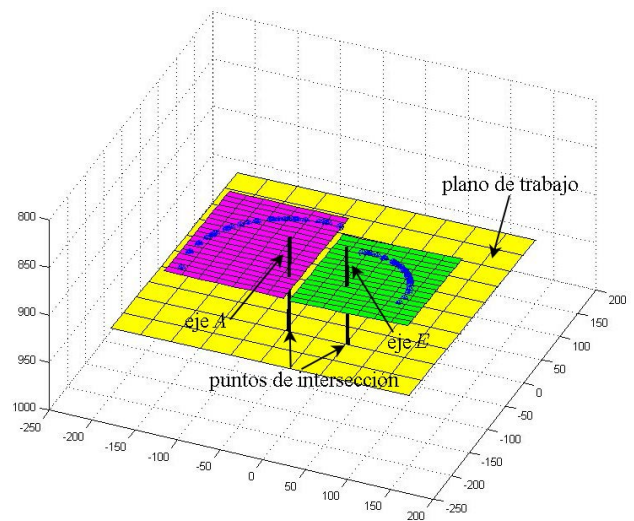


Fig. 13. Ejes de las revolutas A y E , intersectando al plano de trabajo del mecanismo.

Las longitudes de trabajo reales de los eslabones se obtienen de calcular las distancias entre los puntos de intersección de los ejes con el plano de trabajo. Debido a que los ejes de las revolutas B , C y D son móviles, el punto de intersección cambia de posición, por lo que, las longitudes de los eslabones L_1 , L_2 , L_3 y L_4 , se obtuvieron calculando el promedio de

las distancias para las 30 posiciones del mecanismo. Las longitudes del mecanismo calculadas se presentan en la Tabla IV; se considera como el error en la medición la diferencia entre los valores calculados por este método y las obtenidas con la máquina de medición por coordenadas.

TABLA IV

LONGITUDES DE TRABAJO REALES DE LOS ESLABONES DEL MECANISMO, (MILÍMETROS).

Parámetro	Magnitud calculada	Error
L_0	68.477	0.0567
L_1	176.001	0.593
L_2	175.532	0.051
L_3	174.841	0.767
L_4	177.320	0.403

VII. CONCLUSIONES

En esta contribución se presentó una metodología para determinar los parámetros geométricos de robots y mecanismos, basada en una técnica de calibración de cámara. Para mostrar el potencial de este método, se calcularon las longitudes de trabajo reales de un mecanismo plano de cinco barras con 2-GdL. El sistema de medición propuesto es fácil de implementar y requiere una sola cámara, a diferencia de otros métodos donde se necesita de sensores adicionales o de instalar eslabones extra. Además, no es necesario posicionar el robot mediante un sistema de control ni implementar un algoritmo que requiera de un gran número de poses, para determinar los parámetros geométricos del mecanismo.

La caracterización del sistema de medición demostro que cuenta con buena precisión y confiabilidad para determinar los parámetros geométricos del mecanismo analizado. Los resultados obtenidos en este trabajo permiten visualizar el potencial de la metodología propuesta, para el análisis de mecanismos con arquitecturas más complejas tales como robots seriales espaciales, plataformas paralelas, robots móviles, exoesqueletos, etc.

REFERENCIAS

- [1] A. Traslosheros and J. Sebastián, "A method for kinematic calibration of a parallel robot by using one camera in hand and a spherical object," *Advanced Robotics*, 2011.
- [2] J. Ouyang and I. Jawahir, "Ball array calibration on a coordinate measuring machine using a gage block," *Measurement*, vol. 16, no. 4, pp. 219 – 229, 1995.
- [3] R. Garrido and M. A. Trujano, "Stability analysis of a visual pid controller applied to a planar parallel robot," *International Journal of Control, Automation and Systems*, vol. 17, no. 6, pp. 1589–1598, 2019.
- [4] S. Bai and M. Y. Teo, "Kinematic calibration and pose measurement of a medical parallel manipulator by optical position sensors," *Journal of Robotic Systems*, vol. 20, no. 4, pp. 201–209, 2003.
- [5] Y. Zhang and F. Gao, "A calibration test of Stewart platform," in *2007 IEEE International Conference on Networking, Sensing and Control*, London, UK, April 2007, pp. 297–301.
- [6] M. Dinham and G. Fang, "A low cost hand-eye calibration method for arc welding robots," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Guilin, China: IEEE, Dec 2009, pp. 1889–1893.
- [7] G. Meng, L. Tiemin, and Y. Wensheng, "Calibration method and experiment of Stewart platform using a laser tracker," in *2003 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, Washington, DC, USA, Oct 2003, pp. 2797–2802.
- [8] J. M. S. Motta, G. C. de Carvalho, and R. McMaster, "Robot calibration using a 3D vision-based measurement system with a single camera," *Robotics and Computer-Integrated Manufacturing*, vol. 17, no. 6, pp. 487–497, dec 2001.
- [9] Y. Takeda, S. Kawasumi, D. Matsuura, and E. Castillo-Castaneda, "Compliance analysis of 3-RPSR parallel mechanism for movable die drive mechanism of pipe bender," in *Proceedings of 3rd IFToMM International Symposium on Robotics and Mechatronics (ISRMM2013)*, 2013.
- [10] G. Ecorchard and P. Maurine, "Self-calibration of delta parallel robots with elastic deformation compensation," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alta., Canada, Aug 2005, pp. 1283–1288.
- [11] Y.-J. Chiu and M.-H. Perng, "Self-calibration of a general hexapod manipulator with enhanced precision in 5-dof motions," *Mechanism and Machine Theory*, vol. 39, no. 1, pp. 1 – 23, 2004.
- [12] H.-K. Lee, K. Choi, J. Park, and H. Myung, "Self-calibration of gyro using monocular slam for an indoor mobile robot," *International Journal of Control, Automation and Systems*, vol. 10, no. 3, pp. 558–566, 2012.
- [13] E. Hernández-Martínez, C. López-Cajún, and J. Jáuregui-Correa, "Calibration of Parallel Manipulators and their Application to Machine Tools. A State of the Art Survey," *Ingeniería Investigación y Tecnología*, vol. 11, no. 2, pp. 141–154, 2010.
- [14] W. Khalil and S. Besnard, "Self calibration of Stewart-Gough parallel robots without extra sensors," *Robotics and Automation, IEEE ...*, vol. 15, no. 6, pp. 1029–1033, 1999.
- [15] A. Rauf and J. Ryu, "Fully autonomous calibration of parallel manipulators by imposing position constraint," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, vol. 3, Seoul, South Korea, May 2001, pp. 2389–2394.
- [16] A. Rauf, A. Pervez, and J. Ryu, "Experimental results on kinematic calibration of parallel manipulators using a partial pose measurement device," *Robotics, IEEE Transactions on*, vol. 22, no. 2, pp. 379–384, 2006.
- [17] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [18] I. I. O. for Standardization, "Accuracy (trueness and precision) of measurement methods and results," url: <https://www.iso.org/obp/ui/iso:std:iso:5725:-1:ed-1:v1:en>, 12, 1994.
- [19] I. A. R. J. Bonev, "Geometrical method for computing the constant-orientation workspace of 6-PRRS parallel manipulators," *Mechanism and Machine Theory*, vol. 36, no. 1, pp. 1–13, 2001. [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-0035199551&partnerID=40&md5=d00a00ecfa91f835e9bee95687384855>
- [20] J.-Y. Bouguet, "Camera calibration toolbox for matlab," <http://www.vision.caltech.edu/bouguetj>, last access: 05/05/2019.
- [21] V. Schomaker, J. Waser, R. Marsh, , and G. Bergman, "To fit a plane or a line to a set of points by least squares," *Acta crystallographica*, vol. 12, no. 8, pp. 600–604, 1959.

Análisis e Implementación de una Red Neuronal Convolutiva para la Estimación de Pose Utilizando un Sistema de Visión Monocular

Pérez Chong, José Adrián
Tecnológico Nacional de
México, I.T. de La Paz
La Paz, México
perezmsc18@itlp.edu.mx

Martínez Díaz, Saul
Tecnológico Nacional de
México, I.T. de La Paz
La Paz, México
saul.md@lapaz.tecnm.mx

Gómez Torres, José Luis
Tecnológico Nacional de
México, I.T. de La Paz
La Paz, México
jose.gt@lapaz.tecnm.mx

Castro Liera, Iliana
Tecnológico Nacional de
México, I.T. de La Paz
La Paz, México
iliana.cl@lapaz.tecnm.mx

Resumen—En este documento se propone la integración de una Red Neuronal Convolutiva en la estimación de pose con un sistema de visión monocular. Se propone entrenar una Red Neuronal para el reconocimiento de objetos con dimensiones conocidas, para así, obtener una matriz homográfica que permita calcular la pose de la cámara con una escala conocida.

Index Terms—estimación de pose, visión artificial, red neuronal convolutiva, detección de objetos

I. INTRODUCCIÓN

La estimación del movimiento basado en la cámara es un área altamente estudiada debido a su facilidad de implementación y aprovechamiento de recursos limitados.

La Odometría es el estudio de la estimación de la posición de vehículos durante la navegación. Dentro de la Odometría, la técnica que se basa en la estimación de la posición a través de información obtenida en imágenes se conoce como Odometría Visual (OV). Las ventajas de la OV están apegadas al bajo costo de los dispositivos fotográficos y a la gran cantidad de información que brindan las imágenes permitiendo buenos resultados. Uno de los primeros trabajos para estimar el movimiento a través de una cámara fue realizado por [1] y ha ayudado a plantear las bases para la OV.

La inspiración para la odometría visual es puramente biológica, nuestros cerebros utilizan varias pistas para percibir profundidad y el flujo óptico, tales como variaciones en textura, gradientes, oclusión, tamaño de objetos conocidos, sombras, desenfoque, etc. Por lo tanto es normal el querer integrar alguno de estos métodos a los algoritmos conocidos.

La odometría visual se puede implementar monocular, estéreo o con cámaras RGB-D (cámaras con sensor de profundidad), dependiendo del diseño del sistema. La OV estéreo puede estimar la escala inmediatamente a diferencia de la OV monocular, sin embargo, la OV estéreo requiere un proceso de calibración mayor y una sincronización rígida, sin la cual, el error se propaga a lo largo del tiempo.

La implementación monocular es preferida debido al costo reducido que presenta y la posibilidad de aplicarse en estructuras de menor tamaño donde establecer una separación adecuada entre ambas lentes es complicado.

Existen algoritmos aceptados para la estimación del movimiento basado en una secuencia de imágenes como [2]. La OV basada en geometría como se observa en la fig. 1 es capaz de calcular el flujo óptico en secuencias de imágenes y su precisión depende del correcto emparejamiento de características como se demuestra en [3]. Contamos con otro tipo de OV basado en algoritmos de aprendizaje que son entrenados con datos etiquetados, aunque este último no está completamente definido.

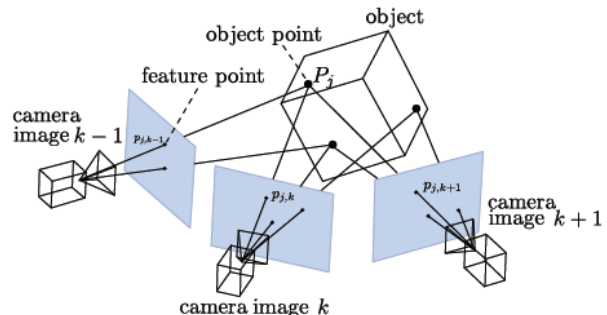


Fig. 1. Conjunto de puntos 3D en una secuencia de imágenes y pose relativa de la cámara.

Al ser la OV un sistema de localización incremental, el error producido no se corrige en ningún momento sino que es acumulado y propagado a futuras estimaciones.

Además del error acumulado siempre se cuenta con valores anómalos (outliers) y la asunción de que el escenario es estático, para combatir estos problemas se utilizan algoritmos robustos como Random Sample Consensus (RANSAC) [4] o técnicas de filtrado incrementando la complejidad computacional del cálculo. Debido a esto, se utilizan diferentes técnicas como la fusión de sensores o algoritmos de aprendizaje con datos designados para la disminución de errores.

Aunque se utilizan secuencias de imágenes para estimar la trayectoria de la cámara, existe mucha información dentro de

las imágenes que no se aprovecha, por ejemplo [5] utiliza un sistema estéreo donde las nubes de puntos 3d sirven para reconocer ubicaciones conocidas y mejorar la estimación de ruta. En [6] se incorpora una base de datos de dimensiones de estructuras conocidas como paredes, edificios, etc.

El propósito de este trabajo es analizar otra alternativa basada en utilizar la información brindada por objetos conocidos dentro de las secuencias de imágenes computadas y de esta forma preservar y/o recuperar la escala a lo largo del trayecto.

Los trabajos actuales de reconocimiento de objetos buscan usar técnicas de aprendizaje automatizado como las redes neuronales convolucionales (CNN, *convolutional neural network*). Uno de los puntos mas interesantes de las CNNs es la habilidad de aprender características de un conjunto de datos y usarlos eficientemente para la tarea asignada.

Muchos trabajos en reconocimiento se han enfocado en recolectar muestras de datos de gran tamaño para generar modelos mas poderosos y mejores técnicas con el objetivo de reducir el sobreentrenamiento. Algunos de las modelos que ha mostrado gran efectividad y eficiencia en los últimos años son AlexNet [7] y GoogleNet [8].

Las CNNs pueden aprender a detectar cientos de objetos a través de millones de imágenes, por lo tanto requieren de modelos con grandes capacidades de aprendizaje. Esta capacidad se puede controlar variando su profundidad y amplitud. A pesar de lo atractivo de las CNNs estas suelen ser relativamente costosas de aplicar en imágenes de alta resolución. Afortunadamente las unidades de procesamiento gráfico (GPU) actuales son lo suficiente potentes como para facilitar el entrenamiento de redes de escalas interesantes.

II. ODOMETRÍA VISUAL MONOCULAR

El sistema de odometría visual monocular que se va a utilizar es uno ya conocido e implementado en trabajos anteriores como [3] y se muestra en la fig. 2. Se realizaron pruebas de distintos algoritmos de extracción de características y se mide el desempeño de cada uno en un conjunto de datos de prueba.

A. Estimación de movimiento

Sea $\mathbf{X} = (x, y, z)^T$ un punto 3D y $x = (u, v)^T$ su proyección en el plano imagen. El modelo de proyección central sería (1)

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} f_u & 0 & u_c & 0 \\ 0 & f_v & v_c & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (1)$$

donde \sim indica equidad hasta cierta escala desconocida, la matriz 3x3 triangular superior \mathbf{K} es la matriz de la cámara modelada a través de los parámetros intrínsecos de la cámara tales como la longitud focal f_u y f_v y el centro focal (u_c, v_c) .

Asumiendo que $\mathbf{X} = (x, y, z)^T$ es un punto estacionario que se mantiene en la misma posición mientras la cámara se

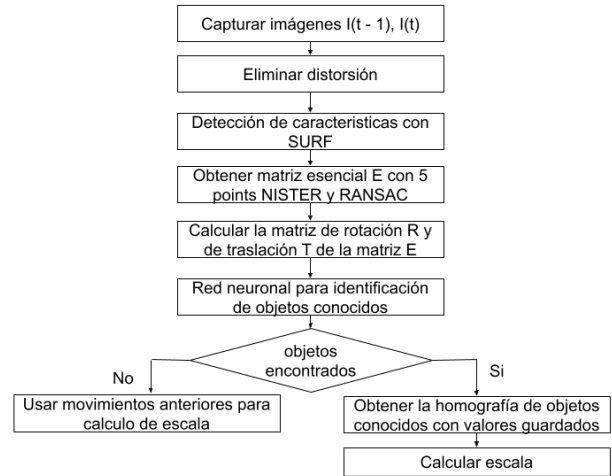


Fig. 2. Etapas del sistema de visión monocular con detección de objetos para estimación de pose de la cámara.

mueve al siguiente instante. La nueva proyección se convierte en la ecuación (2)

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \sim \mathbf{K} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (2)$$

donde la matriz de rotación y el vector de traslación $\mathbf{t} \in \mathbb{R}^3$ representan la transformada euclidiana generada por el movimiento de la cámara.

En el problema de estimación de movimiento, la transformada $\begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix}$ es la incógnita a resolver dadas una serie de correspondencias 3D a 2D tales que $(x, y, z) \leftrightarrow (u', v')$. Esto se conoce como el problema de la perspectiva de n puntos (PnP, *perspective from n points*). Asumiendo toda la matriz de proyección $\mathbf{P} = \mathbf{K} \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix}$ como la incógnita, la solución se puede obtener resolviendo un sistema lineal homogéneo. Esta estrategia se conoce como la transformada lineal directa (DLT, *direct linear transform*).

III. DETECCIÓN DE CARACTERÍSTICAS

Existen dos tipos de detectores de características, los detectores de esquinas y los que detectan áreas o puntos de interés. Los detectores de esquinas usados comúnmente son Moravec [1], Forstner [9], Harris [10], Shi-Tomasi [11], y FAST [12]. Estos detectores suelen no ser robustos ante cambios de escala y de ángulo de visión. Los detectores de puntos de interés son mas efectivos en cuanto a estos cambios por lo que pueden ser mas efectivos al momento de extraer características y emparejarlas en una secuencia de imágenes. Entre los detectores de puntos de interés mas comunes tenemos SIFT [13], SURF [14] y ORB [15]. Las características obtenidas en un instante deben ser emparejadas con las características obtenidas en el instante anterior. Para el seguimiento de características utilizamos el rastreador

KLT debido a su robustez ante movimientos pequeños y movimientos mas grandes.

A. FAST

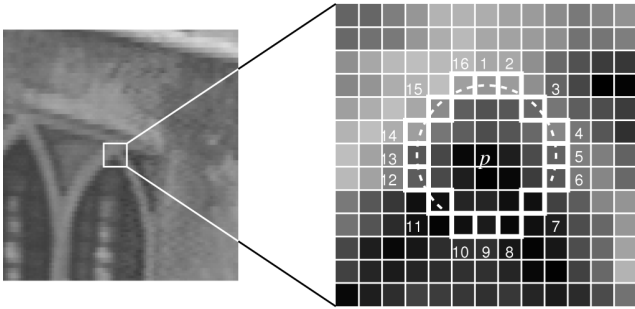


Fig. 3. FAST, punto de interés cuya vecindad tiene una secuencia de pixeles que supera el umbral definido.

Features from Accelerated Segment Test (FAST) es un algoritmo que se encarga de comparar la intensidad del punto de interés p con la intensidad de su vecindad de pixeles como podemos observar en la fig. 3. Para detectar una esquina es necesario que una secuencia de un cierto número de pixeles de la vecindad superen un umbral de intensidad. Este algoritmo se destaca por su velocidad de ejecución que es menor a otros detectores de esquinas pero no es efectivo ante altos niveles de ruido o cambios en escala. Su desempeño correcto depende de una correcta configuración del umbral de intensidad.

B. SURF

El algoritmo *Speed Up Robust Features* (SURF) se encarga de detectar y describir puntos de interés de una manera mas eficiente que su antecesor *Scale Invariant Feature Transform* (SIFT). Como su nombre lo indica, incluye mejoras que aceleran la ejecución y permiten obtener una mayor efectividad. A diferencia de SIFT que se basa en operaciones que utilizan las diferencias de gaussianas, el algoritmo surf se basa en otra aproximación realizada mediante imágenes integrales. La robustez del algoritmo se basa en identificar la orientación del punto de interés y su vecindad de pixeles. La orientación se obtiene mediante operaciones gaussianas en la vecindad circular de pixeles para obtener un vector que describa al punto de interés de forma única.

C. ORB

Oriented FAST and Rotated BRIEF (ORB) es otra alternativa que permite la extracción y descripción de puntos de interés. Es una combinación de la detección FAST con la técnica de descripción de puntos de interés *Binary Robust Independent Elementary Features* (BRIEF). Para lograr la efectividad ante variaciones en escala el algoritmo FAST se aplica en una pirámide con diferentes escalas de la imagen. ORB tambien utiliza una variante de BRIEF capaz de trabajar ante rotaciones. Por lo tanto ORB es robusto ante cambios de escala y rotación. Un punto importante de ORB es el reducido costo computacional que presenta, por lo tanto, es una opción importante al trabajar con sistemas de recursos limitados.

D. Eliminación de valores anómalos

Debido a que la extracción y el emparejamiento de características en imágenes secuenciales no es perfecta, es necesario una forma de eliminar valores anómalos (outliers). RANSAC es un algoritmo iterativo que toma un muestreo aleatorio de nuestros emparejamientos para obtener la matriz esencial, posteriormente pondera esta matriz con el resto de correspondencias. Después de un número de iteraciones el algoritmo termina dando como resultado la matriz con mayor ponderación respecto a los puntos evaluados. Al tener un carácter aleatorio este algoritmo no obtiene siempre el mismo resultado pero en la gran mayoría de los casos es el óptimo.

IV. DETECCIÓN DE OBJETOS

Debido a la dificultad de obtener la escala en sistemas monoculares se ha optado por la creación de una base de datos de dimensiones de objetos conocidos y una red neuronal convolucional entrenada para la detección de dichos objetos. El objetivo es localizar los objetos en la escena computada y a través de una matriz de homografía calcular las dimensiones relativas del objeto en la escena. Si el objeto es plano, la pose de la cámara se puede obtener usando su homografía.

AlexNet es una arquitectura de CNN con 8 capas que consisten de capas de convolución, capas de activación, capas de max pooling y capas densas. Teniendo 5 capas de convolución, 3 de max pooling y 3 densas que terminan en una capa de salida del tipo dense-softmax. El tamaño y orden de las capas lo podemos encontrar en la tabla I.

TABLE I
ARQUITECTURA DE CAPAS DE ALEXNET

Capa	Características	Tamaño	Kernel	Stride	Activación
Input	1	227x226x3	-	-	-
Conv.	96	55x55x96	11x11	4	relu
Pool	96	27x27x96	3x3	2	relu
Conv.	256	27x27x256	5x5	1	relu
Pool	256	27x27x256	3x3	2	relu
Conv.	384	13x13x384	3x3	1	relu
Conv.	384	13x13x384	3x3	1	relu
Conv.	384	13x13x384	3x3	1	relu
Pool	256	6x6x256	3x3	2	relu
Dense	-	9216	-	-	relu
Dense	-	4096	-	-	relu
Dense	-	4096	-	-	relu
Output	-	1000	-	-	Softmax

^aSample of a Table footnote.

En [8] se entrena el modelo con 15 millones de imágenes catalogadas con aproximadamente 22 mil categorías. Los resultados obtenidos muestran errores de 15.3% al aplicarse en los datos de prueba. La estructura se puede aplicar a cualquier otro conjunto de datos de imágenes bidimensionales.

Una vez detectado un objeto conocido se puede trabajar el área de interés y obtener los puntos características y sus descriptores para obtener una homografía proyectiva con las características y dimensiones conocidas. Teniendo la homografía y el objeto detectado podemos calcular la pose de la cámara de una forma mas efectiva.

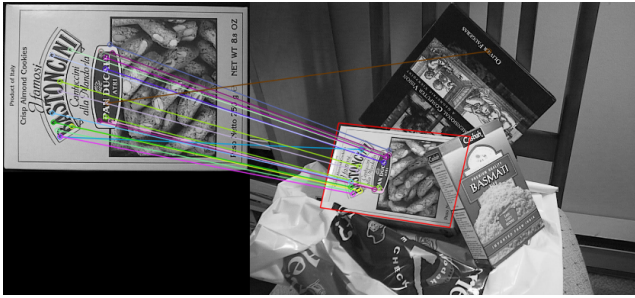


Fig. 4. Emparejamiento de puntos usando homografía.

V. EXPERIMENTOS

Los experimentos se realizaron en el conjunto de datos KITTI [16], el conjunto de datos se conforma de una serie de fotogramas que se obtienen de un recorrido en la calle como se observa en la fig. 5. En dichas pruebas se obtuvieron resultados donde se muestra el error en la estimación de ruta. La elección del algoritmo de extracción y descripción de características no es algo trivial y debe ser tomado en cuenta. Los experimentos se realizaron con el fin de encontrar el extractor de características que se desempeñe mejor. Los resultados se comparan con los valores verdaderos obtenidos por medio de sensores láser.

El detector de esquinas FAST a pesar de ser un detector de esquinas incapaz de trabajar con cambios de escala y rotaciones demostró un buen desempeño como se muestra en la tabla II. Cabe mencionar que el movimiento que realiza la cámara entre un instante t y un instante $t + 1$ no es de gran tamaño, de igual manera el conjunto de datos no contiene cambios bruscos en el ángulo de visión.

TABLE II
DESEMPEÑO DEL ALGORITMO DE OV CON FAST

Frame	Error absoluto en metros	Error acumulado en metros
100	0.215888	2.333499
400	3.820372	47.092479
700	9.252882	235.379759
1000	13.413948	588.010515

En el caso del algoritmo SURF los resultados que podemos ver en III demostraron ser mejores que los demás detectores de características utilizados. Demuestra un desempeño similar a FAST en los primeros frames pero conforme la cámara continua alejándose del origen el resultado se desvía por una cantidad menor que en los otros algoritmos.

TABLE III
DESEMPEÑO DEL ALGORITMO DE OV CON SURF

Frame	Error absoluto en metros	Error acumulado en metros
100	0.387551	2.587993
400	1.897696	51.734162
700	3.305652	129.757265
1000	3.415369	218.565254

La tabla IV nos muestra que ORB fue el de peor desempeño ante la secuencia analizada, aunque al inicio se mantiene en un desvío menor, conforme la cámara rota y aumenta de velocidad en sus movimientos el error aumenta a valores bastante considerables.

TABLE IV
DESEMPEÑO DEL ALGORITMO DE OV CON ORB

Frame	Error absoluto en metros	Error acumulado en metros
100	2.334228	11.414300
400	25.724679	276.235144
700	9.810927	1382.001411
1000	74.890377	2823.852631

Es necesario el análisis de integración de la red neuronal convolucional al algoritmo de OV para poder calcular la escala y evaluar los resultados de la estimación de pose respecto a la estimación sin la detección de objetos. Los experimentos se deberán realizar en un ambiente controlado para minimizar obstáculos que puedan alterar el resultado.



Fig. 5. Frames 90, 100 y 110 de la secuencia del KITTI dataset.

VI. CONCLUSIONES

La efectividad de la OV es dependiente de los algoritmos de extracción de características utilizados, aún con la aplicación del algoritmo correcto existe lugar a la mejora de precisión. La estimación de pose y la OV son campos que pueden mejorarse aprovechando información adicional a la obtenida con los métodos tradicionales. Las CNN presentan una alternativa que merece explorarse mas a fondo para la integración en los métodos de estimación de pose. Mediante métodos de triángulos similares y homografía es posible obtener la escala en una secuencia de imágenes.

REFERENCES

- [1] Moravec, H. P. (n.d.). Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.
- [2] Scaramuzza, D., & Fraundorfer, F. (2011). Tutorial: Visual odometry. IEEE Robotics and Automation Magazine, 18(4), 80–92.
- [3] Chien, H. J., Chuang, C. C., Chen, C. Y., & Klette, R. (2016). When to use what feature? SIFT, SURF, ORB, or A-KAZE features for monocular visual odometry. International Conference Image and Vision Computing New Zealand, 0.

- [4] Fischler, M. A., & Bolles, R. C. (1980). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.
- [5] Mo, J., & Sattar, J. (n.d.). Place Recognition for Stereo Visual Odometry using LiDAR Descriptors.
- [6] Hilsenbeck, S., Moller, A., Huitl, R., Schroth, G., Kranz, M., & Steinbach, E. (2012). Scale-preserving long-term visual odometry for indoor navigation. 2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings.
- [7] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (n.d.). ImageNet Classification with Deep Convolutional Neural Networks.
- [8] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going Deeper with Convolutions.
- [9] Forstner, W. (1987). A feature based correspondence algorithm for image matching. From Analytical to Digital. Proc. Symposium, ISPRS, Commission III, Rovaniemi, 1986, Vol. 3, 150–166.
- [10] Harris, C. G., Pike, J. M., Research, P., & Manor, R. (n.d.). 3D POSITIONAL INTEGRATION FROM IMAGE SEQUENCES.
- [11] Shi, J., & Tomasi, C. (1993). Good Features to Track Introduction.
- [12] Trajković, M., & Hedley, M. (1998). Fast corner detection. Image and Vision Computing, 16(2), 75–87.
- [13] Lowe, D. G. (1999). Object recognition from local scale-invariant features. Proceedings of the IEEE International Conference on Computer Vision, 2, 1150–1157.
- [14] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2007). Speeded-Up Robust Features (SURF).
- [15] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (n.d.). ORB: an efficient alternative to SIFT or SURF.
- [16] Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. International Journal of Robotics Research, 32(11), 1231–1237.

Implementación parcial del algoritmo SURF en FPGA

ISC. Elohim Arturo Torres Manriquez*, Dr. Saúl Martínez Díaz*,
Dr. Israel Marcos Santillán Mendez*, y MSC. José Luis Gómez Torres*

*División de Estudios de Posgrado e Investigación

TECNM /ITLP

La Paz, Baja California Sur, México

Email: torresmsc18@itlp.edu.mx saul.md@lapaz.tecnm.mx israel.sm@lapaz.tecnm.mx jose.gt@lapaz.tecnm.mx

Resumen—En el presente artículo se presenta el diseño y los resultados de la implementación parcial del algoritmo de visión artificial SURF utilizando un FPGA. Para la interconexión de los datos entre el Microprocesador y la lógica reconfigurable del FPGA, se utilizan las herramientas Xillybus, el sistema operativo Xilinx, y los lenguajes C++ y VHDL.

Palabras clave—Visión artificial, SURF, C++, Xillybus, Xlinux, VHDL, FPGA, Microprocesador, PS, PL, Procesamiento de imágenes.

I. INTRODUCCIÓN

Actualmente los algoritmos de visión artificial están ganando un amplio terreno en distintos campos. Hoy en día casi cualquier persona tiene acceso al uso de aplicaciones con estos algoritmos por ejemplo el reconocimiento facial, o alguna aplicación para filtro de imágenes como las que ya tienen la mayoría de *smartphones*. El potencial de este tipo de tecnologías es impresionante, pues permiten el desarrollo de aplicaciones mucho más complejas como la conducción autónoma de vehículos, mapas de profundidad, el reconocimiento de objetos por forma, color, etc. Desafortunadamente aun con los avances que se han tenido con el uso de estos algoritmos existen grandes problemas en los cuales trabajar, pues la mayoría de algoritmos no tienen un equilibrio entre robustez y velocidad de respuesta. La mayoría de estos son muy sensibles al cambio de iluminación, al ruido, a la escala de la imagen, etc. Mientras que los que son menos sensibles a este tipo de condición tardan más tiempo en brindar una respuesta. La estrategia que se presenta en este artículo, es la de implementar un algoritmo bastante eficiente que es el SURF (Speeded-Up Robust Features) por sus siglas en inglés. Este algoritmo está basado en otro llamado SIFT (Scale-invariant feature transform), pero se diferencia de este porque utiliza el enfoque del determinante de la matriz Hessiana para detectar puntos de interés. Estos puntos de interés pueden ser características de la imagen importantes a resaltar, como por ejemplo esquinas, bordes, etc. son útiles para casos como el de detectar una estructura diferente en comparación con el resto en una imagen, lo cual podría representar un obstáculo, y que es muy útil de detectar en tareas como la navegación autónoma.

También puede utilizarse para detectar un mismo objeto encontrando coincidencias en los puntos de interés de una o más imágenes. Se plantea el uso de un FPGA porque tiene una gran capacidad de procesamiento de datos. Además tiene la propiedad de trabajar de forma paralela, a nivel de hardware directamente. Lo que se hace es construir un circuito electrónico y no desarrollar software. Tomando en cuenta que SURF es uno de los algoritmos más robustos para detección de puntos de interés, se utiliza el FPGA para tratar de solventar uno de los puntos débiles de este algoritmo que es el tiempo de procesamiento.

Debido a la gran capacidad de cómputo de los FPGA y al poco consumo de energía que estos requieren, se han propuesto ya algunos trabajos para acelerar algoritmos, como por ejemplo el del programa ExoMars [1] en donde se presenta el codiseño Hardware/Software del algoritmo SURF para su uso en la navegación móvil, teniendo en cuenta las especificaciones del proyecto SPARTAN para el programa ExoMars de la ESA.

Otro artículo que trata directamente sobre la implementación del algoritmo SURF en el FPGA [2] donde se menciona en las conclusiones que se tuvieron que hacer algunos cambios para optimizar el algoritmo para la implementación en la lógica FPGA. También se menciona en los resultados que ha habido una gran preocupación por la posible degradación del algoritmo debido a la pérdida de precisión en la fase de detección. El artículo [8] implementa un algoritmo en el que los resultados arrojaron que puede detectar rostros a velocidades de aproximadamente dos órdenes de magnitud (100X) más altas que la implementación de software que se ejecuta en una CPU de 2.4GHz.

II. MARCO TEÓRICO

SURF El algoritmo SURF es un detector de puntos de interés y generador de descriptores invariante a la rotación y a la escala. Este tiene cuatro fases, las cuales se muestran en la Figura 1 [2].

1. Generación de la imagen integral. Esta se construye a partir de la suma de intensidades de todos los píxeles de la imagen, y que sirve para acelerar procesos posteriores del algoritmo.

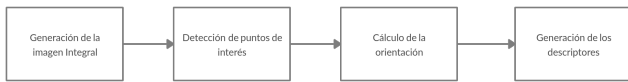


Figura 1. Diagrama etapas del algoritmo SURF.

2. Detección de los puntos de interés. Este algoritmo utiliza la aplicación de filtros Gaussianos y el determinante de la matriz Hessiana para detectar estos puntos a diferentes escalas.
3. Cálculo de la orientación. Una vez encontrados los puntos de interés, se calcula la orientación de estos utilizando el método de Wavelet Haar en x e y . Con esto el algoritmo se vuelve invariante a la rotación, de modo que si la imagen se gira, ambas imágenes se alinean con respecto a ese único punto clave.
4. El último paso es la generación de los descriptores, los cuales tienen información de los píxeles vecinos a los puntos de interés que se encontraron previamente.

Pl Por sus siglas en inglés (Programmable Logic) es la parte interna del FPGA donde se encuentran las celdas que serán configuradas para una función específica dependiendo de como el programador las describa. Esto se realiza con un lenguaje de descripción de Hardware como VHDL, Verilog o ABEL [7].

Ps Por sus siglas en inglés (Processing System) es la parte donde se encuentra el Microprocesador, el cual se encuentra conectado al Pl, y es bastante útil para recibir datos del entorno exterior, por ejemplo conectando sensores a este para que posteriormente estos datos sean procesados por el Pl [7].

Xillynux Es una distribución de Linux, basado en Ubuntu LTS 12.04, el cual incluye un controlador que forma parte del conjunto de Xillybus y, por lo tanto, está preparado para funcionar de forma óptima haciendo uso del mismo [4]. Dicho controlador permite al sistema operativo enviar y recibir datos al Pl, utilizando los ficheros `/dev/xillybus_write_8` y `/dev/xillybus_read_8` para escritura y lectura respectivamente.

Xillybus Este engloba un módulo FPGA, así como el controlador anteriormente mencionado incluido en Xillynux. Consecuentemente, Xillybus es el encargado de hacer posible la comunicación entre el FPGA y sistema operativo [4].

En la Figura 2 se muestra un ejemplo simple del flujo de datos entre el FPGA y el Host alojado en el sistema operativo. Se utiliza de intermediario Xillybus y este a su vez se apoya de estructuras de datos FIFOs, las cuales pueden ser de 8 o 32 bits. En este ejemplo se muestran las señales que se utilizan para que Xillybus tenga un control sobre las FIFOs, conocer su estado y si se puede escribir en ellas [4].

Host : En este proyecto el Host está integrado por un programa hecho en C++ junto con la librería OpenCV

Flujo de datos: Xillybus

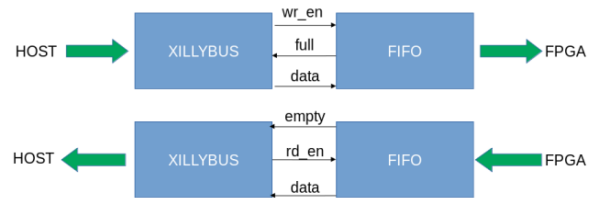


Figura 2. Ejemplo de transmisión de datos entre FPGA y Host mediante uso de FIFOs y Xillybus.

3.2 que se ejecutan en el sistema operativo ya mencionado Xillynux, esto se utiliza para poder leer las imágenes que serán procesadas por el PL del FPGA.

Imagen Integral : Esta tiene el fin de reducir tiempo de procesamiento y obtener en un solo proceso la suma de intensidades de todos los píxeles de la imagen, se obtiene a partir de la suma acumulada de la fila actual, y el valor de la imagen integral en la fila anterior. La imagen integral se utiliza en partes posteriores del algoritmo y se calcula como se muestra en la ecuación. 1.

$$I_{\Sigma}(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad (1)$$

Donde los términos x e y son las posiciones de la matriz de la imagen integral que se irá calculado, y los términos j e i son las posiciones de la matriz de la imagen original.

Una vez obtenida la imagen integral es necesario leer solo cuatro píxeles para obtener la intensidad de la superficie de una cierta región de la imagen, independientemente de su tamaño [2].

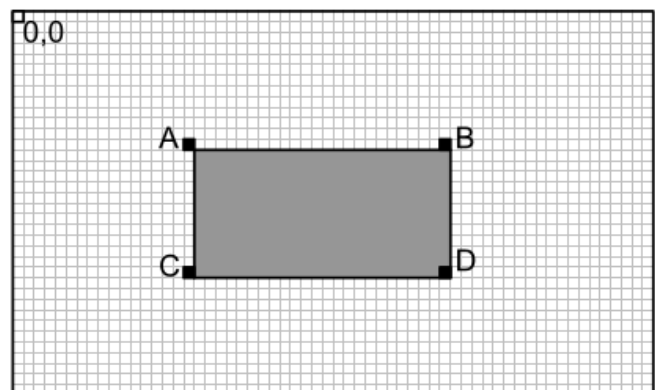


Figura 3. Integral de una región de la imagen, calculada con solo 4 píxeles.

En la Figura 3, se muestra un ejemplo del cálculo de la integral de la zona sombreada, leyendo solo cuatro píxeles y aplicando la operación, mostrada en la ecuación 2.

$$\Sigma = I_{\Sigma}(A) + I_{\Sigma}(D) - I_{\Sigma}(C) - I_{\Sigma}(B) \quad (2)$$

En la ecuación 2, los términos A, B, C Y D, representan la imagen integral en cada uno de estos puntos.

Filtro Gaussiano : Los filtros Gaussianos se utilizan para suavizar la imagen, lo cual da el efecto borroso y que la imagen se encuentra a una mayor distancia, además de este efecto los filtros Gaussianos ayudan a atenuar cierto ruido en las imágenes [9].

Para lograr el efecto de escalado en las imágenes obtenidas, se hace uso de filtros Gaussianos, porque el algoritmo detecta puntos de interés de una misma imagen a diferentes escalas y evalúa cual punto de interés es más significativo a qué escala .

Este algoritmo detecta puntos de la imagen que presenten alguna variación con respecto al resto, lo cual los hace interesantes. El mecanismo que se utiliza para detectar esta variación es el determinante de la matriz Hessiana, la cual sirve como indicador de los cambios locales que suceden alrededor de algún punto con respecto al resto, al final se seleccionan los puntos de interés que tengan un determinante mayor.

El algoritmo utiliza la segunda derivada para encontrar máximos y mínimos, esto es encontrar los puntos en los que hay un cambio local significativo con respecto al resto de la imagen. La derivada con respecto a x encuentra estos cambios de forma horizontal, la derivada con respecto a y encuentra cambios de forma vertical y la derivada con respecto a x e y detecta cambios en forma diagonal. Para combinar estos cambios, se propone utilizar el determinante de la matriz Hessiana, que se construye a partir de estas derivadas, la cual está dada en la ecuación 3.

$$H(x, y) = \det \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (3)$$

En la ecuación 3 se muestran los componentes de la matriz Hessiana, que son las segundas derivadas parciales ya mencionadas, respecto a 'x' o 'y' de la imagen en un punto dado [3].

Kernel : La aplicación de estos filtros se realiza mediante una convolución 2D de la imagen con su respectivo kernel, este kernel se obtiene con la operación que se muestra en la ecuación 4 [9].

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4)$$

En la ecuación 4 x e y representan los índices del kernel que se va calculando, σ es un parámetro que podemos variar dependiendo del grado de suavizado que se desee y π e e son constantes conocidas.

Para obtener distintas escalas de la imagen al aplicar la convolución, se deben utilizar diferentes kernels, estos se

obtienen modificando el parámetro σ de la ecuación cuatro. Cuando se aumenta el valor de este parámetro y se obtiene la imagen filtrada, se consigue el efecto de escalado, y se resaltan estructuras mayores en la imagen. El kernel es, una matriz de coeficientes estrictamente de tamaño m /times m .

Convolución : Una vez que se obtiene el kernel, se aplica una convolución 2D, esta se puede interpretar como un recorrido de izquierda a derecha y de arriba hacia abajo, traslapando cada coeficiente del filtro (o kernel) sobre la imagen. En cada traslape se multiplican los coeficientes por el píxel sobre el cual está, y luego se suman todos los productos. Esta operación se muestra en la ecuación 5 [9].

$$y(m, n) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h(k, l) * x(m - k, n - l) \quad (5)$$

Uso de memoria RAM en FPGA : En los FPGAs se manejan diferentes tipo de arquitecturas dependiendo del fabricante, pero independientemente de esto, existen dos maneras de trabajar con las memorias RAM.

1. Memoria distribuida. En este tipo de memoria lo que se hace es que se interconectan los elementos lógicos del FPGA para formar una memoria. En este tipo de memoria los datos se leen de forma síncrona. Es decir tenemos que esperar al siguiente ciclo de reloj una vez que hayamos puesto la dirección de memoria que queremos leer en la salida.

2. Memoria de bloque. En este tipo de memoria, lo que se hace es utilizar los bloques de memoria que ya están en el FPGA, para no utilizar los recursos lógicos del mismo. Este tipo de memoria puede usarse de forma síncrona como el caso anterior, o de forma asíncrona, es decir, no tenemos que esperar al siguiente ciclo de reloj en el que se pone la dirección de memoria para poder leer el dato en la salida.

En cuanto a funcionamiento ambas son muy similares. Pueden crearse de uno o dos puertos de lectura y uno de escritura. La principal diferencia es la forma de leer los datos [6].

La memoria de bloque tiene la ventaja con respecto a la distribuida en que puede usarse de forma síncrona o asíncrona y además no utiliza elementos lógicos del FPGA que podrían utilizarse para otras tareas.

III. DESARROLLO

III-A. Aspectos generales

El lenguaje de Hardware reconfigurable utilizado en este proyecto es el VHDL, para realizar la síntesis, implementación y generar el archivo bitstream, se utilizó el software Vivado 2018.3, La memoria RAM que se implementó fue la memoria de bloque con anterioridad, en esta la lectura de datos asíncrona y se utilizó principalmente para guardar la imagen integral.

El FPGA en el que se realizaron las pruebas fue una Zybo Zync 7000 Development Board, con las siguientes características. [5]

- Procesador Cortex-A9 de doble núcleo de 650 Mhz.
- Controlador de memoria DDR3 con 8 canales DMA.
- Controladores periféricos de alto ancho de banda: 1 G Ethernet, USB 2.0, SDIO.
- Controlador periférico de bajo ancho de banda: SPI, UART, I²C.
- Lógica reprogramable equivalente a FPGA Artix-7.
 - Celdas lógicas de 28 K.
 - Bloque RAM de 240 KB.
 - 80 DSP.
 - Convertidor analógico digital (XADC) en chip de canal doble, 12 bits y 1 MSPS.

En la cual se implementó el sistema operativo Xillynx, este se encuentra cargado dentro de una memoria microSD que se inserta directamente en el FPGA en el conector J4.

En esa misma memoria se cargó en la raíz el archivo .bit, este archivo es el que nos proporciona el programa Vivado, en el cual se encuentra toda la estructura de los bloques que se desarrollaron, así como las IPs de las FIFOs y Xillybus.

III-B. Interconexión de datos entre PL y PS del FPGA

Para lograr la interconexión entre el PS y el PL, se hace uso de distintas herramientas. Una de estas es un sistema operativo el cual se encuentra cargado en una memoria microSD y que es insertada en el FPGA, este sistema operativo es el Xillynx.

Además de usar este sistema operativo, también es necesario el uso del Xillybus ya que este, actuará de intermediario en el proceso de transmisión de datos, estableciendo dos flujos de comunicación: uno en la dirección FPGA-Host y otro para Host-FPGA. Se utilizan FIFOs de lectura/escritura, para la transmisión de datos.

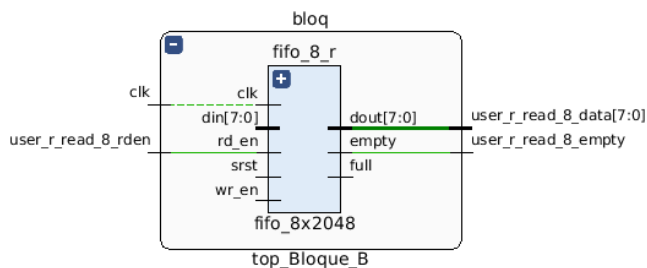


Figura 4. Diagrama FIFO de 8 bits.

En el diagrama de bloque de la Figura 4 proporcionado por el software ya mencionado Vivado, se muestra una FIFO de lectura de datos, implementada en el proyecto, las señales más importantes son *empty*, *full*, *din[7:0]* y *dout[7:0]*, las cuales indican si la FIFO está vacía, llena, y los 8 bits de datos de entrada o de salida respectivamente.

Otro aspecto importante a mencionar es que se creó un programa en lenguaje C++ el cual forma parte del Host junto con el sistema operativo, la función principal de este programa es leer las imágenes, convertirlas

a escala de grises y cargar los datos a un Buffer. Una vez cargados los datos al buffer estos son enviados al fichero `/dev/xillybus_write_8`, esta dirección se utiliza solo para escribir los datos a la FIFO de 8 bits, es necesario mencionar que se utilizan FIFOs de 8 y no de 32 bits que también están disponibles, porque los valores de intensidad de las imágenes están en rango de 0 a 255 y solo se requieren 8 bits para almacenarlos. Posteriormente se realiza el procesamiento de las imágenes en el PL y una vez hecho esto, se utiliza el fichero `/dev/xillybus_read_8` únicamente para leer los datos de la FIFO, estos datos son las salidas del módulo principal del FPGA ya procesados, estos se envían de regreso al Host, en el cual se cargan a una matriz y se despliega su contenido para poder validar los resultados. En la Figura 5 se muestra un diagrama de flujo básico, del funcionamiento del programa en C++.

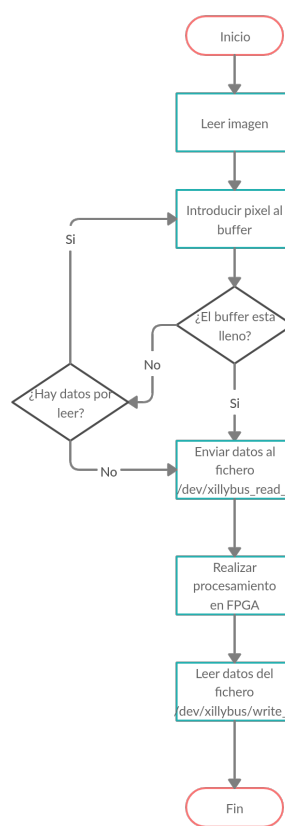


Figura 5. Diagrama de flujo programa en C++.

Esto libera al programador del FPGA de la preocupación acerca de cuándo Xillybus estará disponible para recibir esos datos, ya que este se encargará de ir leyendo las FIFOs cuando no tenga otras tareas que atender, mientras que por el otro lado el programa del Host va relleniéndolas con información. De esta manera se consigue la independencia entre la aplicación FPGA y el Host, cada uno escribe o lee en el momento que pueda hacerlo sin depender del otro. [4]

III-C. Creación de la imagen integral

El primer paso del algoritmo es el cálculo de la imagen integral, la implementación del mismo en el FPGA, se muestra en el diagrama de flujo de la Figura 6.

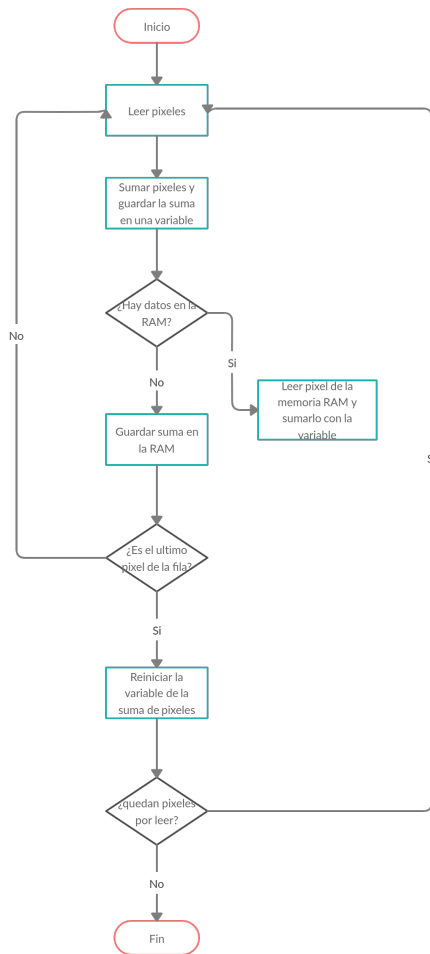


Figura 6. Diagrama del cálculo de la imagen integral en FPGA.

En este proceso, se leen los pixeles que se van recibiendo de la FIFO, se van sumando y almacenando en una variable x . Si estos pixeles pertenecen a la primera fila de la imagen, quiere decir que aun no se han introducido datos en la memoria RAM, que es donde se almacena la imagen integral. Como no hay ningún dato de la imagen integral, cada suma se guarda en una dirección de la memoria RAM. Al terminar de sumar todos los datos de la primera fila se reinicia la variable (x), y se vuelve a comenzar a sumar los pixeles conforme se vayan recibiendo. Como ahora sí hay datos en la imagen integral, cada vez que se realice una suma, se debe leer sólo un dato de la imagen integral de manera consecutiva y sumarlo a la variable (x), posteriormente este dato se guarda en la siguiente posición disponible de la RAM. Al terminar de leer todos los pixeles de la misma fila, se vuelve a reiniciar la variable (x), y se repite el proceso hasta que no haya mas pixeles que leer.

III-D. Aplicación de filtro Gaussiano

Para implementar el filtro Gaussiano en el FPGA, primero se obtuvieron los coeficientes del kernel, vistos en la ecuación 4. Para obtenerlos se utilizó la ayuda del software Matlab. Con el comando: `fspecial('gaussian',3,1)`, se creó una matriz de tamaño 3x3, con una sigma de 1.0.

Debido a que en el FPGA no es tan sencillo el manejo de datos en punto flotante, cada dato de la matriz se multiplicó por 2 elevado a 15, y se transformó a binario tomando la parte entera. Se considero tomar este valor para redondear el valor que se tenía en punto fijo, y poder representarlo con una cadena de bits en el FPGA tratando de perder la menor cantidad de información. Como algunos valores en punto flotante no tienen representación exacta en binario, es imposible no perder algo de información en el proceso. Por esta razón se utilizó una potencia de base 2 y un valor elevado para dicha operación. Este valor puede ser representado como una secuencia de bits en el FPGA como se muestra en la Figura 7. Estos resultados son los que se utilizaron en VHDL para definir el kernel.

```

kernel(0,0) <= "0100110011100"; --0.0751*(2^15)
kernel(0,1) <= "0111111011000"; --0.1238*(2^15)
kernel(0,2) <= "0100110011100"; --0.0751*(2^15)
kernel(1,0) <= "0111111011000"; --0.1238*(2^15)
kernel(1,1) <= "1101000100011"; --0.2042*(2^15)
kernel(1,2) <= "0111111011000"; --0.1238*(2^15)
kernel(2,0) <= "0100110011100"; --0.0751*(2^15)
kernel(2,1) <= "0111111011000"; --0.1238*(2^15)
kernel(2,2) <= "0100110011100"; --0.0751*(2^15)
    
```

Figura 7. Kernel Gaussiano con sigma=1.0 transformado a binario.

El proceso en el FPGA para la aplicación del filtro Gaussiano es el siguiente.

1. Leer y se guardar temporalmente las primeras 3 filas de la imagen.
2. Se aplica la ecuación 5, el resultado de esta operación es un pixel ya filtrado de la imagen. Este se envía a la salida, para ser mostrado en pantalla con ayuda del programa en C++ ya mencionado con anterioridad.
3. Se lee un nuevo pixel de la imagen y se coloca en la fila que corresponda. Empezando desde la primera a la tercera, y dependiendo del orden en que sea leído el pixel, pues cada nuevo pixel irá remplazando los que ya fueron utilizados para realizar el filtro.
4. Hacer es un recorrido en una posición, de izquierda a derecha, de las 3 filas de la imagen almacenada. Una vez que se llegue al final, se vuelve a reiniciar el recorrido de izquierda a derecha, desde la primera columna.
5. Se vuelve a iniciar el proceso desde el paso 2, pero ahora los pixeles que se tienen en estas 3 filas han sido remplazados por los que se han leído en el paso 3.

IV. RESULTADOS

Los diagramas de bloques generados con el software Vivado después de realizar la síntesis y la implementación,

se muestran a continuación.

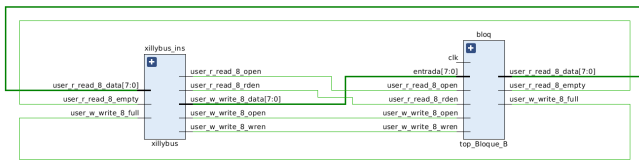


Figura 8. Diagrama interconexión Xillybus top_Bloque_B.

En la Figura 8, se muestra la interconexión del bloque principal, que es el Xillybus y otro bloque denominado top_Bloque_B este sirve como intermediario entre el bloque principal y el resto de bloques. El bloque Xillybus, es el que interactúa directamente con los datos recibidos por el Host y que se encuentra en más alto nivel de jerarquía.

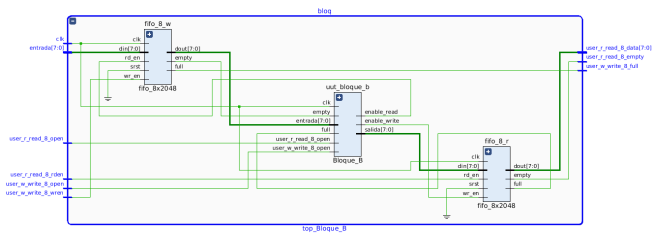


Figura 9. Diagrama interconexión Bloque_B.

En la Figura 9 se puede observar las interconexiones del Bloque_B con las FIFOs, este bloque denominado Bloque_B es donde se implementó el cálculo de la imagen integral y la aplicación del filtro Gaussiano a la imagen.

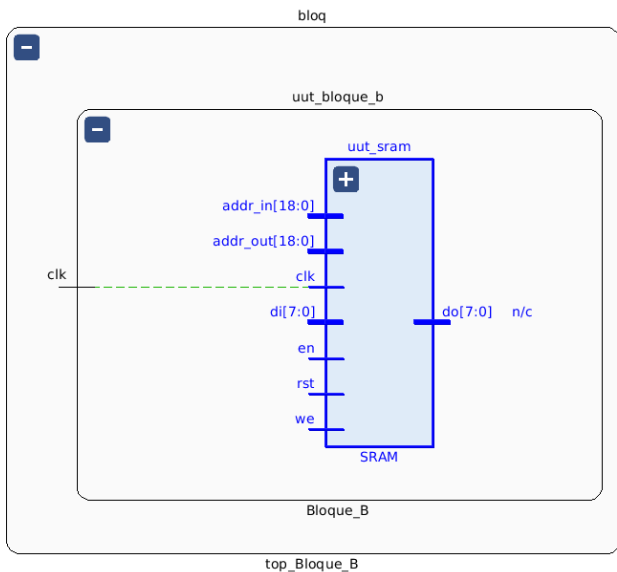


Figura 10. Diagrama memoria RAM.

En la Figura 10 se observa el diagrama de la memoria RAM, que es donde se va guardando y leyendo la imagen

integral, esta se implementó con dos direcciones de lectura, una de entrada, una de salida y señales opcionales, para el reset, deshabilitar o habilitar.

Y por último el diagrama del sistema completo se puede apreciar en la Figura 11.

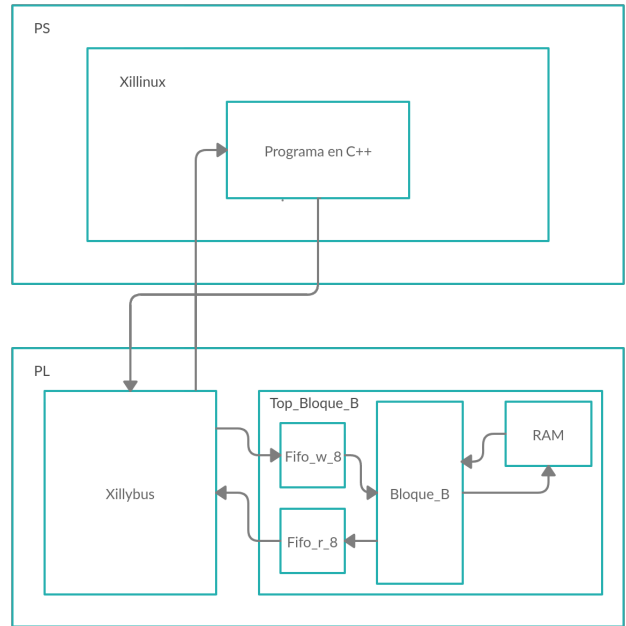


Figura 11. Diagrama interconexión del sistema.

En la Figura 12 se muestra la imagen original, y en la Figura 13 se muestra el resultado de aplicar el filtro Gaussiano en el FPGA.



Figura 12. Imagen original antes de ser filtrada.

En la tabla comparativa de desempeño, se pueden observar los resultados de las pruebas realizadas, entre un programa en C++ en un Microprocesador y el FPGA.

V. CONCLUSIONES

Como se puede apreciar en la tabla comparativa, el FPGA redujo los tiempos de procesamiento de los algoritmos implementados con la misma imagen, la cual tiene un tamaño de 300 x 300 pixeles. Hasta el momento puede



Figura 13. Resultados de aplicación del filtro gaussiano en el FPGA.

 Tabla I
 TABLA COMPARATIVA DE DESEMPEÑO

Dispositivo	Algoritmo	Duración segundos	Pixeles img
Microprocesador	Imagen Integral	0.559256	457 x 630
FPGA	Imagen Integral	0.544164	457 x 630
Microprocesador	Filtro Gaussiano	0.66465	457 x 630
FPGA	Filtro Gaussiano	0.557805	457 x 630
Microprocesador	Imagen Integral	0.914593	1134 x 850
FPGA	Imagen Integral	0.831785	1134 x 850
Microprocesador	Filtro Gaussiano	2.03229	1134 x 850
FPGA	Filtro Gaussiano	0.885309	1134 x 850

parecer que no es mucho lo que se redujo en comparación con lo implementado en el Microprocesador, hecho en lenguaje C++, pero hay que tomar en cuenta 2 cuestiones, la primera es que en el FPGA aumenta un poco el tiempo de procesamiento, porque depende del programa en C++ del Host para poder recibir y devolverlos al Host. Aun así, el tiempo de respuesta es mas rápido. Y la segunda cuestión es que al momento de implementar las demás etapas de algoritmo SURF este tiempo se irá incrementando en ambos dispositivos, pero lo lógico es que el FPGA irá tomando una mayor ventaja.

El algoritmo implementado completamente en el Microprocesador en lenguaje C++ tarda aproximadamente 16.1913 segundos. Como ya se mencionó falta aun implementar las otras etapas del algoritmo SURF en el FPGA para poder realizar más comparaciones de desempeño, pero hasta el momento se cumplen los resultados esperados.

REFERENCIAS

- [1] G. Lentaris, I. Stamoulias, D. Diamantopoulos, K. Siozios, and D. Soudris, An FPGA implementation of the SURF algorithm for the ExoMars programme, presented at the Workshop on Reconfigurable Computing (WRC), At Germany, January 2013.
- [2] J. Svab, T. Krajnik, J. Faigl and L. Preucil, "FPGA based Speeded Up Robust Features,"2009 IEEE International Conference on Technologies for Practical Robot Applications, Woburn, MA, 2009, pp. 35-41.
- [3] Bay H., Tuytelaars T., Van Gool L. (2006) SURF: Speeded Up Robust Features. In: Leonaridis A., Bischof H., Pinz A. (eds) Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol 3951. Springer, Berlin, Heidelberg.
- [4] D. Fernández, "Uso de Xillybus con ZedBoard",Lic. Trabajo final, Teoría de Ingeniería Electrónica, Universidad de Sevilla, Sevilla, 2017. Universidad de Sevilla, Sevilla 2017.

- [5] Digilent. (2016,Abril,11). ZYBO FPGA Board Reference Manual. [Online]. Available: https://reference.digilentinc.com/_media/zybo:zybo_rm.pdf.
- [6] J. Martínez y J. Castillo. Memoria RAM en FPGA. [Online]. Available: <https://vhdl.es/memoria-ram-en-fpga/>.
- [7] Xilinx. Zynq Architecture zync 14.2. [Online]. Available: http://www.ioe.nchu.edu.tw/Pic/CouseItem/4468_20_Zynq_Architecture.pdf.
- [8] C. He, A. Papakonstantinou and D. Chen, "A novel SoC architecture on FPGA for ultra fast face detection,"2009 IEEE International Conference on Computer Design, Lake Tahoe, CA, 2009, pp. 412-418.
- [9] R.C. González, R.E. Woods, Digital ImageProcessing, Pearson Prentice Hall, 2008

Aplicación web para la obtención y visualización de datos provenientes de perfiladores autónomos Argo

Emmanuel Romero*, Iliana Castro*, Leonardo Tenorio-Fernandez†, Marco Castro*

*División de Estudios de Posgrado e Investigación

TecNM/IT de La Paz, La Paz, Baja California Sur, México

Email: romeroqe.93@gmail.com, iliana.cl@lapaz.tecnm.mx, marco.cl@lapaz.tecnm.mx

†Departamento de Oceanología

CICIMAR-IPN, La Paz, Baja California Sur, México

Email: ltenoriof@ipn.mx

Resumen—Argo es un conjunto global con más de 3,800 perfiladores hidrográficos autónomos que miden temperatura y salinidad en los primeros 2,000 m del océano. Los datos recolectados poseen una alta variedad de aplicaciones en campos de investigación, educación y operación. Este trabajo describe el diseño y el desarrollo de una aplicación web que obtiene de forma automatizada datos provenientes de perfiladores hidrográficos autónomos Argo, los filtra por medio de un algoritmo de punto en el polígono, utilizando sus coordenadas geográficas, realiza conversiones en los datos y genera formas de visualización amigables para los usuarios.

Palabras clave—aplicación, web, punto en el polígono, automatización, visualización, manipulación, Argo, perfiladores

I. INTRODUCCIÓN

El uso de instrumentos oceanográficos autónomos se ha convertido en una herramienta muy importante en la oceanografía observacional. Los Perfiladores Hidrográficos Autónomos (PHAs) son herramientas que reducen los costos de las observaciones oceanográficas *in situ*, obteniendo una gran cantidad de perfiles hidrográficos en el tiempo y el espacio, a menor costo respecto a los realizados en cruceros oceanográficos. Cada perfil de cada perfilador es procesado por su Centro de Ensamblaje de Datos (DACs) en un sistema de control de calidad, antes de ser publicado.

Actualmente, existen más de 3,800 PHAs pertenecientes al conjunto global Argo alrededor de los océanos del mundo, que miden temperatura y salinidad desde los 2,000 m de profundidad hasta la superficie del mar o viceversa.

Solo existen once centros de ensamblaje de datos obtenidos por perfiladores Argo en el mundo, a los cuales los países latinoamericanos carecen de participación. Esto impide que los países latinoamericanos y en desarrollo, intervengan en los procesos de calidad de PHAs, aún cuando estos instrumentos estén en las aguas de sus zonas económicas exclusivas.

Existe una alta variedad de aplicaciones científicas para los datos obtenidos por PHAs, estas aplicaciones tienen por objetivo conocer más sobre los océanos. Centros operacionales de distintos países producen análisis globales y regionales utilizando los datos del grupo Argo, con los que se alertan con anticipación anomalías de temperatura, salinidad y cambios en la circulación oceánica, brindan confiabilidad en pronósticos

y ayuda a monitorear las condiciones ambientales que afectan la productividad primaria y toda la cadena trófica.

Los medios por los cuales están disponibles los datos, carecen de herramientas para filtrar y visualizar la información antes de ser descargada, provocando que los usuarios interesados en los datos se enfrenten a archivos binarios que tienen que filtrar manualmente. Para manipular estos archivos, es necesario el uso de software orientado a matrices o conocimientos de programación.

Este trabajo muestra el diseño y desarrollo de una aplicación web que obtiene automáticamente los datos provenientes de los perfiladores, los filtra por medio de un polígono dado y brinda herramientas que facilitan la manipulación y visualización de los datos antes de su descarga.

II. PERFILADORES HIDROGRÁFICOS AUTÓNOMOS ARGO

En la actualidad existen varios modelos de perfiladores autónomos utilizados en Argo, todos trabajan de manera similar pero difieren en sus características de diseño y en las variables que son capaces de medir.

La misión estándar establecida por Argo para estos perfiladores, consiste en el descenso a una profundidad objetivo de 1,000 m, permanece en esta profundidad por 9 días y luego desciende a 2,000 m para iniciar su ascensión a la superficie durante 6 horas mientras miden temperatura y salinidad, en el caso de los perfiladores núcleo. Sin embargo, los perfiladores biogeoquímicos pueden medir un mayor número de variables como temperatura, salinidad, presión, oxígeno disuelto, clorofila y dispersión de luz.

Los satélites y GPS determinan la posición de los perfiladores cuando emergen a la superficie, y los perfiladores transmiten sus datos a los satélites, estos datos son monitoreados por el centro de información de Argo y luego son recibidos por su DAC. Además, los satélites verifican si existe una nueva misión configurada para el perfilador, si es así, la envían al perfilador y este la ejecuta, si no, el perfilador continúa con su misión actual. Cada vez que un PHA realiza un perfil hidrográfico, emerge a la superficie y transmite sus datos.

III. DATOS PROVENIENTES DE PHAS

Los modelos actuales de perfiladores al realizar las mediciones de las variables durante cada perfil, sólo son capaces

de realizar la medición de una de sus variables a la vez, por ejemplo, no pueden medir la temperatura en el mismo instante de tiempo en el que midieron la salinidad.

Debido a esta limitación es necesario su procesamiento, para obtener aproximaciones de las mediciones en instantes de tiempo en los cuales no fueron medidas. Los datos de cada PHA tienen que ser validados, verificados y procesados por un sistema de control de calidad, antes de ser utilizados o publicados. Este sistema de control de calidad consta de dos etapas, el Control de Calidad en Tiempo Real (RTQC) y el Control de Calidad en Modo Diferido (DMQC).

III-A. Controles de calidad

En los DAC, estos datos están sujetos a un proceso de control de calidad llamado RTQC donde se marcan y/o corrigen datos erróneos. El objetivo del RTQC es que los datos estén disponibles dentro de las primeras 24 horas de su transmisión y por lo tanto, estas pruebas son limitadas y automáticas. Estos datos están libres de errores graves en cada una de las variables que mide el perfilador (posición geográfica, temperatura, presión, etc), y deben ser consistentes con la hidrografía de la zona donde se realizó el perfil.

Una vez concluido este proceso, los datos se envían a los dos Centros Globales de Ensamblaje de Datos (GDAC), para que estén disponibles para su descarga.

Dentro de los archivos, los datos sin ajustes se identifican como `DATA_MODE='R'`, mientras que los datos con ajuste se identifican con `DATA_MODE='A'` [1].

El segundo proceso de control de calidad es el DMQC. Los datos provenientes por este control de calidad reemplazan a los datos obtenidos por el RTQC, ya que, durante este proceso estos son sometidos a un escrutinio detallado por parte de expertos oceanográficos, estos datos llegan a tardar un año en ser publicados y se denotan como `DATA_MODE='D'` [1].

Esto se debe a que algunos datos incorrectos no pueden determinarse como incorrectos y algunos datos correctos pueden identificarse erróneamente como incorrectos mediante las pruebas en tiempo real.

III-B. Acceso a los datos

US GODAE Data Server y IFREMER Data Server son los dos GDACs, estos dos servidores son alimentados automáticamente por todos los DACs con la última versión de los datos de forma simultánea, para así garantizar la coherencia entre los dos conjuntos de datos.

Ambos tienen a disposición del usuario medios por los cuales brindan acceso a los datos validados por los controles de calidad, la elección de que GDAC utilizar es cuestión de cual se adapta mejor a las necesidades del usuario.

Además, ofrecen los datos a través de un Identificador de Objeto Digital (DOI), por el Protocolo de Transferencia de Hipertexto (HTTP) y por el Protocolo de Transferencia de Archivos (FTP). Cada mes se genera un DOI asociado a una copia exacta del GDAC, útil para usuarios interesados en usar archivos NetCDF en publicaciones científicas. Para usuarios interesados en el manejo de archivos NetCDF reales,

los GDACs ofrecen los datos a través de HTTP y FTP. Otra alternativa es el servidor *rsync*, el cual proporciona un servicio de sincronización espejo entre el GDAC y el usuario.

Estos archivos modelan un conjunto de datos científicos como una colección de matrices junto con sus sistemas de coordenadas y algunas de sus propiedades auxiliares. Cada variable tiene un tipo, una forma especificada por una lista de dimensiones y un conjunto de otras propiedades [2]. Las interfaces de programación principales para el manejo de estos archivos son C, C++, Java, Python, Matlab, R, entre otros.

IV. APLICACIÓN WEB

IV-A. Selección de los datos

Como la selección de los datos puede ser de interés tanto de escala global como regional, es necesario filtrar los datos por medio de las coordenadas geográficas en las que fueron tomados.

Este es un problema de Punto en el Polígono (PIP), procedente de la geometría computacional y busca determinar si un punto está dentro o fuera de un polígono.

El algoritmo par-impar [3], traza un rayo desde un punto hacia una dirección fija e intersecta los bordes del polígono. Si el punto está en el exterior del polígono, el rayo se cruzará con el borde un número par de veces, de lo contrario, si está en el interior intersectará con el borde un número impar de veces, como se muestra en la Fig. 1.

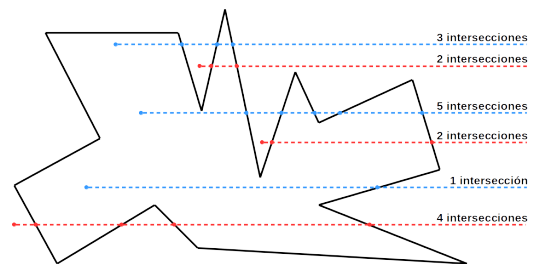


Figura 1: Intersecciones de puntos dentro de un polígono con el algoritmo par-impar.

A pesar de que este algoritmo puede fallar cuando el punto se encuentra en el perímetro de la figura, es suficiente para delimitar áreas de estudio en el océano.

Los datos provenientes de los PHAs y que al menos han pasado por RTQC se descargan automáticamente cada 24 horas por medio del servicio de sincronización del GDAC IFREMER [4]. Para filtrarlos se estableció el polígono que será el área de estudio, se extrajeron los máximos y mínimos de la latitud y longitud de sus puntos, para fijarlos como rango, los perfiles fuera de este rango se descartaron automáticamente.

Debido a que es necesaria la evaluación de cada uno de los puntos no descartados con el polígono mediante el algoritmo PIP elegido, se utilizó programación paralela para que cada núcleo del procesador evaluase equitativamente cierta cantidad de puntos y acelerar este proceso. Para efectos de pruebas de este trabajo se estableció a la Zona Económica Exclusiva (ZEE) de México como área de estudio.

En esta zona México tiene los mismos derechos que en su mar territorial, pero permite la libre circulación de embarcaciones extranjeras y la instalación de diversas vías, por ejemplo oleoductos marinos o diversos tipos de cableado. En la Fig. 2 se muestran con puntos rojos las ubicaciones geográficas de los perfiles que se realizaron dentro del polígono, filtrados por el algoritmo par-impar.

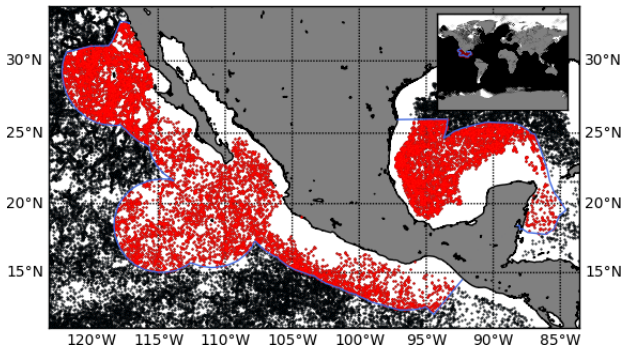


Figura 2: Ubicaciones geográficas filtradas de todos los perfiles dentro del polígono definido.

IV-B. Visualización de los datos

Para poder manipular y visualizar estos datos se exportaron de sus archivos fuente a archivos con Valores Separados por Coma (CSV), se diseñó una Base de Datos (BD) relacional con tablas y atributos semejantes a los contenidos en los archivos NetCDF y se agregaron estos datos filtrados a la BD a través de los archivos CSV generados anteriormente, con el fin, de evitar leer los datos desde los archivos NetCDF cada vez que un usuario los solicite desde la aplicación web.

Con los datos disponibles en la base de datos, por medio de consultas se pueden obtener datos estadísticos, como por ejemplo la cantidad de perfiladores dentro del polígono, la cantidad de perfiles o perfiladores por año, los DACs a los que pertenecen dichos perfiladores, entre otros. En la Tabla I se muestran los perfiladores que han realizado mediciones dentro del polígono dado en el mes de noviembre del 2019.

Podemos observar en la tabla que existe una escasez de perfiladores biogeoquímicos dentro del polígono. Estos 4 PHAs biogeoquímicos son capaces de medir oxígeno además de temperatura y salinidad, pero ninguno de sus datos de oxígeno concluyeron satisfactoriamente el proceso de control de calidad, por lo que no están disponibles. Así que podemos concluir que dentro de la ZEE de México no existen datos de parámetros biogeoquímicos de buena calidad provenientes por PHAs Argo.

Otra forma de visualización fue la utilización de widgets para resumir la información y la generación de gráficas. Pueden generarse gráficas estadísticas de los datos (como se muestra en la Fig. 3), con las que podemos comparar por ejemplo, los tipos de perfiladores presentes dentro del polígono, ver el crecimiento de la cantidad de perfiles o perfiladores por año.

Los widgets nos brindan información útil e inmediata, por ejemplo cuantos DACs están presentes dentro del polígono

Tabla I: Perfiladores y perfiles presentes en la ZEE de México.

DAC	Núcleo		Biogeoquímicos		Perfiles
	Activos	Inactivos	Activos	Inactivos	
AO : AOML	51	114	0	3	32 998
IF : CORIOLIS	6	3	0	1	1 098
ME : MEDS	1	1	0	0	201
Total	58	118	0	4	34 297

dado, la cantidad total de PHAs y cuantos de ellos aún siguen realizando mediciones, así como la cantidad total de perfiles.

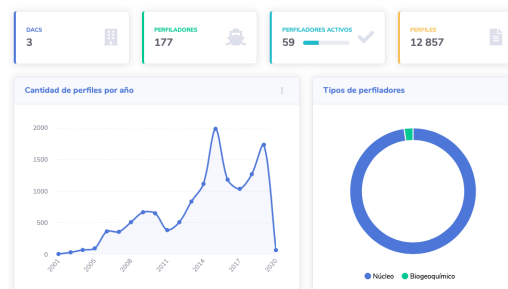


Figura 3: Widgets y gráficas estadísticas de los datos.

También se graficaron los perfiles de temperatura (Fig. 4) y salinidad (Fig. 5), el diagrama de Temperatura-Salinidad (TS) (Fig. 6), la estimación de la trayectoria del perfilador (Fig. 7) y los perfiles de temperatura (Fig. 8) y salinidad (Fig. 9) respecto al tiempo, estos diagramas son básicos para realizar análisis en investigaciones científicas del océano.

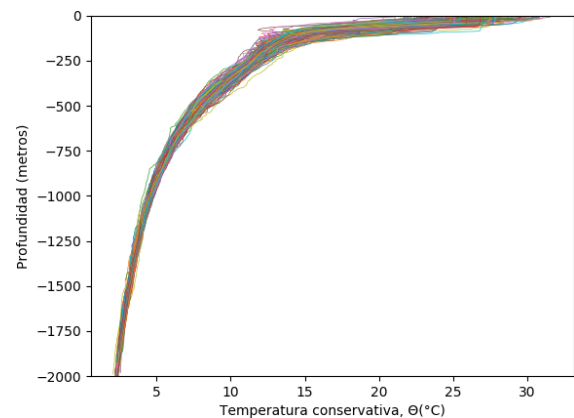


Figura 4: Perfil de temperatura.

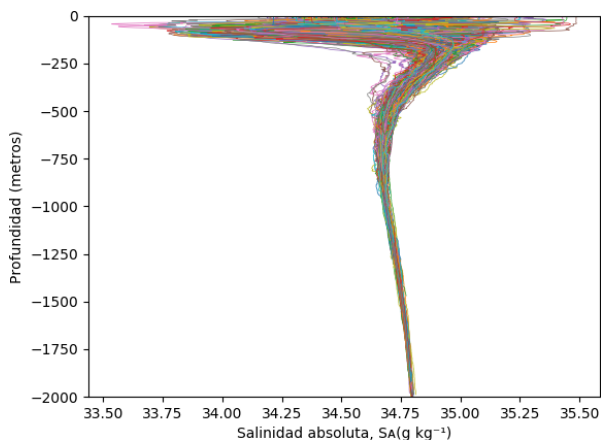


Figura 5: Perfil de salinidad.

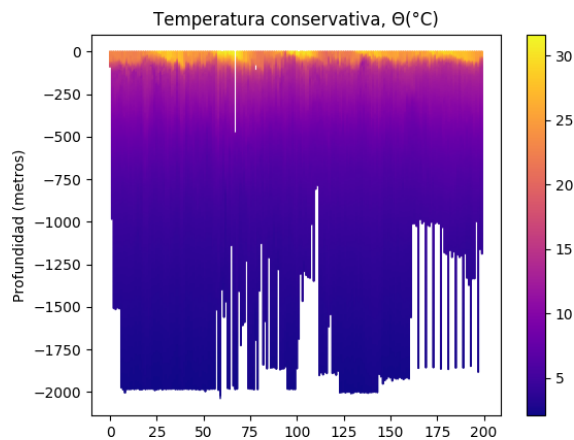


Figura 8: Perfiles de temperatura respecto al tiempo.

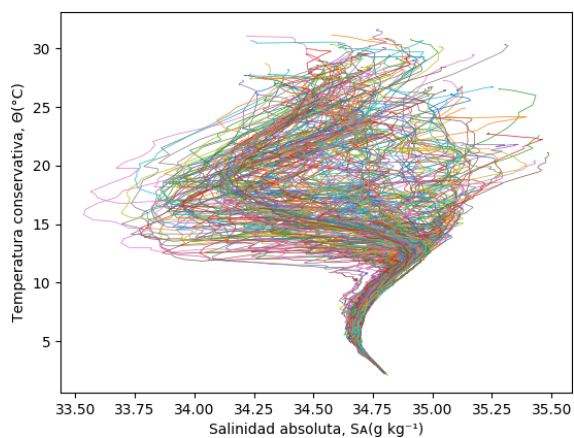


Figura 6: Diagrama TS.

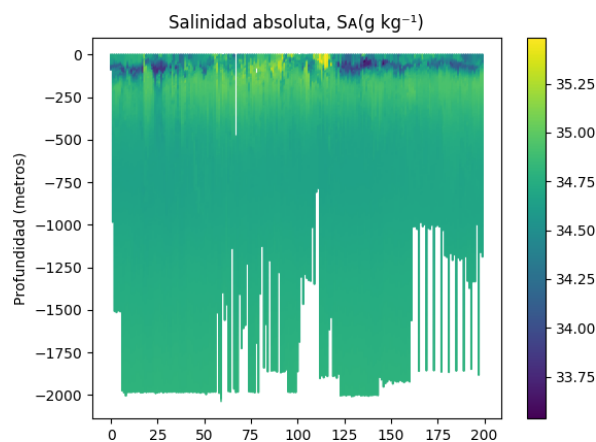


Figura 9: Perfiles de salinidad respecto al tiempo.

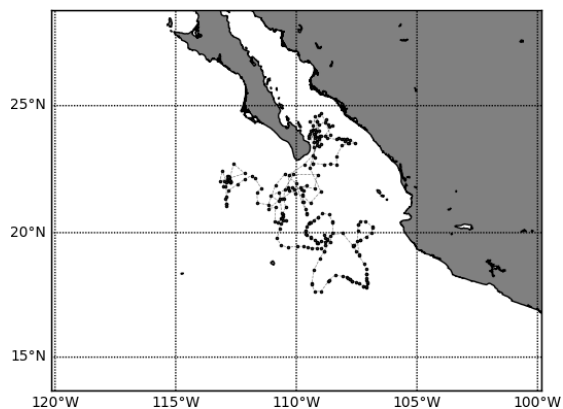


Figura 7: Trayectoria del PHA.

Estas imágenes se guardan en tres tamaños y en los idiomas español e inglés, con el fin de ralentizar lo menos posible el tiempo de carga de la aplicación web y al mismo tiempo tener a disposición del usuario imágenes de buena calidad y en el idioma de su preferencia.

Además usan TEOS-10, durante la 25ª Asamblea en junio de 2009 de la Comisión Oceanográfica Intergubernamental se adoptó a la Ecuación Termodinámica Internacional del Agua de Mar-2010 (TEOS-10), esta define las propiedades termodinámicas del agua de mar y del hielo. Reemplazó a la Ecuación del Estado del Agua de Mar-1980 (EOS-80) como la descripción oficial de las propiedades de agua de mar y hielo en las ciencias marinas [5].

Uno de los cambios más significativos en comparación con EOS-80, es que TEOS-10 usa salinidad absoluta (fracción de masa de sal en agua de mar) en lugar de la salinidad práctica (que es esencialmente una medida de la conductividad del agua de mar) para describir el contenido de sal del agua de mar. Las BDs internacionales (incluidas las de Argo) actualmente registra sus datos como salinidad práctica, para evitar confusiones mientras se elimina su uso en publicaciones científicas.

Por tal motivo, los datos se guardan de la misma forma en la aplicación web, pero se realiza su conversión a temperatura conservativa y salinidad absoluta, cuando se utilizan para los diagramas y gráficas, esta conversión se realiza mediante el uso la caja de herramientas oceanográficas Gibbs-SeaWater (GSW), la cual contiene las subrutinas TEOS-10 para evaluar las propiedades termodinámicas del agua de mar [6].

Durante las pruebas de la generación de los anteriores diagramas se encontraron diagramas TS con datos de salinidad que no reflejaban los patrones comunes de salinidad de los mismos a bajas profundidades, se graficaron los diagrama TS de cada mes en una pequeña área en específico dentro del polígono, con la profundidad como paleta de colores, se agregaron las masas de agua de la zona [7] y los isógramas de densidad según TEOS-10 (Fig. 10).

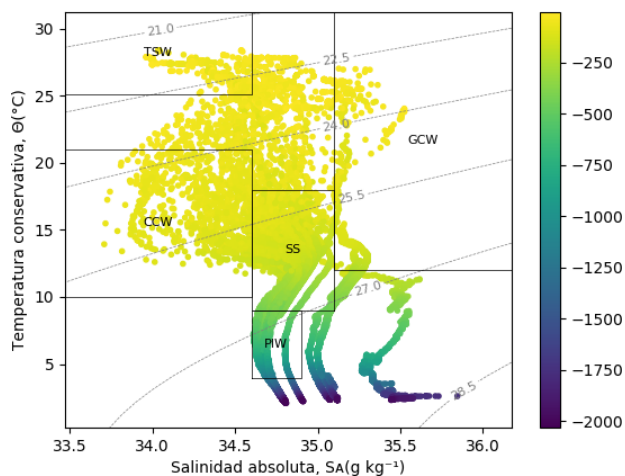


Figura 10: Diagrama TS que muestra corrimientos de salinidad a bajas profundidades en el mes de diciembre.

La Fig. 10 demostró corrimientos en los datos de salinidad. Por lo que se realizó un análisis de los datos, para identificar los datos que presentaban dichas anomalías. Algunos de los datos en RTQC eran los causantes de estas variaciones (Fig. 11).

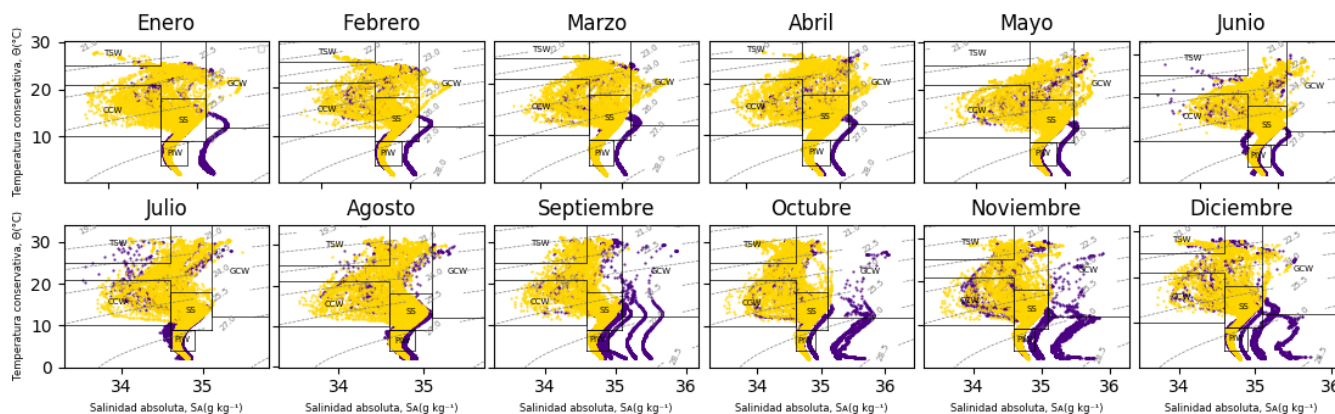


Figura 11: Comparación de datos en RTQC (morado) y DMQC (amarillo).

Por lo que se decidió trabajar solo con datos a los que se les haya aplicado el DMQC, disminuyendo con ello la cantidad de datos disponibles, pero conservando la confiabilidad de los mismos.

Además de las gráficas, otra forma de mejorar la experiencia de visualización de los datos, fue la implementación de la herramienta *ncdump* [2]. Esta herramienta permite convertir los archivos NetCDF a archivos de texto, lo que permite a los usuarios de la aplicación web, la visualización de la estructura y de los datos contenidos dentro de los archivos NetCDF antes de su descarga.

El flujo de los procedimientos descritos anteriormente se muestran en la Fig. 12 y se realizan después de la sincronización espejo de los archivos (descrito en la subsección IV-A), con el fin de que la base de datos, los archivos de texto y las gráficas se actualicen diariamente.

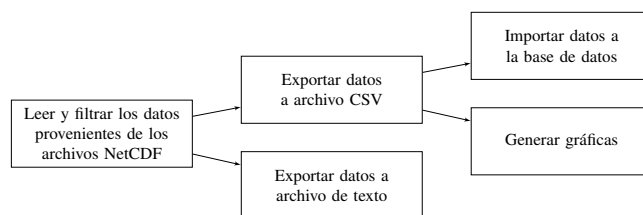


Figura 12: Flujo del procesamiento de los datos.

IV-C. Mapa satelital interactivo

La importancia del desarrollo de una aplicación web para el desarrollo de este trabajo, radica en que los datos de los perfiladores son públicos, deben de estar disponibles y ser de fácil acceso, para cualquier científico o investigador interesado en el uso de los mismos.

Para su desarrollo, se utilizó Laravel 6 LTS, el cual es un poderoso Framework PHP de código abierto con un gran potencial de integración, escalabilidad y facilidad de mantenimiento. Los Frameworks como Laravel empaquetan una colección de componentes de terceros junto con archivos de configuración, proveedores de servicios, estructuras de directorios prescritas y programas de arranque de aplicaciones [8], este código es

generado, verificado y actualizado por una comunidad inmensa de expertos programadores, por lo que las brechas de seguridad son continuamente analizadas y resueltas. Además al escoger una de sus versiones soporte a largo plazo (LTS) se garantizan correcciones de errores por dos años (hasta 3 de septiembre de 2021) y tres años de correcciones de seguridad (hasta 3 de septiembre de 2022).

Con los datos disponibles en la base de datos, las gráficas y los archivos de texto generadas, se desarrolló la herramienta con la cual serán accesibles por los usuarios.

Con la ayuda de la interfaz de programación de aplicaciones (API) gratuita de ArcGIS para JavaScript y la biblioteca de código abierto de JavaScript Leaflet para mapas interactivos (apta para dispositivos móviles).

Se programó un mapa satelital interactivo, que contiene los límites del polígono dado desplegados con una línea color cyan y la última posición en la que cada perfilador envió datos, estas posiciones se representaron con puntos y se codificaron por colores, verde para los perfiladores activos y gris para los inactivos, así como círculos para los perfiladores núcleo y rombos para los biogeoquímicos (Fig. 13).



Figura 13: Herramienta de acceso a los datos.

Cada uno de estos puntos es seleccionable, al seleccionarlo se muestra una ventana emergente (Fig. 14) que permite dibujar la trayectoria del perfilador, también contiene un resumen de los datos del perfilador, sus gráficas y un enlace a su página (Fig. 15), en esta página se encuentran todos sus datos, gráficas y archivos de texto generados, enlaces para la descarga de los datos en NetCDF y CSV, así como un mapa interactivo propio donde se muestra la información de cada uno de sus perfiles.

Para manejar y filtrar los datos se agregó al mapa principal una barra de navegación con tres pestañas: localización, datos y selección.

En la pestaña de localización se encuentran los filtros de los perfiladores, estos controlan los puntos que se muestran en el mapa satelital. Pueden filtrarse por última localización, localización de lanzamiento, por mes, rango de fechas y por un polígono dado. Además permite filtrar por perfiladores activos, inactivos, núcleo, biogeoquímicos o solo mostrar aquellos que se encuentran dentro del polígono.

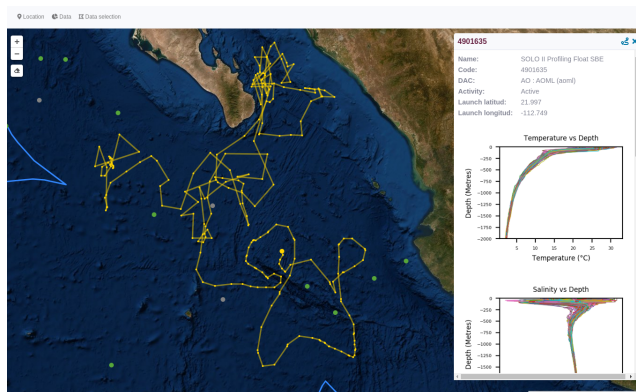


Figura 14: Datos, gráficas y trayectoria de un PHA.

4901635 - SOLO II Profiling Float SBE

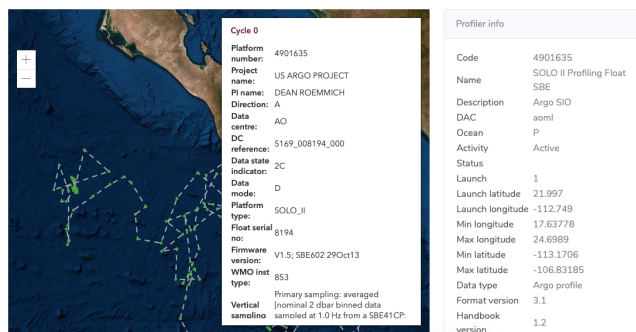


Figura 15: Página de un PHA.

Dentro de la pestaña de selección podemos dibujar un polígono mas pequeño para filtrar los datos y dibujar sus trayectorias (Fig. 16). Este polígono puede ser dibujado mediante latitudes y longitudes máximas y mínimas, por CSV y por medio de dibujo libre con clics dentro del mapa. Los filtros mencionados anteriormente facilitan la localización de los datos para su descarga y visualización.

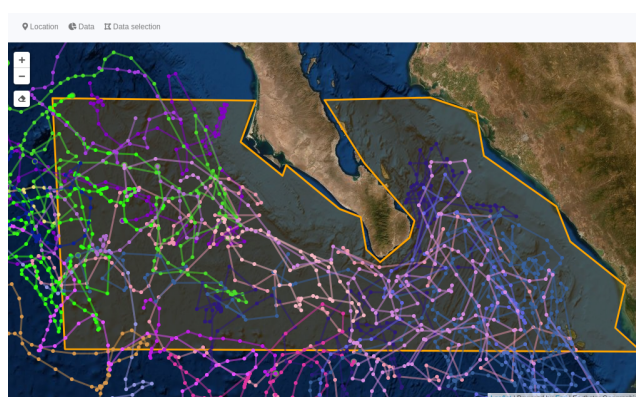


Figura 16: Trayectorias de PHAs filtrados por un polígono dibujado.

Por último en la pestaña de datos se muestran datos estadísticos y diagramas según el filtro seleccionado (Fig. 17).

En una tabla muestran los datos de los perfiladores filtrados, la cantidad de perfiles y perfiladores activos e inactivos, núcleo y biogeoquímicos de cada DAC presente dentro del filtro seleccionado. Además de gráficas de perfiles por año, perfiladores por año y diagramas de tipos de perfiladores.

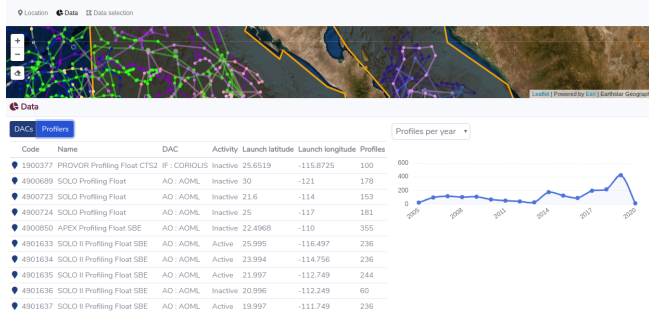


Figura 17: Datos de perfiladores dentro del polígono y gráfica de perfiles por año.

IV-D. Pruebas del algoritmo PIP par-impar y comparación de datos Argo con World Ocean Atlas 2018

Una de las grandes ventajas de esta aplicación web, es que los procedimientos descritos en este trabajo pueden utilizarse con otras bases de datos georeferenciadas.

Se realizaron pruebas con la base de datos World Ocean Atlas 2018 (WOA18), la cual proporciona datos de calidad controlada para calcular las climatologías de temperatura, salinidad, oxígeno disuelto y nutrientes inorgánicos disueltos. Este conjunto de campos climatológicos se derivan de la NCEI World Ocean Database 2018 (WOD18).

Se descargaron los datos mensuales de temperatura [9] y salinidad [10] de la media estadística de cada cuarto de grado ($1/4^\circ$) del 2005 al 2017 y se le aplicó el algoritmo PIP descrito en este trabajo en un polígono frente a Baja California Sur, México (Fig. 18).

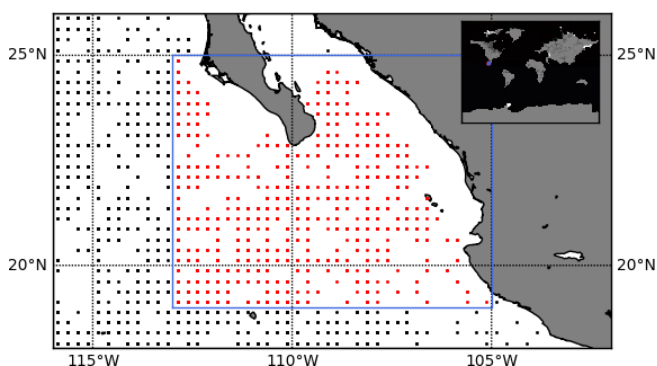


Figura 18: Ubicaciones geográficas filtradas de los datos de la WOA18 dentro de un polígono definido.

Se aplicaron las correcciones de los datos según TEOS-10 y se compararon con los datos provenientes por el DMQC de los PHAs Argo (Fig. 19).

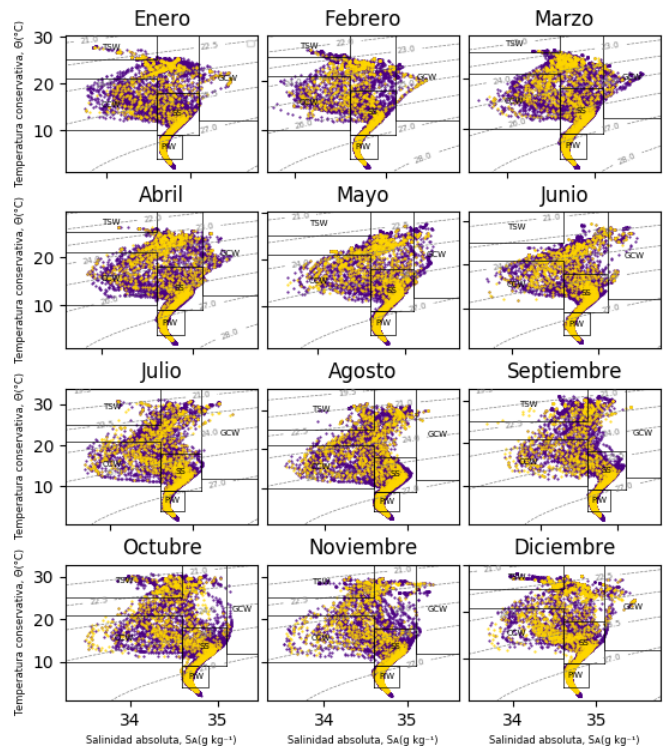


Figura 19: Comparación de datos DMQC Argo (morado) y datos de la WOA18 (amarillo).

Como podemos observar en la figura anterior, los datos provenientes de los PHAs y de la WOA18 de la misma zona, se empalman a profundidades bajas y se encuentran en las mismas masas de agua.

V. CONCLUSIONES Y TRABAJO FUTURO

Se ha presentado el diseño y desarrollo de una aplicación web que obtiene automáticamente los datos obtenidos por PHAs dentro de un polígono dado, que aplica correcciones en las variables de temperatura y salinidad, según la descripción oficial de las propiedades de agua de mar vigente y que implementa formas de visualización y filtrado de los datos, para su fácil localización y descarga.

Esta aplicación permite localizar los PHAs que han realizado mediciones dentro de una zona de estudio, de forma intuitiva y sencilla, las diversas formas de visualización de sus datos facilitan su interpretación y estudio, además las herramientas del mapa interactivo como dibujar las trayectorias, o el filtrado de los datos por medio de un polígono dado, permiten localizar fenómenos oceánicos en el área de interés del usuario.

Tanto el algoritmo PIP como el resto de procedimientos descritos en este trabajo, pueden aplicarse a datos provenientes de otras bases de datos internacionales georeferenciadas que manejen datos de temperatura y salinidad, y con ello enriquecer la base de datos de la aplicación web.

Los datos en DMQC tienen congruencia con otras bases de datos internacionales, lo cual valida este proceso, sin embargo,

demasiados datos tienen que ser descartados debido a los corrimientos presentes en RTQC. Por lo que se propone:

Mejorar la eficiencia del DMQC y corregir el mayor número de datos posibles del RTQC, aplicar técnicas de aprendizaje automático en los controles de calidad, podría facilitar la revisión por parte de los expertos oceanográficos, aumentar el número de observaciones en DMQC y reducir el tiempo en el que los datos provenientes por el DMQC estén disponibles.

Trabajar en los datos para calcular los valores a profundidades estándar, ya que, cada instrumento mide a diferentes profundidades y con tazas de medición diferentes.

Aplicar análisis regresivos en las series temporales de los datos para obtener tendencias en sus variables y aplicar análisis de grupos para encontrar grupos de perfiladores o perfiles con características en común.

AGRADECIMIENTOS

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por su apoyo financiero, al Instituto Tecnológico de La Paz (ITLP), al Centro Interdisciplinario de Ciencias Marinas (CICIMAR) y a la Dr. Laura Sánchez Velasco del Departamento Plancton y Ecología Marina del CICIMAR-IPN, por su apoyo institucional.

REFERENCIAS

- [1] Argo Data Management Team (2019). "Argo user's manual V3.3", doi: 10.13155/29825
- [2] Rew, R. K. and G. P. Davis, "NetCDF: An Interface for Scientific Data Access", IEEE Computer Graphics and Applications, Vol. 10, No. 4, pp. 76-82, July 1990, doi: 10.1109/38.56302
- [3] M. Galetzka and P. Glauner, "A Simple and Correct Even-Odd Algorithm for the Point-in-Polygon Problem for Complex Polygons", Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2017, doi: 10.5220/0006040801750178
- [4] Argo (2000). "Argo float data and metadata from Global Data Assembly Centre (Argo GDAC)". SEANO, doi: 10.17882/42182
- [5] The international thermodynamic equation of seawater - 2010: Calculations and use of thermodynamic properties. Intergovernmental Oceanographic Commission, Manuals and Guides No. 56, UNESCO (English), 196 pp. (2010).
- [6] McDougall, T.J. and P.M. Barker, 2011: "Getting started with TEOS-10 and the Gibbs Seawater (GSW) Oceanographic Toolbox", 28pp., SCOR/IAPSO WG127, ISBN 978-0-646-55621-5.
- [7] E. Portela et al. (2016). "Water masses and circulation in the tropical Pacific off central Mexico and surrounding areas". Journal of Physical Oceanography. 46, doi: 10.1175/JPO-D-16-0068.1
- [8] Matt S. "Laravel: Up and Running A Framework for Building Modern PHP Apps". O'Reilly, 1st edition, 2016, ISBN 978-1491936085
- [9] Locarnini, R. A., A. V. Mishonov, O. K. Baranova, T. P. Boyer, M. M. Zweng, H. E. Garcia, J. R. Reagan, D. Seidov, K. Weathers, C. R. Paver, and I. Smolyar, 2018. World Ocean Atlas 2018, Volume 1: Temperature. A. Mishonov Technical Ed.; NOAA Atlas NESDIS 81, 52 pp.
- [10] Zweng, M. M., J. R. Reagan, D. Seidov, T. P. Boyer, R. A. Locarnini, H. E. Garcia, A. V. Mishonov, O. K. Baranova, K. Weathers, C. R. Paver, and I. Smolyar, 2018. World Ocean Atlas 2018, Volume 2: Salinity. A. Mishonov Technical Ed.; NOAA Atlas NESDIS 82, 50 pp.

Diseño de una aplicación web para control de expedientes médicos del Hospital General Progreso CAAPS

Ing. Antonio Alejandro Sánchez Zapata
Estudiante de la Maestría en Sistemas Computacionales en un Programa PNPC.
Tecnológico Nacional de México /IT de Acapulco
Acapulco, Guerrero, México.
aaszapata18@gmail.com

M.I.D.S. Alma Delia de Jesús Islao
Docente de la Maestría en Sistemas Computacionales.
Tecnológico Nacional de México /IT de Acapulco
Acapulco, Guerrero, México.
alma.islao.ita@gmail.com

Dr. Eduardo de la Cruz Gámez
Docente de la Maestría en Sistemas Computacionales.
Tecnológico Nacional de México /IT de Acapulco
Acapulco, Guerrero, México.
gamezeduardo@yahoo.com

M.T.I. Jorge Carranza Gómez
Docente de la Maestría en Sistemas Computacionales.
Tecnológico Nacional de México /IT de Acapulco
Acapulco, Guerrero, México.
jcarranzamx@gmail.com

Ing. Luis Ángel Castro Silvestre
Estudiante de la Maestría en Sistemas Computacionales en un Programa PNPC
Tecnológico Nacional de México /IT de Acapulco
Acapulco, Guerrero, México
lacsilvestre18@gmail.com

Resumen— En este artículo, se plasma el trabajo interdisciplinario de la Maestría en Sistemas Computacionales con apoyo del CONACYT, impartida en el Instituto Tecnológico de Acapulco. El artículo tiene por meta presentar el desarrollo analítico y visual de una aplicación web que permita el registro de expedientes médicos y reportes médicos del Área de Servicio Amigable para Adolescentes, ubicada dentro del Hospital General Progreso CAAPS, del municipio de Acapulco de Juárez, Guerrero. En la primera parte del artículo se redacta la introducción, el marco teórico, las tecnologías elegidas para el desarrollo del proyecto y el objetivo a cumplir.

En la segunda parte se presentan los diagramas que ayudan a comprender el funcionamiento de la aplicación, se describe el método utilizado y se citan trabajos relacionados. El apartado de conclusiones contiene los resultados del diseño inicial de la aplicación web y la descripción de los módulos. Se hace mención que este artículo forma parte de una serie de trabajos siendo el segundo de estos, el cual, da seguimiento a un artículo previo titulado: Implementación de una Aplicación Web para el Registro y Control de Expedientes Médicos de la Clínica Avanzada de Atención Primaria a la Salud (C.A.A.P.S) [1].

Palabras clave— Python, MVC, SEAPCAAPS, UML.

I. INTRODUCCIÓN

Las Tecnologías de Información reducen el consumo de tiempo y por ende inversiones; esto como consecuencia hace que los trabajadores de las empresas mejoren su productividad, al aprovechar mejor el tiempo y disminuir este en buscar resolver problemas que se presentan día a día, se dice entonces que el utilizar herramientas digitales para realizar tareas de forma más sencilla y mejorar el rendimiento de los empleados, así como

responder con mayor velocidad a tiempos de respuesta, ya que clientes y usuarios cada día buscan una mejor atención y mayor competencia al adquirir y brindar nuevos servicios.

Actualmente, esto ha influenciado en instituciones tanto públicas como privadas, desde escuelas, tiendas departamentales, hoteles e incluso hospitales; sin embargo, no todas estas instituciones han migrado a este tipo de trabajo ya que, no cuentan con el equipo y/o tecnología necesaria, no cuentan con los recursos económicos para obtener la tecnología, o están demasiado acostumbrados a una forma de realizar su trabajo por lo que no migran a nuevas formas de trabajo.

En este caso, el Hospital General Progreso CAAPS mantiene en la actualidad una forma de trabajo manual, la cual sufre pérdida de información en expedientes físicos ya que son registrados a lápiz y papel. Con la creación de esta herramienta se buscan como beneficios principales: El almacenamiento de la información de los pacientes de forma segura y confiable, la reducción del capital invertido en objetos papelería e impresiones de formatos de historia médica, y la implementación de este sistema digital en pro de mejorar el trabajo y manejo de información en el módulo de Servicio Amigable para Adolescentes.

A continuación se describe la estructura por la que está conformada este proyecto.

II. OBJETIVO GENERAL DEL PROYECTO

Implementar una aplicación web para el registro y control de los expedientes de consultas para el Área de Servicio Amigable del Hospital Progreso CAAPS.

III. MARCO CONCEPTUAL

A continuación se describen las herramientas principales utilizadas para el análisis y el desarrollo de la aplicación:

A. Enterprise Architect

Es una herramienta de diseño y modelado visual basada en UML. La plataforma soporta: el diseño y construcción de sistemas de software; modelado de procesos de negocio; y modelado de dominios basados en la industria. Es utilizado por empresas y organizaciones no solo para modelar la arquitectura de sus sistemas, sino también para procesar la implementación de estos modelos en todo el ciclo de vida del desarrollo de la aplicación [2].

Con el apoyo de esta herramienta se han creado los diagramas para dejar más clara la comprensión del funcionamiento de la aplicación, además de identificar los roles que tienen las personas involucradas en el manejo de esta.

B. MySQL

Es un sistema de gestión de bases de datos relacionales de tipo Open Source, rápido, fiable y sencillo de utilizar que puede trabajar en entornos cliente/servidor o incrustados [3].

Esta herramienta resulta ser útil para la generación de la base de datos en la que se guardará la información de los expedientes de historia clínica del Área de Servicio Amigable, además de que puede generarse en muchísimas plataformas diferentes, y nos pueden facilitar al mismo tiempo la creación de los diagramas de entidad relación para mostrar más detalladamente la información de la base de datos.

C. Python

Python es un lenguaje de programación el cual basa su estructura en una sintaxis que favorece la creación de un código

legible y reutilizable [4]. Se trata de un lenguaje de programación multiparadigma, este soporta programación orientada a objetos, programación imperativa y programación funcional, esta última a menor medida. Es un lenguaje interpretado, dinámico y multiplataforma. Una de las ventajas principales de este es la posibilidad de crear un código con gran legibilidad, que ahorra tiempo y recursos, lo que facilita su comprensión e implementación.

Entre los usos que se le da a este lenguaje es en el análisis de datos e implementación de inteligencia artificial, cuenta con un gran número de librerías para procesamiento de datos, y herramientas como Django para el desarrollo web.

La elección de estas herramientas se debe al hecho de que se busca la creación de una herramienta de bajo costo y que logre ser desarrollada de manera rápida, por lo que se utilizan el lenguaje de programación de Python en conjunto con el Framework de Django [5] y la biblioteca de estilos de Bootstrap.

IV. DISEÑO DE LA HERRAMIENTA PROPUESTA

Para la elaboración de los diagramas UML [6] y la construcción de la aplicación, se hace uso de la herramienta Enterprise Architect.

A continuación se describen las actividades que se realizan en el Área de Servicio por medio de un diagrama de modelado de negocio, y las funciones principales de los usuarios involucrados se observan en los siguientes diagramas de casos de uso, contexto de la aplicación, despliegue, clases y Entidad-Asociación.

A. Modelado de Negocio

En este apartado se describen los procesos que se ejecutan en el Área de Servicio (Véase Fig. 1).

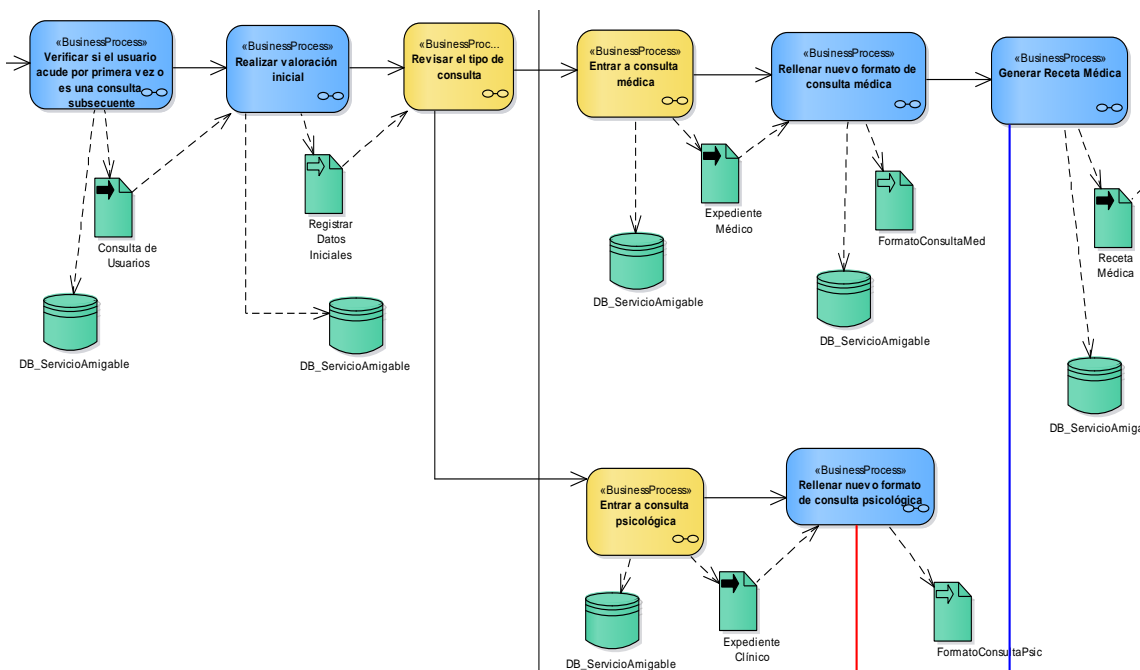


Fig. 1. Modelado de negocio de la herramienta propuesta

En el diagrama ilustrado en la Fig. 1, se observan los procesos que corresponden a las tareas principales que se ejecutan en el Servicio, y se describen a continuación.

- Verificar si el usuario acude por primera vez o es una consulta subsecuente: Se revisa en Recepción si el usuario del Servicio está acudiendo por primera vez a consulta o si ya tiene generado un expediente clínico o médico.
- Realizar valoración inicial: En este proceso se realiza la captura de datos del paciente al entrar al módulo. Se capturan los datos correspondientes a Nombre completo, Fecha de nacimiento, CURP, Sexo, Peso y Estatura.
- Revisar el tipo de consulta: Se verifica si el paciente entrará a consulta médica o psicológica y se le indica a que consultorio será al que entrará.

En caso de que entre a Consulta médica, se realizan los siguientes procesos:

- Entrar a consulta médica: Después de corroborar que el paciente recibirá una consulta médica, pasará al consultorio con el médico correspondiente.
- Registrar nuevo formato de historia médica: Una vez efectuada la consulta y generado el pre-diagnóstico, se rellena el formato de historia médica que después se colocará en el expediente del paciente del Servicio.
- Generar receta médica: Se crea la receta médica del paciente de forma que pase al Área de Farmacia a canjear sus medicamentos escritos en el documento.

En caso de que entre a Consulta psicológica, se realizan los siguientes procesos:

- Entrar a consulta psicológica: Después de corroborar que el paciente recibirá una consulta psicológica, pasará al consultorio con el psicólogo correspondiente.
- Rellenar nuevo formato de historia clínica: Una vez terminada la consulta se rellena el formato de historia clínica que después se colocará en el expediente clínico del paciente del Servicio.

B. Diagrama de Contexto

En este apartado se describen los agentes que intervienen en el Área de Servicio y que harán uso de la aplicación (Véase Fig. 2).

- Pacientes: Este agente externo corresponde a los pacientes que acuden al Área de Servicio Amigable.
- Recepción: Este agente externo corresponde al responsable de la Recepción, atiende a los pacientes del Área de Servicio en cuanto entran a solicitar una consulta, registra sus datos y asigna la consulta.
- Consultorio: Una vez que se asigna la consulta en Recepción, el paciente entra al Consultorio médico o psicológico según sea el caso a recibir atención del responsable. Al finalizar, se emiten las historias, reportes y evaluaciones pertinentes y la Receta médica.

- Farmacia: Al finalizar la consulta del paciente, éste se retira del Servicio y sale del Hospital a la farmacia correspondiente por el medicamento prescrito en la Receta.

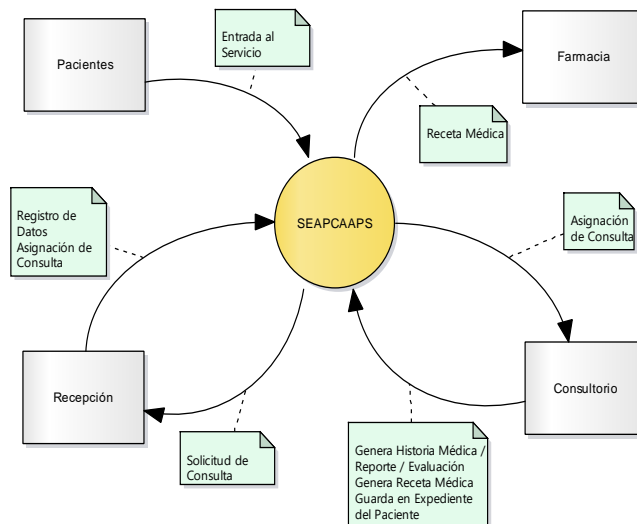


Fig. 2. Diagrama de contexto de la aplicación.

C. Casos de Uso

Como se puede apreciar en los siguientes diagramas de casos de uso (Véase Fig. 3, 4 y 5), las tareas a ejecutar en el sistema están repartidas entre los usuarios involucrados que operan en el Servicio, esto con el objetivo de conseguir un manejo administrable de la aplicación, además de distribuir las tareas en cada uno de estos de acuerdo al puesto que desempeñan dentro del área.

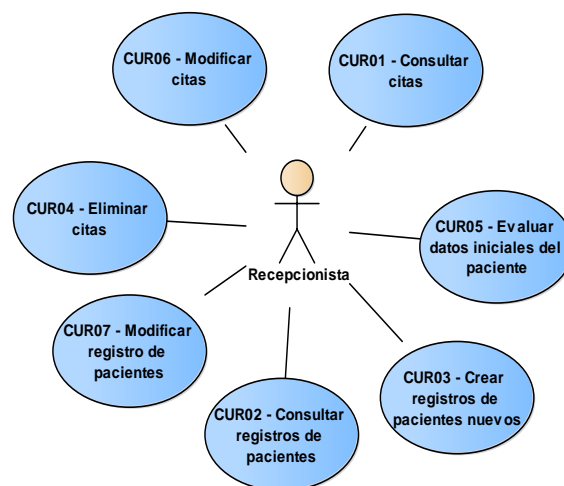


Fig. 3. Casos de uso primarios de Recepcionistas.

En la Fig. 3, se observa el diagrama del actor que corresponde al responsable de la recepción, que como función principal desempeña el registro y la evaluación inicial de los pacientes que acuden al Área de Servicio.

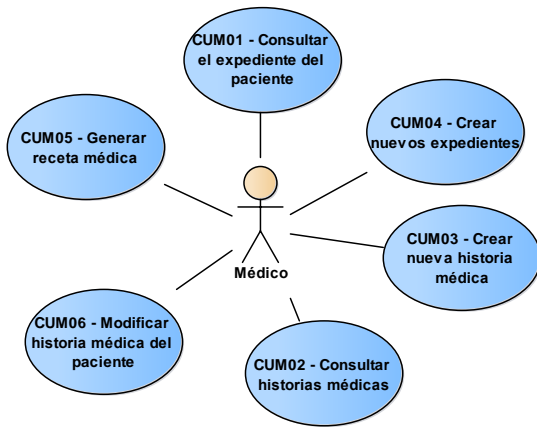


Fig. 4. Casos de uso primarios de Médicos.

En la Fig. 4, se observa el diagrama del actor que corresponde al médico responsable del consultorio, que como función principal desempeña el registro de consultas médicas, reportes de evaluación, notas médicas y entrevista inicial de los pacientes que acuden al Área de Servicio.

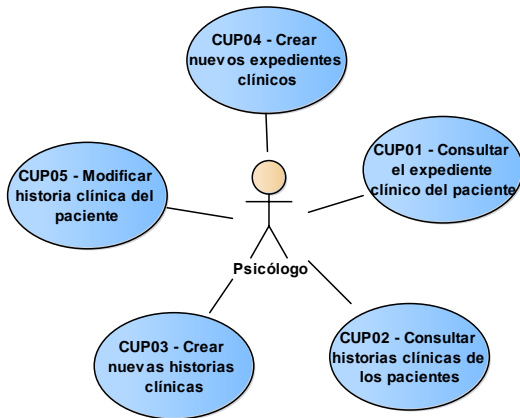


Fig. 5. Casos de uso primarios de Psicólogos.

En la Fig. 5, se observa el diagrama del actor que corresponde al psicólogo responsable del consultorio, que como función principal desempeña el registro de consultas psicológicas, reportes de evaluación, reportes de evolución, reportes de orientación, consejería y entrevista inicial.

D. Diagrama de Clases

A continuación se muestra en la Fig. 6, el diagrama de clases y los atributos que se comunican en la aplicación. Esta capa de abstracción implementa las operaciones CRUD [7] y el patrón de diseño MVC [8] [9], lo que permite que se realicen peticiones a la base de datos reduciendo el tiempo de respuesta, además, de esta manera al actualizar una funcionalidad en el código fuente facilita la mantenibilidad y beneficiando así la seguridad de la aplicación.

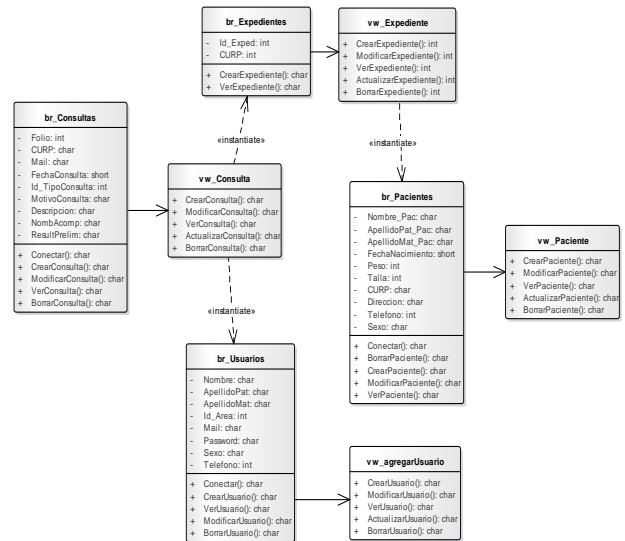


Fig. 6. Diagrama de clases de la herramienta propuesta.

E. Diagrama de Despliegue

Este apartado tiene como objetivo principal mostrar la estructura interna de la aplicación, los objetos que la componen y la comunicación que existe entre cada uno de ellos (Véase Fig. 7).

- PC: Es una computadora de escritorio en la que, por medio de un explorador web (Mozilla, Opera, Chrome) se va a operar la aplicación web dentro del Área de Servicio.
- Dispositivo Móvil: Es un dispositivo portátil (Smartphone, Tablet) en el que, por medio de un explorador web (Mozilla, Opera, Chrome, Safari) se va a operar la aplicación web. Este dependerá del tipo de sistema operativo con el que cuente el dispositivo, sea Android, IOS o Kirin OS, por mencionar algunos ejemplos.
- Servidor de Hosting: En este se va a alojar el contenido de la aplicación. Se divide en dos secciones:
 - Servidor Web: En este se encontrarán los componentes de la WebApp, y se encargará de administrar el acceso a ella.
 - Servidor de DB: Cuenta con el servidor de MySQL, el cual tendrá la responsabilidad de manipular y administrar el acceso a los datos de la DB.

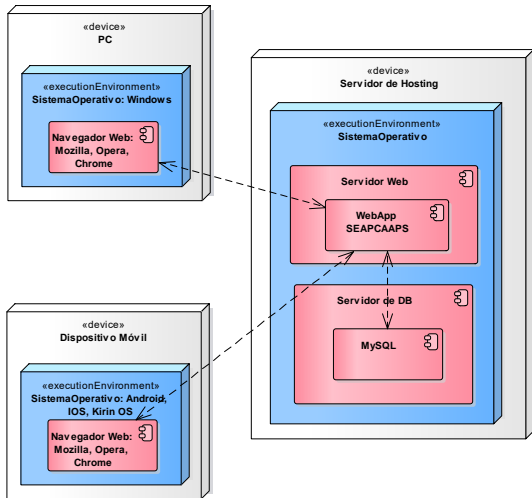


Fig. 7. Diagrama de despliegue de la herramienta propuesta.

F. Diagrama Entidad-Asociación

El sistema tiene como función principal mantener un registro digital de los datos de los pacientes atendidos en el Servicio Amigable para Adolescentes, en el cual a partir de que se establece un inicio de sesión, se verifica que tipo de usuario se conecta al sistema y se establece su inicio de sesión. Cada uno de estos usuarios tiene determinado un tipo en específico de permisos para manipular el sistema, todos estos otorgados por el que es el o la Jefe del Servicio, y por el Administrador del Sistema.

Por lo que es necesario mencionar que este sistema cuenta con dos diagramas de Entidad-Asociación en los que se observan las tablas que se comunican en la interfaz de los usuarios: Médicos, psicólogos, jefe/a de servicio, recepcionistas (Véase Fig. 8), y las tablas que se comunican en la administración y permisos que se otorgan para el manejo de la aplicación (Véase Fig. 9).

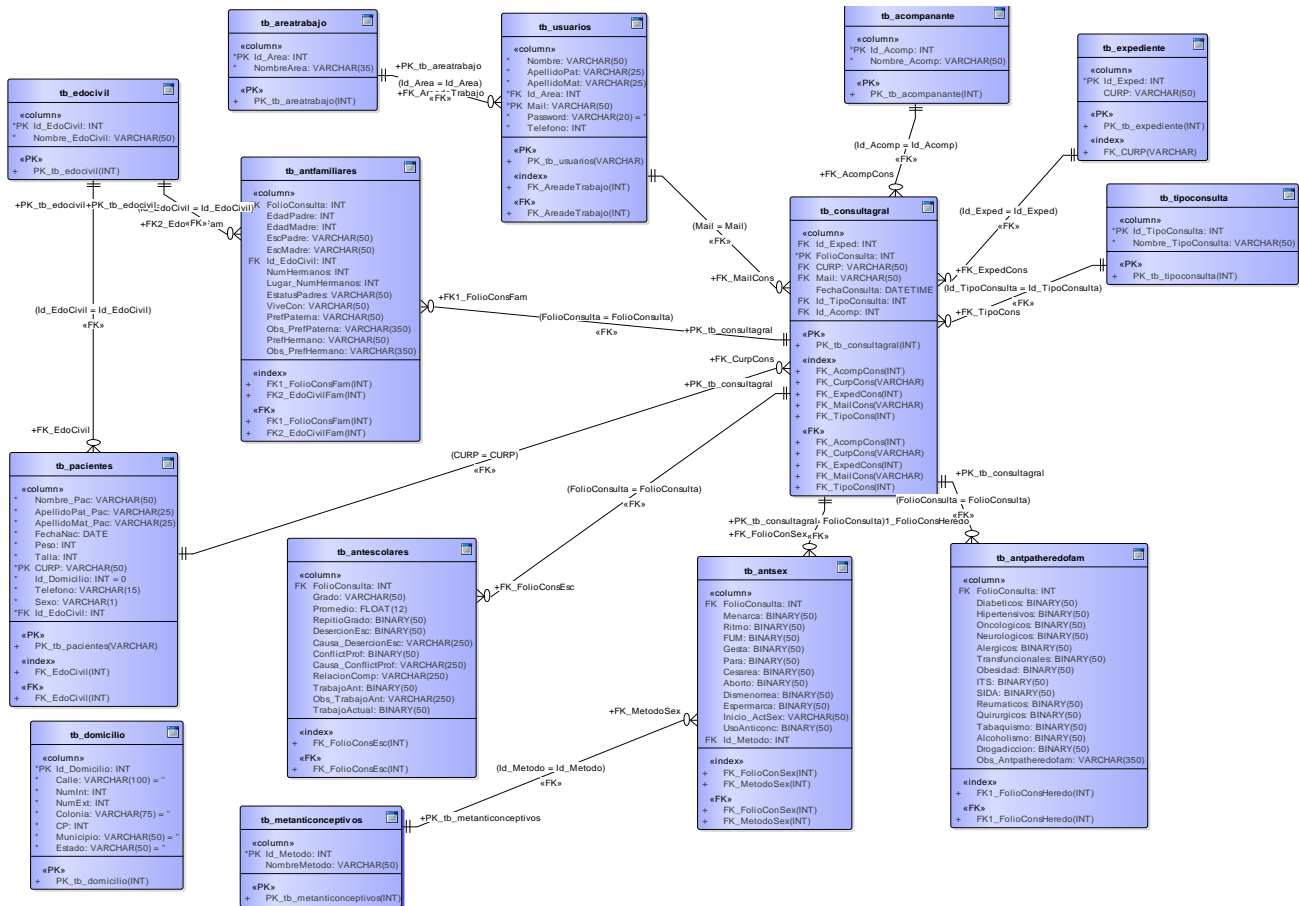


Fig. 8. Diagrama Entidad Asociación de la interfaz para usuarios del Servicio.

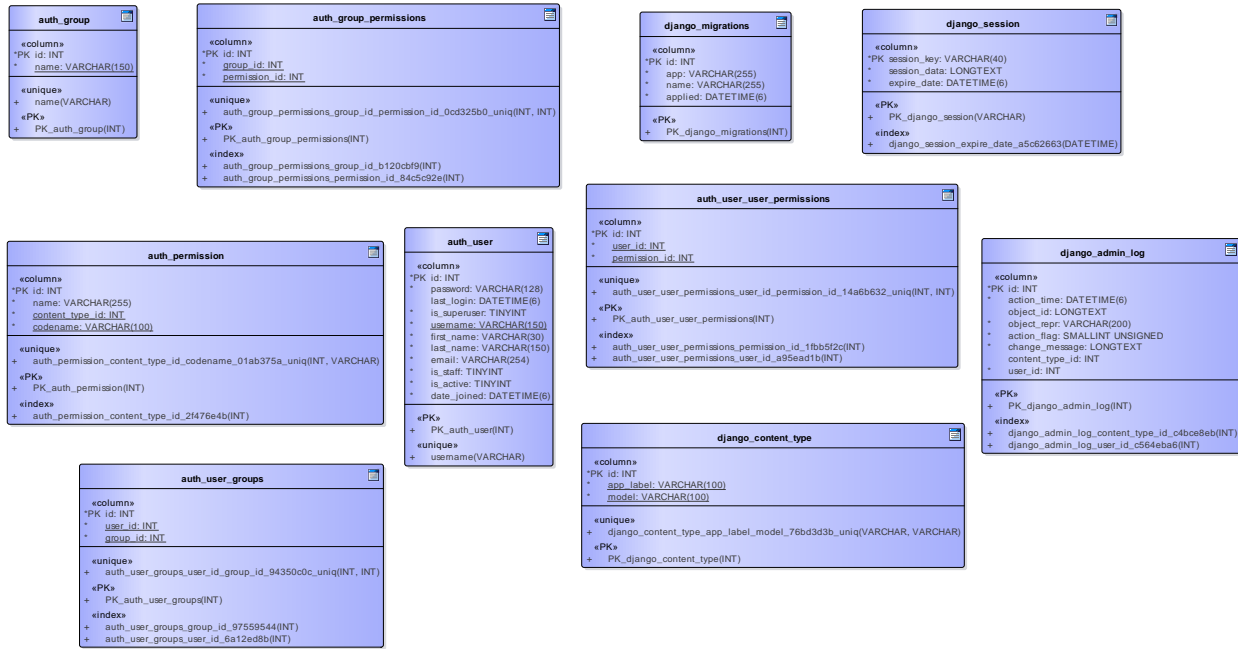


Fig. 9. Diagrama Entidad Asociación del Administrador de permisos y usuarios.

El sistema está conformado por los siguientes módulos:

- **Módulo de Administrador:** Este módulo es por el cual al iniciar sesión se muestra una vista como administrador en el cual se visualizan los datos de usuario y los permisos otorgados por el Administrador o Jefe de Servicio.
- **Módulo de Pacientes:** Este módulo se encarga del registro de los datos de entrada de los pacientes que acuden a consultas, y estos datos solo son registrados por el recepcionista del Servicio. Se capturan y/o actualizan, en caso de ser necesario, los datos de evaluación inicial del paciente tales como nombre del paciente, fecha de nacimiento, peso, estatura, Clave Única de Registro de Población (CURP), dirección y teléfono.
- **Módulo de Expedientes:** Dentro de este módulo se encuentran registrados los expedientes que están asignados a cada uno de los pacientes, estos se dan de alta justo cuando un paciente ha acudido por algún tipo de consulta o consejería al área del Servicio.
- **Módulo de Registro de Consultas:** Este módulo se encarga de la generación de registro de los datos de las consultas otorgadas a los pacientes del Área Servicio Amigable y las recetas médicas expedidas, cabe detallar que estos datos solo son registrados y manipulados por los médicos y psicólogos del Servicio.

Dentro de este módulo se otorgan los siguientes servicios:

- **Entrevista inicial:** Este tipo de consulta se caracteriza por ser la primera que se otorga a

los pacientes del servicio. Consta de un cuestionario inicial en el que se busca recabar los datos personales y familiares del paciente, antecedentes patológicos y no patológicos, personalidad, exploración física, hábitos, actividades recreativas, entre otros.

- **Orientación y consejería sexual:** Este tipo de consulta se caracteriza por otorgar al paciente una consulta de tipo informativa sobre algún tema en particular sobre planificación familiar o información acerca de algún método anticonceptivo.
- **Prescripción de métodos anticonceptivos:** En este tipo de atención, a diferencia del anterior, se limita a la salida de métodos anticonceptivos por solicitud de un paciente del servicio.
- **Anticoncepción de emergencia:** En este tipo de consulta se registra si una paciente acude por el método de anticoncepción de emergencia.
- **Atención de ITS:** En este tipo de consulta se inicia el seguimiento de una infección de transmisión sexual diagnosticada en el paciente que acudió al servicio. Tiene como sucesores futuras consultas para el tratamiento de la ITS detectada de acuerdo al tipo que sea y a la forma en que se le haya sido detectada.
- **Otros servicios**

V. CONCLUSIONES

Este proyecto consta de una serie de trabajos, de los cuales, siendo este el segundo artículo, se describe la tecnología utilizada para el desarrollo y la codificación de la aplicación para expedientes del Servicio Amigable para Adolescentes. Se definieron las funciones de los usuarios involucrados en esta y se documentaron los casos de uso principales para otorgar los permisos en la fase de codificación de la aplicación. Para la parte administrable se utiliza el administrador que otorga el framework Django de Python, ya que esto facilita el manejo de los permisos que se otorgan a los usuarios, además de que cuenta con una bitácora de la base de datos, por lo que se puede enfocar el resto del tiempo en el desarrollo de la funcionalidad de la aplicación.

Como se describió en el artículo previo, se hizo la selección del patrón de diseño de software Modelo Vista Controlador (MVC) por su facilidad para implementarse en el desarrollo web, además de que con este patrón se crea código reutilizable para la mayoría de las interfaces, y se separan las reglas de negocio de la representación visual y el manejo de los datos de la base de datos, lo cual añade seguridad al contenido de la aplicación.

Para la generación de las vistas se utiliza la biblioteca de Bootstrap para facilitar el diseño final, además de lenguaje HTML5, actualmente de los módulos propuestos se encuentran generados el módulo de Administración de sistema, Pacientes y

Expedientes, se han realizado pruebas para el alta y consulta de datos de todos los módulos. El módulo de Registro de Consultas se encuentra en una etapa de representación, en la que solo se visualizan los datos de las consultas.

REFERENCIAS

- [1] Sánchez Zapata, A. A. (15 de Agosto de 2019). Implementación de una Aplicación Web para el Registro y Control de Expedientes Médicos de la Clínica Avanzada de Atención Primaria a la Salud (C.A.A.P.S): Caso "Área de Servicio Amigable (Módulo del Adolescente)". *Memorias del Congreso Internacional de Investigación Academia Journals Puebla 2019*. Puebla, Puebla, México.
- [2] Systems, S. (09 de Febrero de 2019). *Enterprise Architect*. Obtenido de Sparx Systems: <https://sparxsystems.com/products/ea/>
- [3] Oracle. (2014). *Manual de Referencia de MySQL 5.0*. Obtenido de MySQL Community: <https://downloads.mysql.com/docs/refman-5.0-es.pdf>
- [4] Challenger-Pérez, I., Díaz-Ricardo, Y., Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, XX(2), 1-13.
- [5] Loaiza Sojos, E. L., Molina Ríos, J. R., Loja Mora, M. N. & Zea Ordoñez, M. P. (2016). Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python.
- [6] Sommerville, I. (2011). *Ingeniería de Software*. Pearson.
- [7] León Soberón, J. J. (2020). Análisis comparativo de sistemas gestores de bases de datos Postgresql y MySQL en procesos CRUD.
- [8] Fernández, Y., & Díaz, Y. (2012). Patrón Modelo-Vista-Controlador. *Telemática*, Vol. 11, 47-57.
- [9] Pressman, R. (2010). Diseño basado en patrones. En R. Pressman, *Ingeniería del Software: Un enfoque práctico* (págs. 295-316).

Diseño de una Aplicación Móvil para el control de una flotilla vehicular de Recolección de Residuos Sólidos Urbanos

Ing. Leslie Yajaira Peña Vázquez
Estudiante de la Maestría en
Sistemas Computacionales en un
Programa PNPC
Tecnológico Nacional de México
/IT de Acapulco
Acapulco, Guerrero, México
leylapena22@gmail.com

M.T.I. Jorge Carranza Gómez
Docente de la Maestría en
Sistemas Computacionales
Tecnológico Nacional de México
/IT de Acapulco
Acapulco, Guerrero, México
jcarranzamx@gmail.com

M.T.I. Juan Miguel Hernández
Bravo
Docente de la Maestría en
Sistemas Computacionales
Tecnológico Nacional de México
/IT de Acapulco
Acapulco, Guerrero, México
jmhernan@gmail.com

M.T.I. Rafael Hernández Reyna
Docente de la Maestría en
Sistemas Computacionales
Tecnológico Nacional de México
/IT de Acapulco
Acapulco, Guerrero, México
rhernan7@yahoo.com.mx

M.S.C. Juan C. Ramírez Teresa
Docente de Lic. En Sistemas
Comp.
Tecnológico Nacional de México
/IT de Acapulco
Acapulco, Guerrero, México
janramter@gmail.com

Resumen—En este artículo, se plasma el trabajo interdisciplinario de la Maestría en Sistemas Computacionales con apoyo del CONACYT, impartida en el Instituto Tecnológico de Acapulco. El objetivo de este estudio es analizar el sistema de recolección de residuos sólidos urbanos en el municipio de Coyuca de Benítez, Guerrero. Presentando una alternativa de localización GPS para la empresa por medio del diseño de una aplicación móvil que permita obtener el seguimiento de las rutas de recolección de residuos sólidos urbanos, con esta aplicación el administrador podrá observar la posición actual de cada vehículo en tiempo real y el control del personal en operación, conociendo la ruta y horarios mediante el aplicativo móvil.

Palabras Clave—Residuos Sólidos Urbanos (RSU), Aplicación Móvil, GPS.

I. Introducción

La Dirección de saneamiento básico del municipio de Coyuca de Benítez, se encarga de administrar los servicios de limpia, recolección, transporte y disposición final de los residuos sólidos del municipio. Esta empresa tiene como objetivo desarrollar y aplicar acciones que permitan la conservación del medio ambiente en beneficio de la población llevando a cabo una buena organización de recolección de residuos sólidos urbanos del municipio[1].

Actualmente la empresa realiza de forma manual la asignación de las rutas a sus conductores, desconoce la posición

actual de su vehículo debido a que solo se conoce la ubicación del conductor por medio de llamadas telefónicas.

Con la ayuda de la tecnología GPS obtenemos con más precisión la ubicación de cualquier dispositivo móvil. La monitorización de vehículos es útil para la administración de los mismos durante el transcurso del día, por medio de envío de notificaciones cuando los vehículos realicen una parada no establecida la cual conlleva a algún tipo de atraso de la ruta establecida, incluso es útil para los conductores de la empresa.

Las coordenadas de ubicación serán presentadas de manera gráfica en una serie de mapas que nos proporciona el servicio de Google Maps, lo cual es más entendible para el "Administrador". La monitorización automatizada de vehículos puede ser muy útil no sólo para la administración de los mismos, sino incluso para los propios conductores dentro de la empresa [2].

El propósito del desarrollo de esta aplicación es automatizar los procesos que actualmente maneja la empresa de Dirección de Saneamiento Básico del Municipio de Coyuca de Benítez, permitiendo al Coordinador obtener la ubicación en tiempo real de los vehículos, donde verifica por medio de la app que cada conductor desempeñe la ruta de trabajo establecida al inicio del día laboral. Además del seguimiento que se le dará al vehículo a través de los mapas, la aplicación también mostrará al administrador si algún conductor salió de su ruta [3].

Esta App ofrecerá la ubicación en tiempo real de la unidad monitoreada y el control del personal en operación, del mismo modo realiza un envío de notificaciones por paradas no planificadas de la unidad de transporte, donde estas notificaciones generadas son enviadas al Administrador de la aplicación, el cual podrá tomar las medidas necesarias en tiempo y forma.

II. Objetivo General

Desarrollar una aplicación para dispositivos móviles con sistema operativo Android, para el monitoreo de las rutas vehiculares designadas a los conductores de recolección de residuos sólidos urbanos.

Lugar	Nota	Fecha	Fuente	Liga
Guadalajara, Jal.	"Prometen GPS en camiones de basura"	1 Ene. 2009	El Informador	http://limpiemoselsalto.blogspot.mx/2009/01/prometen-gps-en-camiones-de-basura.html
Tijuana, B.C.	"Municipio compra GPS para camiones de basura"	19 Dic. 2012	El Sol de Tijuana	http://www.oem.com.mx/elsoldetijuana/notas/n2813202.htm
Tijuana, B.C.	"Ayuntamiento coloca GPS en obsoletos camiones de basura"	5 Oct. 2013	www.zetatijuana.com	http://zetatijuana.com/2013/10/05/ayuntamiento-coloca-gps-en-obsoletos-camiones-de-basura/
Cajeme, Son.	"Eficientan recolección de basura con GPS en unidades"	19 Mar. 2010	elregionaldesonora	http://www.elregionaldesonora.com.mx/noticia/6632
Cajeme, Son.	"Contarán con GPS camión recolector de basura"	10 Sep. 2011	correorevista.com	http://correorevista.com/sur/contaran-con-gps-camion-recolector-de-basura/
Ciudad Madero, Tamps.	"Contaran con GPS camión recolector de basura"	15 Sep. 2011	Panucoaldia diariodebate	http://panucoaldia.diariodebate.info/index.php/2011/09/d16019-vigilaran-con-gps-a-camiones-recolectores-de-basura/
Matamoros, Tamps.	"Instalarán GPS a camiones de basura de Matamoros"	4 May. 2012	Grupo mi radio	http://www.grupomiradio.mx/portal/?p=55330
Aguascalientes, Ags.	"Controlará GPS rutas de basura"	8 Ago. 2013	Periódico am	http://www.am.com.mx/aguascalientes/local/controlara-gps-rutas-de-basura-37716.html
Guanajuato, Gto.	"Monitorean con GPS camiones de la basura"	7 Ene. 2014	El Sol de León	http://www.oem.com.mx/elsoldeleon/notas/n3248180.htm
Ciudad Juárez, Chih.	"Podrán incluir GPS para eficientar levantamiento de basura"	21 Feb. 2014	Noticias de Chihuahua	http://www.larednoticias.com/noticias.cfm?n=126150
Culiacán, Sin.	"Rastreo GPS para la Gestión de Residuos"	2014	GPS América	http://www.gpsamerica.com.mx/cms/index.php/en/mercados/gobierno/recoleccion-de-basura

Fig. 1 Resumen del uso de Sistemas de Geo Posicionamiento Global (GPS) al Servicio de Recolección de Residuos Sólidos Urbanos en ciudades mexicanas [4].

IV. Descripción del Método

A. Modelo de Negocio

Se elaboró el diagrama de modelado de negocio (Fig. 2), en el cual se muestran los procesos que se llevan a cabo en cada departamento en la Dirección de Saneamiento Básico del Municipio de Coyoaca de Benítez, para realizar el seguimiento de las rutas de recolección de residuos sólidos urbanos.

A continuación, se explica brevemente en que consiste cada proceso del modelado de negocio:

- Registra Asignación de Visitas a Operadores: En este proceso el Coordinador asigna a cada Operador la zona a recorrer durante el día.
- Consultar Ruta Asignada: Cada Operador deberá actualizar datos para poder visualizar la zona asignada por el Coordinador del Departamento Operativo.
- Salir a Ruta: Una vez asignada la ruta a cada Operador este deberá salir a su destino.
- Registrar Visitas: EL Operador deberá registrar cada contenedor visitado en la zona asignada y registrar en el sistema su hora de llegada y salida de este mismo.
- Realizar Reporte: Cada Operador deberá realizar un reporte de las actividades realizadas sobre la zona asignada durante el día, y este podrá ser visualizado en línea por el Coordinador del Departamento Operativo.
- Revisa Visitas: El coordinador podrá visualizar en línea si el Operador cumplió con su ruta asignada, por medio de un mapa que mostrara el recorrido de ese día.

- Revisa Actividades: El coordinador visualiza en el reporte las actividades realizadas por el Operador durante su recorrido.

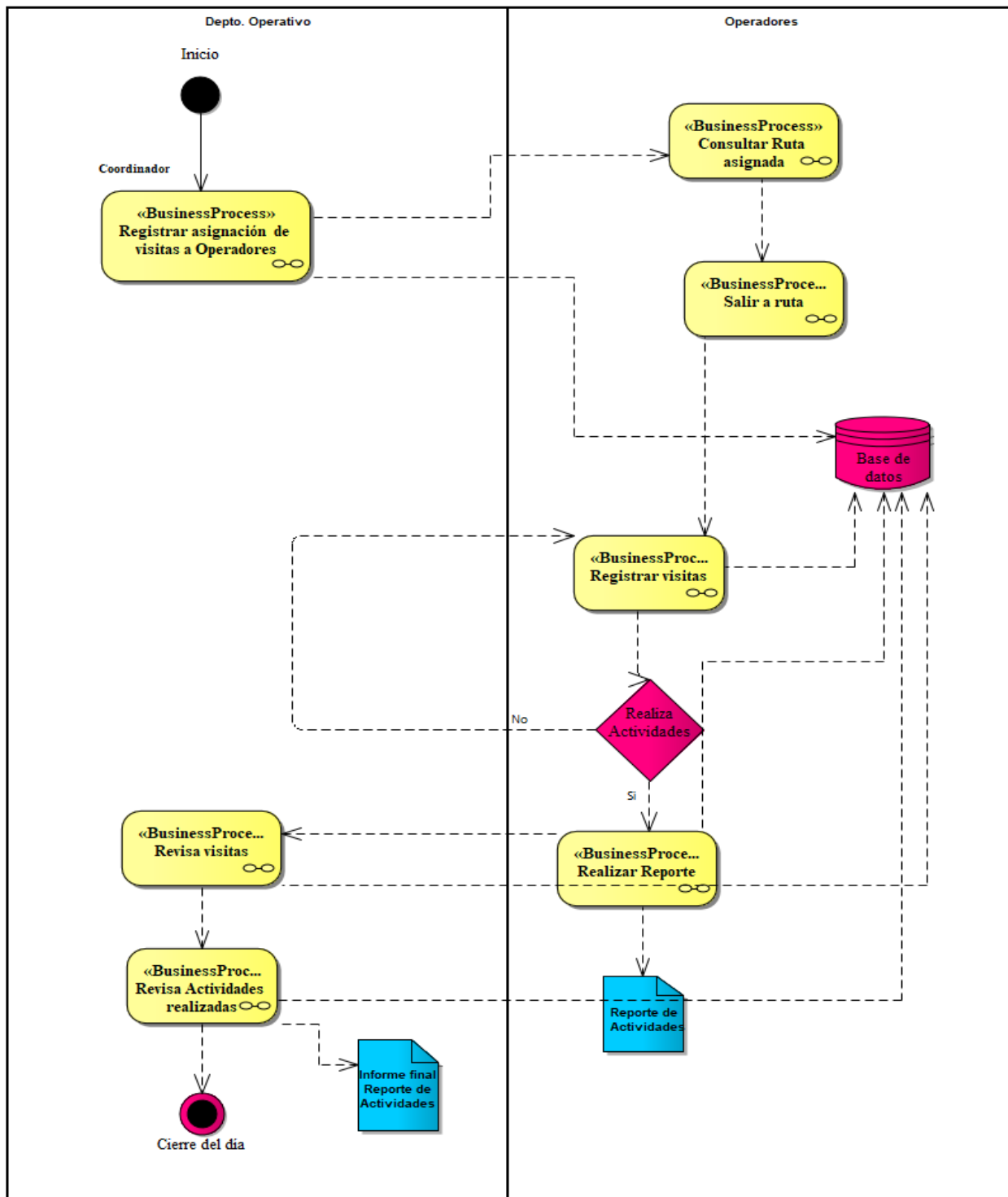


Fig.2 Modelado de Asignación de Ruta

B. Caso de Uso Principal

Caso de uso – Asignación de ruta

Las personas a las cuales están enfocado el uso de esta aplicación son el personal del departamento de atención y seguimiento ciudadano, departamento operativo, auxiliar operativo y brigadas integrales de la

Dirección de Saneamiento Básico del Municipio de Coyuca de Benítez, como punto básico, todo el personal debe poseer un Smartphone o Tablet con sistema operativo Android.

En la **Fig. 3** se muestra la interacción entre el usuario y el sistema, es decir, son las acciones que puede realizar cada usuario del sistema dependiendo del rol que tenga.

DCU-Seleccionar una opción del menú rutas.

- Administrador. Es el usuario que va asignando a cada operador su ruta de recorrido y vehículo a utilizar durante el día.
- Operador. Este usuario al entrar al menú podrá ver su ruta y vehículo asignado y visualizar su mapa del recorrido que tendrá que hacer durante el día.

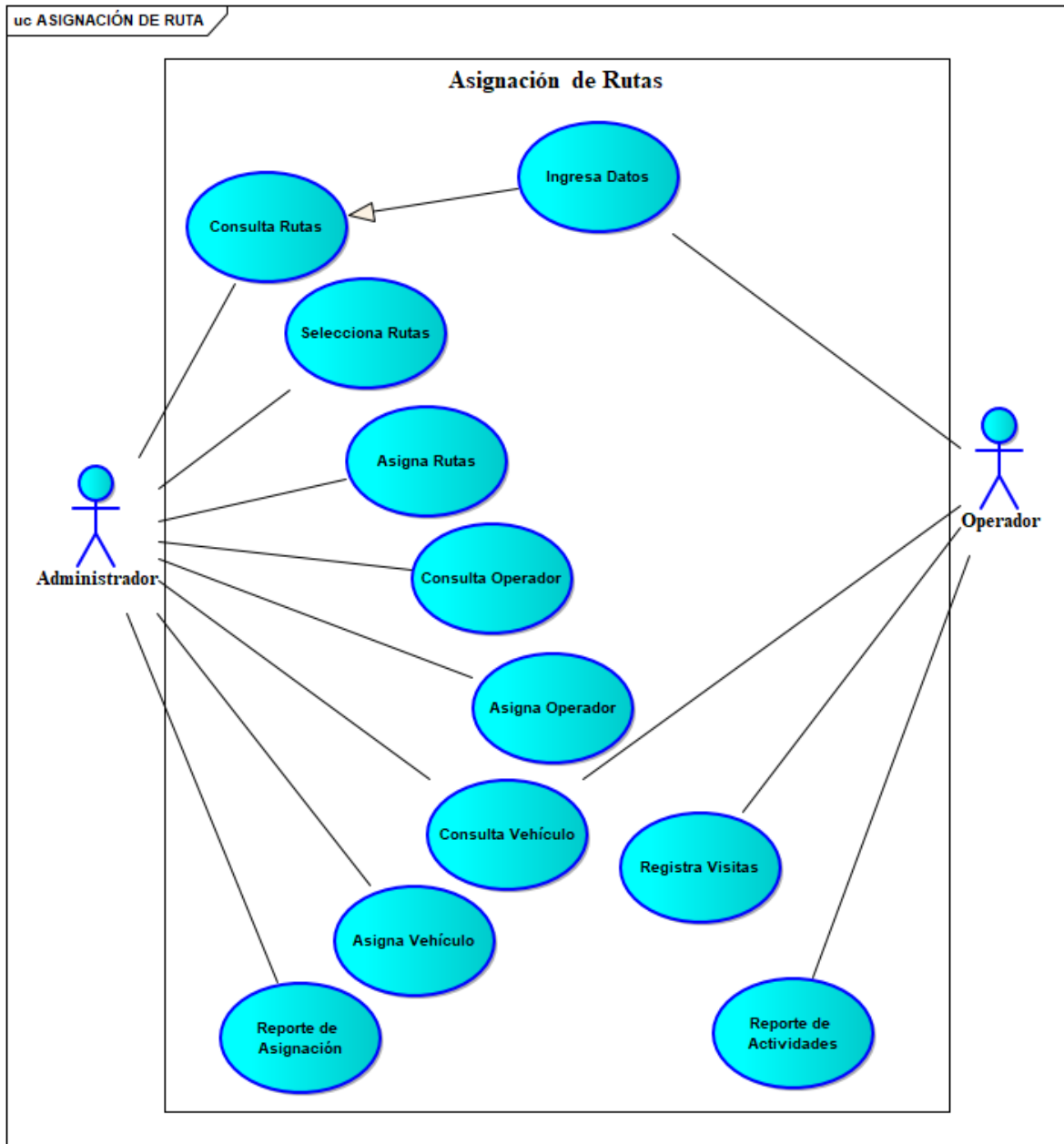


Figura 3 CDU-Asignación de Ruta

ID	Descripción	Característica Asociada
<i>CUR01</i>	Consultar ruta: El administrador ingresa al menú principal de la ventana rutas y consulta las rutas.	CAR-03
<i>CUR02</i>	Selecciona rutas: El operador podrá consultar la ruta del día que le ha sido asignada por el administrador.	CAR-06
<i>CUR03</i>	Asigna rutas: El sistema le muestra al operador un mapa con la trayectoria a recorrer durante el día, una vez que esta haya sido asignada por el administrador.	CAR-06
<i>CUR04</i>	Consulta Operador: el administrador consulta la lista de operadores.	CAR-03
<i>CUR05</i>	Asigna Operador: el administrador asigna un operador a cada ruta.	CAR-03
<i>CUR06</i>	Consulta Vehículo: el administrador consulta los vehículos disponibles durante el día.	CAR-03
<i>CUR07</i>	Asigna Vehículo: el administrador asigna un vehículo a cada operador, este depende de la ruta que le haya sido asignada.	CAR-06
<i>CUR08</i>	Reporte de Asignación: el administrador genera un reporte en el cual aparecen los datos del Operador, la matrícula y tipo de vehículo que le fue asignado para realizar el recorrido de su ruta.	CAR-04

Tabla 1 CDU-Asignación de ruta 1

ID	Descripción	Característica Asociada
<i>CUR01</i>	Ingresar datos: El operador ingresa al sistema para ver las asignaciones del día.	CAR-03
<i>CUR02</i>	Consultar ruta: El administrador ingresa al menú principal de la ventana rutas y consulta las rutas.	CAR-06
<i>CUR03</i>	Consulta vehículo: El operador consulta el vehículo que le fue asignado para realizar el recorrido de su ruta.	CAR-06
<i>CUR04</i>	Registra visitas: El operador registra las visitas a cada contenedor (hora de llegada y hora de salida de cada contenedor) de la ruta que le fue asignada.	CAR-03
<i>CUR05</i>	Reporte de actividades: El operador genera un reporte detallado en el cual especifica cada una de las tareas realizadas a cada contenedor que visito y especifica los kilogramos de residuos sólidos urbanos generados durante el día.	CAR-04

Tabla 2 CDU-Asignación de ruta 2

C. Diagrama de Clases

Se elaboró un sólo diagrama de clases que contempla el proceso analizado (Fig. 4). A continuación se describen cada una de las clases:

- Usuario: Esta clase representa la información de los usuarios que están registrados en la aplicación. Es una clase tipo formulario, donde se tiene que capturar el correo electrónico y contraseña del usuario.
- Contenedores: Esta clase representa la información relacionada con los contenedores a los cuales se les da mantenimiento. El formulario cuenta con el id contenedor y el id zona para conocer la ubicación exacta de este.

- Zona: Esta clase representa la información almacenada de las colonias a las que se realiza el seguimiento, contando con el id zona, nombre de la zona y ubicación.
- Vehículo: Esta clase contiene la zona ala que le toca recorrer a cada vehículo, el id del empleado y la capacidad de carga de este.
- Reportes: Esta clase genera reportes de las actividades realizadas por los Operadores durante el día.
- Empleados: Esta clase contiene los datos de cada empleado, nombre, dirección y teléfono.
- Asignación: Esta clase genera la asignación de vehículo, empleado y zona la que se le brindara el servicio durante el día.

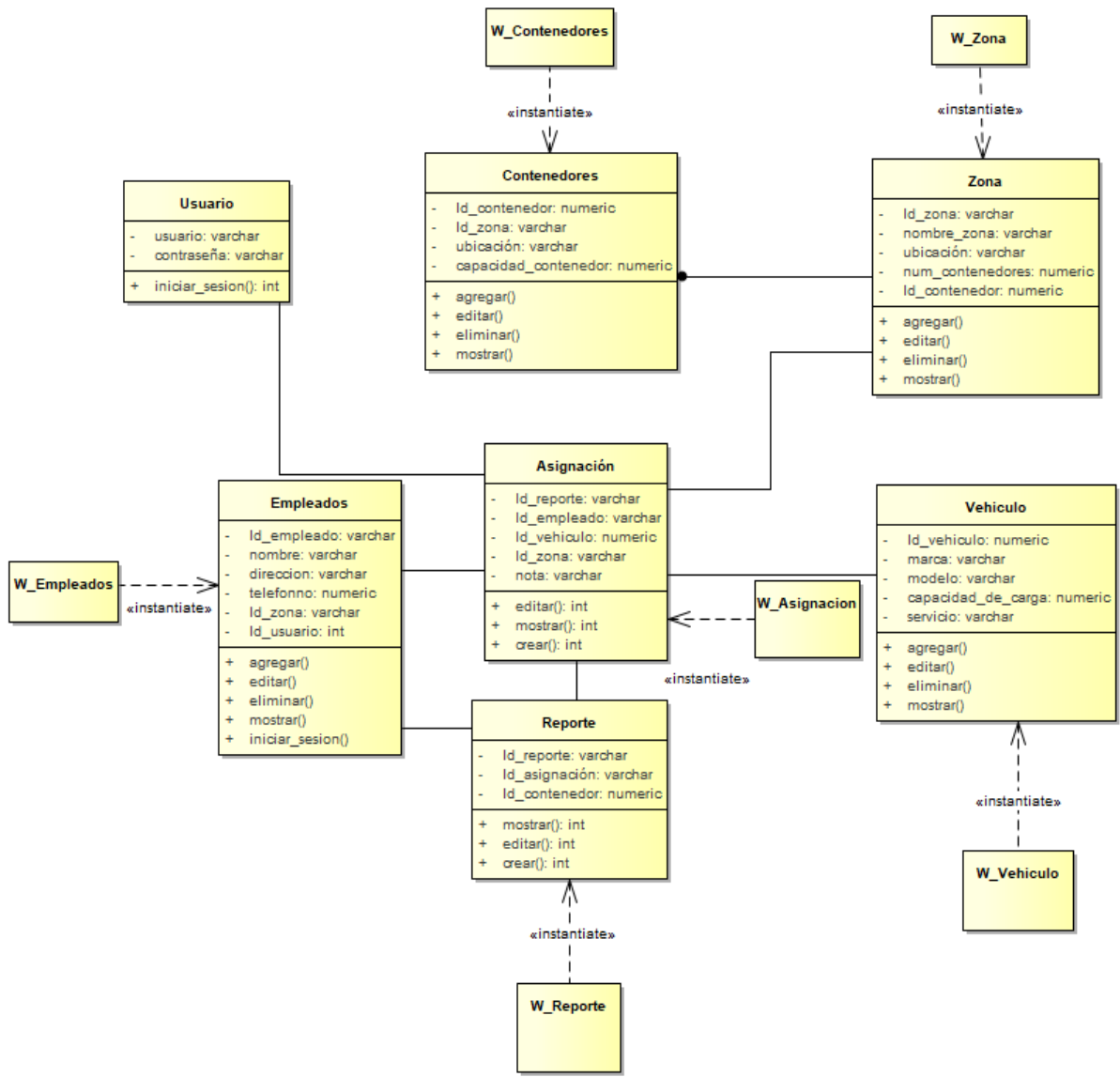


Fig. 4 Diagrama de Clases

D. Modelo Navegacional

Este modelo define el acceso que se le brindara a cada uno de los usuarios, en este caso Administrador y Operador (Fig. 5).

El modelo de navegación tiene como objetivo la representación de los nodos y enlaces de la estructura de hipertexto, y el diseño de las rutas de navegación mediante diagramas de clases UML [5].

Está basado en un grupo de esquemas de navegación que muestran a cada usuario la navegación establecida dentro de la aplicación.

A continuación se muestra una vista abstracta en la cual se ilustran las vistas de la aplicación que obtendrá cada uno de los usuarios.

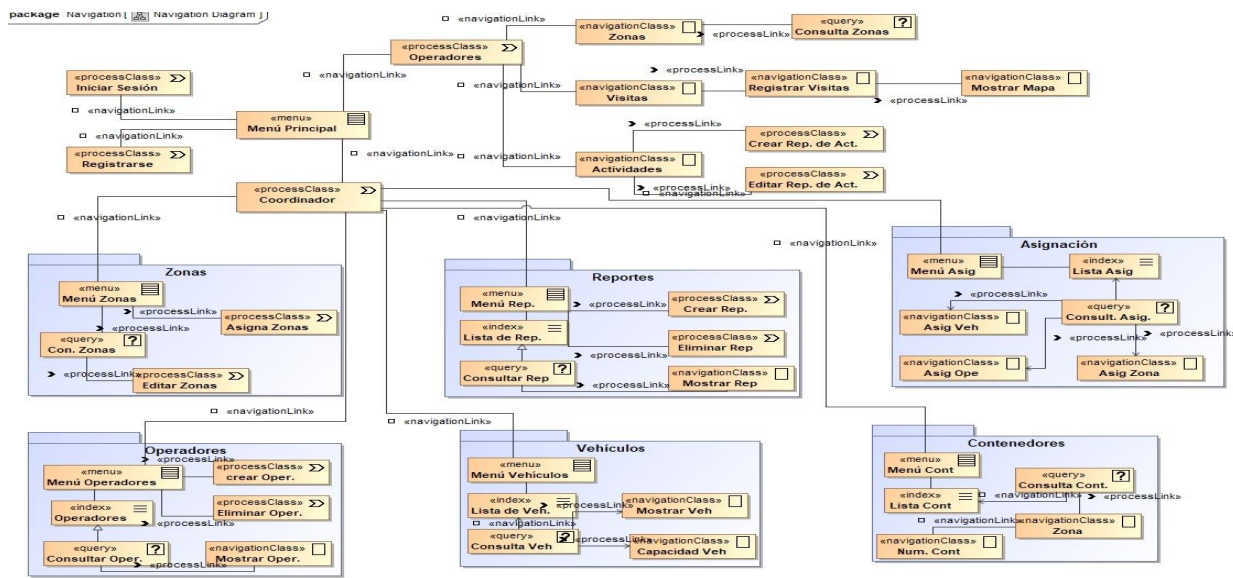


Fig. 5 Modelo Navegacional

E. Diagrama de Despliegue

En el diagrama de paquetes o diagrama de distribución se muestra las partes del sistema de lo que se tiene que instalar en cada nodo (Fig. 6). Se muestra la descripción de este diagrama en la tabla 3.

<i>Elemento</i>	<i>Responsabilidad</i>	<i>Propiedades</i>
<i>Pc del usuario</i>	<i>Computadora personal mediante la cual accederá el administrador del sistema.</i>	<i>Navegador= Google Chrome</i>
<i>Dispositivo Móvil</i>	<i>Dispositivos por medio de los cuales se accederá en el futuro a la aplicación.</i>	<i>IOS, Android, Windows form</i>
<i>Servidor web</i>	<i>El servidor web alberga las capas de presentación y servicios web.</i>	<i>S.O Ubuntu / Windows server</i>
<i>Servidor BD</i>	<i>Este servidor alberga la base de datos.</i>	<i>SQL SERVER</i>

Tabla 3 Descripción de los Nodos en el Diagrama de Despliegue

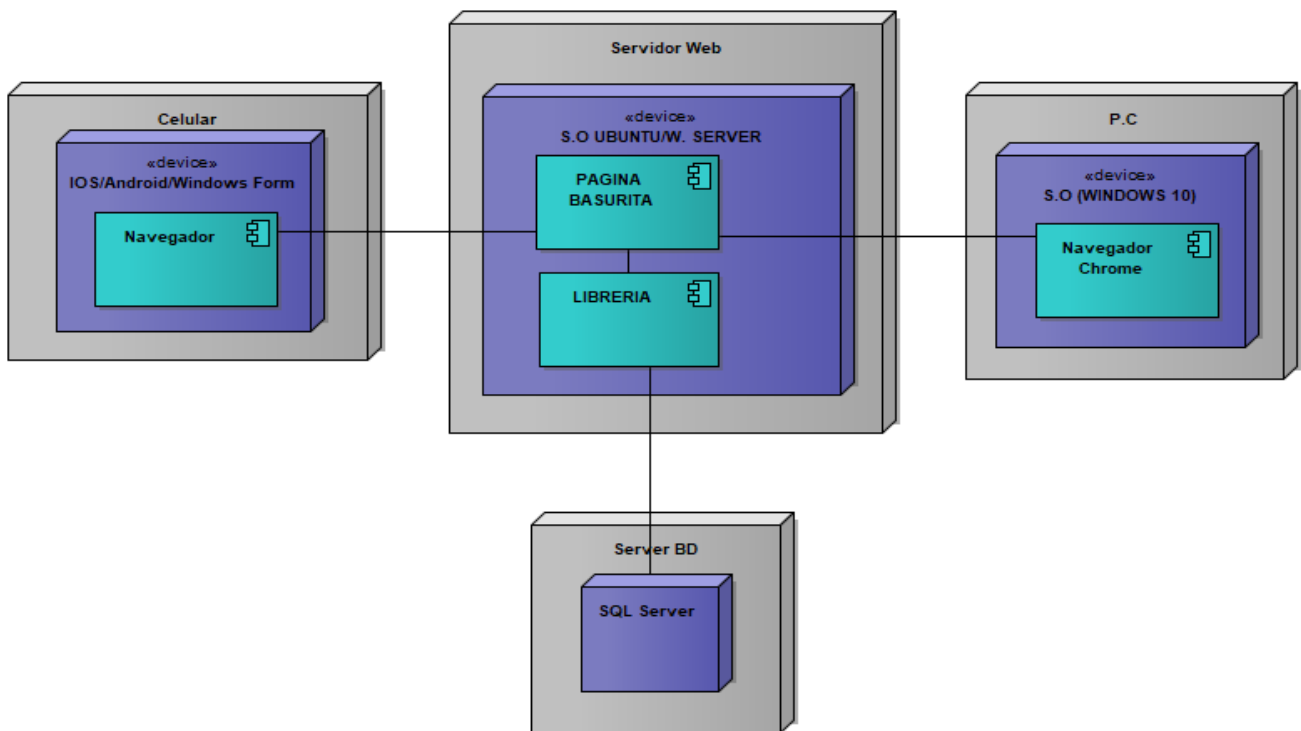


Fig. 6 Diagrama de Despliegue

F. Diagrama Entidad-Relación

Este diagrama ilustra como las entidades se relacionan entre si dentro del sistema (Fig. 7). A continuación se describen cada una de las entidades:

- **Asignación:** Esta tabla genera los reportes de las rutas asignadas a los operadores durante el día.
- **Contenedores:** Esta tabla nos da los datos de cada contenedor, dando su ubicación y capacidad.
- **Empleados:** Esta tabla contiene los datos de cada empleado.
- **Reportes:** Esta tabla genera reportes de las actividades realizadas por los Operadores durante el día.
- **Vehículos:** Esta tabla contiene la zona ala que le toca recorrer a cada vehículo, el id del empleado y la capacidad de carga de este.

- Zona: Esta tabla contiene la ubicación y rutas que abarcará el vehículo de recolección durante su recorrido.

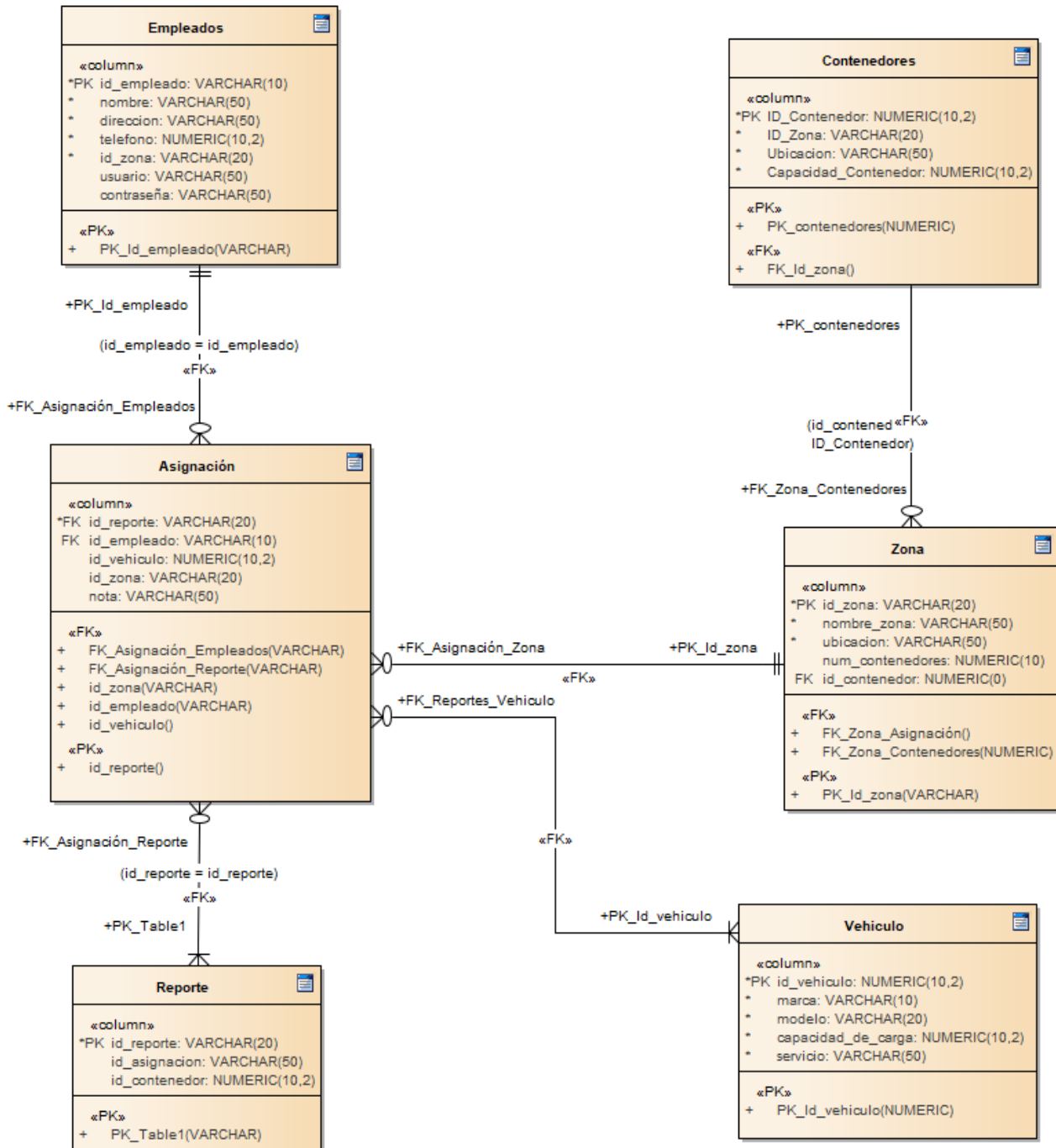


Figura 3 Diagrama Entidad-Relación

V. Conclusiones

En base al análisis realizado en la problemática de la Dirección de saneamiento básico, se generaron los diagramas UML para la creación de los módulos que la integrarán, servirán como base para la codificación y pruebas en trabajos futuros.

El crecimiento en la adquisición de dispositivos móviles junto a la ampliación de las redes de datos resulta muy beneficioso para el desarrollo de esta aplicación.

Con esta aplicación se espera mejorar el proceso que actualmente realiza el Departamento Operativo de la Dirección de Saneamiento Básico del Municipio de Coyuca de Benítez.

Una vez terminadas las etapas anteriores, se posee un completo conocimiento del dominio del problema.

Además de ello, se cuenta con una idea básica de cómo se verán las interfaces. Para comenzar con la implementación se

eligió donde almacenar los objetos y con qué lenguaje o herramienta desarrollar las interfaces.

Referencias

- [1] Dirección de Saneamiento Básico Municipal (2018), Datos estadísticos sobre residuos sólidos, Acapulco, Gro.
- [2] Google Map. (2017). Google Maps. Obtenido de Google Maps APIs: <https://enterprise.google.com/intl/es-419/maps/products/mapsapi.html>. Elissa, "Title of paper if known," unpublished.
- [3] L.Y.P Vazquez, <<<Desarrollo de una Aplicación Móvil para el control de una flotilla vehicular de Recolección de Residuos Sólidos Urbanos >> Congreso Internacional Academia Journals Puebla 2019, vol2, pp 272-277, 2019.
- [4] Betanzo-Quezada, Eduardo. << Evaluación de rutas de recolección de residuos sólidos urbanos con apoyo de dispositivos de rastreo satelital: análisis e implicaciones >>. *Revista internacional de contaminación ambiental*, 32(3), 323-337(2016).
- [5] (Sommerville, 2011) <<ingeniería del Software 9na Edición >>

Patrones de acceso de una aplicación basada en Scrum para la gestión de proyectos de desarrollo de software

Ing. José Raúl López Morales
Estudiante de Maestría en
Sistemas Computacionales
programa PNPC
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
jraul_lopez@hotmail.com

M.C. José Francisco Gazga
Portillo
Docente de Maestría en Sistema
Computacionales
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
ita.gazga@gmail.com

M.T.I. Juan Miguel Hernández
Bravo
Docente de Maestría en Sistema
Computacionales
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
jmhernan@yahoo.com

M.I.D.S. Alma Delia de Jesús
Islao
Docente de Maestría en Sistema
Computacionales
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
alma.islao.ita@gmail.com

Ing. Rogelio Ramírez Silva
Estudiante de Maestría en
Sistemas Computacionales
programa PNPC
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
rrasilva18@gmail.com

Resumen—En este artículo, se plasma el trabajo interdisciplinario de la Maestría en Sistemas Computacionales con apoyo del CONACyT, impartida en el Instituto Tecnológico de Acapulco. El artículo tiene por meta, presentar la implementación de los patrones de unidad de trabajo y repositorio en la capa de acceso a datos, permitiendo agilizar y facilitar la manipulación de los datos respecto de la persistencia de datos a la capa de negocio. Se hace mención a que este artículo forma parte de una serie de trabajos siendo el tercero de estos, el cual da seguimiento a un artículo previo titulado: Diseño de una herramienta informática basada en la metodología Scrum para la gestión del desarrollo de software, escrito por los mismos autores y que fue presentado y publicado en el congreso de Academia Journals Puebla 2019.

Palabras clave—Patrón de diseño, ASP.NET Core, MVC, Repositorio y Unidad de trabajo.

I. INTRODUCCIÓN

Actualmente existen una gran variedad de tecnologías para desarrollar sistemas informáticos que implementan patrones de diseño que permiten agilizar y facilitar el proceso de desarrollo, lo cual es debido a la gran demanda que existe en el mercado, que las empresas requieren de algún tipo de software que demande la menor cantidad de tiempo en su construcción. Cuando se habla de escalar sistemas informáticos, el patrón MVC (Modelo-Vista-Controlador) es ideal para este tipo de escenarios, porque resulta más fácil codificar, depurar y probar algo (ya sea del modelo, vista o controlador) que tenga un solo trabajo ya que resulta complejo actualizar, probar y depurar código que tenga dependencias repartidas entre dos o más de estas tres áreas. Por ejemplo, la lógica de la interfaz de usuario tiende a cambiar con mayor frecuencia que la lógica de

negocios. Si el código de presentación y la lógica de negocios se combinan en un solo objeto, un objeto que contenga lógica de negocios deberá modificarse cada vez que cambie la interfaz de usuario [6]. ASP.NET Core MVC es un marco de desarrollo de aplicaciones web de Microsoft que combina la eficiencia y el orden de la arquitectura de MVC, ideas y técnicas de desarrollo ágil y las mejores partes de la plataforma .NET [1]. Este marco de desarrollo es de código abierto y multiplataforma permitiendo desplegar aplicaciones en diferentes sistemas operativos. Un grupo de especialistas conformado por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides han catalogado un conjunto de patrones. Actualmente, su catálogo se considera una de las fuentes de información más populares sobre patrones de diseño. Dado que los patrones fueron documentados por los cuatro autores, los patrones se denominan patrones de diseño Gang of Four, o GoF. El catálogo GoF incluye 23 patrones de diseño. Los autores han colocado estos 23 patrones en tres categorías; patrones de creación, estructurales y de comportamiento [4]. ASP.NET Core MVC permite utilizar inyección de dependencia, la cual es una técnica que ayuda a crear aplicaciones flexibles y simplifica las pruebas unitarias [1]. Esta técnica permite inyectar servicios antes de que muestre una vista al usuario, una forma de inyectar servicios con esta técnica es a través del patrón *singleton* el cual dicta que solamente se puede crear una instancia de una clase, entonces cuando se crean aplicaciones web con ASP.NET Core MVC que accedan a una base de datos, es obligado crear un contexto (clase) donde se establezca una cadena de conexión a la base de datos. Al inyectar un servicio del contexto a través del patrón *singleton*, se permite mejorar el rendimiento de una aplicación web debido a que, si múltiples usuarios realizan transacciones,

estos podrán acceder a la base de datos a través de un solo objeto del contexto. El patrón *singleton* pertenece a la categoría creacional del catálogo GoF.

A. Patrones de arquitectura de aplicaciones empresariales

Existe otro catálogo de patrones compilado por Martin Fowler. Este catálogo es llamado Patrones de Arquitectura de Aplicaciones Empresariales (en inglés Patterns of Enterprise Application Architecture ó P of EAA). Los patrones de ambos catálogos (GoF y P of EAA) se usan al construir sistemas de software. Sin embargo, los P de EAA están orientados a las aplicaciones empresariales [4]. Una aplicación empresarial es una colección de componentes que proporciona una funcionalidad empresarial que se puede utilizar internamente, externamente o con otras aplicaciones empresariales. Por ejemplo, gestión de pedidos, de inventarios o facturación [2]. Estas aplicaciones suelen ser complejas, altamente escalables y distribuidas. Los patrones del catálogo P of EAA están colocados en 10 categorías: Patrones de lógica de dominio, arquitectónicos de orígenes de datos, de comportamiento relacional de objetos, estructurales relacional de objetos, de mapeo de metadatos de objetos relacionales, de presentación web, de distribución, de concurrencia fuera de línea, de estado de sesión y de base. Las aplicaciones modernas están divididas en tres capas conceptuales: Acceso a datos, lógica de negocio y presentación o interfaz de usuario.

Los patrones unidad de trabajo y repositorio que pertenecen al catálogo P of EAA, permiten obtener la mejor forma de realizar transacciones con la base de datos. A continuación, se explican estos patrones.

B. Patrón repositorio

Emplear una capa de acceso a datos para realizar las operaciones Crear, Leer, Actualizar y Eliminar (en inglés Create, Read, Update y Delete, CRUD) es muy común en las aplicaciones del mundo real. Un componente de acceso a datos aísla los detalles de cómo las operaciones CRUD están tomando lugar en un sistema informático, por ejemplo, cuando se necesita realizar un conjunto de consultas de la misma base de datos desde varios lugares. Aunque un componente de acceso a datos encapsule las operaciones básicas de CRUD, no ofrecerá ayuda para evitar la duplicación de estas operaciones. Por lo tanto, se viene realizando el mismo código en varios lugares de una aplicación. En tales casos, se puede introducir una capa entre las clases de dominio y la capa de acceso a datos. Esta capa se encargará de las operaciones de encapsulación que se pueden reutilizar una y otra vez. Esta capa viene en forma de patrón repositorio. El patrón repositorio media entre la capa de acceso a datos y el resto del sistema. Además, lo hace proporcionando acceso de colección a los datos subyacentes. Una vez que se implementa el patrón de repositorio, el código de la capa de negocio no invocará el componente de acceso a datos directamente. En su lugar, invocará el repositorio para hacer el trabajo. El patrón repositorio ofrece una interfaz de recopilación al proporcionar métodos para agregar, modificar, eliminar y recuperar objetos de dominio.

C. Patrón unidad de trabajo

Una operación comercial involucra múltiples pasos mientras es requerida. La operación puede ser tratada como un solo lote o unidad. Se puede recurrir a las transacciones de la base de datos y realizar la tarea de la siguiente manera:

- Comenzar una transacción en la base de datos.
- Realizar todos los pasos de la operación uno por uno contra la base de datos.
- Comprometer o revertir la transacción.

Aunque este procedimiento parece sencillo, existe un problema: Se están realizando pasos individuales de la operación directamente en la base de datos. Entonces, si una operación involucra n pasos, está creado n operaciones de escritura en la base de datos. Una cantidad tan grande de pequeñas operaciones en la base de datos puede afectar el rendimiento general del sistema. Una mejor forma es capturar pasos individuales y luego enviarlos al motor de la base de datos de una sola vez y luego ejecutarlos en una única transacción, esa es la idea detrás del patrón unidad de trabajo. El patrón de unidad de trabajo realiza un seguimiento de la transacción comercial que se supone altera la base de datos de alguna manera. Una vez que finaliza la transacción comercial, los pasos se reproducen en la base de datos en una transacción para que la base de datos refleje los cambios deseados. Por lo tanto, el patrón de unidad de trabajo rastrea una transacción comercial y la traduce en una transacción de base de datos, en la que los pasos se ejecutan colectivamente como una sola unidad [4].

II. OBJETIVO GENERAL

El objetivo principal que se pretende lograr con la presente propuesta, consiste en desarrollar una herramienta informática que permita la gestión de proyectos de desarrollo de software que basan su construcción, en el empleo de la metodología Scrum, para ello, es importante la implementación de los patrones unidad de trabajo y repositorio en la capa de acceso, haciendo uso del lenguaje de programación C# con el marco de desarrollo ASP.NET Core MVC.

III. IMPLEMENTACIÓN DE LOS PATRONES REPOSITORIO Y UNIDAD DE TRABAJO

A partir de esta sección, se presenta la implementación de los patrones repositorio y unidad de trabajo en la capa de acceso a datos de la herramienta, esta implementación es soportada por el diagrama de clases propuesto en el segundo artículo de la serie [3], en el cual se representó dichos patrones.

A. Repositorio

En este apartado se presenta algunos métodos implementados que proporciona el patrón repositorio al momento de utilizarlo. Las siguientes líneas de código implementan la clase genérica *RepositorioGenerico* la cual hereda la interfaz *IRepositorioGenerico* indicando a través de la palabra reservada *where* que *T* es una clase.

```
public class RepositorioGenerico<T> :
    IRepositoryGenerico<T> where T : class
{
    internal PiScrumDbContext db = null;

    public
    RepositorioGenerico(PiScrumDbContext db)
    {
        this.db = db;
    }
}
```

Dentro de la clase *RepositorioGenerico* se crea un campo el cual es de tipo *PiScrumDbContext* que es una clase que representa el contexto para realizar operaciones en la base de datos. A continuación, se muestran las líneas de código que implementan el método para agregar (*AgregarAsin*) un registro en la base de datos.

```
public virtual async Task<T>
    AgregarAsin(T entity)
{
    db.Set<T>().Add(entity);
    return entity;
}
```

El método *AgregarAsin* recibe como parámetro *entity* de tipo *T*, el cual es proporcionado como argumento para el método *Add* donde a través del método *Set* se obtiene un *DbSet* (que representa la colección de todas las entidades en el contexto) de la entidad para que se pueda guardar la instancia de la entidad (clase). Los métodos *Add* y *Set* son proporcionados por *Entity Framework* el cual es un mapeador de objetos relacionales que permite a los desarrolladores de .NET trabajar con una base de datos utilizando objetos .NET. Las siguientes líneas de código permiten actualizar datos de un registro en la base de datos.

```
public virtual async Task<T>
    ActualizarAsin(T entity, object key)
{
    if (entity == null)
        return null;
    T exist = await
    db.Set<T>().FindAsync(key);
    if (exist != null) {
        db.Entry(exist).CurrentValues.SetVal
        ues(entity);
    }
    return exist;
}
```

El método *ActualizarAsin* presentado, recibe como parámetros *entity* y *key* donde *entity* es de tipo *T* y *key* de tipo *object*. En este método primero se realiza una consulta (*FindAsync*) en la entidad almacenada en *entity* donde se hace un filtrado utilizando el parámetro *key*. La variable *exist* almacena la entidad retornada por la consulta, entonces la segunda condición se cumple y se hace la actualización a través del método *Entry* proporcionando como argumento la entidad retornada a la cual se va actualizar con la información almacenada en *entity* por el método *SetValues*. Los métodos

FindAsync, *Entry* y *SetValues* también son proporcionados por *Entity Framework*. A continuación, en las siguientes líneas se presenta el método para eliminar un registro de la base de datos.

```
public virtual async Task EliminarAsin(T
    entity)
{
    db.Set<T>().Remove(entity);
}
```

El método *EliminarAsin* presentado, recibe como parámetro *entity* de tipo *T* donde almacena la información del registro a eliminar de una entidad específica. Para eliminar un registro simplemente se utiliza el método *Remove* donde se pasa como argumento *entity* y este obtiene un *DbSet* por el método *Set* para realizar la transacción en el contexto.

Como pueden observar en la firma de los métodos presentados tienen dos palabras reservadas *async* y *Task*, esto hace que los métodos sean asíncronos, permitiendo que se puedan hacer múltiples transacciones en el contexto con una sola instancia de este.

B. Unidad de trabajo

En las siguientes líneas de código se muestra la parte esencial de la implementación del patrón unidad de trabajo.

```
public class UnidadDeTrabajo :
    IUnidadDeTrabajo
{
    private readonly PiScrumDbContext db;

    public UnidadDeTrabajo(PiScrumDbContext
        db)
    {
        this.db = db;
    }

    private IRepositoryGenerico<Proyectos>
        Proyectos;

    private IRepositoryGenerico<
        HistoriasUsuario>HistoriasUsuario;

    private IRepositoryGenerico<Sprint>
        Sprint;

    private IRepositoryGenerico<Tareas>
        Tareas;

    private IRepositoryGenerico<
        SeguimientoSprint>
        SeguimientoSprint;

    public IRepositoryGenerico<Proyectos>
        ProyectosRepositorio
    {
        get
        {
```

```

        return Proyectos = Proyectos ?? new
        RepositorioGenerico<Proyectos>(db);
    }
}

public IRepositoryGenerico<
    HistoriasUsuario>
    HistoriasUsuarioRepositorio
{
    get
    {
        return HistoriasUsuario =
        HistoriasUsuario ?? new
        RepositorioGenerico<HistoriasUsuari
        o>(db);
    }
}

public IRepositoryGenerico<Sprint>
    SprintRepositorio
{
    get
    {
        return Sprint = Sprint ?? new
        RepositorioGenerico<Sprint>(db);
    }
}

public IRepositoryGenerico<Tareas>
    TareasRepositorio
{
    get
    {
        return Tareas = Tareas ?? new
        RepositorioGenerico<Tareas>(db);
    }
}

public IRepositoryGenerico<
    SeguimientoSprint>
    SeguimientoSprintRepositorio
{
    get
    {
        return SeguimientoSprint =
        SeguimientoSprint ?? new
        RepositorioGenerico<SeguimientoSpri
        nt>(db);
    }
}

public async Task<int> SaveAsync()
{
    return await db.SaveChangesAsync();
}

```

La clase *UnidadDeTrabajo* tiene el propósito de asegurar que, usando varios repositorios, compartan un mismo contexto de base de datos (variable *db*). De tal forma que, cuando se complete una unidad de trabajo, se pueda invocar al método *SaveChanges* en esa instancia del contexto, asegurando que todos los cambios relacionados se coordinen. Por ello, la clase necesita el método *SaveAsync* y una propiedad para cada repositorio. Las propiedades del repositorio retornan una instancia de repositorio de la cual se han creado instancias con la misma instancia de contexto de base de datos que las demás instancias de repositorio. En el código anterior, se crean variables de clase para el contexto de la base de datos y para cada repositorio (variable *SeguimientoSprint*). En el caso de la variable *db*, se crea una instancia de un nuevo contexto.

Las propiedades del repositorio comprueban si el repositorio existe. De otra forma, se crea una nueva instancia de este, pasando la instancia de contexto (*db*). Lo cual da como resultado, cada uno de los repositorios que comparten la misma instancia de contexto.

IV. RESULTADOS

Este apartado se presenta el uso de los patrones repositorio y unidad de trabajo en la lógica de negocio. El siguiente código es un fragmento del método que realiza el seguimiento del sprint de un proyecto en la herramienta.

```

await
unidadDeTrabajo.SeguimientoSprintReposito
rio.AgregarAsin (Modelo);
//guardando cambios en las entidades de
//tareas e historias de usuario
if (ModeloT.EstimacionEsfuerzoRestante ==
0)
    ModeloT.Estatus = "FINALIZADO";
else
    ModeloT.Estatus = "PROCESO";

if (ModeloH.EstimacionEsfuerzoRestante ==
0)
    ModeloH.Estatus = "FINALIZADO";
else
    ModeloH.Estatus = "PROCESO";

await
unidadDeTrabajo.TareasRepositorio.Actuali
zarAsin (ModeloT,ModeloT.IdTarea);

await
unidadDeTrabajo.HistoriasUsuarioRepositor
io.ActualizarAsin (ModeloH,ModeloH.IdHisto
riaUsuario);

await unidadDeTrabajo.SaveAsync();

```

Del código anterior, las líneas con fondo gris, son donde se implementa el patrón de repositorio y unidad de trabajo. La primera línea en negrita se utiliza el método *Agregar* a través de la propiedad *SeguimientoSprintRepositorio* del repositorio

pertenece a la clase *UnidadDeTrabajo* que implementa el patrón unidad de trabajo. El método *Agregar* requiere un modelo el cual es una clase de tipo *SeguimientoSprint*. De esta misma forma se utiliza para actualizar una tarea y una historia de usuario. Cuando se realizan las transacciones comerciales agregar y actualizar se guardan cambios en la base de datos con las instancias de las entidades *Seguimiento sprint*, *Historias Usuario* y *Tareas*. Estos cambios se hacen con el método *SaveAsync* a través del patrón *unidad de trabajo* como se muestra en el código presentado anteriormente. De esta misma forma se utiliza para actualizar una tarea y una historia de usuario. Cuando se realizan las transacciones comerciales agregar y actualizar se guardan cambios en la base de datos con las instancias de las entidades *Seguimiento sprint*, *Historias Usuario* y *Tareas*. Estos cambios se hacen con el método *SaveAsync* a través del patrón *unidad de trabajo* como se muestra en el código presentado anteriormente.

Cuando se ejecuta el método *AgregarAsin*, **Entity framework** traduce esta instrucción en una instrucción SQL para que el gestor de base de datos pueda interpretarlo y realizar la transacción. En la figura 1 se observa la traducción del método *AgregarAsin* en SQL.

```
INSERT INTO "SeguimientoSprint" ("IdSeguimientoSprint", "FechaRegistro", "IdSprint", "IdTarea",
VALUES (@p2, @p3, @p4, @p5, @p6);
```

Fig 1. Traducción del método *AgregarAsin* a SQL.

De la misma forma, cuando se ejecuta el método *ActualizarAsin* este se traduce a SQL como se observa en la figura 2.

```
UPDATE "Tareas" SET "EstimacionEsfuerzoRestante" = @p7
WHERE "IdTarea" = @p8;
UPDATE "HistoriasUsuario" SET "Estatus" = @p0
WHERE "IdHistoriaUsuario" = @p1;
```

Fig 2. Traducción del método *ActualizarAsin* a SQL.

V. COMENTARIOS FINALES

A. Conclusiones

En este artículo, se presentó la implementación de los patrones repositorio y unidad de trabajo en la capa de acceso permitiendo mejorar el rendimiento al momento de realizar transacciones en la base de datos, también influye el hecho de que los métodos contenidos en el patrón son asíncronos, resolviendo así, el problema de concurrencia. La forma de tratar una transacción comercial el patrón unidad de trabajo, es muy similar al mecanismo de las instrucciones *commit* y *rollback* de SQL, es decir, si en una transacción con *n* cambios falla uno ellos entonces ninguno se refleja en la base de datos esto asegura la integridad de la base de datos. La implementación de estos patrones en la capa de acceso no afecta al momento de cambiar de gestor de base de datos ya que estos patrones como se expuso,

el patrón repositorio tiene la función de crear una capa de abstracción entre la capa de acceso a datos y la lógica de negocio y el patrón unidad de trabajo permite manejar transacciones durante la manipulación de datos utilizando los métodos implementados en el patrón repositorio, es decir, el patrón unidad de trabajo solamente necesita de un contexto sin importar el origen.

B. Trabajos a futuro

Este artículo forma parte de una serie de trabajos que reflejan el progreso del proyecto de tesis de maestría, donde el trabajo a futuro claramente debe abordar la conclusión de la herramienta donde se expondrá, el desarrollo de la aplicación propuesta, haciendo uso de la implementación de los patrones repositorio y unidad de trabajo presentada en este trabajo y haciendo uso de las herramientas de desarrollo de software, implementando la arquitectura propuesta en el primer artículo [5].

REFERENCIAS

- [1] A. Freeman, Pro ASP.NET Core MVC 2, Londres: Apress, 2017.
- [2] IBM, «Aplicaciones empresariales.» IBM Corp., 2017. [En línea]. Available: https://www.ibm.com/support/knowledgecenter/es/SSPLFC_7.3.0/com.ibm.taddm.doc_7.3/UserGuide/c_cmdb_business_apps.html. [Último acceso: 22 Febrero 2020].
- [3] J. F. Gazga Portillo, J. R. López Morales, J. M. Hernández Bravo y A. D. de Jesús Islao, «Diseño de una herramienta informática basada en la metodología Scrum para la gestión del desarrollo de software,» *Academia Journals*, pp. 789-794, 2019.
- [4] B. Joshi, Beginning SOLID Principles and Design Patterns for ASP.NET Developers, Thane: Apress, 2016.
- [5] J. R. López Morales, J. F. Gazga Portillo, J. M. Hernández Bravo y A. D. de Jesús Islao, «Propuesta de una herramienta basada en la metodología Scrum para la gestión del desarrollo de software,» vol. 11, n° 2, 2019.
- [6] S. Smith, «Información general de ASP.NET Core MVC,» Microsoft, 28 01 2020. [En línea]. Available: <https://docs.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-3.1>. [Último acceso: 19 02 2020].

Diseño de una aplicación web para el control del historial clínico, implementando el estándar HL7

Ing. Luis Ángel Castro Silvestre
Estudiante de la Maestría en
Sistemas Computacionales en un
Programa PNP
Tecnológico Nacional de México
/ IT de Acapulco
Acapulco, Guerrero, México
lacsilvestre18@gmail.com

M.T.I. Juan Miguel Hernández
Bravo
Docente de la Maestría en
Sistemas Computacionales.
Tecnológico Nacional de México
/ IT de Acapulco
Acapulco, Guerrero, México.
jmhernan@yahoo.com

M.T.I. Jorge Carranza Gómez
Docente de la Maestría en
Sistemas Computacionales.
Tecnológico Nacional de México
/ IT de Acapulco
Acapulco, Guerrero, México.
jcarranzamx@gmail.com

Dr. José Antonio Montero
Valverde
Docente de la Maestría en
Sistemas Computacionales.
Tecnológico Nacional de México
/ IT de Acapulco
Acapulco, Guerrero, México.
jamontero@infinitum.net

Ing. Antonio Alejandro Sánchez Zapata
Estudiante de la Maestría en
Sistemas Computacionales en un
Programa PNP.
Tecnológico Nacional de México
/ IT de Acapulco
Acapulco, Guerrero, México.
aaszapata18@gmail.com

Resumen— El presente artículo tiene como finalidad realizar un análisis sobre el diseño de una herramienta que facilite el manejo de la información confidencial contenida en las historias clínicas y permita al personal médico tener un control total sobre éstas. Actualmente en los hospitales nacionales de nuestro país se archivan las historias clínicas de los pacientes en carpetas físicas donde uno encuentra todos los datos, antecedentes y pruebas de laboratorios que se le puedan realizar a una persona, esta forma de archivar las historias clínicas de los pacientes limita su atención. Una necesidad sentida del sistema de salud de nuestro país es la Historia Clínica Electrónica (HCE), una información que es vital tenerla de manera expedita y poder acceder a ella sin restricciones de ninguna índole a excepción de la protección de datos del usuario. Antes las circunstancias antes mencionadas, el Laboratorio Estatal de Salud Pública del Estado de Guerrero en sus esfuerzos por el mejoramiento continuo, considero de gran importancia el desarrollo de una aplicación web para la administración y manejo del HCE, utilizando el estándar HL7 (HL7, 2020) para la interoperabilidad.

Palabras clave— LESP, HCE, HL7, ECELESP, OOHDM.

I. INTRODUCCIÓN

En los últimos años se han presenciados a cambios radicales en el uso de las Tecnologías de la Información y la Comunicación (TIC). Como es lógico la medicina no ha escapado a esta revolución tecnológica. Se ha contemplado un cambio trascendental en la forma de generar, consultar y comunicar la información clínica (Sánchez Zapata, 2019). Ya es posible pensar que han desaparecido muchas de las barreras que impedían una comunicación a distancia, simultánea y en cualquier momento con otros profesionales asistenciales.

Además, las TIC suponen importantes avances a nivel de acceso y a nivel de incorporar herramientas de soporte a la decisión.

En este sentido la HCE es una herramienta que favorece la calidad, la seguridad y la continuidad asistencial (Otero, 2014). Permite además tener un control sobre las acciones realizadas. Sin embargo, la complejidad del trabajo médico, la diversidad de los usuarios, personal clínico, y el gran número de sistemas de información implicados hacen que se trate de una tarea difícil. Para conseguir su implementación en un centro sanitario es necesario un activo compromiso de todos los usuarios implicados (Ramos, 2010).

La implementación de la HCE es una decisión estratégica dentro del Laboratorio Estatal de Salud Pública (LESP) que pretende mejorar la efectividad y la eficiencia. También obedece a una realidad marcada por las expectativas de nuestros pacientes y por la aparición de nuevas técnicas médicas. Poder controlar los costos, optimizar los procesos y reasignar los recursos son retos permanentes de cualquier Dirección. El LESP muestra importantes avances en la incorporación de tecnologías de la información para conseguir estos objetivos y la HCE es uno de ellos.

II. OBJETIVO GENERAL DEL PROYECTO

Desarrollar una aplicación web para el Laboratorio Estatal de Salud Pública del Estado de Guerrero, donde a través de dicha aplicación el personal médico podrá registrar y administrar las historias clínicas de los pacientes, haciendo uso del estándar HL7 (HL7, 2020) para la interoperabilidad entre estas.

III. ALCANCES DE LA SOLUCIÓN

Con el desarrollo de este proyecto se busca optimizar e integrar los diferentes recursos de información en el LESP, mediante el uso de tecnologías de conectividad e interoperabilidad que permitan la creación de una red de información. Con esta aplicación, el personal médico consultará, y allí encontrará diagnósticos, tratamientos, medicación y recomendaciones de todos los pacientes afiliados al sistema, de modo que actuará con mayor celeridad. La información se manejará con absoluta privacidad y sólo podrá ser consultada por el personal médico debidamente autorizado.

En el desarrollo de este proyecto, el elemento central lo constituye un repositorio de historias clínicas, que se encarga de regular el acceso a los registros médicos, así como gestionar, almacenar y proporcionar los mecanismos de seguridad de la información de los pacientes que hacen parte del sistema. Los diversos esquemas desarrollados en el proyecto son un excelente referente en cuanto al manejo que se le debe dar al tema de seguridad, sobre todo teniendo en cuenta lo delicado e

importante que es la información almacenada en las historias clínicas.

IV. DISEÑO UTILIZANDO LA METODOLOGÍA OOHDM

El modelo *Object Oriented Hypermedia Desing Methodology* (OOHDM) es una extensión de *Hypertext Desing Model* (HDM) orientada a objetos, hoy en día OOHDM se está convirtiendo en una de las metodologías que más utilizan los diseñadores de aplicaciones web, debido a que en comparación con otras metodologías OOHDM genera mejores resultados (Teixeira, 2018). Haciendo uso de la metodología OOHDM y basándome en las diferentes etapas que propone, se diseñaron los siguientes diagramas:

A. Diagrama de clases

En la fase de contenido de la metodología OOHDM aplicado en el diseño de la aplicación web para el HCE se muestra el siguiente diagrama de clases (Fig. 1), que representa la información con los datos y atributos.

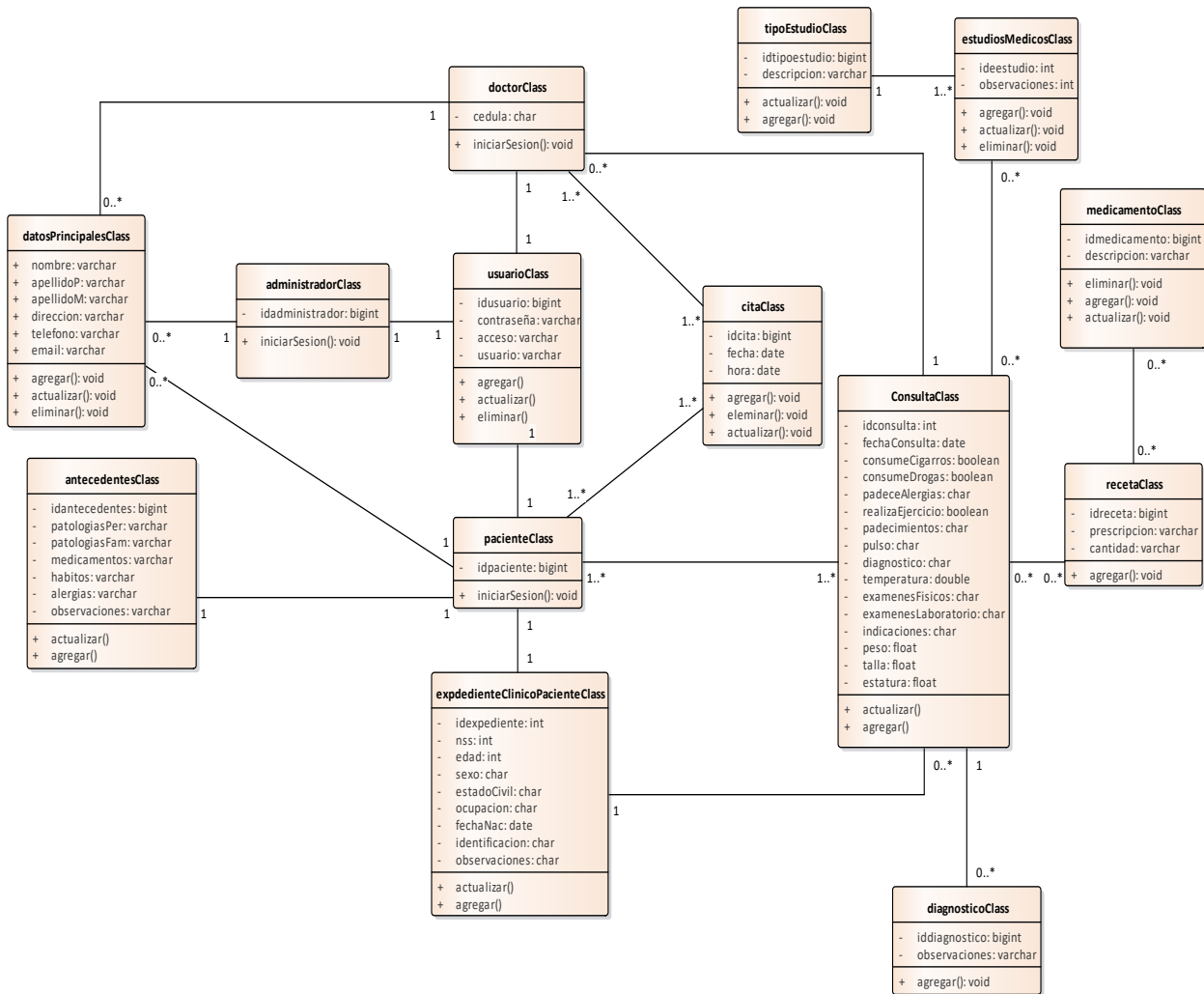


Fig. 1. Diagrama de clases (Elaboración propia).

A continuación, se describen los componentes de la Figura 1:

- **administradorClass:** En esta clase tipo formulario se obtienen los datos personales que se heredan de la clase datosPrincipalesClass.
- **antecedentesClass:** Esta clase se encarga de registrar los antecedentes médicos de los pacientes, así como alergias u otra información relevante para el llenado de su historia clínica.
- **citaClass:** Clase tipo formulario encargada de agendar citas a los pacientes, el personal clínico del laboratorio será el responsable de realizar cada una de las citas.
- **consultaClass:** En esta clase se lleva a cabo el registro de las consultas de cada uno de los pacientes que se atienden, el personal clínico del laboratorio a través de esta clase agregara los datos recabados al historial clínico del paciente.
- **datosPrincipalesClass:** Esta clase contiene los campos para la captura de los datos personales, a partir de esta clase que hereda dichos campos se hace el registro de los datos personales del administrador, pacientes y doctores.
- **diagnosticoClass:** La función principal de esta clase es registrar el diagnostico que el personal clínico realice a un paciente a través de una consulta previa.
- **doctorClass:** En esta clase se registran los datos del personal clínico que ara uso de la aplicación.
- **estudiosMedicosClass:** En esta clase se lleva el registro de los diferentes estudios médicos que el personal clínico le halla solicitado al paciente, cabe destacar que esta clase hereda datos de la clase consultaClass los cuales son necesarios para que dicho estudio se realice.
- **expedienteClinicoPacienteClass:** Esta clase es la principal encargada de almacenar toda la información clínica del paciente, el personal médico se encarga de enriquecerla a través de la información de consultas y estudios médicos que requieran del paciente.
- **medicamentosClass:** Esta clase funciona como conexión para la clase recetaClass.
- **pacienteClass:** En esta clase si registran los datos personales de los pacientes, hereda en su mayoría los campos de la clase datosPrincipales.
- **recetaClass:** Clase encargada de llevar el registro que el personal clínico recete a un paciente.

- **tipoEstudioClass:** Esta clase funciona como conexión entre la clase estudiosMedicosClass, y su principal tarea es asignar que tipo de estudio se realizara el paciente.
- **usuarioClass:** En esta clase se lleva el control de usuarios que harán uso de la aplicación, a través de un formulario se obtiene los datos de dichos usuarios.

B. Modelado de proceso.

La **Fig. 2** representa el diagrama de modelado de negocios, la cual permite tener una mejor comprensión de los procesos que se llevan a cabo. En la Fig. 2 se pueden apreciar tres carriles, los cuales dividen las áreas involucradas para el expediente clínico. A continuación, se describen los procesos:

- **Solicita atención medica:** Es el primer proceso, inicia cuando un paciente solicita los servicios médicos.
- **Solicitar datos:** El personal clínico solicita los datos personales al paciente. En este punto se determina si el paciente ya se encuentra dado de alta en el sistema, si es así continua al siguiente proceso, de lo contrario se registra al paciente. Dentro de este proceso se genera el “Carnet” el cual es único para cada paciente.
- **Agendar cita:** En este proceso se genera una cita, ya sea para ver la evolución de un tratamiento o simplemente una cita general. Esté proceso genera un documento “Cita” el cual contiene los datos de correspondientes de la cita generada.
- **Evaluación:** El personal clínico realiza una evaluación inicial del paciente y hace anotaciones (observa dolencias). Dentro de este proceso se genera el “Registro clínico” indispensable para la creación del expediente médico.
- **Urgencia:** De acuerdo con el diagnóstico, tras la evaluación inicial, se determina si el paciente recibe atención inmediata de acuerdo a la gravedad presentada.
- **Atención paciente:** El personal clínico da atención al paciente, elabora historia clínica electrónica completa y diagnostica tras las pruebas físicas y observaciones que realizó. Se generan dos documentos, uno de ellos el diagnostico que realiza el personal clínico y el otro el expediente del paciente.
- **Registrar datos:** Una vez efectuada la consulta y generado el diagnóstico y el expediente del paciente, se agregan los datos y las observaciones del personal clínico al HCE.
- **Receta médica:** En caso de que el paciente requiera tratamiento, el personal clínico genera la “receta” con

las prescripciones de acuerdo a la consulta. Dentro de este proceso si el médico considera necesario que el paciente se realice estudios de laboratorio se genera la "Orden de laboratorio" con la lista de estudios a realizar.

- **Realizar pruebas de laboratorio:** En este proceso se lleva a cabo las pruebas de laboratorio, generando los resultados de dichos estudios.
- **Guardar información:** El personal clínico al finalizar la consulta procede a guardar la información generada en la HCE.

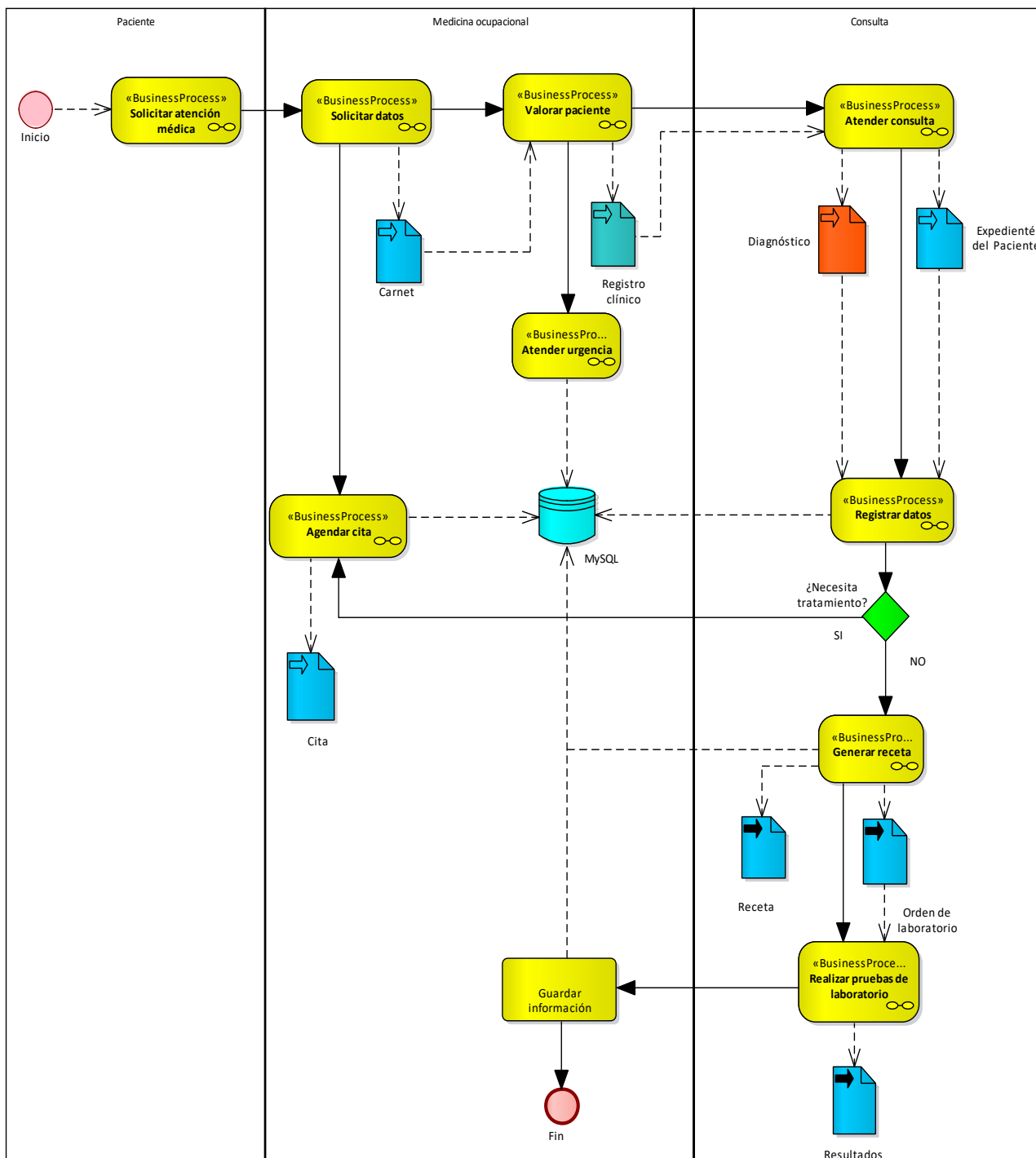


Fig. 2. Modelado de negocio (Elaboración propia).

C. Diagrama Entidad-Asociación

En la Fig. 3 se muestran las tablas que se podrán utilizar para almacenar la información y que estarán disponibles para la aplicación. Este diagrama será el modelo para crear la base de

datos de la aplicación, de llenarse de manera correcta será indispensable para el correcto funcionamiento del proyecto.

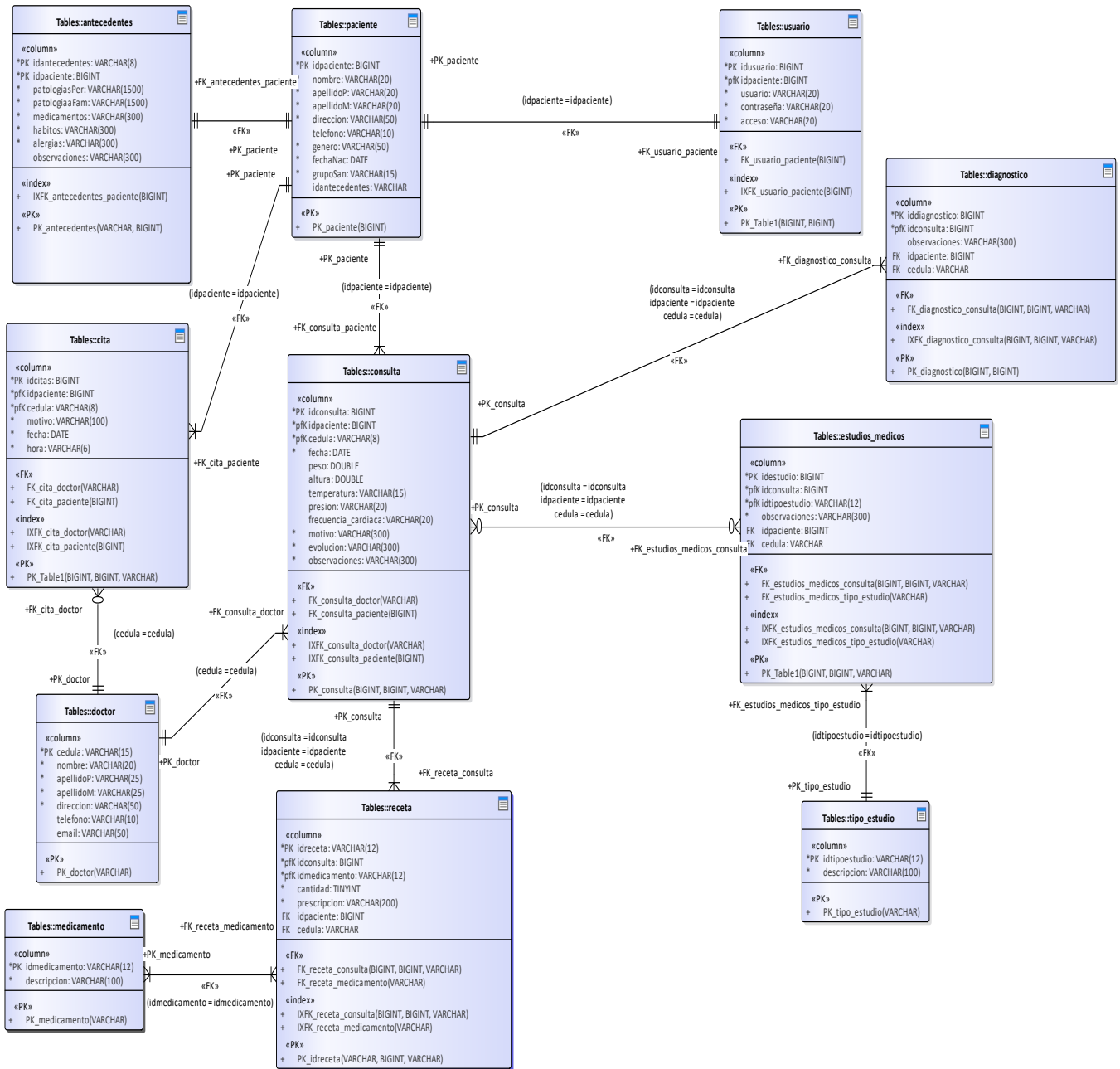


Fig. 3. Diagrama Entidad-Asociación (Elaboración propia).

D. Diseño navegacional.

En este mapa de navegación contextual se muestran las rutas de navegación que podrá realizar el usuario por la aplicación web, primeramente, se deberá autenticar para tener acceso al menú principal y acceder a los diferentes módulos del sistema.

Los submenús son los siguientes: Submenú Doctores, Submenú Expediente, Submenú Pacientes, Submenú Reportes y Submenú Usuarios.

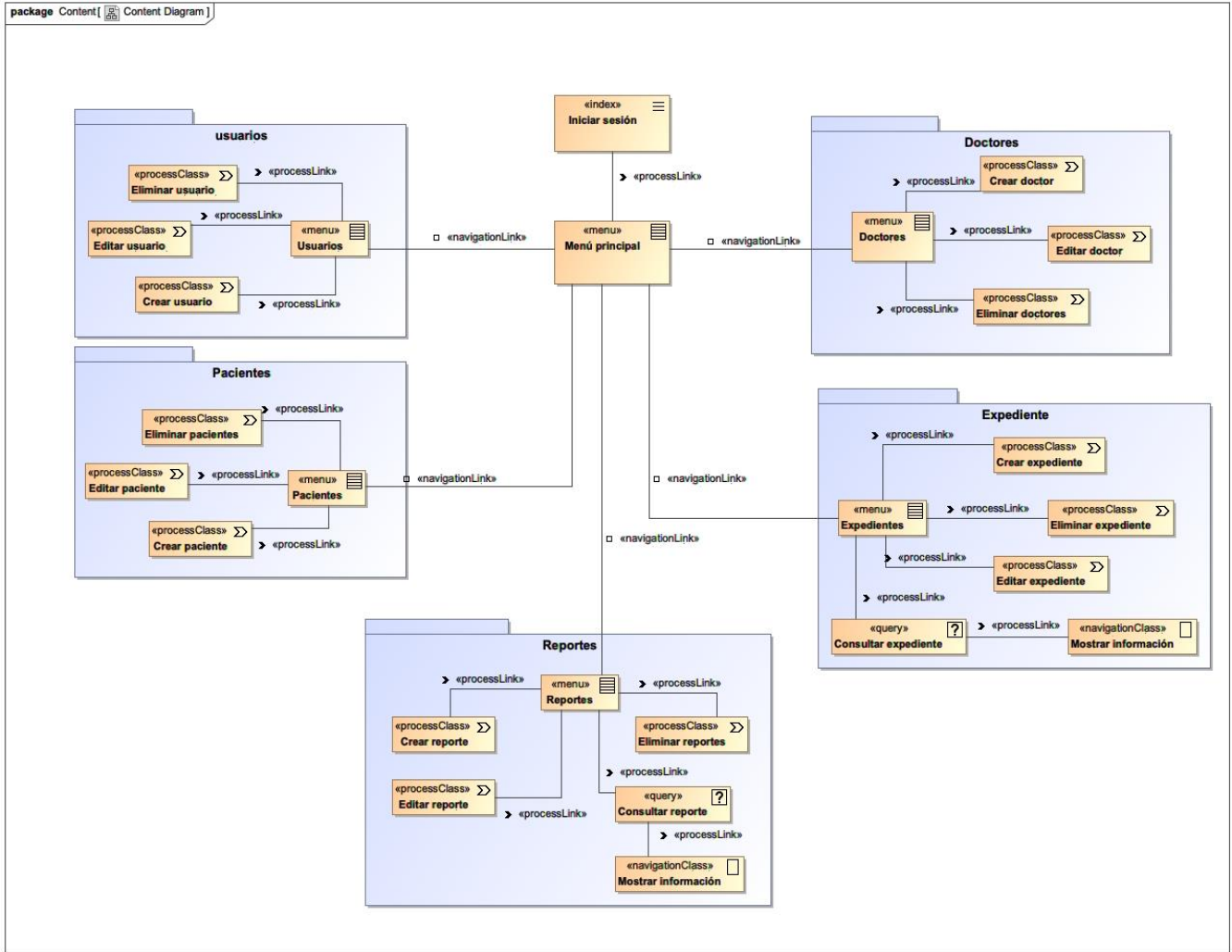


Fig. 4. Diseño navegacional (Elaboración propia).

E. Diagrama de despliegue.

En este diagrama se muestra la distribución de las partes del sistema Web, separando los componentes de cada nodo como se muestra en la Fig. 5.

Descripción:

- **PC:** Computadora de escritorio en la cual se accederá a la aplicación por medio de un navegador web, ya sea Chrome, Firefox o Safari.
- **Dispositivos móviles:** Es un equipo portátil de fácil acceso como podría ser un teléfono celular inteligente o una tableta digital por el cual a través de un navegador web pueda manipularse.
- **Servidor web:** Este servidor contiene los recursos necesarios para el funcionamiento de la aplicación, cuenta con el motor PHP, Apache y la interfaz para la conexión de la BD.

- **Servidor BD:** Cuenta con el sistema gestor de base de datos, que en este caso es MySQL.

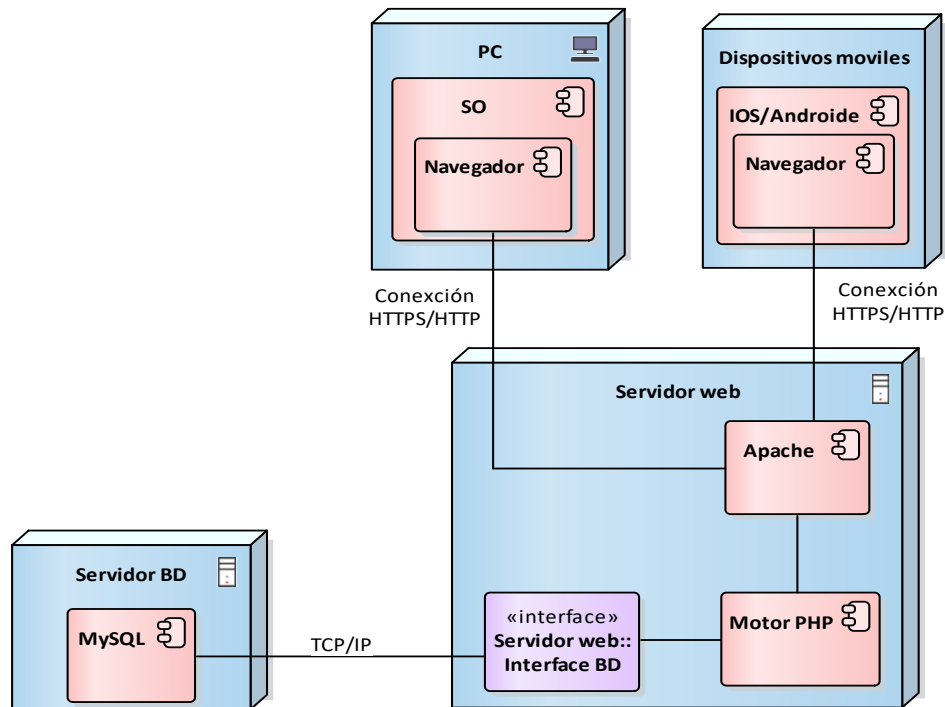


Fig. 5. Diagrama de despliegue (Elaboración propia).

V. ESTÁNDAR HL7

El estándar HL7 está enfocado hacia los desarrolladores de software y fabricantes de equipos médicos, con el objetivo de unificar la forma en que se transmite, intercambia o almacena la información presente en las unidades e instituciones médicas, según un formato común, acordado por todas las partes involucradas. Existen otros estándares dedicados al sector médico, cada uno con un dominio y enfoque muy bien definidos: farmacia, dispositivos médicos, imágenes médicas y seguros. En este sentido, HL7 se dedica al procesamiento, así como la gestión de datos administrativos y clínicos (Castro, 2019). El HL7 se centra en los siguientes campos:

- Gestión de paciente (admisión, altas y traslados).
- Consultas de recursos (habitaciones, camas, dispositivos, etc.) programación de pacientes.
- Programación de procedimientos médicos, resultados, ensayos clínicos.
- Administración financiera.
- Documentos médicos.
- Registros médicos.
- Tratamientos médicos.

Teniendo en cuenta la gran variedad de aplicaciones involucradas en el proceso del acto médico y el requisito de que tengan que intercambiar información entre ellas, es

notablemente que muchas de estas interfaces de comunicación se beneficiarían enormemente del uso de un enfoque estandarizado. Los estándares HL7 vienen específicamente para resolver este problema y aliviar la carga del paso de mensajes y el intercambio de datos entre varias aplicaciones al proporcionar una estructura muy precisa bajo la cual esto debe suceder (Health Level Seven International, 2013).

Una aplicación que usa el estándar HL7 envía a otra aplicación un mensaje de "tipo HL7" generado como resultado de algún evento médico que ocurre en la actividad actual: admitir paciente, transferir paciente, dar de alta paciente, etc. Si la aplicación que recibe los mensajes también cumple con las regulaciones de HL7, entonces es seguro que no faltará información, toda la información recibida se interpretará de la manera correcta y se emitirá una respuesta adecuada. Como consecuencia, el intercambio de información realizado es coherente y eficiente (Castro, 2019).

Los estándares HL7 definen una serie de mensajes que cubren todas las actividades específicas de las unidades médicas. Un mensaje HL7 se compone de segmentos, campos, componentes y subcomponentes, se caracteriza por el tipo de mensaje. Los tipos de mensajes están organizados en diferentes

dominios o campos (por ejemplo, admisión, descarga, transferencia, programación de ensayos clínicos, etc.).

A continuación, se muestra un ejemplo de codificación de una orden de laboratorio en el formato tradicional de HL7.

```
H|\^&|||LESP|||||T||20200224172957
P|1|1|5||CASTRO^SILVESTRE^LUIS
ANGEL||19890905|M||Gro|||||||2020022
4172957|||||1
O|6|1||^102|||||A|||||||O
O|6|1||^702|||||A|||||||O
O|6|1||^EGO|||||A|||||||O
L|6|N
```

VI. CONCLUSIONES

La aplicación permitirá al personal médico, contar con una herramienta de software para el control adecuado del historial clínico electrónico dentro del Laboratorio Estatal de Salud Pública, el diseño de esta aplicación se basa en la norma NOM-024-SSA3-2012, Sistemas de información de registro electrónico para la salud y la norma NOM-004-SSA3-2012, Del expediente clínico. Cumpliendo normativas reguladas por la Secretaría de Salud.

Los beneficios que el estándar HL7 aporta al sistema, son la posibilidad de acceder a los datos médicos y procesarlos de forma independiente en la solución particular elegida por cada institución. Por lo tanto, no tiene importancia quién es el proveedor para una determinada aplicación de software, siempre que cumpla con las especificaciones de los estándares, y no limita de ninguna manera las opciones existentes en la adquisición de otros sistemas.

Las aplicaciones de software utilizadas en los centros médicos deben ser modulares y escalables para satisfacer las necesidades de los beneficiarios. Para garantizar la correcta integración de varios módulos de software, es necesario utilizar los estándares HL7, que proporcionan un conjunto de reglas y algoritmos específicos para el campo médico.

REFERENCIAS

[1] Castro, L. A. (06 de 2019). Propuesta de una Aplicación Web para la administración y manejo del historial clínico electrónico (HCE) en el sector salud, utilizando el estándar HL7 para la interoperabilidad. Obtenido de <http://www.hl7.org>

[2] HL7. (02 de 2020). *Health Level Seven International - Homepage | HL7 Standard*. Obtenido de <http://www.hl7.org>

[3] Health Level Seven International. (2013). HL7 Implementation Guide for CDA Release 2: Patient Generated Document Header Template, Release 1.

[4] Otero, P. (2014). Beneficios y riesgos relacionados con el uso de la historia clínica electrónica. *Archivos Argentinos de Pediatría*, Vol. 109(No. 6), Pág. 476-477.

[5] Ramos, L. J. (2010). Archivo de historias clínicas digitalizado, una solución previa a las Historias Clínicas Electrónicas. (Sociedad Española de Documentación Médica). Vol. 18(No. 2), Pág. 4-10.

[6] Sánchez Zapata, A. A. (15 de Agosto de 2019). Implementación de una Aplicación Web para el Registro y Control de Expedientes Médicos de la Clínica Avanzada de Atención Primaria a la Salud (C.A.A.P.S): Caso "Área de Servicio Amigable (Módulo del Adolescente)". *Memorias del Congreso Internacional de Investigación Academia Journals Puebla 2019*. Puebla, Puebla, México.

[7] Teixeira, L. (2018). Desenvolvimento de aplicações hipermedia utilizando a metodologia OOHDM: a possibilidade de incluir os requisitos funcionais e não funcionais de um sistema. *University of Aveiro*.

Implementación del backend de una aplicación web para el mantenimiento de infraestructura y equipo en el ITSM

Ing. Rogelio Ramírez Silva
Estudiante de Maestría en
Sistemas Computacionales
programa PNPC
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
rrasilva18@gmail.com

M.C. Francisco Javier Gutiérrez
Mata
Docente de Maestría en Sistema
Computacionales
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
fcmata84@hotmail.com

M.I.D.S. Alma Delia De Jesús
Islao
Docente de Maestría en Sistema
Computacionales
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
alma.islao.ita@gmail.com

M.C. Eloy Cadena Mendoza
Docente de Maestría en Sistema
Computacionales
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
eloy_cadena@yahoo.com

Ing. José Raúl López Morales
Estudiante de Maestría en
Sistemas Computacionales
programa PNPC
Tecnológico Nacional de México
/IT de Acapulco
Acapulco Gro., México
jraul_lopz@hotmail.com

Resumen— En el presente artículo, se plasma el trabajo interdisciplinario de la Maestría en Sistemas Computacionales con apoyo del CONACyT, impartida en el Instituto Tecnológico de Acapulco. El objeto de este artículo es, presentar la implementación de la herramienta informática SISCMIIE, la cual es un sistema informático destinado a dar seguimiento y control del mantenimiento de infraestructura y equipo en el Instituto Tecnológico de San Marcos (ITSM), programada la parte del backend en NodeJS en una versión web que permite a los usuarios tener acceso desde cualquier punto geográfico con acceso a internet. Este artículo forma parte de una serie de trabajos siendo el segundo de estos, el cual da seguimiento a un artículo previo titulado: **Propuesta de diseño de un sistema de información web para el seguimiento y control del mantenimiento de infraestructura y equipo** [4].

Palabras clave— *NodeJS, Backend, Mantenimiento, MVC.*

I. INTRODUCCIÓN

Las tecnologías para el desarrollo de aplicaciones web, móviles y de escritorio han aumentado considerablemente, surgiendo herramientas que ayudan a la industria del desarrollo de software profesional, actualmente las tecnologías para el desarrollo de aplicaciones web se dividen en dos grandes intereses, los cuales son, Backend y Frontend.

En el backend se ejecutan las tecnologías que por lo general realizan el trabajo más pesado como el de procesar la información y persistirla en la base de datos, algunas de las tecnologías backend son, por mencionar algunas; Asp.Net Core, Ruby, Java y JavaScript. Las tecnologías ejecutadas del lado del Frontend son herramientas que se utilizan para renderizar la

información almacenada en las bases de datos y que ayudan al usuario final a tener una experiencia agradable al introducir o extraer información.

Para la implementación del backend de la herramienta mencionada en este artículo se hará uso del framework NodeJS, el cual es un entorno JavaScript de código abierto que le permite ejecutar JavaScript en el lado del servidor. Esto permite utilizar el mismo lenguaje para ejecutar código tanto en el cliente como ahora en el servidor. NodeJS utiliza una colección de módulos. Los módulos conforman la funcionalidad central de NodeJS y le permiten hacer cosas como trabajar con el sistema de archivos, con protocolos de red (HTTP, TCP, DNS), datos binarios y la capacidad de comunicarse con la base de datos, es importante mencionar que fue diseñado para construir aplicaciones en red escalables [1].

El lenguaje que se utiliza para el desarrollo de la implementación es JavaScript el cual es un lenguaje de script interpretado que utilizando NodeJS se obtiene la capacidad de hacer cosas como hacer solicitudes desde una base de datos y responder a solicitudes HTTP y crear archivos. Otra característica muy importante por mencionar del lenguaje es que en tiempo de ejecución es multiplataforma, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google [2].

La base de datos utilizada para persistir la información del sistema es mongodb, el cual almacena datos en documentos flexibles similares a JSON, haciendo uso de este tipo de bases de datos NoSQL que rompen los paradigmas de las bases de datos relacionales las cuales dictan ciertas características que

debe cumplir una base de datos, es importante resaltar que la información almacenada en las bases de datos NoSQL se recuperan mucho más rápido que una base de datos relacional [3].

II. OBJETIVO GENERAL

El objetivo principal que se pretende lograr con la implementación del software SISCMIIE es gestionar la información de las anomalías y fallos que presentan los equipos e infraestructuras del ITSM.

III. IMPLEMENTACIÓN DEL BACKEND DE LOS MÓDULOS DE SISCMIIE

A partir de esta sección, se presenta la implementación de los módulos del sistema que trata este artículo, en la figura 1 se observa el diagrama de clases, el cual da soporte al desarrollo de SISCMIIE haciendo uso de la tecnología NodeJS. Las clases en las cuales se basó el desarrollo de la implementación son las siguientes: Catalogo, Artículo, Ticket, DetalleTicket, Plan.

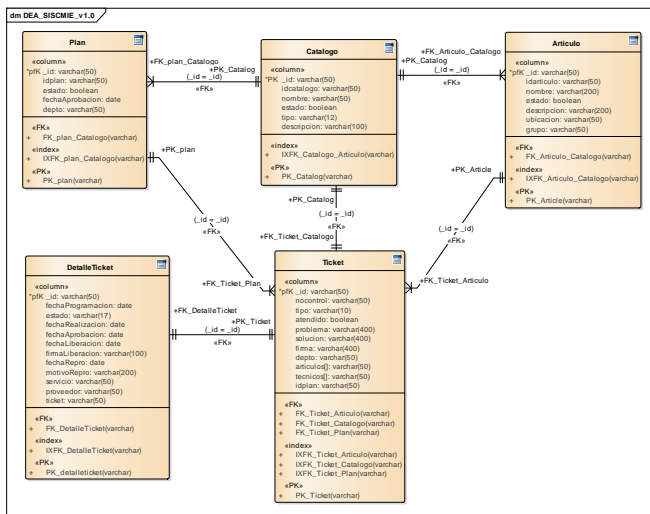


Fig. 1. Diagrama de clases de SISCMIIE.

La primera funcionalidad que se implementó en el API REST (Aplicación ejecutada en el servidor) de SISCMIIE con NodeJS fue la de dar de alta departamentos, servicios, proveedores y grupos que se utiliza para clasificar los equipos e infraestructuras por tipo.

A. Módulo artículo

Este módulo gestiona la parte de los artículos que el sistema administra. A continuación, se presenta la implementación del método para agregar un artículo al sistema, siendo este uno de los más importantes a mencionar debido a que si un artículo no existe en la base de datos no se puede realizar ninguna otra operación sobre sus datos. Las demás funcionalidades que permiten actualizar, buscar y eliminar, si fueron implementadas, aunque en el sistema no se eliminan artículos, más bien cambian de estado, los cuales pueden ser dos, activo o inactivo y al realizar esta operación se utiliza el método para actualizar. En las siguientes líneas de código se muestran la funcionalidad de agregar.

En forma local se envían los datos a la siguiente dirección <http://localhost/article>, los datos son enviados utilizando la aplicación postman para simular un cliente ingresando datos al sistema.

```
app.post('/article', (req, res) => {
  var body = req.body;

  var articulo = new Articulo({
    idarticulo: body.idarticulo,
    nombre: body.nombre,
    estado: body.estado,
    descripcion: body.descripcion,
    ubicacion: body.ubicacion,
    grupo: body.grupo
  });

  articulo.save((err, articuloGuardado) => {
    if(err) {
      return res.status(400).json({
        ok: false,
        mensaje: 'Error creando articulo',
        errors: err
      });
    }
    res.status(201).json({
      ok: true,
      articulo: articuloGuardado
    });
  });
});
```

Como se observa en el código anterior se obtiene a través de la URL un objeto de tipo JSON en el mismo orden que lo dicta el Objeto de tipo Articulo, los datos son recibidos a través del protocolo HTTP y almacenados en la base de datos, si hubiera un error este sería tratado y enviado al cliente que consume el backend, avisando del error ocurrido.

B. Módulo ticket

El módulo de ticket permite gestionar las solicitudes de mantenimiento correctivo que los jefes de cada departamento del ITSM realizan, también es importante mencionar que este mismo módulo se utiliza para realizar el procedimiento de verificación de infraestructuras que se menciona en el primer artículo [4].

Para este módulo se implementaron cuatro métodos que corresponden a un CRUD (Create, Read, Update y Delete) que se utilizan para crear tickets, obtener tickets de la base de datos, actualizar tickets y eliminar tickets. Los cuales se pueden acceder a través de los endpoint o métodos que brindan la funcionalidad dependiendo del verbo (post, put, get y delete)

que brinda el API REST. El endpoint para acceder a estas funcionalidades ejecutando la aplicación de forma local es <http://localhost:3000/ticket>.

```
app.post('/ticket', (req, res) => {
  var body = req.body;

  var ticket = new Ticket({
  });
  ticket.save((err, ticketGuardado) => {
    if(err) {
      return res.status(400).json({
        ok: false,
        mensaje: 'Error creando ticket',
        errors: err
      });
    }
    res.status(201).json({
      ok: true,
      ticket: ticketGuardado
    });
  });
});
```

En las líneas de código anterior muestra el endpoint para crear tickets, el cual recibe a través del protocolo HTTP el cuerpo de tipo JSON con la información a persistir en la base de datos los datos son obtenidos del cuerpo de la petición y mapeados al objeto de tipo Ticket y posteriormente con el método save se guarda la información, en caso de existir un error al guardar la información se envía un error de tipo 400 (se utiliza para indicar al usuario que no está enviando la información correctamente) con un mensaje indicando que ocurrió un error al guardar el ticket y si la persistencia de la información se realiza con éxito, se envía la información que se almacenó en formato JSON para que la persona que realizó la operación realice la tarea que sea necesaria.

Cuando se realiza una verificación que es un procedimiento que se lleva a cabo en la revisión de infraestructuras y equipo del cual se habla en el primer artículo [4], se obtienen un conjunto de datos los cuales son iguales a si realizaras una solicitud de mantenimiento correctivo, es por eso que se optó por utilizar el método de ticket para realizar este procedimiento y almacenarlo en la misma tabla, la única diferencia es que, si se realiza una verificación, el campo tipo, almacena un valor indicando que es una verificación y si es una solicitud se almacena el valor solicitud, posteriormente se reprograma la fecha en que se realizará dicho trabajo y es ahí donde se utiliza el método que se llama DetalleTicket el cual me permite almacenar un detalle de dicha solicitud o verificación.

```
app.post('/', (req, res) => {
  var body = req.body;

  var detalleticket = new Ticket({
    fechaProgramada: body.fechaProgramada,
    estado: body.estado,
    fechaRealizacion: body.fechaRealizacion,
    fechaAprobacion: body.fechaAprobacion,
    firmaAprobacion: body.firmaAprobacion,
```

```
    fechaLiberacion: body.fechaLiberacion,
    firmaLiberacion: body.firmaLiberacion,
    fechaReprogramada: body.fechaReprogramada,
    motivoReprogramada: body.motivoReprogramada,
    servicio: body.servicio,
    proveedor: body.proveedor,
    ticket: body.ticket
  });
```

```
detalleticket.save((err, detalleticketGuardado)
=> {
  if(err) {
    return res.status(400).json({
      ok: false,
      mensaje: 'Error al crear detalleticket',
      errors: err
    });
  }
  res.status(201).json({
    ok: true,
    detalleticket: detalleticketGuardado
  });
});
```

El código mostrado anteriormente corresponde a la funcionalidad que permite agregar detalle a las solicitudes o verificaciones de infraestructuras, los datos que envía el cliente son la fecha en que se programa la atención de la anomalía encontrada o mantenimiento correctivo a resolver, así como el servicio que se realizará, el proveedor que atenderá el servicio si es externo o interno y el identificador único de la solicitud que le corresponde dicho detalle.

La información es recibida por el servidor y almacenada en caso de ser correcta, en caso de que suceda un error al intentar almacenar la información, el servidor enviará un mensaje indicando el error al cliente que realiza la operación en el API REST NodeJS.

C. Módulo plan

La implementación del módulo plan consiste en gestionar información de la base de datos, como estado del plan, los cuales pueden ser en proceso o cerrado, fecha de aprobación del plan y departamento que creó el plan, esta información está relacionada con el módulo de solicitudes, debido a que en el proceso de verificación se hacen solicitudes de los mantenimientos que se realizarán como parte del plan de mantenimiento preventivo que se realizarán durante el semestre en curso en el ITSM. La finalidad de este módulo es asignarle solicitudes de mantenimiento preventivo que posteriormente se le asignarán fecha para ser realizadas.

Los métodos implementados para este módulo son: creación, obtención, actualización y eliminación. La dirección en modo local para acceder a dichos métodos es <http://localhost:3000/plan>, haciendo uso de esta dirección se puede manipular la información de los planes que son creados y están en proceso.

El código mostrado a continuación permite crear un plan, donde los datos se obtienen a través del protocolo HTTP enviados con el verbo post desde el cliente, la información se obtiene del lado del API REST en NodeJS y se almacena si no existe ningún error y si existe algún problema se notifica al cliente que está realizando la operación y se envía un error 400 el cual indica que los datos no se están enviando correctamente.

```
app.post('/plan', (req, res) => {
  var body = req.body;

  var plan = new Plan({
    idplan: body.idplan,
    estado: body.estado,
    fechaAprobacion: body.fechaAprobacion,
    depto: body.depto
  });
  plan.save((err, planGuardado) => {
    if(err) {
      return res.status(400).json({
        ok: false,
        mensaje: 'Error creando plan',
        errors: err
      });
    }
    res.status(201).json({
      ok: true,
      ticket: planGuardado
    });
  });
});
```

IV. RESULTADOS

En este apartado se presenta el resultado de la implementación del API REST en NodeJS.

```
{
  "ok": true,
  "articulo": {
    "estado": false,
    "_id": "5e73fbc7e27120352c4ab6e8",
    "idarticulo": "54321",
    "nombre": "Impresora Epson",
    "descripcion": "Color Negro, Modelo ep-23, Monocromático",
    "ubicacion": "5e6440733fac904658196b63",
    "grupo": "5e6a6d2362444faa94aaef2b"
  }
}
```

Para hacer las peticiones al Backend se utilizó la aplicación postman para no tener que desarrollar un cliente que consuma los datos por el momento. Como se observa en el código

mostrado anteriormente, al crear un artículo el servidor devuelve los datos que se almacenaron en la base de datos, los datos mostrados ya están listos para ser utilizados por cualquier cliente que quiera hacer uso de ellos, es decir, desarrolladores para una aplicación móvil u otra aplicación web que requiera información de los equipos almacenados.

```
{
  "ok": true,
  "ticket": {
    "estado": "ABIERTA",
    "articulos": ["5e583f4898f23361a414bfa4"],
    "_id": "5e655b94f8501f75d0f6e582",
    "idticket": "1",
    "tecnico": "5b3440083fbc905458326b5f",
    "problema": "Reparación de memoria RAM",
    "deptoDestino": "5e6440083fac904658196b5f",
    "deptoOrigen": "5e64403f3fac904658196b60",
    "Aprobacion": "2020-03-08T20:54:44.361Z",
    "createdAt": "2020-03-08T20:54:44.390Z",
    "updatedAt": "2020-03-08T20:54:44.390Z",
  }
}
```

La información en formato JSON que se muestran a continuación, es el resultado devuelta por el servidor SISCMIE al crear un plan, a partir de ese momento la información puede ser mostrada en tablas o manipula de la manera que mejor convenga.

```
{
  "ok": true,
  "plan": {
    "estado": false,
    "_id": "5e66d1890f950d4d60aeb27f",
    "periodo": "ENE-JUN2020",
    "createdAt": "2020-03-09T23:30:17.820Z",
    "updatedAt": "2020-03-19T23:12:53.176Z",
  }
}
```

Las líneas en formato JSON mostradas a continuación, corresponde a los datos devueltos cuando se ingresan información al catálogo general del sistema, en este catálogo se registran departamentos, proveedores, grupos y servicios, en este caso se registró un departamento.

```
{
  "ok": true,
  "catalogo": {
    "estado": true,
    "tipo": "DEPARTAMENTO",
    "_id": "5e66d1680f951f4f60gfb265f",
    "idcatalogo": depto-20,
    "nombre": "Recursos Materiales",
    "descripcion": "Jefatura de materiales",
    "createdAt": "2020-03-09T22:30:17.920Z",
    "updatedAt": "2020-03-19T22:12:53.156Z",
  }
}
```


Los campos de los departamentos, proveedores, grupos y servicios son los mismos, lo único que cambia en cada uno de ellos son los datos que se registran, es por eso que se optó por implementarlo de esta forma, es decir, diferenciarlos por un campo llamado tipo.

V. CONCLUSIONES

En este artículo, se presentó la implementación del backend de un sistema de información web para el seguimiento y control del mantenimiento de infraestructura y equipo, que se encuentra en su etapa de desarrollo, haciendo uso de las tecnologías JavaScript y NodeJS, los cuales son tecnologías fáciles de utilizar y de fácil acceso, pero, sobre todo, tecnologías de uso libre, es decir, con licencias que permiten desarrollar aplicaciones que no te penalizan por hacer uso personal o comercial con productos finales desarrollados con dichas herramientas. NodeJS es una tecnología con la que se pueden desarrollar servicios web rápidos y fáciles, es decir, levantar un servicio se realiza en un tercio del tiempo que llevaría realizarlo con otras tecnologías como Asp.Net Core o Java. Es importante recalcar que NodeJS es una tecnología que tiene buena seguridad y que es respaldada por NPM una biblioteca de librerías online que da soporte a dicho framework.

Hasta este punto la información ya se puede utilizar para hacer pruebas o en su defecto también se puede hacer uso de dicha información para consumirla con una aplicación móvil y desarrollar aplicativos que realicen otras funciones como consultar información de solicitudes pendientes o notificar que no se ha realizado una solicitud que está atrasada, a partir de este punto ya se puede comenzar a utilizar el sistema y darle el rumbo que el cliente considere apropiado. Es importante mencionar que toda la información almacena está protegida haciendo uso de web tokens, es decir, si alguna otra persona ajena al ITSM quisiera consumir la información del sistema esta debe proporcionarles la clave con la que se descripta el token que se envía en cada petición que se realiza para realizan cualquier operación sobre el API REST como Crear, Actualizar, Obtener o Eliminar datos de dicho sistema.

REFERENCIAS

- [1] F. Russ, *Beginning JavaScript. The Ultimate Guide to Modern JavaScript Development: Apress 2019.*
- [2] K. Manuel. ayllusolar. [En línea] 29 de 1 de 2015. [Citado el: 3 de 3 de 2020.] http://ayllusolar.cl/wp-content/uploads/2016/08/node_js_Guia_Principiantes.pdf.
- [3] Zorrilla, Marta. Universidad de Cantabria. [En línea] 3 de 2017. [Citado el: 3 de 3 de 2020.] https://ocw.unican.es/pluginfile.php/2396/course/section/2473/NoSQL_Tema2_MongoDB.pdf.
- [4] R. Ramírez Silva, F. J. Gutiérrez Mata, E. De la Cruz Gámez y E. Cadena Mendoza, «Propuesta de diseño de un sistema de información web para el seguimiento y control del mantenimiento de infraestructura y equipo» vol. 11, n°2, 2019.

Diseño de sistema de cobro por medio de transferencias para el uso en instituciones públicas

Ing. Alejandro Bautista Santos
Estudiante de la Maestría en
Sistemas Computacionales en un
Programa PNPC.
Tecnológico nacional de México
/ Instituto Tecnológico de
Acapulco.
Guerrero, México.
alebtasantos18@gmail.com

M.T.I. Juan Miguel Hernández
Bravo
Docente de Maestría en Sistema
Computacionales.
Tecnológico nacional de México
/ Instituto Tecnológico de
Acapulco.
Guerrero, México.
jmhernan@yahoo.com

M.C. José Francisco Gazga
Portillo
Docente de Maestría en Sistema
Computacionales.
Tecnológico nacional de México
/ Instituto Tecnológico de
Acapulco.
Guerrero, México.
ita.gazga@gmail.com

M.T.I. Rafael Hernández Reyna
Docente de Maestría en Sistema
Computacionales.
Tecnológico nacional de México
/ Instituto Tecnológico de
Acapulco.
Guerrero, México.
rherman7@yahoo.com.mx

Ing. Miguel Angel Jimenez Leal
Estudiante de la Maestría en
Sistemas Computacionales en un
Programa PNPC.
*Tecnológico nacional de México /
Instituto Tecnológico de
Acapulco.*
Guerrero, México.
majleal18@gmail.com

Resumen.—*Los sistemas que utilizan transacciones electrónicas han sido implementados en muchas empresas del mundo, porque estos sistemas facilitan el traspaso de dinero entre entidades bancarias, convenientes al proveer de información significativa requerida dentro de cada transferencia realizada, tal como los detalles de los movimientos realizados dentro de los periodos seleccionados en las diferentes cuentas, así como en los reportes de ingresos de los recursos concernientes a cada concepto que es utilizado en el manejo dentro de las entidades educativas, siendo esto a nivel medio superior y superior, de igual manera estos sistemas comprueban el estado en que se encuentren las transacciones realizadas por medio de los usuarios a los que se les brinda el servicio. En México el uso de las transacciones electrónicas se ha convirtiendo en algo elemental dentro de las empresas privadas, pero en las instituciones públicas donde se requiere solventar el costo de algún concepto que requiera la realización de un pago para algún trámite, no se cuenta con sistemas de este tipo. Este artículo muestra el diseño de una herramienta que proporcionará a las instituciones públicas la oportunidad de obtener un sistema que otorgue la información necesaria en los diferentes conceptos de pago, siendo en este caso el Tecnológico nacional de México / Instituto Tecnológico de Acapulco quien precise en los diferentes departamentos y trámites que realiza, el desarrollo de la aplicación web. Esta herramienta será programada de acuerdo a las especificaciones generales de la institución y podrá mostrar indicadores de los ingresos en cada una de las referencias respectivas a los conceptos generados en tiempo y orden que sean requeridos.*

Palabras Clave— *transacción electrónica, concepto de pago, estancias educativas.*

Consejo Nacional de Ciencia y Tecnología

I. INTRODUCCIÓN

En este artículo se expone el trabajo interdisciplinario realizado en la Maestría en Sistemas Computacionales (MSC) en apoyo del Consejo Nacional de Ciencia y Tecnología (CONACYT), impartida en el Tecnológico Nacional de México (TecNM) / Instituto Tecnológico de Acapulco (ITA).

El objetivo del artículo es: Realizar el diseño de un Sistema web que de manera alternativa a los procedimientos actuales, permita mediante una aplicación web, el uso de transferencias mediante un sistema de pagos electrónicos interbancarios (SPEI) en estancias de educación pública a nivel superior, referente a trámites realizados durante el transcurso de la estancia de los alumnos como estudiantes del plantel.

Las transacciones electrónicas bancarias resultan un sistema brindado diariamente hacia la comunidad, el cual tiene como objetivo ayudar a las personas a transferir dinero de una cuenta bancaria a otra, procedente del mismo banco o de uno externo con alguna sucursal diferente, con la finalidad de economizar tiempo y esfuerzo. Miles de personas lo utilizan diariamente para pagar facturas, cubrir costos de productos o simplemente para hacer llegar recursos a sus familiares.

Por otra parte, las transacciones bancarias se han convertido en una necesidad para la sociedad. Un claro ejemplo es la banca móvil de los diferentes bancos, mediante los cuales se puede acceder a una diversidad de empresas privadas que reciben pagos por medio de transferencias a través de sus aplicaciones tales como *Transfer*, *BBVA móvil*, *Oxxo Pago*, entre otras.

Las aplicaciones que permiten pagos por medio de transferencias han ganado popularidad en la actualidad, obteniendo la atención de los usuarios, y los centros que ofrecen servicios o productos. También se ha vuelto un tema de interés para las empresas, con el desarrollo de esta herramienta buscamos que sea de interés para el ámbito de la educación académica el área en la cual podemos contribuir con el desarrollo de sistemas para otorgar mejoras en los períodos de elaboración de trámites.

Día con día surgen necesidades, las cuales pueden ser resueltas con el uso de las tecnologías y la implementación de nuevos desarrollos de programas con aplicaciones realistas, un ejemplo de esto son los sistemas de banca móvil y sus diferentes usos en las peticiones de los usuarios en la actualidad, los cuales tienen como objetivo la mejorar de la seguridad y eficiencia en el pago de servicios y conceptos.

Actualmente en países desarrollados, como Estados Unidos el porcentaje de incrementos en el uso de las transacciones electrónicas ha sido de un 5.0% para el año 2017 según la revista Forbes en línea, lo que indica un incremento de 82.5 millones de usuarios para ese año, solamente en México.[1] Este tipo de incrementos demuestran que los usuarios buscan siempre sistemas que les permitan un fácil manejo en el uso de sus recursos y al momento de hacer pagos de sus respectivos servicios y facturas correspondientes.

Sin embargo la mayoría de estas aplicaciones muestran un inconveniente al solo prestar servicios para empresas privadas, y no se están desarrollando aplicaciones adaptadas a los sistemas estandarizados en los niveles de educación pública donde se necesita cubrir el costo por la solicitud de trámites requeridos por los estudiantes, lo que ocasiona que los usuarios (estudiantes en su mayoría), no cuenten con un proceso que genere seguridad al realizar sus pagos de manera física por medio de depósitos, por lo que se puede retrasar las actividades, en caso del usuario al tener que invertir jornadas prolongadas de tiempo para realizar un solo requerimiento respectivo a un concepto de trámite, por parte de los administrativos que imparten el servicio dentro de la estancia educativa no tienen las facultades de agilizar el proceso del trámite, si el pago correspondiente no ha sido reflejado y validado por el plantel.

Es por eso que en este artículo se hace la propuesta de un sistema de pagos por medio de transferencias electrónicas para los conceptos correspondientes a trámites dentro de las instituciones educativas públicas.

El resto de este artículo está organizado de la siguiente manera, en la sección 2 se analiza la descripción de la propuesta y un diagrama para el entendimiento del funcionamiento en cuanto a conectividad, en el apartado 3 se muestran los fundamentos teóricos que permitirán seleccionar las piezas claves como son las tecnologías y herramientas a utilizar en el sistema, en la sección 4 se presenta parte de la metodología que se estará utilizando en el diseño y modelado de los diagramas en el desarrollo que se está realizando, los prototipos de las vistas de los módulos del sistema de pagos electrónicos por medio de transacciones se presenta en la sección 5, las conclusiones obtenidas son mostradas en la sección 6.

II. DESCRIPCIÓN DE PROPUESTA DE SOLUCIÓN

A continuación se presenta el modelado de la solución a la problemática, la cual permitirá que las formas de pago del Instituto Tecnológico de Acapulco se puedan realizar por medio del desarrollo de una plataforma web, la cual centralizará los diferentes pagos y otorgará más control de los depósitos realizados, permitiendo el registro de cada una de las transacciones efectuadas en línea, guardando los datos en el servidor del sistema del Instituto Tecnológico de Acapulco, otorgando confiabilidad de la información generada por cada depósito.[2] Siendo la fiabilidad del sistema de suma importancia para la realización de trámites en menor tiempo posible con respecto a la validación de los depósitos, y de esta manera tener el control de cada transacción electrónica de las diferentes operaciones sucedidas en el procedimiento respectivo al seguimiento del proceso de los tramites, considerando estos de vital importancia.

El sistema propuesto para los pagos efectuados por los alumnos del plantel es un complemento, por lo que cada alumno podrá darse de alta por medio de un usuario en la plataforma web que será desarrollada, obteniendo acceso al sistema y con la posibilidad de realizar cada transacción correspondiente al trámite que resulte interesado en efectuar.

Con el nuevo sistema de transacciones, los alumnos ya no tendrán la necesidad de trasladarse a una sucursal bancaria para efectuar los depósitos correspondientes, podrán efectuarlos desde cualquier dispositivo con acceso a internet, debido a que este sistema estará alojado en el servidor del instituto, gracias a la infraestructura con la que cuenta el plantel. El sistema de transacciones estará siempre en comunicación con el servidor del plantel y en constante comunicación a la base de datos, podemos observar la propuesta de conexión que se pretende implementar en el Instituto Tecnológico de Acapulco, (Véase figura 1).

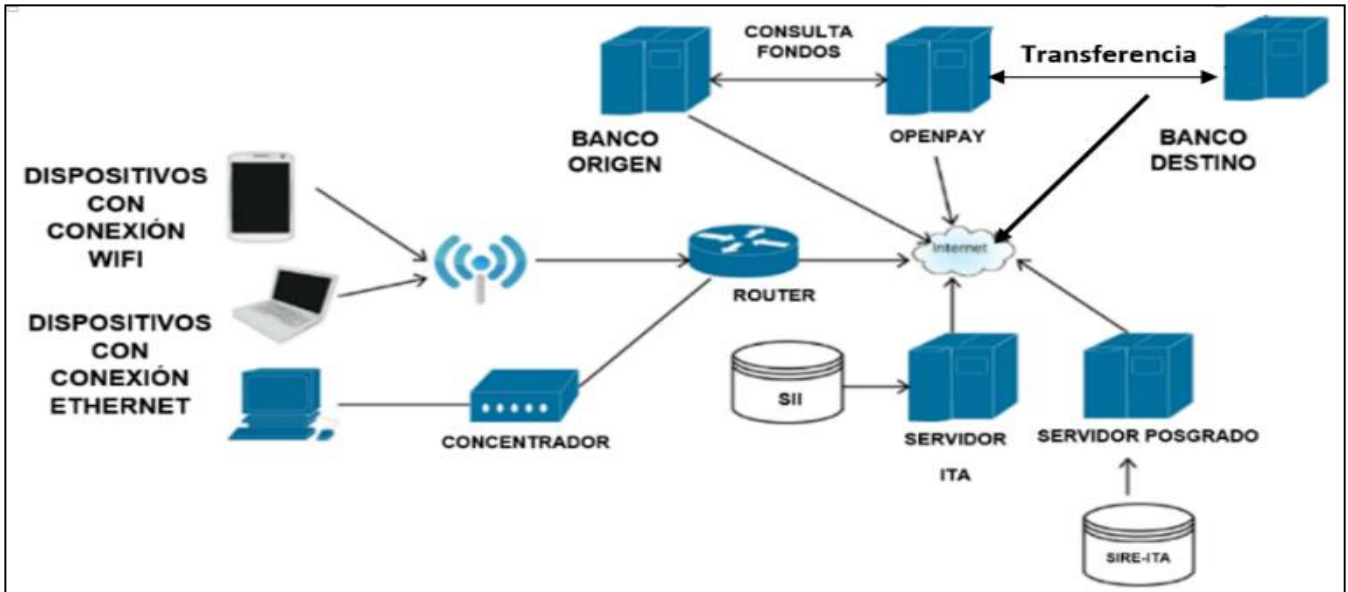


Figura 1. propuesta de de diagrama de arquitectura conexión (caso: Instituto Tecnológico de Acapulco)

III. FUNDAMENTOS TEÓRICOS

El desarrollo de sistemas permite la oportunidad de seleccionar las mejores características de cada uno de los componentes, y esto es referido a elementos que son pieza clave en la arquitectura de los diferentes sistemas, a continuación se presentan los elementos que serán los pilares dentro de la implantación en el desarrollo, resulta sorprendente saber que no son todos los elementos a tener en consideración al momento de codificar, pero si los más relevantes. En los que se encuentran los siguientes.

Protocolos de seguridad: Este tipo de reglas son las que darán la relación necesaria para la transición de datos, dentro de esta comunicación de datos debemos de concentrarnos en conceptos fundamentales. [3]

Integridad de datos: Al ser un sistema que maneja información sensible de suma importancia para los usuarios, se debe de debemos de concentrarnos en que dicha característica conserve su esencia. [4]

Autenticación: Este concepto es centrado en ambas partes involucradas de nuestro sistema, debido a que los datos deben de ser autenticados y los usuarios deben de mostrar dicha veracidad de igual manera al momento de hacer uso de los diferentes módulos del sistema. [5]

Confidencialidad: Dentro de los pasos a seguir en el flujo de la información este es no el más importante pero si, el que se convierte en pieza clave para otorgar confianza de utilización para las personas que están pensando en mover. [6]

Certificado SSL: Certificados utilizados en servicios dentro de los protocolos HTTP (*Hypertext Transfer Protocol*) y HTTPS (*HyperText Transfer Protocol Secure*) estos pueden ser digitales o de clave pública, la mayoría de las ocasiones son las empresas de seguridad las que se encargan de otorgar este servicio; el cual consiste en realizar una verificación de la información por medio de realizar un escáner de la información

contenida dentro del sitio web, que está prestando el servicio o en otros casos ofertando productos, sin lugar a dudas es prioridad conseguir una certificación SSL. [7]

Asp .net identity: esta propiedad de la familia *Asp*, es la que permite la gestión de la identidad de los usuarios y quien les otorga una membresía en el uso de los módulos que lo componen. Este aplicativo de identidad se ha venido utilizando dentro de los entornos de desarrollo de esta tecnología y ha llegado a ser utilizada en la mayoría de las modalidades de las aplicaciones, siendo móvil, web e incluso en las híbridas. Este tipo de identificación ofrecidas por Microsoft permiten que la identidad sea corroborada por medio de sesiones como, *Facebook, google, twitter* entre otros. [8]

IV. METODOLOGÍA

El uso de la metodología en el desarrollo es concerniente al conjunto de técnicas, reglas y maneras de actuar en los procedimientos dentro de los soportes que se documentan en el empleo del diseño de sistemas. El objetivo de que se exponga mediante la documentación en una serie de técnicas en el modelado que permiten los desarrollos de software de calidad, es permitir sistematizar procesos mediante métodos que solucionen problemas planteados. [9]

A. Determinación de Requerimientos

La herramienta en la cual se fundamenta esta fase son los diagramas de casos de usos, los cuales son diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema. Primero que todo es necesaria la recopilación de requerimientos. En este punto, se hace prioridad el identificar los actores y las tareas que ellos deben realizar. Luego, se determinan los escenarios para cada tarea y tipo de actor. Los diagramas de casos de uso, (Véase figura 2) son importantes para visualizar, especificar y documentar el comportamiento del elemento, estos diagramas facilitan que los sistemas, subsistemas y clases sean abordables

y comprensibles, al presentar una vista externa de cómo pueden utilizarse estos elementos en un contexto dado.

Los diagramas de casos de uso también son importantes para probar sistemas ejecutables a través de ingeniería directa y para comprender sistemas ejecutables a través de ingeniería inversa. En el siguiente diagrama podremos observar el caso de uso de manera general, para nuestro sistema propuesto, entre una interacción de la relación que tendrán los casos de uso y los actores, este diagrama propone las vistas principales que tendrá el sistema, siendo este diagrama uno de los más importantes en el modelado dentro de la organización del comportamiento de un sistema, a continuación se describen los puntos del modelo de casos de uso general.

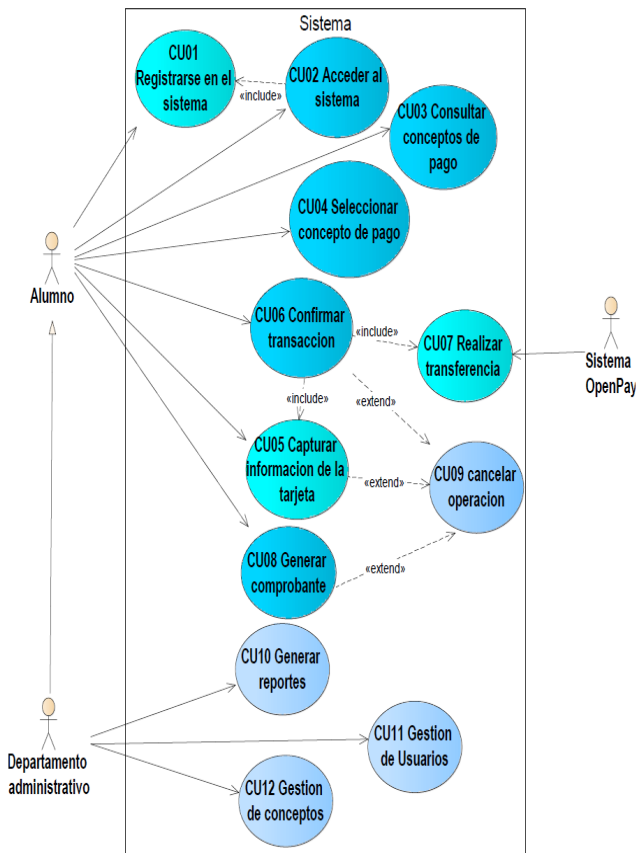


Figura 2. Diagrama de casos de uso general del sistema.

- **CU-01** Permite darse de alta en el sistema, ingresando datos personales con la finalidad de poder realizar alguna transferencia respecto a un concepto de trámite.
- **CU-02** Permite ingresar al sistema, una vez que el usuario se ha registrado, permitiendo que pueda consultar los diferentes apartados, y hacer uso del mismo.
- **CU-03** Permite consultar los diferentes tipos de conceptos con base en los criterios que han establecido los departamentos para la realización de trámites.

- **CU-04** Permite seleccionar uno o varios conceptos de pago, en consecuencia, al trámite que requieran realizar.
- **CU-05** Permite una vez realizado, el proceso de darse de alta, poder ingresar datos correspondientes a la tarjeta que ocupara para realizar la transferencia.
- **CU-06** Permite al sistema consultar los fondos de la tarjeta ingresada y realizar la transferencia al banco-destino correspondiente a la cuenta del plantel.
- **CU-07** El **API-REST** del servidor de **OPENPAY**, procederá a encriptar los datos de transferencia de fondos y otorgara una respuesta de la transacción (exitosa o no exitosa).
- **CU-08** Permite generar el comprobante de la transferencia realizada y consultar los comprobantes de movimientos realizados en periodos anteriores.
- **CU-09** Permite cancelar la operación en cualquier momento durante el proceso. (Esto, antes de realizar la confirmación de transferencia).
- **CU-10** Permite al administrador generar reportes que correspondan a transferencias realizadas por conceptos y periodos respectivamente.
- **CU-11** Permite tener el control de los usuarios que se han registrado en el sistema. (Solo visualización de información, sin poder modificar datos).
- **CU-12** Permite controlar los conceptos de pago. (Crear, Visualizar, actualizar y borrar).
-

B. Diseño Conceptual

Se construye un modelo orientado a objetos que represente el dominio de la aplicación usando las técnicas propias de la orientación a objetos.

La finalidad principal durante esta fase es capturar el dominio semántico de la aplicación en la medida de lo posible, teniendo en cuenta el papel de los usuarios y las tareas que desarrollan.

El resultado de esta fase es un modelo de clases relacionadas que se divide en subsistemas. En el diseño de la base datos que se ocupa para el desarrollo tenemos un esquema propuesto para el sistema aún está en etapa de desarrollo y con el paso del mismo este esquema se hará más grande y con mayor detalle, pero en el actual se tiene nueve tablas, (Véase figura 3).

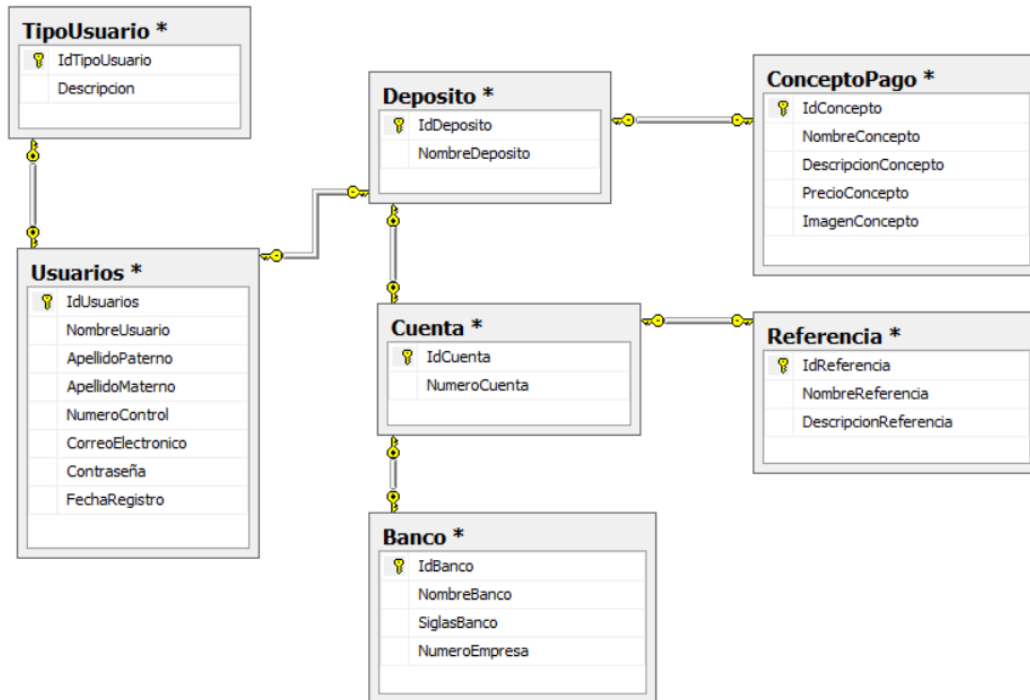


Figura 3. Diseño conceptual de las entidades a utilizar

- **TipoUsuario** el sistema contará con dos usuarios principales que son los administradores que darán creación a los conceptos eventos y a los alumnos que podrán hacer uso de los diferentes apartados que ofrecerá el mismo sistema.
- **Usuarios** podrá hacer la separación entre los dos diferentes usuarios mediante un estado y la descripción.
- **Deposito** una vez realizada la acción de pago la información se verá reflejada mediante esta tabla que guardará todos los datos del depósito realizado.
- **Cuenta** será la tabla que guardará la información que proporcione la institución (banco al cual pertenece la cuenta de la escuela y donde los recursos son puestos).
- **Banco** la información del banco (tanto destino como origen es decir de donde se tomara la información y los recursos así como a donde serán transferidos).
- **Referencia** cada concepto de pago cuenta con información, cada uno con diferentes campos en la empresa (una sección de la cuenta) así como un número de referencia para poder seccionar los recursos.
- **Concepto** pago en los departamentos son generados algunos conceptos de pago concernientes a los trámites que realiza la institución, por tanto esta misma información es guardada en esta tabla.

C. Diagrama de despliegue

Como parte de la propuesta del modelado se tiene la estructura general que muestra los artefactos que pertenecen al mundo real, pues estos permiten una visualización de los aspectos físicos de un sistema, cada uno de sus nodos representan elementos palpables que estarán implicados en el tiempo de ejecución del sistema, el cual debe de contener memoria y gran capacidad de procesamiento para el manejo de los datos que se están insertando y consultando, todo esto durante el uso del sistema. (*El lenguaje unificado de modelado, Booch, Jacobson, Rumbaugh, 2006*).

A continuación se muestra el diagrama de despliegue (Véase figura 4) concerniente a la propuesta del sistema (Véase figura 1), mediante una representación gráfica para los nodos que formaran parte de los recursos computacionales dentro de la arquitectura propuesta para la implementación en el plantel del Instituto Tecnológico de Acapulco.

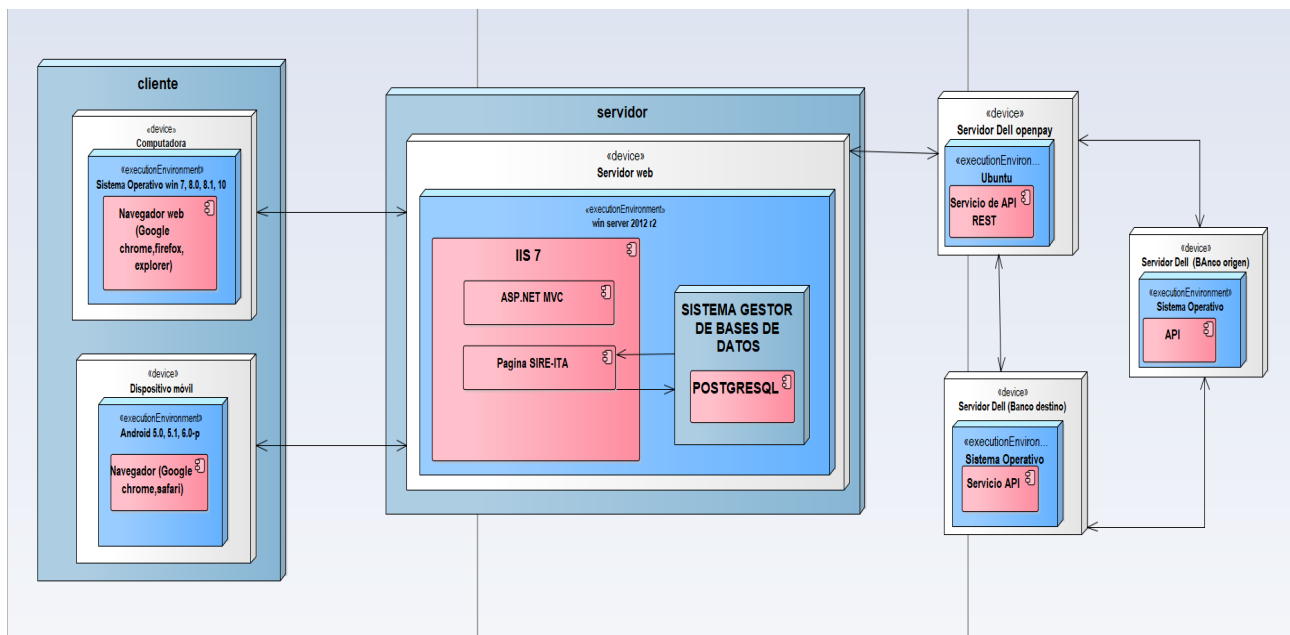


Figura 4. Diagrama de despliegue

V. PROTOTIPO DE VISTAS

El entorno del usuario deberá ser amigable, por tanto el manejo del sistema consiste en vistas sencillas, el primer impacto al momento de entrar a hacer uso del sistema consiste en mostrar al usuario un panorama general que permita conocer, todas las posibilidades que serán posibles realizar, si bien es congruente entender que el sistema consiste en realizar transferencias y obtener esos recursos, el sistema concentrará diversidad de conceptos, así como información de la institución, contacto con la misma y una cuenta respectiva a cada usuario. (Véase figura 5)



Figura 5. Vista principal del sistema

Para poder otorgar confianza al usuario y se sienta con la libertad de poder realizar los trámites concernientes a los pagos, cada uno de los usuarios podrá realizar un registro y así podrá visualizar en un apartado los movimientos que ha realizado, con esto nos referimos a consultas de pagos realizados a los conceptos, en modalidad de reportes por fechas y periodos. (Véase figura 6)

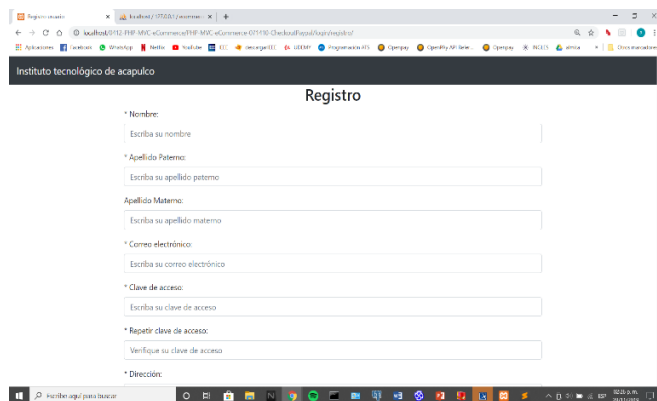


Figura 6. Datos preliminares para crear una cuenta en el sistema

Para el control de los movimientos realizados por los usuarios dentro de cada periodo, es necesario tener un control del recurso que está entrando y la referencia a la cual está siendo ingresada correspondientemente para ello el sistema contará con dos tipos de usuarios los generales que será las personas que podrán realizar transferencias y tendrán la oportunidad de visualizar el entorno de los conceptos de pago y los eventos que requieran cubrir algún costo, y los usuarios administrativos que serán el personal encargado de controlar la información así como la realización de nuevos conceptos y el control de cada uno por medio de reportes. (Véase figura 7)

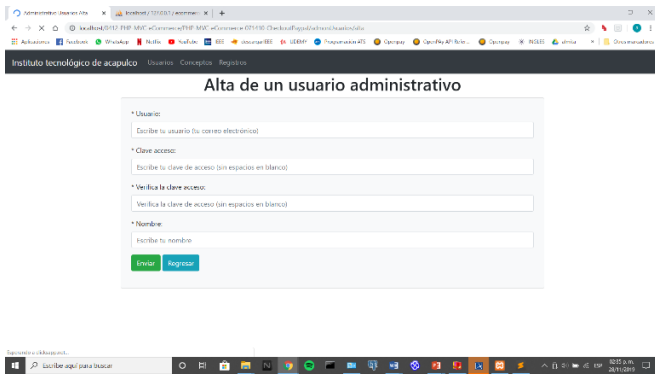


Figura 7. Datos preliminares de un usuario administrativo

El principal motivo, por el cual se ha realizado un estudio y una investigación con el objetivo de crear un sistema que permita realizar pagos concernientes a conceptos de trámites necesarios dentro de la institución, y por tanto deben de ser visualizados de manera clara, concisa y apreciable donde se visualicen y se pueda realizar las trasferencias de dinero. (Véase figura 8)

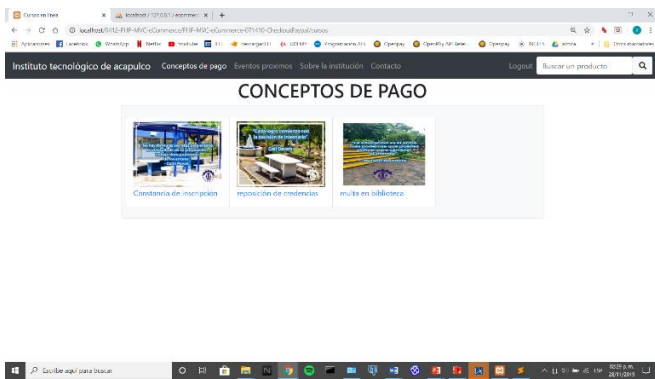


Figura 8. Visualizaciones de conceptos

El sistema que se pretende centralizar y llevar a la realidad, creando un apartado donde se visualice para comodidad del usuario los conceptos más buscados y los que se realizaron menos tiempo al momento de crear el usuario. (Véase figura 9)



Figura 9. Conceptos recientes y más buscados

Generar los reportes que de manera física puedan resaltar la veracidad de los movimientos mediante el sistema y así dar información detallada por periodos que el usuario administrador solicite y requiera en los momentos necesarios. (Véase figura 10)



Figura 10. Formularios de detalles de ingresos por periodos

VI. CONCLUSIÓN

La propuesta de diseño y modelado que se presentó en este artículo corresponde la base para el desarrollo de un sistema de pagos electrónicos interbancarios, para implementar en estancias de educación a nivel medio superior (preparatorias) y superior (universidades), siendo estas las bases que se consideran necesarias para tener un panorama general del desarrollo que será utilizado al momento de crear he implementar la propuesta modelada, se presentaron cuatro apartados, propuesta de solución, fundamentos, metodología con los apartados de determinación de requerimientos, diseño conceptual, diagrama de despliegue y prototipo de vistas. Cada uno de estos diagramas serán los cimientos que darán estructura y permitirán el desarrollo del sistema. Mostrando que tenemos las bases suficientes para realizar el desarrollo y así poder otorgar a las estancias publicas una herramienta que permita realizar este tipo de movimientos con información delicada pero con un sistema confiable y amigable.

VII. REFERENCIAS

- [1] Forbes. (2018). Comercio electrónico representó 5% del PIB de México en 2018. Octubre 22 del 2018, de Forbes México Sitio web: <https://www.forbes.com.mx/comercio-electronico-crece-5-en-mexico/>.
- [2] Ing. Alejandro Bautista Santos M.T.I. Juan Miguel Hernández Bravo M.C. José Francisco Gazga Portillo M.T.I. Rafael Hernández Reyna. (02 julio 2019). PROPUESTA DE UN SISTEMA WEB PARA REALIZAR PAGOS ELECTRÓNICOS INTERBANCARIOS DE TRÁMITES EN ESTANCIAS PUBLICAS: Caso Instituto Tecnológico de Acapulco. Congreso Internacional de Investigación Academia Journals Puebla 2019., 2, 180-186. 13/03/2020, De Memorias del congreso internacional de investigación puebla 2019 Base de datos.
- [3] Mengual Galán, Luis. (2019). Un modelo formal para la especificación, análisis, verificación e implantación de protocolos de seguridad. Lenguajes y Sistemas Informáticos e Ingeniería del Software: Facultad de Informática (UPM) [antigua denominación].
- [4] Bast, Silvia Gabriela | Montejano, Germán Antonio | García, Pablo | Fritz, Estela. (Abril 2015). Evaluación de la integridad de datos en Sistemas de e-Voting. XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015), 2, 170-176. 13/03/2020, De Red de Universidades con Carreras en Informática (RedUNCI) Base de datos.
- [5] Carles Mateu. (2004). Instalación del servidor. En Desarrollo de U Formación de Posgrado aplicaciones web (24). Barcelona: Eureka Media, SL.
- [6] M. Iraburu. (2006). Confidencialidad. 13/03/2020, de universidad nacional autónoma de México Sitio web: <http://scielo.isciii.es/pdf/asisna/v29s3/original5.pdf>
- [7] Angulo Castro, Diana Carolina Henao Leiva, Juan Felipe. (2017). Análisis de herramientas de interceptación para el control de ataques reales de suplantación con certificados SSL. En tesis (10-16). Revista idistrial: Universidad Distrital Francisco José de Caldas.
- [8] Stefan Lazov. (2019). Configurar la identidad ASP.NET. 10 de ene de 2019, de pluralsight Sitio web: <https://www.pluralsight.com/guides/configuring-asp-net-identity>
- [9] Fabián Coelho. (17/05/2019). Significado de Metodología. 13/03/2020, de significados Sitio web: <https://www.significados.com/metodologia/>

Diseño de un sistema Web para toma de decisiones en la asignación de operadores en el SIT ACABÚS

Ing. Edder Felipe Genchi
Estudiante de la Maestría en
Sistemas Computacionales en un
Programa PNPC.
Tecnológico Nacional de México
/IT de Acapulco
Acapulco, Guerrero, México.
efgenchi18@gmail.com

M.T.I. Jorge Carranza Gómez
Docente de la Maestría en
Sistemas Computacionales.
Tecnológico Nacional de México
/IT de Acapulco
Acapulco, Guerrero, México.
jcarranzamx@gmail.com.

M.T.I. Eloy Cadena Mendoza
Docente de la Maestría en
Sistemas Computacionales.
Tecnológico Nacional de México
/IT de Acapulco
Acapulco, Guerrero, México.
eloy_cadena@yahoo.com

Dr. Eduardo de la Cruz Gámez
Docente de la Maestría en
Sistemas Computacionales.
Tecnológico Nacional de México
/IT de Acapulco
Acapulco, Guerrero, México.
gamezeduardo@yahoo.com.com

Resumen— Este artículo se origina por el trabajo interdisciplinario de la Maestría en Sistemas Computacionales con apoyo del CONACYT, impartida en el Instituto Tecnológico de Acapulco. El trabajo se enfoca en el diseño de un sistema Web utilizando notación UML (Lenguaje de modelado Unificado) para apoyar la toma de decisiones y elegir al operador idóneo cuando exista una vacante en el Sistema Integral de Transporte (SIT) ACABÚS. En la primera sección del artículo se presenta la introducción y objetivo general del trabajo de investigación. En la segunda sección se muestran los diagramas y modelos que sirven para representar cómo está estructurado el sistema web, para esto se contempla el Modelado de Negocio, los Casos de Uso, el Diagrama Entidad Asociación (DEA) y el Modelo Navegacional. Este artículo es la continuación de un trabajo anterior que se titula: Propuesta de sistema web para el apoyo de la toma de decisiones en la asignación de operadores en el Sistema Integral de Transporte ACABÚS.

Palabras clave— UML, Modelado de Negocio, ACABÚS, SIT.

I. INTRODUCCIÓN

El SIT ACABÚS es una empresa consolidada en Acapulco Gro., tiene tres años en operación siendo una de las pocas empresas de transporte público que cuentan con seguridad dentro de las estaciones, brindando un servicio de calidad a sus usuarios, comparado con otros medios de transporte, los autobuses padrones y articulados cuentan con aire acondicionado para comodidad de los clientes [1]. La selección del personal que conduce los autobuses se hace en el área de Jefe de Operadores quien se encarga de establecer los horarios, descansos y asignación de unidades; cuando surge un imprevisto el Jefe de Operadores resuelve la problemática lo antes posible para evitar que el servicio de transporte se interrumpa.

Actualmente el proceso de selección de personal en el SIT ACABÚS no está sistematizado, se lleva de manera manual, es por esto que existe la necesidad de agilizar mediante una herramienta, como lo es un sistema Web que permita el almacenamiento de información, clasificación de los operadores, y el apoyo a la selección de los operadores. Al

desarrollar la herramienta disminuirán los tiempos de selección de operadores, mejorando el proceso.

La información obtenida del área de Jefe de Operadores del proceso de selección de operadores sea interpretada, y las características que se piden en el SIT ACABÚS sean lo más cercano a la realidad, sin tener ambigüedad se necesita una notación, un estándar para comprender los diagramas símbolos e imágenes que permitan dar una solución a la problemática existente [2]. Los modelos se usan durante el proceso de ingeniería de requerimientos, ayudan a derivar los requerimientos de un sistema, también durante el proceso de diseño para describir el sistema a los ingenieros que implementan el sistema, y después de la implementación para documentar la estructura y la operación del sistema [3]. Una ventaja que ofrece UML es el desarrollo de aplicaciones globales para la Web, así como la lógica de negocios

La metodología UWE sigue la línea y estándares de UML, que a su vez permite especificar de mejor manera una aplicación Web, incluye varios estereotipos, valores etiquetados y restricciones, facilitando la transición entre modelos y métodos [4]. Esta metodología es utilizada en este trabajo ya que el producto final es un Sistema Web, aplica a la perfección por que define claramente la construcción de los diferentes modelos en esta metodología.

II. OBJETIVO GENERAL DEL PROYECTO

Plasmar el análisis que se hizo en el área de Jefe de Operadores (proceso de selección) del SIT ACABÚS, y diseñar a través de los diferentes diagramas haciendo uso de la notación UML, para desarrollar una herramienta Web que apoye en la toma de decisiones en la asignación de operadores, que permita el manejo de información, así como una correcta clasificación de información.

III. DISEÑO DE LA HERRAMIENTA WEB

El proceso en la selección de operadores se puede representar mediante diagramas y modelados, para entender de manera general las áreas involucradas en dicha selección, esta sección empieza con el Modelado de Procesos de Negocio que permite tener una visión del SIT ACABÚS donde se muestran las responsabilidades, roles y procesos.

A. Modelado de Negocio

En la fig. 1, se muestran cuatro carriles (divisiones horizontales) que son las áreas involucradas para la selección del personal. A continuación, se describen los procesos.

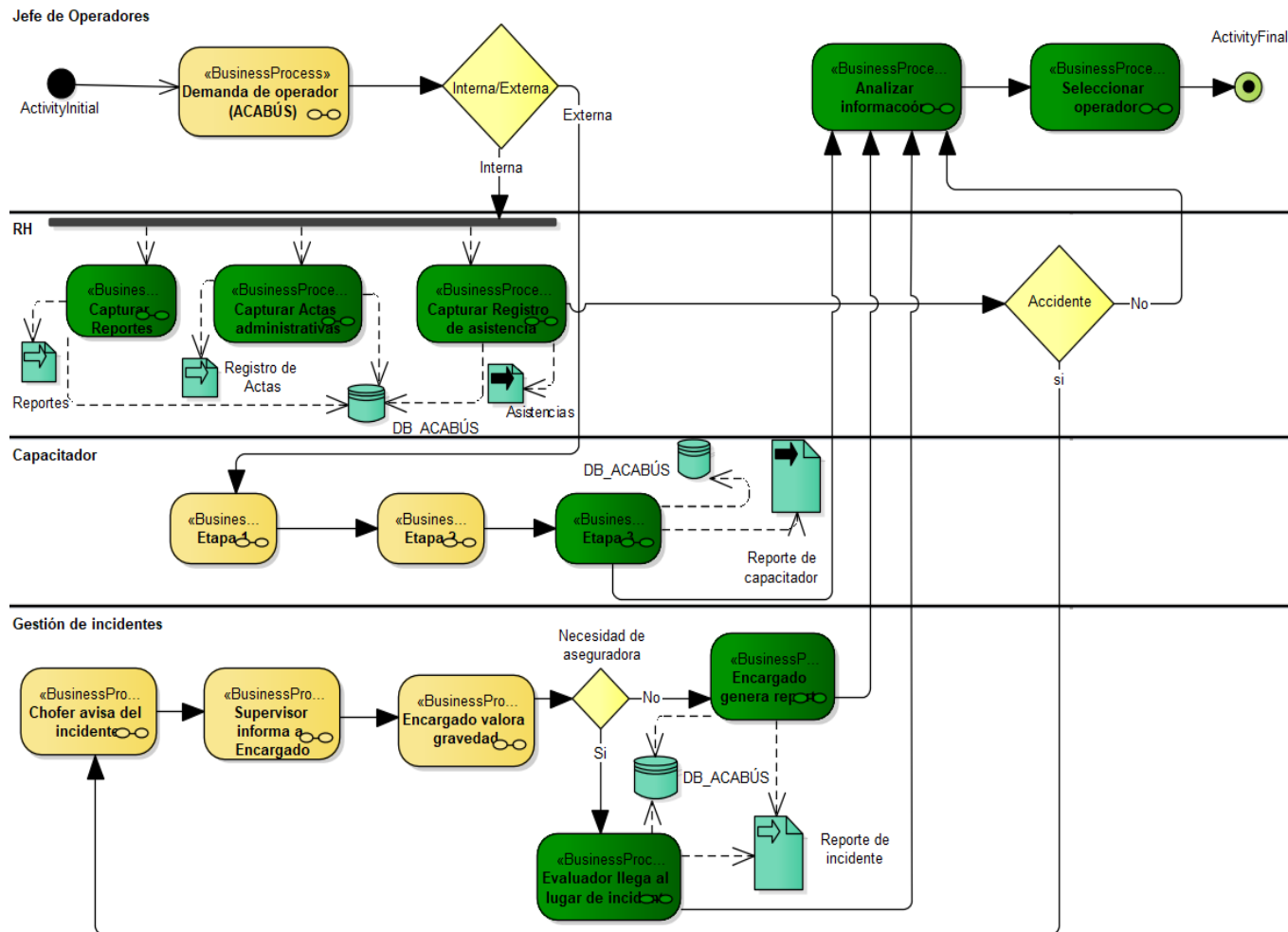


Fig. 1 Modelado de Negocio

- Proceso: **Demanda de Operador (ACABÚS)**. Es el primer proceso y se inicia cuando existe una necesidad de cubrir un puesto para manejar un autobús. En este proceso está involucrado el Jefe de Operadores quien es el encargado de revisar que operador puede ocupar ese lugar.
- Proceso: **Reportes**. El personal de recursos humanos es el encargado de generar los reportes hacia los operadores que incurran en alguna indisciplina, estos reportes pueden ser de los usuarios en general, de los supervisores y de los mismos compañeros de trabajo, hacia los operadores.
- Proceso: **Actas administrativas**. Recursos Humanos es el encargado de generar las actas administrativas hacia los operadores cuando, estas pueden ser generadas cuando un operador tiene tres reportes similares, o cuando hay una indisciplina fuerte.
- Proceso: **Registro de asistencia**. Personal de RH lleva el registro diario de las asistencias de los operadores.
- Proceso: **Chofer avisa del incidente**. El operador comunica del percance al supervisor.
- Proceso: **Supervisor informa a encargado**. El supervisor en turno avisa al jefe de Gestión de incidente.

- Proceso: **Encargado valora gravedad.** El jefe de incidentes revisa si hay necesidad de hablar a la aseguradora para mandar al evaluador.
- Proceso: **Evaluador llega al lugar del incidente.** Si hay necesidad de hablar a la aseguradora se espera hasta que llegue el evaluador para revisar el percance y se genera un documento con los detalles.
- Proceso: **Encargado genera reporte.** De no necesitar la aseguradora, solo se genera el reporte del incidente, y el jefe de incidente es el cargado de hacer esta tarea.
- Proceso: **Etapa 1.** En esta fase el Capacitador realiza diferentes pruebas a los prospectos como son:
 - Encender el autobús
 - Manejar en pendientes
 - Manejar de reversa
 - Estacionarse
- Proceso: **Etapa 2.** El capacitador prosigue en diferentes lugares con otras pruebas para esta fase son:
 - Manejar en colonias con supervisor.
 - Manejar en espacios reducidos.
- Proceso: **Etapa 3.** En esta etapa por ser la última se genera un reporte de los prospectos a manejar un autobús. Además, se incluyen estas pruebas.
 - Manejar en ruta sin pasajeros
 - Manejar en ruta con pasajeros
- Proceso: **Analizar información.** En este proceso es cuando se analiza la información recabada de las diferentes áreas, aquí es donde se revisan detalles y el sistema arrojará un apoyo para la selección de operadores
- Proceso: Selección de operador. Con el apoyo de la herramienta y algoritmo de minería de datos el sistema mostrará la clasificación.

B. Casos de Uso

En el diagrama de modelado de procesos donde interactúa el sistema con el usuario se realiza un diagrama de caso de uso, esto para detallar y ser más entendible lo que se pretende hacer con el sistema Web para el apoyo a los operadores del ACABÚS.

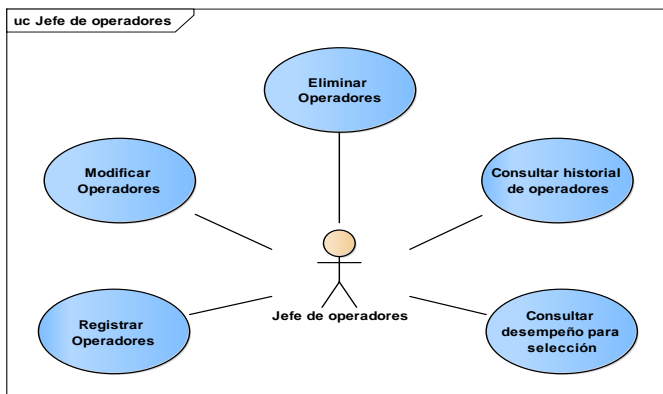


Fig. 2 Caso de uso Jefe de Operadores

El actor de Recursos Humanos es el encargado de generar una serie de reportes y registros de los operadores, como las asistencias diarias, reportes de los usuarios hacia los operadores, y las actas administrativas. El caso de uso de Jefe de Operadores se muestra en la fig. 2.

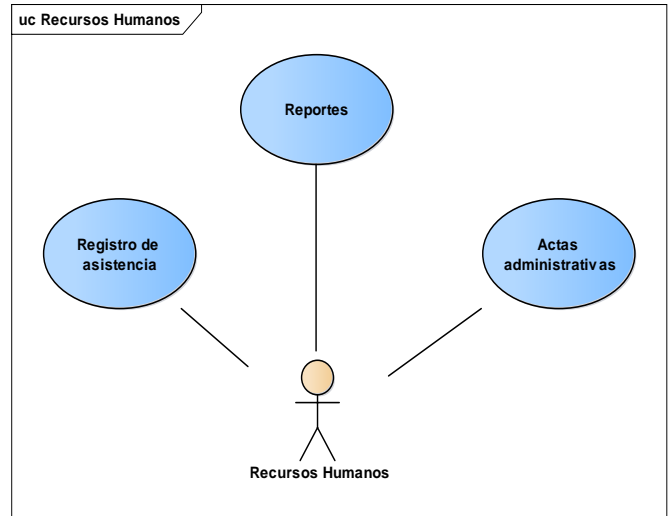


Fig. 3 Caso de uso Recursos Humanos

En el área de Recursos Humanos se registra la asistencia diaria de los operadores, los reportes que hagan los usuarios hacia los operadores y las Actas Administrativas que acumule cada operador, esta información se utiliza para analizar al personal activo que puede ser promovido para manejar otro autobús. El caso de uso de Recursos Humanos se muestra en la fig. 3.

IV. DIAGRAMA ENTIDAD-ASOCIACIÓN (DEA)

En esta parte se muestran las tablas que se podrán utilizar para almacenar la información y que estarán disponibles para la aplicación, así como para realizar las consultas de las descripciones de los operadores. Este diagrama será el modelo para crear la base de datos de la aplicación, de llenarse de manera correcta será indispensable y representará un apoyo al decidir la selección de operadores.

En la fig. 4 se muestra el DEA con las tablas que se utilizan para en el sistema web, la información se centra en el operador, características de cada uno del personal que esta activo para saber sus características, si ha tenido algún choque, información de la licencia del operador y aspectos necesarios que son importantes para manejar un autobús. También hay una tabla de asistencia y reportes, para determinar su comportamiento dentro del trabajo.

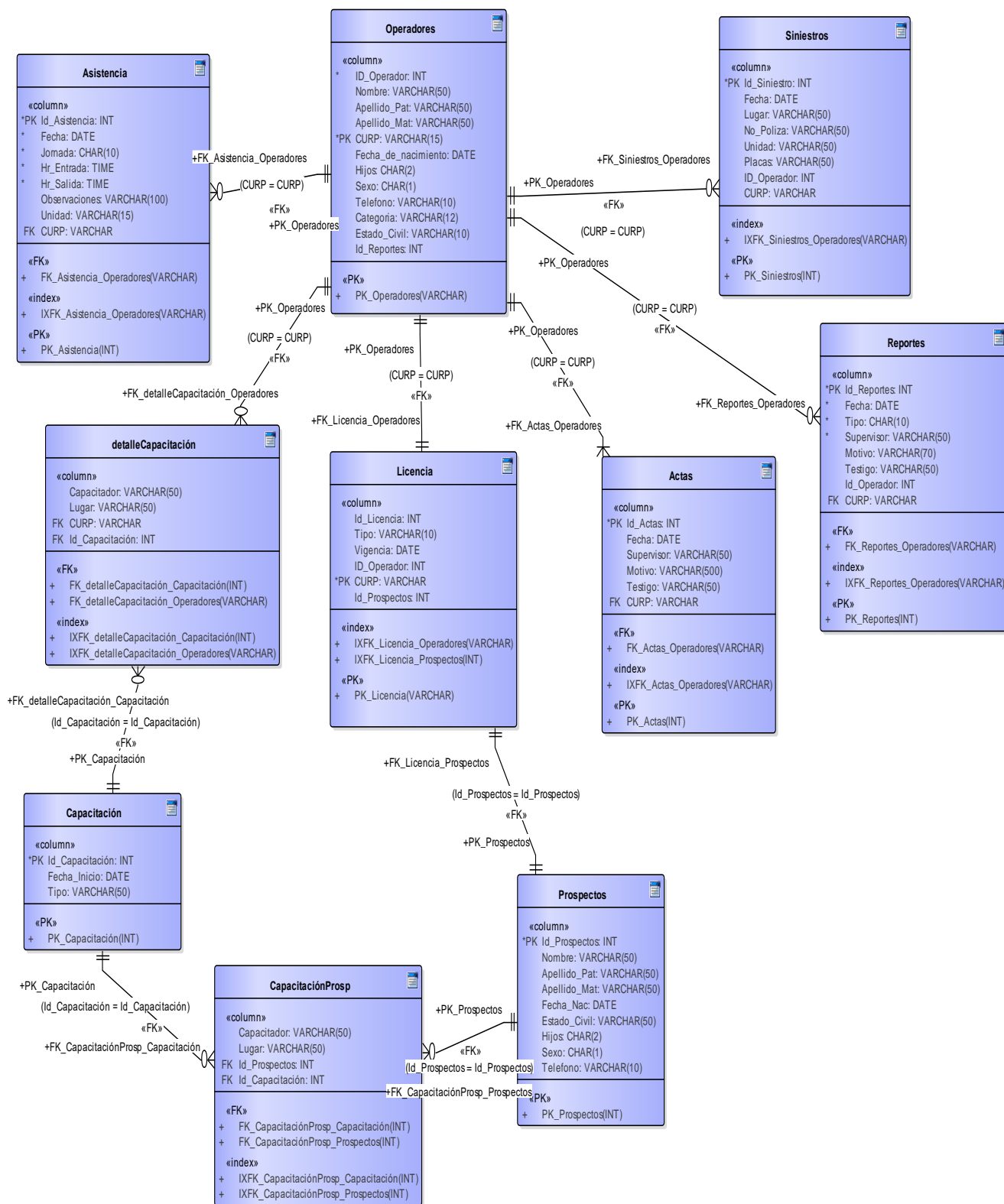


Fig. 4 Diagrama Entidad Asociación

V. DISEÑO NAVEGACIONAL

En la metodología UWE hay una fase donde se presenta la navegación dentro de la aplicación y un conjunto de estructuras como menús, índices y consultas [5].

En este modelo navegacional se representan los nodos y los enlaces de la estructura y el diseño de las rutas navegacionales del prototipo del Sistema Web para el apoyo de la selección del personal del SIT ACABÚS mediante el siguiente diagrama, en el cual se puede observar el estereotipado de la metodología UWE.

Los estereotipos de UWE en la fase de navegación son los siguientes [6]:

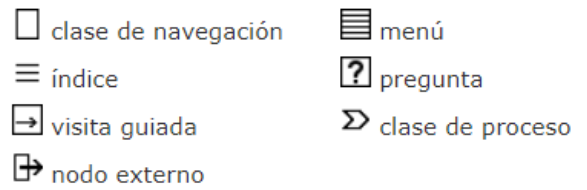


Imagen 1 Estereotipos y sus iconos.

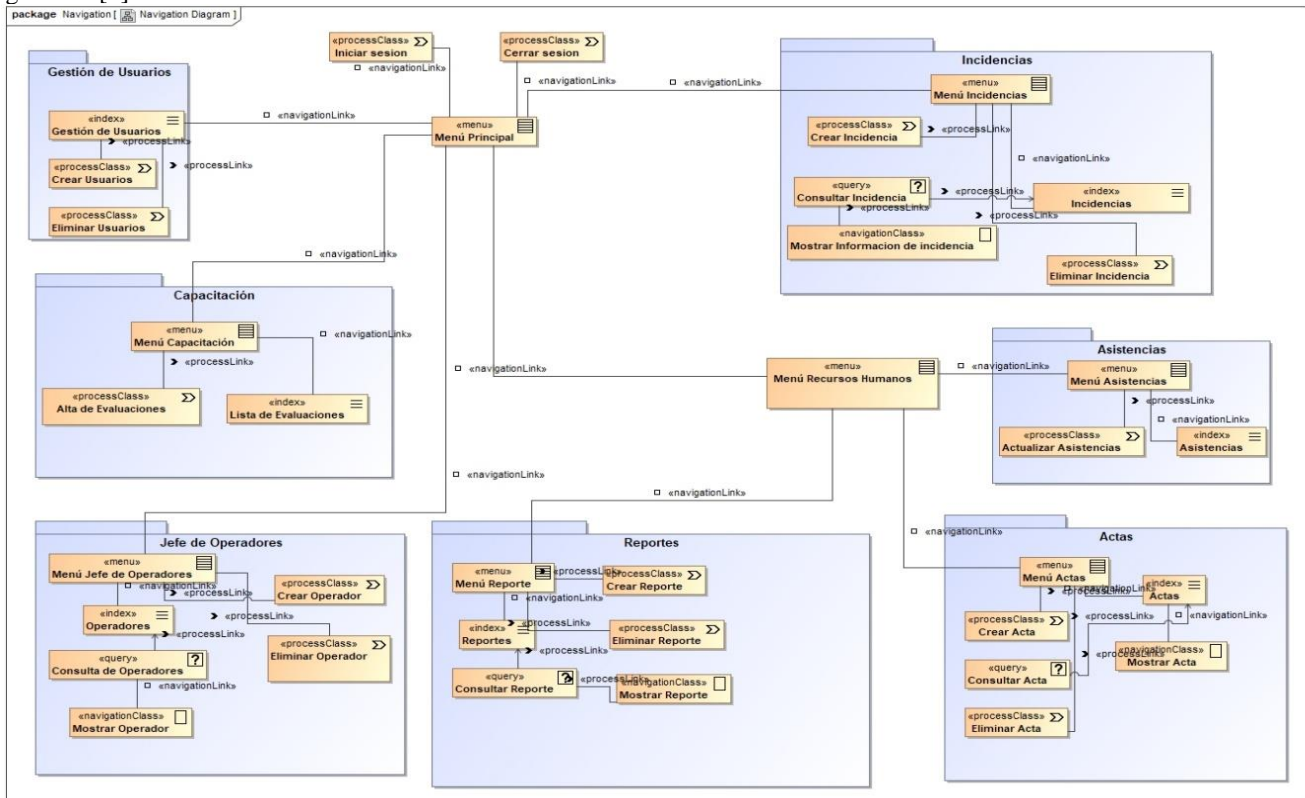


Fig. 5 Modelo navegacional

En este mapa de navegación contextual se muestran las rutas de navegación que podrá realizar el usuario por la aplicación web, previamente, se deberá autenticar para tener acceso al menú principal y acceder a los diferentes módulos del sistema.

Los submenús en el diagrama navegacional son los siguientes:

- Submenú Recursos Humanos.
 - Asistencias
 - Actas
 - Reportes
- Submenú Jefe de Operadores
- Submenú Incidencias.
- Submenú Capacitación

VI. CONCLUSIONES

En este trabajo se presentó el diseño del proyecto de selección de operadores para el SIT ACABÚS, se mostró el modelado de procesos de negocios, con lo que se tiene un panorama general sobre el proceso de selección, desde que se inicia con una vacante pasando por las diferentes áreas obteniendo información y analizando para obtener al candidato idóneo. También se mostró el caso de uso de Jefe de Operadores ya que es el departamento principal donde se toman las decisiones y se tiene la responsabilidad cumpliendo con la tarea de elegir y tener a los mejores operadores manejando un autobús. Los diagramas anteriores darán soporte al desarrollo del Sistema web.

REFERENCIAS

- [1] Felipe, E (15 de Agosto 2019) Propuesta de sistema web para el apoyo de la toma de decisiones en la asignación de operadores en el Sistema Integral de Transporte ACABÚS. *Memorias del Congreso Internacional de Investigación Academia Journals Puebla 2019*. Puebla, Puebla, México.
- [2] Kimmel, P (2010). Guía de planificación de Sistemas BRT. Fondo global del ambiente, 3, 33.
- [3] Somerville, I. (2016). *Ingeniería de Software. México: Pearson*.
- [4] Rotta, D., Pallota, G., Klikailo H., & Belloni, E.. (2016). Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web. En Concurso de Trabajos Estudiantiles(pp 252 - 267). 19º Concurso de Trabajo Estudiantil: JAHO
- [5] Navarro, M. A (2006) Conceptualización, prototipado y proceso de aplicaciones hipermedia.
- [6] Koch, N., Knapp, A., Wirsing, M.. (2016). Tutorial - Navigation Model. Marzo 2020, de Web Engineering Group Sitio web: <https://uwe.pst.ifi.lmu.de/teachingTutorialNavigation.html>