

TECNOLÓGICO  
NACIONAL DE MÉXICO



# COMAROB 2024

**XXVI Congreso Mexicano de Robótica**  
**Asociación Mexicana de Robótica**

Torreón, Coahuila, México  
Del 13 al 15 de noviembre de 2024

## MEMORIAS

*Editado por Miguel Ángel Llama Leal*



ISBN en trámite



# **NUEVAS TENDENCIAS EN INVESTIGACIÓN CIENTÍFICA DE LA ROBÓTICA EN MÉXICO**

**Asociación Mexicana de Robótica e Industria, A.C.**

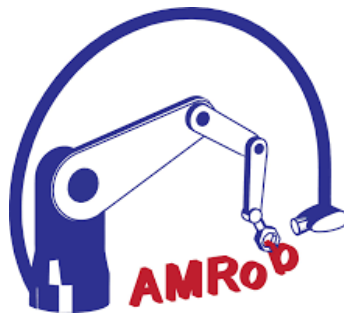
**Editado por**

**Miguel Ángel Llama Leal**

**Colaboradores**

**Tecnológico Nacional de México**

**Instituto Tecnológico de La Laguna**



**Torreón, Coahuila, México, 2024**

**ISBN: en trámite**

# Asociación Mexicana de Robótica e Industria, A.C.

## *JUNTA DIRECTIVA*

### **PRESIDENTE**

Dr. Alejandro Enrique Dzul López  
TecNM/Instituto Tecnológico de La Laguna  
[aedzull@lalaguna.tecnm.mx](mailto:aedzull@lalaguna.tecnm.mx)

### **VICEPRESIDENTE**

Dra. Ivonne Gabriela Zepeda Valencia  
Facultad de Ingeniería, Universidad Autónoma de Baja California  
[zepedavalencia93@gmail.com](mailto:zepedavalencia93@gmail.com)

### **SECRETARIO**

Dr. Ervin Jesús Álvarez Sánchez  
Universidad Veracruzana  
[eralvarez@uv.mx](mailto:eralvarez@uv.mx)

### **TESORERO**

Dr. César Higuera Verdugo  
TecNM/Instituto Tecnológico de La Paz  
[cesar.hv@lapaz.tecnm.mx](mailto:cesar.hv@lapaz.tecnm.mx)

## COMITE ORGANIZADOR

### **Presidente**

Dr. José Irving Hernández Jácquez  
TecNM/Instituto Tecnológico de La Laguna

### **Miembros**

Dr. Alberto Hussein De La Torre Sotelo  
IEEE Monterrey Section

Dra. Patricia Melin Olmeda  
IEEE Computational Intelligence Mexican Council Chapter

Dr. Alejandro Enrique Dzul López  
TecNM/Instituto Tecnológico de La Laguna

Dr. Miguel Ángel Llama Leal  
TecNM/Instituto Tecnológico de La Laguna

Dr. Francisco Ronay López Estrada  
TecNM/Instituto Tecnológico de Tuxtla Gutiérrez

Dr. Víctor Samuel De León Gómez  
TecNM/Instituto Tecnológico de La Laguna

Dra. Ivonne Gabriela Zepeda Valencia  
Facultad de Ingeniería  
Universidad Autónoma de Baja California

Dr. Ervin Jesús Álvarez Sánchez  
Universidad Veracruzana

Dr. César Higuera Verdugo  
TecNM/Instituto Tecnológico de La Paz

## COMITÉ EDITORIAL

### **Presidente**

Dr. Miguel Ángel Llama Leal  
TecNM/Instituto Tecnológico de La Laguna

### **Miembros**

Dr. Alejandro Enrique Dzul López  
TecNM/Instituto Tecnológico de La Laguna

Dra. Ivonne Gabriela Zepeda Valencia  
Facultad de Ingeniería  
Universidad Autónoma de Baja California

Dr. Francisco Ronay López Estrada  
TecNM/Instituto Tecnológico de Tuxtla Gutiérrez

Dr. Víctor Samuel De León Gómez  
TecNM/Instituto Tecnológico de La Laguna

## PRESIDENTE DE PROGRAMA

Dr. Francisco Ronay López Estrada  
TecNM/Instituto Tecnológico de Tuxtla Gutiérrez

## COMITÉ CIENTÍFICO

Dr. Rafael Kelly Martínez – CICESE  
Dr. Juan Manuel Ibarra Zannatha – CINVESTAV  
Dr. Alfonso Pámanes García – TecNM/IT La Laguna  
Dr. Víctor Adrián Santibañez Dávila – TecNM/IT La Laguna  
Dr. Emilio González Galván – UASLP  
Dr. Rubén A. Garrido Moctezuma – CINVESTAV  
Dra. Karla Anhel Camarillo Gómez – TecNM/IT Celaya  
Dr. Ulises Zaldívar Colado – UAS  
Dra. Dora Aydée Rodríguez Vela – UPSIN  
Dr. Jesús Alberto Sandoval Galarza – TecNM/IT La Paz

# Índice

Asociación Mexicana de Robótica e Industria A.C. ....	III
Comité Organizador .....	IV
Comité Editorial .....	V
Presidente de Programa .....	V
Comité Científico .....	V

## Capítulo 1 – Control

### ***ESTRATEGIAS DE CONTROL BASADAS EN FUNCIONES DE LYAPUNOV PARA MANIPULADORES ROBÓTICOS***

Mario Ivan Nava Bustamante, José Luis Meza Medina, Rodrigo Loera Palomo .....	2
---	---

### ***CONTROL NO LINEAL DE UN CONVERTIDOR BOOST CUADRÁTICO CON PROCESAMIENTO DE POTENCIA REDUNDANTE REDUCIDA***

Cuauhtémoc Guerrero, Víctor Santibañez, José Luis Meza-Medina .....	8
---	---

### ***DISEÑO DE UN CONTROLADOR EN TIEMPO-FINITO PARA REGULACIÓN EN ROBOTS MÓVILES NO-HOLÓNOMOS***

Manuel Mera, Héctor Ríos .....	14
--------------------------------	----

### ***CONTROL DE SISTEMAS MULTIAGENTE CON LA PLATAFORMA LEGO EV3***

Zurisadai Flores-Murcia, Daly Y. Lara-Solís, Diego González-Torres, Jesús Santiaguillo-Salinas .....	20
--	----

### ***SOBRE REGULACIÓN DE POSICIÓN ARTICULAR DE ROBOTS MANIPULADORES***

Javier Moreno, Gabriela Zepeda .....	27
--------------------------------------	----

### ***SINCRONIZACIÓN DE 3 SISTEMAS MECÁNICOS DE 1 GDL***

Ervin Jesús Alvarez-Sánchez, José Gustavo Leyva-Retureta, Fernando Aldana-Franco, Rosario Aldana-Franco .....	33
---	----

## Capítulo 2 – Robótica Móvil

<b><i>IMPLEMENTATION OF A HARDWARE SYSTEM FOR THE REMOT ESTATION OF AN ROV/UUV</i></b>	
Marisol Vázquez Tzompantzi, Bruno Yael Silva Morales, Suresh Kumar Gadi, Carlos Horus Acosta Espinoza, Marco Sekehen Gonzalez Tepoz, Andy Alan Castro Valdez .....	39
<b><i>CONTROL DIFUSO EVOLUTIVO PARA UN SISTEMA QUADROTOR</i></b>	
Emmanuel Adrian Espinoza Gómez, Arturo Tellez Velazquez, Esther Lugo González .....	45
<b><i>MATCHING INDOOR NAVIGATION SYSTEMS TO APPLICATION REQUIREMENTS: A COMPARATIVE STUDY</i></b>	
German Cuaya-Simbro, Jorge Armando García Bautista .....	51
<b><i>MÉTODO DE CONTINUACIÓN HOMOTÓPICA PARA LA PLANEACIÓN DE TRAYECTORIAS EN EL ROBOT TURTLEBOT3</i></b>	
María de las Nieves Juárez Manny, Alejandro Enrique Dzul López .....	57
<b><i>INDOOR AUTONOMOUS NAVIGATION FOR QUADCOPTER USING MONOCULAR VISUAL SLAM</i></b>	
Rogelio Enrique Jiménez Trejo, Juan Manuel Ibarra Zannatha .....	63
<b><i>COMMISSIONING OF THE NVIDIA JETSON AGX ORIN MODULE FOR A TRAFFIC SIGNALS DETECTOR BASED ON YOLO</i></b>	
Gibrán López Burrola, Sergio López Hernández, Miguel Ángel Llama Leal, Juan Sifuentes Mijares .....	68

## Capítulo 3 – Diseño de Robots

<b><i>CITLALI: A WHEELED MOBILE ROBOT INSPIRED BY STARFISH</i></b>	
Juan C. Tejada, Ignacio Pietra-Santa, Dariel Cruces-Delmar, E.G. Hernandez Martinez, Alexandro López-González .....	76
<b><i>REMOTELY CONTROLLED VEHICLE BY FACIAL EMG</i></b>	
Alexis Benítez Rosales, Daniel Armando Montiel De León, Rey David Solis Galíndrez, Verónica Esther Arriola Ríos, Sergio Enrique Solís Nájera .....	82

<b>DESIGN OF A LEARNING ROBOT FOR USE IN THERAPIES WITH AUTISTIC CHILDREN: A MECHATRONIC APPROACH</b>	
Arturo N. Cordova-Zacarias, María S. Alejandre-Apolinar, Hugo Amores-Pérez, Irma A. García-Gonzalez, Osmar S. Morales-Varela, Octavio A. García-Alarcón .....	87

<b>PROTOTIPO DE SIMULACIÓN DE UNA MIMOSA PÚDICA</b>	
Rodolfo Romero Herrera, María Fernanda Guillén Pérez .....	93

## Capítulo 4 – Robótica Educativa

<b>ESTRATEGIA PARA LA ENSEÑANZA DE PROGRAMACIÓN A TRAVÉS DE ROBOTS DE MINISUMO</b>	
Mario Salvador Castro Zenil, Alma Alvarado López, Guillermo Rey Peñaloza M. ....	98

<b>IMPLEMENTACIÓN DE PROYECTOS DE ROBÓTICA COMO ESTRATEGIA DIDÁCTICA ENFOCADA EN REFORZAR LOS APRENDIZAJES ESPECÍFICOS PROPUESTOS POR ABET</b>	
Gerardo I. Pérez-Soto, Carlos G. Manríquez-Padilla, Juan José Saucedo-Dorantes, José Villagomez-Moreno, Karla Anhel Camarillo-Gómez .....	104

## Capítulo 5 – Nominados al Premio Rafael Kelly

<b>CONTROL ROBUSTO PARA REGULACIÓN DE POSICIÓN DEL SISTEMA CARRO-PÉNDULO AFECTADO POR PERTURBACIONES CONSTANTES</b>	
Samir Cota, Jesús Sandoval, Javier Moreno-Valenzuela, Víctor Santibáñez, Luis Cervantes-Pérez, César Higuera .....	113

<b>RELAJACIÓN DE CONDICIONES DE EXCITACIÓN PARA UN CONTROLADOR DE SEGUIMIENTO DE TRAYECTORIAS ADAPTABLE CON REALIMENTACIÓN DE SALIDA PARA ROBOTS MANIPULADORES ACCIONADOS POR PAR</b>	
Luis Cervantes-Pérez, Víctor Santibáñez, Jesús Sandoval .....	120

<b>SEGUIMIENTO DE TRAYECTORIA CON MODELO CINEMÁTICO EXTENDIDO PARA UN ROBOT ACKERMAN</b>	
Jasmanik Torres-López, Hiram Garcia-Lozano, Jesús Santiaguillo-Salinas, Jesús Reyes-Rios, Luis Castro-Romero, José Alfonso Jiménez.....	127

<b>CONTROL CONTINUO EN TIEMPO FINITO PARA ROBOTS MANIPULADORES</b>	
Lucero De Santiago Reyes, Víctor Adrián Santibáñez Dávila, Jorge Alberto Villalobos Chin .....	133

<b>CONTROL GESTUAL DE UN ROBOT MÓVIL OMNIDIRECCIONAL MEDIANTE CNN</b> Carlos-A. Salado-Chávez, Matew Herrera-Perez, Karla I. Salado-Chávez, Oscar D. Ramírez-Cárdenas, Esther Lugo González .....	139
<b>ESTIMACIÓN DE LA POSICIÓN PARA EL SEGUIMIENTO DE UN ROBOT MÓVIL CON RUEDAS</b> Servando Encina-Espino, Alejandro Enrique Dzul López, Pedro Castillo, Héctor Ríos Barajas .....	145
<b>GENERACIÓN DE TRAYECTORIAS EN LÍNEA PARA EL CONTROL DE CAMINADO A TRAVÉS DEL MOMENTO ANGULAR</b> Emanuel Márquez-Acosta, Víctor De-León-Gómez, Víctor Santibañez .....	152

## Capítulo 6 – Modelado Cinemático

<b>PRUEBAS EXPERIMENTALES DE TIEMPO REAL EN EL CONTROL CINEMÁTICO DE MOVIMIENTO DE UN ROBOT CUADRÚPEDO</b> M. C. Martínez-Rodríguez, A. Y. Aguilar-Bustos, E. Bugarin, J. L. Muñoz-Torres ....	160
<b>COMPARATIVE ANALYSIS OF FORWARD KINEMATICS IN ROBOTIC MANIPULATORS: HOMOGENEOUS MATRICES, SCREW THEORY, AND QUATERNIONS</b> Jesús A. Medrano-Hermosillo, Abraham E. Rodríguez-Mata, Victor A. Gonzalez-Huitron, Jesús Ricardo López Gutiérrez, Rafael Sandoval Rodríguez, Larbi Djilali .....	166
<b>ANÁLISIS DE ACELERACIÓN DE UN MANIPULADOR PARALELO 6 UPUR</b> Fredy Uriel Nava-Andrés, Manuel Arias-Montiel .....	172

## Capítulo 7 – Estimación

<b>ESTIMACIÓN DE POSTURA EN UN ROBOT AUTOBALANCEABLE USANDO VISIÓN Y REDES NEURONALES SUPERVISADAS</b> Armando Saenz, Issac Gandarilla, Arturo Tadeo Espinoza Fraire, Cristina Saenz .....	179
<b>ESTIMACIÓN Y COMPENSACIÓN DE PERTURBACIONES EN UNA FORMACIÓN EN CADENA SOBRE EJES MÓVILES</b> J. Eduardo Ruiz Ruiz, Julio A. Báez Hernández, Martin Velasco Villa, Alejandro Rodriguez Angeles .....	186

<b>IMPLEMENTACIÓN DE UN OBSERVADOR DE TIPO LUENBERGER PARA EL SISTEMA CAÓTICO DE KAPITANIAK CON EL USO DE TARJETAS DE ADQUISICIÓN DE DATOS</b> Tonalli C. Galicia López, Paul J. Campos Hernández, Diana Gamboa Loaiza, Luis N. Coria de L.R. ....	193
--	-----

## Capítulo 8 – Robótica Humanoide

<b>SISTEMA MULTIMODAL DE RECONOCIMIENTO DE EMOCIONES EMPLEANDO LA INTERACCIÓN CON EL ROBOT HUMANOIDE NAO</b> Vanessa Cristell Torres López, José Anibal Arias Aguilar, Eduardo Sánchez Soto .....	200
<b>ÓRTESIS PARA REHABILITACIÓN DE LA MANO IMPRESA EN 3D</b> Valentina Islas-González, Gonzalo Solano-Ruiz, Angel D. Hernández-Lecourtois, Axel Cid-Trejo, Noemi Hernández-Oliva, Marisol Alejandre-Flores .....	206

## Capítulo 9 – Monitoreo y Adquisición de Datos

<b>MONITOREO IOT MEDIANTE RASPBERRY PI</b> René Sanchez Linares, Rodolfo Romero-Herrera .....	213
<b>SISTEMA DAQ PARA EL MONITOREO Y ANÁLISIS DE SISTEMAS METALÚRGICOS DURANTE EL CICLO TÉRMICO DE SOLIDIFICACIÓN</b> Gonzalo Hedain López-Mera, Diego Sánchez-García, Raúl Dalí Cruz-Morales, David Tinoco-Varela, Alma Alejandra Luna-Gómez .....	219

# CAPÍTULO 1



## CONTROL

# Estrategias de Control basadas en Funciones de Lyapunov para Manipuladores Robóticos

Mario Ivan Nava Bustamante  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de La Laguna*  
México  
navabustamantem@gmail.com

José Luis Meza Medina  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de La Laguna*  
México  
jlmezam@correo.itlalaguna.edu.mx

Rodrigo Loera Palomo  
*CONAHCYT*  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de La Laguna*  
México  
rloerapa@conahcyt.mx

**Resumen**—Originalmente, la teoría de Lyapunov fue concebida específicamente para analizar sistemas en lazo cerrado con ausencia de entradas de control. Si se consideran sistemas con entradas de control, el enfoque basado en las Funciones de Lyapunov de control (*Control Lyapunov Function* o CLF), puede aportar facilidades de análisis. Una CLF es una candidata a ser Función de Lyapunov (*Lyapunov Function* o LF), gracias a que su derivada puede hacerse negativa mediante la elección idónea de la entrada de control. La existencia de una CLF es suficiente (y necesaria) para estabilizar asintóticamente a un sistema nominal. En este trabajo se presenta un procedimiento mediante el cual se puede obtener una CLF que permita proponer esquemas de control para tareas de regulación, considerando robots manipuladores modelados mediante las ecuaciones de Hamilton como caso de estudio. La obtención de este tipo de reguladores permite asegurar la estabilidad asintótica del sistema.

**Palabras clave**—Funciones de Lyapunov de Control, Control de robots, Modelado Hamiltoniano, Control basado en Lyapunov.

## I. INTRODUCCIÓN

Las propiedades dinámicas de las ecuaciones diferenciales representativas de un sistema se pueden determinar y analizar mediante la teoría de Lyapunov. Los teoremas que sustentan la teoría de Lyapunov, requieren de la existencia de una llamada Función de Lyapunov (*Lyapunov Function* o LF)  $V(x)$ . El método de Lyapunov, que fue introducido originalmente como una herramienta de análisis de estabilidad de sistemas de control, ha resultado ser también una herramienta muy útil en el diseño de controladores realimentados considerando sistemas Hamiltonianos. Lo anterior se establece mediante la implementación de una llamada Función de Lyapunov de Control (*Control Lyapunov Function* o CLF) [1] [2] [3]. Una CLF es una candidata a ser Función de Lyapunov (*Lyapunov Function* o LF), gracias a que su derivada puede hacerse negativa mediante la elección idónea de la entrada de control. La existencia de una CLF es suficiente (y necesaria) para estabilizar asintóticamente a un sistema nominal [4] [5] [6] [7]. El diseño de estrategias de control para sistemas Hamiltonianos es un tema de alto interés de investigación. La teoría de Lyapunov y la obtención de CLF's, en conjunto con las ecuaciones de Hamilton como método para el modelado de sistemas, pueden conferir ventajas en aplicaciones

de control automático. Se han reportado algunos trabajos de investigación muy interesantes referentes al modelado Hamiltoniano de sistemas y el Control basado en Lyapunov. En [8] y [9] se implementan Funciones de Lyapunov de Control del tipo barrera y cuadrática, respectivamente, para el control y estabilización de sistemas de orden superior. En [10] se consideran las propiedades de sistemas Hamiltonianos y su aplicación al control de actuadores saturados. Más aun, la teoría Hamiltoniana ha generado recientemente ventajas para el modelado de sistemas y diseño de sistemas de control en lazo cerrado a partir de términos de energía propia del sistema. En [11] se establece una metodología basada en la interconexión integral e inyección de amortiguamiento basada en el control por pasividad, y se aplica en sistemas subactuados considerando perturbaciones acotadas y no acotadas, mientras que en [12] también se utiliza una metodología basada en sistemas pasivos interconectados sin la necesidad de resolver las ecuaciones diferenciales representativas para el modelado de la dinámica del sistema a controlar. En este trabajo de investigación, se aporta un método sistemático y funcional de construcción de CLF's, que permitan diseñar esquemas de control estables sin necesidad de realizar alguna prueba de estabilidad extra, siendo lo mencionado una ventaja respecto a otros procedimientos de diseño de controladores. Dicho proceso se puede resumir en las siguientes etapas:

- 1) Modelar a la planta a controlar mediante las ecuaciones de Hamilton,
- 2) Obtener una CLF para dicho sistema,
- 3) A partir de la CLF obtenida, proponer una ley de control adecuada.

El procedimiento anteriormente descrito, es implementado para la resolución de tareas de regulación en manipuladores robóticos. Para ello fue necesario extender e implementar un teorema reportado en la literatura, de construcción de CLF's para sistemas de una entrada y una salida, a sistemas de múltiples entradas y múltiples salidas. A continuación, se muestra el procedimiento de obtención del modelo Hamiltoniano de manipuladores robóticos.

## II. MODELADO HAMILTONIANO

La formulación Hamiltoniana se caracteriza por la ausencia de ecuaciones de restricción entre las coordenadas. Las ecuaciones de Hamilton tienen como propósito describir el movimiento mediante  $2n$  ecuaciones diferenciales ordinarias de primer orden, donde  $n$  son los grados de libertad del sistema, las cuales reemplazan a las  $n$  ecuaciones diferenciales ordinarias de segundo orden de Lagrange [14] [15]. En mecánica clásica, la transformada de Legendre se usa para obtener el modelado Hamiltoniano de un sistema partiendo del Lagrangiano. En esta sección se muestra a detalle la obtención del modelo de un robot manipulador de 2 grados de libertad.

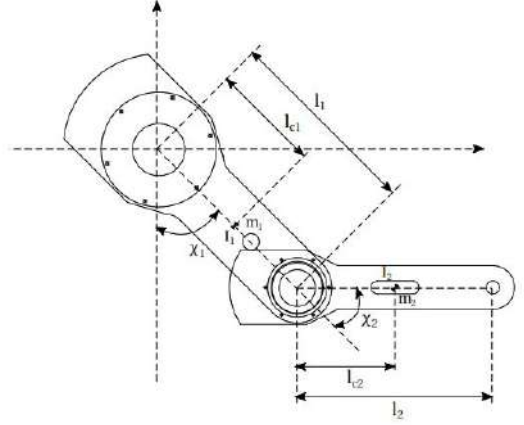


Fig. 1. Robot manipulador CICESE de dos grados de libertad

### A. Ecuaciones de Hamilton

El uso de las ecuaciones de Hamilton es un método que permite obtener la dinámica de un sistema mecánico [13]. Es posible obtener dichas ecuaciones partiendo del Lagrangiano  $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ , el cuál está definido por:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q}), \quad (1)$$

donde  $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}})$  y  $\mathcal{U}(\mathbf{q})$  representan la energía cinética y potencial del sistema, mientras que  $\mathbf{q}$  y  $\dot{\mathbf{q}}$  representan a los vectores de posiciones y velocidades articulares, respectivamente.

Para el modelado Hamiltoniano, se define el vector de estados  $\mathbf{x} = [\mathbf{q} \ \mathbf{p}]^T$ , donde  $\mathbf{p}$  resulta ser un vector de momentos conjugados dado por:

$$\mathbf{p} = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}. \quad (2)$$

El Hamiltoniano de un sistema puede obtenerse de:

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) = \dot{\mathbf{q}}^T \mathbf{p} - \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}), \quad (3)$$

a partir del cual pueden ser obtenidas las ecuaciones de movimiento de Hamilton dadas por:

$$\dot{\mathbf{q}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}, \quad (4)$$

$$\dot{\mathbf{p}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} + \boldsymbol{\tau}. \quad (5)$$

### B. Modelo Hamiltoniano de Robots Manipuladores

Un robot manipulador es un sistema mecánico complejo cuya descripción analítica requiere de ecuaciones diferenciales no lineales. En este trabajo se consideran robots manipuladores conformados por eslabones rígidos conectados mediante articulaciones libres de fricción y elasticidad. En esta sección se muestra el modelo dinámico de un robot manipulador de 2 grados de libertad (Figura 1) cuyos parámetros experimentales se muestran en la Tabla I [18].

Tabla I  
PARÁMETROS DEL ROBOT MANIPULADOR

Notación	Significado	Valor	Unidades
$x_1$	Posición angular del eslabón 1	-	[rad]
$x_2$	Posición angular del eslabón 2	-	[rad]
$l_1$	Longitud del eslabón 1	0.45	Metros [m]
$l_2$	Longitud del eslabón 2	0.45	Metros [m]
$l_{c1}$	Centro de masa del eslabón 1	0.091	Metros [m]
$l_{c2}$	Centro de masa del eslabón 2	0.048	Metros [m]
$m_1$	Masa del eslabón 1	23.902	Kilogramos [m]
$m_2$	Masa del eslabón 2	3.88	Kilogramos [m]
$I_1$	Inercia del eslabón 1	1.266	[kgm <sup>2</sup> /rad]
$I_2$	Inercia del eslabón 2	0.093	[kgm <sup>2</sup> /rad]

El lagrangiano del robot manipulador, libre de fricción, es:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} - \mathcal{U}(\mathbf{q}). \quad (6)$$

donde  $M(\mathbf{q})$  representa la matriz de inercias compuesta por:

$$M(\mathbf{q}) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (7)$$

donde los términos de la matriz de inercia, a partir de los parámetros del robot, resultan ser:

$$M_{11}(\mathbf{q}) = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_1 + I_2,$$

$$M_{12}(\mathbf{q}) = m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_2,$$

$$M_{21}(\mathbf{q}) = m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_2,$$

$$M_{22}(\mathbf{q}) = m_2 l_{c2}^2 + I_2.$$

A partir del lagrangiano (6) y resolviendo (2) y (3), pueden obtenerse el momento conjugado y el Hamiltoniano del sistema, respectivamente:

$$\mathbf{p} = M(\mathbf{q}) \dot{\mathbf{q}}, \quad (8)$$

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^T M(\mathbf{q})^{-1} \mathbf{p} + \mathcal{U}(\mathbf{q}). \quad (9)$$

Resolviendo (4) y (5) pueden ser obtenidas las ecuaciones de movimiento de Hamilton para el robot manipulador de dos grados de libertad:

$$\dot{\mathbf{q}} = M(\mathbf{q})^{-1}\mathbf{p}, \quad (10)$$

$$\dot{\mathbf{p}} = -\frac{\partial}{\partial \mathbf{q}} \left( \frac{1}{2}\mathbf{p}^T M(\mathbf{q})^{-1}\mathbf{p} + \mathcal{U}(\mathbf{q}) \right) + \boldsymbol{\tau}. \quad (11)$$

Finalmente, las ecuaciones de movimiento de Hamilton, en espacio de estados del robot manipulador, resultan ser [18]:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} M(\mathbf{q})^{-1}\mathbf{p} \\ -\frac{\partial}{\partial \mathbf{q}} \left( \frac{1}{2}\mathbf{p}^T M(\mathbf{q})^{-1}\mathbf{p} \right) + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau} \end{bmatrix} \quad (12)$$

donde  $\mathbf{g}(\mathbf{q})$  representa el vector de pares gravitacionales dado por:

$$\begin{aligned} g_1(\mathbf{q}) &= (m_1 l_{c1} + m_2 l_1)g \sin(q_1) + m_2 l_{c2} g \sin(q_1 + q_2) \\ g_2(\mathbf{q}) &= m_2 l_{c2} g \sin(q_1 + q_2). \end{aligned}$$

### III. CONSTRUCCIÓN DE FUNCIONES DE LYAPUNOV DE CONTROL

En esta sección se muestra un procedimiento mediante el cual se puede obtener una CLF que permita proponer esquemas de control para tareas de regulación, considerando al robot manipulador como caso de estudio. La obtención de este tipo de reguladores diseñados a partir de la construcción de una CLF adecuada permite asegurar la estabilidad asintótica del sistema.

Sea un sistema no lineal descrito por [19]:

$$\dot{\mathbf{x}} = F(\mathbf{x}) + G(\mathbf{x})\mathbf{u} \quad (13)$$

donde  $\mathbf{x} \in \mathbb{R}^n$  son los estados del sistema y  $\mathbf{u} \in \mathbb{R}$  es la entrada de control. Se asume que  $F(\mathbf{x})$  y  $G(\mathbf{x})$  pueden ser reestructuradas en la forma:

$$F(\mathbf{x}) = \begin{bmatrix} \phi_{11} & \phi_{12} & 0 & \cdots & 0 \\ \phi_{21} & \phi_{22} & \phi_{23} & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & 0 \\ \phi_{n-1,1} & \phi_{n-1,2} & \phi_{n-1,3} & \cdots & \phi_{n-1,n} \\ \phi_{n1} & \phi_{n2} & \phi_{n3} & \cdots & \phi_{nn} \end{bmatrix} \mathbf{x}, \quad (14)$$

$$G(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \phi_{n,n+1} \end{bmatrix}, \quad (15)$$

para funciones escalares continuas  $\phi_{ij}(t, \mathbf{x})$  satisfaciendo:

$$\phi_{i,j}(\mathbf{x}) = \phi_{i,j}(x_1, \dots, x_i), \quad (16)$$

para  $1 \leq i \leq n$  y  $1 \leq j \leq i + 1$ . Además se debe cumplir:

$$\phi_{i,i+1}(x_1, \dots, x_i) \neq 0, \quad (17)$$

para todo  $x_1, \dots, x_i \in \mathbb{R}$  con  $1 \leq i \leq n$ . La descomposición de (13) puede no ser única. Haciendo uso del Teorema 1 reportado en [19], es posible construir funciones  $s_i$  tal que  $V(\mathbf{x}) = \mathbf{z}^T \mathbf{z}$  será una función de control de Lyapunov para

sistemas dados en (13).

**Teorema 1** [19] Si el sistema (13)-(15) satisface (16) y (17), entonces existen funciones  $s_i$  tal que  $V(\mathbf{x}) = \mathbf{z}^T \mathbf{z}$  es una CLF para el sistema (13)-(15), donde  $\mathbf{z}$  es un vector de transformación en el espacio de estados  $\mathbb{R}^n$ , que se construye usando funciones escalares suaves denotadas por  $s_1(x_1), s_2(x_1, x_2), \dots, s_{n-1}(x_1, \dots, x_{n-1})$ .

$$\begin{aligned} z_1 &= x_1, \\ z_2 &= x_2 - z_1 s_1(x_1), \\ z_3 &= x_3 - z_2 s_2(x_1, x_2), \\ &\vdots \\ z_n &= x_n - z_{n-1} s_{n-1}(x_1, \dots, x_{n-1}). \end{aligned} \quad (18)$$

Cada una de estas funciones dependerán de los componentes de los estados del sistema  $x_1$  hasta  $x_i$ .

Es necesario asegurar que  $\mathbf{z}$  sea un difeomorfismo, asegurando que las funciones  $s_i$  sean biyectivas, diferenciables y posean una inversa  $S^{-1}$  que mantenga invariante el conjunto frontera resultado de cada transformación. En forma matricial, el vector de transformación  $\mathbf{z}$  se puede representar mediante su descomposición en una matriz  $S$  que es multiplicada por el vector de estados  $\mathbf{x}$ :

$$\mathbf{z} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -s_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \pm [s_1 s_2 \cdots s_{n-1}] & \pm [s_2 \cdots s_{n-1}] & \cdots & 1 \end{bmatrix} \mathbf{x}, \quad (19)$$

El inverso de (19) resulta ser:

$$\mathbf{x} = S^{-1}(\mathbf{x})\mathbf{z} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ s_1 & 1 & 0 & \cdots & 0 \\ 0 & s_2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \mathbf{z}. \quad (20)$$

Sustituyendo (20) en la representación en espacio de estados de (13), se obtiene:

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} &= \begin{bmatrix} \phi_{11} & \phi_{12} & 0 & \cdots & 0 \\ \phi_{21} & \phi_{22} & \phi_{23} & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & 0 \\ \phi_{n-1,1} & \phi_{n-1,2} & \phi_{n-1,3} & \cdots & \phi_{n-1,n} \\ \phi_{n1} & \phi_{n2} & \phi_{n3} & \cdots & \phi_{nn} \end{bmatrix} \\ &\quad \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ s_1 & 1 & 0 & \cdots & 0 \\ 0 & s_2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \phi_{n,n+1} \end{bmatrix} \mathbf{u} \end{aligned} \quad (21)$$

Es imprescindible obtener una representación en espacio de estados del sistema en función del vector de transformación

$z$ , es decir, obtener  $\dot{z}$ . Para ello, tomando la derivada de  $z = S(x)x$  en (19) tenemos que

$$\dot{z} = \dot{S}(x)x + S(x)\dot{x}. \quad (22)$$

expresión que puede ser reestructurada en la forma:

$$\begin{aligned} &= \Delta S(x)\dot{x} + S(x)\dot{x}, \\ &= [\Delta S(x) + S(x)]\dot{x}, \\ &= T(x)\dot{x}, \end{aligned} \quad (23)$$

entonces se tiene:

$$\dot{z} = T(x)\dot{x} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \Delta_1 & 1 & 0 & \cdots & 0 \\ \Delta_2 & \Delta_2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta_{n-1} & \Delta_{n-1} & \Delta_{n-1} & \cdots & 1 \end{bmatrix} \dot{x}. \quad (24)$$

Sustituyendo  $\dot{x}$  de (21) en (23), se obtiene la nueva representación en espacio de estados del sistema (13) en función del vector de transformación  $z$ :

$$\begin{aligned} \dot{z} &= T(x) \underbrace{\begin{bmatrix} \phi_{11} + \phi_{12}s_1 & \phi_{12} & \cdots & 0 \\ \phi_{21} + \phi_{22}s_1 & \phi_{22} + \phi_{23}s_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 \\ \phi_{n1} + \phi_{n2}s_1 & \phi_{n2} + \phi_{n3}s_2 & \cdots & \phi_{nn} \end{bmatrix}}_{A(x)} z \\ &+ T(x) \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \phi_{n,n+1} \end{bmatrix} u. \end{aligned} \quad (25)$$

El diseño de las funciones  $s_i$  debe asegurar que el primer elemento de la matriz  $A(x)$  de (25) es definido negativo, es decir,

$$\phi_{11} + \phi_{12}s_1 < 0. \quad (26)$$

A partir del Teorema 1, se sabe que una CLF adecuada para sistemas dados por (13) estará dada por  $V(x) = z^T z$ , siendo su derivada respecto al tiempo [19]:

$$\dot{V} = z^T [A(x) + A^T(x)] z + 2u\phi_{n,n+1}(x)z_n. \quad (27)$$

La obtención de una CLF adecuada para sistemas del tipo (13) permite establecer condiciones para la construcción de leyes de control que aseguren la estabilidad asintótica de sistemas de control bajo el enfoque de casos específicos. A continuación se ejemplifica la construcción de una CLF adecuada que permiten obtener esquemas de control para solventar tareas de regulación considerando un robot manipulador de dos grados de libertad.

#### IV. CONSTRUCCIÓN DE UNA CLF PARA EL CONTROL DE POSICIÓN DE UN ROBOT MANIPULADOR

Se retoma al robot manipulador de dos grados de libertad, cuyas ecuaciones de Hamilton están dadas por:

$$\dot{q} = M(q)^{-1}p \quad (28)$$

$$\dot{p} = -\frac{\partial}{\partial q} \left( \frac{1}{2}p^T M(q)^{-1}p \right) + g(q) + \tau \quad (29)$$

donde  $M(q)$  es la matriz de inercias de dimensión  $n \times n$ ,  $g(q)$  es el vector de pares gravitacionales de orden  $n$ ,  $\tau$  es el vector de  $n$  pares aplicados al robot,  $q$  denota el vector de posiciones angulares y  $p$  el vector de momentos conjugados.

Se define un vector de estados  $x = [\tilde{q}^T \ \dot{p}^T]^T$ , donde  $\tilde{q} = q_d - q$  es el vector errores de posición y  $p$  el vector de momentos conjugados del sistema. El sistema (28)-(29) en su representación en espacio de estados, considerando  $\tau = \frac{\partial}{\partial q} \left( \frac{1}{2}p^T M(q)^{-1}p \right) + g(q) + u$  como ley de control preliminar, resulta ser:

$$\begin{bmatrix} \dot{\tilde{q}} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0_{n \times n} & -M(q)^{-1} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix} \begin{bmatrix} \tilde{q} \\ p \end{bmatrix} + \begin{bmatrix} 0_{n \times n} \\ I_{n \times n} \end{bmatrix} u. \quad (30)$$

A partir de la representación en espacio de estados del sistema, en la forma indicada en (14) y (15), se sabe que  $\Phi_{11} = \Phi_{13} = \Phi_{21} = \Phi_{22} = 0_{n \times n}$ ,  $\Phi_{12} = -M(q)^{-1}$  y  $\Phi_{23} = I_{n \times n}$ .

El vector de transformación  $z$  correspondiente al sistema resulta ser:

$$\begin{aligned} z_1 &= \tilde{q}, \\ z_2 &= p - \tilde{q}S_1(\tilde{q}), \end{aligned} \quad (31)$$

donde  $S_1(\tilde{q}) \in \mathbb{R}^{n \times n}$  y  $\tilde{q} \in \mathbb{R}^n$ , siendo su forma matricial:

$$z = Sx = \begin{bmatrix} I_{n \times n} & 0_{n \times n} \\ -S_1(\tilde{q}) & I_{n \times n} \end{bmatrix} \begin{bmatrix} \tilde{q} \\ p \end{bmatrix}, \quad (32)$$

y su inverso:

$$\begin{bmatrix} \tilde{q} \\ p \end{bmatrix} = S^{-1}z = \begin{bmatrix} I_{n \times n} & 0_{n \times n} \\ S_1(\tilde{q}) & I_{n \times n} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \quad (33)$$

donde  $S \in \mathbb{R}^{2n \times 2n}$ ,  $x = [x_1 \ x_2]^T \in \mathbb{R}^{2n}$  con  $x_1 = \tilde{q}$ ,  $x_2 = \dot{p}$  y  $S_1(\tilde{q})$  es una matriz diagonal de orden  $n \times n$  cuyos elementos son  $s_i(\tilde{q}_i)$  con  $i = 1, 2, \dots, n$ . Sustituyendo (33) en (30), obtenemos:

$$\begin{bmatrix} \dot{\tilde{q}} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} -M(q)^{-1}S_1 & -M(q)^{-1} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0_{n \times n} \\ I_{n \times n} \end{bmatrix} u. \quad (34)$$

Es necesario obtener la representación en espacio de estados del sistema (28)-(29) en función del vector de transformación  $z$ . A partir de (31) se sabe que:

$$\dot{z} = T(x)\dot{x}, \quad (35)$$

siendo

$$T(x) = \begin{bmatrix} I_{n \times n} & 0_{n \times n} \\ \Delta_1 & I_{n \times n} \end{bmatrix}, \quad (36)$$

donde

$$\Delta_1 = -\frac{\partial(S_1(\tilde{\mathbf{q}}))}{\partial(\tilde{\mathbf{q}})} \begin{bmatrix} \tilde{q}_1 & 0 & \cdots & 0 \\ 0 & \tilde{q}_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 \\ 0 & 0 & \cdots & \tilde{q}_n \end{bmatrix} - S_1(\tilde{\mathbf{q}}), \quad (37)$$

y con un cierto abuso de notación,  $\frac{\partial(S_1(\tilde{\mathbf{q}}))}{\partial(\tilde{\mathbf{q}})}$  denota una matriz diagonal de orden  $n \times n$  cuyos elementos son sus respectivas derivadas parciales; es decir, denota a la matriz:

$$\frac{\partial(S_1(\tilde{\mathbf{q}}))}{\partial(\tilde{\mathbf{q}})} = \begin{bmatrix} \frac{\partial(s_1(\tilde{q}_1))}{\partial(\tilde{q}_1)} & 0 & \cdots & 0 \\ 0 & \frac{\partial(s_2(\tilde{q}_2))}{\partial(\tilde{q}_2)} & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 \\ 0 & 0 & \cdots & \frac{\partial(s_n(\tilde{q}_n))}{\partial(\tilde{q}_n)} \end{bmatrix}. \quad (38)$$

Nótese que  $\Delta_1$  es una matriz diagonal de orden  $n \times n$  de funciones dependiente del estado  $\tilde{\mathbf{q}}$ , de la matriz diagonal  $S_1(\tilde{\mathbf{q}})$  con elementos  $s_i(\tilde{q}_i)$ , con  $i = 1, 2, \dots, n$  y de sus derivadas parciales. Resolviendo (35) con (30), se obtiene:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -M(\mathbf{q})^{-1}S_1 & -M(\mathbf{q})^{-1} \\ -\Delta_1 M(\mathbf{q})^{-1}S_1 & -\Delta_1 M(\mathbf{q})^{-1} \end{bmatrix}}_{A(\tilde{\mathbf{q}}, \mathbf{p})} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u}. \quad (39)$$

En base al Teorema 1, una CLF adecuada para el sistema (28)-(29) resulta ser:

$$V(\tilde{\mathbf{q}}, \mathbf{p}) = \tilde{\mathbf{q}}^2 + (\mathbf{p} - \tilde{\mathbf{q}}S_1)^2, \quad (40)$$

siendo la derivada temporal a lo largo de las trayectorias del sistema:

$$\dot{V}(\tilde{\mathbf{q}}, \mathbf{p}) = \mathbf{z}^T \begin{bmatrix} -2M(\mathbf{q})^{-1}S_1 & \mu \\ \mu & -2\Delta_1 M(\mathbf{q})^{-1} \end{bmatrix} \mathbf{z} - 2\mathbf{u}(\mathbf{p} - \tilde{\mathbf{q}}S_1). \quad (41)$$

donde  $\mu = -\Delta_1 M(\mathbf{q})^{-1}S_1 - M(\mathbf{q})^{-1}$ .

Es necesario elegir funciones  $S_i(\tilde{\mathbf{q}})$  que aseguren que el primer elemento de (41) sea definido negativo y con ello poder construir una ley de control nominal adecuada. Primero, se obtienen las condiciones necesarias de  $S_1(\tilde{\mathbf{q}})$  a partir de:

$$\begin{aligned} -2S_1(\tilde{\mathbf{q}}) &< 0, \\ S_1(\tilde{\mathbf{q}}) &= \text{diag}(k_i), \end{aligned} \quad (42)$$

siendo  $k_i$  constantes positivas para  $i = 1, 2, \dots, n$ . Sustituyendo en (37), se obtiene  $\Delta_1 = -\text{diag}(k_i)$  con  $k_i$  siendo constantes positivas para  $i = 1, 2, \dots, n$ .

Una ley de control nominal  $\mathbf{u}$  adecuada para el sistema (28)-(29) puede construirse a partir del último término de la ecuación (41), buscando que dicho término sea definido negativo, con lo cual se propone:

$$\mathbf{u} = -K_v(\mathbf{p} - \tilde{\mathbf{q}}S_1) \quad (43)$$

siendo  $K_v$  una matriz diagonal con elementos positivos. Nótese que la ley de control (43) debe ser complementada con

la ley de control preliminar  $\boldsymbol{\tau}$ , considerada en (30), resultando finalmente:

$$\boldsymbol{\tau} = -K_v(\mathbf{p} - \tilde{\mathbf{q}}S_1) + \frac{\partial}{\partial \mathbf{q}} \left( \frac{1}{2} \mathbf{p}^T M(\mathbf{q})^{-1} \mathbf{p} \right) + \mathbf{g}(\mathbf{q}). \quad (44)$$

La correcta elección de (43) y (42), permite que (41) sea definida negativa, y con ello se asegura la estabilidad asintótica del origen del sistema de control.

## V. RESULTADOS EN SIMULACIÓN

En esta sección, se muestran los resultados obtenidos mediante el controlador propuesto (Control CLF) y su comparación respecto a un control Proporcional-Integral-Derivativo (PID). Se observa como la posición deseada es alcanzada por el sistema de control en ambos eslabones del robot manipulador, y cómo los errores de posición tienden asintóticamente a cero. En cada una de de las pruebas se logró el objetivo sin sobrepasar los límites de pares de entrada de los motores que conforman al manipulador robótico (150 Nm y 20 Nm).

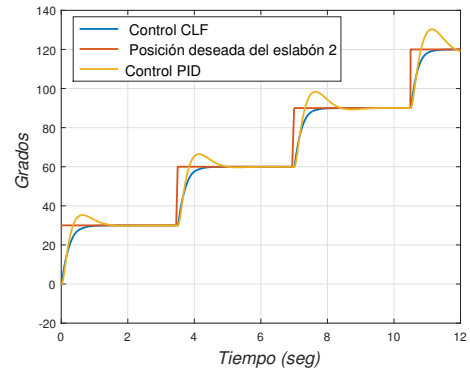


Fig. 2. Posición del eslabón 1

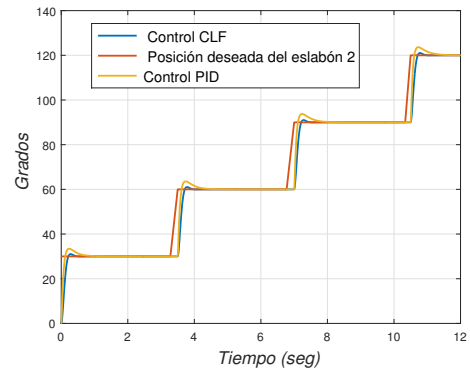


Fig. 3. Posición del eslabón 2

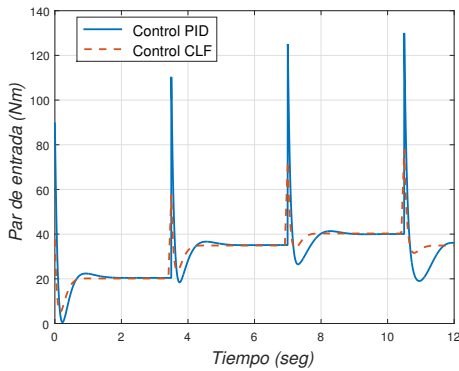


Fig. 4. Par de entrada del eslabón 1

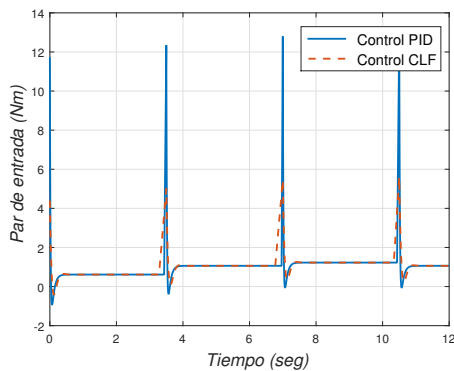


Fig. 5. Par de entrada del eslabón 2

## VI. CONCLUSIONES

Fue implementado un método sistemático de diseño de esquemas de control nominal, en función de la obtención adecuada de CLF para sistemas no lineales de múltiples-entradas múltiples-salidas. Para ello, se hace uso de un vector de transformación  $z$  que permita manipular parámetros de la derivada de la CLF propuesta y con ello asegurar la estabilidad asintótica del sistema. Se busca que estos esquemas de control estabilicen al sistema despreciando la presencia de incertidumbres paramétricas y perturbaciones externas, lo cual podría ser solucionado robustificando los controladores propuestos mediante técnicas de rediseño de Lyapunov. Lo anteriormente mencionado, fue ejemplificado mediante su aplicación teórica para la resolución de tareas de regulación en un manipulador robótico de 2 grados de libertad. Mediante simulaciones numéricas se validó el funcionamiento de la metodología para la construcción de CLF, así como el desempeño de los controladores propuestos. Se establece como trabajos futuros, ampliarlas técnicas de este artículo de investigación a sistemas subactuados, además de considerar el efecto de perturbaciones externas y cambios paramétricos en el sistema de control en lazo cerrado.

## AGRADECIMIENTOS

Los resultados reportados en este artículo fueron obtenidos gracias al apoyo recibido por parte del Instituto Tecnológico

de La Laguna y el Consejo Nacional de Ciencia y Tecnología CONACyT. Especial agradecimiento a dichas instituciones.

## REFERENCIAS

- [1] H.K. Khalil, *Nonlinear systems*, 2nd edition. Prentice Hall, 1996.
- [2] J. Espinoza, "Introducción al análisis de sistemas no lineales", Universidad de Concepción Facultad de Ingeniería, Departamento de Ingeniería Eléctrica.
- [3] G. Şahan and D. Özdemir, "Uniform Asymptotic Stability by Indefinite Lyapunov Functions". *13th Asian Control Conference (ASCC)*, 2022, pp. 1771-1774, doi: 10.23919/ASCC56756.2022.9828046.
- [4] A. Bacciotti y L. Rosier, "Lyapunov functions and stability in control theory", Springer-Verlag, New York, 2nd edition, 2005.
- [5] A. Polyakov y L. Fridman, "Stability notions and Lyapunov functions of sliding mode control systems", *Journal of the Franklin Institute*, 2014.
- [6] Z. Arstein, (1983). "Stabilization with relaxed controls", *Nonlinear analysis*, Vol. 7, pp. 1163-1173, 1983.
- [7] Sontag, E. (1983). "A Lyapunov-like characterization of asymptotic controllability", *SIAM J. Control and Optimization*, Vol. 21, pp. 462-471, 1983.
- [8] N. Sedova and O. Druzhinina, "Stabilization of State Constrained Delayed Systems Using Control-Lyapunov and Control-Barrier Functions," 2022 16th International Conference on Stability and Oscillations of Nonlinear Control Systems (Pyatnitskiy's Conference), Moscow, Russian Federation, 2022, pp. 1-4, doi: 10.1109/STAB54858.2022.9807608.
- [9] Z. Liu, Q. Li, Y. Chen, W. Zhang, Y. Cui and Y. Zhou, "A Quadratic Lyapunov Function-Based Control Scheme for a Class of High-Order Nonlinear Systems," 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 2023, pp. 803-808, doi: 10.23919/CCC58697.2023.10240192.
- [10] W. Jianwen and S. Liying, "Stabilizing of Nonlinear Singular Hamiltonian Systems with Actuator Saturation," 2024 Second International Conference on Cyber-Energy Systems and Intelligent Energy (ICCSIE), Shenyang, China, 2024, pp. 1-5, doi: 10.1109/ICCSIE61360.2024.10698026.
- [11] E. Franco, P. Arpentí, A. Donaire and F. Ruggiero, "Integral IDA-PBC for Underactuated Mechanical Systems Subject to Matched and Unmatched Disturbances," in *IEEE Control Systems Letters*, vol. 8, pp. 568-573, 2024, doi: 10.1109/LCSYS.2024.3399474
- [12] P. Borja, "Interconnection and Damping Assignment Passivity-Based Control Without Partial Differential Equations," 2024 UKACC 14th International Conference on Control (CONTROL), Winchester, United Kingdom, 2024, pp. 131-136, doi: 10.1109/CONTROL60310.2024.10532055.
- [13] H. Goldstein, P. Charles y S. John, *Classical mechanics*, Third edition, 2000.
- [14] Y. Kawano y T. Ohtsuka, "An algebraic approach to Hamilton's canonical equations", *SICE Annual Conference*, pp. 370-375, 2011.
- [15] R. Maheshwari, D. Patil y S. Dwivedi, "Port-Controlled Hamilton Model and Control of Induction Machine with Inverter Output LC Filter", *1st International Conference on Power Electronics and Energy (ICPEE)*, pp. 1-4, doi: 10.1109/ICPEE50452.2021.9358543, 2021.
- [16] B. Deng, Y. Qin, Y. Li, H. Wang y X. Li, "A Novel Approach to Range Doppler SAR Processing Based on Legendre Orthogonal Polynomials", *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 1, pp. 13-17, doi: 10.1109/LGRS.2008.2006572, 2009.
- [17] Guoqi Li y Changyun Wen, "Legendre polynomials in signal reconstruction and compression", *5th IEEE Conference on Industrial Electronics and Applications*, pp. 1636-1640, doi: 10.1109/ICIEA.2010.5514776, 2010.
- [18] R. Kelly y V. Santibáñez, "Control de Movimiento de Robots Manipuladores". Pearson educación, 2003.
- [19] X. Cai, Q. Zhang y R. Lin, "Constructive Control Lyapunov Functions for a Class of Nonlinear Systems", *International Conference on Neural Networks and Brain*, 2005.

# Control No Lineal de un Convertidor Boost Cuadrático con Procesamiento de Potencia Redundante Reducida

Cuauhtémoc Guerrero

Universidad Autónoma de la Ciudad de México  
Ciudad de México, México  
cuauhtemoc.guerrero@uacm.edu.mx

Víctor Santibañez

Tecnológico Nacional de México/  
Instituto Tecnológico de la Laguna  
Torreón, Coahuila, México  
vsantiba@itlalaguna.edu.mx

José Luis Meza-Medina

Tecnológico Nacional de México/  
Instituto Tecnológico de la Laguna  
Torreón, Coahuila, México  
jlmezam@correo.itlalaguna.edu.mx

**Resumen**—La topología del convertidor boost con el principio de procesamiento de potencia redundante reducida es una alternativa de carga para dispositivos eléctricos. En este trabajo se presenta un controlador para tener un seguimiento exponencial del voltaje de la carga. El diseño de dicho controlador se hace teniendo como base el modelo matemático promedio no lineal que se ha obtenido del convertidor. Se presenta una prueba formal del análisis de estabilidad en lazo cerrado del sistema conformado por el convertidor y el controlador.

**Index Terms**—Quadratic Boost Converter, Reduced Redundant Power Processing  $R^2P^2$ , Modelo promedio.

## I. INTRODUCCIÓN

Los convertidores electrónicos de potencia de corriente directa a corriente directa (DC-DC, por sus siglas en inglés) tienen en la actualidad un gran auge de uso como lo es en aplicaciones industriales, el sector automotriz o en carros eléctricos [1], también en fuentes de energía renovable o sistemas de potencia móviles [2], [3]. Los convertidores DC-DC que se utilizan en las fuentes renovables, tales como generadores fotovoltaicos o eólicos, requieren que tengan una amplia conversión de voltaje y eficiencia mejorada [4].

Los interruptores semiconductores de potencia que se utilizan en los convertidores DC-DC tradicionales deben de trabajar a frecuencias de conmutación máxima, sin embargo, están restringidos por el tiempo limitado de respuesta que tienen [1]. Esto conlleva a que las topologías básicas de este tipo de sistemas tengan limitaciones. Por tal motivo, en los últimos años se han desarrollado otras topologías que sobrellevan dichas limitaciones. En particular, el convertidor boost cuadrático es de los más utilizados ya que reduce el número de elementos requeridos del circuito de dicho convertidor [2] y la ganancia de voltaje contiene un término cuadrático por lo que se espera tener una mayor ganancia para el mismo ciclo de trabajo [5]. Un tipo de convertidor boost DC-DC que utiliza el principio de que la transferencia de potencia es reducida desde el puerto de entrada al puerto de salida sobre convertidores interconectados cuando se utiliza una conexión que no está en cascada, se les denomina convertidor boost cuadrático basado en el principio de procesamiento de potencia

redundante reducida ( $R^2P^2$ , por sus siglas en inglés) [6]. Los convertidores con  $R^2P^2$  son en la actualidad de mucho interés en investigadores de electrónica de potencia que se ha realizado análisis teórico en cuanto a su funcionamiento [7].

El  $R^2P^2$  presentado en [2], es de un sólo interruptor, conversión de voltaje cuadrático y, desde un punto de vista teórico, más eficiente y tiene las siguientes ventajas. Sin circuitos adicionales, la energía almacenada en el inductor se envía a la carga proporcionando una ganancia de voltaje para algún convertidor tradicional utilizado como elevador de alta ganancia [1]. Dicho convertidor está compuesto por dos convertidores con un funcionamiento en modo de conducción continua (CCM, por sus siglas en inglés), el primero es de tipo buck-boost y el segundo es de tipo boost. Cabe hacer mención que la técnica  $R^2P^2$  ha sido empleada de forma exitosa en sistemas de potencia para desempeñar regulación de voltaje activo en redes de distribución, como se menciona en [8], aunque los autores lo denominan técnica de regulación serie.

El convertidor cuadrático con  $R^2P^2$  ha sido utilizado también con algunas variantes para obtener ciertas ventajas extras a las que ya presenta. Por ejemplo, en [5] proponen una topología de este tipo de convertidor, así como la definición de funciones de conmutación para tener un amplio rango de operación de voltaje y bajo rizo en la corriente de entrada. Mientras que en [9], por medio de un circuito auxiliar en el convertidor cuadrático, reducen pérdidas de la potencia total. Hay trabajos que se han desarrollado basados en  $R^2P^2$ ; como es el caso de la propuesta presentada en [10] donde los autores diseñan un convertidor Boost-Boost  $R^2P^2$  basados en concepto cuasi-resonante usando técnicas de conmutación suave; o el trabajo de [11] muestran una implementación de un convertidor buck cuadrático utilizando los conceptos  $R^2P^2$  para aplicaciones fotovoltaicas; asimismo, en [12] utiliza un convertidor con  $R^2P^2$  para satisfacer altos niveles de corrientes y bajos voltajes en sistemas fotovoltaicos. Cabe hacer mención que, el modelado y control de convertidores electrónicos ayudará a incluir, en el modelo dinámico de sistemas robóticos, el modelo de los actuadores y sus drives electrónicos de potencia y comparar su comportamiento usando una dinámica

completa de todos los elementos que conforman los sistemas robóticos.

En [2], además de presentar el modelo matemático del convertidor boost cuadrático con  $R^2P^2$ , diseñan un controlador basado en técnicas de control lineal en el dominio de la frecuencia, realizando una linealización del modelo promediado del convertidor utilizando como variable de retroalimentación la corriente de conmutación. En este trabajo, se presenta una estrategia de control del modelo promedio no lineal del convertidor boost cuadrático con  $R^2P^2$  propuesto en [2] aprovechando las ventajas que presenta dicho modelo al escribirlo en una estructura tipo Euler-Lagrange. Para el diseño del controlador, se retroalimentan todas las variables de estado del modelo no lineal del convertidor, considerando que solo se tiene como variable deseada el voltaje de la carga, por lo que se obtienen ecuaciones para obtener el comportamiento de las otras variables para que los errores sean cero y asegurar un seguimiento exponencial del voltaje de salida.

## II. MODELO MATEMÁTICO DEL CONVERTIDOR BOOST CUADRÁTICO CON $R^2P^2$

La topología eléctrica de este tipo de convertidor se puede obtener de dos formas, como se puede observar en la Figura 1: en una se utilizan dos interruptores de potencia, (ver Figura 1a) y en la otra sólo un interruptor, (ver Figura 1b). Ambas topologías, con interruptores ideales, son electrónicamente equivalentes por lo que tienen los mismos modos de funcionamiento [2]. Un modo de funcionamiento es cuando el interruptor de potencia está encendido durante  $t_{on}$  segundos y el otro cuando dicho interruptor está apagado durante  $(1 - D)T_s$  segundos, donde  $D$  es la relación de trabajo (duty ratio) y  $T_s$  el periodo de tiempo de conmutación. La relación de trabajo es definida como  $D = t_{on}/T_s$  mientras que el periodo de tiempo de conmutación como  $T_s = t_{on} + t_{off}$ .

En los dos modos de operación se obtienen dos conjuntos de ecuaciones, los cuales se pueden utilizar para obtener un conjunto de ecuaciones diferenciales por medio del concepto de modelo promediado. Para ello, es necesario tomar el promedio de las variables del convertidor, i.e., el valor de promedio de las corrientes de los inductores  $L_1$  y  $L_2$  así como de los voltajes de los capacitores  $C_1$  y  $C_2$  definidos como  $\langle i_{L1} \rangle$ ,  $\langle i_{L2} \rangle$ ,  $\langle v_{C1} \rangle$  y  $\langle v_{C2} \rangle$ , respectivamente<sup>1</sup>. Con esto, se tienen unas ecuaciones diferenciales promedio como

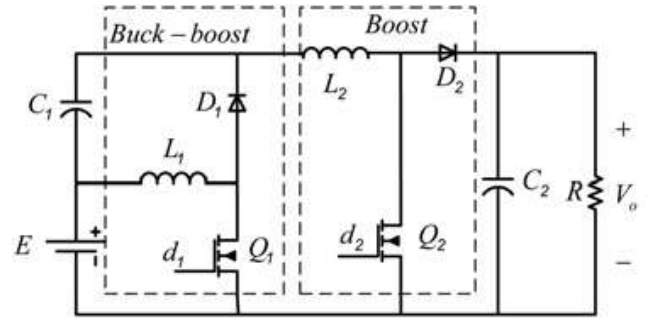
$$\frac{d}{dt} \langle i_{L1} \rangle = \frac{d}{L_1} E - \frac{(1-d)}{L_1} \langle v_{C1} \rangle \quad (1a)$$

$$\frac{d}{dt} \langle i_{L2} \rangle = \frac{1}{L_2} E + \frac{1}{L_2} \langle v_{C1} \rangle - \frac{(1-d)}{L_2} \langle v_{C2} \rangle \quad (1b)$$

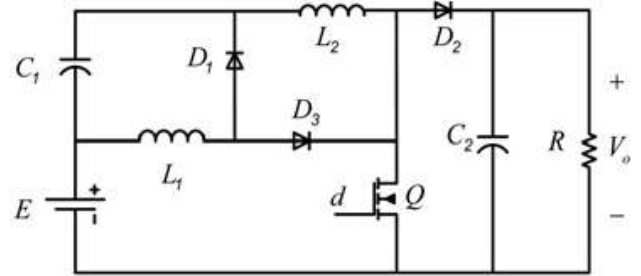
$$\frac{d}{dt} \langle v_{C1} \rangle = \frac{(1-d)}{C_1} \langle i_{L1} \rangle - \frac{1}{C_1} \langle i_{L2} \rangle \quad (1c)$$

$$\frac{d}{dt} \langle v_{C2} \rangle = \frac{(1-d)}{C_2} \langle i_{L2} \rangle - \frac{1}{C_2 R} \langle v_{C2} \rangle \quad (1d)$$

<sup>1</sup>Los autores de [2] utilizan los símbolos  $\langle \cdot \rangle$  en las variables de interés del circuito para indicar que las variables son obtenidas por medio de un proceso de promediado de dos modelos, uno cuando los interruptores de potencia están en conduction y el otro cuando esto están en circuito abierto.



(a) dos interruptores de potencia



(b) un interruptor de potencia

Figura 1: Topologías del convertidor boost cuadrático con  $R^2P^2$  (extraído de [2]).

donde  $d$  es una variable que denota el ciclo de trabajo (duty cycle) variante en el tiempo de la señal de control PWM para el dispositivo semiconductor de potencia.

Si se supone que las variables promedio tienden a ellas mismas, i.e.,  $\langle i_{L1} \rangle \rightarrow i_{L1}$ ,  $\langle i_{L2} \rangle \rightarrow i_{L2}$ ,  $\langle v_{C1} \rangle \rightarrow v_{C1}$  y  $\langle v_{C2} \rangle \rightarrow v_{C2}$ , entonces se puede definir a las corrientes de los dos inductores y a los voltajes de los dos capacitores como los estados del sistema de tal manera que se tiene  $x_1 = i_{L1}$ ,  $x_2 = i_{L2}$ ,  $x_3 = v_{C1}$  y  $x_4 = v_{C2}$  mientras que la fuente de alimentación es  $E$ , las ecuaciones diferenciales (1) se puede reescribir como

$$L_1 \frac{d}{dt} x_1 + (1-d) x_3 = dE \quad (2a)$$

$$L_2 \frac{d}{dt} x_2 - x_3 + (1-d) x_4 = E \quad (2b)$$

$$C_1 \frac{d}{dt} x_3 + x_2 - (1-d) x_1 = 0 \quad (2c)$$

$$C_2 \frac{d}{dt} x_4 - (1-d) x_2 + \frac{1}{R} x_4 = 0 \quad (2d)$$

Se define el vector de estados  $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T \in \mathbb{R}^4$  por lo que en forma matricial el sistema dado en (2) es

$$\mathbf{D} \frac{d}{dt} \mathbf{x} + \mathbf{R} \mathbf{x} + \mathbf{C} \mathbf{x} = \mathbf{q} E \quad (3)$$

donde las matrices  $\mathbf{D} \in \mathbb{R}^{4 \times 4}$ ,  $\mathbf{R} \in \mathbb{R}^{4 \times 4}$  y  $\mathbf{C} \in \mathbb{R}^{4 \times 4}$  están

dadas como

$$\mathcal{D} = \begin{bmatrix} L_1 & 0 & 0 & 0 \\ 0 & L_2 & 0 & 0 \\ 0 & 0 & C_1 & 0 \\ 0 & 0 & 0 & C_2 \end{bmatrix} \quad (4)$$

$$\mathcal{R} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{R} \end{bmatrix} \quad (5)$$

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & (1-d) & 0 \\ 0 & 0 & -1 & (1-d) \\ -(1-d) & 1 & 0 & 0 \\ 0 & -(1-d) & 0 & 0 \end{bmatrix} \quad (6)$$

y el vector de entradas  $\mathbf{q} \in \mathbb{R}^4$  es

$$\mathbf{q} = \begin{bmatrix} d \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (7)$$

### III. RESULTADO PRINCIPAL

#### III-A. Formulación del problema

Considere el modelo obtenido por las ecuaciones diferenciales promediado del convertidor boost cuadrático con  $R^2P^2$  dado en (2). Además suponga que

- S1.** Los parámetros del sistema que conforman el convertidor así como de la carga son constantes y conocidos, todos positivos.
- S2.** Se dispone de la señal de medición de la corriente de los inductores  $L_1$  y  $L_2$ .
- S3.** Se dispone de la señal de medición de los voltajes en los bornes de los capacitores  $C_1$  y  $C_2$ .
- S4.** El voltaje deseado de salida  $V_o$  es una función conocida, suave y acotada con primera derivada.
- S5.** El voltaje de alimentación  $E$  es una función constante y conocida.

Con esto, se desea calcular la relación de trabajo variante en el tiempo  $d$  como señal de control para la conmutación del interruptor de potencia del convertidor boost cuadrático con  $R^2P^2$ , de tal forma que el voltaje en las terminales de la carga tenga un seguimiento exponencial.

#### III-B. Controlador propuesto

La función de la razón de trabajo variante en el tiempo  $d$  se obtiene como

$$d = 1 - \frac{C_2 \dot{x}_{4d} + \frac{1}{R} x_{4d}}{x_{2d}} \quad (8)$$

donde  $x_{2d}$  es la corriente deseada, calculada en el controlador, para el inductor  $L_2$  mientras que  $x_{4d}$  es el voltaje deseado del capacitor  $C_2$  definida de acuerdo a **S4**.

La corriente deseada  $x_{1d}$ , calculada en el controlador, del inductor  $L_1$  se obtiene por medio de una ecuación de estado como

$$\dot{x}_{1d} = \frac{1}{L_1} (dE - (1-d)x_{3d} - K_{IL1}e_1) \quad (9)$$

con  $x_{3d}$  el valor deseado, calculada en el controlador, del voltaje de capacitor  $C_1$ , la variable  $e_1 \in \mathbb{R}$  es el error de la corriente del inductor  $L_1$  definido como  $e_1 = x_{1d} - x_1$  y  $K_{IL1}$  es una constante con la condición

$$K_{IL1} > 0 \quad (10)$$

Asimismo, la corriente deseada  $x_{2d}$ , calculada en el controlador, del inductor  $L_2$  se calcula como

$$\dot{x}_{2d} = \frac{1}{L_2} (E + x_{3d} - (1-d)x_{4d} - K_{IL2}e_2) \quad (11)$$

donde  $e_2$  denota al error de la corriente del inductor  $L_2$ ,  $x_{4d}$  cumple con la suposición **S4** y  $K_{IL2}$  es una constante que cumple la condición

$$K_{IL2} > 0 \quad (12)$$

El voltaje deseado  $x_{3d}$ , calculado en el controlador, del capacitor  $C_1$  se obtiene como

$$\dot{x}_{3d} = \frac{1}{C_1} ((1-d)x_{1d} - x_{2d} - K_{VC1}e_3) \quad (13)$$

donde  $e_3 \in \mathbb{R}$  denota el error del voltaje del capacitor  $C_1$  dado como  $e_3 = x_{3d} - x_3$  y  $K_{VC1}$  es una constante con la condición

$$K_{VC1} > 0 \quad (14)$$

Bajo estas condiciones, el sistema del convertidor boost cuadrático con  $R^2P^2$  tiene un seguimiento asintótico del voltaje del capacitor que está en paralelo a las terminales de la carga.

**Observación.** Las ecuaciones del controlador (8), (9), (11) y (13) se obtienen después de realizar una dinámica en lazo cerrado del sistema dado en (3). Dichas ecuaciones como se proponen aseguran que los errores sean cero y que el voltaje de la carga tenga un seguimiento exponencial, como se explica en la siguiente sección. También cabe hacer mención que, la frase “calculada en el controlador” cuando se menciona como se calcula las corrientes deseadas de los inductores 1 y 2 como el voltaje deseado del capacitor 1 es en el sentido de que se obtienen por medio de las ecuaciones (9), (11) y (13).

### IV. LAZO CERRADO Y DISEÑO DEL CONTROLADOR

#### IV-A. Sistema en lazo cerrado

Se define el vector de errores  $\mathbf{e} = [e_1 \ e_2 \ e_3 \ e_4]^T \in \mathbb{R}^4$  como

$$\mathbf{e} = \mathbf{x}_d - \mathbf{x} \quad (15)$$

donde  $\mathbf{x}_d$  denota al vector de los estados deseados, de forma individual se tiene que los errores de los estados son

$$e_1 = x_{1d} - x_1 \quad (16a)$$

$$e_2 = x_{2d} - x_2 \quad (16b)$$

$$e_3 = x_{3d} - x_3 \quad (16c)$$

$$e_4 = x_{4d} - x_4 \quad (16d)$$

Considerando (3) y la definición del vector de errores (15), se tiene un sistema dinámico de los errores como

$$\mathcal{D}\dot{\mathbf{e}} + \mathcal{R}\mathbf{e} + \mathcal{C}\mathbf{e} = \Psi \quad (17)$$

donde el vector  $\Psi \in \mathbb{R}^4$  es definida como

$$\Psi = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{bmatrix} \quad (18)$$

con los elementos de dicha función  $\Psi$  dados como

$$\psi_1 = -dE + L_1\dot{x}_{1d} + (1-d)x_{3d} \quad (19a)$$

$$\psi_2 = -E + L_2\dot{x}_{2d} - x_{3d} + (1-d)x_{4d} \quad (19b)$$

$$\psi_3 = C_1\dot{x}_{3d} - (1-d)x_{1d} + x_{2d} \quad (19c)$$

$$\psi_4 = C_2\dot{x}_{4d} + \frac{1}{R}x_{4d} - (1-d)x_{2d} \quad (19d)$$

#### IV-B. Diseño del controlador

Con la suposición **S5**, se tiene al sistema con una alimentación constante (por ejemplo, una batería) y, de acuerdo a las suposiciones **S2** y **S3**, se disponen de las señales de las corrientes de los inductores  $L_1$  y  $L_2$  así como de los voltajes de los capacitores  $C_1$  y  $C_2$ . Con lo antes mencionado y con el sistema en lazo cerrado dado en (17), es necesario encontrar una función para la razón de trabajo variante en el tiempo  $d$  así como ecuaciones de estado para los estados deseados  $x_{1d}$ ,  $x_{2d}$  y  $x_{3d}$ . Con esto se logra tener que el sistema tenga un seguimiento del voltaje en las terminales de la carga, con todas las señales internas acotadas.

La razón de trabajo variante en el tiempo  $d$  se calcula como está escrito en (8). Al sustituir (8) en (19d), se tiene que el elemento  $\psi_4$  resulta como

$$\psi_4 = 0 \quad (20)$$

La corriente deseada del inductor  $L_1$  se obtiene por medio de una ecuación de estado dada en (9). Sustituyendo (9) en (19a) el elemento  $\psi_1$  resulta como

$$\psi_1 = -K_{IL1}e_1 \quad (21)$$

También, la corriente deseada del inductor  $L_2$  se calcula como se presenta en (11). Al sustituir en (19b) lo dado en (11) y simplificando se llega a que  $\psi_2$  es

$$\psi_2 = -K_{IL2}e_2 \quad (22)$$

Asimismo, el voltaje del capacitor  $C_1$  es obtenido como se propone en (13). Teniendo (13), el elemento  $\psi_3$  dado en (19d) se reduce a

$$\psi_3 = -K_{VC1}e_3 \quad (23)$$

#### Ecuación dinámica de errores

Con (20), (21), (22) y (23) el vector  $\Psi$  resulta en

$$\Psi = \begin{bmatrix} -K_{IL1}e_1 \\ -K_{IL2}e_2 \\ -K_{VC1}e_3 \\ 0 \end{bmatrix} \quad (24)$$

Sustituyendo (24) en (17) se tiene que la ecuación de errores resulta como

$$\mathcal{D}\dot{\mathbf{e}} = -\bar{\mathcal{R}}\mathbf{e} - \mathcal{C}\mathbf{e} \quad (25)$$

donde la matriz  $\bar{\mathcal{R}} \in \mathbb{R}^{4 \times 4}$ , es dada como

$$\bar{\mathcal{R}} = \begin{bmatrix} K_{IL1} & 0 & 0 & 0 \\ 0 & K_{IL2} & 0 & 0 \\ 0 & 0 & K_{VC1} & 0 \\ 0 & 0 & 0 & \frac{1}{R} \end{bmatrix} \quad (26)$$

#### V. PRUEBA DE ESTABILIDAD

Una función que se puede utilizar para probar estabilidad en el sentido de Lyapunov de (3) es una que relacione la energía total del sistema como

$$\mathcal{H} = \frac{1}{2}\mathbf{x}^T \mathcal{D}\mathbf{x} \quad (27)$$

donde los estados  $\mathbf{x}$  se define en la sección 2 y la matriz  $\mathcal{D}$  en (4).

Tomando en cuenta lo anterior, para proceder a probar la estabilidad del sistema en lazo cerrado (25) se propone utilizar una función como la dada en (27).

Considere una función cuadrática de la siguiente forma

$$\mathcal{H}_e = \frac{1}{2}\mathbf{e}^T \mathcal{D}\mathbf{e} \quad (28)$$

con la matriz  $\mathcal{D}$  dada en (4) y de acuerdo a la suposición **S1**, es definida positiva. Por lo tanto, la función cuadrática  $\mathcal{H}_e$ , dada en (28), cumple con la condición

$$\frac{1}{2}\lambda_{\min}(\mathcal{D})\|\mathbf{e}\|^2 \leq \frac{1}{2}\mathbf{e}^T \mathcal{D}\mathbf{e} \leq \frac{1}{2}\lambda_{\max}(\mathcal{D})\|\mathbf{e}\|^2 \quad (29)$$

donde  $\lambda_{\min}(\mathcal{D})$  y  $\lambda_{\max}(\mathcal{D})$  denotan el valor propio más pequeño y más grande, respectivamente, de la matriz  $\mathcal{D}$  mientras que  $\|\mathbf{e}\|$  la norma euclidiana del error  $\mathbf{e}$ .

La derivada respecto al tiempo de (28), teniendo en cuenta que la matriz  $\mathcal{D}$  contiene elementos constantes de acuerdo a la suposición **S1**, es

$$\dot{\mathcal{H}}_e = \mathbf{e}^T \mathcal{D}\dot{\mathbf{e}} \quad (30)$$

Sustituyendo en (30) el término  $\mathcal{D}\dot{\mathbf{e}}$  que se tiene en (25), considerando la propiedad de antisimetría de la matriz  $\mathcal{C}$  y realizar simplificaciones se llega a

$$\dot{\mathcal{H}}_e = -\mathbf{e}^T \bar{\mathcal{R}}\mathbf{e} \quad (31)$$

con la matriz  $\bar{\mathcal{R}}$  definida en (26), la cual es una matriz diagonal y si cumple con las condiciones (10), (12) y (14) así como con la suposición **S1** para la constante  $R$  de la carga, dicha matriz  $\bar{\mathcal{R}}$  es definida positiva, con lo cual utilizando la teoría de Lyapunov, se concluye estabilidad exponencial del

origen del sistema en lazo cerrado dado por (25). Una cota exponencial explícita del error puede obtenerse utilizando los siguientes movimientos algebraicos. Note que

$$\lambda_{\min}(\bar{\mathcal{R}}) \mathbf{I} \leq \bar{\mathcal{R}} \quad (32)$$

donde  $\lambda_{\min}(\bar{\mathcal{R}})$  es el valor propio más pequeño de la matriz  $\bar{\mathcal{R}}$  mientras que  $\mathbf{I} \in \mathbb{R}^{4 \times 4}$  es una matriz identidad. Entonces, de acuerdo a (32) y realizando manipulación algebraica, se tiene

$$\mathbf{e}^T \bar{\mathcal{R}} \mathbf{e} \geq \frac{2\lambda_{\min}(\bar{\mathcal{R}})}{\lambda_{\max}(\mathcal{D})} \left[ \frac{1}{2} \lambda_{\max}(\mathcal{D}) \right] \|\mathbf{e}\|^2 \quad (33)$$

De acuerdo a la condición que se tiene en (29), la función de desigualdad dada en (33) implica que es

$$-\mathbf{e}^T \bar{\mathcal{R}} \mathbf{e} \leq -\frac{2\lambda_{\min}(\bar{\mathcal{R}})}{\lambda_{\max}(\mathcal{D})} \mathcal{H}_e \quad (34)$$

Considerando la desigualdad (34) para la función dada en (31) se tiene que esta última cumple con

$$\dot{\mathcal{H}}_e \leq -\gamma_e \mathcal{H}_e \quad (35)$$

donde  $\gamma_e = \frac{2\lambda_{\min}(\bar{\mathcal{R}})}{\lambda_{\max}(\mathcal{D})}$ , que de acuerdo a las condiciones para  $\mathcal{D}$  y  $\bar{\mathcal{R}}$  dadas en (29) y (32), respectivamente,  $\gamma_e$  es positiva e independiente del tiempo.

Integrando (35) en el intervalo  $[0, t]$ , se tiene

$$\mathcal{H}_e(t) \leq \mathcal{H}_e(0) + \int_0^t (-\gamma_e) \mathcal{H}_e ds \quad (36)$$

La expresión dada en (36) tiene la estructura para poder aplicar el Lema de Gronwall-Bellman. Al realizar lo antes mencionado, resulta

$$\mathcal{H}_e(t) \leq \mathcal{H}_e(0) e^{-\gamma_e t} \quad (37)$$

De acuerdo a (37) y considerando la propiedad de la función  $\mathcal{H}_e$  dada en (29) se tiene

$$\lambda_{\min}(\mathcal{D}) \|\mathbf{e}\|^2 \leq \mathcal{H}_e(0) e^{-\gamma_e t} \quad (38)$$

Con la propiedad dada en (29), pero con la función  $\mathcal{H}_e$  evaluada en la condición inicial, (38) se reduce a

$$\|\mathbf{e}\| \leq \beta_e \|\mathbf{e}(0)\| e^{-\frac{\gamma_e}{2} t} \quad (39)$$

donde  $\beta_e = \sqrt{\frac{\lambda_{\max}(\mathcal{D})}{\lambda_{\min}(\mathcal{D})}}$ , de igual forma, de acuerdo a la condición para  $\mathcal{D}$  dada en (29)  $\beta_e$  es positiva e independiente del tiempo.

Por lo tanto, la definitividad positiva de las matrices  $\mathcal{D}$  y  $\bar{\mathcal{R}}$  implica que las constantes  $\gamma_e$  y  $\beta_e$  son positivas e independientes del tiempo  $t$ , tal que el vector de error  $\mathbf{e}$ , dado en (39), converge al origen de forma exponencial, i.e.

$$\lim_{t \rightarrow \infty} \|\mathbf{e}(t)\| = 0$$

## VI. RESULTADOS DE SIMULACIÓN

El algoritmo de control propuesto para el convertidor boost cuadrático con  $R^2P^2$  que presenta en este trabajo se implementó en una simulación numérica con el software SIMULINK/MATLAB® con un paso de integración fijo de 0.001 ms y con el método de integración numérica ode8 (Dormand-Prince), utilizando sólo bloques básicos de SIMULINK.

Para realizar la simulación se utilizó el modelo matemático del convertidor boost cuadrático con  $R^2P^2$  dado en (3), la función de la razón de trabajo variante en el tiempo  $d$  (8) y el cálculo de las corrientes de los inductores  $L_1$  y  $L_2$ , (9) y (11) respectivamente, así como del voltaje del capacitor  $C_1$  dada en (13).

Los parámetros del circuito del convertidor utilizados en la simulación son:  $R = 46 \Omega$ ,  $L_1 = 196 \times 10^{-6} \text{ H}$ ,  $L_2 = 767 \times 10^{-6} \text{ H}$ ,  $C_1 = 100 \times 10^{-6} \text{ F}$ ,  $C_2 = 220 \times 10^{-6} \text{ F}$  y la fuente de alimentación  $E = 12 \text{ V}$ .

Las ganancias del controlador que se consideraron que cumplen con las condiciones (10), (12) y (14) para garantizar la estabilidad del sistema en lazo cerrado, son los siguientes valores:  $K_{IL1} = 10$ ;  $K_{IL2} = 100$ ; y  $K_{VC1} = 5$ .

El voltaje de referencia para la carga se consideró como

$$V_{C2d} = \begin{cases} 15 & 0 < t < 0.5 \\ 5t + 12.5 & 0.5 \leq t < 1.5 \\ 20 & 1.5 \leq t \end{cases} \quad (40)$$

En la Figura 2 se presentan las gráficas del voltaje deseado, dado en (40) y de color rojo, así como del voltaje real en color azul. En dicha Figura se observa que el voltaje real realiza un seguimiento de la señal de referencia. Cabe hacer mención que la función dada en (40) no cumple totalmente con la suposición **S4** y a pasar de esto, en la simulación numérica, como se puede ver en la Figura 2, hay un seguimiento del voltaje que se tiene en las terminales de la carga. En el instante que se tiene una señal tipo escalón, a los 0 s, el voltaje de salida alcanza a la referencia a los de 0.0845 s, como se puede ver en el recuadro de acercamiento que se tiene en la Figura 2.

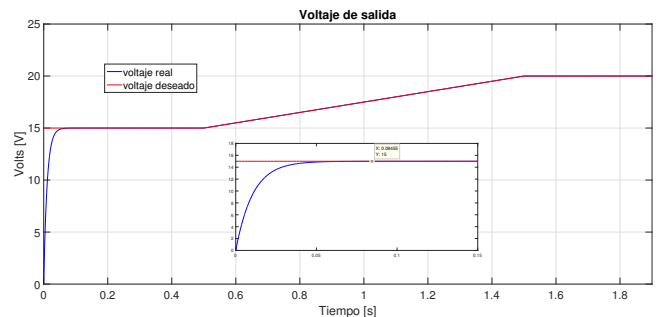


Figura 2: Respuesta de voltaje de salida, en las terminales de la carga.

Para obtener el voltaje en las terminales de la carga de acuerdo a la función (40) se requiere una razón de trabajo

como se muestra en la Figura 3, que como se puede observar en el recuadro de acercamiento de dicha Figura es variante en el tiempo.

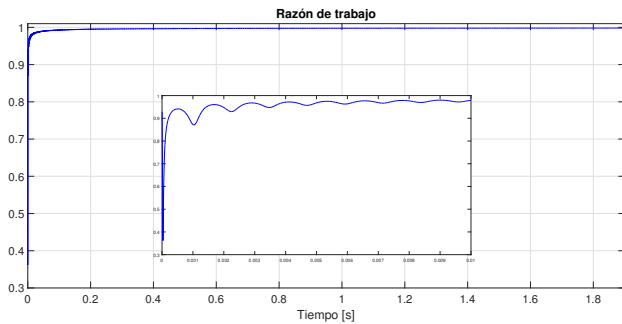


Figura 3: Razón de trabajo requerido para obtener el voltaje deseado en las terminales de la carga.

Una posible fuente de perturbación que se puede tener en el sistema en lazo cerrado, es la retoalimentación de las señales contaminadas de ruido, por lo que se realizó una simulación del sistema considerando que se tiene ruido blanco al voltaje del capacitor 1 por medio de un bloque “Band-Limited White Noise” de Simulink con un parámetro de potencia de ruido de 0.1. Se comparó la respuesta del voltaje de la salida sin ruido y con esta condición (restando dichas señales), por lo que en la Figura se tiene la gráfica. Como se puede observar, hay una diferencia máxima de 8 [mV].

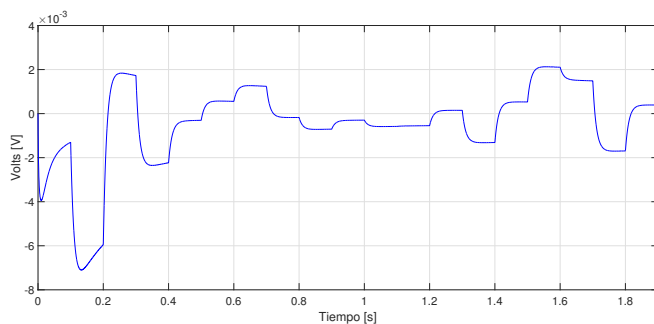


Figura 4: Error de voltaje de salida sin y sin ruido en el voltaje del capacitor 1.

## VII. CONCLUSIONES

En el presente escrito, se muestra un controlador para el convertidor boost cuadrático con  $R^2P^2$  considerando el modelo matemático promedio no lineal que se ha obtenido para dicho convertidor. Dicho controlador es novedoso en el sentido de que en el diseño del control se consideran todas las variables de estado (sin la necesidad de linealizar el modelo promediado), a partir de sólo tener como referencia el voltaje deseado del voltaje de la carga, se calcula la razón de trabajo como señal de control (que es variante en el tiempo) en conjunto con el cálculo de las otras tres variables de estados. El controlador presenta un seguimiento exponencial del voltaje en las terminales de la carga que se tiene en las terminales

de la salida del convertidor. Como trabajo futuro se tiene el implementar físicamente el controlador.

## AGRADECIMIENTOS

Este trabajo fue desarrollando con el apoyo de la Universidad Autónoma de la Ciudad de México (UACM), por el proyecto CONACYT 134534, así como proyectos del TecNM, y las actividades de la red internacional denominada “Red Internacional de control y cómputo aplicados (RICCA)”.

## REFERENCIAS

- [1] R. Selvaraj, K. Chittibabu, M. Anandharaj, and R. Perumal, “Modeling and analysis of hybrid quadratic boost converter in matlab/simulink environment,” in *2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, 2022, pp. 1–6.
- [2] J. A. Morales-Saldaña, R. Loera-Palomo, E. Palacios-Hernández, and J. L. González-Martínez, “Modelling and control of a dc–dc quadratic boost converter with r2p2,” *IET Power Electronics*, vol. 7, no. 1, pp. 11–22, 2014.
- [3] P. Reddy and Y. Obulesu, “A non-cascading dc/dc quadratic boost converter with high voltage gain for pv applications,” *International Journal of Electronics*, vol. 110, 08 2022.
- [4] C. Zogogianni, E. Tatakis, and M. Vekic, “Non-isolated reduced redundant power processing dc/dc converters: A systematic study of topologies with wide voltage ratio for high-power applications,” *IEEE Transactions on Power Electronics*, vol. PP, pp. 1–1, 12 2018.
- [5] P. M. García-Vite, J. C. Rosas-Caro, A. L. Martínez-Salazar, J. d. J. Chavez, A. Valderrábano-González, and V. M. Sánchez-Huerta, “Quadratic buck–boost converter with reduced input current ripple and wide conversion range,” *IET Power Electronics*, vol. 12, no. 15, pp. 3977–3986, 2019.
- [6] R. Loera-Palomo and J. A. Morales-Saldaña, “Family of quadratic step-up dc–dc converters based on non-cascading structures,” *IET Power Electronics*, vol. 8, no. 5, pp. 793–801, 2015.
- [7] M. N. Pefkianaki, C. G. Zogogianni, and E. C. Tatakis, “Investigation of the operational behavior of a dc/dc high step up r2p2 converter,” in *2017 Panhellenic Conference on Electronics and Telecommunications (PACET)*. IEEE, 2017, pp. 1–4.
- [8] J. R. R. Zientarski, M. L. da Silva Martins, J. R. Pinheiro, and H. L. Hey, “Evaluation of power processing in series-connected partial-power converters,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 7, no. 1, pp. 343–352, 2018.
- [9] Z. Jianfeng, X. Jianping, Z. Xiang *et al.*, “A soft-switched high step-up quadratic boost converter with auxiliary circuit,” *Transactions of China Electrotechnical Society*, vol. 32, no. 24, pp. 71–79, 2017.
- [10] E. Netzahuatl-Huerta, J. Morales-Saldaña, I. Reyes-Portillo, and R. Peña-Gallardo, “ $R^2p^2$  boost-boost converter for high efficient power stages,” in *2021 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, vol. 5. IEEE, 2021, pp. 1–6.
- [11] I. Reyes-Portillo, J. Morales-Saldaña, E. Netzahuatl-Huerta, E. Palacios-Hernández, and S. Méndez-Elizondo, “Modeling of a quadratic buck converter based on the r2p2 concept for pv applications,” in *2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, vol. 4. IEEE, 2020, pp. 1–6.
- [12] I. A. Reyes-Portillo, J. Morales-Saldaña, C. Romero-Rivera, and E. Palacios-Hernández, “Design and modeling of a high current ratio converter for pv applications,” *IEEE Latin America Transactions*, vol. 21, no. 10, pp. 1144–1155, 2023.

# Diseño de un Controlador en Tiempo–Finito para Regulación en Robots Móviles No–Holónomos

Manuel Mera

Section of Graduate Studies and Research  
ESIME–UPT, Instituto Politécnico Nacional  
C.P. 07340, CDMX, México  
mmerah@ipn.mx

Héctor Ríos

División de Estudios de Posgrado e Investigación  
Tecnológico Nacional de México/I.T. La Laguna  
C.P. 27000, Torreón, Coahuila, México  
hriosb@lalaguna.tecnm.mx

**Resumen**—En este artículo se presenta un controlador que asegura la convergencia en tiempo–finito de la posición y orientación de un robot móvil no–holónimo a cualquier punto del espacio de estados, desde casi cualquier condición inicial. El diseño de control se basa en resultados previos sobre la convergencia en tiempo–finito del sistema de Heisenberg, conocido también como el integrador de Brockett. Dicho controlador tiene como base el control por vector unitario, una técnica muy conocida en el área de modos deslizantes. Sin embargo, no se requiere el diseño de una superficie deslizante. El desempeño del controlador en tiempo–finito se valida a través de algunas simulaciones numéricas.

**Index Terms**—Robot Móvil Uniciclo, Robots No–Holónomos, Control No lineal.

## I. INTRODUCCIÓN

LOS robots móviles uniciclo (UMRs) han llamado la atención de manera importante durante la última década debido principalmente a su capacidad para moverse libremente de un punto a otro y a la gran diversidad de aplicaciones que estos robots poseen (ver, *e.g.*, [1], [2] y [3]).

En este tipo de robots móviles es muy común, debido a restricciones de acceso a las señales de voltaje/corriente de los motores de las ruedas, considerar el modelo cinemático. El hecho de que el modelo cinemático de esta clase de robots móviles no cumpla con las condiciones necesarias de Brockett, para su estabilización a través de realimentación de estados suave [4], representa una de las principales dificultades en su diseño de control [5]. De este modo, el diseño de controladores por realimentación de estados no–suave o variantes en el tiempo se convierte en una necesidad para dicha clase de sistemas robóticos móviles. Además, es importante resaltar que el modelo cinemático de los robots móviles uniciclo es difeomorfo al sistema de Heisenberg; y por lo tanto, es posible aplicar controladores, diseñados originalmente para la regulación del sistema de Heisenberg, al problema de regulación en robots móviles uniciclo.

El sistema de Heisenberg, también conocido como el integrador de Brockett o el integrador no–holónimo, es un sistema no lineal cuyos campos vectoriales generan el álgebra de Heisenberg (ver, *e.g.*, [6] y [7]). Este sistema es un sistema no–holónimo prototípico para muchos modelos matemáticos de sistemas físicos, incluidos los robots móviles uniciclo. Por

esta razón, el sistema de Heisenberg, y su forma canónica encadenada, han sido muy usados como *benchmark* para el diseño de controladores y el análisis de estabilidad de robots móviles no–holónomos (ver, *e.g.*, [8] y [9]).

En la literatura se pueden encontrar resultados recientes y notables relacionados con el problema de estabilización de sistemas no–holónomos en su forma encadenada. Por ejemplo, en [10], se propone un control adaptable por modos deslizantes que asegura la convergencia en tiempo–finito de la variable de deslizamiento. Sin embargo, la convergencia de los estados del sistema al origen es asintótica. En [11], se propone un controlador que garantiza la convergencia de los estados del sistema al origen en tiempo–finito, pero es necesario que una condición de sector homogénea se cumpla. Por otro lado, aunque existen otros trabajos que garantizan la convergencia en tiempo–finito [12], en tiempo–fijo [13] y en tiempo–predefinido [14], los controladores propuestos requieren señales de control no acotadas, sintonización de muchos parámetros de diseño, o cálculos en línea complejos. Además, el problema de regulación más general, *i.e.*, la convergencia a un punto arbitrario del espacio de estados, normalmente no se aborda. Esto es relevante porque, para esta clase particular de sistemas, un controlador que garantice la convergencia al origen no garantiza necesariamente la convergencia a cualquier punto del espacio de estados. Es importante mencionar que, a pesar de que existen muchos resultados sobre el diseño de controladores para el problema de seguimiento de trayectorias en UMRs (ver, *e.g.*, [15], [16], [17], [18], [19], y las referencias ahí mencionadas), el problema de regulación es completamente diferente al problema de seguimiento de trayectorias debido a las restricciones no–holónomas.

Este trabajo contribuye con el diseño de un controlador que asegura la convergencia en tiempo–finito de la posición y orientación de un robot móvil no–holónimo a cualquier punto del espacio de estados, desde casi cualquier condición inicial. El diseño del control se basa en un controlador en tiempo–finito desarrollado para el sistema de Heisenberg [20]. Dicho controlador tiene como base el control por vector unitario, una técnica muy conocida en el área de modos deslizantes. Sin embargo, no se requiere el diseño de una superficie deslizante, su sintonización es muy simple y la señal de control está

acotada. Estos son puntos claves desde el punto de vista de la implementación.

La estructura del resto del artículo es la siguiente. La definición del sistema, así como el planteamiento a resolver se presenta en la Sección II. La Sección III introduce algunas definiciones relevantes y resultados previos necesarios para el desarrollo teórico de los resultados. La estructura matemática del control propuesto, así como algunos comentarios relevantes relacionados, se presentan en la Sección IV. Mientras que, en la Sección V, se incluyen unas simulaciones con el fin de ilustrar el desempeño y una posible implementación del resultado. Finalmente, en la Sección VI se dan comentarios a manera de conclusión.

**Notación.** Denote  $\mathbb{R}_+ = \{x \in \mathbb{R} : x > 0\}$ ,  $\mathbb{R}_- = \{x \in \mathbb{R} : x < 0\}$  y  $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} : x \geq 0\}$ , donde  $\mathbb{R}$  es el conjunto de los números reales;  $\|\cdot\|$  denota la norma Euclidiana en  $\mathbb{R}^n$ . Defina la función  $[a]^\gamma = |a|^\gamma \text{sign}(a)$ , para cualquier  $\gamma \in \mathbb{R}_{\geq 0}$  y cualquier  $a \in \mathbb{R}$ . El término  $S^1$  es la 1-esfera. El término  $SO(2)$  representa el grupo ortogonal especial en  $\mathbb{R}^2$ , y cualquier elemento de  $SO(2)$  representa una rotación estándar en  $\mathbb{R}^2$ , definida para  $\theta \in S^1$ , por la matriz

$$\mathbf{R}(\theta) := \begin{bmatrix} \text{c}(\theta) & -\text{s}(\theta) \\ \text{s}(\theta) & \text{c}(\theta) \end{bmatrix},$$

donde la notación  $\text{c}(\cdot)$  y  $\text{s}(\cdot)$  representa las funciones trigonométricas  $\cos(\cdot)$  y  $\sin(\cdot)$ , respectivamente.

## II. PLANTEAMIENTO DEL PROBLEMA

Considere el modelo cinemático del UMR (ver, Fig 1):

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

$$\dot{x}(t) = v(t)\text{c}(\theta), \quad (2)$$

$$\dot{y}(t) = v(t)\text{s}(\theta), \quad (3)$$

donde  $x, y \in \mathbb{R}$  denotan la posición del centro de masa del robot móvil,  $\theta \in \mathbb{R}$  representa el ángulo de orientación de las ruedas, mientras que las entradas de control,  $\omega$  y  $v$ , representan la velocidad angular y lineal, respectivamente. Para abreviar la notación es posible definir el vector de estados del UMR como  $\xi = (\theta, x, y)^\top$ , así como sus entradas  $u = (\omega, v)^\top$ .

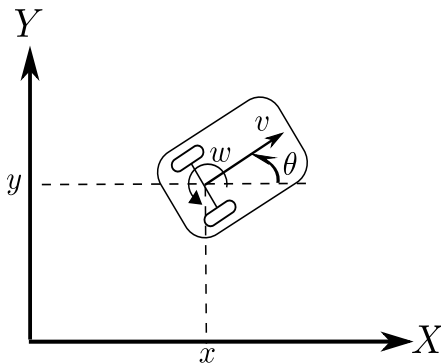


Figura 1. Diagrama del UMR

El problema a resolver es el diseño de un controlador capaz de llevar el UMR a una posición y orientación final deseada, que es arbitraria,  $\xi_r = (\theta_r, x_r, y_r)$ , a partir de casi cualquier posición y orientación inicial  $\xi_0 = (\theta(0), y(0), x(0))$ , en un tiempo finito, *i.e.*,  $\xi(T) = \xi_r$ , con  $0 \leq T < \infty$ . Además, se desea obtener una estimación del tiempo de convergencia  $T$ .

## III. PRELIMINARES

Considere el sistema

$$\dot{x} = f(t, x), \quad t \in \mathbb{R}_{\geq 0}, \quad x(0) = x_0, \quad (4)$$

donde  $x \in \mathbb{R}^n$  es el vector de estado. La función  $f : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  se asume uniformemente localmente acotada en  $t$ . Para  $f$  localmente medible pero discontinua con respecto a  $x$ , las soluciones se entienden en el sentido de Filippov [21]. Ésto es,  $x(t, x_0)$  es solución de (4) si es absolutamente continua, y si satisface la inclusión diferencial

$$\dot{x} \in K[f](t, x) = \overline{\text{co}} \bigcap_{\varepsilon > 0} \bigcap_{\mu N = 0} f(t, B(x, \varepsilon) \setminus N),$$

donde  $\overline{\text{co}}(M)$  representa la cerradura convexa de  $M$ ,  $B(x, \varepsilon)$  representa una bola centrada en  $x$  con radio  $\varepsilon$ ,  $\mu$  es la medida de Lebesgue. Note que las intersecciones se toman sobre todos los conjuntos  $N$  de medida cero, para todo  $\varepsilon > 0$ .

**Definición 1.** [22]. Sea el origen un punto de equilibrio de (4) *i.e.*, sea  $\mathbf{0} \in K[f](t, \mathbf{0})$ . El origen se dice atractivo en tiempo finito, con una región de atracción  $\Omega \subseteq \mathbb{R}^n$ , si existe una función localmente acotada  $T : \mathbb{R}^n \setminus \{\mathbf{0}\} \rightarrow \mathbb{R}_+$  tal que  $x(t, x_0) = \mathbf{0}$ , para cualquier  $x_0 \in \Omega$ , y todo  $t \geq T(x_0)$ . La función  $T$  se llama la función de asentamiento.

Las ecuaciones dinámicas del sistema de Heisenberg son las siguientes

$$\dot{z}(t) = Y^\top(t) \mathbf{J} X(t), \quad (5)$$

$$\dot{X}(t) = Y(t), \quad (6)$$

con los estados  $z \in \mathbb{R}$  y  $X \in \mathbb{R}^2$ , y las condiciones iniciales  $z(0) = z_0$  y  $X(0) = X_0$ , respectivamente. La función  $Y : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^2$  es la señal de control, mientras que  $\mathbf{J} \in \mathbb{R}^{2 \times 2}$  es la matriz anti-simétrica

$$\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

De esta forma es posible definir de manera breve los estados del sistema de Heisenberg como  $q = [z, X^\top]^\top = [z, x_1, x_2]^\top$ , así como sus entradas  $Y = [y_1, y_2]^\top$ .

## IV. DISEÑO DE CONTROL

### IV-A. Transformación a la Forma de Heisenberg

Para el diseño del controlador propuesto, se requiere transformar las coordenadas del UMR dadas en (1)–(3) a la forma equivalente del sistema de Heisenberg (5)–(6), con este fin se presenta el siguiente Lema.

**Lema 2.** El cambio de coordenadas  $\Psi : (\xi, u) \rightarrow (q, Y)$ , dado por

$$\begin{bmatrix} z \\ x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\theta xc(\theta) + \frac{1}{2}\theta ys(\theta) - xs(\theta) + yc(\theta) \\ -\frac{1}{2}(xc(\theta) + ys(\theta)) \\ -\theta \\ -\omega \\ -\frac{1}{2}(v + (yc(\theta) - xs(\theta))\omega) \end{bmatrix},$$

es un difeomorfismo.

*Proof:* La prueba es directa notando que  $\partial\Psi/\partial(\xi, v)$  es suave y no-singular en todo  $\mathbb{R}^5$ . Además, debido a la biyectividad de  $\Psi$  es posible diseñar el control en las coordenadas  $(q, Y)$ , usando  $\Psi^{-1}$ , dada por

$$\begin{bmatrix} \theta \\ x \\ y \\ \omega \\ v \end{bmatrix} = \begin{bmatrix} -x_1 \\ -2x_2c(x_1) + (z - x_1x_2)s(x_1) \\ 2x_2s(x_1) + (z - x_1x_2)c(x_1) \\ -y_1 \\ (z - x_1x_2)y_1 - 2y_2 \end{bmatrix}, \quad (7)$$

y aplicando el controlador en las coordenadas  $(\xi, u)$ . ■

#### IV-B. Estructura del Controlador

Es posible re-plantear el problema dado en la sección II, en las coordenadas  $(q, Y)$ , como: dada una referencia constante  $q_r \in \mathbb{R}^3$ , definiendo el vector de error

$$q(t) - q_r = \begin{bmatrix} z(t) - z_r \\ X(t) - X_r \end{bmatrix} = \begin{bmatrix} \tilde{z}(t) \\ \tilde{X}(t) \end{bmatrix}, \quad (8)$$

y la variable auxiliar

$$\Gamma(t) = \tilde{z}(t) - \tilde{X}^\top(t) \mathbf{J} X_r,$$

diseñar un control  $Y(t)$ , tal que el origen del sistema

$$\dot{\Gamma}(t) = Y^\top(t) \mathbf{J} \tilde{X}(t), \quad (9)$$

$$\dot{\tilde{X}}(t) = Y(t), \quad (10)$$

con el vector de estados

$$\tilde{q}(t) = \begin{bmatrix} \Gamma(t), \tilde{X}^\top(t) \end{bmatrix}^\top,$$

y la condición inicial  $\tilde{q}(0) = \tilde{q}_0$ , sea atractivo en tiempo-finito.

Antes de presentar formalmente el controlador propuesto, se requiere introducir la definición de algunas funciones adicionales y resultados previos sobre el control del sistema de Heisenberg. Primero, considere las funciones

$$\omega(\tilde{q}) = \text{atan}(\Phi(\tilde{q})),$$

$$\Phi(\tilde{q}) = \begin{cases} \frac{2\beta \|\Gamma(t)\|^\frac{1}{2}}{\|\tilde{X}(t)\|}, & \text{si } \tilde{q} \neq \mathbf{0}, \\ 0, & \text{si } \tilde{q} = \mathbf{0}, \end{cases}$$

con los parámetros escalares  $\gamma, \beta \in \mathbb{R}_{\geq 0}$ , y la matriz  $\mathbf{R} \in SO(2)$ . Además, sean los conjuntos  $\mathcal{M}_{\tilde{x}} = \{\tilde{q} \in \mathbb{R}^3 : \tilde{X} = \mathbf{0}\}$ , y

$$\mathcal{D}_\beta = \{\tilde{q} \in \mathbb{R}^3 : |\Phi(\tilde{q})| \leq 2\beta^2\},$$

donde evidentemente  $\mathcal{M}_{\tilde{x}} \cap \mathcal{D}_\beta = \{\mathbf{0}\}$ . Finalmente, se definen las constantes  $\omega_\beta = \text{atan}(2\beta^2)$  y  $C_\beta = c(\omega_\beta)$ .

**Teorema 3.** [20]. Considere el sistema (9)–(10), con la señal de control

$$Y(t) = -\gamma \mathbf{R}(\omega(\tilde{q})) \frac{\tilde{X}(t)}{\|\tilde{X}(t)\|}, \quad (11)$$

donde  $\tilde{X}/\|\tilde{X}\| \in \{\theta \in \mathbb{R}^2 : \|\theta\| \leq 1\}$  para  $\tilde{X} = \mathbf{0}$ , con los parámetros  $\beta \geq |\Gamma_0|^\frac{1}{2}/\|\tilde{X}_0\|$  y  $\gamma \in \mathbb{R}_{\geq 0}$ . El origen del sistema en lazo cerrado es atractivo en tiempo-finito y la región de atracción está dada por el conjunto  $\mathcal{D}_\beta$ . Además,  $\mathcal{D}_\beta$  es un conjunto invariante para las soluciones del sistema en lazo cerrado, y el tiempo de convergencia satisface

$$T_{\tilde{q}}(\tilde{q}_0) \leq \frac{\|\tilde{X}_0\|}{\gamma C_\beta},$$

i.e.,  $\tilde{q}(t) = \mathbf{0}$ , para todo  $t \geq T_{\tilde{q}}(\tilde{q}_0)$ .

#### IV-C. Observaciones de la Señal de Control $Y(t)$

- Las soluciones del sistema (9)–(10) en lazo cerrado con el control (11) son suaves fuera del origen. Sin embargo, debido a que para  $\tilde{q} = \mathbf{0}$  el control (11) es discontinuo, las soluciones se entienden en el sentido de Filippov.
- El control (11) está globalmente acotado dado que, para todo  $t \geq 0$ , se tiene

$$\|Y(t)\| = \left( \gamma^2 \frac{\tilde{X}^\top(t)}{\|\tilde{X}(t)\|} \mathbf{R}^2(\omega(\tilde{q})) \frac{\tilde{X}(t)}{\|\tilde{X}(t)\|} \right)^\frac{1}{2} = \gamma.$$

- Note que, dada cualquier  $\tilde{q}_0 \notin \mathcal{M}_{\tilde{x}}$ , siempre existe  $\beta \in \mathbb{R}_{\geq 0}$ , tal que  $\tilde{q}_0 \in \mathcal{D}_\beta$ . Sin embargo, para  $\tilde{q}_0 \in \mathcal{M}_{\tilde{x}} \setminus \mathbf{0}$ , no existe  $\beta \in \mathbb{R}_{\geq 0}$ , tal que  $\tilde{q}_0 \in \mathcal{D}_\beta$ . Por lo tanto, no es posible asegurar que el origen del sistema (9)–(10) sea atractivo en tiempo-finito con el control (11), si las condiciones iniciales se encuentran en  $\mathcal{M}_{\tilde{x}} \setminus \mathbf{0}$ . Esta restricción, para la aplicación en robots móviles, no representa mayor problema pues si se tiene  $\tilde{q}_0 \in \mathcal{M}_{\tilde{x}} \setminus \mathbf{0}$ , siempre es posible dar un impulso como señal de control inicial para que  $\tilde{q}(t) \notin \mathcal{M}_{\tilde{x}} \setminus \mathbf{0}$ , para el instante posterior al inicial.

#### IV-D. Control en las Coordenadas $(\xi, u)$

Defina el error de regulación en las coordenadas  $(\xi, u)$  como

$$\tilde{\xi}(t) = \begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \\ \tilde{\theta}(t) \end{bmatrix} = \begin{bmatrix} x(t) - x_r \\ y(t) - y_r \\ \theta(t) - \theta_r \end{bmatrix} = \xi(t) - \xi_r.$$

Debido a que  $\Psi$  es un difeomorfismo, y cumple con  $\Psi(\mathbf{0}, \cdot) = (\mathbf{0}, \cdot)$ , la convergencia al origen de  $\tilde{\xi}$ , es equivalente a la convergencia al origen del sistema (9)–(10) con la referencia  $q_r = [I_{3 \times 3}, \mathbf{0}_{3 \times 2}] \Psi(\xi_r, \mathbf{0})$ . De esta forma, el control (11) aplicado en las coordenadas  $(\xi, u)$  al sistema (1)–(3), asegura la convergencia al origen en tiempo finito del vector de error  $\tilde{\xi}$ . Este resultado se presenta formalmente en el siguiente Corolario, el cual se puede deducir directamente del Teorema 3.

**Corolario 4.** *Considere el sistema (1)–(3), con la señal de control*

$$u(t) = [\mathbf{0}_{3 \times 3}, I_{3 \times 2}] \Psi^{-1}(q(t), Y(t)) = S(\xi(t))Y(t),$$

para todo  $t \geq 0$ , con  $Y(t)$ , así como sus parámetros, dados en el Teorema 3 con

$$q(t) = [I_{3 \times 3}, \mathbf{0}_{3 \times 2}] (\Psi(\xi(t), \mathbf{0})), \quad (12)$$

$$q_r = [I_{3 \times 3}, \mathbf{0}_{3 \times 2}] (\Psi(\xi_r, \mathbf{0})), \quad (13)$$

y

$$S(\tilde{\xi}(t)) = \begin{bmatrix} -1 & 0 \\ y(t)c(\theta(t)) - x(t)s(\theta(t)) & -2 \end{bmatrix}.$$

Entonces, para cualquier condición inicial tal que  $y_0c(\theta_0) \neq x_0s(\theta_0)$ , el origen de la dinámica del error de regulación es atractivo en tiempo-finito, y además, el tiempo de convergencia satisface

$$T(\tilde{\xi}_0) \leq \frac{\left\| \begin{bmatrix} \frac{1}{2} (x_0c(\theta_0) + y_0s(\theta_0)) \\ \theta_0 \end{bmatrix} \right\|}{\gamma C_\beta}.$$

*Proof:* Note que de (7) se obtiene que

$$\begin{bmatrix} \omega(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} -y_1(t) \\ (z(t) - x_1(t)x_2(t))y_1(t) - 2y_2(t) \end{bmatrix} = S(\tilde{\xi}(t))Y(t),$$

mientras que (12)–(13) se obtienen directamente de la definición de  $\tilde{q}$  en (8) y de la definición de  $\Psi$ , dada en el Lema 2. Finalmente, la condición  $\tilde{q}_0 \notin \mathcal{M}_{\tilde{x}} \setminus \mathbf{0}$ , mencionada en la observación 3 y expresada en las coordenadas  $(\xi, u)$ , da la restricción  $y_0c(\theta_0) \neq x_0s(\theta_0)$ . La convergencia en tiempo-finito es resultado del Teorema 3. ■

## V. SIMULACIONES

Con el fin de ilustrar el desempeño del controlador propuesto y una posible implementación, se presentan las siguientes simulaciones. Estas simulaciones se llevaron a cabo empleando el método de integración de Euler, con un paso de muestreo igual a 0.001[s].

### V-A. Un Único Punto de Referencia

Para la primer simulación se considera solamente un punto de referencia. Las condiciones iniciales para el UMR son  $\xi_0 = [-\pi/2, -5, 8]^T$  y el punto de referencia a alcanzar es  $\xi_r = [0, 2, -2]^T$ . Por lo tanto, para estas condiciones iniciales los parámetros del control (11) se seleccionan cumpliendo con las condiciones del Teorema 3, como  $\beta = 0.2747$  y  $\gamma = 2$ . Finalmente, se obtiene la cota para el tiempo de convergencia, *i.e.*,  $T \leq 2.6501$ . Los resultados se ilustran en las Figuras 2–4.

En la Figura 2 se muestran las entradas de control  $\omega$  y  $v$ . En la Figura 3 se muestra la trayectoria en el plano  $x-y$ , y la convergencia al punto de referencia desde la condición inicial. Finalmente, la Figura 4 presenta la evolución en el tiempo de los estados del UMR.

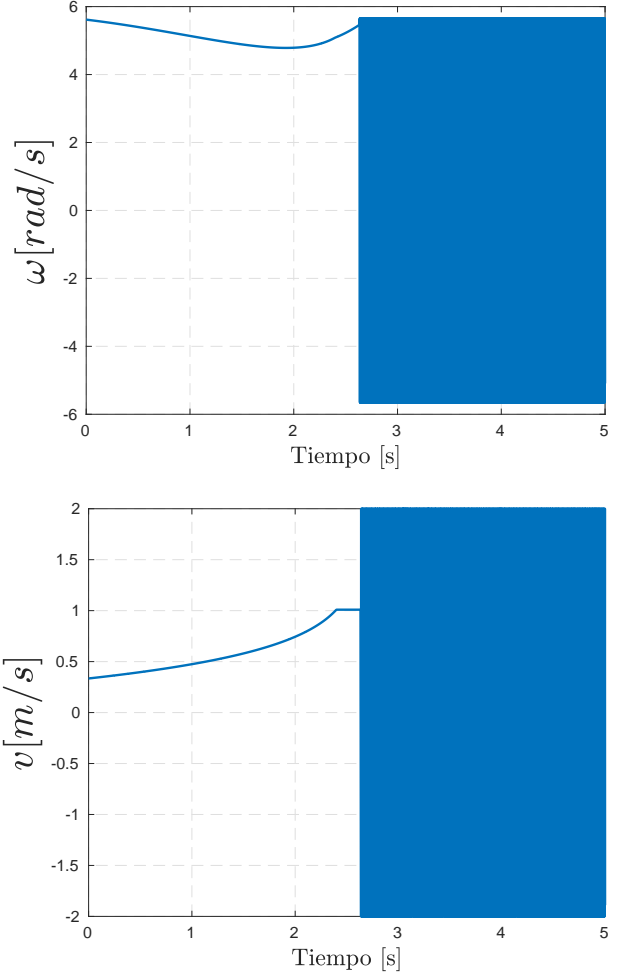


Figura 2. Entradas de Control  $u$

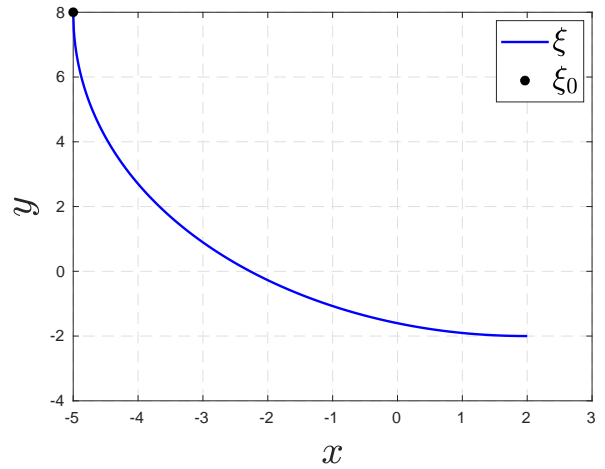


Figura 3. Trayectoria en el Plano  $x-y$

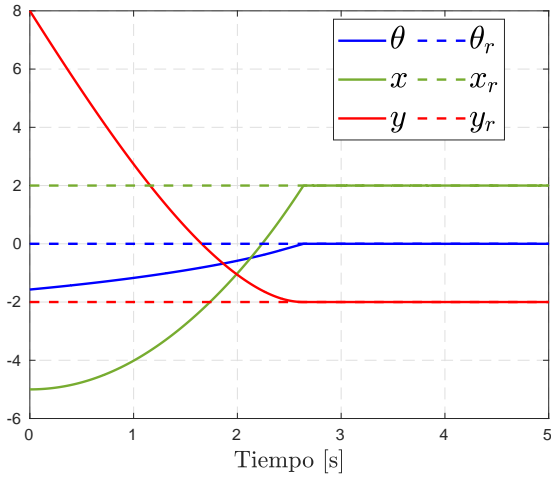


Figura 4. Estados  $\xi$  vs Tiempo

### V-B. Más de Un Punto de Referencia (Seguimiento Punto a Punto)

Para el segundo caso se consideran dos puntos de referencia, los cuales el UMR tiene que alcanzar de forma secuencial. Las condiciones iniciales para el UMR son  $\xi_0 = [0, -6, 7]^T$ , mientras que los puntos a alcanzar de manera secuencial son  $\xi_1 = [-\pi, 1, -1]^T$  y  $\xi_2 = [\pi, -10, 10]^T$ . Los parámetros del controlador se seleccionaron como  $\beta = 0.6380$  y  $\gamma = 2$ . Los resultados se ilustran en las Figuras 5–7.

En la Figura 5 se presentan las señales de control, donde claramente se ven los intervalos de tiempo cuando  $\xi(t) = \mathbf{0}$ . Mientras que la Figura 6 muestra la evolución de la posición del UMR en el plano  $x - y$ . La evolución respecto al tiempo de los estados  $\xi$  se presenta en la Figura 7, donde se puede apreciar la progresión de  $\xi_0$  a  $\xi_2$ , pasando por  $\xi_1$ .

## VI. CONCLUSIONES

En este artículo se introduce el diseño de un control capaz de asegurar la convergencia en tiempo-finito a una referencia arbitraria, dada para la posición y orientación de un robot móvil no-holónomo, desde cualquier condición inicial fuera de un conjunto de medida cero. El controlador, desarrollado previamente para el sistema de Heisenberg, tiene como base el control por vector unitario, una técnica muy conocida en el área de modos deslizantes. Sin embargo, no se requiere el diseño de una superficie deslizante para asegurar la convergencia en tiempo-finito, su sintonización es muy simple y la señal de control está acotada. Estas son características deseables desde el punto de vista de la implementación. Finalmente, el desempeño del controlador en tiempo-finito se validó a través de un par de simulaciones numéricas.

### AGRADECIMIENTOS

Este trabajo fue apoyado en parte por el proyecto SEP-CONACYT-ANUIES-ECOS NORD Proyecto 315597. El trabajo de Manuel Mera fue apoyado en parte por el Proyecto IPN-SIP 20240338. El trabajo de Héctor Ríos fue apoyado en

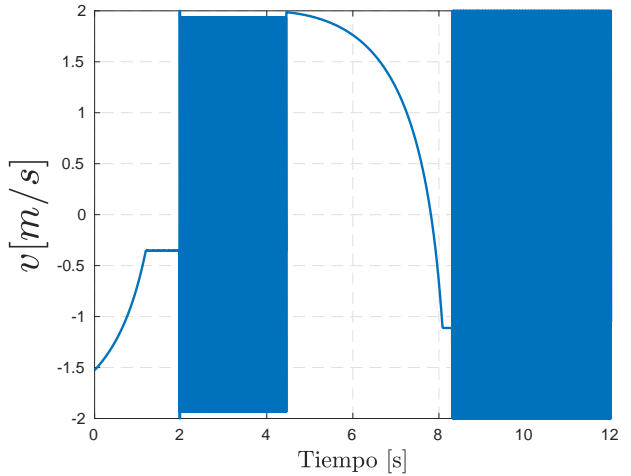
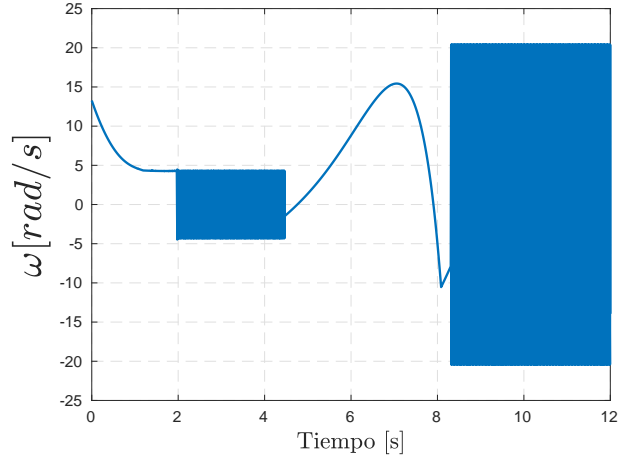


Figura 5. Entradas de Control  $u$

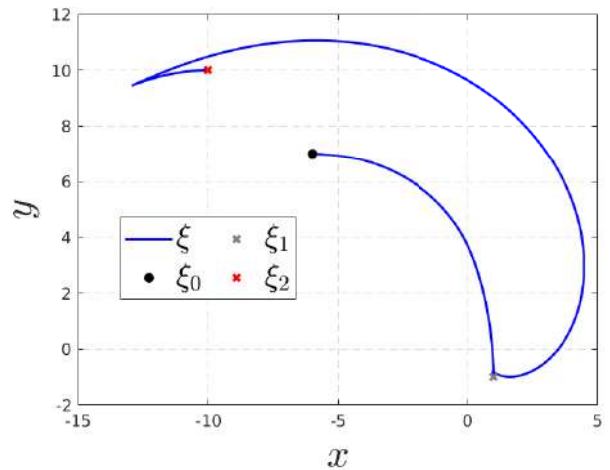


Figura 6. Trayectoria en el Plano  $x - y$

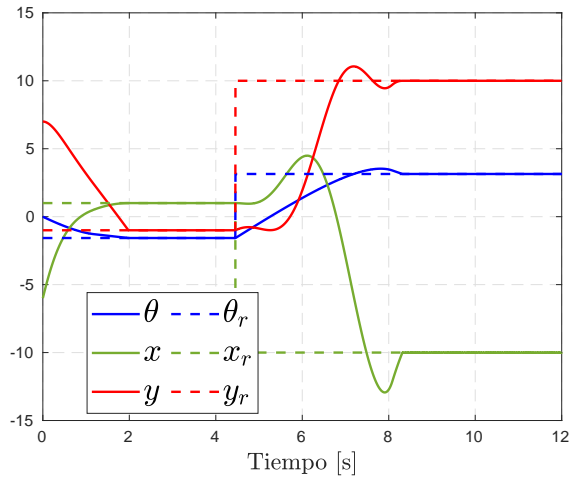


Figura 7. Estados  $\xi$  vs Tiempo

parte por CONAHCYT, Investigadoras e Investigadores por México, CVU 270504 Proyecto 922 y en parte por TecNM Projects.

#### REFERENCIAS

- [1] H. Zhang, J. Sun, and Z. Wang, "Game-theoretical persistent distributed control of nonholonomic robots without global position measurements subject to unknown slippage constraints," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 2, pp. 354–364, 2022.
- [2] M. Khaledyan, T. Liu, V. Fernandez-Kim, and M. de Queiroz, "Flocking and target interception control for formations of nonholonomic kinematic agents," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 4, pp. 1603–1610, 2015.
- [3] M. Zhang and H. Liu, "Game-theoretical persistent tracking of a moving target using a unicycle-type mobile vehicle," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6222–6233, 2014.
- [4] R. W. Brockett, "Control theory and singular riemannian geometry," in *New directions in applied mathematics*. Springer, 1982, pp. 11–27.
- [5] I. Duleba, W. Khefifi, and I. Karcz-Duleba, "Layer, Lie algebraic method of motion planning for nonholonomic systems," *Journal of the Franklin Institute*, vol. 349, no. 1, pp. 201–215, 2012.
- [6] A. M. Bloch, "Nonholonomic mechanics," in *Nonholonomic mechanics and control*. Springer, 2003, pp. 207–276.
- [7] A. Vershik and V. Y. Gershkovich, "Nonholonomic problems and the theory of distributions," *Acta Applicandae Mathematica*, vol. 12, no. 2, pp. 181–209, 1988.
- [8] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [9] N. Marchand and M. Alamir, "Discontinuous exponential stabilization of chained form systems," *Automatica*, vol. 39, no. 2, pp. 343–348, 2003.
- [10] A. Ferrara, G. P. Incremona, and C. Vecchio, "Adaptive multiple-surface sliding mode control of nonholonomic systems with matched and unmatched uncertainties," *IEEE Transactions on Automatic Control*, 2023.
- [11] E. Rocha, F. Castaños, and J. A. Moreno, "Robust finite-time stabilisation of an arbitrary-order nonholonomic system in chained form," *Automatica*, vol. 135, p. 109956, 2022.
- [12] W. Zhu, X. Yu, S. Li, and H. Du, "Finite-time discontinuous control of nonholonomic chained-form systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2022.
- [13] F. Gao, J. Huang, X. Shi, and X. Zhu, "Nonlinear mapping-based fixed-time stabilization of uncertain nonholonomic systems with time-varying state constraints," *Journal of the Franklin Institute*, vol. 357, no. 11, pp. 6653–6670, 2020.
- [14] J. D. Sánchez-Torres, M. Defoort, and A. J. Muñoz-Vázquez, "Predefined-time stabilisation of a class of nonholonomic systems," *International Journal of Control*, vol. 93, no. 12, pp. 2941–2948, 2020.

- [15] M. Mera, H. Ríos, and E. Martínez, "A sliding-mode based controller for trajectory tracking of perturbed unicycle mobile robots," *Control Engineering Practice*, vol. 102, p. 104548, 2020.
- [16] P. Rochel, H. Ríos, M. Mera, and A. Dzul, "Trajectory tracking for uncertain unicycle mobile robots: A super-twisting approach," *Control Engineering Practice*, vol. 122, p. 105078, 2022.
- [17] K. Singhal, V. Kumar, and K. Rana, "Robust trajectory tracking control of non-holonomic wheeled mobile robots using an adaptive fractional order parallel fuzzy PID controller," *Journal of the Franklin Institute*, vol. 359, no. 9, pp. 4160–4215, 2022.
- [18] H. Ríos, M. Mera, and A. Polyakov, "Perturbed unicycle mobile robots: A second-order sliding-mode trajectory tracking control," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 3, pp. 2864–2872, 2024.
- [19] Y. Zhou, H. Ríos, M. Mera, A. Polyakov, G. Zheng, and A. Dzul, "Homogeneity-based control strategy for trajectory tracking in perturbed unicycle mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 1, pp. 274–281, 2024.
- [20] M. Mera and H. Ríos, "Semi-global and robust finite-time regulation of the heisenberg system," *IEEE Transactions on Automatic Control*, vol. 69, no. 10, pp. 7152–7158, 2024.
- [21] A. F. Filippov, *Differential equations with discontinuous righthand sides*. Kluwer Academic Publishers, 1988.
- [22] K. Zimenko, A. Polyakov, and D. Efimov, "On finite-time robust stabilization via nonlinear state feedback," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 16, pp. 4951–4965, 2018.

# Control de sistemas multiagente con la plataforma Lego EV3

Zurisadai Flores-Murcia  
*Depto. de Ingeniería en Mecatrónica*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
zuri3aelec@gmail.com

Daly Y. Lara-Solís  
*Depto. de Ingeniería en Mecatrónica*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
darethls01@gmail.com

Diego Gonzáles-Torres  
*Depto. de Ingeniería en Mecatrónica*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
diegogonzalez3428@gmail.com

Jesús Santiaguillo-Salinas  
*Depto. de Ingeniería en Mecatrónica*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
jsantiaguillo@outlook.com

**Abstract**—En este trabajo se presentan dos estrategias de control con el objetivo de lograr la formación y seguimiento en formación o marcha en un sistema multiagente compuesto por seis robots diferenciales utilizando la plataforma Lego Mindstorms EV3. Para representar al sistema multiagente se utiliza la teoría de grafos y para el diseño de las estrategias de control se considera el modelo cinemático de un punto frontal en cada robot, para el control de formación se emplean grafos directos e indirectos y para la marcha se utiliza un grafo centrado en el líder en donde agentes seguidores hacen una formación con respecto a un agente líder. Los resultados teóricos se validan mediante una plataforma experimental, y se realizan varios experimentos para evaluar el desempeño de ambas estrategias de control.

**Index Terms**—Control de formación, Sistema multiagente, Seguimiento en formación, Teoría algebraica de grafos, Lego Mindstorms EV3, Robots diferenciales.

## I. INTRODUCCIÓN

Los sistemas multiagente se conforman por dos o más agentes inteligentes que interactúan en un mismo entorno [1]. En términos generales, un agente es un ente (persona, robot o dispositivo) que realiza una acción y puede colaborar con otros agentes para resolver una problemática común [2]. Estos sistemas se clasifican en dos categorías: homogéneos y heterogéneos, diferenciándose por los tipos de sensores, actuadores o morfologías de los robots [3].

Una herramienta clave para el análisis y diseño de estos sistemas es la teoría algebraica de grafos, que proporciona un marco para resolver diversos problemas mediante la representación de un conjunto de vértices y aristas [4]. Trabajos como [5] emplea la teoría algebraica de grafos como medio de comunicación para sincronizar un grupo de robots, generando un comportamiento colectivo similar al de un enjambre. De forma complementaria, [6] presenta un modelo de formación líder-seguidor que se centra en mantener una distancia y orientación deseadas entre los seguidores y el líder. Por otro lado, [7] analiza el control en formación utilizando dos enfoques: la rigidez de grafos y el esquema líder-seguidor.

La utilización de sistemas multiagente puede ofrecer un mejor rendimiento en comparación con el uso de un único agente, proporcionando una mayor eficacia y versatilidad en diversos contextos. Por ejemplo, [8] describe un sistema de control y gestión para una flota de robots móviles que se encargan del desplazamiento de cargas en puertos y aeropuertos. Asimismo, [9] propone una estrategia de mapeo visual de entornos mediante el uso de dos robots que trabajan en conjunto para recopilar imágenes.

Del mismo modo, las capacidades de los sistemas multiagente se han investigado mediante plataformas educativas como LEGO Mindstorms EV3. A pesar de ser diseñado para la robótica educativa, es posible expandir su potencial en aplicaciones de investigación utilizando un SO basado en Linux y MicroPython. Por ejemplo, [10] propuso un sistema en el que un grupo de robots LEGO coopera de manera autónoma para transportar un objeto de gran tamaño a una ubicación específica. En otro estudio, [11], se presenta el diseño e implementación de un esquema de control para un enjambre robótico móvil líder-seguidor, basado en un controlador de dinámica de errores utilizando LEGO NXT.

En este trabajo se desarrolla el modelado y diseño de estrategias de control de formación y marcha para un sistema multiagente compuesto por robots móviles diferenciales, utilizando la teoría de grafos para asegurar la coordinación entre los robots. Los resultados teóricos son validados utilizando una plataforma experimental conformada por seis robots móviles diferenciales Lego EV3 y un sistema de posicionamiento local que consta de un conjunto de cámaras OptiTrack. El control de los robots se realiza utilizando el lenguaje Python.

La estructura del trabajo es la siguiente: la sección II presenta el desarrollo matemático del modelo cinemático por punto frontal para  $n$  robots y también se presentan definiciones importantes utilizadas en la teoría algebraica de grafos. En la sección III se define el planteamiento del problema, cuyo objetivo es realizar formaciones y seguimiento de trayectoria

en formación líder-seguidores. En la sección IV se presenta el diseño de las estrategias de control. En la sección V se presentan los resultados experimentales y por último, en la sección VI se presentan las conclusiones y trabajo futuro.

## II. PRELIMINARES

### A. Modelo cinemático del robot diferencial

Considerando un conjunto de  $N$  robots móviles diferenciales, donde la pose del  $i$ -ésimo robot puede representarse como  $q_i = [x_i(t), y_i(t), \theta_i(t)]^T$   $i = 1, \dots, n$ . De acuerdo con [12] y la Fig. 1, la cinemática de cada robot está definida por

$$\dot{q}_i = \begin{bmatrix} \cos \theta_i & 0 \\ \sin \theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} A_i(\theta_i) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad i = 1, \dots, n. \quad (1)$$

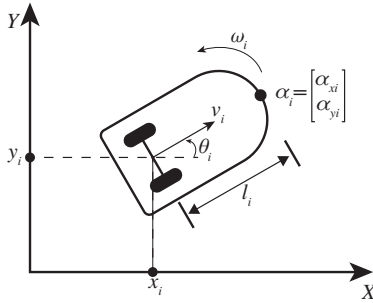


Fig. 1. Esquema del robot móvil diferencial.

Siendo  $\zeta_i = [x_i, y_i]^T$  la posición del  $i$ -ésimo robot en el plano,  $\theta_i$  la orientación respecto al eje  $X$ ,  $A_i(\theta_i)$  se define como la matriz de desacoplamiento,  $v_i$  y  $\omega_i$  son las velocidades lineales y angulares respectivamente. Es evidente que la matriz  $A_i(\theta_i)$  es singular, por lo que, para evitar esta singularidad se propone un modelo cinemático utilizando un punto frontal  $\alpha_i = [\alpha_{xi}(t), \alpha_{yi}(t)]^T$  en donde las nuevas coordenadas que describen la posición del robot son

$$\alpha_i = \begin{bmatrix} x_i + l_i \cos \theta_i \\ y_i + l_i \sin \theta_i \end{bmatrix} \quad (2)$$

donde  $l_i$  representa la distancia del punto medio del eje de las ruedas hasta el punto frontal  $\alpha_i$ . El nuevo modelo cinemático está dado por

$$\dot{\alpha}_i = \underbrace{\begin{bmatrix} \cos \theta_i & -l_i \sin \theta_i \\ \sin \theta_i & l_i \cos \theta_i \end{bmatrix}}_{A_i(\theta_i, l_i)} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad (3)$$

donde  $A_i(\theta_i, l_i)$  es la nueva matriz de desacoplamiento y  $\det(A_i(\theta_i, l_i)) = l_i$ . Podemos garantizar que  $l_i \neq 0$ , esto implica que  $A_i(\theta_i, l_i)$  es no singular, por lo tanto, el modelo cinemático descrito por (3) puede ser ocupado para el diseño de las estrategias de control.

### B. Teoría algebraica de grafos

De acuerdo con [13] la teoría de grafos puede utilizarse para modelar y analizar las interacciones y relaciones entre cada robot que conforman un sistema multiagente.

*Definición 1:* Un grafo  $G = \{V, E, C\}$  consiste en

- Un conjunto de nodos  $V = \{R_1, \dots, R_n\}$ .
- Un conjunto de aristas  $E = \{(R_j, R_i) \in V \times V\}$  con  $i \neq j$ , tal que si  $(R_j, R_i) \in E$  entonces  $R_j \in N_i$ , donde  $N_i$  es el conjunto de vecinos del nodo  $i$ .
- Un conjunto de etiquetas  $C = \{c_{ji} = R_i - R_j\}$  con  $(R_j, R_i) \in E$ .

*Definición 2:* La matriz laplaciana de  $G$  está dada por

$$L(G) = \Delta - A_d \quad (4)$$

donde  $\Delta = \text{diag}\{g_1, \dots, g_n\}$  y  $g_i$  es el grado del conjunto  $N_i$ , el cual define el número de conexiones que posee,  $A_d \in R^{n \times n}$  es llamada matriz de adyacencia con los elementos

$$a_{ij} = \begin{cases} 1 & \text{si } (R_j, R_i) \in E \\ 0 & \text{en otro caso.} \end{cases} \quad (5)$$

*Definición 3:* Un grafo dirigido es aquel en donde cada conjunto de aristas indica que el agente  $j$  puede obtener información del robot  $i$ , pero no al revés. No se permiten auto-aristas  $(i, i)$  a menos que se indique lo contrario.

*Definición 4:* Un grafo no dirigido es aquel en donde cada conjunto de aristas indica que los agentes  $i$  y  $j$  pueden obtener información entre sí. Un grafo no dirigido puede verse como un caso especial de un grafo dirigido, donde una arista  $(i, j)$  en el grafo no dirigido corresponde a las aristas  $(i, j)$  y  $(j, i)$  en el grafo dirigido.

*Definición 5:* Un grafo mixto es una triada  $(V, E, C)$  en donde las aristas pueden ser dirigidas o no dirigidas, permitiendo representar situaciones donde ciertas relaciones son unidireccionales mientras que otras son bidireccionales. Esto proporciona una mayor flexibilidad para modelar las diversas formas de interacción y comunicación entre los nodos (agentes) en el sistema.

## III. PLANTEAMIENTO DEL PROBLEMA

### A. Formación

Sea  $N = \{R_1, \dots, R_n\}$  un conjunto de  $n$  agentes moviéndose en el plano con posiciones  $\alpha_i = [\alpha_{xi}(t), \alpha_{yi}(t)]^T$   $i = 1, \dots, n$  y  $\alpha_i^* = [\alpha_{xi}^*(t), \alpha_{yi}^*(t)]^T$  la posición deseada del agente  $R_i$  respecto a un subconjunto de agentes  $R_j$ , con  $R_j \in N_i$ , para una formación dada en donde

$$\alpha_i^* = \frac{1}{g_i} \sum_{j \in N_i} (\alpha_j + c_{ji}).$$

El objetivo de control es diseñar una estrategia de control  $u_i$  para cada agente  $R_i$  tal que

- Se garantice la convergencia a la formación deseada, i.e.

$$\lim_{t \rightarrow \infty} (\alpha_i(t) - \alpha_i^*(t)) = 0 \quad i = 1, \dots, n.$$

## B. Marcha

Sea  $m(t) = [m_x(t), m_y(t)]^T$  una trayectoria preestablecida continua en el tiempo. Basado en el esquema líder-seguidor, donde  $R_n$  es el agente líder y los agentes restantes son seguidores. Estableciendo  $\alpha_i^*$  como

$$\begin{aligned}\alpha_n^* &= m(t), \\ \alpha_i^* &= \alpha_n + c_{ni} \quad i = 1, \dots, n.\end{aligned}$$

El objetivo de este trabajo es diseñar una estrategia de control tal que

- Se mantenga el seguimiento asintótico de la trayectoria por parte del líder, i.e.

$$\lim_{t \rightarrow \infty} (\alpha_n(t) - m(t)) = 0.$$

- Se garantice la convergencia de la formación deseada por parte de los seguidores con respecto al líder, i.e.

$$\lim_{t \rightarrow \infty} (\alpha_i(t) - \alpha_i^*(t)) = 0.$$

## IV. DISEÑO DE LAS ESTRATEGIAS DE CONTROL

### A. Control de formación

Para el sistema descrito en (3) se propone una estrategia de control basada en una linealización por retroalimentación de estados [14] dada por

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = A_i^{-1}(\theta_i, l_i) u_i \quad i = 1, \dots, n \quad (6)$$

donde  $A_i^{-1}(\theta_i, l_i)$  es la inversa de la matriz de desacoplamiento,  $u_i$  se propone como la entrada de control auxiliar dada por

$$u_i = -k_i(\alpha_i - \alpha_i^*) = -k_i \left( \alpha_i - \frac{1}{g_i} \sum_{j \in N_i} (\alpha_j + c_{ji}) \right) \quad (7)$$

donde  $k_i > 0$  es la ganancia de control para el agente  $R_i$ .

*Proposición 1:* Considerando un grafo de formación directo, indirecto o mixto, el sistema multiagente bajo la ley de control (6)-(7) con  $k_i > 0$  logrará que el sistema en lazo cerrado (3)-(6)-(7) converga exponencialmente a la formación deseada.

*Demostración:* El sistema en lazo cerrado (3)-(6)-(7) resulta

$$\dot{\alpha} = -K((L(G) \oplus I_2)\alpha - C) \quad (8)$$

con  $K = \text{diag}[k_1, \dots, k_n]$ ,  $L(G)$  la matriz laplaciana del sistema,  $\oplus$  denota el producto kronecker,  $I_2$  es la matriz de identidad de  $2 \times 2$ ,  $\alpha = [\alpha_1, \dots, \alpha_n]^T$  y

$$C = \left[ \frac{1}{g_1} \sum_{j \in N_1} c_{j1}, \dots, \frac{1}{g_n} \sum_{j \in N_n} c_{jn} \right].$$

En [15] se demostró que en el sistema en lazo cerrado (8) los agentes convergen exponencialmente a la formación deseada, es decir,  $\lim_{t \rightarrow \infty} (\alpha_i(t) - \alpha_i^*(t)) = 0$ , siempre y cuando la formación deseada esté basada en un grafo de formación

directo, indirecto o mixto. Esta demostración se fundamenta en el uso de la matriz Laplaciana y el teorema de los círculos de Gershgorin [16].  $\square$

### B. Control de marcha

Para el control de marcha bajo el esquema líder-seguidor se propone una estrategia de control análoga a la estrategia de control en formación agregando una prealimentación de la velocidad de marcha

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = A_i^{-1}(\theta_i, l_i) u_i \quad i = 1, \dots, n-1, \quad (9)$$

$$\begin{bmatrix} v_n \\ \omega_n \end{bmatrix} = A_n^{-1}(\theta_n, l_n) u_n, \quad (10)$$

Las variables de control auxiliar están dadas por

$$u_i = -k_i(\alpha_i - \alpha_n - c_{ni}) + \dot{m}(t) \quad (11)$$

$$u_n = -k_n(\alpha_n - m(t)) + \dot{m}(t) \quad (12)$$

con  $\dot{m}(t)$  la velocidad de la trayectoria deseada y  $k_i, k_n > 0$ .

*Proposición 2:* Considerando el sistema multiagente conformado por el agente líder y los agentes seguidores descritos por (3) y el control dado por (9)-(10)-(11)-(12) con  $k_i, k_n > 0$ , en el sistema en lazo cerrado el robot líder converge a la trayectoria deseada, mientras que los seguidores realizan una formación con respecto al líder.

*Demostración:* EL sistema en lazo cerrado resulta

$$\dot{\alpha} = (A \oplus I_2)\alpha + C \quad (13)$$

donde  $C = [k_1 c_{n1} + \dot{m}(t), \dots, k_{n-1} c_{n(n-1)} + \dot{m}(t), k_n m(t) + \dot{m}(t)]$ ,

$$A = -KL(G) - k_n \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix}, \quad (14)$$

y  $K = \text{diag}\{k_1, \dots, k_{n-1}\}$ . Se definen los errores del sistema

$$\begin{aligned}e_1 &= \alpha_1 - \alpha_n - c_{n1} \\ &\vdots \\ e_{n-1} &= \alpha_{n-1} - \alpha_n - c_{n,(n-1)} \\ e_n &= \alpha_n - m(t)\end{aligned}$$

En forma matricial se tiene

$$e = [P \oplus I_2]\alpha - \tilde{C} \quad (15)$$

con  $e = [e_1 \quad \dots \quad e_n]^T$ ,

$$P = L(G) + \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \quad \text{y}$$

$$\tilde{C} = [c_{n1}, c_{n2}, \dots, c_{n,n-1}, m(t)].$$

La dinámica del error resulta

$$\dot{e} = (B \oplus I_2)e \quad (16)$$

donde

$$B = \begin{bmatrix} -k_1 & 0 & 0 & 0 & \dots & 0 & k_n \\ 0 & -k_2 & 0 & 0 & \dots & 0 & k_n \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & 0 & \dots & -k_{n-1} & k_n \\ 0 & 0 & 0 & 0 & \dots & 0 & -k_n \end{bmatrix}.$$

Dado que  $B$  es una matriz triangular superior y  $k_n > 0, k_i > 0 \ i = 1, \dots, n-1$ , se puede concluir que  $B$  es Hurwitz y los errores convergen exponencialmente a cero.  $\square$

## V. TRABAJO EXPERIMENTAL

### A. Funcionamiento

Para validar los resultados teóricos obtenidos, las estrategias de control son implementadas en la plataforma experimental que se muestra en la Fig. 2.

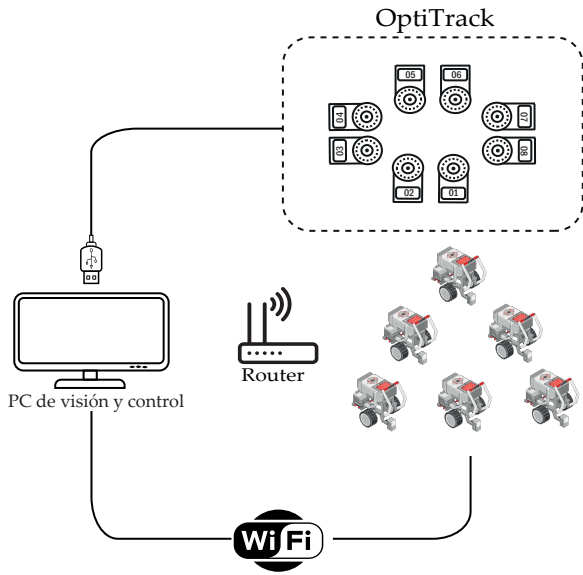


Fig. 2. Plataforma experimental.

La plataforma está compuesta por un sistema de cámaras OptiTrack, que captura la posición y orientación de seis robots Lego EV3. Cada robot está equipado con tres marcadores pasivos acoplados en la parte superior, formando una figura irregular. El sistema captura el centroide de esta figura para obtener las posiciones y orientaciones de cada robot. Para la integración con el software de control, se utiliza el protocolo NatNetClient, un estándar de OptiTrack para la transmisión de datos de captura de movimiento en tiempo real. Un programa en Python implementa el cliente NatNet, que se conecta al servidor Motive, el software de OptiTrack que procesa los datos de las cámaras. A través de esta conexión, el programa en Python recibe las posiciones y orientaciones de cada robot.

El software de control, desarrollado en Python, contiene todas las estrategias de control y se comunica con los robots Lego EV3 utilizando el protocolo Socket INET, facilitando la conexión y transmisión de datos a través de una Red

de Área Local (LAN). El software de control actúa como servidor, esperando las conexiones de los robots Lego EV3, que funcionan como clientes, cada robot está programado en MicroPython y poseen un identificador único, permitiendo que el programa de control identifique a cada agente de forma correcta, para enviarles sus respectivas velocidades.

### B. Experimentación

A continuación se presentan las experimentaciones.

*Experimento 1:* El grafo de formación directa de cadena cíclica empleado en el experimento se muestra en la Fig. 3, dicho grafo es ponderado, es decir, indica las posiciones (en metros) de cada agente respecto a sus vecinos en el plano  $XY$ . Los vectores de posición están dados por:  $c_{65} = [-d_p \cos \theta, -d_p \sin \theta]$ ,  $c_{54} = [d_p \cos \theta, -d_p \sin \theta]$ ,  $c_{43} = [d_p, 0]$ ,  $c_{32} = [d_p \cos \theta, d_p \sin \theta]$ ,  $c_{21} = [-d_p \cos \theta, d_p \sin \theta]$ ,  $c_{16} = [-d_p, 0]$ , donde  $\theta = 60^\circ$  y  $d_p = 0.7 \text{ m}$ . El punto  $\alpha_i$  a controlar se ubica a  $0.10 \text{ m}$  delante del punto medio del eje de las ruedas y será el mismo para cada experimento. Los ganancias utilizadas para este experimento son  $k_1 = k_2 = k_4 = k_5 = k_6 = 0.7, k_3 = 0.84$  y el tiempo de experimentación es de  $60 \text{ s}$ .

La Fig. 4 muestra la gráfica de la posición de los seis agentes en el plano, como se puede observar, cada agente tiene una condición inicial diferente a la formación deseada, conforme transcurre el tiempo cada agente trata de reposicionarse con respecto a su robot vecino hasta llegar a la formación deseada. En la Fig. 5 se puede apreciar que los errores convergen exponencialmente a cero y en aproximadamente  $t = 22 \text{ s}$  se logra la formación deseada, logrando validar la estrategia de control diseñada.

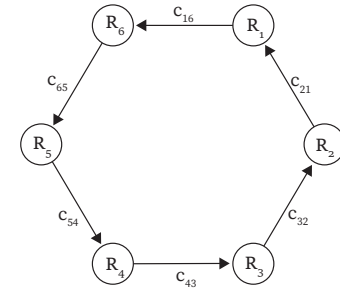


Fig. 3. Grafo empleado en el Experimento 1.

*Experimento 2:* De forma análoga, el grafo de formación indirecta empleado en el segundo experimento se muestra en la Fig. 6. Los vectores de posición son:  $c_{12} = [d_p, 0]$ ,  $c_{21} = [-d_p, 0]$ ,  $c_{32} = [-d_p, 0]$ ,  $c_{23} = [d_p, 0]$ ,  $c_{43} = [-d_p, 0]$ ,  $c_{34} = [d_p, 0]$ ,  $c_{54} = [-d_p, 0]$ ,  $c_{45} = [d_p, 0]$ ,  $c_{65} = [-d_p, 0]$ ,  $c_{56} = [d_p, 0]$ , donde  $d_p = 0.7 \text{ m}$ . Las ganancias de control para este experimento son  $k_i = 0.7, i = 1, \dots, 6$ .

La Fig. 7 muestra la gráfica de posición de los seis agentes en el plano. Similar al Experimento 1, cada agente tiene una posición inicial diferente a la formación deseada, conforme transcurre el tiempo cada agente trata de reposicionarse con respecto a su robot vecino y los errores logran converger exponencialmente a cero en  $t = 22 \text{ s}$ , Fig. 8.

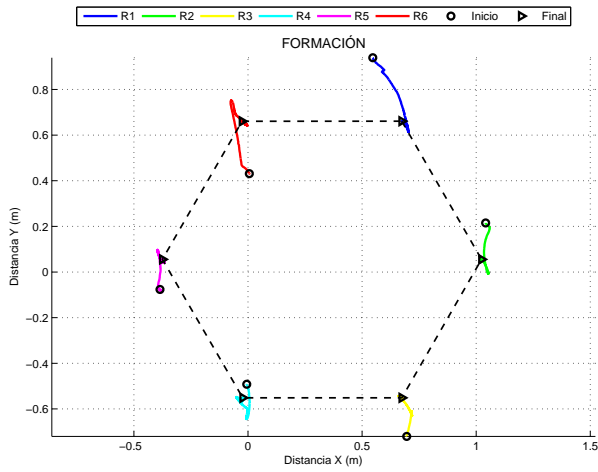


Fig. 4. Formación en el Experimento 1.

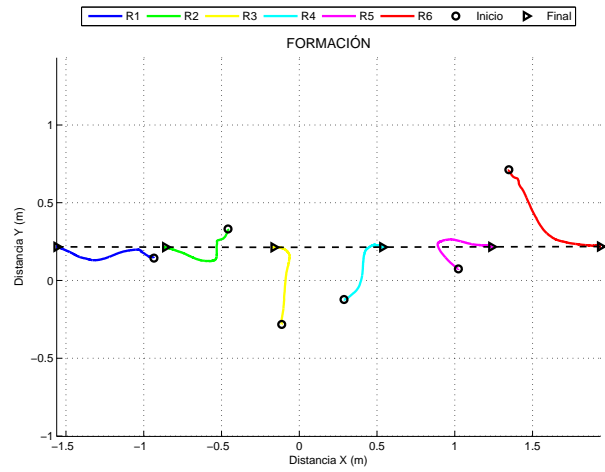


Fig. 7. Formación en el Experimento 2.

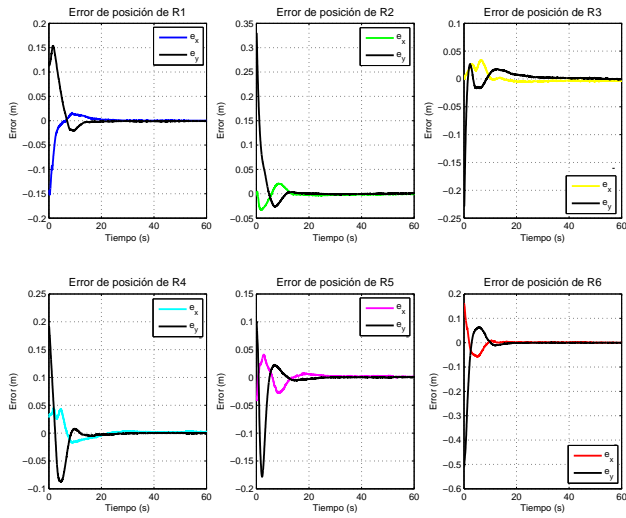


Fig. 5. Errores del Experimento 1.

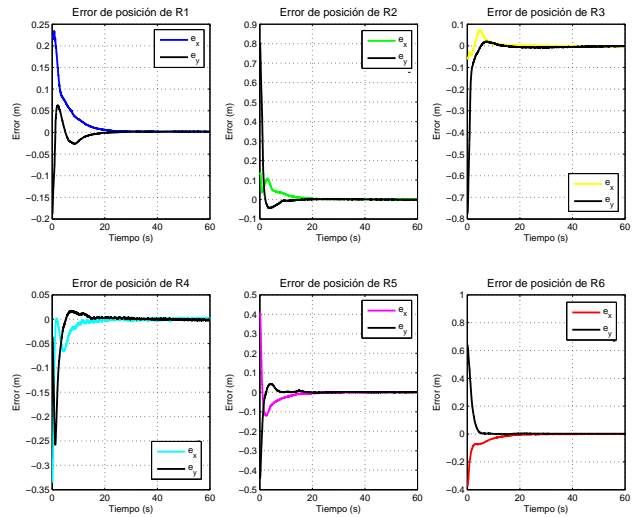


Fig. 8. Errores del Experimento 2.

**Experimento 3:** Para el control de marcha, la trayectoria deseada está definida por:  $m(t) = [0.9 \cos(\frac{4\pi t}{T}) \cos(\frac{2\pi t}{T}), 0.9 \sin(\frac{4\pi t}{T}) \sin(\frac{2\pi t}{T})]^T$ . La Fig. 9 muestra el grafo de formación del robot líder con respecto a sus seguidores con:  $c_{61} = [-d_p, -d_m]$ ,  $c_{62} = [d_p, -d_m]$ ,  $c_{63} = [-d_c, -d_c]$ ,  $c_{64} = [0, -d_c]$ ,  $c_{65} = [d_c, -d_c]$ , siendo  $d_p = 0.30 \text{ m}$ ,  $d_m = 0.15 \text{ m}$  y  $d_c = 0.4 \text{ m}$ . El líder deberá seguir la trayectoria deseada mientras los agentes logran mantener la formación deseada respecto al líder. Las ganancias de control utilizadas son  $k_1 = k_6 = 1.23$ ,  $k_2 = k_4 = 1.53$ ,  $k_3 = 2.1$ ,  $k_5 = 1.26$ , con un periodo  $T = 60 \text{ s}$ .

En la Fig. 10 se puede observar la trayectoria del robot líder

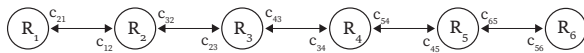


Fig. 6. Grafo empleado en el Experimento 2.

con respecto a la trayectoria deseada y la formación de los agentes seguidores con respecto del líder en distintos instantes de tiempo. Los robots inician fuera de sus posiciones deseadas. El agente líder alcanza a la trayectoria deseada, mientras los seguidores logran la formación deseada con respecto al líder. En la Fig. 11 se puede apreciar que en  $t = 0 \text{ s}$  todos los agentes poseen condiciones iniciales distintas a la deseada y conforme transcurre el tiempo, el agente líder converge a la trayectoria deseada haciendo que los errores decrezcan de forma exponencial, de igual forma, los errores de los seguidores disminuyen de forma exponencial. En ambos casos, tanto los errores del agente líder como de los seguidores se mantienen en una vecindad cercana a cero. Se proporciona un enlace para acceder a todos los videos de los experimentos realizados en este trabajo: [https://drive.google.com/drive/folders/1VdJ1dTJFK7eCzdd\\_2EQ-b7gisMTvwavT?usp=sharing](https://drive.google.com/drive/folders/1VdJ1dTJFK7eCzdd_2EQ-b7gisMTvwavT?usp=sharing).

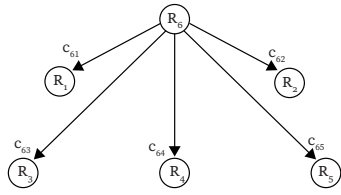


Fig. 9. Grafo del Experimento 3.

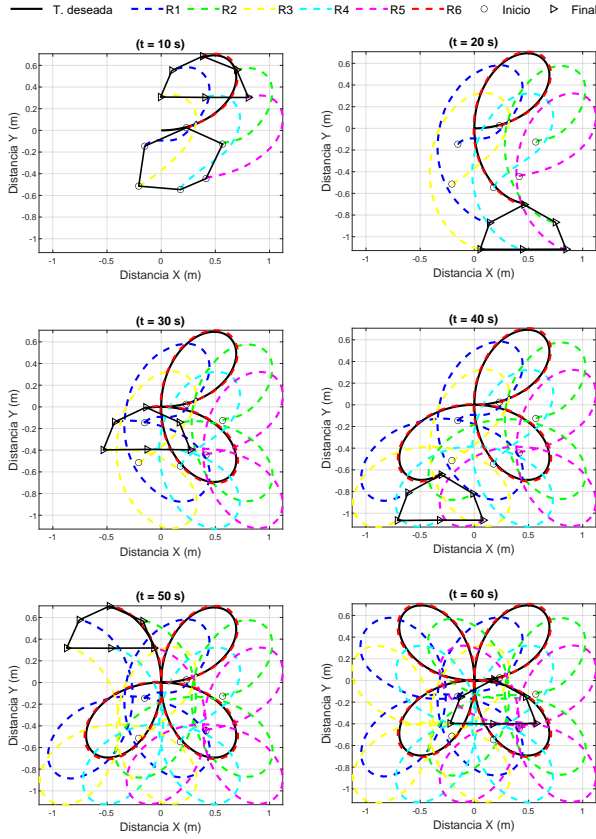


Fig. 10. Trayectorias del agente líder y seguidores.

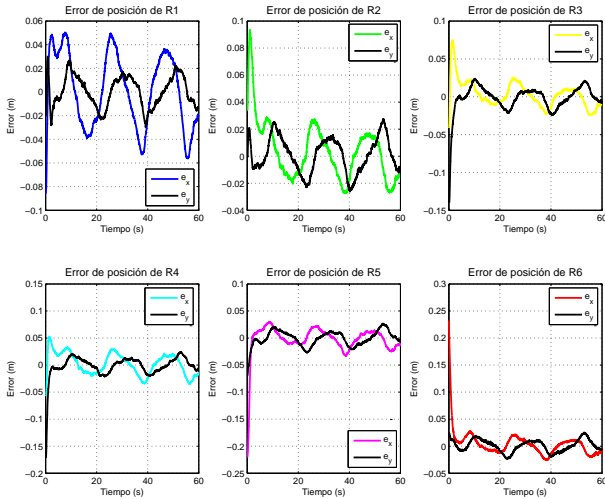


Fig. 11. Errores del Experimento 3.

## VI. CONCLUSIONES

En este trabajo se presentaron dos estrategias de control para lograr la formación y el seguimiento en formación o marcha en un sistema multiagente compuesto por seis robots móviles diferenciales. Para el modelado del sistema multiagente, se empleó la teoría de grafos para representar las interacciones entre los agentes. Para el diseño de las estrategias de control, se consideró como salida a controlar un punto frontal en cada robot. Para el control de formación se emplearon grafos directos e indirectos, mientras que para la marcha se utilizó un grafo centrado en un líder, donde el líder sigue una trayectoria predefinida mientras que el resto logra una formación con respecto al líder. Se presenta un análisis de estabilidad con el cual se garantiza la convergencia exponencial del error mediante las propiedades de la matriz Laplaciana.

Los resultados teóricos se validaron experimentalmente. En los experimentos de formación, las gráficas de posición y error mostraron que los agentes convergen a la formación deseada en aproximadamente 20 segundos. Para el experimento de marcha, el líder converge a una trayectoria deseada mientras los seguidores mantienen la formación deseada respecto al líder. Como trabajo a futuro, se plantea utilizar formaciones variantes en el tiempo y añadir otros tipos de agentes, en particular drones.

## REFERENCES

- [1] C. Peñaranda Cebrián, *Un sistema multiagente para mejorar la localización de un robot NAO*, 2016.
- [2] M. Méndez Martínez, *Control de vuelo en configuración líder-seguidor para aeronaves tipo cuadricóptero*, Master's thesis, Centro de Investigación y de Estudios Avanzados del IPN, Departamento de Ingeniería Eléctrica/Sección de Mecatrónica, 2015.
- [3] J. L. Alfonso-Jimenez, J. Santiaguillo-Salinas, H. N. Garcia-Lozano, and N. Perez-Castro, "Seguimiento en formación para un sistema multiagente heterogéneo," *Pápi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, vol. 12, no. Especial2, pp. 8-13, 2024.
- [4] B. B. I. R., and U. Del Bío-Bío Escuela de Pedagogía en Educación Matemática, "Teoría de grafos," 2013. [Online]. Available: <http://repobib.ubiobio.cl/jspui/handle/123456789/1953>. [Accessed: Sept. 26, 2024].
- [5] W. O. Quesada Moncayo, *Generación de comportamientos de enjambre en robots móviles a través del uso del aprendizaje por refuerzo*, Doctoral dissertation, Universidad Santo Tomás, 2018.
- [6] J. González-Sierra, E. G. Hernandez-Martinez, E. D. Ferreira-Vazquez, J. J. Flores-Godoy, G. Fernandez-Anaya, and P. Paniagua-Contro, "Leader-follower control strategy with rigid body behavior," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 184-189, 2018.
- [7] A. A. Bergfjød, *Formation control: Comparing Leader-Follower and Graph Rigidity Approaches on Differential-Drive Mobile Ground Robots*, Master's thesis, 2022.
- [8] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert, "Multi-robot cooperation in the MARTHA project," *IEEE Robotics & Automation Magazine*, vol. 5, no. 1, pp. 36-47, 1998.
- [9] I. Rekleitis, R. Sim, G. Dudek, and E. Milius, "Collaborative exploration for the construction of visual maps," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2001, vol. 3, pp. 1269-1274.
- [10] J. L. Farrugia, and S. G. Fabri, "Swarm robotics for object transportation," in *UKACC Int. Conf. Control*, 2018, pp. 353-358.
- [11] D. Zermas, "Control of a leader-follower mobile robotic swarm based on the NXT educational LEGO platform," in *IEEE Int. Symp. Industrial Electronics*, 2011, pp. 1381-1386.
- [12] E. G. Hernandez-Martinez, and E. Aranda-Bricaire, "Marching control of unicycles based on the leader-followers scheme," in *IEEE Int. Conf. Industrial Electronics*, 2009, pp. 2265-2270.

- [13] C. Xu, Y. Zheng, H. Su, and H. B. Zeng, "Containment for linear multi-agent systems with exogenous disturbances," *Neurocomputing*, vol. 160, pp. 206-212, 2015.
- [14] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer, 2012.
- [15] E. G. Hernandez Martinez, and E. Aranda Bricaire, "Decentralized formation control of multi-agent robot systems based on formation graphs," 2012.
- [16] H. E. Bell, "Gershgorin's theorem and the zeros of polynomials," *The American Mathematical Monthly*, vol. 72, no. 3, pp. 292-295, 1965.

# Sobre regulación de posición articular de robots manipuladores

Javier Moreno

*Instituto Politécnico Nacional–CITEDI*  
Tijuana, Baja California, 22435, México  
moreno@citedi.mx  
<https://orcid.org/0000-0003-0670-5979>

Gabriela Zepeda

*Facultad de Ingeniería*  
*Universidad Autónoma de Baja California*  
Mexicali, Baja California, 21280, México  
zepedavalencia93@gmail.com  
<https://orcid.org/0000-0002-5272-0750>

**Resumen**—Este artículo presenta un nuevo método de ajuste de ganancias de un controlador no lineal en la parte proporcional, lineal en parte derivativa derivado, con compensación de gravedad para robots manipuladores con fricción viscosa. El análisis del sistema de lazo cerrado se lleva a cabo utilizando una función de Lyapunov. Los resultados numéricos de un manipulador robótico de tres grados de libertad respaldan los resultados teóricos.

**Palabras clave**—Manipulador robótico, estabilidad, aplicaciones, actuador, simulación numérica.

## I. INTRODUCTION

Estudios sobre la regulación de manipuladores y servomecanismos saturados de entrada, que en realidad están definidos por las ecuaciones de Euler-Lagrange, son examinados a continuación. El principio de optimalidad de Lyapunov para la creación de controladores saturados para sistemas no lineales (que incluyen algunos sistemas de Euler-Lagrange) se utilizó en [1]. El impacto de la saturación del actuador en el rendimiento de seguimiento de los sistemas servo fue explorado en [2]. Los autores de [3] presentaron un controlador proporcional-derivativo saturado que asegura el seguimiento de la trayectoria de un manipulador. En un desarrollo más reciente, [4] formuló un algoritmo no lineal que se basa en la saturación de los componentes de la acción de control. Otro antecedente es presentado por una clase de controladores acotados propuestos en [5].

El control proporcional-derivativo (PD) con compensación de gravedad fue introducido en los trabajos seminales [6], [7]. Nótese que en el contexto de los sistemas generales de Euler-Lagrange, la compensación de gravedad corresponde al gradiente de energía potencial, ya que son sistemas generalizados. Para controladores proporcional-derivativo (PD) con compensación de gravedad, se requiere conocimiento de la estructura y los parámetros del término de gravedad del sistema. Esta estructura de control ha sido útil para muchos investigadores en la prueba de mecanismos robóticos y como base para nuevas investigaciones teóricas. Se revisan algunos trabajos. En [8], se examinó una clase de controladores con compensación de gravedad, basados en el paradigma de modelado de energía, para generar funciones de Lyapunov. En [9] se desarrolló un controlador de tipo PD no lineal, donde las ganancias son funciones no lineales. Esto se hizo para mantener la acción de control dentro de las capacidades

máximas de cada actuador, y se logró un control de posición global. En [10] se logró una regulación global a través del principio de invariancia de LaSalle utilizando un PD con un controlador de compensación de gravedad deseado cuando se aplica a un manipulador de robot con una no linealidad de entrada suavizada. El trabajo reportado en [11] presentó un método para ajustar las ganancias del controlador mencionado. En [12], se aplicó un PD con un controlador de compensación de gravedad a un robot manipulador de cuatro grados de libertad.

Las limitaciones de los actuadores robóticos que utilizan la estructura PD con compensación de gravedad se han considerado en algunos trabajos [13], [10], [14]. Sin embargo, un problema aún por resolver implica tener en cuenta las funciones de saturación dura que representan las limitaciones del actuador en el análisis del sistema de lazo cerrado, bajo el supuesto de que el controlador no tiene restricciones, como se hace típicamente en la práctica.

Este artículo examina el controlador proporcional no lineal derivativo lineal con compensación de gravedad introducido en [8], pero, a diferencia de la literatura conocida, se consideran manipuladores robóticos saturados de entrada con fricción viscosa. En otras palabras, el manipulador robótico presenta una no linealidad de saturación dura de entrada, mientras que el controlador no tiene restricciones. Además, se supone que el manipulador robótico está perturbado por una perturbación constante acoplada, lo que implica que el punto de equilibrio de lazo cerrado está desplazado del origen. Bajo un cambio de coordenadas, el equilibrio se desplaza al origen del espacio de estados. La estabilidad asintótica global del punto de equilibrio se muestra utilizando una función de Lyapunov estricta, evitando así la necesidad de utilizar el principio de invariancia como se hizo en trabajos anteriores, incluidos aquellos que tratan la saturación de entrada. Se presenta un estudio de simulación en un manipulador de tres grados de libertad, que confirma los resultados teóricos.

En este artículo, se utilizará la siguiente notación. Las letras minúsculas en negrita denotan vectores, por ejemplo,  $\mathbf{x} \in \mathbb{R}^n$ . La notación  $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$  representa la norma del vector  $\mathbf{x} \in \mathbb{R}^n$ . Las letras mayúsculas en cursiva denotan matrices, por ejemplo,

## II. DINÁMICA DEL MANIPULADOR, ANTECEDENTES Y OBJETIVO DE CONTROL

### II-A. Dinámica del manipulador

El comportamiento de un manipulador robótico de cadena serial con articulación giratoria con  $n$  eslabones se puede expresar de la siguiente manera [15], [16]:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + F_v\dot{\mathbf{q}} = \boldsymbol{\tau}(\mathbf{u}) - \boldsymbol{\delta}, \quad (1)$$

donde  $M(\mathbf{q})$  denota la matriz de inercia  $n \times n$ ,  $C(\mathbf{q}, \dot{\mathbf{q}})$  significa la matriz  $n \times n$  de pares de Coriolis y centrífugos,  $\mathbf{g}(\mathbf{q})$  representa el vector  $n \times 1$  de pares gravitacionales,  $F_v$  es una matriz diagonal  $n \times n$  que es definida positiva y contiene el coeficiente de fricción viscosa para cada articulación, y  $\boldsymbol{\delta}$  es un vector  $n \times 1$  que simboliza una perturbación constante. Específicamente,  $\boldsymbol{\tau}(\mathbf{u})$  es el vector  $n \times 1$  de las entradas de par aplicadas, que son generadas por los actuadores.

A pesar de la alta no linealidad de la ecuación de movimiento (1), se puede demostrar que posee características esenciales que se utilizan para simplificar el diseño del sistema de control.

• **Propiedad 1 [16], [17]:** El modelo de robot (1) satisface las siguientes propiedades.

- Para un robot que solo tiene articulaciones giratorias, el vector de par gravitacional  $\mathbf{g}(\mathbf{q})$  tiene un límite superior para todos los  $\mathbf{q} \in \mathbb{R}^n$ . Esto implica la existencia de constantes finitas  $k'_{gi}$  tales que

$$\sup_{\forall \mathbf{q}} |g_i(\mathbf{q})| \leq k'_{gi} \quad (2)$$

para  $i = 1, 2, \dots, n$ , con  $g_i(\mathbf{q})$  representando el  $i$ -ésimo componente de  $\mathbf{g}(\mathbf{q})$ .

- La matriz de inercia es definida positiva y acotada.
- Dado que el cálculo de  $C(\mathbf{q}, \dot{\mathbf{q}})$  involucra el primer tipo de símbolos de Christoffel, se deduce que la matriz

$$\frac{1}{2}\dot{M}(\mathbf{q}) - C(\mathbf{q}, \dot{\mathbf{q}}) \quad (3)$$

exhibe antisimetría. Además,

$$\dot{M}(\mathbf{q}) = C(\mathbf{q}, \dot{\mathbf{q}}) + C(\mathbf{q}, \dot{\mathbf{q}})^T. \quad (4)$$

- Para todos los  $\mathbf{q}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ ,

$$C(\mathbf{q}, \mathbf{x})\mathbf{y} = C(\mathbf{q}, \mathbf{y})\mathbf{x}. \quad (5)$$

□

### II-B. Modelo de actuador y antecedentes útiles

Para cada  $u_i$  que pertenece a  $\mathbb{R}$  y una constante  $u_i^{max}$  que es mayor que 0, la siguiente notación

$$\text{sat}(u_i) = \min\{\max\{u_i, -u_i^{max}\}, u_i^{max}\} \quad (6)$$

es representativa de la función de saturación dura simétrica. Cuando se expresa como una función vectorial,

$$\text{sat}(\mathbf{u}) = [\text{sat}(u_1) \ \dots \ \text{sat}(u_n)]^T.$$

Dado que el par de entrada del robot está limitado por la capacidad de potencia de los actuadores, se puede expresar como

$$\boldsymbol{\tau}(\mathbf{u}) = \text{sat}(\mathbf{u}), \quad (7)$$

donde  $\mathbf{u} \in \mathbb{R}^n$  se entiende como el par de control deseado, sin tener en cuenta la saturación del actuador. El modelo (7) incorpora el modelo del actuador que se está examinando en este artículo.

**Propiedad 2:** Considere la función de saturación dura  $\text{sat}(u_i)$  en (6). Entonces, para todo  $y_i \in \mathbb{R}$  y  $|w_i^*| \leq u_i^{max} - \varepsilon$ , se satisface la desigualdad

$$y_i[\text{sat}(y_i + w_i^*) - w_i^*] \geq \psi_i(y_i) \quad (8)$$

, donde  $\psi_i(y_i) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  es una función definida positiva.

*Demostración:* Ver [18]. □

Un ejemplo de una función que satisface la propiedad 3 se da por  $\psi_i(y_i) = \varepsilon \tanh(y_i)^2$ , con  $\varepsilon = u_i^{max} - |w_i^*|$ .

Como preámbulo a la presentación del controlador en estudio, primero se recuerda la función tangente hiperbólica

$$\tanh(x_i), \quad \forall x_i \in \mathbb{R}.$$

Con esta definición, se tiene el siguiente vector:

$$\text{tanh}(\mathbf{x}) = [\tanh(x_1) \ \dots \ \tanh(x_n)]^T.$$

Ahora, se define la función

$$\phi(x_i) = \tanh(x_i + q_i^*) + \alpha_i, \quad \forall x_i \in \mathbb{R}, \quad (9)$$

donde

$$q_i^* = -\tanh^{-1}(\alpha_i)$$

y  $\alpha_i \in \mathbb{R}$  es una constante tal que  $|\alpha_i| < 1$ . La función  $\text{tanh}^*(x_i)$  también se puede ordenar en un vector

$$\boldsymbol{\phi}(\mathbf{x}) = [\phi(x_1) \ \dots \ \phi(x_n)]^T.$$

La siguiente propiedad de  $\phi(x_i)$  está inspirada en una clase de funciones definidas en [14] y [19].

**Propiedad 3.** La función  $\phi(x_i)$  en (9) exhibe las siguientes propiedades

- $\phi(0) = 0$ .
- $\phi(x_i)$  satisface  $\phi(x_i)x_i > 0$  para todo  $x_i \neq 0$ .
- $|\phi(x_i)| \leq k_f$  para todo  $x_i \in \mathbb{R}$ .
- $\phi(x_i)$  es diferenciable para todo  $x_i \in \mathbb{R}$ , es estrictamente creciente, y  $|\partial\phi(x_i)/\partial x_i| \leq k_J$ , para todo  $x_i \in \mathbb{R}$ .
- La función

$$l_i(x_i) = \int_0^{x_i} \phi(x_i) dx_i$$

es definida positiva y radialmente ilimitada.

- Además, siempre existe  $k_{U_i} > 0$  tal que

$$l_i(x_i) \geq \frac{k_{U_i}}{2} \phi(x_i)^2. \quad (10)$$

□

La propiedad 3 se cumple con  $k_f = 2$ ,  $k_J = 1$  y  $\kappa = 1$ . Además,

$$\begin{aligned} l(x_i) &= \int_0^{x_i} \tanh(x_i) dx_i \\ &= \ln(\cosh(x_i + q_i^*)) + \alpha_i x_i - \ln(\cosh(q_i^*)). \end{aligned}$$

Entonces, se tiene  $\ln(\cosh(x_i + q_i^*)) + \alpha_i x_i - \ln(\cosh(q_i^*)) \geq \frac{1}{2}\phi(x_i)^2$ . En consecuencia,  $k_{U_i} = 1$ .

Se puede demostrar la existencia de otras funciones continuas que presentan las mismas características que la función tangente hiperbólica. De hecho, se introduce una generalización de este concepto en la referencia [14].

**Supuesto 1.** Considerando que la propiedad 1, ecuación (2), se satisface, la suma de los términos de gravedad y la perturbación cumple lo siguiente:

$$|g_i(\mathbf{q}) + \delta_i| < u_i^{max}, \quad i = 1, \dots, n, \quad \forall \mathbf{q} \in \mathbb{R}^n, \quad (11)$$

lo que implica

$$sat(\mathbf{g}(\mathbf{q}) + \boldsymbol{\delta}) = \mathbf{g}(\mathbf{q}) + \boldsymbol{\delta}. \quad \square$$

### II-C. Objetivo de control

Suponga que  $\mathbf{q}_d \in \mathbb{R}^n$  es el vector de desplazamiento de la articulación deseado. El error de posición se caracteriza entonces como

$$\tilde{\mathbf{q}}(t) = \mathbf{q}(t) - \mathbf{q}_d. \quad (12)$$

Dados los preliminares indicados, el desafío es formular una entrada de control sin restricciones  $\mathbf{u}(t)$  para el modelo de manipulador saturado de entrada (1)–(7) de modo que el vector de desplazamiento de la articulación  $\mathbf{q}(t) \in \mathbb{R}^n$  converja asintóticamente al vector de desplazamiento de la articulación constante deseado  $\mathbf{q}_d \in \mathbb{R}^n$ , es decir,

$$\lim_{t \rightarrow \infty} \tilde{\mathbf{q}}(t) = \mathbf{q}^*, \quad (13)$$

con  $\tilde{\mathbf{q}}(t)$  definido en (12) y  $\mathbf{q}^*$  una constante que depende de la perturbación constante  $\boldsymbol{\delta}$  y las ganancias de control. Si  $\boldsymbol{\delta} = 0$  [Nm], la entrada de control diseñada  $\mathbf{u}(t)$  garantiza

$$\lim_{t \rightarrow \infty} \tilde{\mathbf{q}}(t) = 0. \quad (14)$$

En otras palabras, la entrada de control  $\mathbf{u}(t)$  que se va a diseñar debe garantizar cierto grado de robustez ante perturbaciones constantes acopladas y al mismo tiempo debe garantizar una regulación global cuando no haya perturbaciones.

### III. CONTROL NO LINEAL EN LA PARTE PROPORCIONAL, LINEAL EN PARTE DERIVATIVA, Y CON COMPENSACIÓN DE GRAVEDAD

Con el fin de estabilizar la dinámica del sistema (1), que contiene una función de saturación dura de entrada y una perturbación constante, se considera el algoritmo de control dado por [8]

$$\mathbf{u} = -K_p \mathbf{tanh}(\tilde{\mathbf{q}}) - K_d \dot{\tilde{\mathbf{q}}} + \mathbf{g}(\mathbf{q}), \quad (15)$$

donde  $\mathbf{tanh}(\tilde{\mathbf{q}})$  es una función vectorial cuyos elementos satisfacen la propiedad 3, mientras que  $K_p$  y  $K_d$  son matrices definidas positivas diagonales. Defina una matriz definida positiva  $K = diag\{k_i\}$ , y constantes estrictamente positivas  $\epsilon$  y  $\beta_1$  de modo que

$$\begin{aligned} K_p &= \epsilon K, \\ K_d &= \beta_1 K. \end{aligned} \quad (16)$$

El diseño del controlador (15) se alinea con un algoritmo de compensación gravitacional que además tiene una parte no lineal proporcional y otra lineal derivativa. De hecho, la ley de control (15) fue introducida en el manuscrito [8], como un caso particular de una clase de reguladores motivados por el método de diseño de moldeo de energía.

En primer lugar, al sustituir el controlador (15) en el modelo de manipulador (1) se obtiene el sistema de lazo cerrado. Bajo el supuesto 1, el equilibrio es

$$\begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} -\mathbf{tanh}^{-1}(\epsilon^{-1} K^{-1} \boldsymbol{\delta})^T \\ 0 \end{bmatrix}. \quad (17)$$

Observe que la ganancia  $K$  debe ser lo suficientemente grande para garantizar la existencia del punto de equilibrio. En concreto, la desigualdad matricial

$$I - \epsilon^{-1} K^{-1} \boldsymbol{\delta} > 0 \quad (18)$$

debe cumplirse.

Se define

$$\mathbf{q}^* = -\mathbf{tanh}^{-1}(\epsilon^{-1} K^{-1} \boldsymbol{\delta}) \quad (19)$$

y el cambio de variable

$$\mathbf{x} = \tilde{\mathbf{q}} - \mathbf{q}^*.$$

Por lo tanto, se tiene la función

$$\phi(\mathbf{x}) = \mathbf{tanh}(\mathbf{x} + \mathbf{q}^*) + \epsilon^{-1} K^{-1} \boldsymbol{\delta},$$

cuyos componentes  $\phi(x_i)$  satisfacen realmente la propiedad 3.

**Proposición 1:** *El punto de equilibrio (17) resultante del sistema de lazo cerrado dado por el manipulador (1), y el controlador (15), es globalmente asintóticamente estable siempre que se cumpla el supuesto 1, la desigualdad (18), y se logren las siguientes desigualdades matriciales:*

$$\beta_1 K_u F_v - \epsilon M(\mathbf{q}) > 0, \quad (20)$$

y teniendo en cuenta que  $\mathbf{q} = \mathbf{q}_d + \mathbf{x} - \mathbf{q}^*$ ,

$$\begin{aligned} D_v(\mathbf{x}) &= \text{Sym}\{\beta_1 F_v - \epsilon M(\mathbf{q}) \phi'(\mathbf{x}) \\ &\quad - \epsilon C(\mathbf{q}, \phi(\mathbf{x}))\} > 0, \end{aligned} \quad (21)$$

donde  $\phi'(\mathbf{x}) = \partial \phi(\mathbf{x}) / \partial \mathbf{x}$ , y como se explicó anteriormente  $\text{Sym}\{A\} = \frac{1}{2}[A + A^T]$ .

*Demostración:* Ver [18]. □

Cuando no se presenta ninguna perturbación, es decir,  $\boldsymbol{\delta} = 0$  [Nm], el controlador (15) garantiza el objetivo de control establecido en el límite (14).

**Proposición 2.** *Considerando que no se presenta la perturbación, es decir,  $\boldsymbol{\delta} = 0$ , el punto de equilibrio  $[\tilde{\mathbf{q}}^T \quad \dot{\tilde{\mathbf{q}}}^T]^T = 0$  es globalmente asintóticamente estable.*

*Demostración:* Ver [18]. □

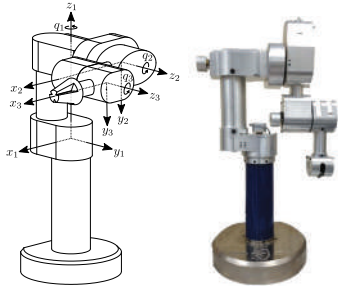


Figura 1. Esquema del manipulador con marcos de referencia de modelado y fotografía en posición inicial.

#### IV. RESULTADOS DE LA SIMULACIÓN

Se han obtenido resultados numéricos con el modelo de un manipulador real de tres grados de libertad. Los marcos de referencia y la fotografía del modelado del manipulador se muestran en la figura 1. Los elementos del modelo del manipulador (1) y los parámetros relacionados se encuentran en el apéndice. Este robot experimental se utilizó en los trabajos anteriores [19] y [22]. Sin embargo, en este artículo mostramos simulaciones numéricas como un paso previo hacia una futura implementación en tiempo real.

A continuación se describen las condiciones en las que se llevaron a cabo las simulaciones. En primer lugar, los límites de par del actuador están dados por  $u_1^{m\acute{a}x} = 10.0$  [Nm],  $u_2^{m\acute{a}x} = 12.5$  [Nm], y  $u_3^{m\acute{a}x} = 3$  [Nm], mientras que la posición deseada de la articulación es

$$\mathbf{q}_d(t) = \begin{bmatrix} 3 \\ \pi/2 + 3 \\ 3 \end{bmatrix} \text{ [rad]}. \quad (22)$$

Se ha usado el siguiente vector de perturbaciones constantes:  $\boldsymbol{\delta} = [5 \ 0.1 \ 0.5]$  [Nm]. Véase el modelo del robot (1). Las ganancias seleccionadas como  $\epsilon = 1$ ,  $\beta = 0.495$  y  $K = \text{diag}\{200, 1000, 300\}$ . Cabe señalar que para el vector de perturbación  $\boldsymbol{\delta}$  y las ganancias seleccionadas para el controlador propuesto (15) satisfacen las condiciones de estabilidad establecidas en las Proposiciones 1 y 2, es decir, se cumplen las desigualdades matriciales (20) y (21). Se selecciona la matriz  $K_u = \text{diag}\{1, 1, 1\}$  en la desigualdad matricial (20). Para la posición de articulación deseada  $\mathbf{q}_d$  en (22) y el modelo de manipulador robótico descrito en el apéndice, se tuvo que diseñar un algoritmo de búsqueda numérica para encontrar los valores apropiados de  $\epsilon$  y  $\beta$  que satisfagan las condiciones de la proposición 1. Por lo tanto, el límite (13) está garantizado globalmente con el controlador propuesto. Todas las simulaciones se llevaron a cabo utilizando las condiciones iniciales  $\mathbf{q}(0) = [0 \ \pi/2 \ 0]^T$  [rad], y  $\dot{\mathbf{q}}(0) = [0 \ 0 \ 0]^T$  [rad/s], lo que significa que las articulaciones 2 y 3 están en posición vertical hacia abajo y el manipulador está en reposo. La figura 2 muestra la evolución temporal de las posiciones de las articulaciones  $q_1(t)$ ,  $q_2(t)$  y  $q_3(t)$  obtenidas numéricamente con el controlador (15). Las acciones de control aplicadas  $\text{sat}(u_1(t))$ ,  $\text{sat}(u_2(t))$  y  $\text{sat}(u_3(t))$  se muestran en la figura 3. El controlador (15) produce una acción de control más allá

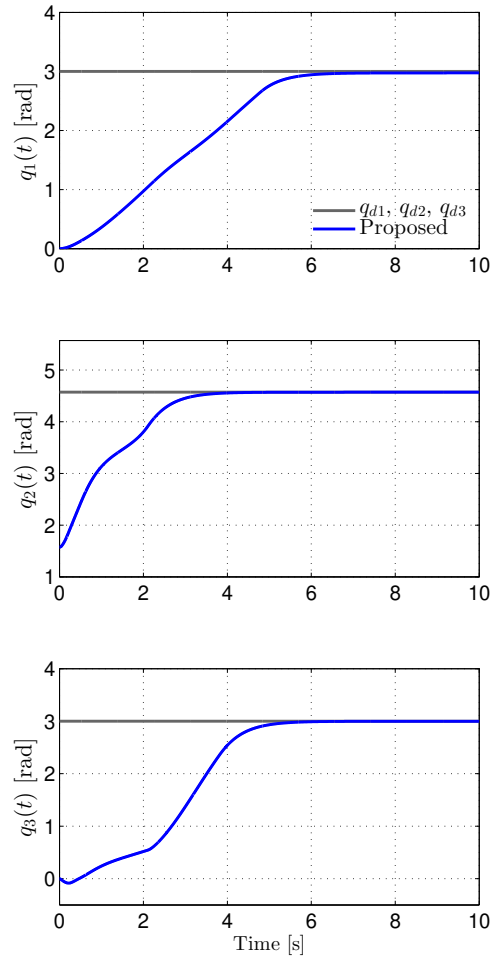


Figura 2. **Resultados de la simulación numérica:** Evolución temporal de las posiciones articulares  $q_1(t)$ ,  $q_2(t)$  y  $q_3(t)$  con la técnica de control (15).

de la saturación. Sin embargo, dado que se cumplen todas las condiciones establecidas en la proposición 1, se satisface el límite (13) con  $\mathbf{q}^*$  definido en (19). Esto implica que se presenta un error de posición en estado estacionario. Debido al alto valor de la ganancia  $K$ , es despreciable. Se puede afirmar que los resultados numéricos presentados son consistentes con la teoría introducida presentada en este documento.

#### V. CONCLUSIONES

En este documento se ha discutido un controlador de tipo PD con compensación de gravedad para robots manipuladores con entrada saturados. El precio pagado para demostrar la estabilidad asintótica global del punto de equilibrio a través de una función de Lyapunov estricta fue la complejidad del análisis. Sin embargo, las condiciones de estabilidad resultantes fueron fáciles de satisfacer. La teoría se validó aplicando el controlador a un sistema manipulador de tres grados de libertad. Se cumplieron todas las condiciones requeridas para la estabilidad y los resultados numéricos fueron satisfactorios.

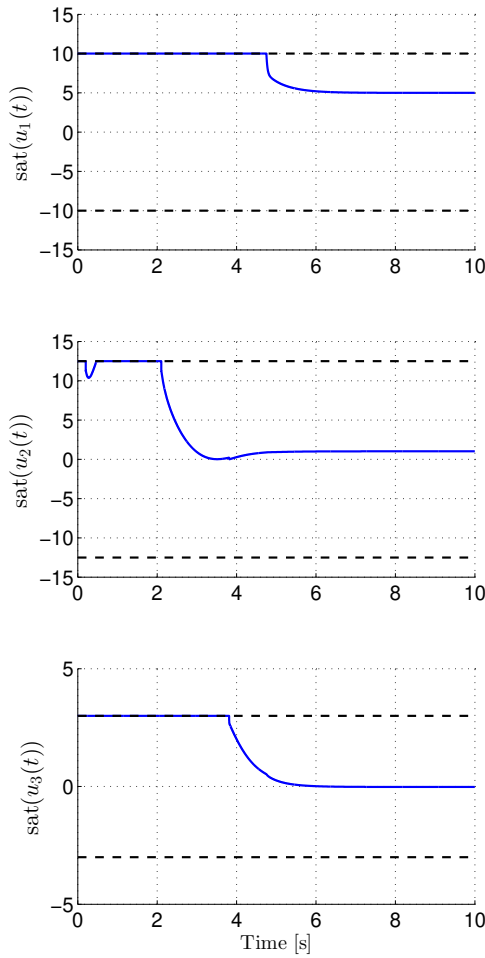


Figura 3. **Resultados de la simulación numérica:** Evolución temporal de la acción de control  $sat(u_1(t))$ ,  $sat(u_2(t))$  y  $sat(u_3(t))$  para la técnica de control (15).

## REFERENCIAS

- [1] R. D. Robinett, G. G. Parker, H. Schaub and J. L. Junkins, "Lyapunov optimal saturated control for nonlinear systems," *J. Guid. Control Dyn.*, vol. 20, no. 6, pp. 1083-1088, 1997.
- [2] M. Goldfarb and T. Sirithanapipat, "The effect of actuator saturation on the performance of PD-controlled servo systems," *Mechatronics*, vol. 9, no. 5, pp. 497-511, 1999.
- [3] Y. Su and C. Zheng, "A saturated PD plus scheme for asymptotic tracking of robot manipulators", in *Proc. Int. Conf. Robot. Biomimetics, December*, Guilin, China, 2009, pp. 853-858.
- [4] Y. Guo, B. Huang, A. J. Li and C. Q. Wang, "Integral sliding mode control for Euler-Lagrange systems with input saturation," *Int. J. Robust Nonlinear Control*, vol. 29, no. 4, 1088-1100, 2019.
- [5] E. Clemente, M. C. Rodríguez-Liñán, M. Meza-Sánchez, L. Monay-Arredondo, L. Herrera, "A class of bounded and partially bounded nonlinear controllers for first and second order dynamical systems," *IEEE Control Systems Letters*, vol. 6, pp. 1028-1033, 2022.
- [6] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *J. Dyn. Syst. Meas. Contr.*, vol. 103, pp. 119-125, June 1981.
- [7] S. Arimoto and F. Miyazaki, "Stability and robustness of PD feedback control with gravity compensation for robot manipulators," in F. Paul and D. Youcef-Toumi, Eds., *Robotics: Theory and Applications*, DSC-vol. 3. New York: Amer. Soc. Mechanical Engineers, 1986.
- [8] V. Santibáñez and R. Kelly, "Strict Lyapunov Functions for Global Regulation of Robot Manipulators," in *Proc. of the IEEE International*

*Conference on Robotics and Automation*, vol. 1. no. 3. 21-27 May 1995, pp. 2758-2763.

- [9] R. Kelly and R. Carelli, "A class of nonlinear PD-type controllers for robot manipulators," *J. Robotic. Syst.*, vol. 13, no. 13, pp. 793-802, 1996.
- [10] A. Zavala-Río and V. Santibáñez, "A natural saturating extension of the PD-with-desired-gravity-compensation control law for robot manipulators with bounded inputs," *IEEE Trans. Robot.*, vol. 23, pp. 386-391, 2007.
- [11] A. Perrusquia, J. A. Flores-Campos and C. R. Torres-San-Miguel, "A novel tuning method of PD with gravity compensation controller for robot manipulators", *IEEE Access*, vol. 8, pp. 114773-114783, 2020.
- [12] A. Dadbin, A. Kalhor, and M. T. Masouleh, "A comparison study on the dynamic control of OpenMANIPULATOR-X by PD with gravity compensation tuned by oscillation damping based on the phase-trajectory-length concept," in *Proc. of the 2022 8th International Conference on Control, Instrumentation and Automation (ICCIA)*, March 2022, pp. 1-7.
- [13] V. Santibáñez and R. Kelly, "On global regulation of robot manipulators: Saturated linear state feedback and saturated linear output feedback," *European Journal of Control*, vol. 3, no. 2, pp. 104-113, 1997.
- [14] J. Moreno-Valenzuela, J. Moyrón, M. Martínez-Lopez, and M. Jiménez-Quiroz, "Global asymptotic stability of input-saturated one degree-of-freedom Euler-Lagrange systems with Rayleigh dissipation under nonlinear control," *International Journal of Control*, vol. 96, no. 8, pp. 2100-2112, 2023.
- [15] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*, London: Springer-Verlag, 2000.
- [16] R. Kelly, V. Santibáñez, and A. Loría, *Control of Robot Manipulators in Joint Space*. London: Springer-Verlag, 2005.
- [17] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Dynamics and Control*, 2nd ed., Hoboken, NJ: John Wiley & Sons, Inc., 2020.
- [18] J. Moreno y G. Zepeda, *Reporte interno: Sobre regulación de posición articular de robots manipuladores*, disponible a solicitud, octubre de 2024.
- [19] J. Moreno-Valenzuela, J. Moyrón, and J. Montoya-Cháirez, "Limited integrator anti-windup-based control of input-constrained manipulators," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 2, pp. 1738 - 1748, 2024.
- [20] H. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River: Prentice-Hall, 2002.
- [21] R. Sepulchre, M. Jankovic, and P. V. Kokotovic, *Constructive Nonlinear Control*, London: Springer-Verlag, 1997.
- [22] J. Moreno-Valenzuela, M. Martínez-Lopez, and J. Sandoval, "Static Anti-Windup for Global Regulation of Constrained Euler-Lagrange Systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, to appear, 2024.

## APÉNDICE: DESCRIPCIÓN DEL MANIPULADOR ROBÓTICO

En este apéndice, se describe el modelo del manipulador experimental utilizado para llevar a cabo las simulaciones. Los elementos de la matriz de inercia están dados por

$$\begin{aligned}
 M_{11} &= \theta_1 + \theta_3 \cos^2(q_2) - \theta_2 \sin^2(q_2 + q_3) \\
 &\quad + \theta_4 (\cos(2q_2 + q_3) + \cos(q_3)), \\
 M_{12} &= \theta_7 \sin(q_2 + q_3) + \theta_6 \sin(q_2), \\
 M_{13} &= \theta_7 \sin(q_2 + q_3), \\
 M_{22} &= \theta_5 + \theta_8 + 2\theta_4 \cos(q_3), \\
 M_{23} &= \theta_8 + \theta_4 \cos(q_3), \\
 M_{33} &= \theta_8.
 \end{aligned}$$

Dado que  $M(\mathbf{q})$  es simétrico,  $M_{12} = M_{21}$ ,  $M_{13} = M_{31}$  y  $M_{32} = M_{23}$ . Los elementos de la matriz de fuerzas centrífugas

Tabla I  
PARÁMETROS DEL MANIPULADOR EXPERIMENTAL EN LA FIGURA 1  
USADO PARA REALIZAR LAS SIMULACIONES NUMÉRICAS.

Parámetro	Valor	Unidad	Parámetro	Valor	Unidad
$\theta_1$	3.3002	$kg\ m^2$	$\theta_8$	0.2838	$kg\ m^2$
$\theta_2$	0.7924	$kg\ m^2$	$\theta_9$	1.0357	$kg\ m$
$\theta_3$	-0.0554	$kg\ m^2$	$\theta_{10}$	0.1888	$kg\ m$
$\theta_4$	0.1279	$kg\ m^2$	$\theta_{11}$	8.3387	$kg\ m^2/s$
$\theta_5$	0.3441	$kg\ m^2$	$\theta_{12}$	2.1720	$kg\ m^2/s$
$\theta_6$	0.0556	$kg\ m^2$	$\theta_{13}$	3.5735	$kg\ m^2/s$
$\theta_7$	0.0498	$kg\ m^2$	$g$	9.81	$m/s^2$

y de Coriolis son

$$\begin{aligned}
 C_{11} &= -\dot{q}_3(\theta_4(\sin(2q_2 + q_3) + \sin(q_3)) \\
 &\quad + \theta_2 \sin(2q_2 + 2q_3))/2 \\
 &\quad - \dot{q}_2((\theta_2 \sin(2q_2 + 2q_3))/2 + \theta_4 \sin(2q_2 + q_3)) \\
 &\quad - \dot{q}_2 \theta_3 \cos(q_2) \sin(q_2), \\
 C_{12} &= -\dot{q}_1((\theta_2 \sin(2q_2 + 2q_3))/2 + \theta_4 \sin(2q_2 + q_3)) \\
 &\quad - \dot{q}_2 \theta_3 \cos(q_2) \sin(q_2) + \dot{q}_3 \theta_7 \cos(q_2 + q_3), \\
 C_{13} &= -\dot{q}_1(\theta_4(\sin(2q_2 + q_3) + \sin(q_3)) \\
 &\quad + \theta_2 \sin(2q_2 + 2q_3))/2 \\
 &\quad + \dot{q}_2 \theta_7 \cos(q_2 + q_3) + \dot{q}_3 \theta_7 \cos(q_2 + q_3), \\
 C_{21} &= \dot{q}_1(\theta_4 \sin(2q_2 + q_3) + \theta_3 \cos(q_2) \sin(q_2)) \\
 &\quad \dot{q}_1(\theta_2 \sin(2q_2 + 2q_3))/2, \\
 C_{22} &= -\theta_4 \dot{q}_3 \sin(q_3), \\
 C_{23} &= -\theta_4 \sin(q_3)(\dot{q}_2 + \dot{q}_3), \\
 C_{31} &= \theta_2 \dot{q}_1 \sin(q_2 + q_3) \cos(q_2 + q_3) \\
 &\quad + \theta_4 \dot{q}_1 (\sin(2q_2 + q_3) + \sin(q_3))/2, \\
 C_{32} &= \theta_4 \dot{q}_2 \sin(q_3), \\
 C_{33} &= 0.
 \end{aligned}$$

El vector de gravedad está dado por

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} 0 \\ -\theta_9 g \cos(q_2) - \theta_{10} g \cos(q_2 + q_3) \\ -\theta_{10} g \cos(q_2 + q_3) \end{bmatrix}.$$

La matriz que contiene los coeficientes de fricción viscosa está dada por  $F_v = \text{diag}\{\theta_{11}, \theta_{12}, \theta_{13}\}$ . Finalmente, los parámetros identificados se dan en la Tabla I.

# Sincronización de 3 sistemas mecánicos de 1 GDL

Ervin Jesús Álvarez-Sánchez  
Facultad de Ingeniería  
Mecánica y Eléctrica  
Universidad Veracruzana  
Xalapa, México  
eralvarez@uv.mx

José Gustavo Leyva-Retureta  
Facultad de Ingeniería  
Mecánica y Eléctrica  
Universidad Veracruzana  
Xalapa, México  
guleyva@uv.mx

Fernando Aldana-Franco  
Facultad de Ingeniería  
Mecánica y Eléctrica  
Universidad Veracruzana  
Xalapa, México  
faldana@uv.mx

Rosario Aldana-Franco  
Facultad de Ingeniería  
Mecánica y Eléctrica  
Universidad Veracruzana  
Xalapa, México  
raldana@uv.mx

**Resumen**—En este trabajo se presenta la implementación de un método de sincronización de sistemas mecánicos de 1 grado de libertad (GDL) aplicada a 3 péndulos con diferencias paramétricas, por lo que sus comportamientos dinámicos son diferentes en amplitud y frecuencia. Los resultados obtenidos demuestran la eficacia de la técnica utilizada, al tener un sistema de referencia y dos sistemas seguidores, lográndose que se tenga una sincronización asintótica aún y cuando se tienen variaciones en la señal de control del sistema de referencia.

**Palabras Clave**—sincronización, sistemas mecánicos, péndulos, 1 gdl

## I. INTRODUCCIÓN

El trabajar con diferentes tipos de sistemas para realizar trabajo colaborativo resulta problemático en la mayoría de las ocasiones, ya que las diferencias paramétricas o el tipo de sistema provocan que se tengan diferencias en las señales de salida que pueden llegar a ser perjudiciales para cumplir con la tarea asignada. Es por esto que en los últimos años se ha trabajado con el enfoque de la sincronización aplicada a distintos campos del conocimiento para evitar los retardos o diferencias entre los sistemas, ya sean biológicos [1], eléctricos [2], vehículos no tripulados [3], caóticos [4] o mecánicos.

La sincronización entre los comportamientos dinámicos de los sistemas es crucial para el funcionamiento adecuado de los dispositivos en cuestión, ya que pueden estar compuestos por múltiples componentes que deben llevar a cabo un trabajo de manera coordinada para lograr un objetivo o una meta en común [5]. Este problema no solo radica en el objetivo de sincronizar los comportamientos de los sistemas, sino que se debe tener un convergencia en tiempo finito hacia el valor o la dinámica deseada [6], por lo que se han utilizado diferentes técnicas para lograrlo, tales como observadores de estado [7], orden reducido [8] y Hamiltoniana [9], por mencionar algunas.

Específicamente en los sistemas mecánicos la importancia de la sincronización entre ellos radica en los diferentes problemas que pueden surgir, ya que se pueden llegar a generar vibraciones que afecten la estabilidad y el funcionamiento del sistema [10], llegando incluso a desgastar o dañar algunos de los componentes, además de que debido a esto puede existir un consumo de energía no esperado, lo que también puede producir un bajo rendimiento.

Trabajos como el presentado en [11] demuestran la manera en que se pueden sincronizar múltiples sistemas mecánicos cuando tienen diferentes parámetros, lo que llega a implicar diferentes comportamientos dinámicos. En [12] se utilizan

péndulos suspendidos sobre una viga para demostrar la forma en que se puede lograr la sincronización entre ellos, tanto en modo espejo como en modo antifase, en ausencia de gravedad, así como en presencia de esta. Por otra parte, trabajos como el presentado en [13] muestran cómo es posible realizar la sincronización de dos péndulos mediante vibradores rotacionales que se encuentran al interior de los péndulos, lográndose una sincronización en fase o en antifase, dependiendo de las condiciones de control de los vibradores, mostrando una posible aplicación en una máquina trituradora, remarcando la importancia de la sincronización de sistemas mecánicos.

En este trabajo se lleva a cabo la implementación de la técnica presentada en [14] con la finalidad de lograr la sincronización de 3 sistemas mecánicos en una configuración referencia-seguidores, se utiliza la estructura matemática para el modelo matemático de un péndulo para cada uno de los sistemas, sin embargo, se imponen variaciones paramétricas entre ellos para poder corroborar que incluso con estas diferencias es posible llevar a cabo la sincronización de sus movimientos angulares. Además, se añade un cambio en la señal de control del péndulo de referencia, ya que al principio se trabaja con una típica señal senoidal continua para posteriormente cambiarla por una señal del tipo escalonada que vuelve a cambiar a una señal senoidal de amplitud distinta a la primera. Lo anterior para incrementar la dificultad de seguimiento de la señal para los péndulos restantes.

Para realizar lo anterior, primero se dan a conocer algunas definiciones al respecto del tipo de sistema a utilizar, así como la manera en que se resuelve el problema de sincronización y la forma en que se diseñan los controles que permiten obtener el seguimiento, para finalmente mostrar simulaciones numéricas de los resultados obtenidos.

## II. DEFINICIONES PRELIMINARES

Considere 3 sistemas mecánicos de 1 grado de libertad, que están descritos por el modelo matemático (1)

$$M_i \ddot{q}_i(t) + C_i \dot{q}_i(t) + G_i q_i(t) = \tau_i(t) + v_i(t) \quad (1)$$

para  $i = 1, 2, 3$ , donde  $q_i(t) \in \mathfrak{R}$  es la posición,  $M_i$  es la constante de inercia,  $C_i$  es el coeficiente de fricción,  $G_i$  es una constante de fuerzas gravitacionales,  $\tau_i(t)$  es la entrada de control que produce un movimiento en el sistema y  $v_i(t)$  es una señal de acoplamiento que permite a los sistemas sincronizarse.

La representación en el espacio de estados de (1) es la mostrada en (2)

$$\frac{d}{dt} \begin{bmatrix} q_i(t) \\ \dot{q}_i(t) \end{bmatrix} = \begin{bmatrix} \dot{q}_i(t) \\ f_i(\cdot) + b_i v_i(t) \end{bmatrix} \quad (2)$$

$$y(t) = q_i(t)$$

donde

$$f_i(\cdot) = M_i^{-1} [\tau_i(t) - C_i \dot{q}_i(t) - G q_i(t)]$$

$$b_i = M_i^{-1}$$

#### A. Problema de sincronización

El problema está definido por la forma en que los 3 sistemas tengan una sincronización asintótica, es decir

$$\lim_{t \rightarrow \infty} \|q_i(t) - q_j(t)\| = 0, \quad \forall i, j \in \{1, 2, 3\}, \quad i \neq j \quad (3)$$

Para resolver este problema se definen las variables de error entre los sistemas como se muestra en (4)

$$e_{i,j}(t) = q_i(t) - q_j(t), \quad i \neq j \quad (4)$$

Con lo que se infiere que el sistema  $i$  debe sincronizarse con el sistema  $j$ .

Por medio de las variables de error se pueden definir el conjunto de funcionales (5)

$$\begin{aligned} \epsilon_1(t) &= \beta_{1,2} e_{1,2}(t) + \beta_{1,3} e_{1,3}(t) \\ \epsilon_2(t) &= \beta_{2,1} e_{2,1}(t) + \beta_{2,3} e_{2,3}(t) \\ \epsilon_3(t) &= \beta_{3,1} e_{3,1}(t) + \beta_{3,2} e_{3,2}(t) \end{aligned} \quad (5)$$

donde los términos  $\beta_{i,j}$  son constantes positivas que son denominadas como fuerzas de acoplamiento entre cada uno de los sistemas.

Si se considera al sistema 1 como el sistema de referencia, su dinámica no se vería afectada por los demás sistemas y sus estados deberían converger hacia el estado de este sistema, por lo que además  $v_1(t) = 0$ .

Sustituyendo (4) en (5) para cada uno de los sistemas a sincronizar con el sistema 1 se tendría (6)

$$\begin{aligned} \epsilon_2(t) &= \beta_{2,1} [q_2(t) - q_1(t)] + \beta_{2,3} [q_2(t) - q_3(t)] \\ \epsilon_3(t) &= \beta_{3,1} [q_3(t) - q_1(t)] + \beta_{3,2} [q_3(t) - q_2(t)] \end{aligned} \quad (6)$$

Definiendo (6) en forma matricial se obtiene una ecuación que describe la funcional para sincronización (7)

$$\eta(t) = \Theta p(t) - R q_1(t) \quad (7)$$

donde

$$\eta(t) = [\epsilon_2(t) \quad \epsilon_3(t)]^T$$

$$p(t) = [q_2(t) \quad q_3(t)]^T$$

$$\Theta = \begin{bmatrix} \beta_{2,1} + \beta_{2,3} & -\beta_{2,3} \\ -\beta_{3,2} & \beta_{3,1} + \beta_{3,2} \end{bmatrix}$$

$$R = \begin{bmatrix} \beta_{2,1} \\ \beta_{3,1} \end{bmatrix}$$

A la matriz  $\Theta$  se le llama matriz de conexión entre los sistemas, lo que permite visualizar la fuerza de conexión para la sincronización entre los mismos.

#### B. Señales de acoplamiento

La dinámica de las variables funcionales para la sincronización está dada por (8)

$$\frac{d}{dt} \begin{bmatrix} \eta(t) \\ \dot{\eta}(t) \end{bmatrix} = \begin{bmatrix} \dot{\eta}(t) \\ \Theta F(\cdot) + \Theta B V(t) - R f_1(\cdot) \end{bmatrix} \quad (8)$$

$$y_\eta(t) = \eta(t)$$

donde

$$F(\cdot) = [f_2(\cdot) \quad f_3(\cdot)]^T$$

$$B = \begin{bmatrix} b_2 & 0 \\ 0 & b_3 \end{bmatrix}$$

$$V(t) = [v_2(t) \quad v_3(t)]^T$$

De (8) es posible visualizar que un control que resuelve el problema y asegura una convergencia asintótica está dado por (9)

$$V(t) = (\Theta B)^{-1} [R f_1(\cdot) - \Theta F(\cdot) - K_2 \dot{\eta}(t) - K_1 \eta(t)] \quad (9)$$

Si se analiza la dinámica en lazo cerrado se tendría (10)

$$\ddot{\eta}(t) + K_2 \dot{\eta}(t) + K_1 \eta(t) = 0 \quad (10)$$

Con lo cual, mediante una selección adecuada de las constantes  $K_1$  y  $K_2$ , se asegura una convergencia de  $\eta(t)$  a cero, ya que al utilizar la solución para una ecuación cuadrática se tienen las raíces (11)

$$\eta_{1,2} = \frac{-K_2 \pm \sqrt{K_2^2 - 4K_1}}{2} \quad (11)$$

Donde si  $K_2 > 2\sqrt{K_1}$  la convergencia a cero es no oscilatoria, además de que dependiendo de las magnitudes también se podría asegurar una velocidad de convergencia.

Al tener una convergencia de  $\eta(t)$  hacia cero en (10) y al evaluar esto en (7) resulta en (12)

$$p(t) = \Theta^{-1} R q_1(t) \quad (12)$$

Lo que implica que los sistemas 2 y 3 tienden hacia el sistema de referencia 1 con la fuerza de conexión dependiendo de los valores  $\beta$  que se elijan.

### III. SINCRONIZACIÓN DE 3 SISTEMAS MECÁNICOS

Para llevar a cabo la sincronización de 3 sistemas mecánicos se utiliza la ecuación proveniente del modelo matemático de un péndulo, el cual está compuesto por una barra y un disco, como se muestra en la Fig. 1.

La barra tiene una longitud  $L$ , un espesor  $h$ , una masa  $m_B$  y una inercia  $I_B$ , mientras que el disco tiene un radio  $r$ , una masa  $m_D$  y una inercia  $I_D$ , todo el sistema cuenta con fricción en el punto de rotación.

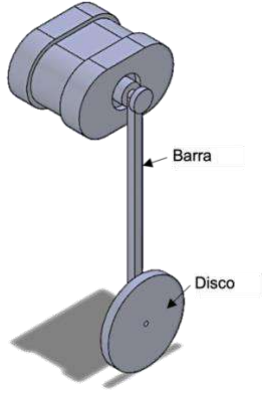


Fig. 1. Sistema pendular utilizado.

El modelo matemático lineal del péndulo está dado por (13)

$$I\ddot{\theta}(t) + b\dot{\theta}(t) + K\theta(t) = \tau(t) \quad (13)$$

donde

$$I = \left(\frac{L^2}{3} + \frac{h^2}{12}\right)m_B + \left(L^2 + \frac{r^2}{2}\right)m_D$$

$$K = gL\left(\frac{m_B}{2} + m_D\right)$$

#### A. Restricciones de los sistemas referencia-seguidores

El péndulo cuya dinámica servirá como referencia será tratado como un péndulo típico que es utilizado en los cursos iniciales de modelado matemático o de análisis de mecanismos. Lo cual aunque es una limitante para su implementación real, sirve para mostrar la diferencia de comportamiento debido a este tipo de suposiciones.

Para este tipo de péndulo la barra es una barra sólida rígida indeformable con una masa y un espesor despreciables ( $m_B = 0 \text{ kg}$ ,  $h = 0 \text{ m}$ ), además el disco tiene un radio también despreciable ( $r = 0 \text{ m}$ ), es decir, que solo se tiene la barra como una restricción mecánica que permite tener un soporte de movimiento para una masa puntual en su extremo opuesto al punto de soporte para rotación, por lo que los términos referidos en (13) relacionados estas condiciones son cero, obteniéndose los coeficientes indicados en (14).

$$\begin{aligned} I_1 &= L^2 m_D \\ K_1 &= gL m_D \end{aligned} \quad (14)$$

Para el primer sistema seguidor se plantea que el disco no exista para el modelo del péndulo, o en otras palabras, que sea de masa y radio despreciable ( $m_D = 0 \text{ kg}$ ,  $r = 0 \text{ m}$ ). Además se considere la barra como una barra delgada homogénea, tomando en cuenta la suposición de que es espesor es despreciable, por lo que  $h = 0 \text{ m}$ , es decir que solo se contemple su longitud, por lo que sus coeficientes quedarían como en (15).

$$\begin{aligned} I_2 &= \frac{L^2}{3} m_B \\ K_2 &= gL\left(\frac{m_B}{2}\right) \end{aligned} \quad (15)$$

El segundo sistema seguidor se mantendría sin restricciones, por lo que las mismas características de los coeficientes serían las mostradas en (16).

$$\begin{aligned} I_3 &= \left(\frac{L^2}{3} + \frac{h^2}{12}\right)m_B + \left(L^2 + \frac{r^2}{2}\right)m_D \\ K_3 &= gL\left(\frac{m_B}{2} + m_D\right) \end{aligned} \quad (16)$$

Utilizando la suposición de que los 3 péndulos tienen un coeficiente de fricción rotacional acorde a sus propias características, junto con las restricciones (14)-(16) y considerando que  $L = 30 \text{ cm}$ ,  $h = 2 \text{ cm}$ ,  $r = 5 \text{ cm}$ ,  $m_B = 160 \text{ g}$ ,  $m_D = 210 \text{ g}$ , se tienen los valores de los coeficientes para cada sistemas mostrados en la tabla 1.

TABLA I. COEFICIENTES DE LOS SISTEMAS

Péndulo	Coeficiente		
	$I \text{ [kg}\cdot\text{m}^2]$	$C \text{ [N}\cdot\text{m}\cdot\text{s]}$	$K \text{ [N}\cdot\text{m}]$
1 (Referencia)	0.0189	0.22	0.61803
2 (Seguidor 1)	0.0048	0.068	0.23544
3 (Seguidor 2)	0.02397	0.29	0.85347

Estas restricciones permiten tener una comparación entre sistemas que cuentan con el mismo modelo matemático, pero con parámetros ligeramente distintos, lo que sucede usualmente cuando se trabaja en el control de sistemas mecánicos, ya sea por diseño, por modificaciones realizadas o por cuestiones estructurales propias de cada fabricante.

#### B. Señales de control

Considerando que entre los sistemas se tiene la misma fuerza de conexión entre ellos, es decir  $\beta_{2,1} = \beta_{2,3} = \beta_{3,1} = \beta_{3,2} = 1$ , la matriz de acoplamiento  $\Theta$  y el vector de acoplamiento  $R$  resultantes se muestran en (17).

$$\begin{aligned} \Theta &= \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \\ R &= [1 \quad 1]^T \end{aligned} \quad (17)$$

Por otra parte, la matriz relacionada con los controles sincronizantes resultante, utilizando los coeficientes de la tabla 1, está dada en (18).

$$B = \begin{bmatrix} (0.0048)^{-1} & 0 \\ 0 & (0.02397)^{-1} \end{bmatrix} \quad (18)$$

Sustituyendo (17) y (18) en (8) se obtienen las expresiones numéricas (19) y (20) de dinámica de las variables funcionales de sincronización.

$$\begin{aligned} \ddot{\eta}_1(t) &= 2f_2(\cdot) - f_3(\cdot) + 416.6667v_1(t) \\ &\quad - 41.7188v_2(t) - f_1(\cdot) \end{aligned} \quad (19)$$

$$\begin{aligned} \ddot{\eta}_2(t) &= -f_2(\cdot) + 2f_3(\cdot) - 208.3333v_1(t) \\ &\quad + 83.4376v_2(t) - f_1(\cdot) \end{aligned} \quad (20)$$

Donde

$$f_1(\cdot) = \left(\frac{1}{0.0189}\right)[\tau_1(t) - 0.22\dot{q}_1(t) - 0.6183q_1(t)]$$

$$f_2(\cdot) = \left(\frac{1}{0.0048}\right)[\tau_2(t) - 0.068\dot{q}_2(t) - 0.23544q_2(t)]$$

$$f_3(\cdot) = \left( \frac{1}{0.02397} \right) [\tau_3(t) - 0.29\dot{q}_3(t) - 0.85347q_3(t)]$$

Las señales de control (21) y (22), se obtienen al sustituir (17) y (18) en (9).

$$v_2(t) = A_1 \begin{pmatrix} f_1(\cdot) - f_2(\cdot) - \frac{2}{3}[K_2\dot{\eta}_1(t) + K_1\eta_1(t)] \\ -\frac{1}{3}[K_2\dot{\eta}_2(t) + K_1\eta_2(t)] \end{pmatrix} \quad (21)$$

$$v_3(t) = A_2 \begin{pmatrix} f_1(\cdot) - f_3(\cdot) - \frac{1}{3}[K_2\dot{\eta}_1(t) + K_1\eta_1(t)] \\ -\frac{2}{3}[K_2\dot{\eta}_2(t) + K_1\eta_2(t)] \end{pmatrix} \quad (22)$$

donde  $A_1 = 0.0048$  y  $A_2 = 0.02397$ .

Las dinámicas en lazo cerrado se obtiene al sustituir (21) y (22) en (19) y (20), de tal manera que se cumple (10), como se muestra en (23)

$$\begin{aligned} \ddot{\eta}_1(t) + K_2\dot{\eta}_1(t) + K_1\eta_1(t) &= 0 \\ \ddot{\eta}_2(t) + K_2\dot{\eta}_2(t) + K_1\eta_2(t) &= 0 \end{aligned} \quad (23)$$

Utilizando los mismos valores  $K_2 = 4$  y  $K_1 = 3$ , para ambas expresiones de (23), se logra asegurar la convergencia a cero en las dinámicas  $\eta_1(t)$  y  $\eta_2(t)$ . Además de acuerdo con (12) se tiene (23).

$$\begin{bmatrix} q_2(t) \\ q_3(t) \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} q_1(t) \quad (23)$$

Lo que demuestra que  $q_3(t) = q_2(t) = q_1(t)$  y que por lo tanto los 3 sistemas están sincronizados, ya que sus posiciones angulares son iguales, teniendo en cuenta que  $q_1(t) = \theta_1(t)$ , teniendo en cuenta que  $q_2(t) = \theta_2(t)$  y que teniendo en cuenta que  $q_3(t) = \theta_3(t)$ .

### C. Simulaciones numéricas

Las simulaciones se llevan a cabo en el software Simulink® con el solver ode4 (Runge-Kutta) y un periodo de muestreo de 0.001 s.

Para cada uno de los péndulos se tiene un control de movimiento distinto con la finalidad de verificar que a partir de la dinámica propia se tiende a la sincronización. Para el caso del péndulo de referencia se tiene un control  $\tau_1$  que es una composición de tres señales, inicialmente se tiene una forma senoidal dada por  $\sin(2t)$ , que una vez transcurridos 6 segundos se cambia a una secuencia de escalones la cual, a los 16 segundos, vuelve a cambiar a una forma senoidal dada por  $0.5\sin(2t)$ , como se muestra en la Fig. 2.

Para el segundo péndulo se utiliza  $\tau_2 = 0.3\sin(t)$ , mientras que para el tercer péndulo se utiliza  $\tau_3 = \sin(3t)$ . En la Fig.3 se muestra la respuesta de los 3 péndulos no sincronizados.

Se puede apreciar que aunque los péndulos 2 y 3 tienen un comportamiento senoidal, la diferencia entre amplitud y frecuencia hacen que el comportamiento propio de cada uno de ellos no se acerque al del péndulo 1, ni que se pueda observar que existe una sincronización de forma natural. Además de que los cambios en la señal del péndulo de seguimiento no son típicos de un sistema convencional que solo maneja una forma.

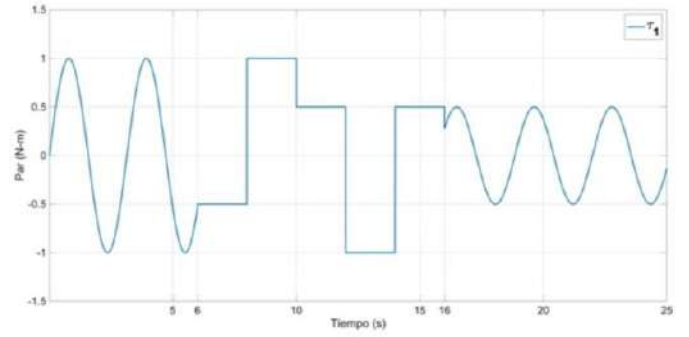


Fig. 2. Señal de control para el péndulo referencia.

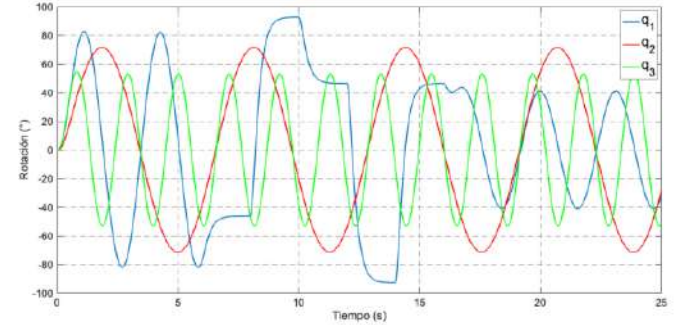


Fig. 3. Respuesta no sincronizada de los péndulos

Para poder apreciar la sincronización de los péndulos, se hace uso del control  $\tau$  de cada uno de ellos, con la finalidad de mostrar que los sistemas se encuentran en un movimiento como el que se aprecia en los primeros segundos de la Fig. 3.

En  $t = 5$  segundos, se activa el control sincronizante para el péndulo 2, mientras que para el péndulo 3 se hace a los 7 segundos. La respuesta angular de los péndulos sincronizados se muestra en la Fig. 4, en donde se puede apreciar que aún y cuando se da un cambio en la señal de referencia en el péndulo 1, los otros dos péndulos la siguen.

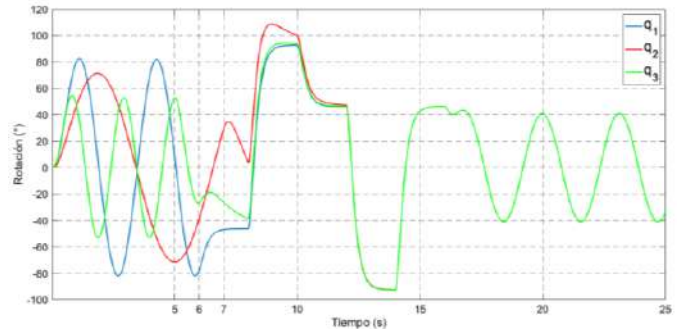


Fig. 4. Señal de salida de los 3 péndulos sincronizada.

Aunque se la convergencia no es inmediata una vez que se activan los controles sincronizantes, los errores entre los péndulos seguidores y el péndulo de referencia tiende a cero, como se aprecia en la Fig. 5.

Se observa que aunque para el péndulo 3 la señal de control sincronizante se activa a los 5 segundos, el sistema tarda aproximadamente otros 5 segundos para que el error  $e_{3,1}$  llegue a cero. Por otra parte, el error  $e_{2,1}$  entre el péndulo 2 y el péndulo 1, tiene una convergencia a cero también en aproximadamente 5

segundos, esto debido a que se utilizaron las mismas ganancias  $K_2$  y  $K_1$  para ambos sistemas. Sin embargo, también es importante señalar, que una vez que se ha logrado la sincronización, esta no se pierde aún y cuando se dan cambios en la forma de la señal de referencia.

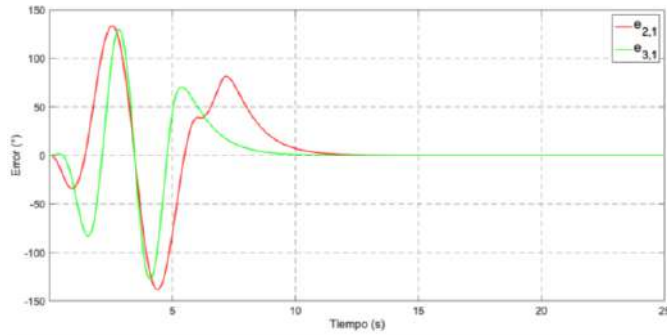


Fig. 5. Error de sincronización.

Las señales de control sincronizante se muestran en la Fig. 6, donde se puede observar que inicialmente parten de cero al no estar actuando, para posteriormente compensar las diferencias entre las dinámicas de los péndulos 2 y 3, tanto de amplitud como de frecuencia, para poder realizar el seguimiento de la señal del péndulo de referencia 1.

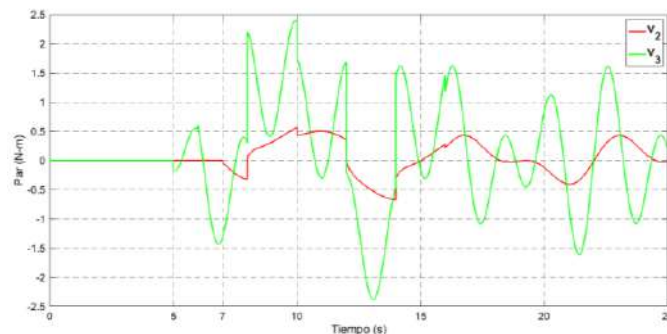


Fig. 6. Señales de controles para sincronización.

Se pueden observar los cambios necesarios tanto en forma como en amplitud que deben de realizar los controles sincronizantes para que los péndulos 2 y 3 puedan seguir al péndulo 1, además de que el efecto de la diferencia paramétrica, así como la amplitud y frecuencia de los controles originales entre los sistemas resulta más evidente en estas señales.

#### IV. CONCLUSIONES

La aplicación de la técnica de sincronización permite demostrar que es posible llevar a cabo un seguimiento adecuado a la señal del sistema guía, sin importar los cambios que se den en dicho sistema, ya sea de amplitud, frecuencia o forma. Además que se pueden tener sistemas que estructuralmente son iguales pero paraméricamente cuentan con diferencias y aún así se logra la sincronización.

Se observa que los resultados de amplitud de las señales de control sincronizantes compensan adecuadamente los cambios dinámicos de los sistemas, por lo que podría estudiarse la comparación entre la potencia de trabajo de un actuador para verificar que no es saturado por los controles.

Sin embargo es importante aclarar que este comportamiento se logra debido a que se conocen todos los parámetros de los sistemas sincronizados, por lo que resulta importante visualizar lo que ocurriría cuando se tienen diferencias en las magnitudes de los parámetros, las cuales podrían representar dinámicas no deseadas.

En un trabajo a futuro se podría estudiar el comportamiento ante perturbaciones acotadas desconocidas junto con identificadores paraméricos dentro de la dinámica de cada sistema, además del uso de no linealidades.

#### REFERENCIAS

- [1] S. Shahal, A. Wurzberg, I. Sibony, H. Duadi, E. Shniderman, D. Weymouth, N. Davidson and M. Fridman, "Synchronization of complex human networks". *Nature Communications*, vol 11, 3854, Aug. 2020. <https://doi.org/10.1038/s41467-020-17540-7>.
- [2] A. Sajadi, R. W. Kenyon & B.M. Hodge, "Synchronization in electric power networks with inherent heterogeneity up to 100% inverter-based renewable generation". *Nature Communications*, vol. 13, 2490 (2022). <https://doi.org/10.1038/s41467-022-30164-3>.
- [3] D. Wang, S. S. Ge, X. Liang and D. Li, "Time-Synchronized Formation Control of Unmanned Surface Vehicles," in *IEEE Transactions on Intelligent Vehicles*, doi: 10.1109/TIV.2024.3371431.
- [4] X. Liu, C. Li, S. S. Ge and D. Li, "Time-Synchronized Control of Chaotic Systems in Secure Communication," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 9, pp. 3748-3761, Sept. 2022, doi: 10.1109/TCSI.2022.3175713.
- [5] H. Nijmeijer and A. Rodríguez-Angeles, *Synchronization of Mechanical Systems*. World Scientific. 2003.
- [6] D. Li, K. P. Tee, L. Xie and H. Yu, "Time-Synchronized Control for Disturbed Systems", in *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 8703-8715, Sept. 2022, doi: 10.1109/TCYB.2021.3054589.
- [7] C. Aldrete-Maldonado, P. J. Prieto-Entenza, C. Plata-Ante, L. Trujillo-Reyes, O. García-Alarcón, y R. Ramírez-Villalobos, «Sincronización de sistemas caóticos de múltiple enrollamiento mediante observador de estado extendido», *ICBI*, vol. 11, n.º Especial2, pp. 110-116, sep. 2023. <https://doi.org/10.29057/icbi.v11iEspecial2.10799>.
- [8] J. A. Rodríguez Liñan, y J. De León Morales. "Sincronización generalizada en orden reducido para sistemas caóticos". *Ingenierías*, vol. 12, no. 45, pp. 6-16, Dic. 2009.
- [9] J. C. Nuñez-Perez, M. A. Estudillo-Valdez, y A. Calvillo-Téllez, "Sincronización MIMO Hamiltoniana usando Ampliación de Muestreo", *ICBI*, vol. 11, n.º Especial2, pp. 22-31, sep. 2023. <https://doi.org/10.29057/icbi.v11iEspecial2.10850>.
- [10] P. R. Carvalho and M. A. Salvi, "Synchronization and chimera state in a mechanical system", in *Nonlinear Dynamics*, vol. 102, pp. 907-925, jan 2020. <https://doi.org/10.1007/s11071-019-05441-4>.
- [11] P.R. Carvalho and M.A Savi, "Synchronization and chimera state in a mechanical system". *Nonlinear Dyn*, vol. 102, pp. 907-925, Jan. 2020. <https://doi.org/10.1007/s11071-019-05441-4>.
- [12] K. Czolczynski, P. Perlikowski, A. Stefanski and T. Kapitaniak, "Synchronization of pendula rotating in different directions". *Commun Nonlinear Sci Numer Simulat*, vol. 17, pp. 3658-3672, Jan. 2012. doi:10.1016/j.cnsns.2012.01.014.
- [13] Z. Li, X. Zhang, W. Chen and B. Wen, "Synchronization characteristics of two vibrator-driven pendulums". *Alexandria Engineering Journal*, vol. 64, pp. 907-921, Nov. 2023.
- [14] D. Rosas, J. Alvarez & E. Alvarez, "Robust synchronization of arrays of uncertain nonlinear second-order dynamical systems". *Nonlinear Dynamics*, vol. 67, pp. 2735-2746, Sept. 2011.

# CAPÍTULO 2



## ROBÓTICA MÓVIL

# IMPLEMENTATION OF A HARDWARE SYSTEM FOR THE REMOTE STATION OF AN ROV/UUV

Marisol Vázquez Tzompantzi  
Secretaría de Investigación y  
Posgrado

Instituto Politécnico Nacional  
México City, México  
mvazquez@ipn.mx

<https://orcid.org/0000-0003-3892-7293>

Bruno Yael Silva Morales  
Unidad Profesional Interdisciplinaria en  
Ingeniería y Tecnologías Avanzadas  
Instituto Politécnico Nacional  
Mexico City, Mexico

bsilvam1200@alumno.ipn.mx  
<https://orcid.org/0009-0007-3570-4904>

Suresh Kumar Gadi  
Centro de Investigaciones y Estudios  
Avanzados del Instituto Politécnico  
Nacional  
Mexico City, Mexico  
research@skgadi.com  
<https://orcid.org/>

Carlos Horus Acosta Espinoza  
Unidad Profesional Interdisciplinaria en  
Ingeniería y Tecnologías Avanzadas  
Instituto Politécnico Nacional  
Mexico City, Mexico  
<https://orcid.org/0009-0007-1767-1950>

Marco Sekehen Gonzalez Tepoz  
Unidad Profesional Interdisciplinaria en  
Ingeniería y Tecnologías Avanzadas  
Instituto Politécnico Nacional  
Mexico City, Mexico  
<https://orcid.org/0009-0006-2689-1851>

Andy Alan Castro Valdez  
Unidad Profesional Interdisciplinaria en  
Ingeniería y Tecnologías Avanzadas  
Instituto Politécnico Nacional  
México City, Mexico  
<https://orcid.org/0009-0005-6429-3222>

**Abstract**— This project focused on developing a remote-control station for an unmanned underwater vehicle (UUV). A Jetson Nano 2GB and a standard keyboard were utilized to implement an algorithm that translates keyboard inputs into control signals, enabling the operation of four Blue Robotics T200 thrusters and four servomotors. The main objective of the project was to design and build an efficient and precise remote-control system capable of maneuvering the UUV in multiple dimensions. The keyboard was used to control movement along the X and Y axes of the robot, while specific keys managed movement along the Z axis and YAW rotation.

The system architecture relies on a TCP/IP socket for communication between the keyboard and the Jetson Nano. Python scripts developed in this project handle reading data from the keyboard, processing these signals, and converting them into discrete commands sent to the UUV. The system also includes visual feedback via OpenCV to display the sensor data and monitor the status of the control signals. (*Abstract*)

**Keywords**—ROV, UUV, Jetson Nano, PMOD JSTK2, TCP/IP, Python, REMOTE-CONTROL.

## I. INTRODUCTION (HEADING 1)

The remote control of remotely operated underwater vehicles (ROVs) presents significant challenges due to the need for precision, reliability, and low latency in control signals, as noted by Griffiths [1]. This is crucial for managing multiple degrees of freedom, such as movement along the X, Y, and Z axes, along with yaw, pitch, and roll rotations, under variable and often adverse underwater conditions, as described by Hawkins [2].

The underwater exploration of natural water bodies such as lakes, rivers, seas, cenotes, and underwater caves, as well as artificial water bodies including dams, reservoirs, and distribution tanks, demands advanced technological solutions. The inspection of submerged structures often requires specialized resources and poses risks to human divers due to factors such as hydrostatic pressure, lack of oxygen, extreme temperatures, and encounters with hazardous substances and wildlife. In this context, Unmanned Underwater Vehicles

(UUVs) offer a safe and efficient alternative to carry out these challenging tasks as outlined by [3].

This article focuses on the design and implementation of an efficient and precise remote control system for the UUV using a Jetson Nano 2GB and a standard keyboard, supported by digital design fundamentals as described by Brown [4] and Chávez [5] for submarine communications. The goal is to translate keyboard inputs into control commands and process them in real-time to ensure low-latency transmission to the UUV 'Don Francisco,' as depicted in 'Fig 1.' The objective is to ensure reliable communication and appropriate response of the UUV to received commands, thereby optimizing its navigation and manipulation capabilities in challenging underwater environments.



Fig. 1: UUV Don Francisco.

## II. OBJETIVES

### A. Design and implement a signal discretization algorithm on the Jetson Nano 4GB

An algorithm was designed to convert the keyboard inputs into discrete data usable by the Jetson Nano. This algorithm interprets key presses corresponding to movements along the

X, Y, and Z axes, as well as yaw, pitch, and roll rotations, converting them into precise digital values.

**B. Integrating the Jetson Nano 2GB modules with a Keyboard for reading control signals**

A physical and logical connection was established between the Jetson Nano 2GB and the keyboard to ensure accurate reading of control signals. This integration enabled continuous and precise capture of input data.

**C. Develop a Reliable and Efficient TCP/IP Communication Interface.**

Develop a reliable and efficient TCP/IP communication interface to facilitate seamless data transmission between the Jetson Nano and the control system. This interface ensures low-latency communication, allowing for real-time processing of control signals and feedback from the UUV.

**D. Implement movement and rotation controls for the UUV.**

The discrete data from the keyboard inputs were translated into control commands that managed the movement along the X, Y, Z axes, and the YAW rotation of the UUV. These controls enabled precise and smooth maneuverability of the vehicle.

**E. Provide a visual interface for monitoring the system status.**

A visual interface shown in 'Fig. 2' was created to allow the operator to monitor the system status in real time. This interface displays relevant information such as images acquired by the robot's camera, orientation, and internal temperature.



Fig. 2: HMI's Visual Interface

**III. PROBLEM DEFINITION**

The remote control of Unmanned Underwater Vehicles (UUVs) poses significant challenges due to the need for precision, reliability, and low latency in control signals. This is crucial for managing multiple degrees of freedom, such as movement along the X, Y, and Z axes, along with yaw rotation, under variable and often adverse underwater conditions.

This article focuses on the design and implementation of an efficient and precise remote control system for the UUV using a Jetson Nano 2GB and a standard keyboard. The goal is to translate keyboard inputs into discrete commands and process them in real-time to ensure low-latency transmission to the UUV. The aim is to ensure reliable communication and appropriate response of the UUV to received commands, thereby optimizing its navigation and manipulation capabilities in challenging underwater environments.

Specifically, the challenge is to structure a Human-Machine Interface (HMI) to control a robot, implementing the mentioned components. To better understand the problem, the physical architecture of the UUV has been diagrammed, and the physical architecture of the HMI is proposed.

To analyze the issue, the information from authors Cross [6] and Hurst [7] has been used as a basis.

**A. UUV Physical Architecture.**

"Fig. 3" shows how the UUV and its key components are physically composed, as well as the interactions between them.

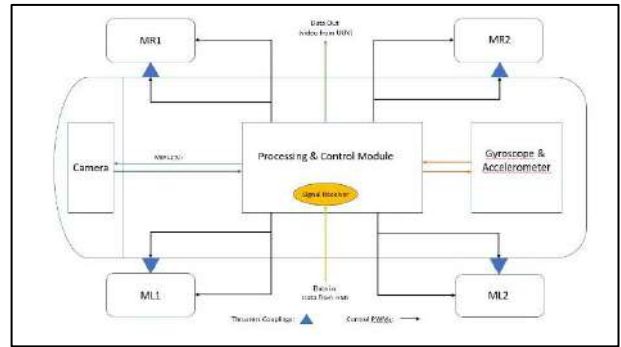


Fig. 3: Physical Architecture from UUV.

**1) Processing & Control Module:**

a) Responsible for processing data and controlling the various components of the UUV.

b) Receives input data from the HMI.

**2) Gyroscope & Accelerometer:**

a) It measures the orientation and acceleration of the UUV, the data is sent to the Processing and Control Module for analysis and use in vehicle control.

**3) Camera:**

a) Captures video that is sent to the Processing and Control Module via the USB interface.

b) The processed video is sent out of the UUV as data output.

**4) Thrusters:**

a) ML1 and ML2 (Motor Left 1 and Motor Left 2): Engines on the left side of the UUV.

b) MR1 y MR2 (Motor Right 1 y Motor Right 2): Engines on the right side of the UUV.

c) They are mounted on the UUV's hull and receive PWM (Pulse Width Modulation) control signals from the Control Module to adjust their speed and direction.

**B. HMI's Physical architecture.**

"Fig. 4" shows the proposed physical architecture for the Human Machine Interface (HMI) of the UUV control system. Each component and its interaction are described below.

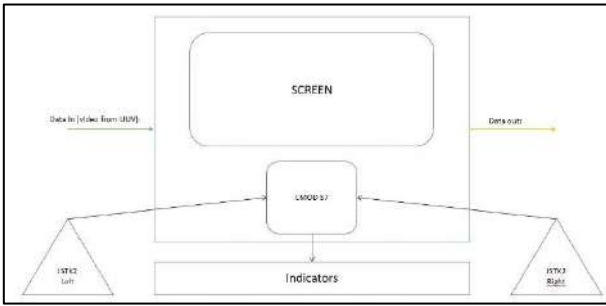


Fig. 4: Physical Architecture from HMI.

1) SCREEN (Monitor):

a) Displays the video received from the UUV and relevant information for the operator.

2) CMOD S7:

- a) Acts as an electronic control unit in the HMI.
- b) Receives signals from keyboard inputs.
- c) Processing the control signals and send this to UUV.

3) Keys "A, S, W, D" as JSTK2 Left and "up, down, left, right" as JSTK2 Right (Joysticks Izquierdo y Derecho):

4) Indicators (LEDs):

They show information about the signals that are being sent to the UUV from the HMI.

C. Hardware design Implementation

"Fig. 5" shows a VHDL code snippet exemplifying one of the hardware modules in the HMI design that sends instructions to the robot.

```

cody:
while True:
    # Recibir el tamaño de los datos
    data_size = b''
    while len(data_size) < 4:
        data_size += conn.recv(4 - len(data_size))
    size = struct.unpack('I', data_size)[0]

    # Recibir los datos
    data = b''
    while len(data) < size:
        data += conn.recv(size - len(data))

    # Deserializar los datos
    received_data = pickle.loads(data)

    # Manejar señales del teclado
    ctrl_pressed = received_data.get('ctrl_pressed', False)
    for key in ['w', 's', 'a', 'd', 'up', 'down']:
        if received_data.get(f'{key}_pressed', False):
            handle_pwm(key, ctrl_pressed)
    
```

Fig. 5: Python Control Code

1) Entity Statement:

a) The UUV\_Controller module is defined with its input and output interfaces, allowing for the processing of control signals from the keyboard and the transmission of commands to the UUV.

b) Inputs: keyboard signals for movement along the X, Y, and Z axes, as well as YAW rotation (corresponding to joystick inputs in the previous implementation). These signals are processed to generate control commands for the UUV).

c) Output: control\_signals (control signals for UUV).

2) Functional Architecture:

a) A loop that continuously runs, checking for input data and processing it in real-time, is used. This loop simulates the behavior of a process that executes on the rising edge of the clock, with conditions to handle resets or disconnections as needed..

b) When the reset is activated, the control signals are reset to zero.

c) On the rising edge of the clock, keyboard inputs are processed and discretized control signals are generated.

3) Signal Processing:

a) The signals are encoded and sent to the UUV's control electron unit.

IV. METODOLOGY

A functional analysis using IDEF0 has been conducted on the UUV to propose the HMI and establish the operation sequence of the robot, building upon a prior functional analysis of the UUV as indicated in the methodology by X. Yan [8].

These diagrams express the function to be analyzed within a block, the inputs, outputs, the mechanisms or tools, and the control or models that will be employed.

A. IDEF0's UUV

The diagram shown in "Fig 6" states that the movement of the robot is the function of interest, the input is the control signal, and the outputs are the information that is sent to the user, the dimming of the lighting, and the new position of the robot.

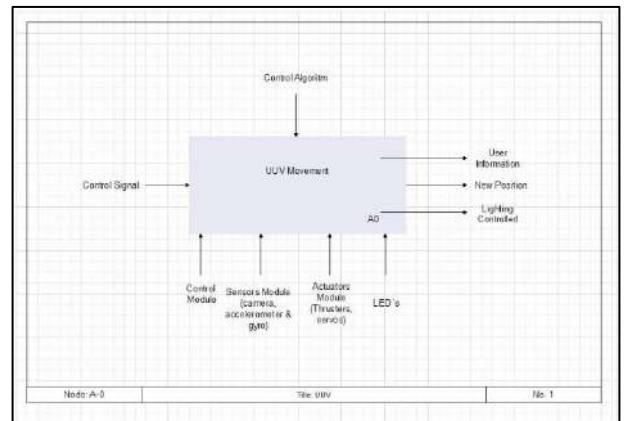


Fig. 6: UUV's IDEF0.

To obtain these outputs, a control algorithm intervenes that, when reading the control signal, makes changes to the robot's actuators. This movement function is broken down into more functions that are shown in the diagram of "Fig. 7", here are shown the different functions that are executed so that the robot moves when receiving the control signal from the HMI, once the signal is decoded the robot moves, this in turn modifies its linear and angular velocity parameters, in

addition to the fact that the intensity of the lighting is modified.

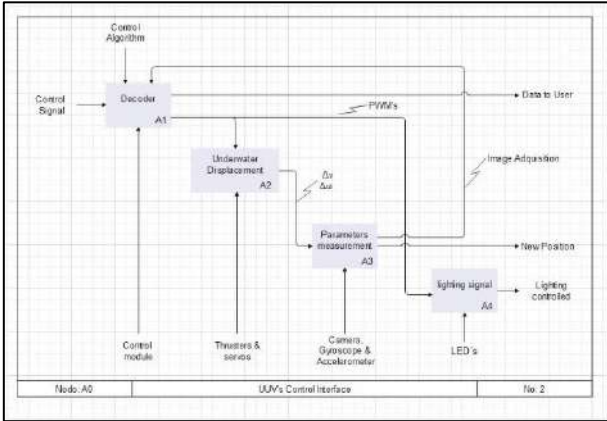


Fig. 7: UUV's control IDEF0.

The data of the change in position and the images captured by the robot are sent to the HMI so that the robot operator can interpret them.

The robot's actuators, sensors, LED lighting system and control modules are involved in this process.

**B. IDEF0's HMI**

Having clear the functional process of the UUV, the control strategy implemented in the HMI has been proposed, which is shown in the IDEF0 of "Fig. 8".

The main node forms the control, whose inputs are made up of the joystick modules and their buttons, which after the process will result in a control signal.

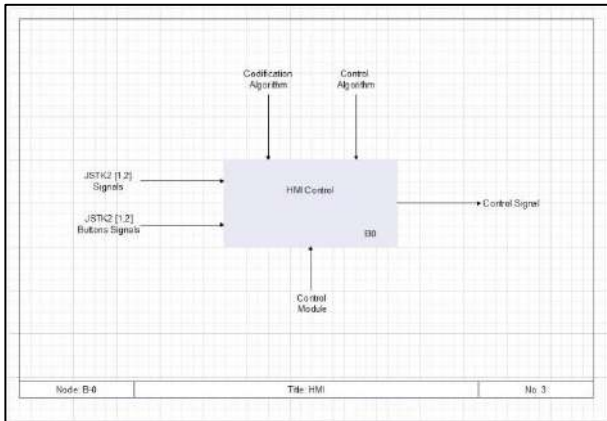


Fig. 8: HMI's IDEF0

This process involves control coding algorithms and the HMI control module itself, which is based on the CMOD S7.

In "Fig. 9", the complete process of operation of the robot is shown, it is shown in which stages the signals of the joysticks and buttons are read, where the algorithms and the control module intervene.

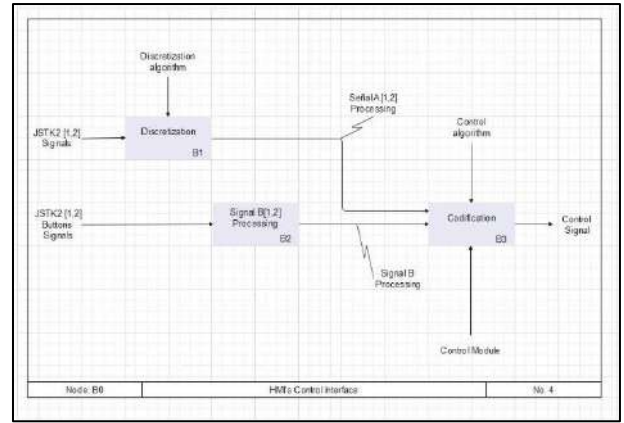


Fig. 9: HMI's Control IDEF0

These diagrams make it easier to understand how different systems work and thus the hardware design process.

**C. Thruster performance**

"Another important factor in the development of UUV control is the performance of the thrusters, which significantly impacts the robot's battery life according to Claus [9]. For this, static tests were conducted with the thrusters and a dedicated hardware model designed in an FPGA specifically for testing the performance of the motors. The code shown in 'Fig. 10' is the synthesized result of the testing algorithm for these tests, while 'Fig. 11' shows the schematic diagram generated by that hardware design..

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Motor_Control is
    Port ( clk_12MHz : in STD_LOGIC;
          pwm_signal : out STD_LOGIC;
          button : in STD_LOGIC);
end Motor_Control;

architecture Behavioral of Motor_Control is
    signal pwm_counter : integer range 0 to 239999; -- Counter for PWM period (12 MHz / 50 Hz = 240,000)
    signal pwm_width : integer range 0 to 1500; -- PWM width in microseconds (1500 us)
    signal thruster_ctrl : STD_LOGIC_VECTOR(1 downto 0) := "00"; -- Default: stop the thruster
    signal esc_initialized : boolean := false; -- Flag to indicate if ESC is initialized

begin
    process(clk_12MHz)
    begin
        if rising_edge(clk_12MHz) then
            -- Initialize ESC by sending a 1500 us PWM signal
            if not esc_initialized then
                if pwm_counter < pwm_width then
                    pwm_signal <= '1';
                else
                    pwm_signal <= '0';
                end if;

                if pwm_counter = pwm_width then
                    esc_initialized <= true;
                end if;
            else
                -- Change main based on button press
                if button = '1' then
                    if thruster_ctrl = "00" then
                        thruster_ctrl <= "01";
                    elsif thruster_ctrl = "01" then
                        thruster_ctrl <= "10";
                    else
                        thruster_ctrl <= "00";
                    end if;
                end if;

                -- Set pwm width based on thruster_ctrl
                case thruster_ctrl is
                    when "00" =>
                        pwm_width <= 1100; -- Forward thrust
                    when "01" =>
                        pwm_width <= 1400; -- Forward thrust
                    when "10" =>
                        pwm_width <= 1600; -- Reverse thrust
                    when "11" =>
                        pwm_width <= 1900; -- Reverse thrust
                    when others =>
                        pwm_width <= 1500; -- Default: stop the thruster
                end case;

                -- generate PWM signal based on pwm_width
                if pwm_counter < pwm_width then
                    pwm_signal <= '1';
                else
                    pwm_signal <= '0';
                end if;
            end if;

            -- Increment the PWM counter
            pwm_counter <= pwm_counter + 1;

            -- Reset the PWM counter at the end of the period
            if pwm_counter = 239999 then
                pwm_counter <= 0;
            end if;
        end process;
    end Behavioral;

```

Fig. 10: Hardware design from Thrusters Test

The diagram represents the FPGA hardware architecture for thruster operation.

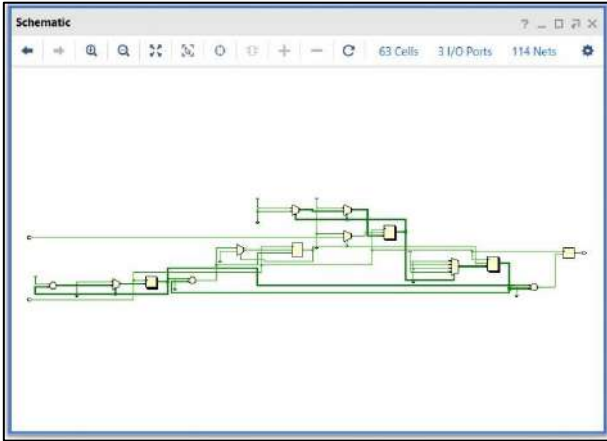


Fig. 11: Thrusters Test Eschematic by RTL.

In "TABLE 1" the data in multiples of 100 of the static tests of the thrusters are shown as well as their physical and electrical parameters when they are controlled.

TABLE I. THRUSTERS PARAMETROS

PWM ( $\mu$ s)	RPM	[A]	[V]	[W]	Thrust [Kg f]	Efficiency
1100	2976	17.03	12	204.4	-2.90	14.2
1200	2457	8.9	12	106.8	-1.95	18.2
1300	1815	3.3	12	39.6	-1.02	25.8
1400	1023	0.5	12	6.0	-0.32	52.9
1500	0	0	12	0.0	0.00	0.0
1600	1009	0.5	12	6.0	0.39	65.8
1700	1796	3.3	12	39.6	1.28	32.3
1800	2456	8.9	12	106.8	2.46	23.1
1900	2995	16.91	12	202.9	3.71	18.3

The board was built by measuring the different electrical parameters by varying the PWM in the Electronic Speed Controller that delivers power to the motor.

A MATLAB license was used to analyze the complete data table with a greater PWM range and with them obtain graphs of power, force and efficiency of the thrusters when implementing the control.

## V. RESULTS & DISCUSSION

### a) Power

The graph in "Fig. 12" shows the relationship between the revolutions per minute (RPM) of the thrusters and the PWM signal from the ESC input.

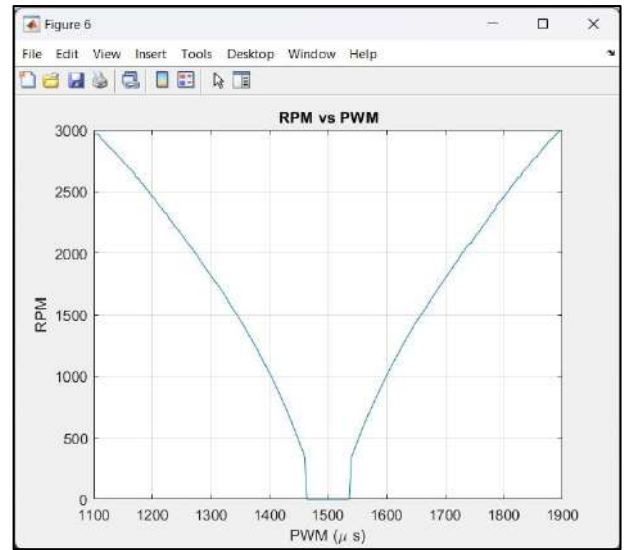


Fig. 12: RPM vs PWM's from Electronic Control Unit

A linear relationship between the PWM value and the RPM can be observed up to certain points, after which the RPM stabilizes.

### b) RPM vs Thrust

The graph in "Fig. 13" shows the power consumed by the thrusters as a function of the thrust generated, measured in kilogram-force (Kg f)

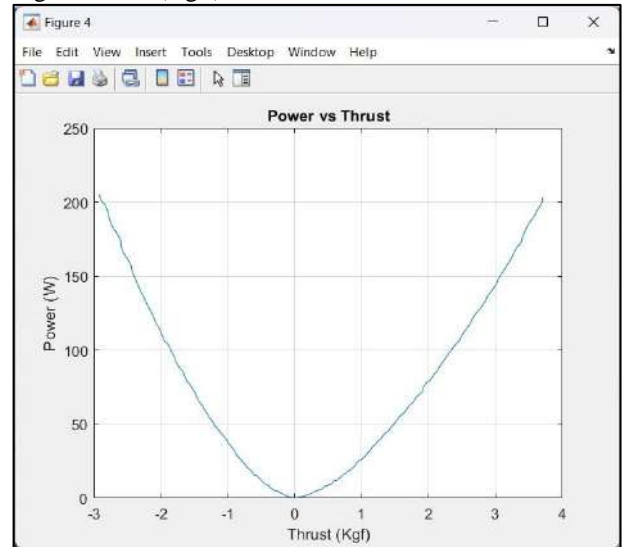


Fig. 13: Power vs Thrust.

The relationship between power and thrust is critical to understanding how energy consumption varies with propellant load.

c) *Efficiency*

The graph in "Fig. 14" shows the efficiency of the thrusters as a function of the PWM signal from input to the ESC. Efficiency is measured in grams per watt (g/W).

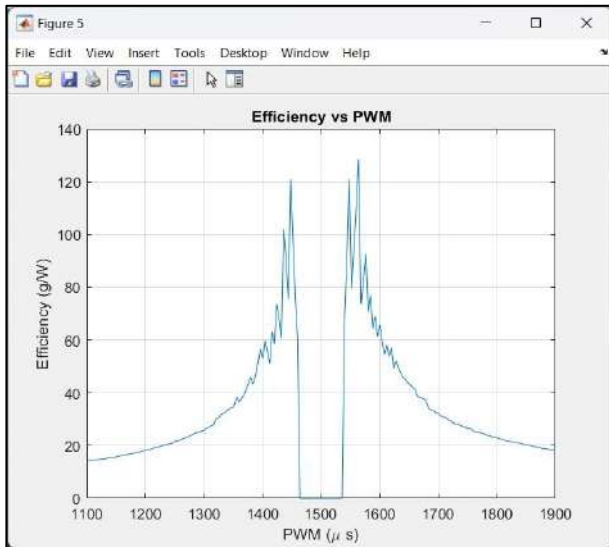


Fig. 14: Efficiency RPM vs PWM's from Electronic Control Unit

CONCLUSIONS

The implementation of the Jetson Nano in the control interface enhances performance and response speed while providing a user-friendly environment for developing advanced Human-Machine Interfaces (HMI).

The linear behavior of RPMs vs PWMs is crucial to determine the operational limits of the thrusters and ensure that the system's capabilities are not exceeded, which could lead to premature wear or mechanical failures.

It is observed that at higher levels of thrust, the power required increases significantly. Planning the UUV maneuvers to operate in the most efficient ranges of this curve

will allow optimal energy management, extending the battery life and, therefore, the vehicle's autonomy.

It is observed that the efficiency is maximum in the range of [1400-1500] microseconds of PWM values, indicating that the system should preferably operate in these ranges to minimize power consumption and maximize thrust. Operating outside of these ranges can result in inefficient energy use, reducing the range of the UUV.

ACKNOWLEDGMENT

Thanks to the IPN's Secretaría de Investigación y Posgrado for the funding and administrative support during the development of this project.

REFERENCIAS

- [1] G. Griffiths, TECHNOLOGY AND APPLICATIONS OF AUTONOMOUS UNDERWATER VEHICLES, London: Taylor & Francis Group, 2003.
- [2] J. Hawkins and A. Allcock, Oceanography and Marine Biology, CRC Press, 2019.
- [3] G. Griffiths and Stevenson, "Open ocean operational experience with the Autosub-1 autonomous underwater vehicle," in *Proceedings 11th Unmanned Untethered Submersible Technology Symposium*, Durham, New Hampshire, 1999, pp. 1-12.
- [4] S. Brown and Z. V, Fundamentals of digital logic with VHDL Design, Ciudad de México: McGraw-Hill, 2000.
- [5] J. Chávez, Sistema de comunicaciones para submarinos autonomos empleando ultrasonido, Ciudad de México: M. S. Tesis, CINVESTAV, 2019.
- [6] N. Cross, Engineering Design Methods Strategies for Product Design, 3ra ed, Wiley. Reino Unido: John Wiley & Sons, LTD., 2000.
- [7] K. Hurst, Engineering Design Principles, Elsevier Science & Technology Books, 1999.
- [8] X. Yan and R. Zante, A Mechatronics Design Process and Its Applications. Case Studies in Mechatronics - Applications and Educations, London, Dordrecht: Springer, 2010.
- [9] B. Claus and R. Bachmayer, "Energy optimal depth control for long range underwater vehicles with applications to a hybrid underwater glider," in *Autonomous Robots*, 2016, p. 1307.

# Control difuso evolutivo para un sistema quadrotor

Emmanuel Adrian Espinoza Gómez  
*División de Estudios de Posgrado*  
*Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca  
emmanueladrianespinoza@gmail.com

Arturo Tellez Velazquez  
*Instituto de computación*  
*Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca  
vitriion@hotmail.com

Esther Lugo González  
*Instituto de Electrónica y Mecatrónica*  
*Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca  
elugog@mixteco.utm.mx

**Abstract**—Se presenta un esquema de control difuso optimizado con un algoritmo evolutivo diferencial implementado en un quadrotor de 6 grados de libertad. El objetivo es mejorar el desempeño de un sistema quadrotor para el seguimiento de trayectorias tridimensionales. En este trabajo se muestra que un estimador difuso empleado para sintonizar las ganancias de un controlador PID, potencia el rendimiento del quadrotor, particularmente cuando lleva una carga de 300 gramos en una trayectoria vertical. Se propone un esquema de optimización evolutivo diferencial para un controlador compuesto por 6 sistemas de inferencia difusa (FIS). El diseño del optimizador permite minimizar simultáneamente 4 funciones objetivo. Los resultados obtenidos muestran que el esquema propuesto atenúa un 50% la oscilación de una carga útil al retroalimentar la posición angular en el algoritmo de optimización, en comparación con ignorar la carga tanto en el diseño del optimizador como del controlador.

**Index Terms**—posición de quadrotor, control difuso evolutivo, evolución diferencial, optimización evolutiva diferencial.

## I. INTRODUCTION

De acuerdo con la relación entre el controlador y el modelo controlado, en [1] dividen los métodos de control de estabilidad de cuadricópteros en tres categorías: control clásico, control óptimo y control inteligente. La estabilidad de un quadrotor contiene varios aspectos: comportamientos estables como altitud, seguimiento de trayectoria, robustez ante parámetros de modelos inciertos cuando no se dispone de un conocimiento completo o preciso del modelo, la tolerancia al fallo del rotor, y el rechazo de perturbaciones, como la capacidad de rechazo de ráfagas de viento. Estos cuatro aspectos constituyen la estabilidad de un quadrotor y se requiere que el controlador diseñado tenga una buena respuesta transitoria y robustez en condiciones adversas o perturbaciones.

Un quadrotor de cuerpo rígido y simétrico tiene tres grados de libertad rotacionales y tres grados de libertad de desplazamiento lineal. El seguimiento de posición del quadrotor en el espacio tridimensional consiste en manipular automáticamente la posición y orientación del quadrotor para alcanzar la trayectoria deseada. La dinámica del quadrotor es no lineal, se puede representar en un espacio de estados de 6 grados de libertad, 12 variables de estado y 4 entradas de control. Al agregar una carga útil se modifica la inercia

del quadrotor en función de la masa y el ángulo del alambre que sostiene la carga, haciendo más compleja la tarea de seguimiento, esto implica el uso de un sistema de control robusto.

En [2], se muestra que un controlador difuso tiene mejor rendimiento que un control basado en linealización por retroalimentación, para controlar la posición de un quadrotor con carga útil sujeta mediante un manipulador de dos eslabones a lo largo de una trayectoria. Su controlador difuso utiliza ganancias de escala para el error y la salida de control, de modo que se tiene el mismo universo de discurso para entradas y salidas, empleando 11 funciones de pertenencia simétricas triangulares centrados en cero, utilizando el modelo de inferencia difusa Mamdani con la composición max-min. En [3] se comparan dos controladores difusos, uno de tipo 1 y otro de tipo 2, en el que se concluye que el controlador tipo 1 o Mamdani es eficiente para controlar la altitud del quadrotor, cuando una carga útil se sujeta con un eslabón del manipulador anclado a la aeronave.

Existen sistemas de control híbridos que involucran sistemas difusos y controles clásicos, tal como sucede en [4], donde se demuestra que el uso de un estimador difuso para sintonizar las ganancias de un controlador PID (Proporcional-Integral-Derivativo) mejora el rendimiento del quadrotor, con una carga de 300 gramos a lo largo de una trayectoria vertical. En [5], se emplea la metodología de control basado en pasividad, interconexión y asignación de amortiguamiento sistemas subactuados (IDA-PBC) para controlar la posición de un quadrotor que utiliza una cámara externa para retroalimentar la posición de su carga útil en el espacio. El quadrotor recorre la distancia en 11 segundos y la carga alcanza la posición deseada en un tiempo de tres segundos. El controlador presenta oscilaciones mínimas de la posición de la carga durante el tiempo total de regulación.

En [6], se desarrolla un control basado en las propiedades de pasividad y energía; los resultados demuestran un desempeño adecuado para regular la posición de la carga alcanzando la referencia en 5 segundos con un ángulo de oscilación máximo de 20 grados. El esquema DIC (Dynamic Inversion Control), presentado en [7], es utilizado para reducir el tiempo de establecimiento que se consigue con un controlador PID,

utilizando el error de posición de la carga suspendida.

Además de los esquemas lineales y no lineales, es posible utilizar algoritmos inteligentes para optimizar el rendimiento de los controladores. Por ejemplo, en [8] se expone el uso de un control por modos deslizantes junto a una red neuronal de base radial que aproxima los valores de las ganancias del controlador. Se demuestra que el uso de algoritmo de optimización mejora significativamente el rendimiento del controlador.

En [9], se realiza la comparación de un controlador no lineal frente al esquema PID, para posicionar la carga a lo largo de una trayectoria circular a una distancia constante del suelo. Los resultados favorecen al control no lineal consiguiendo anular los errores de posición (X,Y), una vez alcanzada la altura (Z) especificada, a los 4 segundos. En [10] Klausen y Fossen presentan los resultados a nivel de simulación, de un controlador no lineal, derivado de la técnica Backstepping, para controlar la posición de la carga durante una ruta con forma espiral de 50 cm de diámetro. En [11] se propone un esquema de control adaptativo para la posición de la carga suspendida del quadrotor, utilizando una conexión rígida, se utiliza un modelo dinámico de seis grados de libertad asumiendo los efectos dinámicos de la carga como perturbación externa. La implementación del controlador robusto genera un error de seguimiento máximo de 5 cm sobre las dimensiones (X, Y y Z) durante el transporte de la carga por una trayectoria en forma de espiral.

El problema de controlar la posición de una carga útil suspendida de un quadrotor tiene dos vertientes, se controla la posición del centro geométrico de la aeronave o se controla la posición del centro geométrico de la carga útil. Esta investigación se centra en controlar la posición del quadrotor con carga útil bajo el siguiente enfoque, la carga es asumida como una perturbación por lo que su dinámica no se considera en el diseño del controlador, en su lugar, la posición de la carga es retroalimentada a la función de optimización evolutiva para ser atenuada. Se desarrolla y valida una propuesta a la tarea de seguimiento de trayectorias espaciales para la posición del quadrotor, la cual se divide en dos: primero se realiza el control de cuatro variables de estado (3 angulares y 1 traslacional), después se ejecuta la generación automática de las referencias para dos variables angulares lo que supone el control de las dos dimensiones subactuadas.

#### A. Modelo Matemático

La dinámica del sistema quadrotor se modela para comprobar el rendimiento del controlador propuesto. Utilizando el método de Euler, se obtiene la dinámica de las posiciones angulares y lineales del quadrotor, asumiendo la carga útil como una perturbación. Asimismo, se modelan las dos dinámicas angulares de la posición de la carga, en función de la posición del quadrotor. Después, estas dinámicas se linealizan alrededor

de un punto de equilibrio dando como resultado las funciones de transferencia de (1) hasta (8):

$$G_1 = \frac{56.95s + 4391}{s^3 + 105s^2 + 870s + 4430} \quad (1)$$

$$G_2 = \frac{65s + 4560}{s^3 + 109s^2 + 1023s + 2935} \quad (2)$$

$$G_3 = \frac{105}{s^2 + 413s} \quad (3)$$

$$G_4 = \frac{1.63}{s^2 + 5s} \quad (4)$$

$$G_5 = \frac{-190s + 567}{57.95s^2 + 4400s} \quad (5)$$

$$G_6 = \frac{-276.4s + 743.5}{61s^2 + 4463s} \quad (6)$$

$$G_{\theta,load} = \frac{\theta}{u_1} = \frac{56.95s + 4391}{s^3 + 105s^2 + 870s + 4430} \quad (7)$$

$$G_{\phi,load} = \frac{\phi}{u_2} = \frac{65s + 4560}{s^3 + 109s^2 + 1023s + 2935} \quad (8)$$

Donde  $G_1, G_2, G_3$  son las funciones de transferencia correspondientes a las variables de estado X, Y, Z. Y las funciones  $G_4, G_5, G_6$  corresponden a las variables de estado Alabeo, Cabeceo, y Guiñada. La simulación del controlador optimizado propuesto se realiza en *Matlab*, asumiendo los parámetros físicos (ver Tabla 1). Estos datos físicos del sistema quadrotor son obtenidos experimentalmente en [12].

TABLE I  
PARÁMETROS FÍSICOS

Parámetro	valor	unidad
$m_{load}$	0.2	Kg
$m_{quad}$	1.0	Kg
M	1.2	Kg
L	0.2	m
$I_x$	$2.35e^{-3}$	$kg\ m^2$
$I_y$	$2.353e^{-3}$	$kg\ m^2$
$I_z$	$5.262e^{-2}$	$Kg\ m^2$
$K_t$	$3.13e^{-5}$	$Ns^2$
$m_{load}$	0.2	Kg
$l_{load}$	0.10	m
$I_l$	0.001	$Kg\ m^2$
$b_x$	0.7	N/(m/s)
$b_y$	0.8	N/(m/s)

## II. DISEÑO DE SISTEMA DIFUSO EVOLUTIVO

### A. Control difuso

Al linealizar la dinámica del sistema quadrotor es posible controlar independientemente cada grado de libertad. En esta investigación se propone un sistema de inferencia difuso para cada una de las variables de estado establecidas. El sistema de control propuesto consta de seis controladores difusos, uno para cada grado de libertad del quadrotor. Estos sistemas de inferencia difusa se disponen para controlar tanto la orientación como la traslación de la aeronave según el diagrama

de la Figura 1, en la que se agrupan el control de elevación (izquierda), control de traslación (derecha), lo que facilita la comprensión de lo que sucede internamente en el sistema.

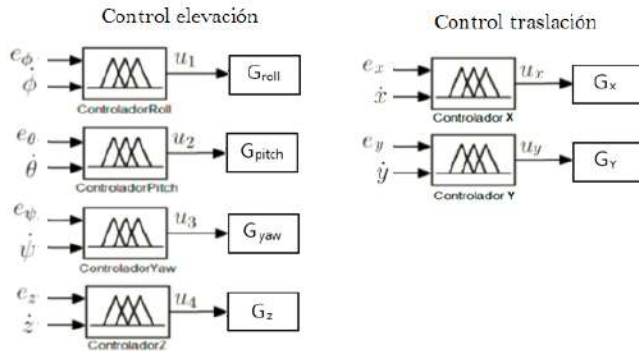


Fig. 1. Esquema de control difuso para seguimiento del sistema quadrotor

Los seis sistemas difusos comparten la misma estructura en su arquitectura. Cada sistema tiene como variables de entrada el error de posición y la derivada de la posición. Estas variables se fusifican y se procesan para inferir el valor de la variable de salida que es una señal de control (en el caso de los controladores de elevación) y una señal de referencia (en el caso de los dos controladores de traslación). La Tabla 2 muestra el tipo de funciones con las que se define el grado de pertenencia de cada variable. Tomando como base el trabajo de Tellez-Velasquez et al. [13], el error de posición se define con una función Z, tres funciones Gaussianas y una función S; la derivada de posición se define con una función Z una función Gaussiana y una función S; por último, las variables de salida se definen con una función Z, cinco funciones Gaussianas y una función S. En la segunda columna se muestran los parámetros requeridos por cada variable dando un total de 30 parámetros para definir cada uno de los sistema de inferencia difusa.

TABLE II  
FUNCIONES DE PERTENENCIA

Variable difusa (tipo)	Funciones	Número parámetros
error de posición (entrada)	Z, G, G, G, S	10
derivada posición (entrada)	Z, G, S	6
señal control (salida)	Z,G,G,G,G,S	14

### B. Optimización evolutiva diferencial

El algoritmo evolutivo diferencial es un optimizador estocástico cuyo funcionamiento consiste en la evolución a partir de una población inicial de potenciales soluciones. En este caso, el individuo es el vector de parámetros que definen a un sistema de control. Como se explicó en la sección anterior, se requieren 15 parámetros para definir el

controlador difuso para un grado de libertad, de manera que se utilizan 90 parámetros para realizar la tarea de seguimiento de trayectorias con el sistema quadrotor.

De manera similar que en [13], este trabajo propone un esquema evolutivo diferencial para minimizar 4 funciones objetivo descritas en (9). Se trata de una función objetivo dependiente del estado de las variables Alabeo y X, una dependiente del estado de las variables Cabeceo e Y, una dependiente del estado de la variable Guiñada y una dependiente de la altura Z. La ecuación (9) es una representación de las 4 funciones objetivo ( $f_{\psi y}$ ,  $f_{\theta x}$ ,  $f_{\phi}$  y  $f_z$ ), que son independientes entre sí, esto quiere decir que el optimizador busca minimizar 4 objetivos al mismo tiempo:

$$f_{obj} = [f_{\psi y}(\psi, x), f_{\theta y}(\theta, y), f_{\phi}(\phi), f_z(z)] \quad (9)$$

Las funciones objetivo son combinaciones de errores de posición y algunas características cuantificables de la respuesta del sistema durante el seguimiento de trayectorias. De (10) hasta (13) son las 4 funciones objetivo propuestas:

#### Función 1:

$$f_{\psi x}(\psi, x) = \sqrt{\frac{E_{\psi}^2 + E_x^2}{2}} \quad (10)$$

donde:

$$E_{\psi} = (\alpha_{11} e_{\psi_{ss}} + \alpha_{12} e_{\psi_{MP}} + \alpha_{13} e_{\psi_{ie}} + \alpha_{14} e_{\psi_{er}})$$

$$E_y = (\alpha_{15} e_{y_{ss}} + \alpha_{16} e_{y_{MP}} + \alpha_{17} e_{y_{ie}} + \alpha_{18} e_{y_{rizo}})$$

En la Tabla III se describe la nomenclatura utilizada para nombrar los términos de la ecuación, el subíndice de cada variable permite intuir fácilmente a que métrica hace referencia cada término de las siguientes funciones objetivo.

TABLE III  
DESCRIPCIÓN DE VARIABLES

Índices de error Alabeo ( $\psi$ )	Descripción
$e_{\psi_{ss}}$	Error en estado estacionario de $\psi$
$e_{\psi_{MP}}$	Sobre impulso máximo de $\psi$
$e_{\psi_{ie}}$	Error global de $\psi$
$e_{\psi_{rizo}}$	Cantidad de oscilación o rizo en $\psi$
$e_{y_{ss}}$	Error en estado estacionario de Y
$e_{y_{MP}}$	Sobre impulso máximo de Y
$e_{y_{ie}}$	Error global de Y
$e_{y_{rizo}}$	Cantidad de oscilación o rizo en Y

#### Función 2:

$$f_{\theta y}(\theta, y) = \sqrt{\frac{E_{\theta}^2 + E_y^2}{2}} \quad (11)$$

donde:

$$E_{\theta} = (\alpha_{21} e_{\theta_{ss}} + \alpha_{22} e_{\theta_{MP}} + \alpha_{23} e_{\theta_{ie}} + \alpha_{24} e_{\theta_{er}})$$

$$E_x = (\alpha_{25} e_{x_{ss}} + \alpha_{26} e_{x_{MP}} + \alpha_{17} e_{x_{ie}} + \alpha_{18} e_{x_{rizo}})$$

**Función 3:**

$$f_{\phi}(\phi) = \sqrt{\frac{E_{\phi}^2}{2}} \quad (12)$$

donde:

$$E_{\phi} = (\alpha_{21} e\phi_{ss} + \alpha_{22} e\phi_{MP} + \alpha_{23} e\phi_{ie} + \alpha_{24} e\phi_{er})$$

**Función 4:**

$$f_z(z) = \sqrt{\frac{E_z^2}{2}} \quad (13)$$

donde:

$$f_z(z) = (\alpha_{25} ez_{ss} + \alpha_{26} ez_{MP} + \alpha_{17} ez_{ie} + \alpha_{18} ez_{rizo})$$

1) *Hiper-parámetros:* Los resultados presentados en la siguiente sección se obtienen con los valores  $\alpha$  para cada una de las cuatro funciones objetivo,  $[\alpha_{\psi}, \alpha_y] \rightarrow f_{\psi y}(\psi, y)$ ,  $[\alpha_{\theta}, \alpha_x] \rightarrow f_{\theta x}(\theta, x)$ ,  $[\alpha_{\phi}] \rightarrow f_{\phi}(\psi)$  y  $[\alpha_z] \rightarrow f_z(z)$ .

$$\begin{pmatrix} \alpha_{\psi} \\ \alpha_y \\ \alpha_{\theta} \\ \alpha_x \\ \alpha_{\phi} \\ \alpha_z \end{pmatrix} = \begin{pmatrix} \alpha_{11} = 0.5 & \alpha_{12} = 0.2 & \alpha_{13} = 0.05 & \alpha_{14} = 0.25 \\ \alpha_{15} = 0.4 & \alpha_{16} = 0.2 & \alpha_{17} = 0.2 & \alpha_{18} = 0.1 \\ \alpha_{21} = 0.35 & \alpha_{22} = 0.5 & \alpha_{23} = 0.1 & \alpha_{24} = 0.15 \\ \alpha_{25} = 0.35 & \alpha_{26} = 0.25 & \alpha_{27} = 0.45 & \alpha_{28} = 0.05 \\ \alpha_{31} = 0.4 & \alpha_{32} = 0.4 & \alpha_{33} = 0.1 & \alpha_{34} = 0.1 \\ \alpha_{41} = 0.6 & \alpha_{45} = 0.2 & \alpha_{43} = 0.15 & \alpha_{44} = 0.05 \end{pmatrix}$$

Se realiza una búsqueda en malla para definir el valor de la tasa de cruce (CR) y el factor de amplificación (F). Para esta dinámica específica, se logra un rendimiento óptimo con una Tasa de cruce de 1.6 y un Factor de amplificación de 0.8.

**C. Implementación**

El código que implementa la tarea de seguimiento de trayectorias con el algoritmo difuso evolutivo se compone por dos módulos principales: el de evaluación y el de optimización. El módulo de evaluación contiene la información tanto del esquema de control como de la dinámica del sistema; este es accedido por el módulo de optimización cuya función es estimar los parámetros de control con base a las funciones objetivo.

1) *Módulo de evaluación:* El módulo de evaluación se subdivide en funciones dedicadas a la solución de las siguientes tareas específicas:

- Definición de parámetros de simulación dinámica.
- Generación de señales de referencia para las dimensiones X, Y, Z y  $\psi$ .
- Cálculo de la dinámica del sistema quadrotor con carga útil.
- Creación y definición de seis sistemas de inferencia difusa.
- Edición de sistemas de inferencia para establecer los parámetros de control recibidos.
- Creación e inicialización de variables internas.
- Ciclo principal: simulación y respuesta de seguimiento de trayectorias.
- Cálculo de índices de error.
- Cálculo de 4 funciones objetivo.
- Retorno de valores objetivo.

2) *Módulo de optimización:* El módulo de optimización depende de dos parámetros utilizados en el algoritmo de evolución diferencial: factor de amplificación  $F$  y tasa de cruce  $CR$ . Este módulo devuelve al mejor individuo o vector solución. De manera general el funcionamiento de este módulo se resume en tareas que se ejecutan secuencialmente y se enlistan a continuación:

- Definición de variables involucradas en algoritmo evolutivo
- Generación aleatoria de la población inicial
- Validación y ordenamiento de parámetros de cada individuo generado
- Ciclo inicial: Iteración de función de evaluación con todos los individuos de la población inicial
- Ciclo principal: optimización
- Cálculo y visualización de superficie de errores

**III. RESULTADOS**

En esta sección se analizan los resultados obtenidos en la implementación del esquema de control difuso evolutivo para el seguimiento de una trayectoria tridimensional. El experimento de seguimiento se realizó 10 veces, los resultados de esta investigación representan el rendimiento promedio del esquema difuso evolutivo en estos experimentos. Así mismo, se detallan las condiciones bajo las cuales se llevo a cabo la simulación de los controladores y la dinámica del sistema quadrotor y la carga útil.

1) *Configuración experimental:* La simulaciones del sistema dinámico y el controlador difuso evolutivo se ejecutan en Matlab Simulink 2018. El modelo y los valores de los parámetros que definen la dinámica del quadrotor se detallan en la sección 2, a continuación se presenta un resumen de los parámetros y configuraciones necesarios para replicar el experimento.

El controlador difuso, optimizado mediante evolución diferencial, consta de seis sistemas de inferencia difusa (FIS). Como se explicó en la sección 3, los seis FIS comparten la misma arquitectura, cada uno recibe el error de posición y la derivada de la posición para calcular una señal de control. Los parámetros Factor de Amplificación y Tasa de Cruce definidos mediante una búsqueda en malla, así como el resto de parámetros utilizados para definir el algoritmo de optimización evolutivo diferencial se muestran en la tabla IV.

TABLE IV  
PARÁMETROS DE OPTIMIZACIÓN

variable	valor	nombre
popSize	50	tamaño de la población
n_gens	60	número de generaciones
tol	0.0001	tolerancia
CR	0.2	tasa de cruce
F	0.4	factor de amplificación

**A. Análisis de resultados**

El esquema de control difuso evolutivo se valida ante las trayectorias mostradas en la Figura 2. Se busca exigir

movimiento en los ejes X e Y, con el objetivo de conocer el comportamiento de la carga útil durante el traslado. Las líneas punteadas muestran las trayectorias de las referencias para cada grado de libertad.

Las trayectorias impuestas para las variables de estado (X, Y, Z,  $\phi$ ) son creadas por un generador de trayectorias, mientras que las referencias ( $\psi$ ,  $\theta$ ) se generan automáticamente por dos de los sistemas de inferencia. Las posiciones X, Y y Z alcanzan la referencia deseada en 2.9 s, 3.3 s y 1.6 s. respectivamente.

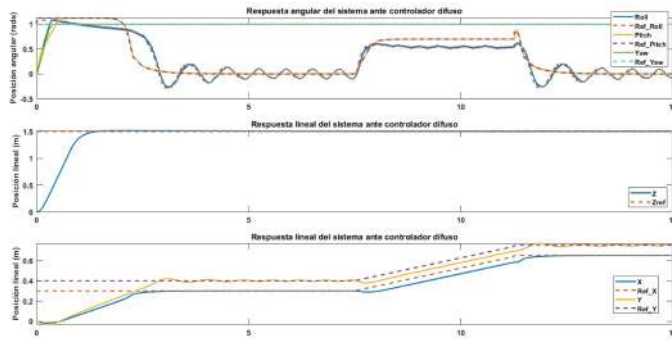


Fig. 2. Respuesta de sistema difuso ante doble escalón

1) *Carga útil*: La dinámica de la carga útil no se modela en el diseño del control difuso pero se asume como una perturbación durante el vuelo del quadrotor. Para comprobar la robustez del algoritmo de optimización se incluyen dos términos que involucran la diferencia entre la posición angular de la carga y el eje vertical en las funciones objetivo que contienen los errores de X e Y respectivamente.

Con la finalidad de evitar oscilaciones no deseadas en el quadrotor debido a la posición de la carga, se realiza una penalización en las funciones objetivo  $f_{\psi_x}$  y  $f_{\theta_y}$  con la que se obtiene los siguientes resultados. La oscilación mas grande respecto al eje X alcanza 0.11 radianes (6 grados) y respecto al eje Y la carga oscila hasta 0.22 radianes, equivalente a 12.6 grados. La diferencia de comportamientos entre un eje y otro se debe a que se asigna diferente coeficiente de fricción a las componentes de movimiento de la carga durante la simulación (ver Figura 3).

Por otro lado, se realiza la optimización sin tomar en cuenta el ángulo de la carga, la posición máxima alcanzada sin penalizar la oscilación es de 0.2 radianes (11.4 grados) en el caso del ángulo medido respecto al eje X. La máxima posición alcanzada respecto al eje Y es de 0.53 radianes (30.5 grados). En la tabla V se comparan las posiciones de la carga útil para los dos casos, cuando se penaliza la carga útil durante la optimización y cuando no se hace. Con esto se demuestra que es posible reducir la oscilación de la carga retroalimentando la posición angular directamente al algoritmo evolutivo diferencial, bajo el enfoque presentado en este documento.

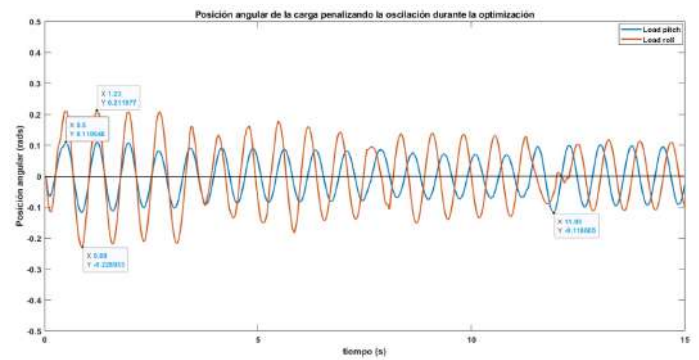


Fig. 3. Respuesta de carga al penalizar la oscilación durante optimización

TABLE V  
OSCILACIONES DE LA CARGA ÚTIL

Posición	Con penalización de la oscilación de la carga	Sin penalización de la oscilación de la carga
Oscilación máxima en X	0.11 radianes	0.20 radianes
Oscilación máxima en Y	0.22 radianes	0.53 radianes

#### IV. CONCLUSIONES

El sistema presentado en este trabajo propone un esquema evolutivo diferencial diseñado para minimizar 4 funciones objetivo, correspondientes a un controlador difuso. El esquema de control difuso, optimizado mediante el algoritmo Evolutivo Diferencial, ofrece una solución eficiente y robusta para el seguimiento trayectorias usando quadrotores con cargas útiles. El optimizador evolutivo consigue prevenir el sobre impulso y eliminar el error en estado estacionario en el plano XY, con una rapidez de  $11 \frac{cm}{s}$  ( $\pm 2$ ). Además, al incorporar la transformada de Fourier de la posición angular como término de penalización en las funciones objetivo, se logra atenuar la oscilación de la carga útil, a pesar de que el diseño de los controladores difusos no consideran la dinámica de la carga.

Los resultados demuestran que el esquema evolutivo diferencial permite una optimización eficiente y robusta, reduciendo significativamente la oscilación de la carga útil en un 52 % en el eje X y un 41 % en el eje Y. Las posiciones X, Y y Z del quadrotor alcanzan las referencias deseadas rápidamente en 2.9 s, 3.3 s y 1.6 s, respectivamente, mostrando que el sistema es tanto estable como rápido en su respuesta. La combinación de técnicas de optimización evolutiva y control difuso proporciona mejoras significativas en estabilidad, precisión y rapidez del sistema, lo que sugiere su viabilidad para aplicaciones en la industria y la investigación.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present

them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

#### REFERENCES

- [1] Pang S. and Tian L. ,”Control Methods for Quadrotor’s Stability”, Journal of Physics: Conference Series 1948, 202, pp.11-15
- [2] Khalifa A. , Fanni M. , Ramadan A., and Abo-Ismael A., ”Adaptive Intelligent Controller Design for a New Quadrotor Manipulation System”, 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 1666-1671, doi: 10.1109/SMC.2013.287.
- [3] Wicaksono, Hendi and C. Buyung. ”T1-Fuzzy vs T2-Fuzzy stabilize Quadrotor Hover with payload position disturbance,” International Journal of Applied Engineering Research 9.22, 2014, pp.17883-17894.
- [4] I. V. Gue, and A. Chua. ”Development of a Fuzzy GS-PID Controlled Quadrotor for Payload Drop Missions,” Journal of Telecommunication, Electronic and Computer Engineering 10 2018, pp. 55-58.
- [5] M. E. Guerrero, D. A. Mercado, R. Lozano and C. D. García, ”Passivity based control for a quadrotor UAV transporting a cable-suspended payload with mini- mum swing,” 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 6718-6723, doi: 10.1109/CDC.2015.7403277.
- [6] Guerrero-Sánchez, María-Eusebia, et al. Robust Control for a Quadrotor Transporting a Payload,” 2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON). IEEE, 2019.
- [7] Das, Hemjyoti. ”Dynamic inversion control of quadrotor with a suspended load,” IFAC-PapersOnLine 51.1, 2018, pp.172-177.
- [8] Luo, Cai, Zhenpeng Du, and Leijian Yu. ”Neural network control design for an unmanned aerial vehicle with a suspended payload,” Electronics 8.9, 2019, pp.931.
- [9] Sadr, Sara, S. Ali A. Moosavian, and Payam Zarafshan. ”Dynamics modeling and control of a quadrotor with swing load,” Int. J. Robot. Res., 2014.
- [10] Klausen, Kristian, Thor I. Fossen, and Tor Arne Johansen. ”Nonlinear control of a multirotor UAV with suspended load,” 2015 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2015.
- [11] Wang, Chen, et al. ”Trajectory tracking control for quadrotor robot subject to payload variation and wind gust disturbance,” Intell Robot Syst 83, 2016, pp.315-333.
- [12] Li, Jun, and Yuntang Li. ”Dynamic analysis and PID control for a quadrotor,” 2011 IEEE International Conference on Mechatronics and Automation. IEEE, 2011.
- [13] A. Tellez Velazquez, H. Molina Lozano, L.A. Villa Vargas, R. Cruz Barbosa, E. Lugo Gonzalez, I.Z. Batirshin, and I.J. Rudas. ”A Feasible Genetic Optimization Strategy for Parametric Interval Type - 2 Fuzzy Logic Systems,” Int. J. Fuzzy Syst. vol.20, pp.318 - 338, 2018.

# Matching Indoor Navigation Systems to Application Requirements: A Comparative Study

German Cuaya-Simbro\*  
*Maestría en Sistemas Computacionales  
Tecnológico Nacional de México  
ITS del Oriente del Estado de Hidalgo  
Apan, Hidalgo  
<http://orcid.org/0000-0001-6303-154X>*

Jorge Armando García Bautista  
*Ingeniería en Sistemas Computacionales  
Tecnológico Nacional de México  
ITS del Occidente del Estado de Hidalgo  
Mixquiahuala, Hidalgo  
<https://orcid.org/0009-0005-4407-8234>*

**Abstract**—This paper presents a comparative analysis of radio frequency (RF) and vision-based indoor navigation systems, aiming to provide practical guidance for system selection. While vision systems generally offer higher accuracy and adaptability, they are computationally demanding and sensitive to environmental conditions. Conversely, RF systems provide wider coverage and lower power consumption but often exhibit lower precision. To bridge this gap, this research proposes a comprehensive evaluation framework considering both quantitative metrics (accuracy, precision, cost) and qualitative factors (environment, application requirements). By comparing different system configurations, the study provides valuable insights into selecting the optimal indoor navigation solution for various robotic applications, such as logistics, healthcare, and manufacturing.

**Index Terms**—Radio Frequency-based, vision-based, cost-benefit, comparative analysis, real application.

Indoor navigation is a term used in describing the procedures involved with determining the location of a mobile robot within a house or any other closed environment, and, without use GPS satellites, it is a real, hard problem. Indoor navigation for mobile robots has gained significant traction in recent years due to its potential to revolutionize various industries, including logistics, healthcare, and manufacturing Huang et al. (2023).

Several methods exist for the indoor positioning and indoor navigation. For example, a popular method involves using radio frequency emitted by Wi-Fi or Bluetooth signals. On the other hand, there are approaches that use computational vision which offer several advantages over popular methods, Otero et al. (2020), Huang et al. (2023), and Fusic et al. (2023). Vision-based indoor navigation systems have high precision, rich environment perception, and adaptability to changes, for applications such as autonomous mobile robots and security and surveillance, however, they can be more computationally demanding and sensitive to lighting conditions and occlusions. On the contrary, radio wave-based systems are generally better suited for applications that require wide coverage and low power consumption, such as warehouse automation and retail environments. Therefore, it is important have or know a basic reference to determine where and when we can use a radio wave-based, or vision-based system, from the last, our work proposes a basic comparison and highlights when it would can be used some of the principal kind of indoor navigation

system.

Researchers have conducted numerous studies to compare RF-based and vision-based indoor navigation systems, evaluating their performance under various conditions and applications. The following points summarize the key findings:

- **Accuracy:** Vision-based systems generally achieve higher precision and accuracy compared to RF-based systems, especially in environments with well-defined features and adequate lighting Huang et al. (2023).
- **Coverage:** RF-based systems typically offer wider coverage due to the ability of RF waves to penetrate obstacles Wu (2022).
- **Computational Complexity:** Vision-based systems require more computational resources compared to RF-based systems, which may limit their use on resource-constrained robots Haznedar et al. (2023).
- **Cost:** RF-based systems are generally less expensive to implement and maintain compared to vision-based systems Wu (2022).
- **Adaptability:** Vision-based systems tend to be more adaptable to changes in the environment, such as furniture rearrangement or lighting variations Morar et al. (2020).
- **Real-Time Performance:** RF-based systems typically offer lower latency, making them suitable for applications requiring real-time navigation Huang et al. (2023).

Of the above, in general, the reported comparisons are related to the performance of wave-based and vision-based systems, but do not provide a guide to determine which type of system is better for a specific application. So, we present a proposal of comparison, and discussion about when we can use some kind of indoor position system principally focused in get the best cost-benefit.

The goal and contribution of this research is to show the comparison of 1 radio frequency and 2 vision-based indoor navigation systems, to guide the election of the best indoor position system for a mobile robot, that allow us to reach the best cost-benefit.

## I. MATERIAL AND METHODS

### A. Radio frequency (RF) indoor position system

RF indoor navigation systems are based on the propagation and measurement of radio signals. Radio waves emitted from reference points or beacons interact with the environment and reach the target object. By measuring the signal strength, propagation time, or angle of arrival (AoA) of the received signals, the system can calculate the target object's position relative to the reference points Wu (2022).

Zare et al. (2021) mentions that Wi-Fi technology is the most popular in indoor positioning, and that almost all modern facilities have ready-to-use equipment. Therefore, there is no specific electronic card or device that is most commonly used in works that use Wi-Fi technologies. In general, the electronic devices used in these works are Wi-Fi access points, Wi-Fi routers, and mobile devices used to collect Wi-Fi signal data. Additionally, it is mentioned that applying Wi-Fi to a robot is technically simple, but typically requires combining with other optimization methods, such as neural networks, deep learning, and filters.

1) *Beacon System*: A beacon system is a wireless technology that utilizes small, battery powered transmitters to periodically send out Bluetooth or WiFi signals. These signals can be picked up by nearby devices with Bluetooth or WiFi capabilities, enabling location-based services and applications Vossen et al. (2017). In our case the reference nodes and the mobile node to be located, with which the indoor location tests were performed, are NodeMCU development boards, which use the ESP8266 processor and are capable of connecting to WiFi networks and can also be configured as an Access Point or as a Wi-Fi scanner to detect nearby networks. A more detailed description of the system can be consulted at García-Bautista and Cuaya-Simbro (2022).

### B. Vision indoor position system

Vision indoor navigation systems utilize cameras to capture images of the environment and extract meaningful information for localization and path planning. These systems use various computer vision techniques, such as feature detection, image matching, and SLAM (Simultaneous Localization and Mapping), to estimate the robot's position and orientation within the indoor environment Morar et al. (2020).

1) *Vision-Based Localization with Landmarks*: Vision-based localization techniques utilize cameras to extract meaningful information from the environment for determining the position and orientation of a mobile robot or device. Landmark-based localization is a prominent method that relies on identifying and matching visual features, such as distinctive shapes, patterns, or objects, to a pre-built map of landmarks, Huang et al. (2023).

2) *Vision-Based Localization with Centroids*: Vision-based localization techniques utilize cameras to extract meaningful information from the environment for determining the position and orientation of a mobile robot or device. Centroid-based localization is a prominent method that relies on identifying and

tracking the centroids of objects or features in the environment, Huang et al. (2023).

### C. Comparison measures to indoor position system

To compare indoor positioning systems based on radio frequency (RF) and computer vision (CV) and guide the selection of the best indoor positioning system for a mobile robot that achieves the best cost-benefit, we consider the following evaluation metrics..

1) *Accuracy*: Accuracy refers to the closeness of the estimated location to the actual location of the user. It is typically measured in meters or centimeters. A common way to measure the accuracy metric is:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N |d_i - p_i| \quad (1)$$

Where:  $N$ : Represents the total number of samples (number of location estimates).  $i$ : Index for iterating through each sample.  $d_i$ : Represents the actual distance for sample  $i$ .  $p_i$ : Represents the estimated distance for sample  $i$ . To calculate the average of the absolute differences between the actual distances ( $d_i$ ) and the estimated distances ( $p_i$ ) for all  $N$  samples we used the follow equation:

$$d_i = \sqrt{(x_o - x_i)^2 + (y_o - y_i)^2} \quad (2)$$

Where:  $d_i$ : Represents the distance between the real point and  $i$  point.  $x_o, y_o$ : Coordinates of the real point.  $x_i, y_i$ : Coordinates of the  $i$  point.

2) *Precision*: Precision refers to the consistency of the estimated location. It indicates how close the repeated estimates are to each other. And we used the next equation to calculate it:

$$\text{Precision} = \sqrt{\frac{\sum_{i=1}^N (d_i - \bar{d})^2}{N}} \quad (3)$$

Where:  $N$ : Represents the total number of samples (number of location estimates).  $i$ : Index for iterating through each sample.  $d_i$ : Represents the actual distance for sample  $i$ .  $\bar{d}$ : Represents the mean of all the actual distances ( $d_i$ ). We can use  $\frac{1}{N} \sum_{i=1}^N d_i$  to represent the mean.

3) *Cost-Effectiveness*: Cost-effectiveness refers to the balance between the system's performance and its overall cost. Cost-effectiveness metrics that we used:

*Initial Investment (II)*: Hardware costs, software licenses for the location engine and any additional software needed for integration, plus system installation and configuration costs, were calculated as follow:

$$II = C_{hardware} + C_{software} + C_{installation} \quad (4)$$

*Accuracy Under Variable Conditions (AUVc)*: This metric will depend on your specific application and environment. You can define it based on the average error under different conditions system's accuracy under different lighting conditions, presence of obstacles, or potential sources of signal interference that are specific to the semi-controlled environment, even

include any other parameter that is considered important and that may affect the operation of the system.

## II. RESULTS

### A. RF based indoor location system

First, we built a low cost indoor location system, using academic hardware, specifically ESP32 development card, the Figure 1 shows the system built using 3 ESP32, and a detail description of the develop of this system can be consult in García-Bautista and Cuaya-Simbro (2022).

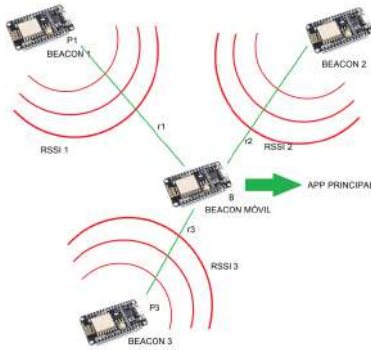


Fig. 1. Configuration of the location system using Beacons and a trilateration system.

1) *Vision-based location system with landmarks detection:* For develop this system, we used a cellular camera mounted in a mobile robot as a vision sensor and an accelerometer-gyro sensor MPU6050 to obtain the direction of the robot.

The original landmark technique uses a comparison between the histogram of the landmark obtained by the robot's camera and compared with the histogram of images stored in memory to obtain the localization. However, for this work, a modified technique was implemented in which no histogram comparison is used, but the landmarks were used to estimate the distance to them, as can be seen in Figure 2.

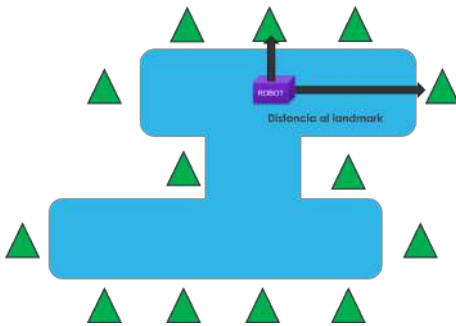


Fig. 2. Landmarks are placed by the environment and the robot's vision calculates the distance to them.

Using a simplified form of the triangular similarity technique and knowing:  $W$ : Real width of the object in cm,  $P$ : Apparent width in pixels,  $F$ : Focal length,  $D$ : Real distance from the object to the camera, the distance from the camera to the landmark was calculated with  $D = (W \times F)/P$ . And the

position was estimated by comparing the current state of the state space with the distance to the corresponding landmark and the current Yaw direction angle obtained by the MPU6050 sensor.

To implement this localization technique, a cell phone was adapted to the robot structure as a vision sensor and the video signal was transmitted to the computer via wireless with the free DroidCam application and the images were processed with the OpenCV library.

Sixteen landmarks were placed in the environment to obtain the approximate location in the environment, as shown in Figure 3.



Fig. 3. Environment for localization with landmarks.

This technique gave good results estimating the position when the robot was manually placed in the states. However, in the autonomous tests there were two major problems: the first one was that sometimes the Machine Vision confused the nearby landmarks returning an incorrect state. The second problem often occurred when changing direction. Since the camera kept getting images when turning, the algorithm would get confused and detect the wrong landmark, thus returning an incorrect distance and of course an incorrect state. That is, the system is sensible to movements of the robot and the detection of the position of the landmarks.

2) *Vision-based location system with centroid detection:* A localization technique based on Computer Vision with centroid detection was implemented, using a Logitech C920 HD Pro webcam as the vision sensor. The central idea is to use segmentation to extract from the image the shape, position and direction of the robot.

A Logitech C920 HD Pro webcam was placed at the top of the environment, i.e., on the ceiling, as a vision sensor, as shown in Figure 4.

Two geometric figures were placed on the top of the robot, which are used as position (rectangle) and direction (triangle) markers. The Algorithm 1 was used to obtain the markers from a enter environment image, this algorithm leverages computer vision techniques, segmentation and filtering, to knowing the location of the rectangle. The position is estimated by calculating the centroid  $(x, y)$  of the square position marker using the moment of inertia calculations and converting pixel coordinates to centimeters.



Fig. 4. Logitech C920 HD Pro webcam at the top of the environment.

Then we developed an application, using the Algorithm 1, with C# .NET Framework for Windows, which allows the use of the OpenCV Machine Vision library. The result of the segmentation and filtering of the robot markers is shown in Figure 5.



Fig. 5. Result of segmentation and filtering of placeholders marker.

3) *Environment*: The physical environment for testing the indoor location systems was a square space measuring 144.00 cm by 144.00 cm, with a total area of 20,736.00  $cm^2$ . The test points, as shown in Figure 6, were randomly selected within this space.

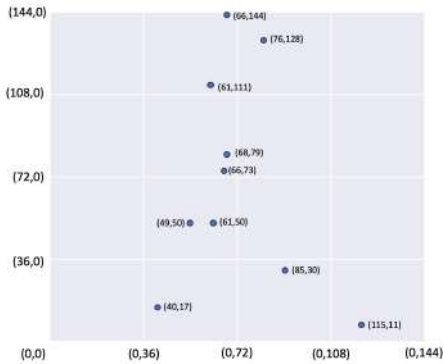


Fig. 6. Test environment. Where each point represent a referent point to compare the results obtained with each of indoor position systems built.

4) *Discussion*: Ten random points with  $x, y$  coordinates in centimeters were selected within the test environment, using the environment's origin  $(0, 0)$  as a reference. The accuracy and precision of each developed indoor location system were calculated at these points using the equations from Section 3 and presented in Table I.

### Algorithm 1 Segmentation and filtering of robot markers

```

CREATE a kernel structure element of size 2x2
CREATE a kernel2 structure element of size 1x1
CREATE a copy of the original image src named dst
CREATE a grayscale image grayImg
APPLY blur filter to image src and save in grayImg
CONVERT image src to grayscale and save in grayImg
APPLY threshold to image grayImg to make it binary
ERODE image grayImg
PERFORM morphological opening operation with kernel on gray-
Img
PERFORM morphological closing operation with kernel2 on gray-
Img
DILATE image grayImg
FIND contours in image grayImg and store them in contours and
hierarchy
if any contours found then
    for each contour i in contours do
        CALCULATE the area of contour i
        ADD the contour index and its area to the dictionary
    end for
end if
SORT the dictionary by contour areas from largest to smallest and
take the largest
for each contour in the item (only one, since we take the largest)
do
    GET the index (key) and area of the contour
    if the area is greater than 100 then
        GET the current contour (CurrentImage) and its sides
        APPROXIMATE the contour CurrentImage to a polygon
shape and save it in Sides
        CALCULATE the moments of the current contour
        CALCULATE the centroid:
        ptCenter.X = moments.M10 / moments.M00
        ptCenter.Y = moments.M01 / moments.M00
        GET the number of sides of the figure (NumOfSides)
        if NumOfSides is equal to 4 (it is a square or rectangle)
then
            ASSIGN the centroid ptCenter to robot
            DRAW the contour on the image src
            CALCULATE the minimum bounding rectangle of the
contour
            GET the corners of the rectangle and convert them to
integers
            DRAW the rectangle and the centroid on the image src
        end if
    end if
end for

```

To assess cost-effectiveness, initial investment (II) was calculated for each system:

- $II_{(RF)} = 400 + 0 + 500 = 900MXN \approx 50USD$ , where Hardware = 3 ESP32 boards, wire, and power banks, Software we used only free software, and Installation = cost of attaching 3 cards with batteries to the wall.
- $TI_{Landmarks} = 2000 + 0 + 800 = 2800MXN \approx 155USD$  where Hardware = 1 SAMSUNG Celular Galaxy A05 4GB RAM 64GB3, cables, and accelerometer-gyroscope sensor MPU6050, Software we used only free software, and Installation = MPU6050 cellular and sensor configuration costs, printing and

TABLE I  
REPORT OF SYSTEMS' ACCURACY AND PRECISION.

Real position (cm)		Radio frequency system (cm)			Landmark vision system (cm)			Centroids vision system (cm)		
$x_o$	$y_o$	$x_i$	$y_i$	$d_i$	$x_i$	$y_i$	$d_i$	$x_i$	$y_i$	$d_i$
49.0	50.0	16.0	15.0	68.0	38.0	65.0	26.0	53.0	56.0	10.0
65.0	73.0	26.0	38.0	75.0	58.0	58.0	23.0	61.0	77.0	9.0
66.0	144.0	41.0	64.0	105.0	75.0	120.0	33.0	61.0	150.0	11.0
40.0	17.0	20.0	18.0	21.0	49.0	29.0	21.0	34.0	15.0	8.0
85.0	30.0	5.0	20.0	90.0	97.0	11.0	31.0	81.0	25.0	9.0
76.0	128.0	31.0	48.0	125.0	68.0	130.0	10.0	73.0	125.0	6.0
68.0	79.0	48.0	44.0	55.0	54.0	87.0	22.0	72.0	83.0	8.0
115.0	11.0	45.0	9.0	72.0	125.0	15.0	14.0	111.0	15.0	8.0
61.0	50.0	6.0	20.0	85.0	49.0	41.0	21.0	63.0	52.0	4.0
61.0	111.0	36.0	61.0	75.0	69.0	90.0	29.0	56.0	114.0	8.0
Accuracy ( $\bar{d}$ )				77.1	23.0			8.1		
Precision (Standard Deviation)				28.0	7.2			2.0		

placement of Landmarks.

- $TI_{Centroids} = 7000 + 0 + 1500 = 8500MXN \approx 475USD$  where Hardware = 1 laptop DELL Latitude 5490, Core i7-8650U, 16GB DDR4 RAM, 512 GB SSD, Logitech C920 HD Pro, cabling, and support system, Software we used only free software, and Installation = laptop and webcam configuration costs, development of the support system for camera placement.

Accuracy Under Variable Conditions (AUVc) was evaluated considering factors such as lighting, obstacles, signal interference, system autonomy, response time, scalability, and instrumentation complexity. The results are summarized in Table II.

TABLE II  
REPORT OF SYSTEMS' AUVc.

	Light variations	Presence of obstacles	Signal interference	
RF	No effect	Slight affect	High effect	
Landmarks	High effect	High effect	Slight affect	
Centroids	High effect	No effect	No effect	
	Autonomy duration (hours)	Response time (s)	Scalability	Instrumentation complexity
RF	<72	<.3	Easy	Easy
Landmarks	<72	between (.5, 1)	Complicate	Moderate
Centroids	<8	>1	Moderate	Complicate

The vision systems demonstrated superior performance in indoor object localization, aligning with previous research. However, the landmark-based system exhibited weaknesses in determining y-axis position, likely due to robot movement and calibration issues. Improving robot motion control could enhance this system's accuracy.

While vision systems excelled in accuracy, the RF system offered a significantly lower initial investment. Although it showed lower accuracy and precision in this study, utilizing more advanced hardware like Minew MBM01 beacons could improve performance while maintaining a budget-friendly solution still have an II value of less than 250 USD, which could support the choice of this type of technology when there is a limited budget for the implementation of an indoor location system.

Table II provides a qualitative comparison of the systems under varying conditions. For environments with changing lighting and obstacles, where high accuracy is not critical, the RF system might be suitable (e.g., production plants, mines). For applications demanding long battery life and easy implementation, the RF system is also a viable option. Con-

versely, vision systems excel in accuracy and precision-critical applications such as automated manufacturing or patient care, despite potential limitations in response time.

The comparative analysis presented in Tables I and II can serve as a valuable resource for researchers and practitioners considering indoor location system development. By carefully evaluating system requirements and available resources, informed decisions can be made regarding the most appropriate technology.

### III. CONCLUSIONS

This study presented the construction and evaluation of two common indoor location systems: RF-based and vision-based. The quantitative analysis, including accuracy and precision metrics, aligned with previous research findings.

A qualitative evaluation and comparison of the different location systems has been proposed and carried out, complementing the quantitative evaluation. Together, both evaluations can be used as a guide to determine the best type of system to use according to the application, environment, situations and characteristics where the location system will be used, and examples have been given of how, based on the results, one or the other location system can be used.

The results demonstrate that the optimal system choice depends on specific application requirements, such as accuracy needs, environmental conditions, and cost constraints. While vision systems generally offer higher precision, RF systems provide advantages in terms of cost and robustness in challenging environments.

Future research could explore hybrid systems combining RF and vision technologies to leverage the strengths of both approaches.

### IV. ACKNOWLEDGEMENTS

This work has been carried out thanks to the support granted by Tecnológico Nacional de México / ITS del Oriente/Occidente del Estado de Hidalgo. In the same way, the joint effort of all the authors of the manuscript.

### REFERENCES

- S. Fusic, T. Sugumari, et al. A review of perception-based navigation system for autonomous mobile robots. *Recent Patents on Engineering*, 17(6):13–22, 2023.
- J. A. García-Bautista and G. Cuaya-Simbro. Revista internacional socio-innova-tec del altiplano. *Revista Internacional Socio-Innova-Tec del Altiplano*, 2(2):1–16, 2022.
- Y. Haznedar, G. Z. G. Aydin, and Z. Turgut. Hybrid indoor positioning for smart homes using wifi and bluetooth low energy technologies. *J. Ambient Intell. Smart Environ.*, 15(1):63–87, jan 2023. ISSN 1876-1364. doi: 10.3233/AIS-220484. URL <https://doi.org/10.3233/AIS-220484>.
- J. Huang, S. Junginger, H. Liu, and K. Thurow. Indoor positioning systems of mobile robots: A review. *Robotics*, 12(2), 2023. ISSN 2218-6581. doi: 10.3390/robotics12020047. URL <https://www.mdpi.com/2218-6581/12/2/47>.

- A. Morar, A. Moldoveanu, I. Mocanu, F. Moldoveanu, I. E. Radoi, V. Asavei, A. Gradinaru, and A. Butean. A comprehensive survey of indoor localization methods based on computer vision. *Sensors*, 20(9):2641, 2020.
- R. Otero, S. Lagüela, I. Garrido, and P. Arias. Mobile indoor mapping technologies: A review. *Automation in Construction*, 120:103399, 2020. ISSN 0926-5805. doi: <https://doi.org/10.1016/j.autcon.2020.103399>. URL <https://www.sciencedirect.com/science/article/pii/S0926580520309791>.
- G. Vossen, S. Dillon, F. Schomm, and F. Stahl. A classification framework for beacon applications. *Open Journal of Internet of Things (OJIOT)*, 3(1):1–11, 2017.
- P. Wu. Comparison between the ultra-wide band based indoor positioning technology and other technologies. *Journal of Physics: Conference Series*, 2187(1):012010, feb 2022. doi: [10.1088/1742-6596/2187/1/012010](https://doi.org/10.1088/1742-6596/2187/1/012010). URL <https://dx.doi.org/10.1088/1742-6596/2187/1/012010>.
- M. Zare, R. Battulwar, J. Seamons, and J. Sattarvand. Applications of wireless indoor positioning systems and technologies in underground mining: A review. *Mining, Metallurgy & Exploration*, pages 1–16, 2021.

# Método de Continuación Homotópica para la Planeación de trayectorias en el robot Turtlebot3

María de las Nieves Juárez Manny  
TecNM / Instituto Tecnológico de La Laguna  
Torreón, Coahuila, México  
mary\_manny@hotmail.com

Alejandro Enrique Dzul López  
TecNM / Instituto Tecnológico de La Laguna  
Torreón, Coahuila, México  
aedzul@lalaguna.tecnm.mx

**Resumen**—En este artículo se implementa el Método de Continuación Homotópica (MCH), en el robot Turtlebot3, para atender el problema de planeación de trayectorias. Lo anterior es logrado utilizando el ambiente de programación ROS y el lenguaje de programación Python. El MCH es un método utilizado en la planificación de trayectorias debido a que garantiza encontrar una trayectoria exitosa, libre de colisión, y que puede implementarse con bajos recursos de cómputo, como en las tarjetas de desarrollo SBC (Single Computer Board). A través de resultados en simulación se muestra la eficacia del método.

**Keywords**—Método de continuación homotópica, robot diferencial, ROS.

## I. INTRODUCCIÓN

Un robot, para su operación, requiere de varios módulos entre los que destacan el de percepción, el de planeación y el de control; en el caso del desarrollo de robots móviles autónomos, la planeación de trayectorias es un área de interés, debido a que es una de las principales características en dicho tipo de robots [1].

El sistema de navegación determina la manera de transitar por el espacio de operación, planificando la ruta a seguir de manera confiable, es decir, evitando obstáculos estáticos y/o dinámicos. Así mismo, dicha ruta o camino deberá cumplir con ciertos requerimientos, por ejemplo, que consuma la menor cantidad de recursos, etc. [2].

En la literatura, el problema de planificación se aborda dependiendo de la información del entorno, en el cual se moverá el robot. Así, al contar con un conocimiento completo del entorno, el problema se clasifica como planeación de movimiento global, en caso contrario, como planeación de movimiento local. Entre los métodos para modelar el entorno están los métodos de mallas, topológicos, geométricos, etc. [3].

En el método de continuación homotópica, la representación del entorno se realiza mediante un sistema de ecuaciones representativas, donde los obstáculos pueden ser considerados como círculos o alguna otra forma geométrica, teniendo así un sistema de ecuaciones del que se busca obtener una solución que genere una trayectoria libre de colisiones [4]-[5].

Este trabajo fue apoyado parcialmente por el CONAHCyT y el TecNM.

El stack de navegación de ROS (Robot Operating System) incorpora algoritmos de planificación tanto global como local, con el fin de dar soporte a la navegación [6]. Sin embargo, solamente algoritmos básicos se encuentran definidos para la planificación global, incluyendo el algoritmo de Dijkstra, A\*. En este trabajo, se tiene como objetivo la incorporación del MCH como un algoritmo de planificación global en ROS.

El resto del documento está organizado como sigue: en la sección II se abordan algunos conceptos básicos sobre robots móviles con ruedas, describiendo el modelo cinemático que es utilizado en este trabajo para el robot TurtleBot3. Posteriormente, en la sección III, se expone el método de continuación homotópica y cómo éste es aplicado para generar la trayectoria libre de obstáculos. En la sección IV se proporcionan las características principales del robot TurtleBot3, se describe el algoritmo para la implementación del método de continuación homotópica en la plataforma de desarrollo ROS, y se muestran los resultados obtenidos. Finalmente, en la sección V se presentan las conclusiones y trabajo a futuro.

## II. ROBOT MÓVIL DIFERENCIAL

Los robots móviles se caracterizan por contar con un mecanismo que les permite desplazarse en su entorno libremente. Este mecanismo está ligado al cuerpo rígido (base o chasis) y es llamado sistema de locomoción [7].

En el caso de los robots móviles con ruedas (RMR), la locomoción está compuesta por un sistema de ruedas acopladas al chasis, la cual proporciona el movimiento relativo con respecto al piso. El tipo y número de ruedas, así como su distribución en el chasis, determinan la movilidad de un RMR.

En este trabajo se utiliza como RMR al robot Turtlebot3 Waffle Pi, el cual es un robot diferencial, es decir, es un vehículo con dos ruedas fijas que comparten un mismo eje de rotación, además de una rueda orientable cuya función es mantener al robot balanceado.

Las dos ruedas fijas son controladas de manera independiente, lo que permite no sólo que el robot avance, sino también que pueda rotar sobre sí mismo (alrededor de un eje perpendicular al piso que pasa por el punto medio entre las ruedas). Un robot móvil diferencial es uno de los tipos más

básicos pero más utilizados de RMR; esto se debe en gran medida a que, por lo general, el robot ocupa un área más pequeña en comparación con otros tipos de RMR. Además de las dos ruedas fijas coaxiales, es común que un robot diferencial cuente con una tercera rueda, del tipo orientable descentrada; esta rueda no está actuada y solamente sirve como punto de apoyo, así que es común despreciarla en el modelo cinemático.

La Figura 1 muestra el diagrama cinemático de un RMR diferencial. En la figura, las coordenadas  $(x, y)$  dan la posición del punto  $P$  (que se supone que está ubicado en el punto medio entre las ruedas fijas) con respecto al marco de referencia fijo (inercial)  $\Sigma_o(X_o, Y_o)$ . Por otra parte, el marco  $\Sigma_m(X_m, Y_m)$  está unido al robot, tiene como origen el punto  $P$  y su eje  $X_m$  apunta hacia el frente del robot, de modo que el ángulo  $\theta$  da la orientación con respecto al marco  $\Sigma_o$ . Los parámetros  $r$  y  $L$  corresponden al radio de las ruedas y a la distancia entre el punto  $P$  y cada una de las ruedas, respectivamente.

En un RMR diferencial, generalmente se considera que las señales de entrada (es decir, las que producen el movimiento) son las velocidades angulares en las dos ruedas actuadas. Si  $\varphi_i$  y  $\varphi_d$  son los desplazamientos angulares de la rueda izquierda y derecha, respectivamente, entonces  $\omega_i = \dot{\varphi}_i$  y  $\omega_d = \dot{\varphi}_d$  serían las señales de entrada al robot. Cada una de estas velocidades angulares aportan al avance del punto de contacto entre la rueda y el piso; estas velocidades son representadas en la Figura por los vectores  $v_i$  y  $v_r$ , estando dadas por

$$v_i = r\omega_i, \quad v_d = r\omega_d.$$

Se le llama *uniciclo* a todo vehículo que utiliza una sola rueda para moverse sobre el piso. Teóricamente, el uniciclo requiere de dos entradas: la velocidad lineal  $v$  en la dirección de la rueda, y la velocidad angular sobre el eje vertical  $\omega$ , que permite generar un cambio de dirección. Dado que la mayoría de los algoritmos de planificación de movimiento entregan una velocidad lineal y una velocidad angular, es conveniente trabajar con un modelo que emplee este tipo de entradas al sistema.

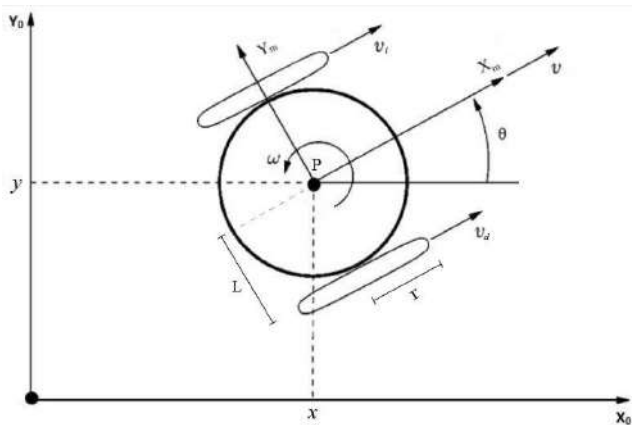


Figura 1. Diagrama cinemático de un RMR tipo diferencial.

En el caso del robot diferencial, la transformación de las velocidades lineales de avance de las ruedas y la velocidad lineal y angular del robot se obtienen usando las siguientes expresiones:

$$v = \frac{v_d + v_i}{2}, \quad \omega = \frac{v_d - v_i}{2L}. \quad (1)$$

Una vez hecha esta transformación de velocidades, el robot diferencial puede ser visto como un uniciclo (y por eso también se le conoce como *robot móvil tipo uniciclo*). Estas  $v$  y  $\omega$  son las magnitudes que pasan por el punto  $P$  y apuntan en la dirección de avance y de rotación del robot (ver la Figura 1).

Con el *modelo cinemático directo de velocidad* (MCDV) de un RMR, se obtiene la variación de la pose del robot, es decir,  $\dot{x}$ ,  $\dot{y}$  y  $\dot{\theta}$  en términos de las velocidades de entrada al robot. Si esas velocidades de entrada son  $v$  y  $\omega$ , entonces el MCDV para el robot uniciclo es [8]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (2)$$

o, si se usa la relación (1), se llega al MCDV en el que las entradas son  $v_i$  y  $v_d$ .

En [1] y [8] se pueden encontrar más detalles sobre el modelado cinemático de los robots móviles en general, y con diferentes mecanismos de locomoción, además del modelo diferencial.

### III. MÉTODO DE CONTINUACIÓN HOMOTÓPICA

Los métodos de homotopía también se conocen como métodos de continuación o métodos de deformación continua; estos se utilizan para resolver sistemas de ecuaciones algebraicas no lineales, de la forma:

$$f(x) = 0 : \quad \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (3)$$

Los métodos de homotopía crean una deformación continua en el sistema  $f(x)$ , añadiendo una función  $G(x)$ , convirtiendo el sistema original (3) a otro de la forma:

$$H(x, \lambda) = \lambda f(x) + (1 - \lambda)G(x) = 0. \quad (4)$$

$G(x)$  determina el tipo de homotopía que se utiliza, tal es el caso, de homotopía de punto fijo, donde  $G(x) = x - x^0$  [9]; en el método de homotopía de Newton  $G(x) = f(x) - f(x_0)$ , sustituyendo en (4), se tiene:

$$H(x, \lambda) = \lambda f(x) + (1 - \lambda)(f(x) - f(x_0)) = 0, \quad (5)$$

donde  $\lambda$  se conoce como parámetro de homotopía,  $x_0$  es un punto inicial,  $H(x, \lambda) \in \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ ,  $x \in \mathbb{R}^n$  y  $\lambda \in [0, 1]$ . Cuando  $\lambda = 0$ , el sistema se reduce al problema  $H(f(x), 0) = f(x) - f(x_0) = 0$ . En tanto, si  $\lambda = 1$ , se obtiene el problema original  $H(f(x), 1) = f(x) = 0$ . Las soluciones que se obtienen del sistema de homotopía durante la deformación del parámetro  $\lambda$ , desde  $\lambda = 0$  hasta  $\lambda = 1$ , forman una familia de curvas homotópicas, conocidas como curvas de solución [9]- [10].

### III-A. Método de homotopía para planeación de trayectorias

En este trabajo, el método de homotopía se aplica al problema de planeación de trayectorias en un espacio de dos dimensiones, donde se considera que el robot parte de un punto inicial  $A$  y debe llegar a un punto final  $B$ , donde el espacio está normalizado, por lo que  $A = (0, 0)$  y  $B = (1, 1)$ .

Los obstáculos están representados por la ecuación:

$$W(x, y) = \sum_{i=0}^{k-1} \frac{p_i}{Ob_i(x, y)}, \quad (6)$$

donde  $k$  es el número de obstáculos en el mapa,  $p_i$  es un parámetro de repulsión del  $i$ -ésimo obstáculo,  $Ob_i(x, y)$  describe la forma de los obstáculos; en este trabajo se considerarán obstáculos de forma circular, por lo que cada obstáculo estará representado por la ecuación:

$$C_i(x, y) = (x - x_i)^2 + (y - y_i)^2 - r_{c_i}^2 = 0, \quad (7)$$

donde  $(x_i, y_i)$  es el centro del obstáculo de forma circular con radio  $r_{c_i}$ .

El sistema de ecuaciones con solución en el punto  $B(x_b, y_b)$ , es formulado como la intersección de dos líneas con pendientes  $m_1$  y  $m_2$ :

$$\begin{aligned} f_1(x, y) &= -y + m_1x + (b - m_1a) = 0, & x, y \in \mathbb{R}, \\ f_2(x, y) &= -y + m_2x + (b - m_2a) = 0, & x, y \in \mathbb{R}, \end{aligned} \quad (8)$$

es decir, en el espacio normalizado  $[0, 1]$ ,  $f_1(x_b, y_b) = 0$  y  $f_2(x_b, y_b) = 0$ . Los obstáculos presentes en el espacio se incorporan en el sistema de ecuaciones anterior, representando una singularidad en  $L_2$ , quedando el sistema transformado como:

$$\begin{aligned} L_1(x, y) &= f_1(x, y) = 0, \\ L_2(x, y) &= f_2(x, y) + W(x, y) - Q = 0. \end{aligned} \quad (9)$$

El método de homotopía crea curvas continuas con un efecto de atracción al punto de intersección en  $B$ , y un efecto de repulsión en los obstáculos [4]. Sustituyendo en (5):

$$\begin{aligned} H_1(f_1(x, y), \lambda) &= f_1(x, y) - (1 - \lambda)f_1(x_0, y_0), \\ H_2(f_2(x, y), \lambda) &= f_2(x, y) - (1 - \lambda)f_2(x_0, y_0), \end{aligned} \quad (10)$$

en donde  $(x_0, y_0)$  es el punto inicial  $A$ .

Para trazar la curva de solución, se ha seguido el esquema propuesto en [4], el cual utiliza el algoritmo de seguimiento hipersférico [11] y, como esquema predictor-corrector, se aplica el método de Newton-Rapson.

## IV. IMPLEMENTACIÓN NUMÉRICA

En este trabajo se utilizó el robot TurtleBot3 [12], el cual es un robot móvil diferencial de bajo costo; para su operación se utiliza la plataforma de desarrollo ROS. El primer TurtleBot fue creado en la empresa Willow Garage en 2010. La versión TurtleBot3 es comercializada en tres opciones, denominadas Burger, Waffle y Waffle Pi; la versión con la que se trabajó es

la configuración Waffle Pi, ver Figura 2. Algunos parámetros importantes se muestran en la TABLA I.

El robot TurtleBot3 cuenta con una tarjeta Raspberry Pi modelo B+, Figura 3, que hace las veces del procesador central; dicha tarjeta se comunica con la tarjeta OpenCR, Figura 4, la cual se comunica con los motores Dynamixel de Robotis, mediante un protocolo de comunicación Half Duplex UART, para el control de velocidad y aceleración.

La tarjeta Raspberry Pi 3 Model B+, contenida en el TurtleBot3, cuenta con las siguientes especificaciones:

- Un chip Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC con 1.4GHz.
- 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2.
- Puerto CSI para conectar una cámara Raspberry Pi.
- Un puerto Micro SD para cargar el SO y almacenar datos.



Figura 2. Robot TurtleBot3 Waffle Pi.

TABLA I  
ESPECIFICACIONES DEL ROBOT TURTLEBOT3

Descripción	Valor	Unidad
Diámetro exterior	0.340	[m]
Altura	0.092	[m]
Peso	3.58	[kg]
Distancia entre las ruedas	0.235	[m]
Diámetro de las ruedas	0.072	[m]
Resolución del encoder	508.8	[pulsos/rev]



Figura 3. Raspberry Pi model B+.

Por otro lado, la tarjeta OpenCR 1.0 (Open-source Control Module for ROS), es una tarjeta ARM Cortex-M7 con un chip de la serie STM32F7, la cual controla los motores de las llantas

del robot; el software Dynamixel SDK también se encuentra disponible bajo licencia de Open Source en [13].

La tarjeta OpenCR 1.0, maneja cálculos de punto flotante y soporta grandes cantidades de datos; para garantizar este alto desempeño funciona a 216 MHz. Cuenta con diferentes periféricos para dar soporte a diferentes dispositivos, entre ellos Arduino, los motores Dynamixel (via TTL y RS485), sensores como la cámara Raspberry Pi y el lidar LDS-01.



Figura 4. Tarjeta OpenCR1.0.

Además, el robot utiliza un sensor laser (lidar) que podría ser utilizado para realizar la tarea de SLAM, para la detección y adquisición de datos del entorno, con un rango de 360°.

#### IV-A. ROS

ROS es un entorno de trabajo de código libre que cuenta con un conjunto de librerías y herramientas de código abierto. ROS se creó pensando en desarrollar software robusto para sistemas robóticos; en el entorno de trabajo de ROS cada módulo tiene una función que puede ser reemplazada o actualizada sin que los demás módulos se vean afectados [14].

ROS proporciona los servicios estándar de un sistema operativo, tales como abstracción del hardware, control de dispositivos de bajo nivel, paso de mensajes entre procesos, etc. Un sistema desarrollado en ROS cuenta con los siguientes elementos de software:

**Nodo (node):** Un nodo es un proceso que realiza algún tipo de procesamiento en el sistema. Los nodos se combinan dentro de un grafo para compartir información entre ellos. El grafo se forma uniendo los nodos mediante tópicos como se muestra en la Figura 5. Por ejemplo, se puede tener un nodo para controlar un sensor láser (lidar), otro nodo para enviar la señal a los motores de un robot, y otro nodo para la construcción de mapas. Cada nodo tendrá sus propios algoritmos para procesar la información que recibe.

**Tópico (topic):** Los tópicos son canales de comunicación entre los nodos, los cuales llevan información, es decir, datos relevantes para que puedan ser procesados por los nodos; a estos canales se pueden subscribir más de un nodo. Por ejemplo, stage, que es un simulador de robots, tiene la capacidad de emitir un tópico corresponde a la odometría del robot. El nodo que emite información no controla a otro nodo que está suscrito a dicho tópico. La información de este tipo es llamada asíncrona, es decir, se emite el dato cuando sea conveniente sin esperar que se responda a él. A su vez, ROS

permite la comunicación síncrona, mediante un mecanismo de petición/respuesta (*request/response*), para lo cual se utilizan servicios.

**Servicios (service):** Una cualidad importante en los servicios es que se encuentran bloqueados, por lo que interrumpen la ejecución del programa hasta que reciben la respuesta al servicio que han solicitado. Así, una vez que se ha hecho un llamado a un servicio, éste procesará hasta que una respuesta sea enviada.

**Paquete (package):** El software en ROS está organizado en paquetes. Un paquete puede contener un nodo, una librería, un conjunto de datos, o cualquier cosa que pueda constituir un módulo. Los paquetes pueden organizarse en pilas (stacks).

**Pila (stack):** Conjunto de nodos que juntos proporcionan alguna funcionalidad (por ejemplo, la pila de navegación).

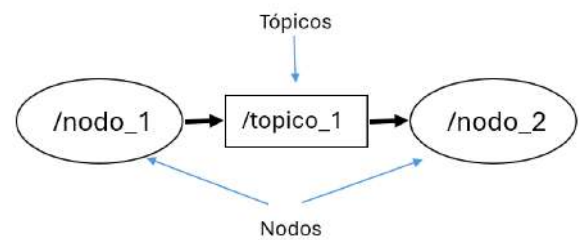


Figura 5. Comunicación entre nodos mediante tópicos.

Para la implementación del método de homotopía, se ha creado un paquete en ROS llamado *homotopy*, entre los nodos más relevantes se encuentran el nodo *homotopy* y *Turtlebot3*. La Figura 6 muestra el esquema de comunicación entre los nodos *homotopy* y *Turtlebot3*.

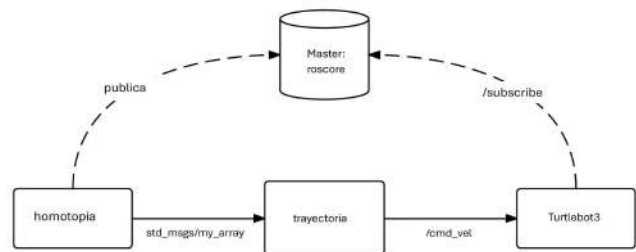


Figura 6. Comunicación entre nodos mediante tópicos.

En el nodo *homotopia* se ha implementado el MCH, con el Algoritmo 1; las variables de entrada son valores numéricos, los cuales han sido tomados de [4] y [9]; como resultado, se obtiene una estructura de datos, que es la trayectoria en un formato  $(x_i, y_i), i = 0, \dots, N$ ; la estructura es un parámetro de entrada para el nodo *Turtlebot3*, el cual se encarga de enviar las velocidades lineales y angulares al robot para recorrer la trayectoria, para lo cual, se utilizan las ecuaciones en (1) y (2), como se detalla en el Algoritmo 2.

---

**Algoritmo 1** Algoritmo MCH para generación de trayectoria libre de colisión

---

**Entrada:**

Pendientes:  $m_1, m_2$ ;  
Obstáculos:  $Ob_1, Ob_2, \dots, Ob_k$   
Parámetros de repulsión:  $p_1, p_2, \dots, p_k$ ;  
Tolerancia:  $tol$ ;  
Máximo de iteraciones:  $Max$ .

**Salida:** Lista con las soluciones encontradas:  $t_0 = (x_0, y_0), t_1 = (x_1, y_1), \dots, t_n = (x_N, y_N)$ .

Sustituir los valores de entrada en (9).  
Obtener el sistema de ecuaciones homotópicas (10) y el Jacobiano  $J(x)$  a partir de (10).  
Inicializar  $x_j$  con la aproximación inicial  $x_0$   
Obtener  $J(x_j)$  y  $H(x_j)$   
Evaluar el determinante de  $J(x_j)$   
**while**  $i < Max || H(x_j) \approx tol$  **do**  
  **if**  $i == 1$  **then**  
     $t_1 = (x_0, y_0, 0)$   
     $(c_x, c_y, c_\lambda) = (x_0, y_0, 0)$   
    Cálculo del vector predictor  
     $(x_p, y_p, \lambda_p) = (c_x, c_y, c_\lambda)$   
  **end if**  
  Aplicar Newton-Raphson  
   $t_{i+1} = (x_{i+1}, y_{i+1}, \lambda_{i+1})$   
  **if**  $i > 1$  **then**  
    Detectar corrección hiperesférica  
    Corregir reversión  
  **end if**  
   $(c_x, c_y, c_\lambda) = (x_{i+1}, y_{i+1}, \lambda_{i+1})$   
**end while**

---

#### IV-B. Resultados en simulación

En simulación, se tomó en cuenta un área de  $3 \times 3$  [m], como obstáculos se consideraron bloques de unicel en forma cilíndrica, los cuales serán obstáculos de forma circular en la implementación del método de Homotopía. La Figura 7 muestra la configuración del espacio de trabajo con dos y tres obstáculos, sin embargo, la simulación se realizó para el primer caso.

Como computadora, se utilizó una laptop Dell Pentium N3710 a 1.60 GHz, la cual cuenta con sistema operativo Linux Ubuntu 16.04, y ROS Kinetic Kame; el código se realizó con Python3.

En la TABLA II se muestran las coordenadas que se utilizaron para los obstáculos  $Ob_1$  y  $Ob_2$ . En la Figura 8 se muestran las tres curvas de solución: mientras las curvas  $C_1$  y  $C_2$  rodean los obstáculos, la curva  $C_3$  marca una trayectoria que va desde el punto inicial  $(0, 0)$  al punto final  $(1, 1)$ , sin colisionar con los obstáculos representados por los puntos en azul.

En la Figura 9, se muestra una captura de Gazebo, en el que se ha simulado la trayectoria obtenida para TurtleBot3.

---

**Algoritmo 2** Recorrido de trayectoria

---

**Entrada:**

Trayectoria: Lista de puntos de la trayectoria en formato  $(x_i, y_i), i = 0, \dots, N$ .

**Salida:** *True* si el recorrido fue exitoso.

Obtener desde *trayectoria*, la orientación deseada en cada posición  $p_i = (x_i, y_i)$   
**for**  $i = 1, 2, \dots, N$  **do**  
  Obtener la posición del robot  $p_r = (x_r, y_r)$  y orientación del robot *angulo\_r*.  
  Obtener el error entre la pose actual del robot y la pose deseada  $p\_err$ .  
  Obtener el error entre la orientación actual del robot y el ángulo de orientación deseada en el punto  $p_i$  y asignarlo a *angulo\_err*.  
  
  **if**  $p\_err == 0 \ \&\& \ \text{angulo\_err} == 0$  **then**  
    Enviar velocidades 0 a */cmd\_vel*  
  **else**  
    Modificar la velocidad lineal y angular en el mensaje *Twist*  
    publicar a */cmd\_vel* el mensaje *Twist*  
  **end if**  
**end for**  
Detener el robot.  
Regresar *True*.

---



Figura 7. Espacio para pruebas experimentales.

TABLA II  
PARÁMETROS DE LOS OBSTÁCULOS DEL EXPERIMENTO

Obstáculo	$x_i$	$y_i$	$r_{c_i}$	$p_i$
$Ob_1$	0.35[m]	0.35[m]	0.1[m]	0.0001
$Ob_2$	0.7[m]	0.6[m]	0.1[m]	0.0001

Los resultados obtenidos, muestran que el método de continuación homotópica genera una trayectoria libre de colisión, por lo que el robot móvil puede desplazarse desde el punto inicial  $A = (0, 0)$  al punto final  $B = (1, 1)$  sin colisionar con los obstáculos cilíndricos.

La distancia entre el robot y el obstáculo  $Ob_1$  se muestra en la Figura 10, y la distancia entre el robot y el obstáculo  $Ob_2$

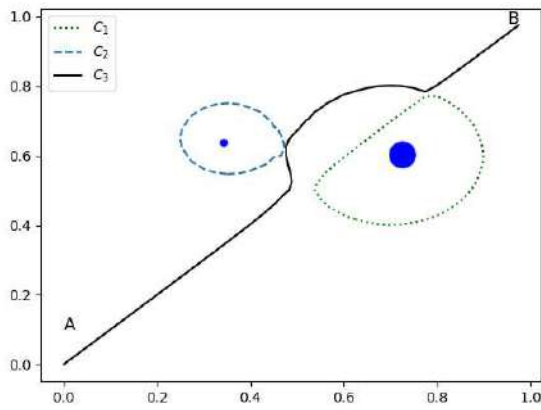


Figura 8. Curvas homotópicas para el caso de dos obstáculos.

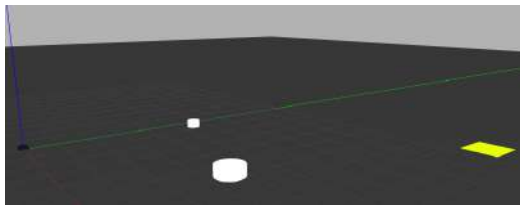


Figura 9. Mundo para simulación en Gazebo.

se muestra en la Figura 11. Se puede observar que la distancia entre el robot y los obstáculos disminuye conforme se acerca a los obstáculos, y una vez que los ha pasado, se incrementa la distancia, es decir, numéricamente el método generó una trayectoria libre de colisiones.

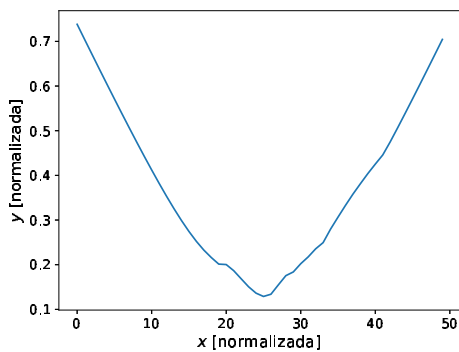


Figura 10. Distancia entre el centro del robot y el obstáculo  $Ob_1$

## V. CONCLUSION

En este trabajo se implementa el método de continuación homotópica en un algoritmo de planeación de trayectorias, donde se consideran obstáculos de forma circular y donde las pruebas numéricas se realizan con el lenguaje python3. La trayectoria obtenida fue implementada para el robot TurtleBot3.

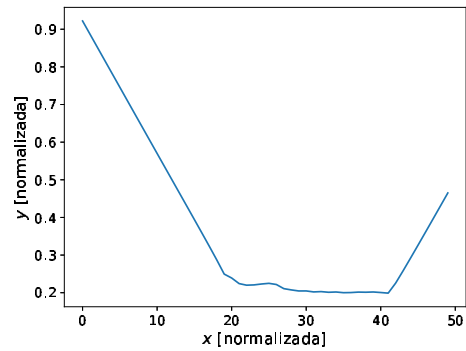


Figura 11. Distancia entre el centro del robot y el obstáculo  $Ob_2$ .

Los resultados obtenidos muestran una trayectoria libre de colisiones en un ambiente con obstáculos estáticos. Como trabajo futuro se implementará un paquete para el método de continuación homotópica que consideré obstáculos con otras formas diferentes a la circular, además de obtener la validación con resultados experimentales.

## REFERENCIAS

- [1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, "Introduction to Autonomous Mobile Robots," MIT Press, 2011.
- [2] LaValle, Steven M., "Planning Algorithms," 2006, Cambridge University Press, USA.
- [3] Lixing Liu and Xu Wang and Xin Yang and Hongjie Liu and Jianping Li and Pengfei Wang, 2023, "Path planning techniques for mobile robots: Review and prospect," Expert Systems with Applications.
- [4] Diaz-Arango, Gerardo and Vazquez-Leal, Hector and Hernandez-Martinez, Luis and Sanz, M.T. and Sandoval-Hernandez, Mario, 2017, "Homotopy Path Planning for Terrestrial Robots Using Spherical Algorithm," IEEE Transactions on Automation Science and Engineering.
- [5] G. Diaz-Arango et al., "A Software Tool for Teaching the Homotopy-Based Path Planning Method for Mobile Robot Applications," 2023 20th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico, 2023, pp. 1-6, doi: 10.1109/CCE60043.2023.10332846.
- [6] ROS.org, "Pila de navegación," Ago. 2024. [Online]. Available: <https://wiki.ros.org/navigation>.
- [7] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, "Robotics: Modelling, Planning and Control," Springer, 2009.
- [8] C. Canudas de Wit, B. Siciliano, and G. Bastin (Eds.), "Theory of Robot Control," Springer, 1997.
- [9] Quemada-Villagómez Miriam Lucero, López-González María de la Luz, Oliveros-Muoz Juan Manuel, Jiménez-Islas Hugo, 2022, "Enseñanza del método de continuación homotópica con seguimiento hipersférico para estudiantes de ingeniería." Acta univ.
- [10] Shenyu Liu and Mohamed Ali Belabbas, 2019, "A Homotopy Method for Motion Planning," Optimization and Control, Arxiv.
- [11] K. Yamamura, "Simple algorithms for tracing solution curves," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Aug. 1993.
- [12] Robotis Co. Ltd, "Robot Platform - Turtlebot3," Ago. 2024. [Online]. Available: <https://emmanual.robotis.com/docs/en/platform/turtlebot3/overview/>
- [13] Robotis Co. Ltd, "The OpenCR 1.0 e-Manual," Ago. 2024. [Online]. Available: <https://github.com/ROBOTIS-GIT/OpenCR>
- [14] ROS org, "Robot Operating System (ROS)" Ago. 2024. [Online]. Available: <http://www.ros.org>.

# Indoor autonomous navigation for quadcopter using monocular visual SLAM

Rogelio Enrique Jiménez Trejo  
Departamento de Control Automático  
Cinvestav  
rogelio.jimenez@cinvestav.mx  
CDMX, México

Juan Manuel Ibarra Zannatha  
Departamento de Control Automático  
Cinvestav  
jibarra@cinvestav.mx  
CDMX, México

**Abstract**—This article presents a method for autonomous navigation based on a monocular SLAM (Simultaneous Localization and Mapping) applied to a quadcopter in unknown indoor environments, to solve problems like the indoor building inspection and mapping after an earthquake. A solution for the orientation and position control is given. Here, an orientation control based on a RANSAC (RANdom SAMple Consensus) regression line orientation applied to the ORB-SLAM2 point cloud is proposed. ORB stands for Oriented FAST (Features from Accelerated Segment Test) and Rotated BRIEF (Binary Robust Independent Elementary Features). Also the centering (distance) control is proposed using the same data. Wall detection for the corridors is made by dense point clustering applying Axes Aligned Bounding Boxes (AABB) and also to detect different possible scenarios the corridors cloud have. Here we rectify the robot pose using an Extended Kalman Filter (EKF) due to the possible triangulation error.

**Index Terms**—drone, quadcopter, ORB-SLAM2, autonomous navigation, orientation control, RANSAC, EKF.

## I. INTRODUCTION

### A. Proposed Solution

From precision agriculture to rescue task, supervision, building inspection, mapping, etc., outdoor and indoor applications for multi rotor drones are quickly increasing. In other of safety operation, many of this applications need an efficient autonomous indoor navigation system.

Indoor autonomous navigation can be solved easily using reactive methods based on information from infrared or ultrasonic proximity sensors, even using cameras in a elemental way. But, in building inspection, the interest application in this work, where we also need information about the localisation of some features (victims, structural failures, blocked passages, etc.) it will better to use some sophisticated methods like video SLAM. Then, our challenge is to use a camera as the only sensor to navigate in a unknown indoor environment after a disaster, mapping this environment, localizing and identifying some features of interest, using a Parrot Bebop, the dron used in our experiments.

To cope with this challenge we propose to use ORB-SLAM2 [1] to map an indoor 3D environment and to perform the autonomous navigation in that environment. We propose the use of RANSAC [2] to obtain a geometrical model of

the environment after an earthquake, as well as the pose of the drone. Also, a cluster method is proposed for the wall identification and to model the possible obstacles produced by some kind of disaster.

### B. Paper description

In section II we present some papers related to MAV applications related to our problem; nevertheless, no one attaches this issue as we do so, nor with the method used here. In section III we present the mathematical model of the drone, the sensor we use to make control (camera), and the method of sensing the environment via point cloud (ORB-SLAM2). The structure developed and the used operative system is also presented.

In section IV we explain the control laws proposed for navigation using the position and orientation as the output, and the angular and linear velocity as control variable. The use of RANSAC for getting the pose and AABB clustering algorithm for getting the navigating decision is explained too. In section V we explain the conclusions and results. Also we propose some applications for this project.

## II. STATE OF THE ART

From the beginning of the integration of the mobile robotics with advanced perception systems at the end of the last century, the problem of mapping and exploration unknown environments has been treated using different approaches resulting in new research areas as Probabilistic Robotics and SLAM. [3]. There has been work in mapping and exploration for single wheeled robot systems, very useful when there are no obstacles in the environment. Nevertheless, the use of MAVs (Micro Aerial Vehicles) are more appropriate to supervise indoor environments after an accident (earthquake, fire, explosions, etc.).

S. Fusic and T. Sugumari [4] present a deep review of existing literature, focusing on approaches that rely on the use of different kind of sensors like RGB and RGB-D cameras, Lidar and IMI to navigate autonomous robots in unknown

environments. This paper provides an overview of the current state-of-the-art in perception-based navigation and discusses the challenges and limitations of this approach.

In their seminal paper Raúl Mur *et al* [1] propose the well known ORB-SLAM algorithm to map an out-door environment offline, and in other experiments, maps a in-door environment in-line with a hand-held camera. ORB-SLAM is the most popular SLAM system of the last decade. Sergio García *et al* [8] approach a SLAM for the same model of Micro Aerial Vehicle (MAV). Their monocular visual SLAM and Sensor fusion (using a IMU) estimate a drone’s position via Extended Kalman Filter (EKF) They got well results for the mapping and indoor navigation, but it depends of human control and needs extra sensors increasing the cost and the processing requirements.

Sobers *et al* [9] used a sonar sensor and an IR sensor to detect indoor environment and get the relative vehicle location for a mono rotor Micro Aerial Vehicle. Nevertheless, this project used a wall following method; that is, only can navigate autonomously to map one wall at time, has no more information.

Putra *et al* [10] propose to use IMU as the sensor for a non-visual MAVs navigation due to its tiny size. But using a MAV with a small and simple camera is possible to get much more information. Tiemann *et al* [11] propose an autonomous navigation for a quadcopter using Ultra Wide Band (UWB) localization. This method could be accurate but it cannot avoid obstacles and the navigation only can be applied in a known indoor environment with UWB nodes installed.

### III. MATERIALS

#### A. Drone

A Parrot Bebop drone is used to perform the experiments, whose kinematic model [4] is done by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ (\cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi)) - u \\ (\cos(\phi) \sin(\theta) \cos(\psi) - \sin(\phi) \sin(\psi)) - u \\ \hat{w} - w \\ \hat{v} + \phi \\ \hat{u} + \theta \\ \hat{\psi} \\ \hat{\psi} - \psi \end{bmatrix} \quad (1)$$

As we can see, this model is a nonlinear system. But what is important to us is the capability to read the rectilinear position across the axis  $y$  and the angular position across the  $z$  axis,  $\psi$ . And the control variables are the velocity in these two axis  $\dot{y}$  and  $\dot{\psi}$

#### B. Camera

In this project the only sensor we use is the fish eye camera of the Parrot Bebop drone, even if it has other sensors. The corresponding camera matrix  $M_C$  and distortion vector  $\vec{d}$  given by the manufacturer are given by [7] :

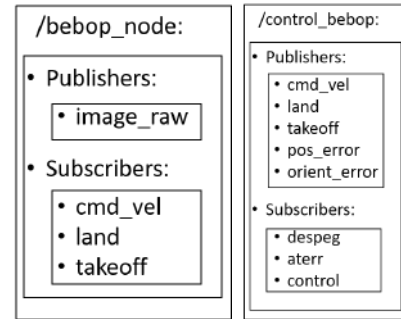
$$M_C = \begin{bmatrix} 396.17782 & 0 & 322.453185 \\ 0 & 399.798333 & 174.243174 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$\vec{d} = [-0.001983 \quad 0.015844 \quad -0.003171 \quad 0.001506 \quad 0] \quad (3)$$

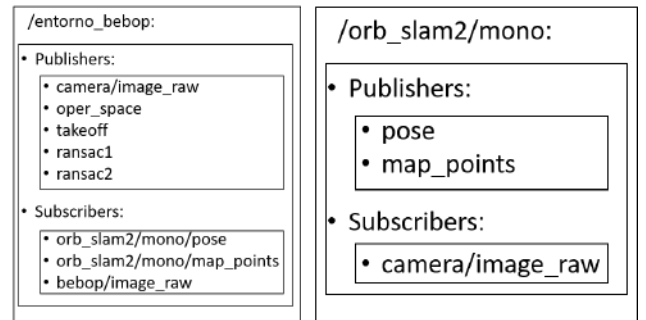
#### C. Robotics Operating System

We use the Melodic version of Robotics Operating System (ROS). For the Parrot Bebop and Parrot Bebop 2 there is the package *bebop\_autonomy* . The organization of the ROS nodes is like in the Fig.1. Also a ORB-SLAM2 ROS node is developed by AppliedAI [5].

ROS has some **nodes** who do some tasks and run methods; within these nodes, there are other elements known as send messages to a digital place called **topic**. Also the scripts for the nodes can be written in many programming languages (e.g. python, java, C++) so the language is not a hindrance. ROS use topics for upload information in message way.



(a) Node of the Bebop (b) Node for the package for the ROS connection computing process



(c) Node for the data analysis (d) Node for ORB-SLAM2 [1]

Fig. 1. Nodes used for the bebop’s data analysis and control within ROS

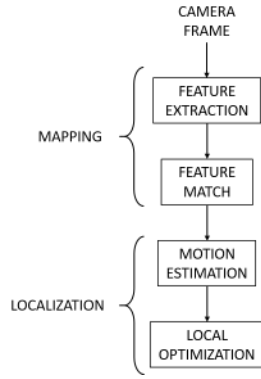


Fig. 2. Local process of the monocular ORB-SLAM2 system

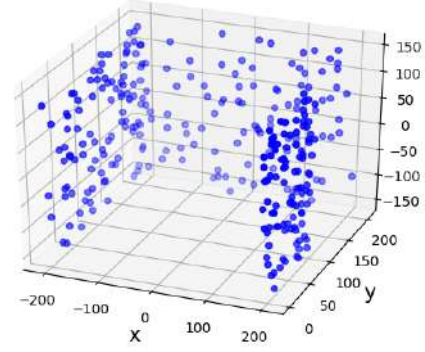


Fig. 3. Point cloud 3D map

#### D. ORB-SLAM2

We use ORB-SLAM 2 (Oriented FAST and Rotated BRIEF), one of the most popular videoSLAM systems, having monocular, stereo and RGB-D versions. ORB-SLAM2 has a few parameters to be tuned:

- **Number of features.** number of features extracted from the image.
- **Scale Factor.** Is the pyramid decimation ratio  $1 < s_f$ ; when  $s_f = 2$  it gives the classical pyramid
- **Number of the pyramid's levels.**  $n_l$
- **Initial FAST points.** Is the maximum initial threshold of FAST points needs the ORB-SLAM2 to start working.
- **Minimum FAST points.** Is the minimum threshold for the FAST points.

The chosen values for each parameter are showed in Table I. The SLAM system needs an image  $f_{(t-1)}$  (camera frame) which extract the features from, then needs a second frame  $f_t$  to extract its features and match them for a 2D localization. After that, with the Camera matrix  $M_C$ , maps the features in a space as 3D points and estimates the camera's motion (hence, robot's one) and make a local optimization for the Self Localization. This process can be watched easier in Fig. 2.

TABLE I  
 PROPOSED VALUES FOR THE ORB-SLAM2 PARAMETERS

nFeatures	scaleFactor	nLevels	iniThFAST	minFAST
2000	1.3	8	30	7

### IV. CONTROL AND NAVIGATION

#### A. Transformation

ORB-SLAM2 node publish two important messages: the pose and pointcloud. The drone pose is given by a vector  $\mathbf{P} = [x, y, z, \vec{q}]$  and the pointcloud by  $p_c = [x_i, y_i, z_i]$  with  $i = 0, 1, \dots, n$ ; its lenght will be increase every iteration. Is easier to apply the method making a transformation from world reference  $R_W$  to drone reference  $R_D$  it is made by a base change matrix taking  $p_c$  and the rotation matrix  $M_\theta$

around  $x$  axis,  $M_\phi$  around  $y$  axis and  $M_\psi$  around  $z$  axis. These matrix are defined as:

$$M_{r\theta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$M_{r\phi} = \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$M_{r\psi} = \begin{bmatrix} \cos \psi & \sin \psi & 0 & 0 \\ -\sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

and the vector for axial transformation, that is, the vector which change the origin of the referential  $\vec{C}_R$ . The vector is formed by:

$$\vec{C}_R = \begin{bmatrix} x_{new} \\ y_{new} \\ z_{new} \\ 1 \end{bmatrix} \quad (7)$$

multiplying all the matrix (4), (5) and (6) next Transformation matrix is obtained:

$$M_T = M_{r\theta} * M_{r\phi} * M_{r\psi} + [\mathbf{0}_{4 \times 3} \quad \vec{C}_R] \quad (8)$$

#### B. Orientation Control using RANSAC

To get the desired orientation, a regression of a dense point segment is proposed. A Random Sample Consensus (RANSAC) regression is used. Splitting the point cloud in two subspaces: Left subspaces  $S_1 = (x, y^+, z_{lim}^{lim})$ , and right subspace  $S_2 = (x, y^-, z_{lim}^{lim})$ , where upper and lower points are suppressed because they are considered as noise. The application of a RANSAC regression permits to obtain two

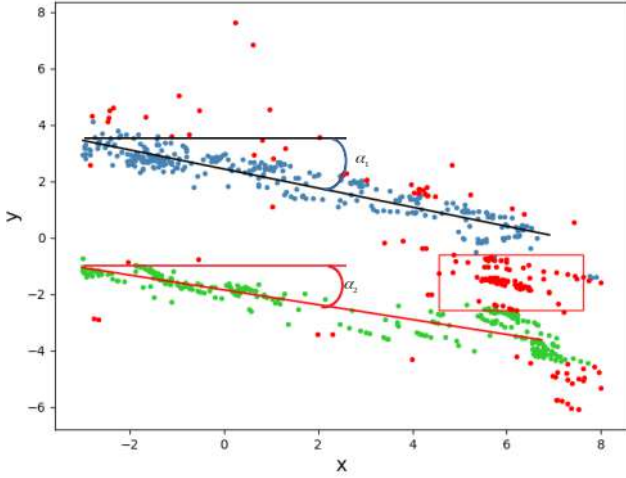


Fig. 4. Orientation measurement with RANSAC

point arrangements used to generate the left and right lines  $r_1$  and  $r_2$ , taking two points of each array  $r_1(x_1, y_1)$  y  $r_1(x_2, y_2)$  to get their slopes  $m_1$  y  $m_2$ :

$$m_1 = \frac{y_{r_{1,1}} - y_{r_{1,2}}}{x_{r_{1,1}} - x_{r_{1,2}}}$$

$$m_2 = \frac{y_{r_{2,1}} - y_{r_{2,2}}}{x_{r_{2,1}} - x_{r_{2,2}}}$$

and is possible to get its angles  $\alpha_1$  y  $\alpha_2$  as shown in Figure 4:

$$\alpha_1 = \arctan(m_1)$$

$$\alpha_2 = \arctan(m_2)$$

Now, the desire angle  $\psi_d$  for the drone trajectory is defined as:

$$\psi_d = -\frac{\arctan(m_1) - \arctan(m_2)}{2} + \arctan(m_2) \quad (9)$$

and then the control law proposed to cope with the orientation error  $e_o = \psi_d - \psi$  is:

$$K_o = b \tan(ae_o) \quad (10)$$

where the coefficients values are chosen as  $a = 1.959$  and  $b = 0.032$ , and the control  $K_o$  is bounded within the range  $[-1, 1]$  radians.

### C. Lateral Position Control using RANSAC

The line equations  $r_1$  and  $r_2$  are given as

$$r_1 = m_1x + b_1 \quad (11)$$

$$r_2 = m_2x + b_2 \quad (12)$$

where  $b_1$  and  $b_2$  are the distance from the drone to the left and right walls respectively. The drone must navigate at the

center of the corridor. Then desired position  $y_d$  is calculated by:

$$y_d = \frac{b_1 + b_2}{2} \quad (13)$$

Finally, the control law to cope with the lateral position error  $e_p = y_d$  is:

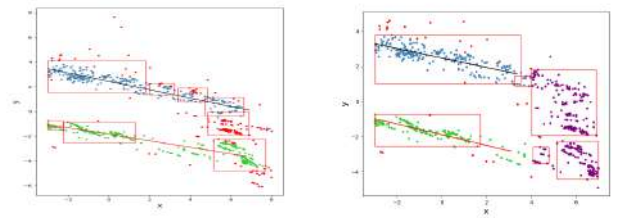
$$K_p = b_p \tan(a_p e_p) \quad (14)$$

where the coefficients take the same values used in eq. 10

### D. Longitudinal Control

To control the forward displacement and the possible changes of direction is needed to identify the structure of the corridors. This is done analysing the 3D point cloud to determine the end of a corridor or the change of its direction. To do that, we proposed a clustering process using Axis Aligned Bounding Boxes (AABB), with parameters bounding factor and minimum number of point to cluster, as showed in Fig. 5(b). To obtain better cluster we use a bounding factor of 0.3 and the same minimum number of point to cluster of 30, proposed in the previous case, as showed in Fig. 5(a). In this figure the non clustered points are considered as noise, and it is easy to identify the end of the corridor and its continuation to the right hand. In Fig. 6 a new scenario is showed where the end of corridor is not detected yet.

When a change of direction of a corridor is detected, the coordinate of the corresponding AABB are used to determine the coordinates of the point to stop the displacement and the turn angle to continue the navigation.



(a) Clustering for wall detection (b) Cluster for wall detection changing the bounding factor

Fig. 5. Detection of the walls in a corridor with turn to right

## V. RESULTS

### A. Conclusions

The 3D map obtained by ORB-SLAM2 was very useful for the dron navigation in a unknown environment, resulting in a trajectory by the median of the corridor, with a well identified turning points. The wall identification was made successfully despite the outliers due the noise. The pose of the drone and the 3D map have been rectified with a EKF. The obtained map can be used to generate a path planning to navigate in a environment previously visited and mapped.

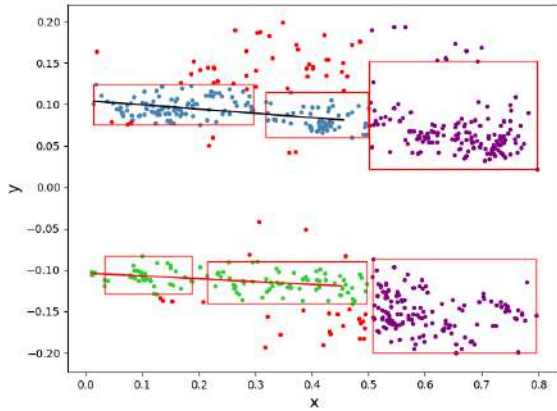


Fig. 6. Clustering for the corridor ahead detection

We observed some scale error during the mapping after turning, but the implemented control and the selected references could avoid this error. The scale error can be rectified with a more accurate matrix calibration and setting the ORB-SLAM2 features.

The indoor navigation in unknown environments was successfully completed, and the drone could detect walls and scenarios with the cluster proposed method. We recommend to keep the AABB clustering to avoid robot collision. Also the orientation and corridor centering method and control with the scenarios, and wall detection may be applied to other mobile robots types, as wheeled robots, and humanoids.

### B. Future Work

- **3D mapping.** Once the scale error is rectified, the environment could be successfully mapped.
- **Obstacle avoiding.** In the future work, we can use the clustering to detect vertical obstacles and to control the needed altitude to avoid them.
- **Navigation.** This system can be used, with some minor changes, to navigate in another class of indoor environments as halls and offices.

### C. Application

This project can be applied in some risk activities which people is doing yet, as review of damaged buildings for many reasons (e.g. fire, earthquakes, explosive warning). Also this method can be applied in land mobile robots for agro-industry in some tasks whose must be more accuracy. Using just one sensor for the indoor autonomous navigation is possible to decrease the production costs in individual or industrial production, making the technology application more available to more people.

## REFERENCES

- [1] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.
- [2] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography", Communication Association and Computing Machine, 24(6), pp.381-395, 1981.
- [3] S. Thrun. "Exploration and Model Building in Mobile Robot Domains." In Proc. IEEE Intl. Conf. on Neural Networks, pp. 175-180, 1993.], [S. Thrun, D. Fox and W. Burgard. "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots." Machine Learning, 31, pp. 29-53, 1998.
- [4] Paz, Claudio and Infante, Gabriel and Carballo, Jeremías and Báez, Federico and Cavenio, Cristian. 2014. Implementación de un Filtro Extendido de Kalman para la Estimación de la Orientación de un UAV utilizando el estándar CMSIS.
- [5] Applied AI [https://github.com/appliedAI-Initiative/orb\\_slam\\_2\\_ros](https://github.com/appliedAI-Initiative/orb_slam_2_ros).
- [6] C K Ashis, K. Sharma, "", unpublished.
- [7] "<https://www.manualpdf.es/parrot/bebop-drone/especificaciones>"
- [8] S. García, M. E. López, R. Barea, L. M. Bergasa, A. Gómez and E. J. Molinos, "Indoor SLAM for Micro Aerial Vehicles Control Using Monocular Camera and Sensor Fusion," 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC), Bragança, Portugal, 2016, pp. 205-210, doi: 10.1109/ICARSC.2016.46.
- [9] Sobers, D. & Chowdhary, Girish & Johnson, Eric. (2009). Indoor Navigation for Unmanned Aerial Vehicles. 10.2514/6.2009-5658.
- [10] J. Putra and D. E. Saputra, "Autonomous Drone Indoor Navigation Based on Virtual 3D Map Reference," 2022 International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, 2022, pp. 01-05, doi: 10.1109/ICISS55894.2022.9915133.
- [11] J. Tiemann, A. Ramsey and C. Wietfeld, "Enhanced UAV Indoor Navigation through SLAM-Augmented UWB Localization," 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, USA, 2018, pp. 1-6, doi: 10.1109/ICCW.2018.8403539.

# Commissioning of the NVIDIA Jetson AGX Orin module for a traffic signals detector based on YOLO

Gibran López Burrola  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de la Laguna*  
Torreón, Coahuila, 27000  
m.glopezb@correo.itlalaguna.edu.mx

Sergio López Hernández  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de la Laguna*  
Torreón, Coahuila, 27000  
m.slopezh@correo.itlalaguna.edu.mx

Miguel Ángel Llama Leal  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de la Laguna*  
Torreón, Coahuila, 27000  
mllama@correo.itlalaguna.edu.mx

Juan Sifuentes Mijares  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de la Laguna*  
Torreón, Coahuila, 27000  
jsifuentesm@correo.itlalaguna.edu.mx

**Abstract**— This work presents the configuration and commissioning of an NVIDIA Jetson AGX Orin module in order to pair and use its CUDA cores with OpenCV libraries. The implementation of CUDA cores is tested with third-party Deep Learning models. In addition, a benchmark is presented on six different computers with and without GPUs, taking the average inference time of the YOLOv8n model as indicator. Finally, a real-time Traffic Signals Detector (TSD) is trained; it is based on a pre-trained YOLOv8m model and trained on the NVIDIA Jetson AGX Orin module using CUDA cores. Some detections and metrics of the presented TSD are shown to evaluate its performance.

**Index Terms**— NVIDIA Jetson AGX Orin, CUDA cores, YOLOv8, real-time object-detector, traffic signals detector.

## I. INTRODUCTION

Artificial Intelligence (AI) is constantly evolving. That has opened the possibility for new significantly important applications in various fields such as engineering, security, and industry. AI gives rise to sophisticated Computer Vision Systems (CVSs) due to the integration of advanced Deep Learning (DL) tools with classical Vision Systems (VSs); traditionally, VSs are commanded by rules proposed by a human skilled in the field. Nowadays, the use of DL algorithms that can learn and automatically adapt to different scenarios provides a VS with the ability to generalize a task without trained personnel assigning the rules as in classic VSs.

Convolutional Neural Networks (CNNs) are a popular DL tool commonly used in CVSs to develop Object Detectors (ODs) [1]; their popularity is due to their performance over different tasks like flaw detector[2], image classifier [3] and image segmentation [4]. CNNs can learn and find patterns from raw data [5]; besides this, they can generalize from models previously trained for a specific task; this particularity is well known as Transfer Learning (TL). It is an effective technique used when there is a limitation in the amount of

data; a pre-trained model can efficiently set new model's weights without a large amount of data [6]. This strategy accelerates the development of new models and considerably improves their performance. The main purpose of an OD is to endow Embedded Systems (ESs) with the ability to generate accurately their own criteria from visual data taken from the surrounding environment. An ES is a computer system based on a microprocessor designed to do specific tasks, often, in real-time [7]. Nowadays, there are some popular ODs in the state of the art, such as YOLO (You-Only-Look-Once) [8], SSD [9], RetinaNet [10], Faster R-CNN [11], or Mask R-CNN [12]. Some OD's direct applications in ESs are real-time object tracking [13], facial recognition [14], quality inspection in manufacturing [15], public security [16], medical image analysis [17], Autonomous Vehicles (AVs) [18], among others.

AVs are one of the most emblematic technological advances of the century due to their many applications: public and private transportation [19, 20]; logistic and warehouse supply [21]; agriculture and farming [22]; emergency services and public safety [23, 24]; military defense [25], besides others. AVs imply an autonomous navigation system, which means there is no direct human intervention during the mobilization of the vehicle.

Autonomous navigation systems commonly process the data in three main stages: The sensing and perception stage; the mapping and localization stage; and the making-decision and control stage [26]. The sensing and perception stage is where the vehicle acquires data about the surrounding environment via integrated sensors like cameras, lidars, radars, ultrasonic sensors, etc.; these sensors take data in real-time. CNNs can analyze the received data to classify and detect objects or identify traffic signals and road lines, to mention a few. The mapping and localization stage uses the data received from the sensors in order to create a virtual environment that maps the physical environment around the AV; simultaneously, the vehicle's location is estimated within the virtual environment. Finally, the making-decision and control stage selects specific

This work was supported by TecNM Projects (Tecnológico Nacional de México/ I. T. La Laguna), CONAHCYT México and RICCA TecNM Network.

decisions as a result of certain circumstances, considering the data from the sensor and the virtual environment; frequently, these decisions are chosen based on different techniques such as reinforcement learning or probabilistic methods, to mention a few. The control mechanism takes these decisions and transforms them into actions like the vehicle's steering, acceleration, and braking. All of this is to achieve a safe and smooth route. Each stage implies the execution of complex calculations to process all the received data. To reduce data latency and be able to process data in real-time, computations have to be computed locally in the AV [27]. Onboard data computing is profitable to the making-decision and control stage; decisions may be made in fractions of seconds without depending on external networks or cloud services. That implies security and reliability, especially when network connectivity is limited or unreliable [28]. A solution for this inconvenience is the implementation of powerful processing units onboard. Besides the compute capacity inherent to the processing unit, it is crucial to consider other factors like portability, compatibility, durability, and sizing.

In 2023, NVIDIA announced its latest and novel set of computing modules: the NVIDIA Jetson Orin modules in the nano, AGX, and industrial versions. These modules are designed to integrate ES and develop DL projects that require high parallel computational costs. Currently, due to its computational capabilities, price and dimensions, the Jetson AGX Orin module is the best option for being placed onboard laboratory mobile robots. It is an innovative product, in consequence, the literature related to this module and available to the public is limited; it addresses specific problems and lacks generality, which makes the implementation and integration of the module for DL projects challenging. The main contribution of this work is a general solution to the NVIDIA Jetson AGX Orin module implementation challenge, in addition, a Traffic Signals Detector (TSD) is trained using CUDA (Compute Unified Device Architecture) cores in order to test their performance which is compared with other computers.

This work is organized as follows. Section II details main specifications and the initial setup of the Jetson AGX Orin module; furthermore, it explains how to enable the module's available hardware to make the most of the resources of the integrated GPU. Section III introduces the YOLO model and how it works; besides this, a performance comparative of YOLOv8 model on different computers is presented. Section IV presents methodology and experimental results of a TSD based on YOLOv8 model. Finally, Section V presents the conclusions obtained and the expected future work of the research project.

## II. COMMISSIONING OF JETSON AGX ORIN

According to [29] Jetson AGX Orin module has 612 CPU cores, 2048 CUDA cores on GPU, 64 Tensor cores on GPU, and 2 NVDLA cores; it works up to 275 [TOPS] in a power range of 15 to 60 [W]; it size is  $100 \times 100 \times 71.64$  [mm]. One of the main reasons for selecting this module is the CUDA cores integrated into NVIDIA GPUs. CUDA cores

are specialized platforms, accommodated and optimized for parallel computing tasks. Parallel processing uses as many cores as possible to perform general-purpose mathematical calculations [30]. CUDA cores use thousands of lightweight threads in parallel to perform a specific task, unlike other parallel processors that cannot perform as many threads. Another motivation is the NVIDIA Tensor cores. They are processing units designed to accelerate DL tasks, particularly those involving  $4 \times 4$  matrix multiply-accumulate operations with (1) structure [31].

$$D_{4 \times 4} = A_{4 \times 4} \times B_{4 \times 4} + C_{4 \times 4} \quad (1)$$

Some of the most popular applications of CUDA and Tensor cores are autonomous navigation, image-video processing, real-time object detectors, DL algorithms, simulations of physical processes and solving traditional algebraic operations [32–34]. The potential of these cores benefits from using complementary libraries such as OpenCV. According to [35], OpenCV is an open source library that contains low-level functions for image processing, as well as high-level algorithms, such as object identification, facial recognition, classification of actions in videos, etc. Pairing the OpenCV library with CUDA cores becomes challenging when online documentation is limited. Algorithm 1 presents the methodology that solves this from the initial configuration to the run of the YOLOv8 model.

---

**Algorithm 1:** Pairing and use of CUDA cores with OpenCV library

---

- 1 Set initial settings;
  - 2 Install and verify the NVIDIA JetPack package;
  - 3 Install and verify the NVIDIA Vision Programming Interface;
  - 4 Install and verify Git package;
  - 5 Create and activate a virtual environment in Python $\geq 3.8$ ;
  - 6 Clone and access directory “ nano\_build\_opencv” obtained from:  
[https://github.com/mdegans/nano\\_build\\_opencv](https://github.com/mdegans/nano_build_opencv);
  - 7 Edit and save the script “requirements” with the current JetPack setting;
  - 8 Run the script “build\_opencv.sh”;
  - 9 Clone and access “yolov8” directory obtained from:  
<https://github.com/ultralytics/ultralytics>;
  - 10 Open the script “requirements” and delete “Pytorch” and “Torchvision” lines from it;
  - 11 Run yolov8 requirements;
  - 12 Test YOLO model on a video or picture.
- 

Step 1 refers to location, network settings, date and time. Step 2 provides current information about the module; for the used module, main information is shown in Table I. Step 3 installs libraries that guarantee the use of CUDA cores. Step 4 installs the tool to obtain third-party resources. Step 5 ensures compatibility between new installed packages. Steps 6 and

9 get third-party packages for OpenCV and YOLO settings, respectively. Step 7 edits the script with the current CUDA arch BIN (8.7 for Orin modules). Steps 8 and 11 install third-party packages for OpenCV and YOLO installations, respectively. Step 12 tests YOLO model.

The module operating system is the GNU/Linux distribution Ubuntu 20.04 LTS. It has 64 [GB] of RAM and 1 [TB] SSD is added in local memory in order to locate all the third-party repositories. The boot is set in this SSD with the NVIDIA SDK Manager Software.

TABLE I JetPack resources

JetPack	5.1.2
L4T	35.4.1
CUDA arch BIN	8.7
CUDA	11.4.315
TensorRT	8.5.2.2
VPI	2.3.9
OpenCV	4.5.4 with CUDA
Python	3.8.10

A. CUDA cores activation test

“Opencv\_zoo” is an OpenCV repository available on GitHub. This repository contains various DL models tuned for OpenCV with benchmarks on different platforms. It can be used to test models executed on CPU and GPU. Algorithm 2 is follow in order to get “Opencv\_zoo” repository.

**Algorithm 2:** Getting “Opencv\_zoo” models

- 1 Clone and access directory “ opencv\_zoo” obtained from:  
[https://github.com/opencv/opencv\\_zoo?tab=readme-ov-file](https://github.com/opencv/opencv_zoo?tab=readme-ov-file);
- 2 Access to “model” directory and try yoloX detector.

YoloX model [36] is run in Jetson AGX Orin in order to test CUDA cores activation. When the model uses CUDA cores, the frame rate is up to 20 Frames Per Seconds (FPS), while when using CPU, it is up to 6 FPS; that means CUDA cores activation is correctly made and actually improves the frame rate.

III. YOLO MODEL

YOLO algorithm is a real-time OD open source that uses one single-stage object detector CNN, which means that object detection is made in a single sage [37]. Unlike an image classifier that completely catalogs the content of the image, YOLO computes the objectness score of each object in an image, even if the rest of the objects belong to different categories. Inspection tasks [38], public security [39], and AVs [40] are some of the applications that have been given to the YOLO algorithm.

Roughly, when the algorithm is run on an image, it predicts a bounding box that encloses an object and associates it to an objectness score for a class; this is made splitting the input image into a squared grid of  $S \times S$  pixels, getting  $m$  frames

[41]. After this, each frame is passed through the CNN in order to extract a characteristic map. When a frame match with certain characteristics, the objectness score is assigned to the frame within a bounding box [42].

The main YOLO stages are visually shown in Figures 1 and 2, taking a dog’s picture as input. Subfigure 1a shows how a dog’s picture is divided in a  $S \times S$  grid. The input image is resized according to the size of the grid, typically  $640 \times 640$  pixels. Subfigure 1b shows the  $m$  frames that appear on the resized image. Subfigure 2a shows how the face of the dog is enclosed and cataloged in the class “Dog” with an objectness score of 93%; Subfigure 2b shows the parameters of the YOLO model, where  $(x_c, y_c)$  is the geometric center of the bounding box,  $B_h$  and  $B_w$  are the height and width of the bounding box, respectively; while  $I_h$  and  $I_w$  are the height and width of the image.

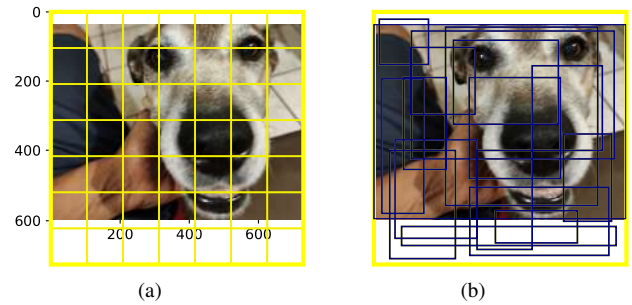


Fig. 1. YOLO stages: (a) Gridding and resizing (b)  $m$  frames

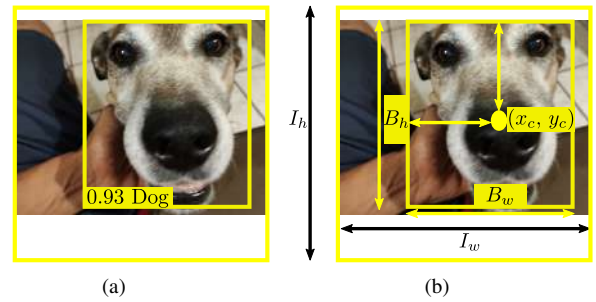


Fig. 2. YOLO stages: (a) Detection (b) Parameterization

At the beginning of 2023, the DL project development company, Ultralytics, launches the YOLOv8 model and its five different versions: n, s, m, l and x. YOLOv8’s architecture divides a CNN into two main parts: the backbone and the head. The backbone consisting of 53 convolutional layers and the head comprises multiple convolutional layers followed by fully connected layers responsible for predicting bounding boxes, classes and objectness scores [43].

A. YOLOv8n benchmark

The pre-trained YOLOv8n.pt model runs on a video, that can be found at <https://youtu.be/ATNCpGQCthc>, on the Jetson module and five other computers. Both, the specifications of

each computer and corresponding Average Inference Times (AITs) [sec] are displayed in the Table II; the corresponding RAM is shown in [GB]. Model performance is shown in best to worst order.

TABLE II YOLOv8n.pt performance benchmarking

Processor	RAM	GPU	AIT
Intel Core i7-10870H	32	RTX 3050Ti	0.0214
ARM Cortex-A78AE	64	NVIDIA Ampere	0.0377
AMD Ryzen 54500U	16	None	0.0711
Intel Core i5-7400	8	None	0.0891
Intel Core i7-4510U	8	None	0.3107
Intel Atom E3950	8	None	0.3804

The benchmark shows that the Jetson module has the second-best performance, however it is important to consider that the first place corresponds to a laptop that, due to its dimensions and weight, is difficult to attach to a laboratory mobile robot; while the dimensions and weight of the Jetson module make coupling to laboratory mobiles feasible.

#### IV. TRAINING A TRAFFIC SIGNALS DETECTOR MODEL

The pre-trained YOLOv8m.pt model is used to design a Traffic Signals Detector (TSD). The process is apportioned into five main stages like Algorithm 3 shows.

---

#### Algorithm 3: Development of the TSD

---

- 1 Data collection and labeling;
  - 2 Distribution of the data set in the training, validation and test subsets;
  - 3 Selection and configuration of the training environment;
  - 4 Training, validation and testing of the model;
  - 5 Fine tuning.
- 

##### A. TSD configuration

In step 1 the data set used is composed by a total of 836 images that are obtained from the platform Mapillary. These images are selected to match with the local traffic signals. Data labeling is made with the webpage Makesense. In the step 2, 24 classes are defined such as: No parking, Parking, Students, Stop, Traffic light, Heavy vehicle, among other traffic signal. The training set uses 711 images of the total; the validation set randomly takes 125 images from the training set; the test set consists of 125 images unknown to the model. For the step 3, the framework is Python using Ultralytics' library. Some important hyperparameters to consider when setting up the training framework are "patience", "batch size", and "image sizing". The "patience" is set to stop the training stage after 100 consecutive epochs without improvement in the validation performance. The batch size is set in order to process 16 images before performing a weight update. Finally, the image sizing is set so that all input images are resized to a width and height of 640 pixels. The optimization settings are left as the default in the library with a confidence threshold set in 0.7; this value determines which predictions are kept. For the step

4, the selected pre-trained model is YOLOv8m.pt; the training stage is composed by 200 epochs. The pre-trained model is selected by benchmarking various models like YOLOv8n and YOLOv8s with different training epochs: 15, 20, 55 and 100 epochs. The selected model with the presented settings has the best performance, it is obtained as a result of the step 5.

##### B. Experimental results

Figure 3 shows some of the TSD detections extracted from the testing step. Subfigures 3a, 3b, and 3c correctly detect their corresponding signals. Subfigure 3d incorrectly detects a "No Left Turn" signal when it should be "No Right Turn". Table III shows the metrics and losses at the last epoch in the training stage. The training and validation losses indicate how well the model is learning, while the metrics (precision, recall, and mean Average Precision (mAP)) show how well the model detects objects.

TABLE III TSD Metrics and losses at the end of training

Value	Description
0.21205	Classification loss during training
0.45029	Precision for bounding boxes
0.40448	Recall for bounding boxes
0.43508	Mean Average Precision (mAP) at 50% IoU
5.6585	Classification loss during validation

Figure 4 shows the Confusion Matrix Normalized (CMN) of the TSD. It is a visual tool that evaluates the model's performance by comparing the predicted classes with the true classes. It is desired that the values of the CMN be on the red boxes line. Formally, Confusion Matrix (CM) is typically a square matrix defined as

$$CM = \begin{bmatrix} T_p & F_p \\ F_n & T_n \end{bmatrix} \in \mathbb{R}^{C \times C}, \quad (2)$$

where  $T_p$ ,  $F_p$ ,  $F_n$ , and  $T_n$  are true positive, false positive, false negative, and true negative values for each class;  $C$  is the number of classes.  $T_p$  are correctly identified objects of a given class,  $F_p$  are incorrectly identified objects of a given class,  $F_n$  are fails to detect an object that is in the image, and  $T_n$  are successes to detect an object that is not in the image [44]. CMN is obtained by global normalization of (2) like

$$CMN_{ij} = \frac{CM_{ij}}{\sum_{ij} CM_{ij}}$$

Note how "SpeedLimit" is repeated, this is because the 10 to 80 Km/hr signals are labeled as "SpeedLimit" but independently detected during model training in order to improve TSD performance. Figure 5 shows the F1-confidence curve of the TSD. According to [45], the F1-score is a metric that evaluates the model's performance. It combines precision and recall, providing a balanced measure, such that

$$F1\text{-score} = \frac{2(\text{Precision})(\text{Recall})}{\text{Precision} + \text{Recall}}$$

The F1-confidence curve is a valuable tool for fine-tuning that plots the F1-score against different confidence thresholds. The peak of the curve is the optimal confidence threshold where



Fig. 3. TSD detections: (a) “Traffic\_lightSign” 89% and “Stop” 97% (b) “Heavy\_Vehicle” 97% (c) “NoParking” 93% and “Student” 95% (d) “NoL\_Turn” 94% and “NoParking” 93%

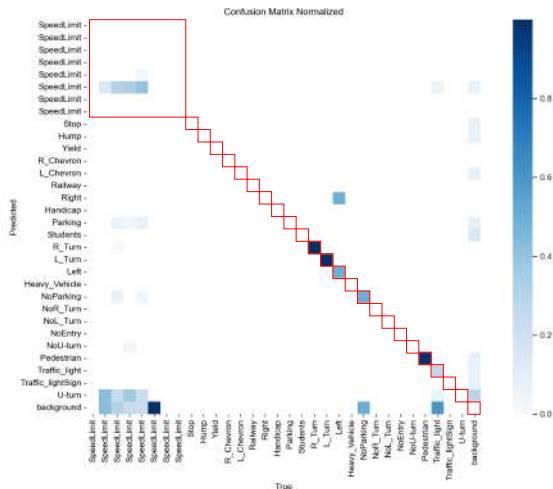


Fig. 4. TSD Confusion Matrix Normalized

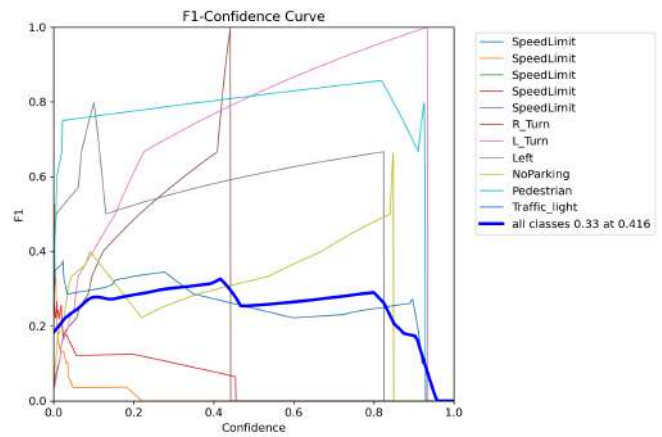


Fig. 5. F1-confidence curve

the F1-score is maximized. This value is often recommended as the operating point for the model.

## V. CONCLUSIONS

This work presents the methodology and initial setting of the NVIDIA Jetson AGX Orin module for CUDA cores activation. Also, the YOLOv8n model is benchmarking in various computers concluding that the Jetson module is the best option to implement in a laboratory vehicle due to

its weight, size and high computing capacity. A real-time traffic signals detector is also presented, using the YOLOv8m model and trained on the proposed Jetson module. The results obtained in the experimental stage show that various traffic signals are successfully detected. However, the model appears to be overfitted since training losses are low but validation losses are higher. According to the obtained F1-confidence curve, it is recommended to set a confidence threshold in the [0.33, 0.416] range. Another significant contribution of this work is the labeling and obtaining of the selected dataset used in the TSD's training. In addition to these products, the training code can be found in the repository [46].

#### A. Future work

As future work, it is expected to integrate the NVIDIA Jetson AGX Orin module into a laboratory vehicle. The onboard module will provide the vehicle with a high data processing capacity, that together with the integration of perception sensors and the development of trajectory generators, is expected to give it a sufficient level of autonomous navigation. The TSD's overfitting problem can be handled by considering different strategies, such as augmenting the dataset's size, using regularization techniques [47], tuning hyperparameters, or adding additional training epochs.

#### ORCID

Gibran López: <https://orcid.org/0009-0007-6130-7638>  
Sergio López: <https://orcid.org/0000-0002-1061-3762>  
Miguel Llama: <https://orcid.org/0000-0002-6280-0981>  
Juan Sifuentes: <https://orcid.org/0000-0001-6213-0001>

#### REFERENCES

- [1] S. Altaha and K. Riad, "Machine learning in malware analysis: Current trends and future directions." *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 1, 2024.
- [2] M. Słoński, K. Schabowicz, and E. Krawczyk, "Detection of flaws in concrete using ultrasonic tomography and convolutional neural networks," *Materials*, vol. 13, no. 7, p. 1557, 2020.
- [3] X. Dong, C. J. Taylor, and T. F. Cootes, "Defect classification and detection using a multitask deep one-class cnn," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1719–1730, 2021.
- [4] B. Kayalibay, G. Jensen, and P. van der Smagt, "Cnn-based segmentation of medical imaging data," *arXiv preprint arXiv:1701.03056*, 2017.
- [5] F. Alharbi, "Sampling strategies for tackling imbalanced data in human activity recognition," Ph.D. dissertation, Goldsmiths, University of London, 2021.
- [6] J. A. Campos-Leal, A. Yee-Rendon, I. F. Vega-Lopez, G. Beltran-Gutierrez, and J. R. Lopez-Arellano.
- [7] E. A. Lee and S. A. Seshia, *Introduction to embedded systems: A cyber-physical systems approach*. MIT press, 2016.
- [8] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.
- [9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.
- [10] M. Cheng, J. Bai, L. Li, Q. Chen, X. Zhou, H. Zhang, and P. Zhang, "Tiny-retinanet: a one-stage detector for real-time object detection," in *Eleventh international conference on graphics and image processing (ICGIP 2019)*, vol. 11373. SPIE, 2020, pp. 195–202.
- [11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.
- [12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [13] R. Lienhart, L. Liang, and A. Kuranov, "A detector tree of boosted classifiers for real-time object detection and tracking," in *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, vol. 2. IEEE, 2003, pp. II–277.
- [14] V. Ghenescu, R. E. Mihaescu, S.-V. Carata, M. T. Ghenescu, E. Barnoviciu, and M. Chindea, "Face detection and recognition based on general purpose dnn object detector," in *2018 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE, 2018, pp. 1–4.
- [15] X. Zhou, X. Xu, W. Liang, Z. Zeng, S. Shimizu, L. T. Yang, and Q. Jin, "Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1377–1386, 2021.
- [16] A. Wilkowski, M. Stefańczyk, and W. Kasprzak, "Training data extraction and object detection in surveillance scenario," *Sensors*, vol. 20, no. 9, p. 2689, 2020.
- [17] R. Yang and Y. Yu, "Artificial convolutional neural network in object detection and semantic segmentation for medical imaging analysis," *Frontiers in oncology*, vol. 11, p. 638182, 2021.
- [18] Q. Zhao, T. Sheng, Y. Wang, F. Ni, and L. Cai, "Cfenet: An accurate and efficient single-shot object detector for autonomous driving," *arXiv preprint arXiv:1806.09790*, 2018.
- [19] A. Siddiq and T. A. Taylor, "Ride-hailing platforms: Competition and autonomous vehicles," *Manufacturing & Service Operations Management*, vol. 24, no. 3, pp. 1511–1528, 2022.
- [20] S. Li, H. Tavafoghi, K. Poolla, and P. Varaiya, "Regulating tncs: Should uber and lyft set their own rules?" *Transportation Research Part B: Methodological*, vol. 129, pp. 193–225, 2019.
- [21] C. Dirican, "The impacts of robotics, artificial intelli-

- gence on business and economics,” *Procedia-Social and Behavioral Sciences*, vol. 195, pp. 564–573, 2015.
- [22] D. Vernon, “Robotics and artificial intelligence in africa [regional],” *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 131–135, 2019.
- [23] A. Chehri, G. Jeon, I. Fofana, A. Imran, and R. Saadane, “Accelerating power grid monitoring with flying robots and artificial intelligence,” *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 48–54, 2021.
- [24] S. Sarker, L. Jamal, S. F. Ahmed, and N. Irtisam, “Robotics and artificial intelligence in healthcare during covid-19 pandemic: A systematic review,” *Robotics and autonomous systems*, vol. 146, p. 103902, 2021.
- [25] B. Kempinski and C. Murphy, *Technical challenges of the US Army’s ground combat vehicle program*. Congressional Budget Office Washington, US-DC, 2012.
- [26] R. Spangenberg, D. Goehring, and R. Rojas, “Pole-based localization for autonomous vehicles in urban scenarios,” in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 2161–2166.
- [27] A. Apostu, F. Puican, G. Ularu, G. Suci, G. Todoran *et al.*, “Study on advantages and disadvantages of cloud computing—the advantages of telemetry applications in the cloud,” *Recent advances in applied computer science and digital services*, vol. 2103, 2013.
- [28] S. Goundar, *Edge Computing: Technology, Management and Integration*. BoD—Books on Demand, 2023.
- [29] L. S. Karumbunathan, “Nvidia jetson agx orin series,” 2022.
- [30] J. Lemeire, J. G. Cornelis, and E. Konstantinidis, “Analysis of the analytical performance models for gpus and extracting the underlying pipeline model,” *Journal of Parallel and Distributed Computing*, vol. 173, pp. 32–47, 2023.
- [31] C. A. Navarro, R. Carrasco, R. J. Barrientos, J. A. Riquelme, and R. Vega, “Gpu tensor cores for fast arithmetic reductions,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 72–84, 2020.
- [32] J. Zhou, *Scalable parallel programming for high performance seismic simulation on petascale heterogeneous supercomputers*. University of California, San Diego, 2014.
- [33] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja *et al.*, “Parallel convolutional processing using an integrated photonic tensor core,” *Nature*, vol. 589, pp. 7840, pp. 52–58, 2021.
- [34] D. B. Skillicorn and D. Talia, “Models and languages for parallel computation,” *Acm Computing Surveys (Csur)*, vol. 30, no. 2, pp. 123–169, 1998.
- [35] E. Cervera, “Gpu-accelerated vision for robots: Improving system throughput using opencv and cuda,” *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 151–158, 2020.
- [36] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “Yolox: Exceeding yolo series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.
- [37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [38] F. Mushtaq, K. Ramesh, S. Deshmukh, T. Ray, C. Parimi, P. Tandon, and P. K. Jha, “Nuts&bolts: Yolo-v5 and image processing based component identification system,” *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105665, 2023.
- [39] A. Kumar, A. Kalia, K. Verma, A. Sharma, and M. Kaushal, “Scaling up face masks detection with yolo on a novel dataset,” *Optik*, vol. 239, p. 166744, 2021.
- [40] P. S. Perumal, M. Sujasree, S. Chavhan, D. Gupta, V. Mukthineni, S. R. Shimgekar, A. Khanna, and G. Fortino, “An insight into crash avoidance and overtaking advice systems for autonomous vehicles: A review, challenges and solutions,” *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104406, Sep. 2021. [Online]. Available: <https://doi.org/10.1016/j.engappai.2021.104406>
- [41] S. López Hernández *et al.*, “Control de movimiento de robots usando técnicas de aprendizaje automático,” 2022.
- [42] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A review of yolo algorithm developments,” *Procedia computer science*, vol. 199, pp. 1066–1073, 2022.
- [43] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics yolov8,” 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [44] D. Fourure, M. U. Javaid, N. Posocco, and S. Tihon, “Anomaly detection: How to artificially increase your f1-score with a biased evaluation protocol,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 3–18.
- [45] H. Huang, H. Xu, X. Wang, and W. Silamu, “Maximum f1-score discriminative training criterion for automatic mispronunciation detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 787–797, 2015.
- [46] G. López Burrola, S. López Hernández, and M. Á. Llama Leal, “Traffic signals detector using yolov8,” <https://data.mendeley.com/datasets/g6mnc24f5b/3>, 2024, accessed via Mendeley Data 09, 17th, 2024. [Online]. Available: <https://data.mendeley.com/datasets/g6mnc24f5b/3>
- [47] R. Moradi, R. Berangi, and B. Minaei, “A survey of regularization strategies for deep models,” *Artificial Intelligence Review*, vol. 53, no. 6, pp. 3947–3986, 2020.

# CAPÍTULO 3



## DISEÑO DE ROBOTS

# Citlali: a wheeled mobile robot inspired by starfish

Juan C. Tejada  
Doctorado en Ciencias de la Ingeniería  
Universidad Iberoamericana  
Ciudad de Mexico, 01219, México  
Email: a2206166@correo.uia.mx

Ignacio Pietra-Santa  
Department of Engineering  
Studies for Innovation  
Universidad Iberoamericana  
Ciudad de México, 01219, México  
Email: a2169394@correo.uia.mx

Dariel Cruces-Delmar  
Department of Engineering  
Studies for Innovation  
Universidad Iberoamericana  
Ciudad de México, 01219, México  
Email: a2169394@correo.uia.mx

E.G. Hernandez-Martinez  
InIAT Institute of Applied Research and Technology  
Universidad Iberoamericana  
Ciudad de México, 01219, México  
Email: eduardo.gamaliel@ibero.mx

Alexandro López-González  
Department of Engineering Studies for Innovation  
Universidad Iberoamericana  
Ciudad de México, 01219, México  
Email: alexandro.lopez@ibero.mx

**Abstract**—Our planet harbors immense biodiversity, including Asteroidean invertebrates, commonly known as sea stars. These organisms, with their water vascular system and ambulacral feet, offer intriguing inspiration for robotic design. This paper presents the development of a bio-inspired robotic system modeled after the morphology and locomotion of sea stars. The proposed robot, named Citlali (from the Nahuatl word for "star"), features five arms equipped with ambulacral wheels, allowing omnidirectional movement. The design and manufacturing processes of the robotic platform are detailed, including the kinematic model and control system. The control system uses an artificial potential field approach to enable the robot to follow desired trajectories. Experimental results demonstrate the robot's capability to execute circular trajectories with varying control gains. Citlali exemplifies the potential of bio-inspired designs in advancing mobile robotics as an alternative to conventional omnidirectional systems.

## I. INTRODUCTION

Asteroidean invertebrates, commonly known as starfish, are particularly remarkable and exhibit a diverse range of species found in various marine environments. Belonging to the class Asteroidea within the phylum Echinodermata, starfish are characterized by their pentaradial symmetry, with a flattened body composed of a central disc and five or more arms. There are approximately 1,900 extant species of starfish distributed across the world's oceans, such as the Atlantic, Pacific, Indian, Arctic, and Antarctic Oceans. They inhabit a wide range of depths, from the intertidal zone to abyssal depths exceeding 6000 m [1], [2].

Starfish move using a water vascular system. Environmental water enters through the madreporite, circulates from the stone canal to the ring and radial canals, and finally reaches the ampullae in the ambulacral feet. Each foot, composed of an internal ampulla and an external podium, expands and contracts to adhere to and release from the substrate. A diagram of the water vascular system is shown in Fig. 1. This unique mode of locomotion, which involves wave-like movements of the ambulacral feet, enables starfish to navigate various terrains,

including vertical surfaces [3]. Most starfish move slowly, such as *Dermasterias imbricata* at about  $15 \text{ cm min}^{-1}$ , while some, like *Luidia foliolata*, can move faster at speeds of  $2.8 \text{ m min}^{-1}$  [4], [5].

The unique locomotion of starfish presents a fascinating challenge and inspiration for robotics. For instance, a modular robot driven by pneumatic artificial muscles mimics starfish tube feet with multiple suction cups [7]. Other designs include a low-cost soft quadrupedal robot utilizing tendon-based mechanisms [8], a starfish-like soft robot demonstrating bending and stretching control [9], and a system of magnetically actuated microtubes for omnidirectional movement [10]. These bio-inspired robots showcase remarkable capabilities such as crawling on flat ground, navigating viscous terrains, and adapting to various environmental challenges [11]. The development of such robots holds significant potential for applications in complex and harsh environments [12].

This work presents the development of a bio-inspired robotic system modeled after the morphology and locomotion of starfish. The proposed robot, Citlali, features five arms equipped with ambulacral wheels, allowing omnidirectional movement. The robot is controlled by an artificial potential field. This paper is structured as follows: Section II introduces the bio-inspired design of the robot, including the ambulacral wheels and the proposed kinematic model; Section III outlines the detailed design and manufacturing process, along with the proposed control scheme; Section IV presents the performance of the robot when executing a series of circular trajectories; and finally, Section V discusses the conclusions of this work and outlines proposed future research directions.

## II. BIO-INSPIRED TECHNICAL APPROACH

In this work, the morphology of Asteroidea serves as inspiration for establishing a novel design for mobile robots. The objective is to develop a robotic platform capable of omnidirectional movement on a plane. This development focuses on two bio-inspired parameters: the first is the morphological

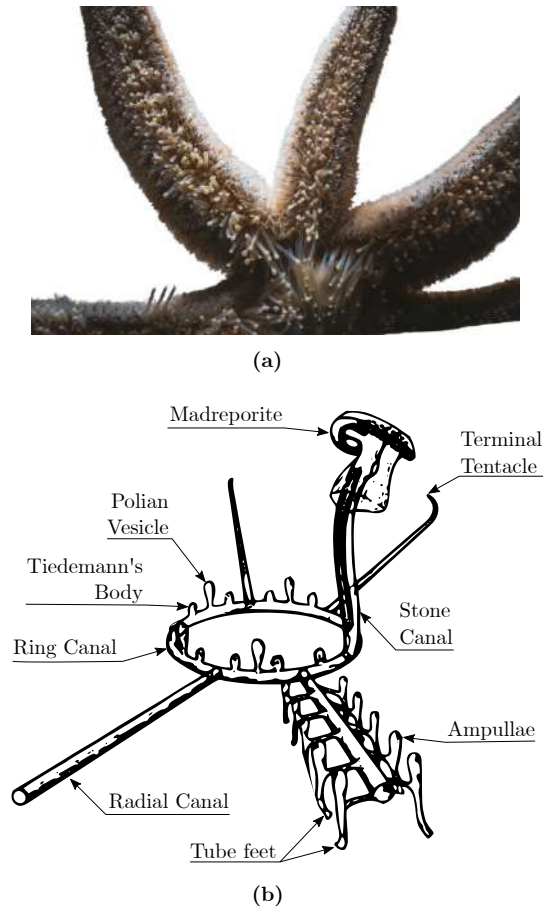


Figure 1: (a) Detail of tube feet of an Asteroidea. (b) Diagrammatic representation of the asteroid water-vascular system. Adapted from [6].

aspect, specifically the number of arms and actuators, which enable combined movements in the Cartesian plane. The second parameter is inspired by the ambulacral feet of starfish, which facilitate horizontal movement.

*A. Morphological Design*

The proposed architecture is based on the shape of a starfish. It consists of five arms connected to a central core, with each arm functioning as an actuator that facilitates system movement.

Fig. 2 illustrates the preliminary design proposal at the architectural level for the robotic platform.

*B. Proposed Actuator with Ambulacral Wheels*

The proposed actuator aims to replicate the function of ambulacral feet, which enable horizontal movement through friction with the surface. This design modifies the surface of a wheel to incorporate the morphology of these feet, transforming it into an ambulacral wheel. The design ensures that each foot has a small, tangential contact area, similar to a mecanum wheel, reducing friction. Each arm will be equipped with three ambulacral wheels, enhancing the friction required

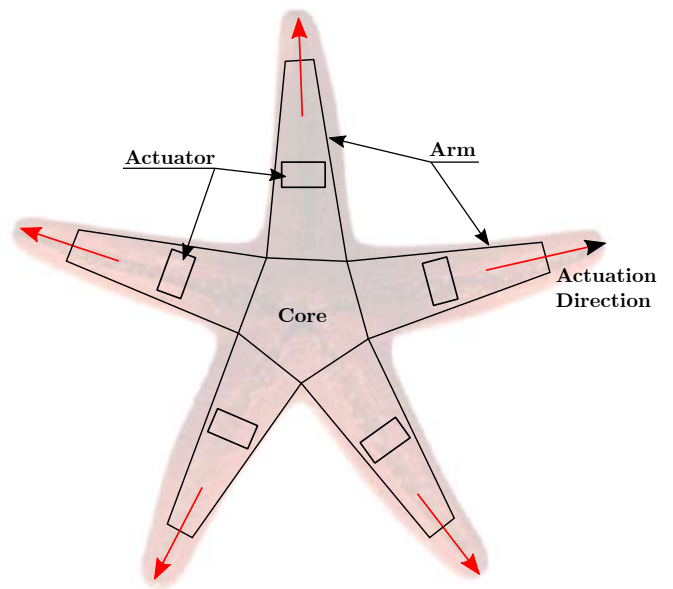


Figure 2: Morphological structure for the robot.

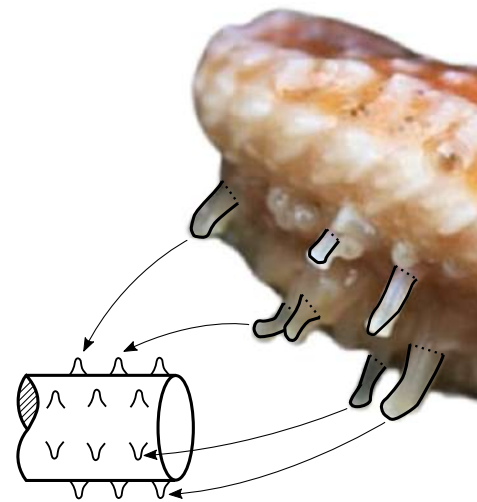


Figure 3: Proposal for ambulacral wheel.

to achieve movement in the desired direction, as shown in Fig. 2. The proposed design and arrangement of the ambulacral feet on the wheel are depicted in Fig. 3.

*C. Citlali Robot Conceptual Design*

Based on the elements described in the previous sections, the robotic system will comprise a total of five arms, each equipped with three ambulacral wheels. The proposed hardware architecture for the robot is illustrated in Fig. 4.

The system comprises a robot, a computer responsible for processing trajectory control, and a Vicon camera system that measures the robot's position in the workspace and provides feedback for kinematic control. The robot consists of five arms,

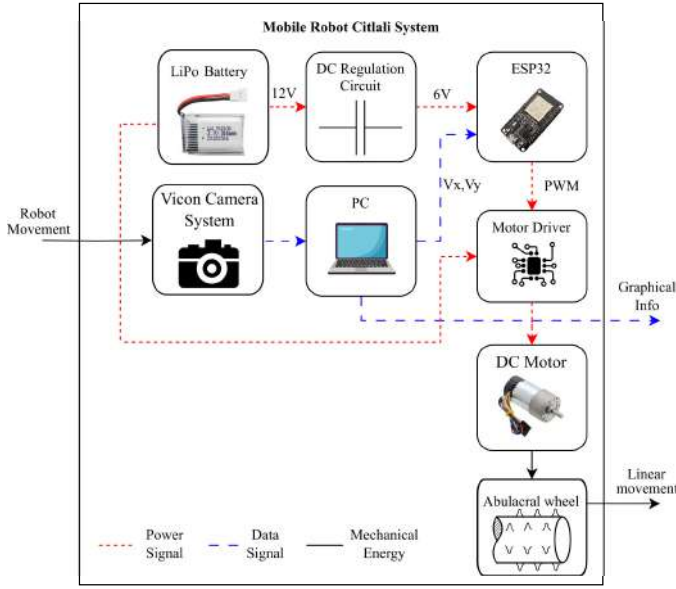


Figure 4: Hardware architecture for the robotic system.

Table I: Geometric parameters for the kinematic model.

Arm	$\alpha$ [ $^\circ$ ]	$r$ [m]	$l$ [m]	$\beta$ [ $^\circ$ ]
$A_1$	90	0.0175	0.153	90
$A_2$	162	0.0175	0.153	90
$A_3$	234	0.0175	0.153	90
$A_4$	306	0.0175	0.153	90
$A_5$	18	0.0175	0.153	90

each equipped with three ambulacral wheels. These wheels are driven by Pololu-4843 DC motors through a gear train, with each motor controlled by a Pololu Jrk G2 24v13 driver. An onboard ESP32 Devkit v1 receives speed commands from the central computer and converts them into PWM signals for each driver to control arm movement. The robot is powered by a 1300 mA h LiPo battery, with a voltage regulator circuit supplying voltage to the motors and the microcontroller.

#### D. Kinematic model

To derive the kinematic model, the robot was analyzed using the methodology proposed by Canudas de Wit et al. [13]. The bio-inspired multi-wheeled ambulacral system has been treated as having a single fixed wheel per arm, simplifying the kinematic analysis. Fig. 5 illustrates the kinematic diagram.

The kinematic model is divided into the posture model (referring to the global position of the robot in the absolute coordinate system) and the configuration model (referring to the description of the velocity of each wheel). According to Fig. 5, Table I provides the geometric parameters to calculate the models.

The robot moves in the  $XY$  plane, without the possibility to rotate around the  $z$ -axis. Therefore, the kinematic posture model is given by,

$$\dot{z}_G = R\Sigma\xi \quad (1)$$

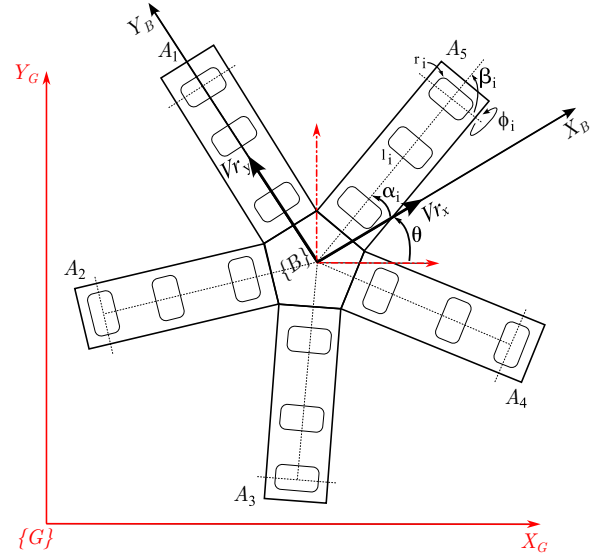


Figure 5: Citlali robot Kinematic diagram.

where,

$$\dot{z}_G = \begin{bmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{\theta}_G \end{bmatrix}$$

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\xi = \begin{bmatrix} V_{r_x} \\ V_{r_y} \end{bmatrix}$$

The kinematic configuration model is given by,

$$\dot{\phi}_i = E\Sigma\xi \quad (2)$$

Where,

$$E = J_2^{-1}J_1$$

The  $J_2 - J_1$  matrices result from the geometric and movement restriction analysis, given by:

$$J_1 = \begin{pmatrix} -\sin(\alpha_i + \beta_i) & \cos(\alpha_i + \beta_i) & l_i \cos(\beta_i) \end{pmatrix}$$

$$J_2 = \begin{bmatrix} r_1 & 0 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 & 0 \\ 0 & 0 & r_3 & 0 & 0 \\ 0 & 0 & 0 & r_4 & 0 \\ 0 & 0 & 0 & 0 & r_5 \end{bmatrix}$$

From Table I and the above equations, the kinematic models for the robotic system are given by:

*Kinematic posture model*

$$\begin{bmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{\theta}_G \end{bmatrix} = \begin{bmatrix} V_{r_x} \cos(\theta) + V_{r_y} \sin(\theta) \\ V_{r_y} \cos(\theta) - V_{r_x} \sin(\theta) \\ 0 \end{bmatrix} \quad (3)$$

*Kinematic configuration model*

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \\ \dot{\phi}_5 \end{bmatrix} = \begin{bmatrix} 57.14V_{r_y} \\ 17.66V_{r_x} - 54.34V_{r_y} \\ -33.59V_{r_x} - 46.23V_{r_y} \\ 33.59V_{r_x} - 46.23V_{r_y} \\ 54.34V_{r_x} + 17.66V_{r_y} \end{bmatrix} \quad (4)$$

III. DESIGN AND MANUFACTURING

A. *Prototype manufacturing*

After deriving the kinematic model and finalizing the conceptual design of the robot, we proceeded with the detailed design phase in preparation for manufacturing. The mechanical system’s design parameters were chosen with the manufacturing method in mind, specifically utilizing 3D printing with fused filament fabrication (FFF) using PLA as the filament material.

For the fabrication of the ambulacral wheels, a flexible material was selected to replicate the deformation behavior of ambulacral feet. Formlabs elastic 50A resin [14] was employed, using selective laser sintering for 3D printing. The design of the mechanical system is depicted in Fig. 6.

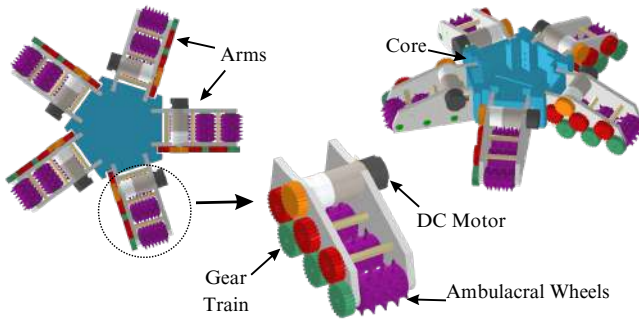


Figure 6: Citlali robot mechanical design.

The electronic system design involved the development and fabrication of two PCBs aimed at integrating the DC motor drivers, the ESP32 microcontroller, and voltage conditioning for the various components.

After completing the detailed design phase, the robot was manufactured, and the subsystem parts were integrated. The Citlali robot has a final weight of 1.2 kilograms. The fully assembled robot is depicted in Fig. 7.

B. *Control system*

An effective approach for trajectory planning involves the use of artificial potential fields. In this approach, the robot’s movement is influenced by a potential  $\mathcal{U}$ . The trajectory planning is generated incrementally based on the force exerted by the negative gradient of the potential  $-\nabla\mathcal{U}(P)$ , indicating

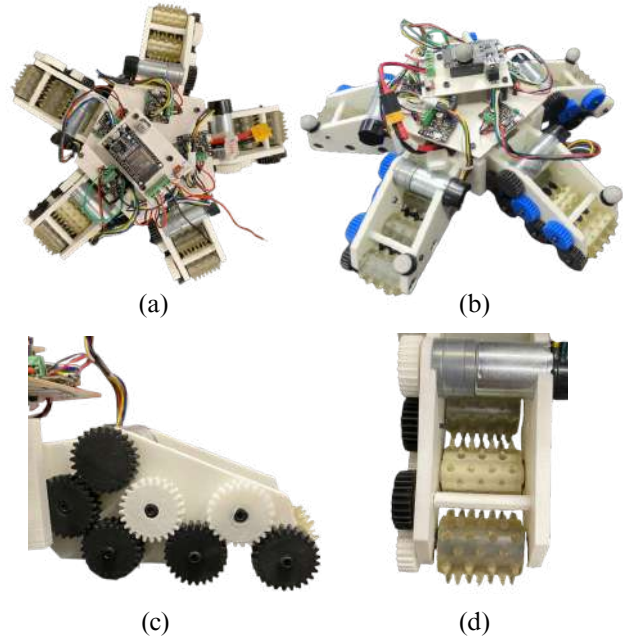


Figure 7: Different views of the Citlali robot. (a) Upper view. (b) Isometric view. (c) Lateral view of an arm. (d) Upper view of an arm and detail of the ambulacral wheels.

the most favorable direction of movement [15]. The resulting trajectory represents a path of least resistance, dynamically adjusted by the gradient of the combined potential field.

The attractive potential guiding the robot to the target point  $P_f$  can be modeled as a paraboloid:

$$\mathcal{U}(P) = \frac{1}{2}k_a\|e(P)\|^2 \quad (5)$$

where  $k_a > 0$  is a positive scaling factor and  $e = P_f - P$  is the error vector. This function is positive definite with a global minimum at  $P_f = 0$ , and the resulting force is:

$$f(P) = -\frac{d}{dx}\mathcal{U}(P) = k_a e(P) \quad (6)$$

The proposed control architecture for the robot is illustrated in Fig. 8. The global coordinates  $\{G\}$  are defined in the experimental setup, and the robot position  $\{B\}$  is obtained via the Vicon System.

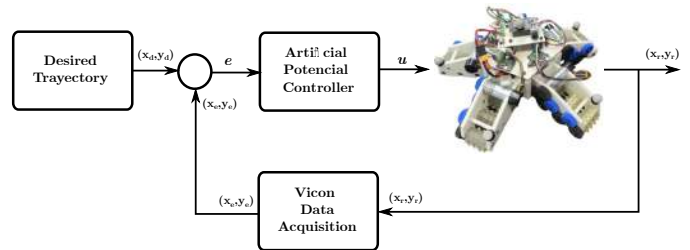


Figure 8: Control scheme.

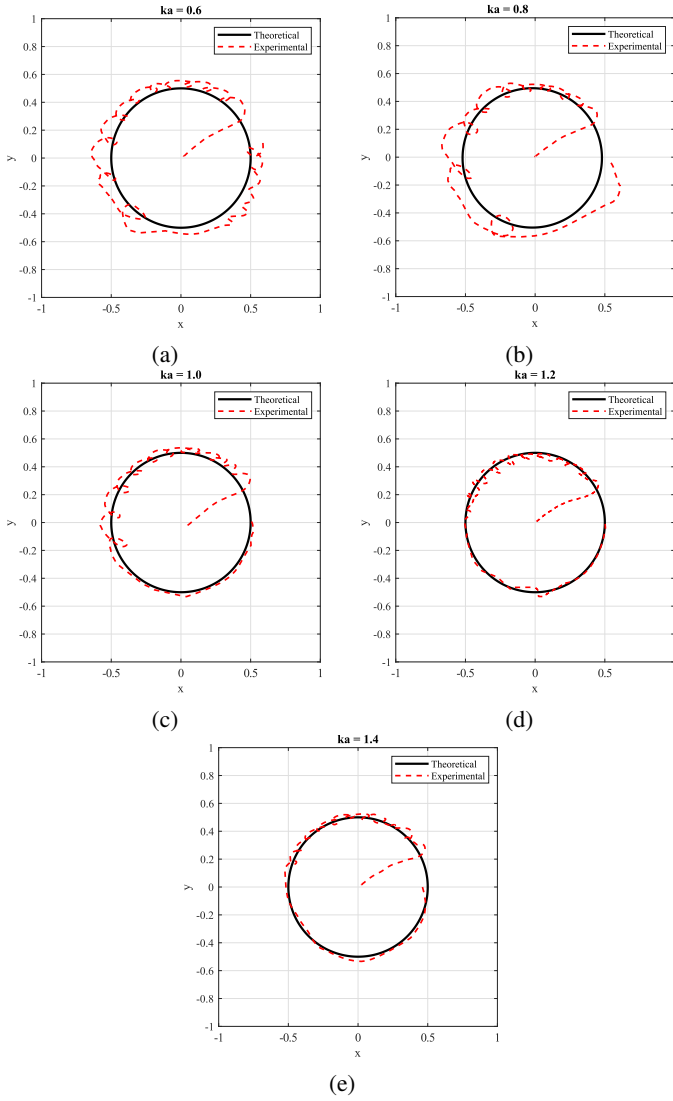


Figure 9: Trajectory results for different control gains.

#### IV. EXPERIMENTAL EXPLORATION AND RESULTS

Once the integration of the robotic system was completed, various tests were performed to verify the behavior of the proposed mobile robot. The control scheme described in the previous section was applied, and different control gains  $K_a = [0.6, 0.8, 1, 1.2, 1.4]^T$  were implemented to evaluate the performance.

To compare the robot's response to the controller, circular trajectories with a center at  $(0, 0)$  and a radius of 0.5 meter were executed.

The results of the obtained trajectories can be seen in Fig. 9.

To quantify the performance of each experiment, the mean squared error (MSE) between the theoretical and experimental positions for  $x$  and  $y$  was calculated. The results are presented in Table II.

All proposed trajectories were executed successfully, demonstrating the capability of the Citlali robot to follow

Table II: Mean squared error for the experiments.

$K_a$	$MSE_x$	$MSE_y$
0.6	0.0276	0.015
0.8	0.0740	0.0506
1	0.0169	0.0080
1.2	0.0125	0.0039
1.4	0.0135	0.0078

the desired paths under varying control parameters. The system's best performance was observed when control gains exceeded 1. In this range, the robot exhibited smooth and continuous motion, closely matching the theoretical trajectory with minimal deviations. This was particularly evident in circular trajectories, where the ambulacral wheels efficiently adjusted to changes in direction, ensuring precise omnidirectional movement.

Conversely, at lower gains (below 1), the robot encountered notable challenges. A significant issue was the presence of a dead zone in the control response, which severely limited the robot's ability to initiate smooth movement. This dead zone, caused by insufficient motor actuation, resulted in a delay between the command and the actual motion of the robot, leading to accumulated errors over time. As the robot struggled to overcome this delay, it produced intermittent movement spikes, where sudden bursts of motion attempted to compensate for the earlier inaction. These movement spikes caused the robot to deviate from its intended trajectory, particularly in sharp turns or when changing direction abruptly.

The increase in control gains not only improved the responsiveness of the system but also reduced the occurrence of these movement spikes. With higher gains, the control system was able to apply more aggressive corrective actions, allowing the robot to follow the planned trajectory more closely. The resulting movements were more fluid, with fewer deviations from the desired path, particularly in the circular trajectory experiments. However, it was observed that very high gains introduced slight oscillations in the robot's motion, indicating that an optimal gain range exists where the balance between responsiveness and stability is maintained.

Overall, these results suggest that tuning the control gains is critical for achieving optimal trajectory tracking. Gains below 1 lead to performance issues due to delayed control responses, while gains significantly above 1 may introduce new challenges, such as oscillatory behavior. The optimal range for this system appears to be around 1.2, where both trajectory accuracy and system stability are maximized.

#### V. DISCUSSION AND CONCLUSION

A bioinspired robotic system based on the morphology and locomotion principles of Asteroidea was successfully developed, achieving the primary objective of creating an omnidirectional platform. The ambulacral wheels, modeled after the tube feet of starfish, served effectively as traction and actuation elements, enabling smooth and multidirectional

movement with reduced friction. This design allowed the robot to meet the movement requirements of the bioinspired concept, providing a unique solution compared to traditional omnidirectional robots.

The integrated system demonstrated the ability to follow desired trajectories, with experiments showing high precision when using a control scheme based on attractive potential fields. For circular trajectories, mean squared error (MSE) values of 0.0125 and 0.0039 for the  $x$  and  $y$  coordinates, respectively, were obtained with a control gain of 1.2. These low error rates highlight the system's capability to accurately track trajectories when control parameters are properly tuned.

However, the system faced challenges related to nonlinearities caused by friction in the non-actuated arms, especially at lower control gains. This friction, arising from the contact between the ambulatory wheels and the surface, introduced disturbances that affected the robot's trajectory. Additionally, the initial kinematic model, which did not account for orientation control, was found to be insufficient. The lack of orientation management led to occasional drift and instability, particularly in complex maneuvers.

To address these issues, future research will focus on developing a more robust control strategy, possibly incorporating a nonlinear controller or observer to handle the nonlinear effects of friction and improve overall trajectory stability. A hardware update and a revised kinematic model incorporating orientation control will be explored to enhance the system's stability and accuracy in motion. Further experimentation with varied trajectory shapes and environments will be conducted to assess the robot's adaptability and refine the control system for improved performance.

#### ACKNOWLEDGMENT

The authors would like to thank Universidad Iberoamericana Ciudad de México and CONAHCYT for grant number 1228748.

#### REFERENCES

- [1] J. Pechenik, *Biology of the Invertebrates*. feb: McGraw-Hill Education, 2014.
- [2] R. C. Brusca, *Invertebrates*. jan: Sinauer Associates is an imprint of Oxford University Press, 2016.
- [3] J. M. L. (editor), *Starfish: Biology and Ecology of the Asteroidea*. Johns Hopkins University Press, 2013. [Online]. Available: [libgen.li/file.php?md5=54672a16d530983fce8eefdf517948b6](http://libgen.li/file.php?md5=54672a16d530983fce8eefdf517948b6)
- [4] M. Cavey and R. Wood, "Specializations for excitation-contraction coupling in the podial retractor cells of the starfish *stylasterias forreri*," *Cell and Tissue Research*, vol. 218, no. 3, Jul. 1981. [Online]. Available: <http://dx.doi.org/10.1007/BF00210108>
- [5] G. A. Kerkut, "The forces exerted by the tube feet of the starfish during locomotion," *Journal of Experimental Biology*, vol. 30, no. 4, p. 575–583, Dec. 1953. [Online]. Available: <http://dx.doi.org/10.1242/jeb.30.4.575>
- [6] E. Hennebert, M. Jangoux, and P. Flammang, *Starfish: Biology and Ecology of the Asteroidea*. Johns Hopkins University Press, 2013, ch. Functional biology of asteroid tube feet, pp. 24–36.
- [7] A. Poungrat and T. Maneewarn, "A starfish inspired robot with multi-directional tube feet locomotion," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2017, pp. 712–717.

- [8] M. Munadi, M. Ariyanto, J. D. Setiawan, and M. F. A. Ayubi, "Development of a low-cost quadrupedal starfish soft robot," in *2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 2018, pp. 225–229.
- [9] J. Zou, M. Yang, and G. Jin, "A five-way directional soft valve with a case study: a starfish like soft robot," in *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*, 2020, pp. 130–134.
- [10] X. Yang, R. Tan, H. Lu, and Y. Shen, "Starfish inspired milli soft robot with omnidirectional adaptive locomotion ability," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3325–3332, 2021.
- [11] S. Mao, E. Dong, M. Xu, H. Jin, F. Li, and J. Yang, "Design and development of starfish-like robot: Soft bionic platform with multi-motion using sma actuators," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013, pp. 91–96.
- [12] S. Mao, E. Dong, S. Zhang, M. Xu, and J. Yang, "A new soft bionic starfish robot with multi-gaits," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2013, pp. 1312–1317.
- [13] C. C. de Wit, B. Siciliano, and G. Bastin, *Theory of Robot Control*. Springer London, 1996.
- [14] "Familia de resinas: Flexible y Elastic — formlabs.com," <https://formlabs.com/latam/materials/flexible-elastic/>, [Accessed 25-06-2024].
- [15] E. G. Hernandez-Martinez and E. Aranda-Bricaire, "Convergence and collision avoidance in formation control: A survey of the artificial potential functions approach," in *Multi-Agent Systems*, F. Alkhateeb, E. A. Maghayreh, and I. A. Doush, Eds. Rijeka: IntechOpen, 2011, ch. 6. [Online]. Available: <https://doi.org/10.5772/14142>

# Remotely controlled vehicle by facial EMG

Alexis Benítez Rosales  
*Facultad de Ciencias*  
*Departamento de Física, UNAM*  
Mexico City, Mexico  
alexisbntzr@ciencias.unam.mx

Daniel Armando Montiel De León  
*Facultad de Ciencias*  
*Departamento de Física, UNAM*  
Mexico City, Mexico  
danielmontiel\_leon@ciencias.unam.mx

Rey David Solis Galíndrez  
*Facultad de Ciencias*  
*Departamento de Física, UNAM*  
Mexico City, Mexico  
reyda@ciencias.unam.mx

Verónica Esther Arriola Ríos  
*Facultad de Ciencias*  
*Departamento de Matemáticas, UNAM*  
Mexico City, Mexico  
v.arriola@ciencias.unam.mx

Sergio Enrique Solís Nájera  
*Facultad de Ciencias*  
*Departamento de Física, UNAM*  
Mexico City, Mexico  
solisnajera@ciencias.unam.mx

**Abstract**—In this project, an electromyogram (EMG) system was developed with the aim of controlling a vehicle. The project involved the design and implementation of an EMG system capable of capturing muscle signals from the masseter muscle and converting them into commands. A communication system was designed between two development boards using HC-05 Arduino Bluetooth modules in a master-slave configuration: one board analyzed the EMG signals and sent commands to the other board, which executed control functions: moving forward, turning, and stopping the 4 motors installed on the vehicle. The aim of the present research work, was to control a vehicle via wireless communication from EMG device.

**Index Terms**—Open Hardware, remote-controlled vehicle, Electromyogram

## I. INTRODUCTION

Technological advancements and the integration of knowledge have driven the creation of projects aimed at addressing specific needs and solving emerging problems. An example of this is the use of electrophysiological monitoring devices, such as electromyographs (EMG) or electrooculographs (EOG), which have provided innovative solutions for clinical diagnosis, prosthetics controlled by muscle signals, brain-computer interfaces, wearable health monitoring devices, and even remote control of vehicles.

The relationship between these biopotentials, technology, and human development is evident, as they open the door to a new generation of devices that improve quality of life, promote rehabilitation for people with motor disabilities, and enhance physical performance through real-time data analysis and feedback.

## II. PROJECT INFORMATION

### A. Statement of the problem / Justification

Remotely controlled vehicles initially emerged for military purposes, particularly in aircraft used during war. By the 1950s, non-military land vehicles began to appear, and in subsequent decades, radio-controlled vehicles became popular in toys as well as in technological and medical devices.

In the medical field, examples include remotely controlled motorized wheelchairs and the Da Vinci robot, created in 2000 for surgeries. These technologies use remote control methods like WiFi, Bluetooth, radiofrequency, or infrared. In Mexico, this technology began to be applied in the second half of the 20th century, with a notable patent in 2009 for the wireless control of a "material-handling vehicle." [1]

Given that remote control technology has seen limited development in specific medical applications, particularly in vehicle control, this project develops an electromyograph (EMG) based control for a four wheeled vehicle. Each mechanism wheel is controlled by an independent motor. The EMG signal analysis allows to differentiate three movement commands for the platform: moving forward, turning, and stopping. The movement is the executed by an electronic board.

## III. BASICS

Biosignal controlled devices are electronics gadgets that allow the users to interact with electronic instruments. These systems convert biological signals to electronic signals that can be interpreted as an electronic activation for a specific device or tool. Electromyogram (EMG) is a technique to evaluate and record the electrical activity produced by skeletal muscles. The biosignals produced by the electrochemical activity of cells, together with the electrical signals of a muscle, are called biopotential [2]. EMG signals are useful in various biomedical applications, such as rehabilitation, prosthesis control, and in our case, to control the movement of a vehicle [3].

Although any part of the human body that has a muscle can be used, this work focuses on the detection of electromyography signals from the face, particularly those from the masseter muscle, that is responsible for chewing: it moves the jaw vertically (upwards) and participates to a lesser extent in other jaw movements.

For signal conditioning and control, Arduino development boards are a good option. Arduino is a free software and hardware development company. Using the C and C++ language it is possible to carry out different tasks [4, 5].

As an extension to the Arduino board, it is necessary to use drivers to control the motors. Drivers are devices used to control the power supply to high-power motors via commands from a microcontroller. They are the interpreter between the board and the motors. The driver we use is an HW-039 BTS7960, which can handle voltages up to 27 V and currents up to 43 A. The analyzed driver features 8 connections to interact with the microcontroller, from which 6 were used in the configuration.

The pin-out connections include GND and VCC for system power. Additionally, the right enable and left enable connections (R\_EN and L\_EN pins) were used to enable input and output signals, respectively, while the right and left pulse width modulation connections (RPWM and LPWM pins) are used to control both the direction and speed of the motor.

DC motors SY34-9E4 were used, which are high-power direct current motors. These motors have a nominal speed of 4000 RPM, a maximum PWM of 255, and operate at a nominal voltage of 24 V; however, they were powered with 10 V and programmed to a speed of 100 PWM for forward movement and 70 PWM for turning. They feature a starting torque ranging from 0.5 Nm to 1 Nm, and a continuous torque ranging from 0.2 Nm to 0.5 Nm.

Wi-Fi or Bluetooth could have been used for remote communication. Since the EMG reader and the car controller will keep constant communication, and we expect the human controller to sit on the car (small distance) we chose to use the latter option. Working with Bluetooth involved using a board with Bluetooth embedded like the ESP32 development board or the HC-05 module Bluetooth that can be integrated to any development board. As the project progressed, the ESP32 board had communication issues with the slave microcontroller, so we proceeded to use HC-05 modules, which has a reach length of up to 30m. This module is a component that provides wireless communication through a Bluetooth connection between two devices, in this case, between the Arduino board that will analyse the EMG signal to issue the command, and the one that will communicate with the drivers to control the wheels.

#### A. Goals

The principal objective of this work is to control a vehicle wirelessly using facial muscle movements. The particular objectives are:

- To design and implement an EMG circuit capable of capturing biopotentials from facial muscles.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If mixed units must be used, clearly state the units for each quantity to be used in the equation.
- To define and associate a biopotential (through specific movements that do not affect the subject) with a specific action, whether it be moving, stopping, or turning the vehicle.

- Perform a basic algorithm on an open source development board (Arduino) that detects the biopotential of the EMG and, via Bluetooth, transmits instructions to a slave microcontroller for the movement of the vehicle.
- Implement the control hardware and code for a robotic base that will act as the vehicle.

## IV. METHODOLOGY

Throughout the project, important elements were discarded and added depending on their effectiveness, however, the key materials were the Arduino UNO and Arduino MEGA development boards, BTS7960 drivers, electrodes and analog components to condition the biosignal (AD620, RC filters, envelope detector, among others). It could be said that the project was divided into three large phases: Obtaining the signal given by facial muscles; Conditioning the signal from the Arduino IDE software; and Movement of the vehicle from the conditioning.

### A. Electromyogram acquisition

For this first stage, the AD620 instrumentation amplifier was used with a gain of 67.7dB (56Ω resistance), as well as a 1μF capacitor to avoid the DC component; then a 30 Hz filtering stage with a 22 μF capacitor and 220Ω resistance followed by a non-inverting amplifier with two 120Ω resistors. Then we sought to make an envelope detector using a 1N5819 diode, 0.22 μF capacitor and appropriate resistance (1000Ω) for the time constant and period of the signal, as shown in Figure 1.

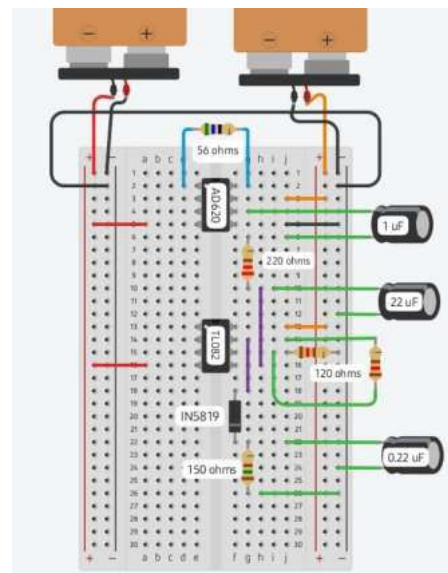


Fig. 1. EMG circuit at protoboard.

Three electrodes are used to obtain the signal as the muscle moves: two which must be placed on the same muscular fibre to detect electric potential, and a third one, called the reference, which must be placed on a different muscle where electrical activity must be almost negligible to play the role of ground [4].

It is important to mention that the muscle that was considered initially for signal acquisition was the frontalis muscle, which generates the potential when the eyebrows are raised. However, when working with this muscle, a lot of irregularities in its voltage arose as well as problems with placing the electrodes. Therefore we changed to acquiring the signal from the masseter muscle, which produced a greater potential in its contraction and eased the placing of electrodes.

Starting from here, a good location for the electrodes was sought by placing them on the masseter as shown in Figure 2. The placement of the reference was varied between the mastoid and the earlobe. It was observed that there were no changes in the signal as the placement of the reference varied. By closing the mouth and clenching the teeth with the jaw, a specific action potential is generated in the aforementioned muscle.

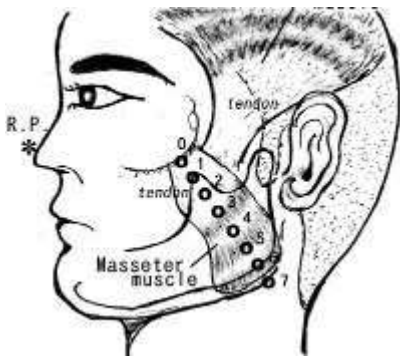


Fig. 2. Electrode placement. Taken from Extraction of the EPP component from the surface EMG [diagram], Toshifumi Kumai, 2009.

Initially, we worked with the direct output signal of the AD620 instrumentation amplifier and the capacitor, eliminating the DC component. It was at an advanced stage of the project that it was proposed to work with the signal after the envelope detector.

### B. Biosignal Conditioning (Programming)

Using Arduino development boards, analog components and components related to the omnidirectional robotic base, the analog signal is obtained, subsequently processed, conditioned and used to operate the actuator that moves the robotic base simulating a vehicle.

For the first part only the Arduino UNO is used. This stage was named “Master” while the second part in which the Arduino MEGA is used was named “Slave” since they fulfill these characteristics of giving and receiving orders.

a) *Master*: In this phase we worked with the Arduino UNO to obtain the biosignal. This stage required a lot of time and simulations, since it was necessary to see the appropriate gesture and threshold to choose the action to be performed by the actuator (the motors that move the vehicle). Subsequently, from an HC-05 bluetooth module the action was sent to the Arduino MEGA (also with a module for wireless communication).

Several conditions are set in the master code to choose specific gestures. When analyzing the biosignal, it was decided that the action threshold should exceed around 1.4 V (since from a basal state to contraction this voltage value was obtained as a difference limit). From this the following gestures/actions were obtained.

- Forward: Contract the muscle for more than 1.4 seconds.
- Turn: Contract the muscle more than 1 time in less than 1.4 seconds
- Stop: Contract the muscle once in less than 1.4 seconds.

During the reading of the signal, a reset of the gestures and thresholds obtained is carried out every 2.7 seconds in order to obtain another gesture again. In the same way, the action given before the restart mentioned above was maintained.

Finally, a number related to each action is sent through the Bluetooth module (“1” to advance, “2” to turn, “3” to stop). In Figure 3 you can see the Arduino UNO board connected to the HC-05 module and receiving the EMG signal.

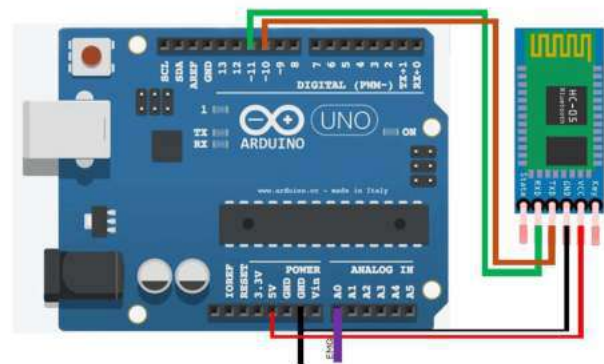


Fig. 3. Connection diagram for Arduino UNO.

b) *Slave*: During this phase the Arduino MEGA was used. From the HC-05 module, the numbers related to each action were received, and within the code the functions corresponding to each action were established. For example, in the case of stopping, all outputs were set to a low state, while for turning or moving forward, the correct activation of the direction of rotation was carried out for each motor individually as will be explained in more detail in section IV-C.

Being 255 pwm (pulse-width modulation) the maximum speed of the motors, 100 pwm was chosen for the forward action and 70 pwm for turning. With this, the vehicle would not move too fast (although the speed could also be changed in the slave code depending on different needs).

### C. Vehicle Movement (Actuator)

During the actuator stage, 4 BTS7960 drivers were used, the Arduino MEGA (using the pwm pins and the digital pins) and a 24 V lithium battery. The complicated part of this stage was the connection of each driver with its motor and the output pins of the Arduino MEGA to give the action signal.

Figure 4 shows the Arduino MEGA board connected to the other HC-05 module and its connection to only one of the four drivers.

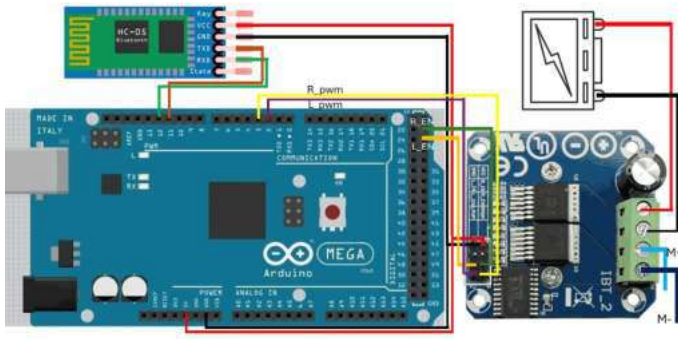


Fig. 4. Connection diagram for Arduino MEGA including the connection to a driver.

A wooden board was used on which all the components of this stage were placed, avoiding movement and improving their organization. Figure 5 shows the vehicle with the circuit installed on the board.



Fig. 5. Robotic base photograph and it's components.

## V. RESULTS AND ANALYSIS

The first part of the project, was obtaining the biopotential. Figure 6 shows a fragment of the processed signal, where it can be seen that its greatest potential is around 3V. This previous analysis was made in the language python separately from the algorithm used to move the vehicle, the analysis was made only to see the biopotencial obtained by the EMG.

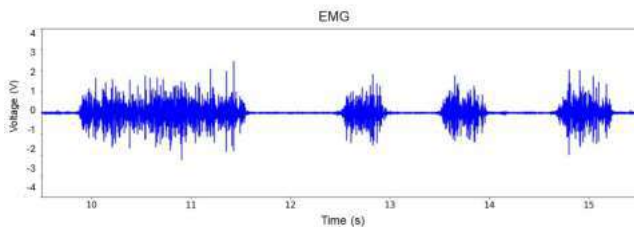


Fig. 6. Output signal from EMG.

One relatively long contraction (approximately 2 seconds) and three short contractions can be observed.

In some cases the signal was not in a basal state, that is, the voltage corresponding to rest was 5 V or -5 V. We believe that this event is due to a malfunction in the symmetrical power supply of the amplifier instrumentation; however, the capacitor was used to reduce this unwanted voltage.

Afterwards, the analysis of the rectified signal and its envelope was carried out; from this new signal the code created on the Arduino cards can be carried out.

This determines the duration of the gestures, the thresholds to be overcome, as mentioned in the section IV-B. In the Figure 7 you can see the same signal after this analysis.

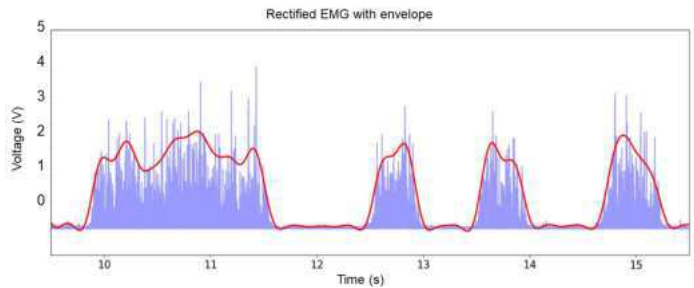


Fig. 7. EMG signal rectification and its envelope.

For the second stage of the project, the algorithm used was developed in the Arduino's free software language, and many tests were carried out. This part was worked on and simulated at home, but it required communication tests. It was after many tests that it was proposed to work with the biopotential of the masseter muscle instead of the frontalis muscle.

Once the assembly of the vehicle was completed, a movement test was carried out before controlling it with the EMG: the biosignal was simulated with a potentiometer, the action of advancing, turning and moving was sent through Bluetooth, and it was performed on the vehicle.

Finally, the stages were put together: the EMG signal was received by the Arduino UNO and, on the basis of this signal, it sent information to the Arduino MEGA, which was on the vehicle.

There was one particular challenge at this stage: The Arduino UNO had difficulty detecting the EMG signal, probably because the direct analog signal was being used which lacked continuous intervals. Likewise, the difference in frequencies in the basal and contraction states is a factor in the previous challenge, since with the basic algorithm a consistent signal cannot be obtained in both states.

The vehicle is able to move through the potentials of the jaw, however, due to the difficulty of registering the values in the different states, there is the possibility that the vehicle may take a while to perform the actions of the gestures made. Due to the previous challenge, it was tried using the EMG envelope signal using an analog envelope detector.

However, the Arduino UNO did not detect the signal and therefore did not send the message to the Arduino MEGA: all the time it remained in the stop command, regardless of the signal.

That is why it was decided to continue using the direct output signal of the instrumentation amplifier. Figure 8 shows the signal used to move the vehicle, notice that there is no consistent signal in some time intervals, causing the challenge commented above.

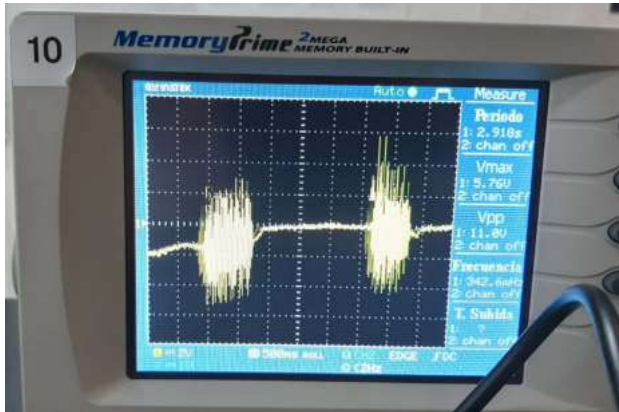


Fig. 8. EMG analog signal on an oscilloscope.

The analysis of the signal shown at the beginning of this section can solve the problem of detecting the signal as given by the Arduino and its sampling rate. Communication between programming languages could be carried out to obtain a cleaner signal suitable for processing, compared to the envelope detector or the raw signal. The problem that would arise is that communication must be almost instantaneous, so the code and the computer used must be as fast and efficient as possible.

## VI. CONCLUSIONS

An EMG circuit capable of recording biopotentials from facial muscles was designed and implemented. Using these signals and a low-complexity algorithm, the vehicle was controlled via Bluetooth. It is important to note that these results can be applied in various fields, such as medicine, where this system could be adapted to motorized wheelchairs for people with movement limitations, or in controlling a land vehicle to assist individuals with mobility impairments by transporting objects or manipulating devices within a limited range.

The use of EMG signals and data transmission via Bluetooth, along with the implementation of a more advanced algorithm using higher-level programming languages for improved signal analysis and near-real-time processing, could be expanded to develop more advanced and personalized assistive devices. This implementation of higher level programming could solve the faced challenge in a better way using the envelope directly in the algorithm. This would allow for better precision in controlling the vehicle.

In our case, the integration of EMG and Bluetooth technologies in the future could aid in the creation of more integrated and autonomous assistive systems, including the potential for obstacle detection when controlling the vehicle.

## ACKNOWLEDGMENT

The support of professors J. López-Aparicio, A. Villarreal-López, E. A. Jiménez-Díaz, G.A. Román-Chong and E. Christy Mckiernan is appreciated for their constant support and guidance for the development of this project. We also thank DGAPA UNAM PROYECTO PAPIME PE103924 AND PAPIME PE104223 for the funding provided.

## REFERENCES

- [1] J. Kraimer, J. Pollack, T. Wellman, E. Jensen, M. Crabill, K. Dykema. "Sistemas Y Metodos Para Controlar De Manera Remota Un Vehiculo De Manejo De Materiales ". Número de patente: MX/a/2009/002803
- [2] T. Kumai, "Extraction of the EPP component from the surface EMG," J Vis Exp. vol. 34, p. 1653, 2009.
- [3] D. Farina et al., "The Extraction of Neural Information from the Surface EMG for the Control of Upper-Limb Prostheses: Emerging Avenues and Challenges," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 22, no. 4, pp. 797-809, July 2014.
- [4] J. S. Artal-Sevil, A. Acón, J. L. Montañés and J. A. Domínguez, "Design of a Low-Cost Robotic Arm controlled by Surface EMG Sensors," 2018 XIII Technologies Applied to Electronics Teaching Conference (TAAE), La Laguna, Spain, 2018, pp. 1-8
- [5] R. Wan and C. Zhong, "Real time detection of muscle fatigue using arduino based surface EMG frequency and amplitude measurements," Journal of Student Academic Research, vol. 5 No. 1, 2024

# Design of a learning robot for use in therapies with autistic children: a mechatronic approach

Arturo N. Cordova-Zacarias    María S. Alejandre-Apolinar    Hugo Amores-Pérez    Irma A. García-Gonzalez  
*TecNM campus Xalapa    TecNM campus Xalapa    TecNM campus Xalapa    TecNM campus Xalapa*  
197o02053@itsx.edu.mx    salome.aa@xalapa.tecnm.mx    hugo.ap@xalapa.tecnm.mx    irma.gg@xalapa.tecnm.mx

Osmar S. Morales-Varela  
*TecNM campus Xalapa*  
187o07190@itsx.edu.mx

Octavio A. García-Alarcón  
*TecNM campus Xalapa*  
<https://orcid.org/0000-0002-0951-4476>

**Abstract**—Autism spectrum disorder (ASD) affects the configuration of the nervous system and brain function and is evidenced by brain impairments that appear differently in different individuals. These impairments impact people’s language patterns and communication. These individuals examine their environment in a particular way, facing obstacles to expression. Studies have shown that mechanical movements are what most attract attention in children with this condition. This paper presents the design of a robot for use in therapies with autistic children and this work is a collaboration between researchers from the Intituto de Investigaciones Cerebrales of the Universidad Veracruzana (IICE- Brain Research Institute of the University of Veracruz) and the Instituto Tecnológico Superior de Xalapa (ITSX -Technological Institute of Xalapa).

**Index Terms**—Robot, Prototype, Interaction, ASD, Therapy.

## I. INTRODUCTION

Autism spectrum disorders (ASD) are a diverse group of conditions. They are characterized by some degree of difficulty with social interaction and communication. Other characteristics are atypical patterns of activities and behaviours, such as difficulty with transition from one activity to another, a focus on details and unusual reactions to sensations [1].

The abilities and needs of autistic people vary and can evolve over time. Autism often has an impact on education and employment opportunities.

For this reason, attempts have been made to use robots to interact with people who have this disorder. Numerous therapeutic approaches have been created to support children with ASD in developing their cognitive skills, improving their daily living abilities, and fostering their active involvement in the community. The integration of assistive technologies into therapy sessions exemplifies this effort. The growing prevalence of ASD has underscored the need for novel technologies that can optimize and augment current therapeutic practices. A brief review of the literature and its applications is provided below:

In [2] a long a long-term study of children with autism playing with a robotic pet is presented. The study took inspiration from non-directive play therapy to encourage children’s proactivity and initiative-taking. At [4] is presented an investigation which robot features stimulate better responses from children with low-functioning autism in robot-assisted therapy.

The study concluded that features like a toy’s “face” and “moving limb” can improve engagement and facial expressions but might not contribute to other social communication skills. In [6] explore the use of the social robot Probo as a social story-telling agent for children with ASD, showcasing its potential in facilitating social communication. [7] investigated if the social robot Probo could help children with autism identify situation-based emotions. The findings showed that children’s performance in identifying sadness and happiness improved with the robot’s assistance. In [8], they compared the effectiveness of social stories delivered through social robots versus computer displays for children with ASD. The study suggested that the delivery method could influence the effectiveness of social stories. In [10] described the design, development, and application of RoboParrot as a screening tool to diagnose children with ASD. The results indicated that it was effective, highlighting the importance of robot-assisted screening. [9] investigated the long-term effects of LEGO therapy with a humanoid robot for children with ASD. The study reported positive impacts on social initiations and engagement. In [13] focused on developing accessible robots to improve the social skills of individuals with autism. [14] presented a robot-assisted behavioral intervention system for children with ASD, focusing on improving eye contact and social interaction skills. In [17] developed a low-cost, socially assistive robot and robot-assisted intervention for children with ASD. Field trials showed significant improvements in spontaneous utterances, social interaction, joint attention, and requesting behaviors. In [15] designed a parrot-inspired robot for robot-assisted therapy for learning and social interaction of children with ASD. The results indicated that the children were attracted to the robot and happy to interact with it. [18] explored how robots can be practically implemented into current education and therapy interventions for children with ASD through a co-creation study involving individuals with autism, parents, and professionals. [19] presented a feasibility study protocol for a randomized controlled trial investigating the effectiveness of using a humanoid robot (Kaspar) to improve social skills in children with ASD. In [20] measured engagement in robot-assisted autism therapy through a cross-cultural study, finding significant differences in engagement

displays between Japanese and Serbian children with ASD and in [23] proposed a personalized machine learning framework for robot perception of affect and engagement in autism therapy, emphasizing the importance of tailoring robot interactions to individual needs. [21] proposed deep-learning neural network architectures for automatically estimating children's visual attention during robot-assisted therapy, showcasing the potential of AI in assessing engagement.

As you have seen there's a lot of robot's-based therapies to work with ASD. This work is a collaboration between researchers from the UV-IICE and ITSX researchers. The "bamboo classroom" is currently in operation, where therapies are carried out on children with a history of autism and where prototypes are used to monitor, interact and assess the progress of the therapy in patients. The robot presented in this work is one of the new prototypes that seeks to better understand the autism spectrum disorder, generate relevant knowledge, propose strategies for its management and that can benefit patients from the use of this robot in motor and communication skills.

The development of these project begins with exploration. The principal objective is to investigate and analyze the specific needs of children with autism in the context of teaching and learning, as well as to evaluate the technologies and methodologies that exist in robotics applied to special education.

#### A. Objectives to be achieved in therapies

The person with autism exhibits unusual and atypical behaviors that cause difficulties in relating and adapting to their family and social environment (Hernández et al., 2015). These individuals often suffer from other medical and psychiatric conditions, which are variably related and contribute to a wide range of clinical manifestations. In addition, they often experience motor difficulties and lack coordination in their movements (Arberas, et al., 2019). When it comes to language characteristics in autistic syndrome, the absence of speech or a delay in its development is observed. In this sense, the main signs are reflected in the inability to maintain a social conversation (González et al., 2016). Identifying medical conditions or factors that trigger the disorder facilitates more effective monitoring of development, appropriate genetic counseling, prevention in relation to avoidable risk factors and, in the future, based on molecular discoveries, potential specific treatment (Arberas, et al., 2019).

There are various activities designed to work with children who have Autism Spectrum Disorder (ASD). Some of these activities are more complex and may need to be broken down into several parts or taught in steps (Soto, 2017). Some of these activities are listed below:

- Imitation: The child is provided with duplicate materials to mimic actions demonstrated by the therapist, such as putting a spoon in a cup or a ball in a box.
- Play: The use of cars and trains is encouraged to stimulate imagination and language development, with guidance provided to ensure appropriate play.

- Inserting, screwing, and fitting: This activity involves placing a piece and then giving instructions, starting with a single piece and gradually increasing complexity. Color sequences and templates are used to enhance difficulty.
- Mosaics: The child creates figures and discriminates colors in designs, starting with simple tasks like aligning pieces and progressing to more complex activities like placing pegs around shapes or on painted forms.
- Color learning: The child is prompted to touch a specific color, with assistance provided initially and gradually reduced. Once a color is mastered, the process is repeated with new colors, and the child is encouraged to generalize their learning by identifying colors in various objects and using cards.
- Repetitive behaviors: Children with ASD may engage in repetitive movements like rocking, spinning objects, or hand flapping, which can capture their attention and interfere with their ability to focus on relevant information. These behaviors can also be highly self-reinforcing, making external rewards less effective in promoting adaptive behaviors.
- Sensory sensitivities: Many children with ASD have specific sensory preferences and may dislike physical contact or certain textures, like new clothes or accessories. They may also avoid messy materials like paint or sand and show aversion to foods with certain textures.
- Visual and sensory stimuli: While they may not enjoy costumes or physical contact, children with ASD are often drawn to sensory stimuli like flashing lights, tactile textures, repetitive sounds, or visually stimulating patterns, such as spinning wheels or flashing toys.

In this work we develop an articulated robot prototype for activities focused on teaching-learning processes for children with autism which is adapted to this project seeking to innovate and develop new ways in this case of applying the mechanics of the structure and with it what it implies the displacement and movement of the robot's limbs, in such a way that it is a product that proposes a new technological application.

## II. ROBOT CONFIGURATION

The brain of these robot is a raspberry pi 4, an arduino board that interacts with the raspberry and with the sensors and actuators. These are the principal objects to interact with children in therapy. A little camera is integrated to detect colors and interactions of children in therapy too. Finally, SolidWorks software and 3D print was used for the design of the robot.

#### A. Robot displacement configuration

We start from the no holonomic robot configuration with a differential configuration and the shape that the base will have. A square base is chosen due to the adjustment and final assembly of different components. The base has various perforations which perform a specific function to work with other parts of the robot. The differential configuration requires

that the robot move freely on any surface and support a certain weight so we used all-terrain wheels from the Pololu brand with a load capacity of more than 15 kilograms and two suport free-movement wheels (Fig.1) .



Fig. 1. Principal and free-movement wheels.

Finally two DC motors are connected directly to the Pololu wheels and are attached directly to the base to prevent slipping and failures at the movement of the weels (Fig.2).

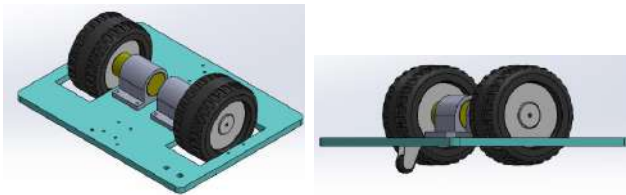


Fig. 2. Final base-wheels-motors assembly.

The 3D print and implementation with coordinates of its diferencial configuration is presented in Fig.3

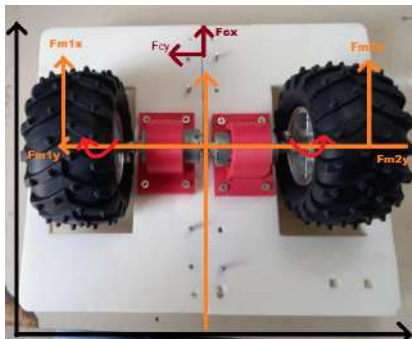


Fig. 3. Final 3D print wheels-base implementation.

### B. Robot body configuration

For the construction of the body, a square shape was chosen, because when assembling the mechanical and electronic components, it is easier and more convenient to handle (Fig.4). The 3D print and implementation is presented in Fig.5

1) *Robot arms:* The robot arms were designed taking into account that they must form a square or a 90-degree angle to be able to have a better perspective of movement. SG90 DC servomotors are used to implement the movement of the arm. The arm is directly connected to the servo motor to be able to

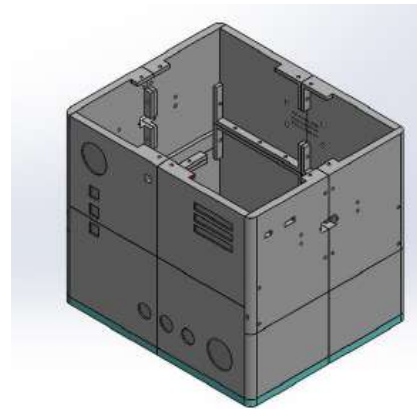


Fig. 4. Robot body-walls.

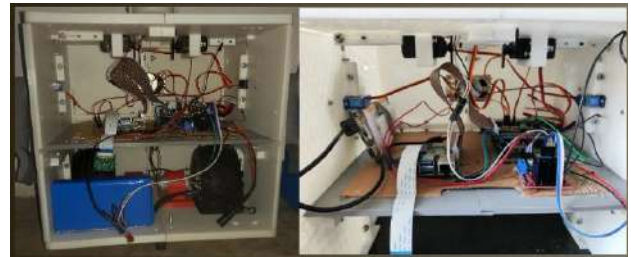


Fig. 5. Final 3D print body.

perform the movement established in the programming. The arms are then assembled to a servo motor guard, designed to protect it and prevent damage. This guard is attached to the lateral wall (Fig.6).

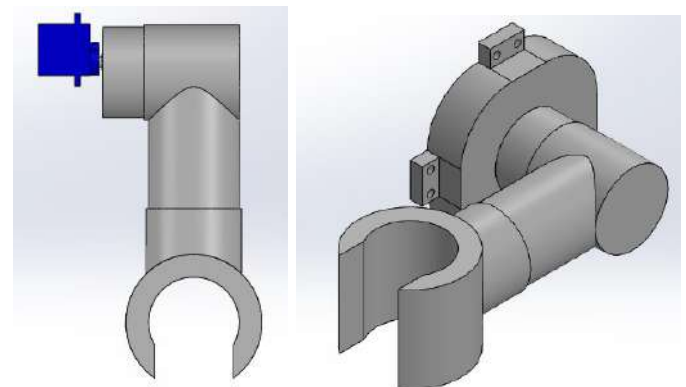


Fig. 6. Robot arm design.

The 3D print and implementation is presented in Fig.7

2) *Robot head:* The head design was made in three parts and because it is easier to put the electronic components inside it, as well as better and faster 3D printing (Fig.8).

The 3D print and implementation is presented in Fig.9

3) *Robot body:* The design consists of four walls and three floors. The first floor is the base where the motors and wheels are located, the second for the electronic components and finally the third for the head and neck of the robot. Each wall



Fig. 7. Final 3D arm implementation.



Fig. 8. Robot head with three parts.



Fig. 9. Final 3D neck and head implementation.

and base is divided into four parts for later joining due to the measurements of the 3D printer. The arms are assembly to the lateral walls with the guard and the servomotor (Fig.10). For the neck, two locks are assembled to the holes in the

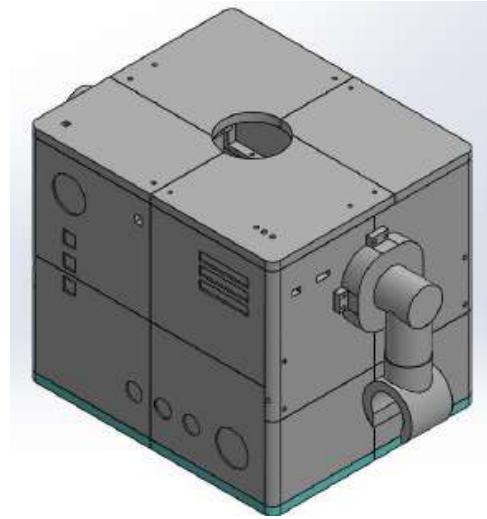


Fig. 10. Robot body and arm assembly.

neck to be able to make the vertical movement of the head through the two servomotors that are in the upper part of the body. Then the MG 995 servomotor is assembled to the neck, which is responsible for moving the head to obtain the horizontal movement. And finally, the neck is assembled to the two servomotors, which allow the vertical movement of the head (Fig.11). For the head part, the three parts are assembled.

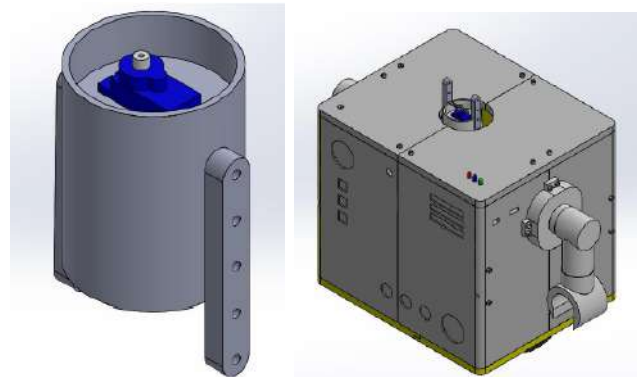


Fig. 11. Neck locks and neck-motors assembly.

First, the base is attached to the lower part of the head and, once the lower part of the head is assembled, the upper part is attached to the lower part to have the complete head. Finally, the camera lens is assembled, the head is assembled to the neck and then to the body of the robot to have the complete assembly (Fig.12).

The 3D print and implementation is presented in Fig.13

### C. Electronics and software configuration

Below in Fig.14 is the connection diagram that was based on the system. It consists of 3 phases: reaction, control and

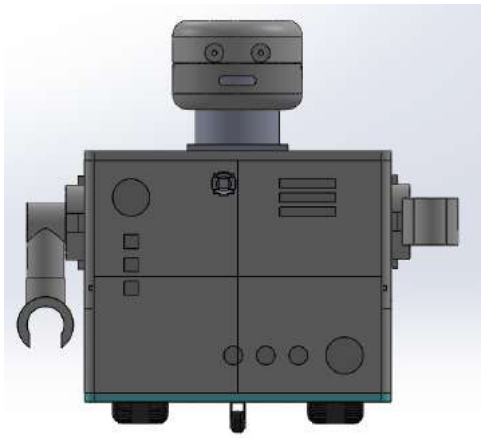


Fig. 12. Robot assembled.



Fig. 13. Final 3D robot implementation.

robotic field indicators. The indicators L1, L2 and L3 will be three red LEDs (sadness or anger), blue (surprise) and green (happiness) where each one shows the user's behavior towards the device. In the control there are the batteries, a TIP120 sound amplifier transistor, a Raspberry Pi 4B board that carries the automation in a Python code program. Also the modules such as the PCA9685 servo motor control module that helps in the control and correct function of the servo motors used and the L298N relay module which helps in the control and correct function of the traction motors of the prototype.

Finally, in the robotic field we find the servo motors, M1 and M2 motors, LED indicators and the buzzer. Servomotors 1 and 2 works as a mechanism in the robot's arms, servomotors 3 and 4 in the operation of the neck, servomotor 5 in the head turning mechanism and finally servomotors 6 and 7 for an eyes mechanism that is not implemented yet. In the head are two LEDs that simulate the blinking of the eyes and finally the speaker transmits the sounds related to interacting with the user.

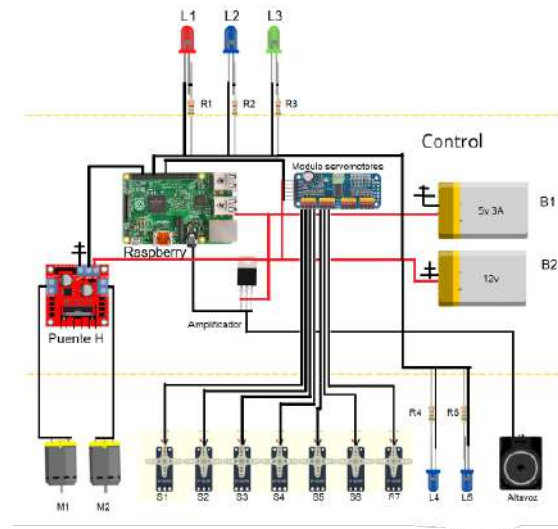


Fig. 14. Final 3D robot implementation.

Additionally, a camera is connected to the Raspberry Pi with a facial and color recognition algorithm to interact with patients.

### III. INTERACTION OBJETIVES WITH HUMANS

This robot is designed to interact with both neurodivergent and neurotypical patients, hence its robot form, which is intended to be friendly and engaging for them.

The robot's chest is equipped with a camera that, in conjunction with the Raspberry Pi's processing and implemented software, detects colors and shapes in established routines. The color of the LEDs on the robot's shoulder, together with the color of its eyes, is meant to interact with children and display "emotional states." Similarly, there is a speaker through which the robot can ask children to interact or express the results of activities.

Finally, the robot's differential configuration and the implementation of motors in its head and arms allow it to perform actions to interact with patients.

### IV. CONCLUSIONS AND FUTURE WORK

In this work, the mechatronic design of a robot to interact with children with autism was presented. The robot is used in the bamboo classroom of the Instituto Tecnológico Superior de Xalapa. Mechanical designs implemented in 3D and the electronic implementation were presented. Currently, the robot is being experimented with in interaction with patients to see if the interaction is successful or if some adjustments need to be made to the software and to see how long the design will take to make some hardware adjustments.

### REFERENCES

- [1] Centers for Disease Control and Prevention, What Is Autism Spectrum Disorder, accessed on 31st July 2024 on url <https://www.cdc.gov/ncbddd/autism/facts.html>.

- [2] Francois, D.; Powell, S.; Dautenhahn, K. A long-term study of children with autism playing with a robotic pet Taking inspirations from non-directive play therapy to encourage children's proactivity and initiative-taking. *Interact. Stud.* **2009**, *10*, 324–373.
- [3] Palsbo, S.E.; Hood-Szivek, P. Effect of Robotic-Assisted Three-Dimensional Repetitive Motion to Improve Hand Motor Function and Control in Children With Handwriting Deficits: A Nonrandomized Phase 2 Device Trial. *Am. J. Occup. Ther.* **2012**, *66*, 682–690.
- [4] Lee, J.; Takehashi, H.; Nagai, C.; Obinata, G.; Stefanov, D. Which Robot Features Can Stimulate Better Responses from Children with Autism in Robot-Assisted Therapy? *Int. J. Adv. Robot. Syst.* **2012**, *9*, 6.
- [5] Robins, B.; Dautenhahn, K.; Ferrari, E.; Kronreif, G.; Prazak-Aram, B.; Marti, P.; Iacono, I.; Gelderblom, G.J.; Bernd, T.; Caprino, F.; et al. Scenarios of robot-assisted play for children with cognitive and physical disabilities. *Interact. Stud.* **2012**, *13*, 189–234.
- [6] Vanderborght, B.; Simut, R.; Saldien, J.; Pop, C.; Rusu, A.S.; Pintea, S.; Lefeber, D.; David, D.O. Using the social robot Probo as a social story telling agent for children with ASD. *Interact. Stud.* **2012**, *13*, 348–372.
- [7] Anamaria, P.C.; Ramona, S.; Sebastian, P.; Jelle, S.; Alina, R.; Daniel, D.; Johan, V.; Dirk, L.; Bram, V. Can the social robot probo help children with autism to identify situation-based emotions? A series of single case experiments. *Int. J. Hum. Robot.* **2013**, *10*, 24.
- [8] Pop, C.A.; Simut, R.E.; Pintea, S.; Saldien, J.; Rusu, A.S.; Vanderfaillie, J.; David, D.O.; Lefeber, D.; Vanderborght, B. Social robots vs. computer display: Does the way social stories are delivered make a difference for their effectiveness on asd children? *J. Educ. Comput. Res.* **2013**, *49*, 381–401.
- [9] Barakova, E.I.; Bajracharya, P.; Willemsen, M.; Lourens, T.; Huskens, B. Long-term LEGO therapy with humanoid robot for children with ASD. *Expert Syst.* **2015**, *32*, 698–709.
- [10] Dehkordi, P.S.; Moradi, H.; Mahmoudi, M.; Pouretemad, H.R. The Design, Development, and Deployment of RoboParrot for Screening Autistic Children. *Int. J. Soc. Robot.* **2015**, *7*, 513–522.
- [11] Ueyama, Y. A Bayesian Model of the Uncanny Valley Effect for Explaining the Effects of Therapeutic Robots in Autism Spectrum Disorder. *PLoS ONE* **2015**, *10*, e0138642.
- [12] Coeckelbergh, M.; Pop, C.; Simut, R.; Peca, A.; Pintea, S.; David, D.; Vanderborght, B. A Survey of Expectations About the Role of Robots in Robot-Assisted Therapy for Children with ASD: Ethical Acceptability, Trust, Sociability, Appearance, and Attachment. *Sci. Eng. Ethics* **2016**, *22*, 47–65.
- [13] Tennyson, M.F.; Kuester, D.A.; Casteel, J.; Nikolopoulos, C. Accessible robots for improving social skills of individuals with autism. *J. Artif. Intell. Soft Comput. Res.* **2016**, *6*, 267–277.
- [14] Yun, S.S.; Kim, H.; Choi, J.; Park, S.K. A robot-assisted behavioral intervention system for children with autism spectrum disorders. *Robot. Auton. Syst.* **2016**, *76*, 58–67.
- [15] Bharatharaj, J.; Huang, L.L.; Al-Jumaily, A.; Elara, M.R.; Krageloh, C. Sociopsychological and physiological effects of a robot-assisted therapy for children with autism. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 11.
- [16] Bharatharaj, J.; Huang, L.L.; Elara, M.R.; Al-Jumaily, A.; Krageloh, C. Robot-Assisted Therapy for Learning and Social Interaction of Children with Autism Spectrum Disorder. *Robotics* **2017**, *6*, 4.
- [17] Boccanfuso, L.; Scarborough, S.; Abramson, R.K.; Hall, A.V.; Wright, H.H.; O'Kane, J.M. A low-cost socially assistive robot and robot-assisted intervention for children with autism spectrum disorder: Field trials and lessons learned. *Auton. Robot.* **2017**, *41*, 637–655.
- [18] Huijnen, C.; Lexis, M.A.S.; Jansens, R.; de Witte, L.P. How to Implement Robots in Interventions for Children with Autism? A Co-creation Study Involving People with Autism, Parents and Professionals. *J. Autism Dev. Disord.* **2017**, *47*, 3079–3096.
- [19] Mengoni, S.E.; Irvine, K.; Thakur, D.; Barton, G.; Dautenhahn, K.; Goldberg, K.; Robins, B.; Wellsted, D.; Sharma, S. Feasibility study of a randomised controlled trial to investigate the effectiveness of using a humanoid robot to improve the social skills of children with autism spectrum disorder (Kaspar RCT): A study protocol. *BMJ Open* **2017**, *7*, 10.
- [20] Rudovic, O.; Lee, J.; Mascarell-Maricic, L.; Schuller, B.W.; Picard, R.W. Measuring Engagement in Robot-Assisted Autism Therapy: A Cross-Cultural Study. *Front. Robot. Ai* **2017**, *4*, 1–17.
- [21] Di Nuovo, A.; Conti, D.; Trubia, G.; Buono, S.; Di Nuovo, S. Deep Learning Systems for Estimating Visual Attention in Robot-Assisted Therapy of Children with Autism and Intellectual Disability. *Robotics* **2018**, *7*, 25.
- [22] Feng, Y.L.; Jia, Q.X.; Wei, W. A Control Architecture of Robot-Assisted Intervention for Children with Autism Spectrum Disorders. *J. Robot.* **2018**, *2018*, 3246708.
- [23] Rudovic, O.; Lee, J.; Dai, M.; Schuller, B.; Picard, R.W. Personalized machine learning for

# Prototipo de simulación de una Mimosa púdica

Rodolfo Romero Herrera  
Escuela Superior de Cómputo  
Sección de Estudios de Posgrado e Investigación  
Instituto Politécnico Nacional  
Ciudad de México, México  
rromero@ipn.mx

María Fernanda Guillén Pérez  
Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías  
Avanzadas  
Instituto Politécnico Nacional  
Ciudad de México, México  
mguillenp1800@alumno.ipn.mx

**Resumen**— Las plantas, como la Mimosa púdica, exhiben movimientos rápidos en respuesta a estímulos externos a través de estructuras especializadas llamadas pulvínulos. Este fenómeno, conocido como sismonastia, permite que la mimosa se defienda de depredadores al plegar sus folíolos de manera rápida. El estudio de estos movimientos no solo revela mecanismos de defensa en las plantas, sino también posibles aplicaciones en la creación de sensores o robots blandos.

Inspirados por el movimiento de la Mimosa púdica, desarrollamos un prototipo utilizando una impresora 3D. Este se encuentra conectado a un circuito que incluye una Raspberry Pi Pico W, un sensor y un servomotor. Mediante un código, logramos simular el movimiento característico de la Mimosa púdica.

**Palabras Clave:** mimosa, folíolo, prototipo, movimiento.

## I. INTRODUCCIÓN

Las plantas muestran movimientos sofisticados y versátiles que les permiten ser eficientes energéticamente y ser muy flexibles, lo cual encaja perfectamente con los requerimientos principales para el diseño de robots blandos. Sin embargo, aún existe un bajo número de robots inspirados en plantas en comparación con los que existen inspirados en animales. [1]

Por lo tanto, es importante continuar implementando los estudios existentes del movimiento de las plantas, en especial de aquellas cuyos movimientos son iniciados y dirigidos por estímulos externos, pues este tipo de comportamientos son muy útiles en el diseño de sistemas con sensores, entre otros.

Una de las plantas cuyo estudio podría aportar en esta implementación de robots blandos es la *Mimosa púdica*, conocida por diversos nombres en la República Mexicana como “dormilona”, “sensitiva”, entre otros términos, la mimosa es un pequeño arbusto perenne, reconocido por su habilidad de plegar de manera rápida sus folíolos que conforman sus hojas en respuesta al tacto.

## II. MECANISMO DE LA MIMOSA PÚDICA.

La mimosa está conformada por un eje primario, constituido por un pecíolo que une el tallo con la hoja, y un raquis foliar (el eje central de la hoja). De este eje primario, parten varios pares de ejes secundarios llamados raquis de las pinnas, los cuales sostienen a los folíolos (las hojas de la mimosa). Cada una de estas divisiones tiene en su base un pulvínulo: un tejido de color oscuro encargado de engrosarse o contraerse para generar movimientos, como se muestra en la Figura 1. [2]

El ángulo de la hoja o folíolo cambiará debido a la turgencia en las células del pulvínulo, una parte o estructura en la base del pecíolo muy especializada, en un proceso de ósmosis, donde entran iones de potasio a la célula, provocando que el medio interno se haga hipertónico respecto al exterior y que se produzca esa turgencia. [3]

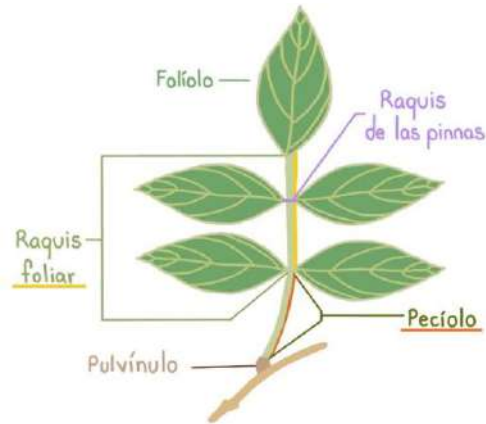


Fig. 1 Elementos de la Mimosa púdica

Dependiendo de si dicha turgencia tiene lugar en las células flexoras o extensoras (figura 2), los folíolos se mantendrán abiertos, se cerrarán, o, en ocasiones, descenderá toda la hoja.

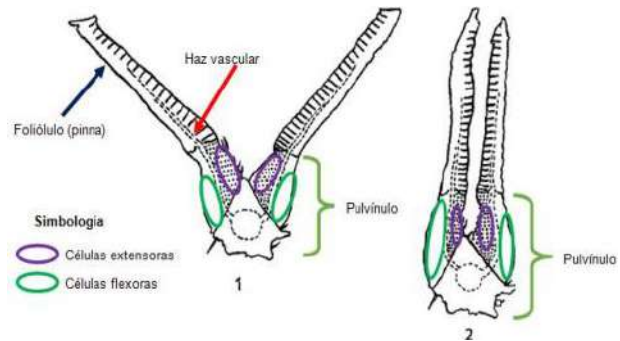


Fig. 2 Pulvínulo de la Mimosa púdica.

Esta respuesta natural presente en la planta se conoce como sismonastia, un tipo de movimiento foliar mecánico que sirve como mecanismo de defensa ante sus depredadores. Al replegarse en un gran porcentaje, da la apariencia de ser una planta marchita; además, esta postura le ayuda a no perder demasiada agua durante las horas de calor y a protegerse del viento al reducir su superficie.

Así mismo, los folíolos tienen un movimiento nocturno conocido como nictinastia, que señala el ritmo circadiano vegetal regulado por la luz. Durante este tipo de movimiento, la mimosa mantiene sus folíolos extendidos durante el día y plegados por la noche. Sin embargo, este movimiento tiene un costo, ya que la planta pierde el 40% de su capacidad de fotosíntesis (la captación de energía solar) al cerrarse y tarda entre 15 y 20 minutos en volver a abrirse.

Las respuestas de la mimosa a los estímulos externos están reguladas internamente y se determinan por procesos químicos que involucran la difusión de iones en forma de potenciales de acción, los cuales se propagan de célula a célula, en analogía con la propagación de los impulsos nerviosos a lo largo de las neuronas.

En múltiples trabajos realizados sobre las mimosas, se ha observado que un estímulo eléctrico al tacto de los folíolos envía una señal que desencadena diversos movimientos, dependiendo de la intensidad del estímulo y del área involucrada.

### III. PROTOTIPO DE LA MIMOSA

El prototipo se creó utilizando el programa “Autodesk Inventor Professional 2024”, donde se diseñó una estructura que permitiera el movimiento de cierre de los folíolos, la cual consta de un total de cinco piezas.

#### A. Base de los folíolos

Representa al raquis foliar de la mimosa, en el centro lleva un rectángulo de 150 mm de largo y 5 mm de ancho, al que están unidos un total de 20 círculos: 10 en cada lado. Estos a su vez tienen un cilindro al centro para colocar en este a los folíolos.

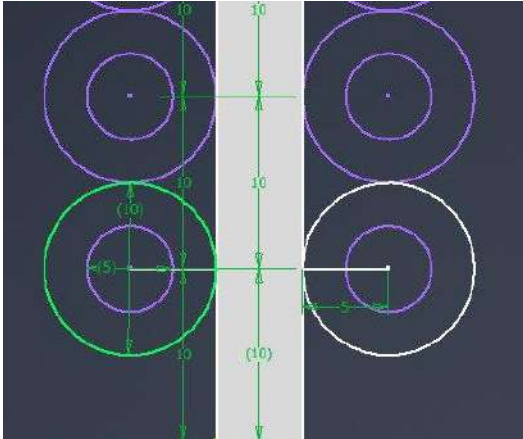


Fig. 3 Medidas de la base de los folíolos [mm].



Fig. 4 Base de los folíolos.

#### B. Base de los folíolos (Tapa)

Tiene la misma cantidad de espacios para los folíolos, su función es mantenerlos fijos permitiendo el correcto movimiento de estos.



Fig. 5 Base de los folíolos (Tapa).

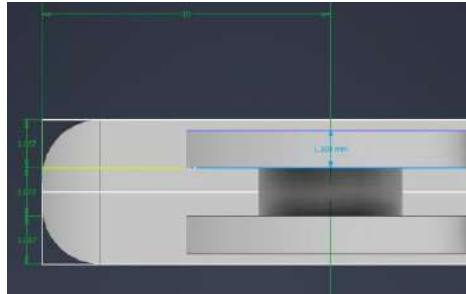


Fig. 6 Vista lateral de la base ensamblada.

#### C. Folíolos

Tienen una forma elíptica inspirada en los folíolos de las mimosas (figura 7). Su raquis de las pinnas es más largo que el de la mimosa, lo que permite colocar la cubierta de forma correcta sin interferir en el movimiento de los folíolos.



Fig. 7 Mimosa.



Fig. 8 Modelo de los folíolos.

#### D. Cubiertas

Juntas, conforman un cilindro. Estas mitades están compuestas por 20 compuertas rectangulares que permiten el paso del raquis de las pinnas de cada uno de los folíolos. De tal forma que, una vez armado todo el prototipo, al jalar la base de los folíolos, los raquis de las pinnas se moverán hacia

arriba, como ocurre con la mimosa cuando entra en contacto con ella.

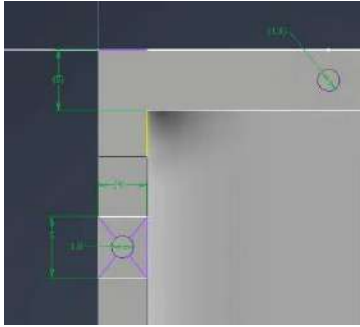


Fig. 9 Vista frontal de la cubierta.

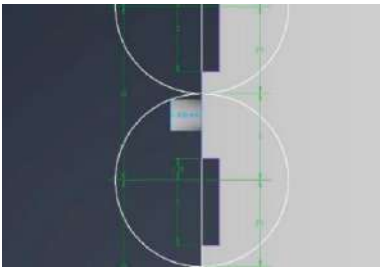


Fig. 10 Vista lateral de la cubierta.



Fig. 11 Cubiertas.

#### IV. PROTOTIPO DE LA MIMOSA

La impresión del modelo de nuestra mimosa se realizó en una impresora 3D modelo Elegoo Neptune 3 Pro, utilizando un filamento PLA tipo cristal. Al tener medidas milimétricas, fue necesario lijar las piezas después de ser impresas.

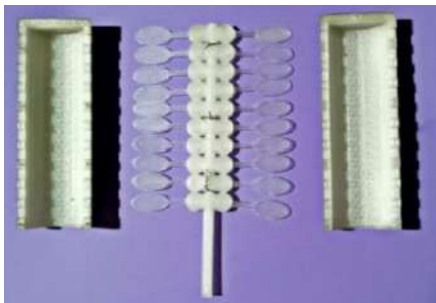


Fig. 12 Prototipo de la mimosa.

La simulación del movimiento de la mimosa requería de componentes que permitieran que al contacto se realizara el movimiento, por tanto, se utilizaron los siguientes componentes para la realización del circuito:

1) Raspberry Pi Pico W

- 2) Protoboard
- 3) Sensor de movimiento PIR
- 4) Servomotor SG90
- 5) Jumpers tipo hembra-macho
- 6) Jumpers tipo macho-macho
- 7) Cabezales macho

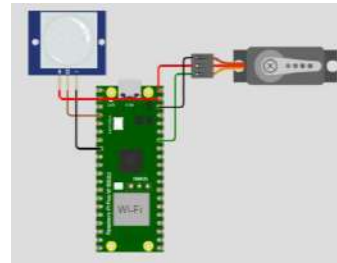


Fig. 13 Configuración del circuito.

Para conectar el prototipo al circuito, se fabricó una cubierta de lámina, como se muestra en la Figura 14. Esta cubierta, junto con un resorte y una palanca, permite que el servomotor accione la base de los folíolos.

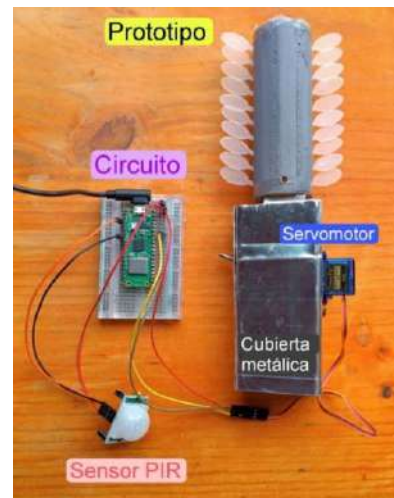


Fig. 14 Ensamble de los componentes.

#### V. FUNCIONAMIENTO

Se desarrolló un código en el programa Thonny utilizando MicroPython, en el cual se configuraron las salidas de la Raspberry Pi, se declararon los componentes necesarios y se definió el movimiento del servomotor para que el prototipo emulara el comportamiento de la mimosa al detectar movimiento. Además, se programó la reapertura de los folíolos tras un tiempo determinado desde el estímulo, en este caso de manera representativa en 20 segundos, en lugar de los 15-20 minutos que tarda la planta real.

Para darle un aspecto más natural a la mimosa, pintamos el prototipo de verde y colocamos los componentes del circuito en una “maceta”. La protoboard se ubicó junto a la cubierta metálica y el sensor PIR se posicionó dentro de la cubierta, sobre una base, de modo que solo detectara estímulos de contacto directo con la mimosa, gracias a las ranuras en la parte superior de la cubierta.

Finalmente, activamos el prototipo. Una vez conectada la Raspberry Pi y detectada la señal de movimiento, el

servomotor se activó, moviendo la base de los folíolos y provocando el cierre exitoso de la mimosa.



Fig. 15 Prototipo antes de detectar el movimiento.



Fig. 16 Prototipo justo después de detectar el movimiento.



Fig. 17 Prototipo 20 segundos después de detectar el movimiento.

## VI. CONCLUSIONES

Se logró exitosamente la creación del prototipo de la mimosa, y gracias a la correcta colocación del sensor, se pudo simular el característico movimiento de sus hojas. Al detectar objetos cercanos, el mecanismo de cierre se activa al entrar en contacto con la planta. No obstante, es fundamental continuar investigando el movimiento de las plantas para mejorar la simulación de la mimosa y explorar las múltiples aplicaciones que el estudio de este tipo de comportamientos puede ofrecer.

## RECONOCIMIENTO

Los autores agradecen el apoyo recibido para desarrollo de este proyecto al IPN (Instituto Politécnico Nacional).

## REFERENCIAS

- [1] S. D. Cezan, H. T. Baytekin, and B. Baytekin, "Self-regulating plant robots: Bioinspired heliotropism and nyctinasty," *Soft Robotics*, vol. 7, no. 1, pp. 1–12, Jan. 2020, doi: 10.1089/soro.2019.0036.
- [2] V. Betancur, D. Conde & E. Zapata El movimiento como estrategia : relación biomecánica entre el tipo de estímulo físico y la respuesta mecánica de la Mimosa Púdica. [online]. Disponible en: <http://hdl.handle.net/20.500.11912/3715>.
- [3] L. Olvera Uribe, "Cultivo in vitro de Mimosa Pudica L., la planta sensitiva de México", Tesis de licenciatura, Benemérita Universidad Autónoma de Puebla, 2018. [online]. Disponible en: <https://hdl.handle.net/20.500.12371/8302>.
- [4] M. Rodríguez, «La dormilona, ni tan mimosa, ni tan púdica», *CMS*, vol. 62, n.º 4, pp. 67–68, ene. 2023.
- [5] R. Traslosheros López, L. Giordano y F. Hernández Cabrera. "Circuito eléctrico equivalente del pulvínulo de la Mimosa Pudica L. – Repositorio Institucional UANL". Bienvenido al Repositorio Institucional UANL - Repositorio Institucional UANL.

# CAPÍTULO 4



# ROBÓTICA EDUCATIVA

# Estrategia para la enseñanza de programación a través de robots de minisumo

Mario Salvador Castro Zenil  
TecNM Instituto Tecnológico  
Superior de Pátzcuaro  
Pátzcuaro, México  
mcastro@itspa.edu.mx  
0000-0001-6880-1072

Alvarado López Alma  
TecNM Instituto Tecnológico  
Superior de Pátzcuaro  
Pátzcuaro, México  
aalvarado@itspa.edu.mx  
0000-0002-2995-6749

Guillermo Rey Peñaloza M.  
TecNM Instituto Tecnológico  
Superior de Pátzcuaro  
Pátzcuaro, México  
grey@itspa.edu.mx  
0000-0003-2795-670X

**Resumen**—La enseñanza de la programación en la ingeniería es un proceso complejo, el cual históricamente en la Carrera de Ingeniería Biomédica ha representado un reto, pues la mayoría de los estudiantes que ingresan a este programa educativo no han tenido en su formación bachiller acercamiento a temas de programación. La robótica de competencia, como lo son los robots de minisumo, generan en los estudiantes de nuevo ingreso motivación por aprender, para poder participar en diversos concursos, y si bien han logrado la implementación de prototipos funcionales, los conceptos básicos de programación no son comprendidos de la mejor manera y esto se ve representado en sus evaluaciones y en la deficiencia de conceptos al implementar la programación en otras asignaturas. Lo que se propone es construir un analizador léxico y sintáctico para la evaluación de pseudocódigo que sirva para manipular un robot de minisumo controlado por Bluetooth, y de esta forma se comprendan los conceptos básicos como funciones, asignaciones, estructuras de decisión y ciclos, teniendo como salida en el caso de que el pseudocódigo esté bien escrito un código en lenguaje Arduino funcional para poder ser grabado en un microcontrolador Arduino, y que nos permita manipular un minirobot móvil.

**Palabras Clave:** Robot, Minisumo, Pseudocódigo, A

**Abstract**— The teaching of programming in engineering is a complex process, which historically in the Biomedical Engineering career has represented a challenge, since most of the students who enter this educational program have not had in their baccalaureate education an approach to programming topics. Competition robotics, such as mini-sumo robots, generate in new students motivation to learn, in order to participate in various competitions, and although they have achieved the implementation of functional prototypes, the basic concepts of programming are not understood in the best way and this is represented in their evaluations and in the deficiency of concepts when implementing programming in other subjects. What is proposed is to build a lexical and syntactic analyzer for the evaluation of pseudocode that serves to manipulate a Bluetooth-controlled mini-sumo robot, and in this way the basic concepts such as functions, assignments, decision structures and cycles are understood, having as output in the case that the pseudocode is well written a code in functional Arduino language to be recorded in an Arduino microcontroller, and that allows us to manipulate a mobile minirobot.

**Keywords**— Robot, Minisumo, Pseudocode, Arduino, Lexer, Parser

## I. INTRODUCCIÓN

La enseñanza de la programación en el ámbito de la ingeniería se ha transformado significativamente en las últimas décadas. El enfoque pedagógico se centraba en lenguajes de programación convencionales y algoritmos abstractos. Sin embargo, la integración de la robótica y el uso del pseudocódigo como herramienta didáctica ha emergido como una metodología efectiva para enseñar conceptos básicos de programación, como condicionales y ciclos, de manera más intuitiva y práctica.

La programación es una habilidad fundamental para los ingenieros, ya que nos permite resolver problemas complejos, automatizar procesos y desarrollar nuevas tecnologías. Según Wing[5], la programación es un componente esencial del pensamiento computacional, una habilidad indispensable en la educación contemporánea de ingeniería. Sin embargo, los estudiantes a menudo enfrentan desafíos significativos al aprender a programar, especialmente cuando se les presenta de manera abstracta y teórica.

Las metodologías tradicionales de enseñanza de la programación en ingeniería, a menudo incluyen la resolución de problemas en lenguajes de programación específicos, como C, Java o Python. No obstante, estudios han demostrado que los enfoques prácticos y basados en proyectos pueden mejorar significativamente la comprensión y retención de conceptos de programación [2].

La robótica educativa es una estrategia que ha ganado popularidad en la enseñanza de la programación. La robótica ofrece una plataforma tangible para aplicar conceptos abstractos, facilitando una comprensión más profunda y práctica de la programación. La investigación de Chambers et al. [6] muestra que los estudiantes que aprenden a programar utilizando robots no solo encuentran el proceso más atractivo, sino que también muestran una mayor retención de los conceptos.

El pseudocódigo es una herramienta poderosa en la enseñanza de la programación, especialmente para principiantes. Su uso permite a los estudiantes enfocarse en la lógica y estructura de los algoritmos sin la carga adicional de la sintaxis específica de un lenguaje de programación. Como señala McConnell [3], el pseudocódigo es esencial para enseñar el diseño algorítmico y la resolución de problemas de manera efectiva.

La integración de la robótica y el pseudocódigo en la enseñanza de la programación combina lo mejor de ambos mundos. Por un lado, la robótica proporciona una plataforma

interactiva y motivadora para aplicar conceptos de programación. Por otro lado, el pseudocódigo permite a los estudiantes desarrollar y refinar su lógica algorítmica sin preocuparse por los detalles sintácticos. Este enfoque dual facilita una comprensión más holística y profunda de la programación [1].

Existen herramientas que nos permiten la enseñanza a través de pseudocódigo, una de ellas es Raptor, es un software de licencia abierta, basado en el uso de diagramas de flujo, que facilita el aprendizaje de conceptos básicos de programación a través de pseudocódigo. Su interfaz intuitiva y visual permite a los estudiantes diseñar y ejecutar algoritmos sin preocuparse por la sintaxis específica de un lenguaje de programación. Según Carlisle et al. [4], Raptor ha demostrado ser eficaz en la enseñanza de estructuras de control y conceptos algorítmicos a estudiantes de ingeniería.

Dos de los conceptos fundamentales en la programación son las estructuras condicionales y los ciclos. Las estructuras condicionales permiten a los programas tomar decisiones basadas en condiciones específicas, mientras que los ciclos permiten la repetición de instrucciones hasta que se cumpla una condición particular. La enseñanza de estos conceptos a través de la robótica y el pseudocódigo puede ser particularmente efectiva. Por ejemplo, al programar un robot para moverse y reaccionar a su entorno, los estudiantes pueden ver directamente cómo las decisiones (condicionales) y las repeticiones (ciclos) afectan el comportamiento del robot.

Diversos estudios de caso han demostrado la efectividad de este enfoque. Por ejemplo, un estudio realizado por Eguchi [7] encontró que los estudiantes que participaron en programas educativos basados en robótica mostraron mejoras significativas en sus habilidades de programación y resolución de problemas. Además, los ejercicios prácticos con robots permiten a los estudiantes experimentar de primera mano cómo se implementan y ejecutan las estructuras condicionales y los ciclos en situaciones del mundo real.

## II. METODOLOGÍA

### A. Planteamiento del Problema

En el Programa Educativo de Ingeniería Biomédica, históricamente más del 80% de los estudiantes de nuevo ingreso no tienen conocimientos de programación, lo cual implica un reto, pues los porcentajes de reprobación en la asignatura de fundamentos de programación son altos. Al buscar alternativas de enseñanza que permitieran reducir los índices de reprobación y deserción, se optó por introducir el desarrollo de prototipos de robótica móvil, buscando que se genera un proceso atractivo y que pudieran “ver” en acción, si bien esta estrategia mejoró los índices, la comprensión de los fundamentos y su aplicación en la resolución de otros problemas aún era deficiente.

Trabajamos en el desarrollo de un pseudocódigo que permitiera aprender los conceptos básicos, sin la rigidez de los lenguajes formales, para esto se realizó un pseudo-lenguaje a través de ANTLR, lo cual permitió mejorar en el proceso de comprensión de conceptos básicos como lo son las estructuras condicionales y los ciclos.

Por lo cual, se plantea un proceso que incorpore ambos conceptos, poder evaluar que el pseudocódigo tenga una sintaxis y estructura correcta y que este se transforme en código Arduino, la implementación de este Lenguaje Robot está en la operación de un robot de mini-sumo controlado por Bluetooth.

### B. Lenguaje Robot con AntLR

Lenguaje Robot esta administrado a través de una interfaz gráfica, en la cual vamos a poder escribir nuestro programa en pseudocódigo, basado en las reglas definidas y que pasa por el analizador léxico y sintáctico para determinar si cumple con las reglas definidas en el lenguaje. Para esto se hace uso de la herramienta ANTLR.

ANTLR (ANother Tool for Language Recognition) es una herramienta de generación de analizadores léxicos y sintácticos. Es una herramienta poderosa que facilita la creación de compiladores, intérpretes y otras herramientas de procesamiento de lenguajes. Nos permite definir gramáticas formales para lenguajes, tanto a nivel léxico como sintáctico, usando una notación de gramática específica. A partir de dichas definiciones, se genera automáticamente código fuente en varios lenguajes de programación, incluidos Java, C#, Python y otros.

Esta herramienta es un software que genera analizadores pred-LL(k), y él mismo utiliza un analizador pred-LL(k) para leer los ficheros en los que están escritas las reglas. ANTLR admite acciones en sus reglas, además de otras prestaciones como paso de parámetros, devolución de valores o herencia de gramáticas.

Se desarrolla un analizador léxico, donde se verifican los “tokens”, que son las palabras reservadas del lenguaje, los espacios blancos, los saltos de línea, los comentarios, las literales, los operadores aritméticos, los operadores de comparación, los operadores lógicos y el proceso de indentación, como forma de organización del código. Podemos observar, por ejemplo el código para los tokens y los operadores lógicos en la figura 1.

```
1 tokens{
2   INM= "immediatly";
3   SLW= "slow";
4   R_VOID= "void";
5   R_MOVE= "move";
6   R_TURN= "turn";
7   R_STOP= "stop";
8   R_TAKE= "take";
9   R_VEL= "velocity";
10  R_IF= "if";
11  R_THEN= "then";
12  R_ELSE= "else";
13  R_ENDIF= "endif";
14  R_ENDW= "endw";
15  R_WHILE= "while";
16  R_FW="forward";
17  R_BW="backward";
18  R_DO= "do";
19  R_VARI="vari";
20  R_VARBOOL="varbool";
21  R_VARS="vars";
22  BEGIN= "begin";
23  END= "end";
24 }
25 OP_OR= "OR";
26 /**AND**/
27 OP_AND= "AND";
28 /**NOT**/
29 OP_NOT= "NOT";
```

Fig. 1. Definición de Tokens

Para el Analizador Sintáctico se tendrá un lookahead igual a cuatro, como se observa en la Figura 2, en la identificamos la definición de las opciones y la instrucciones para construir el AST, que después se usa para obtener los tokens;  $K=4$  es el mínimo necesario para diferenciar nuestros tokens.

En esta parte podemos observar la construcción de las reglas de nuestro lenguaje, por ejemplo, como se definen las estructuras de control y condicionales. Para cuestiones como

```
if a>5 then
    b=6;
endif
while a==5 do
    fa="hola";
endw
```

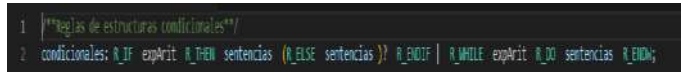


Fig. 2. Definición de reglas de estructuras condicionales

Una vez que tenemos definido nuestro lenguaje y las reglas de este, procedemos a la creación de la interfaz gráfica, donde podemos poner en ejecución nuestro pseudocódigo y nos permite evaluar los códigos que los estudiantes diseñen, para verificar la conformidad del uso de las reglas del lenguaje, es decir, su léxico y del uso del mismo, es decir, su sintaxis.

En la Figura 3 podemos observar la interfaz de desarrollo de la aplicación, en este caso se hizo uso de la herramienta NetBeans y el ANTLR a través Java. En la Figura 4, podemos ver la Interfaz de la aplicación.



Fig. 3. Interfaz de desarrollo de la Aplicación



Fig. 4. Área de trabajo del Lenguaje Robot

En la Figura 5, podemos observar un código de ejemplo de un programa escrito en Robot Lenguaje y en la Figura 6, se observa el resultado del análisis, donde se separa cada uno de los tokens, en el caso de que el resultado del análisis sea correcto.

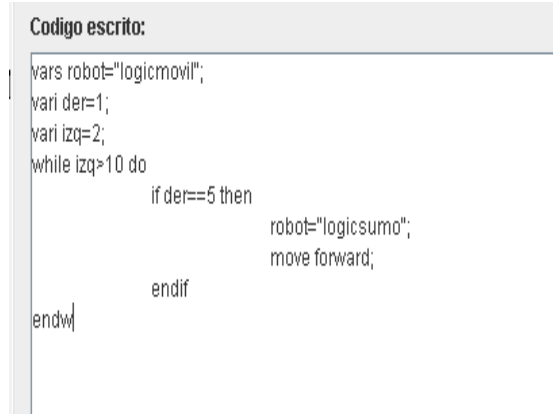


Fig. 5. Código en Lenguaje Robot

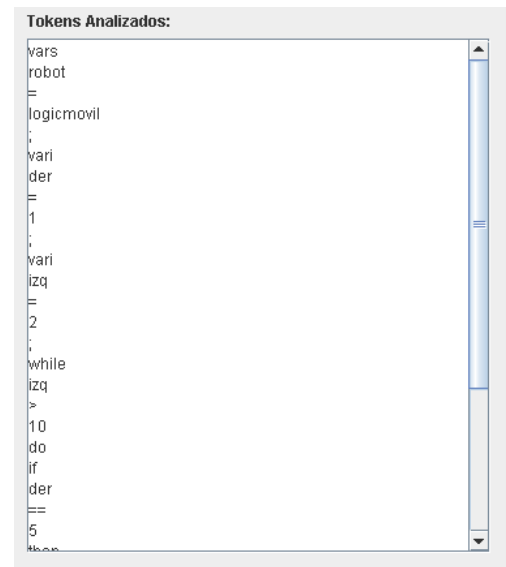


Fig. 6. Resultado del Tokenizador

### III. ROBOT DE MINISUMO

Para la construcción del robot móvil, se realiza el diseño del mismo para su manufactura e impresión en 3D, como el proceso está pensado para estudiantes de primer semestre, no se genera un proceso completo de control, sino se hace uso de un control bluetooth que nos permite manipularlo como un carro de control remoto, pero siguiendo las características de un robot de competencia.

Para esto se utiliza una serie de componentes básicos, como motores pololu 50:1, tracked Wheel de 22 dientes, un batería LiPo de 11 volts, un controlador de motores TB67H420FTG, y como cerebro, un Arduino Nano 33 BLE, el cual incorpora un módulo Bluetooth.

La carcasa del robot es muy simple, una base donde se anclan los motores y se puedan colocar los componentes electrónicos y una placa, para poder empujar a los rivales fuera

del dojo. El diseño se realiza en Fusión 360, dado que es un software que nos permite contar con todas sus herramientas a través de una licencia educativa.

Para la laminación de los modelos se utilizó el software Prusa Slicer como podemos verlo en la Figura 7 y Figura 8. El ensamblado de las piezas se realiza de una forma muy intuitiva como se observa en la figura 9.

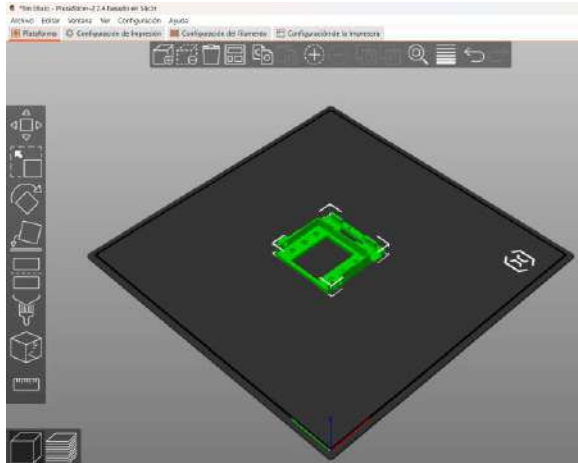


Fig. 7. Laminado de la base

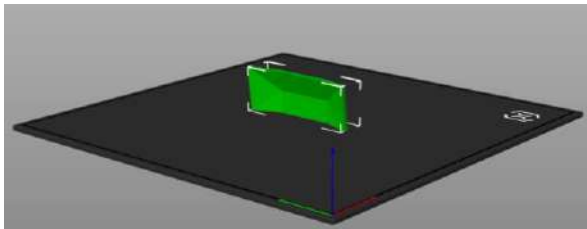


Fig. 8. Laminado de la rampa



Fig. 9. Armado del Robot

#### IV. RESULTADOS

##### A. Analizador y Traductor

Para realizar el proceso de traducción, se implementa un Lexer (Analizador Léxico), el cual es un programa que convierte una secuencia de caracteres en una secuencia de elementos o tokens, y un Parser (Analizador Sintáctico), que analiza la secuencia de tokens o elementos léxicos producida por un analizador léxico, para determinar su estructura gramatical, agrupando los elementos léxicos en conjuntos mayores según las

reglas de la gramática, ambos se desarrollan en Python. Para esta implementación, solo se definen en el lexer, las indicaciones de inicio, mientras, si y asignaciones, como se ve en la Figura 10.

```
def lexer(pseudocode):
    tokens = []
    lines = pseudocode.strip().split('\n')
    for line in lines:
        line = line.strip()
        if line.startswith('INICIO'):
            tokens.append(('INICIO', 'INICIO'))
        elif line.startswith('MIENTRAS'):
            tokens.append(('MIENTRAS', 'MIENTRAS'))
        elif line.startswith('HACER'):
            tokens.append(('HACER', 'HACER'))
        elif line.startswith('FIN_MIENTRAS'):
            tokens.append(('FIN_MIENTRAS', 'FIN_MIENTRAS'))
        elif line.startswith('SI'):
            tokens.append(('SI', 'SI'))
        elif line.startswith('ENTONCES'):
            tokens.append(('ENTONCES', 'ENTONCES'))
        elif line.startswith('FIN_SI'):
            tokens.append(('FIN_SI', 'FIN_SI'))
        elif ':-' in line:
            tokens.append(('ASIGNACION', line))
        elif line.endswith(';'):
            tokens.append(('FUNCION', line))
        elif line.startswith('FIN'):
            tokens.append(('FIN', 'FIN'))
        else:
            continue
    return tokens
```

Fig. 10. Código para la definición del Lexer

El parser está diseñado para analizar los tokens y poder realimentar si la semántica del código no es correcta, como ejemplo de este proceso observamos la Figura 11, donde tenemos la función para evaluar la estructura comparativa SI.

```
def si(self):
    self.pos += 1
    if self.pos < len(self.tokens) and self.tokens[self.pos][0] == 'ENTONCES':
        self.pos += 1
        while self.pos < len(self.tokens) and self.tokens[self.pos][0] not in ['FIN_SI', 'FIN_MIENTRAS', 'FIN']:
            if self.tokens[self.pos][0] == 'FUNCION' or self.tokens[self.pos][0] == 'ASIGNACION':
                self.pos += 1
            else:
                self.error(f"Token inesperado en si: {self.tokens[self.pos][1]}")
        if self.pos < len(self.tokens) and self.tokens[self.pos][0] == 'FIN_SI':
            self.pos += 1
        else:
            self.error("Se esperaba 'FIN_SI' al final del bloque SI")
    else:
        self.error("Se esperaba 'ENTONCES' después de la condición SI")
```

Fig. 11. Parser para la estructura comparativa SI

La interfaz grafica tiene dos apartados, en el primero se coloca el pseudocódigo, el cual es evaluado por el parser y en caso de ser correcto, se llama a la función traductora, la cual transforma el pseudocódigo al lenguaje Arduino. El cual ya incluye para esta aplicación, las librerías para el control de los motores y la lectura a través del bluetooth, del cual esperará recibir las instrucciones ADELANTE, ATRÁS, IZQUIERDA, DERECHA. Como podemos observarlo en la Figura 12.

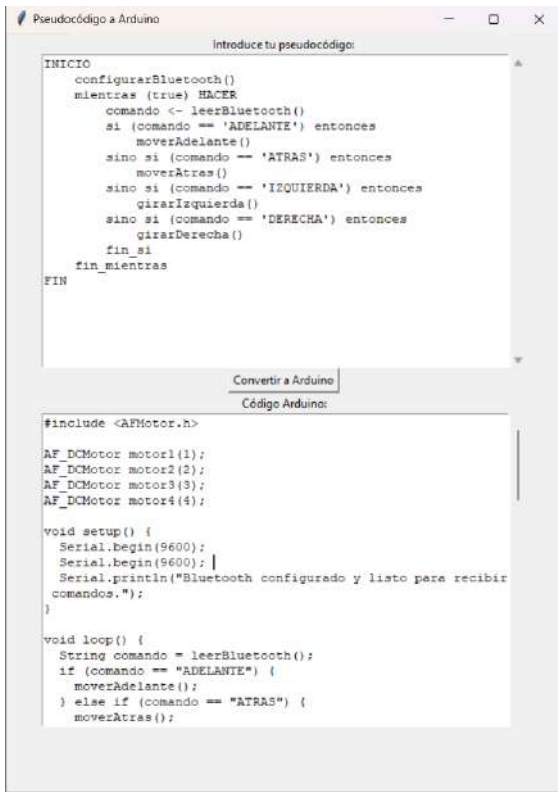


Fig. 12. Interface de la aplicación que evaluó un código correcto

En caso de tener un error en el pseudocódigo en la ventana de traductor, se coloca el mensaje que indica el error en el pseudocódigo, para que el usuario pueda verificar su código y buscar corregir el error.

En la figura 13, se observa ver la interfaz gráfica, en la cual se evalúa un pseudocódigo correcto y como salida, se tiene la traducción a un código funcional, en la figura 14, podemos ver el mensaje que se emite cuando hay un error en la codificación del pseudocódigo.

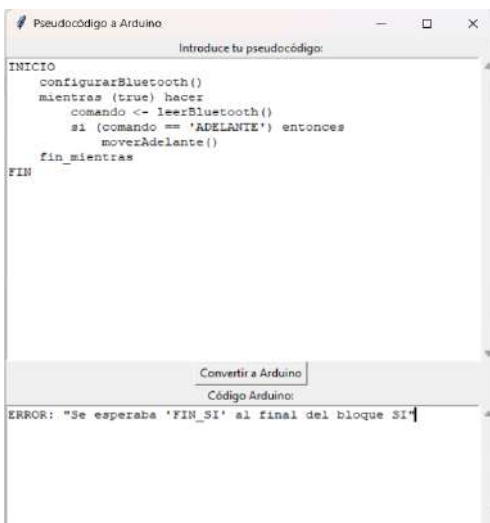


Fig. 13. Interface con un error en el pseudocódigo al no incluir el FIN\_SI

El proceso de traducción se da a través de una función que incorpora las partes fundamentales de la estructura de un programa en Arduino como lo es el Setup y el Loop, e introduce las instrucciones traducidas de la evaluación del pseudocódigo, como podemos verlo en la Figura 14.

```

1 def pseudocode_to_arduino(pseudocode):
2     tokens = lexer(pseudocode)
3     parser = Parser(tokens)
4     try:
5         parser.parse()
6     except Exception as e:
7         return str(e)
8     arduino_code = []
9     arduino_code.append("#include <AFMotor.h>")
10    arduino_code.append("")
11    arduino_code.append("AF_DCMotor motor1(1);")
12    arduino_code.append("AF_DCMotor motor2(2);")
13    arduino_code.append("AF_DCMotor motor3(3);")
14    arduino_code.append("AF_DCMotor motor4(4);")
15    arduino_code.append("")
16    arduino_code.append("void setup() {")
17    arduino_code.append("  Serial.begin(9600);")
18    for token_type, token_value in tokens:
19        if token_type == "FUNCION" and 'configurarbluetooth' in token_value:
20            arduino_code.append("  // Configurar Bluetooth")
21            arduino_code.append("")
22            arduino_code.append("void loop() {")
23        elif token_type == "INICIO":
24            arduino_code.append("  // INICIO DEL PROGRAMA")
25        elif token_type == "MIENTRAS":
26            arduino_code.append("  while (true) {")
27        elif token_type == "HACER":
28            arduino_code.append("    // HACER")
29            arduino_code.append("  }")
30        elif token_type == "FIN_MIENTRAS":
31            arduino_code.append("  }")
32        elif token_type == "SI":
33            arduino_code.append("    if (")
34        elif token_type == "ENTONCES":
35            arduino_code.append("    {")
36        elif token_type == "SINO":
37            arduino_code.append("    } else {")
38        elif token_type == "FIN_SI":
39            arduino_code.append("    }")
40        elif token_type == "FUNCION":
41            func_name = token_value.split('(')[0]
42            if func_name in ['moveradelante', 'moveratras', 'girarderecha', 'girarizquierda']:
43                arduino_code.append("      " + token_value + ";")
44        elif token_type == "ASIGNACION":
45            arduino_code.append("      " + token_value.replace('<-;', '=');")
46        elif token_type == "FIN":
47            arduino_code.append("  // FIN DEL PROGRAMA")
48    arduino_code.append("}")
49    return "\n".join(arduino_code)
50
    
```

Fig. 14. Código que traduce el código Robot a código Arduino

## V. CONCLUSIONES

La integración de la robótica y el pseudocódigo en la enseñanza de la programación facilita una comprensión más profunda y práctica de conceptos fundamentales como condicionales y ciclos. La robótica proporciona una plataforma tangible para aplicar estos conceptos, mientras que el pseudocódigo permite a los estudiantes enfocarse en la lógica algorítmica sin distraerse con la sintaxis específica de un lenguaje de programación.

El pseudocódigo, al ofrecer una representación simplificada de los algoritmos, permite a los estudiantes concentrarse en la estructura y lógica de sus programas. Esto es especialmente útil para principiantes que pueden sentirse abrumados por la sintaxis de los lenguajes de programación tradicionales. El empleo de esta tecnología es complementaria a herramientas como Raptor, que utilizan diagramas de flujo para representar el pseudocódigo, y buscamos que complementen el proceso de enseñanza a través de proporcionar una interfaz visual que facilita la comprensión de conceptos algorítmicos.

Este proceso ha sido implementado durante el ciclo escolar 2023-2024, a los estudiantes de la asignatura de fundamentos de programación, en este ciclo se tuvieron dos grupos de nuevo ingreso, en uno se implemento esta metodología, y en otro se siguió un proceso tradicional, bajo esta metodología se tuvo una deserción del 5% y una reprobación del 18%, mientras que en el

grupo donde se siguió una metodología tradicional se tuvo una deserción del 12% y una reprobación del 40%, porcentajes acordes al historial de la asignatura en el programa educativo, como se observa en la figura 15.

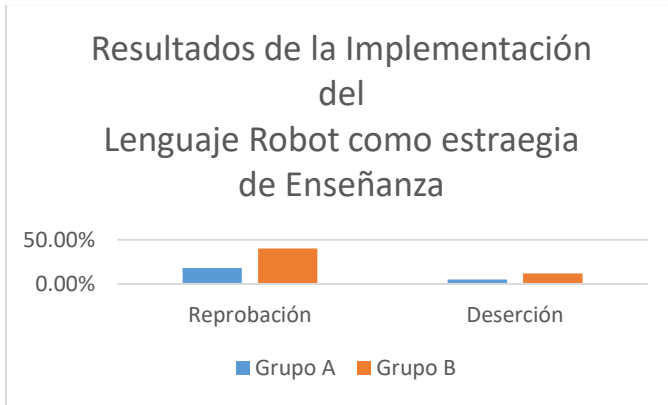


Fig. 15. Gráfica de resultados de implementación de la metodología

Derivado de estos resultados, se implementará la metodología durante el periodo 2024-2025 en los grupos de nuevo ingreso, puesto que en primer semestre se oferta la asignatura de fundamentos de programación, el trabajo futuro contempla seguir evaluando la eficiencia en la implementación de este proceso, para obtener datos que nos permitan tener mayor soporte de las ventajas de implementar esta metodología.

#### REFERENCES

- [1] Fagin, B. (2001). Using robots to teach Java. *ACM SIGCSE Bulletin*, 33(3), 92-96.
- [2] Guzdial, M., & Soloway, E. (2002). Teaching the Nintendo generation to program. *Communications of the ACM*, 45(4), 17-21.
- [3] McConnell, S. (2004). *Code Complete*. 2nd ed. Redmond: Microsoft Press.
- [4] Carlisle, M. C., Wilson, T. A., Humphries, J. W., & Hadfield, S. M. (2005). RAPTOR: a visual programming environment for teaching algorithmic problem solving. *ACM SIGCSE Bulletin*, 37(1), 176-180.
- [5] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- [6] Chambers, J. M., Carbonaro, M., & Rex, M. (2007). Scaffolding knowledge construction through robotic technology: A middle school case study. *Electronic Journal for the Integration of Technology in Education*, 6, 55-70.
- [7] Eguchi, A. (2014). Educational Robotics for Promoting 21st Century Skills. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 8(1), 5-11.

# Implementación de proyectos de robótica como estrategia didáctica enfocada en reforzar los aprendizajes específicos propuestos por ABET

Gerardo I. Pérez-Soto  
Facultad de Ingeniería

Universidad Autónoma de Querétaro  
Santiago de Querétaro, México  
israel.perez@uaq.mx

Carlos G. Manríquez-Padilla  
Facultad de Ingeniería

Universidad Autónoma de Querétaro  
San Juan del Río, México  
carlos.manriquez@uaq.mx

Juan José Saucedo-Dorantes  
Facultad de Ingeniería

Universidad Autónoma de Querétaro  
San Juan del Río, México  
juan.saucedo@uaq.mx

José Villagomez-Moreno  
Facultad de Ingeniería

Universidad Autónoma de Querétaro  
San Juan del Río, México  
jose.villagomez@uaq.mx

Karla Anhel Camarillo-Gómez

Departamento de Ingeniería Mecánica  
Tecnológico Nacional de México en Celaya  
Celaya, México  
karla.camarillo@itcelaya.edu.mx

**Abstract**—En este trabajo se presenta una estrategia didáctica enfocada en la integración de las habilidades y conocimientos descritos en los objetivos de aprendizaje sugeridos por el Consejo de Acreditación para Ingeniería y Tecnología, ABET. La estrategia didáctica propuesta consiste en asignar al alumno el desarrollo de una solución a un problema real de ingeniería considerando criterios técnicos, éticos, sociales y ambientales; posteriormente, el alumno debe implementar la solución mediante el uso de un robot manipulador serial de seis grados de libertad y plasmar los resultados mediante un reporte técnico. La estrategia didáctica propuesta busca una formación integral del alumno que proporcione tanto aprendizaje significativo como habilidades y capacidades para resolver problemas reales presentando la solución mediante reporte técnico escrito y oral. Los resultados, tras aplicar la estrategia durante dos años consecutivos, muestran una notable mejoría en habilidades técnicas y en competencias blandas como lo es el trabajo en equipo, manejo de estrés, comunicación oral entre otras, garantizando que los objetivos de aprendizaje del ABET sean alcanzados.

## INTRODUCCIÓN

La calidad de la información y el desarrollo de nuevas tecnologías son herramientas adicionales para transmitir conocimientos a los alumnos en esta era. Los métodos de enseñanza siguen siendo los componentes principales en el proceso educativo porque permiten la conceptualización del entorno desde un punto de vista científico; además, se han usado desde el inicio de la educación a la fecha y han ido evolucionando con los avances tecnológicos [1], [2], [3].

Las teorías de aprendizaje de Piaget y Vygotsky establecen que la práctica facilita el proceso de aprendizaje ya que permite tener una experiencia que relaciona un nuevo conocimiento con respecto a uno previo [4]. Actualmente, la tendencia es que se desarrollen los conocimientos objetivo en conjunto con las habilidades blandas [3], [5], [6].

En México, la mayoría de las instituciones de educación superior están sujetas a un proceso de acreditación que garantice

la calidad de la institución o de un programa educativo. El programa educativo de Ingeniería Electromecánica de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro, Campus San Juan del Río ha estado acreditada por el Consejo de Acreditación para Ingeniería y Tecnología, ABET (por sus siglas en inglés, Accreditation Board for Engineering and Technology) desde 2016 [7].

Combinando los objetivos de aprendizaje de ABET con las teorías de aprendizaje y el acceso a nuevas tecnologías e información, se pueden establecer estrategias didácticas que impacten positivamente en la formación de profesionistas aptos para el entorno laboral [8], [9], [5].

Analizando algunos trabajos previos, Arocena *et al.* en [10] se da un contexto de sociedad digital, en donde las tecnologías de información y comunicación tienen un gran impacto en la educación, y las palabras clave de la revisión son específicamente la presencia de la robótica en la educación, representando el desarrollo de habilidad individual en el alumno. Sumado a esta la revisión, Marcos Pablos *et al.* [11] presentan una revisión de robots didácticos para la educación de futuros profesionales en ciencias de la salud, remarcando la necesidad de integrarlos en la formación de los mismos para crear en un corto o mediano plazo programas de salud que integren dichas necesidades. También otros trabajos enfatizan que la integración de la robótica como herramienta tecnológica de aprendizaje ayuda a preparar a los alumnos para el futuro [12], [13], [14].

Por otro lado, Ekström y Pareto en [15] presentan una estrategia didáctica en donde el alumno pueda aprender enseñando, asignándole el rol de tutor sobre un robot, y como resultado se presenta la conceptualización de diseño para futuros robots. En [16], se refuerza el punto de vista de Ekström y Pareto, pues se aplica la estrategia antes mencionada a futuros profesores de

informática, y cuyo resultado fue el aprendizaje de actividades prácticas y entendimiento de nuevas situaciones y problemas. En [17] y [18], se deduce que la integración de la robótica en la educación alienta a los alumnos a estudiar y aprender más.

Además, los estándares de acreditación para los programas educativos de nivel licenciatura, son parte importante del panorama de los requisitos que los profesionistas deben tener para adaptarse, así como también representan la adaptabilidad del programa en sus respectivos campos [19].

El diseño de técnicas de enseñanza, estrategias didácticas y los estándares del programa educativo, son un factor importante para el impacto de aprendizaje del alumno y los resultados del programa en sí [20].

En este documento se proponen estrategias didácticas para lograr un aprendizaje significativo de las materias que conforman el plan de estudios y, en conjunto, se busca desarrollar habilidades y competencias que les facilitarán la integración al mercado laboral. Lo anterior, también significa que se garantizarán los indicadores que establece ABET.

#### DESCRIPCIÓN DE LOS RESULTADOS DE APRENDIZAJE DEL ESTUDIANTE

Para poder hacer frente a los nuevos desafíos que trae consigo la industria 4.0 y la rápida evolución de la tecnología, la carrera de Ingeniería Electromecánica, ofertada por la Universidad Autónoma de Querétaro, definió un perfil de egreso para el estudiante compuesto por cinco capacidades técnicas (PEO), las cuales son:

- 1) Tener la capacidad de aplicar sus conocimientos matemáticos, científicos y técnicos para identificar y resolver problemas de ingeniería.
- 2) Tener la capacidad de diseñar elementos mecánicos y sistemas para transformar y ensamblar la materia, proporcionar soluciones, mejoras y mantenimiento a los sectores de automoción, ferrocarril, medicina y tecnología.
- 3) Tener la capacidad de comunicarse con claridad y trabajar en equipos multidisciplinarios.
- 4) Tener la capacidad de desarrollar experimentos e interpretar sus resultados basados en ingeniería asistida por ordenador.
- 5) Tener la capacidad de integrar nuevas herramientas en la práctica de la ingeniería.

Para lograr estas capacidades, la agencia de acreditaciones internacionales ABET, estipula como uno de los criterios de acreditación que se debe definir las habilidades, conocimientos y comportamientos que el estudiante deberá adquirir a medida que avanza en el programa educativo [7]. Para ello, se definieron siete objetivos de aprendizaje (SO, por sus siglas en inglés), listados a continuación:

- 1) Los estudiantes deben demostrar la capacidad de identificar, formular y resolver problemas complejos de

ingeniería utilizando principios de ingeniería, ciencias y matemáticas.

- 2) Los estudiantes aplican el diseño de ingeniería para producir soluciones que satisfagan necesidades específicas teniendo en cuenta la salud pública, la seguridad y el bienestar, así como factores globales, culturales, sociales, ambientales y económicos.
- 3) Capacidad para comunicarse eficazmente con una variedad de audiencias.
- 4) Capacidad para reconocer responsabilidades éticas y profesionales en situaciones de ingeniería y emitir juicios informados, que deben considerar el impacto de las soluciones de ingeniería en contextos globales, económicos, ambientales y sociales.
- 5) La capacidad de funcionar eficazmente en un equipo cuyos miembros juntos brindan liderazgo, crean un entorno colaborativo e inclusivo, establecen metas, planifican tareas y cumplen metas.
- 6) La capacidad de desarrollar y realizar experimentaciones adecuadas, analizar e interpretar datos y utilizar el criterio de ingeniería para sacar conclusiones.
- 7) La capacidad de adquirir y aplicar nuevos conocimientos según sea necesario, utilizando estrategias de aprendizaje adecuadas.

Los SO antes mencionados ayudan a describir lo que los estudiantes deben de saber y deben de ser capaces de hacer al momento de graduarse. El estudiante al lograr un criterio de desempeño avanzado en los siete SO desarrolla las capacidades técnicas descritas en los cinco PEO. La relación que guardan los SO con los PEO se describe en la Tabla I.

TABLA I  
RELACIÓN DE LOS SO CON LOS PEO

PEO	SO1	SO2	SO3	SO4	SO5	SO6	SO7
1	X	X		X		X	X
2	X	X				X	X
3			X		X		
4	X					X	X
5	X					X	X

Por lo anterior, se puede decir que los estudiantes al desarrollar los SO adquirirán una serie de capacidades que les permitirán desempeñarse de manera adecuada en el ámbito industrial. Por ello, es importante desarrollar estrategias didácticas que promuevan la inclusión de las habilidades descritas en los siete SO.

#### DESCRIPCIÓN DE LA ESTRATEGIA DIDÁCTICA

La estrategia didáctica propuesta en este trabajo tiene como objetivo integrar todas las habilidades y conocimientos descritos en los SO para resolver un problema de ingeniería dentro del área de la robótica. Para ello, se utiliza como base de la estrategia didáctica la generación de trayectorias y el diseño tanto de órganos terminales (OT) como de equipo periférico (EP) para un robot industrial de seis grados de libertad.

La estrategia didáctica propuesta consiste en desarrollar un caso de estudio donde el alumno tendrá que proponer una solución a un problema práctico de ingeniería cumpliendo una serie de requisitos técnicos, de entre los cuales destacan: la adecuada selección de materiales y procesos de manufactura, restricciones de tiempo y masa, apego a normatividad vigente, criterios de seguridad entre otros. Una vez diseñada la solución, el alumno deberá validar el adecuado funcionamiento de la misma al desarrollar un prototipo funcional y probarlo en laboratorio bajo condiciones controladas. Posteriormente, el alumno deberá emitir un juicio de valor mediante un informe técnico y comunicará los resultados a una audiencia especializada mediante una presentación oral. Este proceso se describe en la Figura 1.

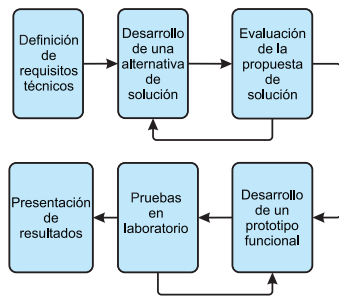


Fig. 1. Descripción del desarrollo de la estrategia didáctica.

La parte fundamental de la estrategia se centra en la integración de un robot manipulador industrial comercial de seis grados de libertad, y la generación de trayectorias mediante el controlador comercial con que viene equipado el robot, al diseño de la solución del problema en cuestión. Para este trabajo, se utilizó el robot manipulador serial de seis grados de libertad Yaskawa MH5S II equipado con un controlador MOTOMAN DX200 mostrado en la Figura 2.



Fig. 2. Robot industrial empleado en el desarrollo de la estrategia didáctica propuesta.

Como parte del diseño de la estrategia didáctica, se definieron diversos rubros de evaluación, los cuales, están ligados a los SO y permiten tanto desarrollar en el estudiante las habilidades necesarias para fortalecer los PEOs, como ayudar al docente a definir los criterios técnicos necesarios para delimitar el desarrollo del caso de estudio. Es importante recalcar que el

caso de estudio debe buscar resolver una problemática real presentada ya sea en el ámbito industrial o en la vida diaria, tomando en consideración que:

- 1) El caso de estudio debe poder dividirse en dos etapas; La primera enfocada al diseño y simulación tanto del OT como de los EP, y la segunda enfocada en la fabricación del prototipo funcional y en el diseño de trayectorias para cumplir con la tarea descrita en el caso de estudio.
- 2) La complejidad de la tarea descrita debe ser tal que pueda resolverse a lo largo del ciclo escolar.
- 3) Debe promover el trabajo colaborativo y el pensamiento analítico.
- 4) Debe promover la integración de tecnologías emergentes para el diseño y fabricación del OT y EP.
- 5) Debe considerar como criterio fijo la integración del robot industrial antes mencionado.
- 6) Los alumnos deben aplicar conocimientos de diversas áreas de la ingeniería para formular la solución al problema.

De manera general, los rubros de evaluación y la relación que guardan con los SO se muestra en la Tabla II.

Como se muestra en la Tabla II, tanto los rubros como los criterios de evaluación se diseñaron de tal forma que promovieran las habilidades, destrezas y capacidades descritas en los siete SO, con lo que se garantiza que la estrategia didáctica tenga un impacto positivo en el desarrollo de las capacidades descritas en los PEO del programa educativo.

#### IMPLEMENTACIÓN DE LA ESTRATEGIA DIDÁCTICA

Para mostrar la efectividad de la estrategia didáctica propuesta en este trabajo, se dará una descripción detallada de la implementación de la misma en un grupo de ocho alumnos de licenciatura, cursando el penúltimo semestre marcado en su plan de estudios, durante un periodo escolar compuesto por 17 semanas.

Los alumnos se integraron en equipos de cuatro personas, cada equipo trabajó en el desarrollo de un sistema de fabricación de empaques de uretano para fijar parabrisas automotrices, el cual, debía fabricar un empaque de uretano en menos de 60 segundos sobre el contorno, marcado en color amarillo, de la sección de parabrisas mostrado en la Figura 3.

Como especificaciones de diseño, los alumnos deben cumplir con las restricciones siguientes:

- Las masa máxima del OT no debe sobre pasar los 500 g.
- Debe poderse fijar al robot mediante cuatro tornillos M5.
- La alimentación eléctrica o neumática del OT y EP debe de ser externa y no puede estar dentro del área de trabajo del robot.
- El EP unicamente debe de sostener la sección de parabrisas.
- La posición inicial y final del parabrisas debe de ser constante.

TABLA II  
RELACIÓN DE LOS SO CON LOS RUBROS DE EVALUACIÓN.

Rubro	Criterio de evaluación	SO
Trabajo en equipo	El alumno muestra evidencia del trabajo en equipo mediante la asignación de roles y responsabilidades en el desarrollo del caso de estudio.	5
Implementación física	El alumno muestra evidencia tanto de la implementación física adecuada de la solución propuesta, como de la aplicación de nuevo conocimiento para recolectar y analizar datos provenientes de la implementación de la solución para formular un juicio de valor.	6, 7
Prototipo funcional	El alumno muestra evidencia tanto de la aplicación de los principios básicos de ingeniería, ciencias y matemáticas, como de nuevo conocimiento e implementación de tecnologías de manufactura emergentes en el diseño de un prototipo funcional que cumpla con especificaciones técnicas, normas de seguridad e higiene, así como con las consideraciones éticas, económicas y sociales vigentes.	1, 2, 4, 7
Informe técnico	El alumno tiene la habilidad de plasmar los resultados provenientes del diseño y aplicación de la solución al caso de estudio analizado, acompañado de un juicio de valor sobre los resultados obtenidos en un reporte escrito empleado vocabulario técnico adecuado, conceptos de ingeniería y matemáticas, así como la justificación técnica de las decisiones tomadas a lo largo del diseño de la solución.	1, 3, 5
Presentación oral	El alumno comunica de forma oral los resultados provenientes del diseño y aplicación de la solución al caso de estudio analizado, empleando material audiovisual adecuado, vocabulario técnico y muestra evidencia de la repartición de roles y responsabilidades durante el desarrollo de la solución al caso de estudio analizado.	3, 5



Fig. 3. Sección de parabrisas a emplear en el caso de estudio.

- El robot y operador deben de estar protegidos de cualquier derrame de uretano.
- El robot no debe ejercer más del 60% del torque máximo permitido por articulación.
- El uretano debe de ser proporcionado de forma automática durante todo el proceso.
- El proceso de aplicación de uretano debe de cumplir con la normativa vigente.
- No se permite interacción del operador durante el proceso.

Dentro del reporte técnico, los alumnos deben plasmar:

- Introducción.
- Planteamiento del problema.
- Criterios de diseño finales y normas oficiales que deben cumplir.
- Selección y justificación de los componentes y materiales

comerciales.

- Evolución de los diseños.
- Diseño final (diagramas e imágenes) lo suficientemente claro para poder ser reproducido por terceros.
- Justificación del porque consideran que ese diseño es el que mejor resuelve el problema.
- Conclusiones y bibliografía.
- Cronograma de actividades y asignación de responsabilidades.

Para la presentación oral, los alumnos deben elaborar material audiovisual que les permita exponer el análisis y desarrollo del caso de estudio, así como los resultados obtenidos en idioma inglés. Para ello, los alumnos disponen de un máximo de 20 minutos y se les evalúa los siguientes rubros:

- Uso de vocabulario técnico.
- Manejo del escenario.
- Expresión corporal.
- Uso adecuado de los recursos audiovisuales.
- Modulación de la voz.
- Emplear un discurso adecuado al tipo de público.
- Resolución de dudas.
- Adecuada gestión del tiempo.

Las entregas de las evidencias se distribuyeron de forma que el diseño del OT y EP se tuviera a mitad del ciclo escolar, mientras que para el reporte técnico y la presentación oral se asignó como fecha límite la última semana de clases, con ello los autores consideran que los alumnos cuentan con suficiente tiempo para atender las observaciones realizadas al diseño del OT y EP, e implementar la solución propuesta.

### RESULTADOS DE LA ESTRATEGIA DIDÁCTICA

Como resultado de la implementación de la estrategia didáctica propuesta, cada equipo de alumnos presentaron e implementaron dos propuestas de solución, las cuales, cumplían con las condiciones establecidas en la evidencia. En ambos casos, los alumnos desarrollaron los mecanismos de sujeción para el parabrisas mostrados en la Figura 4.

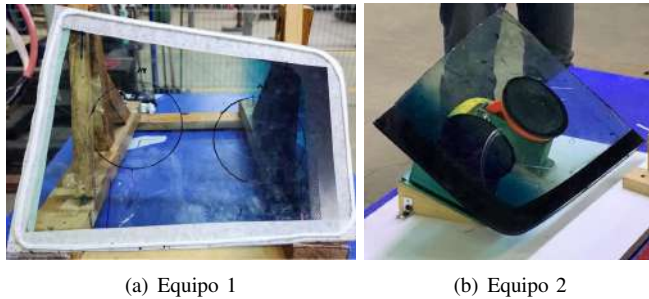


Fig. 4. Mecanismos de sujeción

Durante el diseño de los mecanismos de sujeción anteriormente mostrados, los alumnos consideraron tanto la inclinación del parabrisas, como los puntos de apoyo que este debe de tener acorde a la estrategia de generación de trayectoria utilizada. Como se muestra en la Figura 5, el equipo 1 empleó una estrategia de generación de trayectorias basada en una distribución homogénea de puntos para generar una elipse, mientras que el equipo 2 basó su estrategia de generación de trayectorias en la combinación de movimientos siguiendo una línea recta con movimientos siguiendo una función circular.

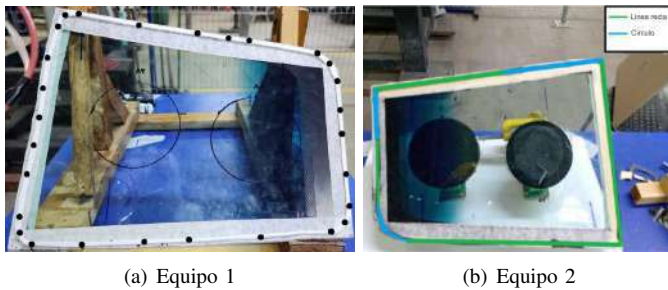
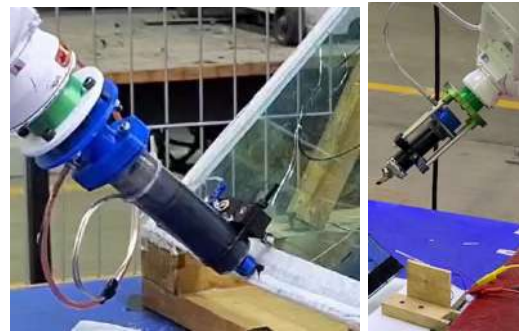


Fig. 5. Estrategia de generación de trayectorias.

Para suministrar el material para la elaboración del empaque, en la Figura 8(b) se observa que ambos equipos consideraron utilizar la inyección de uretano mediante un cilindro neumático fabricado empleando manufactura aditiva.

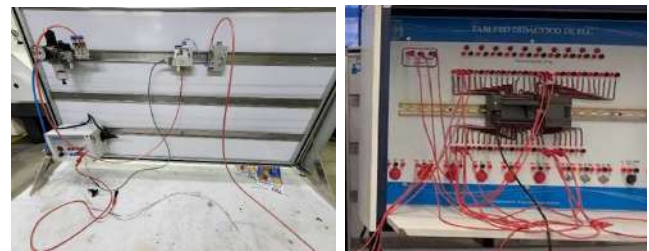
La diferencia fundamental en ambos diseños de inyector se encuentra en la posición del sensor que controla el inicio de la inyección de uretano. El equipo 1 posicionó el sensor en la punta del OT, Figura 6(a), mientras que el equipo 2 colocó el sensor en la esquina inferior derecha del parabrisas, Figura 6(b). Para la etapa de control, el equipo 1 empleó un sistema electroneumático controlado por medio de un arreglo de relevadores, Figura 8(a). Por su parte, el equipo 2 utilizó las señales digitales provenientes de un PLC para controlar



(a) Equipo 1 (b) Equipo 2

Fig. 6. Diseño del OT.

el sistema electroneumático de inyección de uretano, Figura 8(b).



(a) Equipo 1 (b) Equipo 2

Fig. 7. Etapa de control.

Como resultado de la implementación de la solución propuesta, en la Figura 8, se observa como ambos equipos lograron generar el empaque de uretano en menos del tiempo requerido, para posteriormente exponer los resultados mediante una presentación oral.



(a) Equipo 1 (b) Equipo 2

Fig. 8. Resultados de la implementación de la solución.

Tanto para la presentación oral, como para el reporte técnico detallando el desarrollo de la solución propuesta, ambos equipos mostraron evidencia suficiente del cumplimiento de todos los rubros descritos en la sección anterior.

### EVALUACIÓN DEL IMPACTO DE LA ESTRATEGIA DIDÁCTICA EN EL DESARROLLO DE LOS PEO

Para evaluar la influencia de la estrategia didáctica en el adecuado desarrollo de los PEO por parte de los estudiantes,

al inicio de cada uno de los cuatro periodos académicos en los que se implementó la estrategia didáctica propuesta se llevó a cabo una evaluación diagnóstica, la cual, consta de una serie de preguntas técnicas, tanto orales como escritas, acompañadas de un caso de estudio donde los alumnos deben de ser capaces de desarrollar una solución a un problema técnico basado en la mecánica elemental, mediante la implementación de diversas herramientas de ingeniería. La intención de la evaluación diagnóstica es medir el grado de desarrollo de las cinco capacidades técnicas (PEO) descritas en el perfil de egreso de la carrera. Como resultado de estas evaluaciones, en la Figura 9 se muestra el nivel de desarrollo para cada PEO que presentaron los estudiantes. El nivel de desarrollo está clasificado en tres grados, básico, intermedio y avanzado, cada uno de ellos marcado en color verde, amarillo y azul respectivamente. El grado de desarrollo básico se refiere a la etapa en la que el alumno muestra una comprensión fundamental de la capacidad, evidenciando conocimientos teóricos elementales y presentando un desempeño limitado al realizar tareas, por lo que requiere supervisión o asistencia. En contraste, el grado de desarrollo intermedio se caracteriza por una comprensión más profunda, permitiendo al alumno aplicar la capacidad en una amplia gama de situaciones. En este nivel, el alumno es capaz de realizar tareas de complejidad moderada de manera independiente, aunque puede necesitar apoyo en desafíos más complejos. Finalmente, el grado de desarrollo avanzado indica que el alumno posee un dominio completo de la capacidad en cuestión, lo que le proporciona las herramientas necesarias para enfrentar desafíos complejos de manera autónoma y aplicar su conocimiento en diversas situaciones. Además, en este nivel, el alumno tiene la capacidad de enseñar o guiar a otros en el desarrollo de tareas relacionadas.

Con los resultados de la evaluación diagnóstica se encontró que, en su mayoría, los alumnos cuentan con un nivel de desarrollo intermedio en los PEOs relacionados con las capacidades teórico-técnicas (Figura 9(a) y Figura 9(e)) y de elaboración (Figura 9(d)), sin embargo, presentan un rezago considerable en el desarrollo de capacidades relacionadas con la comunicación oral, Figura 9(c), y con la aplicación de las capacidades teórico-técnicas, Figura 9(b).

Al implementar la estrategia didáctica propuesta a un total de 40 alumnos repartidos en cuatro periodos académicos, cada uno compuesto por 17 semanas de clase, se logró incrementar el nivel de desarrollo de los PEO, pasando de predominar los grados de desarrollo básico e intermedio a grados de desarrollo intermedios y avanzados, como lo indica la Figura 10.

Al comparar los resultados, se puede observar que existe un incremento en el número de alumnos que alcanzaron el grado de desarrollo avanzado, mientras que se presenta un decremento en el número de alumnos que mantuvieron el grado de desarrollo básico. Para el PEO 1, el incremento de alumnos con grado de desarrollo avanzado posterior a la implementación de la estrategia didáctica aumentó 31%, de forma similar, para el PEO 2 el incremento fue del 20%, mientras que

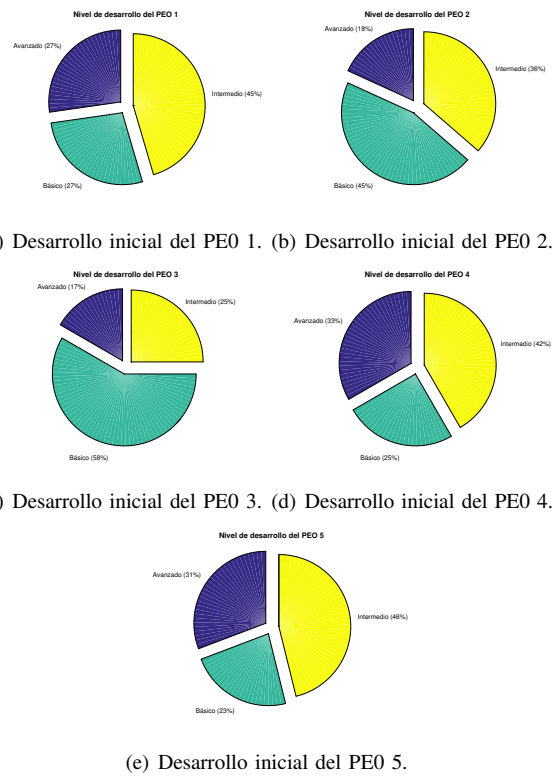


Fig. 9. Resultados de la evaluación inicial del grado de desarrollo de los PEO.

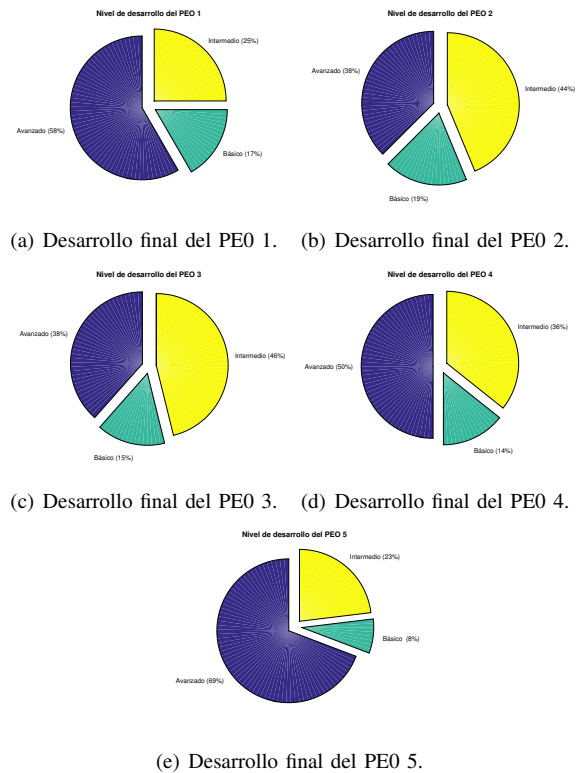


Fig. 10. Resultados de la evaluación final del grado de desarrollo de los PEO.

para el PEO 3 el aumento en el número de estudiantes alcanzó el 20%, por su parte el PEO 4 experimentó un incremento del 17%, por último, el PEO 5 presentó un incremento del 38%. Este incremento en los niveles de desarrollo no solo resalta la eficacia de la estrategia en la formación técnica de los estudiantes, sino que también sugiere una mejora en la aplicación práctica de conocimientos, lo cual es fundamental para su preparación profesional.

## CONCLUSIONES

En este trabajo se presentó una propuesta de estrategia didáctica enfocada en la integración de todas las habilidades y conocimientos descritos en los SO sugeridos por la agencia de acreditaciones internacionales ABET. La estrategia didáctica presentada se fundamenta en proporcionar al estudiante un conjunto de especificaciones vinculadas a un problema real de ingeniería en el ámbito de la robótica. El objetivo es que el alumno desarrolle una solución al problema planteado, siempre en consonancia con la normatividad vigente en la disciplina. Este enfoque implica la consideración de diversos criterios, incluidos los éticos, sociales, económicos y medioambientales, lo que fomenta una formación integral y responsable en futuros profesionales del área. Una vez desarrollada la solución, deberá implementarla con la ayuda de un robot manipulador industrial comercial de seis grados de libertad; posteriormente, el alumno debe ser capaz de plasmar los resultados en un reporte técnico y exponer los mismos mediante una presentación oral. Los principales desafíos encontrados por los estudiantes durante la implementación de la estrategia didáctica fueron; la comprensión del problema, los tipos de normatividad y regulación aplicable al entorno donde se desarrolla el problema, la identificación de limitaciones técnicas, el proceso de diseño de soluciones y el análisis de resultados, así como la visualización del impacto social, ético y medioambiental de la solución propuesta. Para superar estos desafíos, los alumnos emplearon diversos recursos, como tutoriales introductorios, mentorías interdisciplinarias ofrecidas por docentes de la facultad, lecturas complementarias sobre normativas aplicables, análisis guiados de resultados, así como herramientas de prototipado rápido y diseño asistido por computadora. Además, implementaron diversas estrategias para la adecuada gestión del tiempo.

Como resultado de la implementación de la estrategia didáctica propuesta, en los últimos cuatro periodos académicos, se ha encontrado que los alumnos muestran una notable mejoría en habilidades técnicas como lo son: la selección de materiales, la selección de proceso de manufactura, el diseño CAD enfocado a manufactura aditiva, consulta de normas técnicas, tolerancias y holguras en ensambles, procesamiento de señales, así como en la implementación de diversas estrategias de control. Además, los alumnos mostraron evidencias de un incremento en las habilidades blandas, tales como: trabajo en equipo, liderazgo, administración de tiempo, manejo de estrés, expresión oral, así como la integración de trabajo en grupos multidisciplinarios.

Por lo anterior, los autores consideran que, al implementar las estrategias didácticas presentadas en este documento, los estudiantes no solo lograrán un aprendizaje significativo al integrar los conocimientos adquiridos en las materias que conforman el plan de estudios, sino que también se familiarizarán con el uso y aplicación de diversas herramientas y tecnologías, y con la identificación de sus ventajas y desventajas al aplicarse a la solución de un problema real. Con lo cual, se desarrollan habilidades y competencias que les facilitarán la integración al mercado laboral. Lo anterior, también significa que se garantiza el cumplimiento de los objetivos de aprendizaje establecidos por ABET.

## REFERENCES

- [1] Gutiérrez Escobar, M., López Fernández, R., Yanes Seijo, R., Llerena Bermúdez, M., Rosa Rodríguez, M. N., Olano Rivalta, M., "Medios de enseñanza con nuevas tecnologías versus preparación de los docentes para utilizarlos", *Medisur*, 11 (2), 167-175, 2013.
- [2] Solórzano-Barberán, G.M., "Tecnologías de información y comunicación (TIC) en la educación, *Polo del Conocimiento*", 6 (3), 2246-2260, 2021.
- [3] Marchiori, D. M., Gouveia-Rodrigues, R., Popadiuk, S., Wagner Mainardes, E., "The relationship between human capital, information technology capability, innovativeness and organizational performance: An integrated approach", *Technological Forecasting and Social Change*, 177, 2022.
- [4] Nurhasnah, N., Nana, S., Martin, K., "Learning Theories According to Constructivism Theory", *Journal International Inspire Education Technology*, 3(1), pp. 19-30, April 2024.
- [5] Francis, A. M., "Examining industry expectations and classroom standards: A review of writing assessment in professional and technical writing classes", *Writing Pedagogy*, 14 (2), 217, 2022.
- [6] Garcés, G., Peña, C., "Adapting engineering education to BIM and industry 4.0: A view from Kolb's experiential theory in the laboratory", *Ingeniare: Revista chilena de ingeniería*, 30 (3), 497-512, 2022.
- [7] ABET, "Accreditation Policy and Procedure Manual", ABET, 2023.
- [8] Swabey, M. A., Johnson, M. C., "Satisfying ABET criterion using an industrial Microelectronic Skills Incubator", 2015 IEEE International Conference on Microelectronics Systems Education, Pittsburgh, PA, pp. 28-31, May 2015.
- [9] Williams, J. M., "Transformations in technical communication pedagogy: Engineering, writing, and the ABET engineering criteria 2000", *Technical Communication Quarterly*, 10 (2), 149-167, 2009.
- [10] Arocena, I., Huegun-Burgos, A., Rekalde-Rodríguez, I., "Robotics and Education: A Systematic Review", *TEM Journal*, 1, 379-387, 2022.
- [11] Marcos-Pablos, S., García-Peñalvo, F. J., "More than surgical tools: a systematic review of robots as didactic tools for the education of professionals in health sciences", *Advances in Health Sciences Education*, 27, 1139-1176, 2022.
- [12] Drakatos, N., Stompou, E., "The perspective of STEM education through the usage of Robotics", *World Journal of Advanced Research and Reviews*, 18 (3), 901-913, 2023.
- [13] Chatzopoulos, A., Tzerachoglou, A., Priniotakis, G., Papoutsidakis, M., Drosos, C., Symeonaki, E., "Using STEM to Educate Engineers about Sustainability: A Case Study in Mechatronics Teaching and Building a Mobile Robot Using Upcycled and Recycled Materials", *Sustainability*, 15 (21), 15187, 2023.
- [14] Collado Oporto, C. G., Zevallos Gonzales, W. F., Castillo Perdomo, A., Tejada Manrique, E. E., Carpio Rivera, M. A., "Teaching Industrial Robotics to Undergraduate Engineering Students in Latin America", *NeuroQuantology*, 20 (8), 1776-1783, 2022.

- [15] Ekström, S, Pareto, P, “The dual role of humanoid robots in education: As didactic tools and social actors”, *Education and Information Technologies*, 27, 12609-12644, 2022.
- [16] Piedade, J., Dorotea, N., Pedro, A., Matos, J. F., “On Teaching Programming Fundamentals and Computational Thinking with Educational Robotics: A Didactic Experience with Pre-Service Teachers”, *Education Sciences*, 10 (9), 214, 2020.
- [17] Wiltgen, F., “Practical Robotics as a Hands-on Tool in Teaching”, *TRANSFORMACJE*, 4 (115), 72-87, 2022.
- [18] Edgar Lopez-Caudana, Guillermo Rodríguez-Abitia, Sandra Martínez-Pérez, Paloma Anton-Ares, Maria-Soledad Ramírez-Montoya, “Scenarios of the use of robotics as a support tool for teaching”, Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality, 38-43, 2021.
- [19] Baker, F., *Assessment of mechanical engineering skills: a synthesis of industry and academic graduate level curriculum requirements*, Ph. D. Thesis, Mississippi State University, USA, 2022.
- [20] Grimm, M. J., “Design as a Feature of Biomedical Engineering Education—Satisfying ABET and Preparing Students to Address Clinical Needs”, *Journal of Biomechanical Engineering*, 142 (11), 110802, 2020.

# CAPÍTULO 5



## NOMINADOS AL PREMIO RAFAEL KELLY

# Control robusto para regulación de posición del sistema carro-péndulo afectado por perturbaciones constantes

Samir Cota

*Tecnológico Nacional de México  
Instituto Tecnológico de La Paz\**  
La Paz, B.C.S.,23080, México  
e-mail: M22310006@lapaz.tecnm.mx

Jesús Sandoval\*

*Tecnológico Nacional de México  
Instituto Tecnológico de La Paz*  
La Paz, B.C.S.,23080, México  
e-mail: jesus.sg@lapaz.tecnm.mx

Javier Moreno-Valenzuela

*Instituto Politécnico Nacional  
CITEDI*  
Tijuana, B.C.,22435, México  
e-mail: moreno@citedi.mx

Víctor Santibáñez

*Tecnológico Nacional de México  
Instituto Tecnológico de La Laguna*  
Torreón, Coahuila, 27001, México  
e-mail: vasantibanezd@lalaguna.tecnm.mx

Luis Cervantes-Pérez

*Tecnológico Nacional de México  
Instituto Tecnológico de La Laguna*  
Torreón, Coahuila, 27001, México  
e-mail: d.lecervantesp@lalaguna.tecnm.mx

César Higuera

*Tecnológico Nacional de México  
Instituto Tecnológico de La Paz*  
La Paz, B.C.S.,23080, México  
e-mail: cesar.hv@lapaz.tecnm.mx

**Resumen**—Se presenta el diseño de un controlador robusto para la regulación del sistema carro-péndulo afectado por perturbaciones constantes. El control por moldeo de energía ha sido empleado como marco teórico para el diseño del controlador propuesto, destacando la adición de un observador no lineal para tratar con la perturbación de fuerza desconocida actuando sobre el carro. El análisis de estabilidad está basado en la teoría de Lyapunov. Simulaciones numéricas con el modelo del carro-péndulo validan el desempeño satisfactorio del controlador propuesto.

**Index Terms**—Control por moldeo de energía, sistemas mecánicos subactuados, perturbaciones, observador no lineal, teoría de Lyapunov theory.

## I. INTRODUCCIÓN

Los sistemas mecánicos subactuados representan un desafío en el control automático debido a la limitación de actuadores con respecto al número de grados de libertad. Esta configuración dificulta la aplicación de técnicas convencionales de control para la regulación de esta clase de sistemas mecánicos [1], [2]. Existe una gran cantidad de trabajos que abordan diferentes estrategias para el diseño de controladores robustos, algunos de ellos son los siguientes [3]–[6], etc. Por otra parte, el método por moldeo de energía ha surgido como una poderosa herramienta para diseñar estrategias de control robusto que puedan amortiguar la presencia de perturbaciones de manera efectiva. Este enfoque se basa en utilizar una función escalar para el diseño del controlador como para el análisis de estabilidad. Existen algunos trabajos que abordan este enfoque, véase [7], [8] y [9]. A su vez, también existen trabajos de estudios comparativos entre diferentes esquemas de diseño, un ejemplo es mostrado en [10].

\*Autor de correspondencia..

El presente trabajo aborda el diseño de un controlador robusto usando el método por moldeo de energía en presencia de perturbaciones constantes desconocidas aplicado, esta metodología fue implementada al sistema subactuado carro-péndulo el cual es un sistema que ya ha sido abordado en diferentes trabajos bajo diferentes enfoques como por ejemplo [11], [12], etc. La principal contribución de este trabajo es el robustecer la ley de control diseñada en [11], mediante la implementación del observador propuesto en [13]. Se muestran las condiciones necesarias de las ganancias del controlador para asegurar que el sistema en malla cerrada garantice el rechazo a perturbaciones de fuerza constante y garantizar la estabilidad del sistema.

El resto del trabajo está organizado de la siguiente manera. En la Sección II se presenta un método de diseño de controladores para sistemas mecánicos ante perturbaciones constantes, utilizando el método de moldeo de energía. En esta sección se detalla la formulación del problema de control, un observador no lineal y la ley de control propuesta. La Sección III aborda el modelo dinámico del carro-péndulo utilizando la formulación Hamiltoniana en presencia de una perturbación constante. Se describe la implementación de la linealización parcial, se describe el diseño del controlador robusto y se realiza un análisis de estabilidad para validar el objetivo de control de regulación de posición. En la Sección IV se muestran los resultados de simulaciones numéricas que demuestran el desempeño del controlador. Finalmente, las conclusiones se presentan en la Sección V.

## II. CONTROL POR MOLDEO DE ENERGÍA DE SISTEMAS MECÁNICOS SUBACTUADOS CON PERTURBACIONES CONSTANTES

Basado en [14], se presenta una extensión de un método de diseño de controladores para sistemas mecánicos subactuados con perturbaciones de par constantes basado en la metodología de moldeo de energía. La formulación inicia con una descripción hamiltoniana del sistema mecánico a ser controlado, donde el hamiltoniano está dado por

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^T M(\mathbf{q})^{-1} \mathbf{p} + \mathcal{U}(\mathbf{q}) \quad (1)$$

donde  $\mathbf{q} \in \mathbb{R}^n$  y  $\mathbf{p} \in \mathbb{R}^n$  son los vectores de posiciones generalizadas y momentos, respectivamente,  $M(\mathbf{q}) = M(\mathbf{q})^T > 0$  es la llamada matriz de inercia. La función de energía potencial  $\mathcal{U}(\mathbf{q})$  se asume como una función al menos una vez diferenciable con respecto a  $\mathbf{q}$ . El modelo dinámico sin fricción viscosa de la clase de sistemas mecánicos subactuados considerados en este trabajo está dado por

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ -I_{n \times n} & 0_{n \times n} \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{q}} \mathcal{H}(\mathbf{q}, \mathbf{p}) \\ \nabla_{\mathbf{p}} \mathcal{H}(\mathbf{q}, \mathbf{p}) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_n \\ G[\mathbf{u} + \delta] \end{bmatrix} \quad (2)$$

donde  $G \in \mathbb{R}^{n \times m}$  es la matriz de distribución de las entradas de control, con  $m < n$  y  $\text{rango}\{G\} = m$ ,  $\mathbf{u} \in \mathbb{R}^m$  es el vector de pares de entradas de control y  $\delta \in \mathbb{R}^m$  es una perturbación constante desconocida. Continuando con la descripción del modelo, de acuerdo a la formulación hamiltoniana el momentum  $\mathbf{p}$  se define como

$$\mathbf{p} = M(\mathbf{q})\dot{\mathbf{q}} \quad (3)$$

donde  $\dot{\mathbf{q}}$  es el vector de velocidades generalizadas.

La novedad de este método consiste en proponer una malla cerrada con una estructura hamiltoniana específica. Con este fin, se define la siguiente función escalar

$$\mathcal{H}_a(\mathbf{q}_a, \mathbf{p}_a) = \frac{1}{2} \mathbf{p}_a^T M_a(\mathbf{q}_a)^{-1} \mathbf{p}_a + \mathcal{U}_a(\mathbf{q}_a) \quad (4)$$

se denomina hamiltoniano deseado, donde  $M_a(\mathbf{q}_a) \in \mathbb{R}^{n \times n}$  es una función matricial simétrica, definida positiva y diferenciable para todo  $\mathbf{q}_a \in \mathbb{R}^n$ . Además, se supone que  $\mathcal{U}_a(\mathbf{q}_a)$  es una función definida positiva localmente y continuamente diferenciable con un punto mínimo aislado en  $\mathbf{q}_a = \mathbf{0}_n$ . Se introduce la siguiente transformación de coordenadas:

$$\mathbf{q}_a = \alpha(\mathbf{q}) - \phi(t), \quad (5)$$

$$\mathbf{p}_a = M_a(\mathbf{q}_a)\dot{\mathbf{q}}_a, \quad (6)$$

siendo  $\mathbf{q}_a \in \mathbb{R}^n$  y  $\mathbf{p}_a \in \mathbb{R}^n$  los vectores de posiciones “generalizados” y momentos deseados, respectivamente. Además, los vectores  $\alpha(\mathbf{q})$  y  $\phi(t)$  tienen la estructura

$$\alpha(\mathbf{q}) = [\alpha_1(\mathbf{q}), \alpha_2(\mathbf{q}), \dots, \alpha_n(\mathbf{q})]^T, \quad (7)$$

$$\phi(t) = [\phi_1(t), \phi_2(t), \dots, \phi_n(t)]^T, \quad (8)$$

respectivamente. Las funciones  $\alpha_i(\mathbf{q})$  son continuamente diferenciables con respecto a  $\mathbf{q}$ , para  $i = 1, \dots, n$ , elegidas para

asegurar que el  $\text{rango}\{W(\mathbf{q})\} = n$ , para cualquier  $\mathbf{q} \in \mathbb{R}^n$ , tal que  $W(\mathbf{q})^{-1}$  exista, donde  $W(\mathbf{q})$  es la matriz jacobiana del vector  $\alpha(\mathbf{q})$ , es decir,

$$W(\mathbf{q}) = \frac{\partial \alpha(\mathbf{q})}{\partial \mathbf{q}}. \quad (9)$$

Además, las funciones  $\phi_i(t)$  se asumen que son dos veces diferenciables. De (5), se deduce que

$$\dot{\mathbf{q}}_a = W(\mathbf{q})\dot{\mathbf{q}} - \dot{\phi}(t). \quad (10)$$

Al sustituir  $\dot{\mathbf{q}}_a$  de (10) en (6), y  $\dot{\mathbf{q}} = M(\mathbf{q})^{-1} \mathbf{p}$  de (3), produce

$$\mathbf{p}_a = T_a(\mathbf{q}_a, \mathbf{q})\mathbf{p} - M_a(\mathbf{q}_a)\dot{\phi}(t), \quad (11)$$

donde

$$T_a(\mathbf{q}_a, \mathbf{q}) = M_a(\mathbf{q}_a)W(\mathbf{q})M(\mathbf{q})^{-1} \quad (12)$$

con  $\text{rango}\{T_a(\mathbf{q}_a, \mathbf{q})\} = n$ . Con el fin de tratar con el término  $\delta \in \mathbb{R}^m$ , de (2) se introduce la siguiente definición inspirada en [13],

$$\tilde{\delta} = \hat{\delta} - \delta, \quad (13)$$

donde  $\hat{\delta} \in \mathbb{R}^m$  es la perturbación estimada, que se calcula mediante un observador no lineal que se mostrará más adelante. Además, se define convenientemente la siguiente función escalar

$$\mathcal{S}_a(\xi_a) = \frac{1}{2} \xi_a^T \xi_a \quad (14)$$

siendo  $\xi_a \in \mathbb{R}^n$  un vector apropiado dado por

$$\xi_a = \begin{bmatrix} \tilde{\delta} \\ \mathbf{0}_{n-m} \end{bmatrix}. \quad (15)$$

La matriz de inyección de amortiguación  $\mathcal{D}_a(\mathbf{q}_a, \mathbf{p}_a) \in \mathbb{R}^{n \times n}$  es una matriz simétrica semidefinida positiva. Para simplificar, de ahora en adelante se denotará:

$$\begin{aligned} M &= M(\mathbf{q}), & M_a &= M_a(\mathbf{q}_a), & H &= H(\mathbf{q}, \mathbf{p}), \\ \mathcal{U} &= \mathcal{U}(\mathbf{q}), & \mathcal{H}_a &= \mathcal{H}_a(\mathbf{q}_a, \mathbf{p}_a), & \mathcal{U}_a &= \mathcal{U}_a(\mathbf{q}_a), \\ T_a &= T_a(\mathbf{q}_a, \mathbf{q}), & \mathcal{D}_a &= \mathcal{D}_a(\mathbf{q}_a, \mathbf{p}_a), & \mathcal{S}_a &= \mathcal{S}_a(\xi_a), \\ W &= W(\mathbf{q}), & G^\perp &= G^\perp(\mathbf{q}). \end{aligned}$$

### II-A. Formulación del problema de control

El problema de control consiste en diseñar una entrada de control  $\mathbf{u}$  que al sustituirse en el sistema de malla abierta se verifique el siguiente sistema de malla cerrada:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q}_a \\ \mathbf{p}_a \\ \xi_a \end{bmatrix} = \begin{bmatrix} 0_{n \times n} & I_{n \times n} & 0_{n \times n} \\ -I_{n \times n} & -\mathcal{D}_a & -\mathcal{A} \\ 0_{n \times n} & \mathcal{A}^T & -\mathcal{B} \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{q}_a} \mathcal{H}_a \\ \nabla_{\mathbf{p}_a} \mathcal{H}_a \\ \nabla_{\xi_a} \mathcal{S}_a \end{bmatrix} \quad (16)$$

donde  $\mathcal{H}_a$  es una función diferenciable cuya estructura fue definida en (4). Las matrices  $\mathcal{A}$  y  $\mathcal{B}$  tienen dimensión  $n \times n$ , y se definen de la siguiente manera

$$\mathcal{A} = \begin{bmatrix} T_a G & \vdots & 0_{n \times (n-m)} \end{bmatrix}, \quad (17)$$

$$\mathcal{B} = \begin{bmatrix} \epsilon_0 G^T W M^{-1} G & \vdots & 0_{n \times (n-m)} \end{bmatrix}, \quad (18)$$

siendo  $\epsilon_o$  una constante estrictamente positiva apropiada. Sustituyendo las definiciones anteriores en (16) se obtiene:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{q}_a \\ \mathbf{p}_a \\ \tilde{\delta} \end{bmatrix} = \begin{bmatrix} \nabla_{p_a} \mathcal{H}_a \\ -\nabla_{q_a} \mathcal{H}_a - \mathcal{D}_a \nabla_{p_a} \mathcal{H}_a - T_a G \tilde{\delta} \\ G^T T_a^T \nabla_{p_a} \mathcal{H}_a - \epsilon_o G^T W M^{-1} G \tilde{\delta} \end{bmatrix}. \quad (19)$$

Se puede verificar que el origen del espacio de estados  $\begin{bmatrix} \mathbf{q}_a^T & \mathbf{p}_a^T & \tilde{\delta}^T \end{bmatrix}^T = \mathbf{0}_{2n+m}$  es un punto de equilibrio aislado.

### II-B. Observador no lineal

A continuación, introducimos una selección particular de  $\hat{\delta}$  de (13) para garantizar el rechazo a perturbaciones de par constantes, de manera que

$$\hat{\delta} = G^T [\gamma + \beta(\mathbf{p}_a)], \quad (20)$$

donde  $\gamma, \beta \in \mathbb{R}^n$  son funciones vectoriales definidas a continuación. De la derivada temporal de (13) se obtiene

$$\dot{\hat{\delta}} = \dot{\delta}. \quad (21)$$

Sustituyendo la derivada temporal de (20) en (21) y sustituyendo (16), se obtiene

$$\begin{aligned} \dot{\hat{\delta}} &= G^T [\dot{\gamma} + [\nabla_{p_a} \beta] \dot{\mathbf{p}}_a], \\ &= G^T \left[ \dot{\gamma} + [\nabla_{p_a} \beta] \left[ -\nabla_{q_a} \mathcal{H}_a - \mathcal{D}_a \nabla_{p_a} \mathcal{H}_a - T_a G \tilde{\delta} \right] \right]. \end{aligned} \quad (22)$$

La selección de  $\dot{\gamma}$  y  $\beta(\mathbf{p}_a)$  son cruciales para el análisis de estabilidad del sistema de lazo cerrado deseado. Para verificar que (22) corresponde a la tercera fila de (19), la derivada temporal de la función vectorial  $\gamma$  debe satisfacer

$$\dot{\gamma} = [\nabla_{p_a} \beta] [\nabla_{q_a} \mathcal{H}_a + \mathcal{D}_a \nabla_{p_a} \mathcal{H}_a] + M^{-1} W \mathbf{p}_a. \quad (23)$$

Sustituyendo (23) en (22), obtenemos:

$$\dot{\hat{\delta}} = G^T \left[ M^{-1} W \mathbf{p}_a - [\nabla_{p_a} \beta] T_a G \tilde{\delta} \right] \quad (24)$$

Seleccionando adecuadamente la función  $\beta(\mathbf{p}_a)$  como:

$$\beta(\mathbf{p}_a) = \epsilon_o \nabla_{p_a} \mathcal{H}_a = \epsilon_o M_a^{-1} \mathbf{p}_a, \quad (25)$$

y sustituyendo (25) en (24), se obtiene

$$\dot{\hat{\delta}} = G^T M^{-1} W \mathbf{p}_a - \epsilon_o G^T W M^{-1} G \tilde{\delta} \quad (26)$$

por lo que (26) corresponde a la tercera fila de (19).

### II-C. Ley de control

La ley de control que resuelve el problema de control formulado en la sección II-A, se presenta en la siguiente Proposición.

**Proposición 1.** Considere el modelo dinámico dado por (1)-(3). Sean  $M_a, \mathcal{U}_a$  y  $\phi(t)$  sean soluciones del siguiente sistema de ecuaciones diferenciales parciales (EDP):

$$\begin{aligned} G^\perp \left[ \nabla_{\mathbf{q}} \mathcal{H} - T_a^{-1} [\nabla_{\mathbf{q}_a} \mathcal{H}_a + \mathcal{D}_a \nabla_{\mathbf{p}_a} \mathcal{H}_a] - \dot{T}_a \mathbf{p} + \dot{M}_a \dot{\phi}(t) \right. \\ \left. + M_a \ddot{\phi}(t) \right] = \mathbf{0}_{n-m}, \end{aligned} \quad (27)$$

donde  $G^\perp(\mathbf{q}) \in \mathbb{R}^{(n-m) \times n}$ , tal que se cumple la propiedad matemática  $G^\perp(\mathbf{q})G(\mathbf{q}) = \mathbf{0}_{(n-m) \times m}$ . Entonces, el sistema en malla cerrado deseado (19) se verifica con una ley de control por realimentación dada por

$$\mathbf{u} = \mathbf{u}_{es} + \mathbf{u}_{di} - \hat{\delta}, \quad (28)$$

donde

$$\mathbf{u}_{es} = [G^T G]^{-1} G^T \left[ \nabla_{\mathbf{q}} \mathcal{H} - T_a^{-1} [\nabla_{q_a} \mathcal{H}_a + \dot{T}_a \mathbf{p} - \dot{M}_a \dot{\phi}(t) - M_a \ddot{\phi}(t)] \right], \quad (29)$$

$$\mathbf{u}_{di} = -[G^T G]^{-1} G^T [T_a^{-1} \mathcal{D}_a \nabla_{p_a} \mathcal{H}_a], \quad (30)$$

$$\hat{\delta} = G^T [\gamma + \beta(\mathbf{p}_a)], \quad (31)$$

siendo  $\mathbf{u}_{es}$  y  $\mathbf{u}_{di}$  las acciones de control llamadas moldeo de energía e inyección de amortiguación, respectivamente,  $\hat{\delta}$  es la perturbación estimada, con  $\gamma$  y  $\beta$  seleccionados como en (23) y (25), respectivamente.

*Demostración.* La derivada temporal de  $\mathbf{p}_a$  a partir de (11) está dada por

$$\dot{\mathbf{p}}_a = \dot{T}_a \mathbf{p} + T_a \dot{\mathbf{p}} - \dot{M}_a \dot{\phi}(t) - M_a \ddot{\phi}(t). \quad (32)$$

Sustituyendo  $\mathbf{p}$  de (3) en (32) resulta

$$\dot{\mathbf{p}}_a = \dot{T}_a \mathbf{p} + T_a [-\nabla_{\mathbf{q}} \mathcal{H} + G[\mathbf{u} + \delta]] - \dot{M}_a \dot{\phi}(t) - M_a \ddot{\phi}(t). \quad (33)$$

Igualando  $\dot{\mathbf{p}}_a$  de (32) con la segunda fila de (19) da como resultado la siguiente ecuación:

$$\begin{aligned} T_a [-\nabla_{\mathbf{q}} \mathcal{H} + G[\mathbf{u} + \delta]] &= -\nabla_{q_a} \mathcal{H}_a - \mathcal{D}_a \nabla_{p_a} \mathcal{H}_a \\ &\quad - T_a G \tilde{\delta} - \dot{T}_a \mathbf{p} + \dot{M}_a \dot{\phi}(t) + M_a \ddot{\phi}(t) \end{aligned} \quad (34)$$

la cual puede ser escrita como

$$\begin{aligned} G[\mathbf{u} + \delta] &= [-T_a^{-1} [\nabla_{q_a} \mathcal{H}_a + \mathcal{D}_a \nabla_{p_a} \mathcal{H}_a \\ &\quad - T_a G \tilde{\delta} - \dot{T}_a \mathbf{p} + \dot{M}_a \dot{\phi}(t) + M_a \ddot{\phi}(t)] \\ &\quad + \nabla_{\mathbf{q}} \mathcal{H}]. \end{aligned} \quad (35)$$

Premultiplicando ambos lados de la ecuación (35) por la matriz invertible  $\begin{bmatrix} G^\perp \\ G^T \end{bmatrix} \in \mathbb{R}^{n \times n}$ , resulta

$$\begin{aligned} \begin{bmatrix} G^\perp \\ G^T \end{bmatrix} G[\mathbf{u} + \delta] &= \begin{bmatrix} G^\perp \\ G^T \end{bmatrix} [-T_a^{-1} [\nabla_{q_a} \mathcal{H}_a + \mathcal{D}_a \nabla_{p_a} \mathcal{H}_a \\ &\quad - T_a G \tilde{\delta} - \dot{T}_a \mathbf{p} + \dot{M}_a \dot{\phi}(t) + M_a \ddot{\phi}(t)] + \nabla_{\mathbf{q}} \mathcal{H}]. \end{aligned} \quad (36)$$

Observe que la primer fila de (36) corresponde a la EDP definida en (27). De la segunda fila de (36) se obtiene la ley de control (28) requerida para verificar (19) la cual está dada por:

$$\begin{aligned} \mathbf{u} &= [G^T G]^{-1} G^T \left[ \nabla_{\mathbf{q}} \mathcal{H} - T_a^{-1} [\nabla_{q_a} \mathcal{H}_a + \mathcal{D}_a \nabla_{p_a} \mathcal{H}_a \right. \\ &\quad \left. - T_a G \tilde{\delta} - \dot{T}_a \mathbf{p} + \dot{M}_a \dot{\phi}(t) + M_a \ddot{\phi}(t)] - \hat{\delta}, \end{aligned} \quad (37)$$

la cual corresponde a (29), donde se ha utilizado la definición (13), así como la selección de funciones (20), (23) y (25). Sin

embargo, a primera vista, no es obvio obtener directamente la segunda fila en (19) simplemente sustituyendo (37) en (2). La razón se basa en que la matriz  $G$  no es invertible, dado que por definición no es de rango completo. Por lo tanto, es necesario desarrollar (36) y encontrar soluciones en (27) tales que (37) sea válida, para lo cual se requiere que (ver (32) y (34)):

$$\dot{\mathbf{p}}_a = -\nabla_{\mathbf{q}_a} H_a - D_a \nabla_{\mathbf{p}_a} H_a - T_a G \tilde{\delta}. \quad (38)$$

A continuación, de (4), se puede obtener

$$\nabla_{\mathbf{p}_a} \mathcal{H}_a = \nabla_{\mathbf{p}_a} \left( \frac{1}{2} \mathbf{p}_a^T M_a^{-1} \mathbf{p}_a + \mathcal{U}_a \right) = M_a^{-1} \mathbf{p}_a. \quad (39)$$

Sustituyendo la definición de  $\mathbf{p}_a$  dada en (6) en (39) resulta

$$\nabla_{\mathbf{p}_a} \mathcal{H}_a = \dot{\mathbf{q}}_a. \quad (40)$$

Finalmente, concluimos que (38) y (40) corresponden a las dos primeras filas de (19). Bajo la selección de las funciones  $\hat{\delta}$ ,  $\gamma$  y  $\beta$  como (20), (23) y (25), respectivamente, la dinámica del error de estimación dado en (26) corresponde a la tercera fila de (19). Esto completa la prueba de la Proposición 1.  $\square$

#### II-D. Regulación de posición: Objetivo de control y criterio de diseño

Para regulación de posición articular, el objetivo de control es lograr que las posiciones articulares del sistema mecánico subactuado  $\mathbf{q}$  tiendan asintóticamente a la posición constante deseada admisible  $\mathbf{q}_d$ , es decir,

$$\lim_{t \rightarrow \infty} [\mathbf{q}(t) - \mathbf{q}_d] = \mathbf{0}_n, \quad (41)$$

para cualquier configuración inicial  $(\mathbf{q}(0), \mathbf{p}(0))$  del sistema mecánico subactuado que inicie desde un conjunto  $\mathcal{B} \subseteq \mathbb{R}^{2n}$ , donde el origen  $[\mathbf{q}_a^T \ \mathbf{p}_a^T]^T = [\mathbf{0}^T \ \mathbf{0}^T]^T$ , sea el único equilibrio. Por otra parte, como criterio de diseño en este trabajo se tiene interés en verificar lo siguiente:

$$\lim_{t \rightarrow \infty} [\hat{\delta}(t) - \delta] = \mathbf{0}_m. \quad (42)$$

### III. SISTEMA CARRO-PÉNDULO

El sistema carro-péndulo es un mecanismo subactuado constituido por un carro que se desplaza sobre una superficie plana con un poste fijado mediante un pivote sin fricción sobre el carro (Figura 1). Este mecanismo es subactuado porque tiene dos grados de libertad, la rotación  $q_1$  del poste y el desplazamiento horizontal  $q_2$  del carro y solo tiene un actuador, la fuerza externa  $F$  aplicada al carro.

#### III-A. Modelo dinámico

En ausencia de fricción, el modelo dinámico del sistema carro-péndulo afectado por una perturbación constante  $\delta$  está dado por

$$m_p l_p^2 \ddot{q}_1 + m_p l_p \cos(q_1) \ddot{q}_2 - m_p g l_p \sin(q_1) = 0, \quad (43)$$

$$m_p l_p \cos(q_1) \ddot{q}_1 + [M_c + m_p] \ddot{q}_2 - m_p l_p \sin(q_1) \dot{q}_1^2 = F + \delta, \quad (44)$$

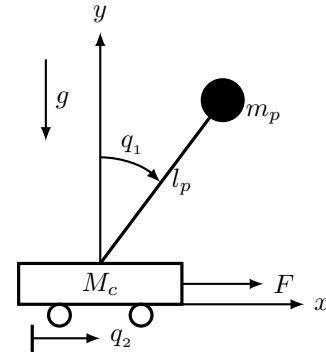


Figura 1: Diagrama esquemático del sistema carro-péndulo.

donde  $M_c$  denota la masa del carro,  $l_p$  es la longitud del péndulo y  $m_p$  su masa,  $F \in \mathbb{R}$  es la entrada de control de fuerza, y  $g$  es la aceleración de la gravedad.

El modelo dinámico del carro péndulo puede ser expresado en términos de las ecuaciones de Euler-Lagrange con

$$M = \begin{bmatrix} \alpha & \beta \cos(q_1) \\ \beta \cos(q_1) & \gamma \end{bmatrix}, \quad \mathbf{g}(\mathbf{q}) = - \begin{bmatrix} m_p g l_p \sin(q_1) \\ 0 \end{bmatrix},$$

con  $\alpha = m_p l_p^2$ ,  $\beta = m_p l_p$ ,  $\gamma = M_c + m_p$  y  $\omega = m_p g l_p$ . Para poder desarrollar este caso particular utilizando la metodología vista en la sección II, necesitamos implementar el siguiente método.

#### III-B. Linealización parcial

A continuación se utiliza la linealización parcial por realimentación mediante la entrada de control de planta  $F$  para facilitar la aplicación del marco teórico propuesto en la sección II. Aplicando el método de linealización parcial por realimentación descrito por [15] se obtiene:

$$\frac{d}{dt} \begin{bmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \frac{\omega}{\alpha} \sin(q_1) - \frac{\beta}{\alpha} \cos(q_1) \sigma \\ \sigma \end{bmatrix} \quad (45)$$

con

$$\sigma = v + \frac{\alpha \delta}{\det[M]}. \quad (46)$$

Con la entrada de control expresada como:

$$F = \left[ \frac{\det[M]}{\alpha} \right] v + \frac{\beta \omega}{\alpha} \sin(q_1) \cos(q_1) - \beta \sin(q_1) \dot{q}_1^2, \quad (47)$$

donde  $v$  es una entrada de control “auxiliar” y  $\det[M] = \alpha \gamma - \beta^2 \cos^2(q_1)$ . El modelo dinámico reducido (45) obtenido con la linealización por realimentación parcial puede escribirse mediante la estructura hamiltoniana (2) en términos del vector de estado  $[q_1 \ q_2 \ p_1 \ p_2]^T$  como se muestra a continuación:

$$\frac{d}{dt} \begin{bmatrix} q_1 \\ q_2 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \frac{\omega}{\alpha} \sin(q_1) - \frac{\beta}{\alpha} \cos(q_1) \sigma \\ \sigma \end{bmatrix}. \quad (48)$$

donde el hamiltoniano está dado por

$$\mathcal{H} = \frac{1}{2}[p_1^2 + p_2^2] + \frac{\omega}{\alpha} \cos(q_1) \quad (49)$$

y el momento  $\mathbf{p} = [p_1 \ p_2]^T$  definido en (3) resulta

$$\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \mathcal{M}\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (50)$$

con las matrices de “inercia” y “distribución” relacionadas con el modelo dinámico (45), dadas por

$$\mathcal{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} -\frac{\beta}{\alpha} \cos(q_1) \\ 1 \end{bmatrix}, \quad (51)$$

así como la nueva función de “energía potencial” definida en (49), tomada de (45):

$$\mathcal{U} = \frac{\omega}{\alpha} \cos(q_1). \quad (52)$$

### III-C. Diseño de un control robusto

Considere el modelo (48) con  $G$  dado en (51). Sean  $\alpha(\mathbf{q}) = K [q_1 \ q_2]^T$  y  $\phi(t) = K [q_{d1} \ q_{d2}]^T$  las variables seleccionadas, donde  $K = I_{2 \times 2}$  y  $\mathbf{q}_d = [q_{d1} \ q_{d2}]^T = [0 \ 0]^T$  es el vector de posiciones deseadas admisibles. Nótese que el término  $\nabla_{\mathbf{q}} \mathcal{H}$  puede expresarse como  $\nabla_{\mathbf{q}} \mathcal{U}$  dado que  $\mathcal{H}$  cuenta con la estructura vista en (49) donde la energía potencial  $\mathcal{U}$  es el único elemento que depende de  $\mathbf{q}$ . Por tanto, si definimos  $D_a$  como

$$D_a = k_v M_a G G^T M_a, \quad (53)$$

donde  $k_v$  es una constante estrictamente positiva seleccionada convenientemente, entonces la EDP de (27) se simplifica de la siguiente manera:

$$G^\perp [\nabla_{\mathbf{q}} \mathcal{U} - T_a^{-1} \nabla_{\mathbf{q}_a} \mathcal{U}_a] = \mathbf{0}_{n-m}. \quad (54)$$

Para el diseño de la ley de control  $v$  fijamos la matriz  $M_a$  constante como se muestra a continuación:

$$M_a = \begin{bmatrix} d_1 & d_2 \\ d_2 & d_3 \end{bmatrix}, \quad (55)$$

donde las constantes  $d_j$  deben seleccionarse adecuadamente para garantizar que  $M_a$  sea una matriz definida positiva, es decir,  $d_1 > 0$  y  $\det[M_a] = d_1 d_3 - d_2^2 > 0$ . El siguiente paso es definir la estructura del aniquilador izquierdo de  $G$ , el cual fue fijado como

$$G^\perp = \begin{bmatrix} 1 & \frac{\beta}{\alpha} \cos(q_1) \end{bmatrix} \quad (56)$$

el cual verifica la propiedad de las matrices  $G^\perp G = 0$ . Reemplazando (52) y (56) en (54) se obtiene la EDP simplificada:

$$c_1 \frac{\partial \mathcal{U}_a}{\partial q_{a1}} + c_2 \frac{\partial \mathcal{U}_a}{\partial q_{a2}} = -\det[M_a] \omega \sin(q_{a1}), \quad (57)$$

donde

$$c_1 = d_3 - d_2 \frac{\beta}{\alpha} \cos(q_{a1}), \quad c_2 = d_1 \frac{\beta}{\alpha} \cos(q_{a1}) - d_2.$$

De [11], la solución para la EDP mostrada en (57) se obtiene utilizando el lenguaje simbólico Matlab, y está dada por

$$\mathcal{U}_a = -\frac{\det[M_a] \omega}{d_2} \ln \left( \frac{d_2 \beta \cos(q_{a1}) - d_3 \alpha}{d_2 \beta - d_3 \alpha} \right) + \frac{1}{2} k_p z^2(q_{a1}, q_{a2}), \quad (58)$$

siendo  $k_p$  una constante arbitraria, una selección adecuada de las constantes  $d_2$  y  $d_3$  asegura  $[d_2 \beta \cos(q_{a1}) - d_3 \alpha] > 0$  y  $[d_2 \beta - d_3 \alpha] > 0$ , tal que garantizan la positividad de  $\mathcal{U}_a$ , y la función  $z(q_{a1}, q_{a2})$  resulta

$$z(q_{a1}, q_{a2}) = \frac{d_1}{d_2} q_{a1} + \mu_1 \operatorname{arctanh} \left( \mu_2 \tan \left( \frac{q_{a1}}{2} \right) \right) + q_{a2}, \quad (59)$$

donde  $\mu_1$  y  $\mu_2$  con constantes definidas como:

$$\mu_1 = \frac{2 \det[M_a] \alpha}{d_2 \sqrt{d_2^2 \beta^2 - d_3^2 \alpha^2}}, \quad (60)$$

$$\mu_2 = \frac{d_2 \beta + d_3 \alpha}{\sqrt{d_2^2 \beta^2 - d_3^2 \alpha^2}}. \quad (61)$$

Se puede comprobar que la función de energía potencial deseada  $\mathcal{U}_a$  en (58) es continua, tiene un mínimo local en  $[q_{a1} \ q_{a2}]^T = [0 \ 0]^T$  y es definida positiva si se verifican las siguientes desigualdades:

$$k_p > 0, \quad (62)$$

$$d_2 \beta > d_3 \alpha, \quad (63)$$

para todo  $|q_{a1}| < \epsilon < \frac{\pi}{2}$ , donde

$$\epsilon = a \cos \left( \frac{d_3 \alpha}{d_2 \beta} \right). \quad (64)$$

Además, a partir de (58) el gradiente de la función de energía potencial deseada  $\mathcal{U}_a$  viene dado por

$$\nabla_{\mathbf{q}_a} \mathcal{U}_a = \begin{bmatrix} \psi_1(q_{a1}) + k_p z(q_{a1}, q_{a2}) \psi_2(q_{a1}) \\ k_p z(q_{a1}, q_{a2}) \end{bmatrix} \quad (65)$$

con  $z(q_{a1}, q_{a2})$  definida en (59) y las variables  $\psi_i(q_{a1})$  en (65) están dadas por:

$$\psi_1 = \frac{\det[M_a] \omega \sin(q_{a1})}{d_2 \beta \cos(q_{a1}) - d_3 \alpha}, \quad \psi_2 = \frac{d_1 \beta \cos(q_{a1}) - d_2 \alpha}{d_2 \beta \cos(q_{a1}) - d_3 \alpha}.$$

Finalmente, tomando en cuenta (28), (45) y (46), la entrada de control auxiliar  $v$  queda

$$v = \frac{\begin{bmatrix} -\frac{\beta}{\alpha} \cos(q_{a1}) \end{bmatrix} [\det[M_a] \nabla_{q_{a1}} \mathcal{U} - d_3 \nabla_{q_{a1}} \mathcal{U}_a + d_2 \nabla_{q_{a2}} \mathcal{U}_a]}{\det[M_a] \left[ 1 + \left[ \frac{\beta \cos(q_{a1})}{\alpha} \right]^2 \right]} + \frac{d_2 \nabla_{q_{a1}} \mathcal{U}_a - d_1 \nabla_{q_{a2}} \mathcal{U}_a}{\det[M_a] \left[ 1 + \left[ \frac{\beta \cos(q_{a1})}{\alpha} \right]^2 \right]} - k_v \left[ p_{a2} - \frac{\beta}{\alpha} \cos(q_{a1}) p_{a1} \right] - \frac{\alpha \hat{\delta}}{\det[M]}, \quad (66)$$

$$\hat{\delta} = \gamma + \epsilon_0 [\Omega \dot{q}_{a1} + \dot{q}_{a2}], \quad (67)$$

$$\dot{\gamma} = \Omega [\epsilon_0 [r_1 + \rho_1] + p_{a1}] + \epsilon_0 [r_2 + \rho_2] + p_{a2}, \quad (68)$$

con

$$r_1 = [\psi_1(q_{a_1}) + k_p z(q_{a_1}, q_{a_2}) \psi_2(q_{a_1})] \frac{d_3}{\det[M_a]} - [k_p z(q_{a_1}, q_{a_2})] \frac{d_2}{\det[M_a]}, \quad (69)$$

$$r_2 = -[\psi_1(q_{a_1}) + k_p z(q_{a_1}, q_{a_2}) \psi_2(q_{a_1})] \frac{d_2}{\det[M_a]} + [k_p z(q_{a_1}, q_{a_2})] \frac{d_1}{\det[M_a]}, \quad (70)$$

$$\rho_1 = k_v [\Omega^2 p_{a_1} + \Omega p_{a_2}], \quad (71)$$

$$\rho_2 = k_v [\Omega p_{a_1} + p_{a_2}], \quad (72)$$

$$\Omega = -\frac{\beta}{\alpha} \cos(q_{a_1}). \quad (73)$$

### III-D. Análisis de estabilidad

Para obtener la dinámica de lazo cerrado, en términos de (5) y (6) y sustituimos la acción de control  $v$  de (66) en (48) y después de algunas manipulaciones algebraicas obtenemos:

$$\frac{d}{dt} \begin{bmatrix} q_{a_1} \\ q_{a_2} \\ p_{a_1} \\ p_{a_2} \\ \tilde{\delta} \end{bmatrix} = \begin{bmatrix} \frac{d_3 p_{a_1} - d_2 p_{a_2}}{\det[M_a]} \\ \frac{d_1 p_{a_1} - d_2 p_{a_2}}{\det[M_a]} \\ -\psi_1(q_{a_1}) - k_p z(q_{a_1}, q_{a_2}) \psi_2(q_{a_1}) - \mathcal{Y}_1 \\ -[d_1 \Omega + d_2] \tilde{\delta} \\ -\mathcal{Y}_2 - [d_2 \Omega + d_3] \tilde{\delta} \\ p_{a_1} \Omega + p_{a_2} - \epsilon_0 [\Omega^2 + 1] \tilde{\delta} \end{bmatrix} \quad (74)$$

donde

$$\mathcal{Y}_1 = k_v [\mathcal{X}_1(q_{a_1}) p_{a_1} + \mathcal{X}_2(q_{a_1}) p_{a_2}] - k_p z(q_{a_1}, q_{a_2}), \quad (75)$$

$$\mathcal{Y}_2 = k_v [\mathcal{X}_3(q_{a_1}) p_{a_1} + \mathcal{X}_4(q_{a_1}) p_{a_2}], \quad (76)$$

con

$$\mathcal{X}_1(q_{a_1}) = -\beta \cos(q_{a_1}) \left[ \frac{d_2 \alpha - d_1 \beta \cos(q_{a_1})}{\alpha^2} \right], \quad (77)$$

$$\mathcal{X}_2(q_{a_1}) = \frac{d_2 \alpha - d_1 \beta \cos(q_{a_1})}{\alpha}, \quad (78)$$

$$\mathcal{X}_3(q_{a_1}) = -\beta \cos(q_{a_1}) \left[ \frac{d_2 \beta \cos(q_{a_1}) - d_3 \alpha}{\alpha^2} \right], \quad (79)$$

$$\mathcal{X}_4(q_{a_1}) = \frac{d_2 \beta \cos(q_{a_1}) - d_3 \alpha}{\alpha}. \quad (80)$$

Se puede verificar que el origen  $[\mathbf{q}_a^T \ \mathbf{p}_a^T \ \tilde{\delta}]^T = [0_2^T \ 0_2^T \ 0]^T$  es el único equilibrio de (74) en el siguiente conjunto

$$\mathcal{B} = \left\{ \begin{bmatrix} \mathbf{q}_a \\ \mathbf{p}_a \\ \tilde{\delta} \end{bmatrix} \in \mathbb{R}^5 : q_{a_1} \in (-\epsilon, \epsilon) \right\} \quad (81)$$

donde  $\epsilon < \frac{\pi}{2}$ , con  $\epsilon$  definido en (64).

Dado que el sistema de lazo cerrado (74) es autónomo, la estabilidad puede demostrarse mediante el método directo de

Lyapunov y el Teorema de Barbashin-Krasovskii. A continuación, se puede seguir el procedimiento realizado en [11] para asegurar el objetivo de control (41). La función de Lyapunov para analizar (74) está dada por

$$\mathcal{V}_L = \mathcal{H}_a + \frac{1}{2} \tilde{\delta}^2, \quad (82)$$

y su derivada temporal a lo largo de las trayectorias de (74) resulta

$$\dot{\mathcal{V}}_L = -k_v [p_{a_1} \Omega + p_{a_2}]^2 - \epsilon_0 [\Omega^2 + 1] \tilde{\delta}^2. \quad (83)$$

donde  $\Omega = -\frac{\beta}{\alpha} \cos(q_{a_1})$ . Entonces, aplicando el teorema de Barbashin-Krasovskii se puede concluir que el origen es un punto de equilibrio localmente asintóticamente estable y el objetivo de control (41) se cumple localmente.

## IV. RESULTADOS DE SIMULACIÓN

Las configuraciones del sistema, así como las ganancias del controlador fueron seleccionadas como en [11] y son descritas a continuación. La configuración inicial fue:  $[q_1(0) \ q_2(0) \ p_1(0) \ p_2(0) \ \delta(0)]^T = [10^\circ \ 0 \ 0 \ 0 \ 0]^T$ . Las ganancias del controlador fueron:  $k_p = 12$ ,  $k_v = 0,3$ ,  $d_1 = 1$ ,  $d_2 = 2$  y  $d_3 = 5$ . La ganancia seleccionada para el observador que mostro mejor desempeño fue  $\epsilon_0 = 3$ . El método de integración utilizado para esta serie de simulación fue ODE23, con un error de tolerancia de  $1 \times 10^{-3}$ . La posición deseada admisible fue  $\mathbf{q}_d = [0 \ 0]^T$ , esto es, se buscó llevar el carro al origen y al péndulo a su posición vertical superior. La perturbación de fuerza constante aplicada al carro fue arbitrariamente establecida como:

$$\delta = \begin{cases} 0 \text{ [N]}, & 0 \leq t < 10 \text{ [s]}, \\ 5 \text{ [N]}, & 10 \leq t \leq 25 \text{ [s]}. \end{cases} \quad (84)$$

Se aprecia en las Figuras 2 y 3 el cumplimiento del objetivo de control (41), aún en presencia de la perturbación (84), tal y como se predijo en la teoría presentada en la sección II.

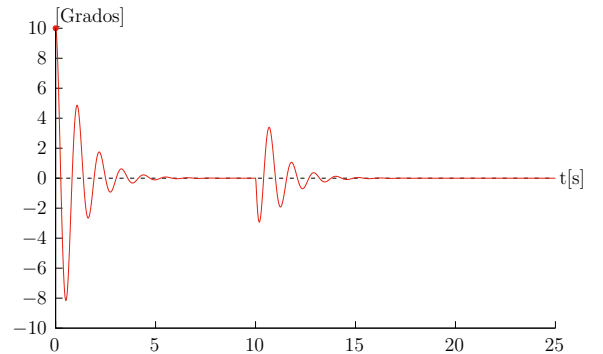


Figura 2: Evolución temporal de la posición del péndulo,  $q_1(t)$ , ante la perturbación de fuerza (84).

Por otra parte, la Figura 4 muestra la evolución temporal de la entrada de control de fuerza  $F$ . Se puede observar que después de aplicada la perturbación (84) la magnitud de la

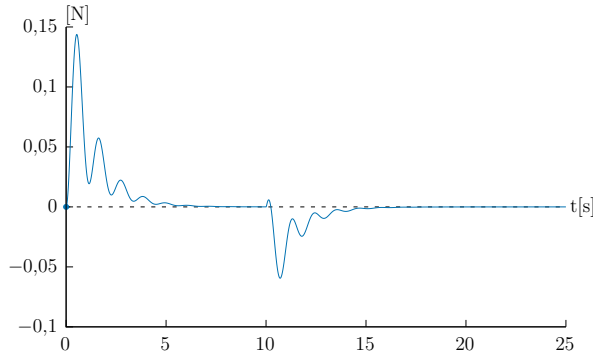


Figura 3: Evolución temporal de la posición del carro,  $q_2(t)$ , ante la perturbación de fuerza (84).

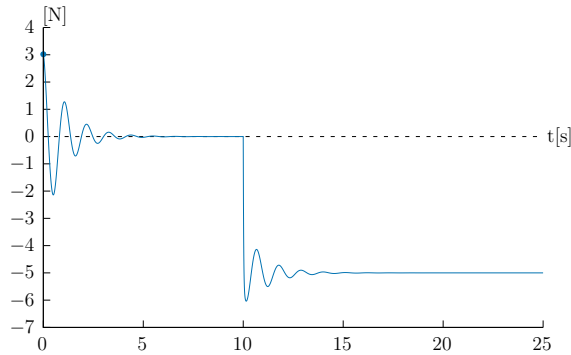


Figura 4: Señal de la entrada de control  $F(t)$  definida en (47) con  $v$  dada por (66)-(68) ante la perturbación de fuerza (84).

señal de control tiende a  $-5$  [N], esto es, tiende a compensar la perturbación con el mismo valor pero con signo opuesto.

Finalmente, una comparación de la perturbación de fuerza (84) y su estimación  $\hat{\delta}$  es mostrada en la Figura 5. Al hacer un acercamiento a la gráfica se puede verificar una rápida respuesta transitoria del observador que prácticamente ambas señales son casi iguales, tal que se cumple el criterio de diseño (42).

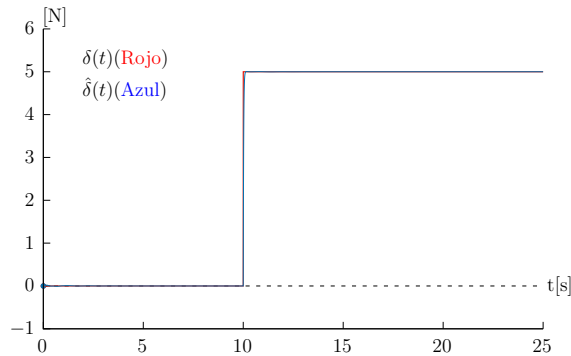


Figura 5: Comparación entre las respuestas de la perturbación de fuerza (84) y su estimación  $\hat{\delta}$  calculada vía (67)-(68).

## V. CONCLUSIONES

Se ha presentado un control robusto por moldeo de energía para el sistema carro péndulo afectado por una perturbación de fuerza constante. Se demostró el desempeño del nuevo controlador mediante simulaciones de carro péndulo utilizando MATLAB y Simulink. Lo novedoso de la propuesta del controlador radica en la inclusión de un observador no lineal que permite compensar la perturbación de fuerza constante que actúa sobre el carro.

## AGRADECIMIENTOS

Este trabajo fue parcialmente apoyado por las subvenciones CONAHCYT CVU, 1106239, Proyecto TecNM Clave 20021.24-P, Proyecto TecNM Clave 20827.24-P y Red Internacional de Control y Cómputo Aplicados (RICCA) del TecNM.

## REFERENCIAS

- [1] R. Ortega, M. W. Spong, F. Gomez-Estern, and G. Blankenstein, "Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment", *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1218–1233, 2002.
- [2] J. Moreno-Valenzuela, and C. Aguilar-Avelar, *Motion control of underactuated mechanical systems*, Springer International Publishing AG, 2018.
- [3] N.P.I. Aneke, "Control of underactuated mechanical systems", Phd Thesis, Technische Universiteit Eindhoven, 2003. <https://doi.org/10.6100/IR5595092003>.
- [4] M. Aminsafae, and M. Shafiei, "A robust approach to stabilization of 2-DOF underactuated mechanical systems", *Robotica*, vol.38, pp. 2221–2238, 2017.
- [5] A. Perrusquia, and W. Yu, "Robust control under worst-case uncertainty for unknown nonlinear systems using modified reinforcement learning", *International Journal of Robust and Nonlinear Control*, vol. 30, no. 7, pp. 2920–2936, 2020.
- [6] A. Donaire, J. G. Romero, R. Ortega, B. Siciliano, and M. Crespo, "Robust IDA-PBC for underactuated mechanical systems subject to matched disturbances", *International Journal of Robust and Nonlinear Control*, vol.27, no. 6, pp. 1000–1016, 2017.
- [7] J. G. Romero, R. Ortega, and A. Donaire, "Energy shaping of mechanical systems via PID control and extension to constant speed tracking", *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3551–3556, 2016.
- [8] J. G. Romero, A. Donaire, and R. Ortega, "Robust energy shaping control of mechanical systems", *Systems & Control Letters*, vol. 62, no. 9, pp. 770–780, 2013.
- [9] J. G. Romero, "Robust energy shaping control of nonlinear systems", Phd Thesis, Université Paris, 2013.
- [10] A. Hfaiedh, A. Chemori, and A. Abdelkrim, "Stabilization of the inertia wheel inverted pendulum by advanced IDA-PBC based controllers: Comparative study and real-time experiments", *17th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2020, pp. 753–760.
- [11] J. Sandoval, R. Kelly, and V. Santibáñez, "A speed regulator for a force-driven cart-pole system", *International Journal of Systems Science*, vol. 53, no. 2, pp. 412–430, 2022.
- [12] C. Manrique, C. Pappalardo, and D. Guida, "A parametric study of a deep reinforcement learning control system applied to the swing-up problem of the cart-pole", *Applied Sciences*, vol.10, no. 24, pp. 1–19, 2020.
- [13] J. Sandoval, L. Cervantes-Perez, V. Santibáñez, J. Moreno-Valenzuela, and R. Kelly, "A GES joint position trajectory tracking smooth controller of torque-driven robot manipulators affected by disturbances", *International Journal of Robust and Nonlinear Control*, vol.34, no. 2, pp. 1032–1053, 2024.
- [14] J. Sandoval, R. Kelly, V. Santibáñez, and J. Villalobos-Chin, "Energy regulation of torque-driven robot manipulators in joint space", *Journal of the Franklin Institute*, vol. 359, no. 4, pp. 1427–1456, 2022.
- [15] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, Munich, Germany, 1994.

# Relajación de condiciones de excitación para un controlador de seguimiento de trayectorias adaptable con realimentación de salida para robots manipuladores accionados por par

Luis Cervantes-Pérez  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de La Laguna*  
Torreón, México  
d.lecervantesp@lalaguna.tecnm.mx

Víctor Santibáñez  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de La Laguna*  
Torreón, México  
vasantibanezd@lalaguna.tecnm.mx

Jesús Sandoval  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de La Paz*  
La Paz, México  
jesus.sg@lapaz.tecnm.mx

**Abstract**—Este artículo presenta un nuevo controlador adaptable compuesto para robots manipuladores totalmente actuados accionados por par, utilizando, por un lado, una metodología de estimación de parámetros recientemente presentada en la literatura basada en una etapa de mínimos cuadrados con factor de olvido y la técnica de extensión y mezcla dinámica del regresor, y por otro lado, utilizando una forma extendida de parametrización de sistemas mecánicos basada en la ecuación de balance de potencia que permite incorporar fuerzas disipativas. Las señales del estimador son combinadas con un controlador adaptable directo con realimentación de salida, lo que permite garantizar estabilidad global uniforme asintótica del origen de un nuevo sistema en lazo cerrado. Con esto, se verifica la convergencia asintótica de los parámetros estimados a los valores reales con una condición de excitación menos estricta en un nuevo regresor. Esto último contrasta con la condición de excitación persistente que es necesaria en la versión adaptable directa del mismo controlador. Finalmente, se ilustra el desempeño del controlador propuesto mediante simulaciones numéricas de un robot manipulador de dos grados de libertad.

**Index Terms**—Estimación de parámetros, Robots manipuladores, Estabilidad, Seguimiento, Persistencia de excitación

## I. INTRODUCCIÓN

En este artículo se aborda el problema del diseño de un controlador adaptable para el seguimiento de trayectorias de robots manipuladores accionados por par basado en técnicas modernas de estimación de parámetros. Las técnicas de identificación de parámetros de sistemas dinámicos, que son la base del control adaptable, constituyen una línea de investigación que ha sido abordada desde hace varias décadas [1], [2], [3], [4], [5], [6].

Particularmente para el caso de robots manipuladores, diversas técnicas de estimación de los parámetros del modelo dinámico han sido reportadas [7], [8], [9], [10], [11]. Fundamentalmente, estos métodos de identificación se centran en las señales medibles de los sistemas para utilizar ciertas configuraciones particulares que involucran la relación entre las entradas y salidas disponibles, así como los parámetros del sistema. La formulación matemática de estas configuraciones

específicas se denomina parametrización del sistema, lo que conduce a una ecuación de regresión, ya sea lineal (ERL) o no lineal (ERNL). En términos generales, esta característica determina qué tipo de algoritmos de identificación paramétrica pueden ser implementados en cada sistema. Los robots manipuladores son una clase de sistemas dinámicos que pueden ser parametrizados mediante una ERL a través del modelo dinámico [12]. Sin embargo, el uso de esta forma de parametrización implica construir regresores con términos que demandan un alto costo computacional de cálculo.

Las técnicas clásicas de estimación de parámetros se representan mediante leyes de adaptación que proponen una dinámica deseada del error paramétrico. Las formas más utilizadas de estas son conocidas como el gradiente descendente o mínimos cuadrados, las cuales parten de una ERL y por sí mismas garantizan el acotamiento de los errores de identificación. Para lograr la convergencia a cero de los errores paramétricos utilizando estas leyes de adaptación se requiere garantizar una condición muy exigente en los regresores denominada excitación persistente (EP). En la práctica, esta condición es complicada de verificar y restringe la clase de trayectorias o sistemas en los cuales se pueden implementar estos estimadores. Por lo que las líneas actuales de investigación se centran en relajar la condición EP por condiciones menos restrictivas.

Con el fin de cumplir diferentes objetivos de control como el seguimiento de trayectorias en presencia de incertidumbres paramétricas, las leyes de adaptación pueden ser combinadas con leyes de control para garantizar condiciones de estabilidad de un sistema en lazo cerrado particular cuyos estados son las señales del error de posición y velocidad. En esta combinación, bajo una misma ley de control, cuando la ley de adaptación involucra únicamente los errores de posición y velocidad entonces se denomina controlador adaptable directo [13]. Cuando la ley de adaptación involucra únicamente al error de identificación entonces se denomina controlador adaptable indirecto [14]. Cuando se combinan tanto los errores

de seguimiento como el error de identificación en la ley de adaptación entonces se denomina controlador adaptable compuesto [11]. El presente trabajo combina un estimador de parámetros recientemente presentado en [15] en conjunto con un controlador adaptable directo con realimentación de salida presentado en [16] y [17]. Este nuevo controlador garantiza el cumplimiento de los objetivos de control de seguimiento de trayectorias y la convergencia de los parámetros estimados a los valores reales relajando las condiciones de excitación en un nuevo regresor.

En comparación con la literatura actual, las principales contribuciones del presente artículo son las siguientes:

- Se propone una extensión de la parametrización de una clase de sistemas mecánicos basado en la ecuación de balance de potencia presentada en [18], para incluir parámetros relacionados a fuerzas disipativas como la fricción.
- En comparación con el controlador adaptable directo con realimentación de salida de [16] y [17] se relajan las condiciones de excitación persistente por una condición denominada excitación por intervalo (EI) en un nuevo regresor, lo cual garantiza convergencia de los parámetros estimados a los valores reales de forma asintótica.
- Se presenta un análisis de estabilidad detallado y completo basado en la teoría de Lyapunov. En particular, se establecen claramente las condiciones en las ganancias del controlador, de modo que se demuestra que el origen de un nuevo sistema en lazo cerrado es globalmente uniformemente asintóticamente estable (GUAS) con una función de Lyapunov estricta.
- Se presentan simulaciones numéricas para validar el comportamiento y desempeño del controlador propuesto, donde las condiciones de EI son usadas para comparación con el controlador dado en [17]. Confirmándose la superioridad en la convergencia a cero del error paramétrico del esquema propuesto en contraste con el controlador dado en [17], cuya convergencia del error paramétrico no tiende a cero.

A lo largo del documento, se usará la notación  $\lambda_{\min}\{A\}$  y  $\lambda_{\max}\{A\}$  para indicar el más pequeño y más grande valor propio, respectivamente, de una matriz acotada simétrica definida positiva  $A(\mathbf{x})$  para cada  $\mathbf{x} \in \mathbb{R}^n$  (denotada por  $A > 0$ ). La notación  $A \geq 0$  significa que la matriz  $A$  es semidefinida positiva. La norma Euclidiana de un vector  $\mathbf{x}$  está definida como  $\|\mathbf{x}\| = (\mathbf{x}^T \mathbf{x})^{1/2}$ , y para una matriz  $A$  es definida como la correspondiente norma inducida  $\|A\| = (\lambda_{\max}\{A^T A\})^{1/2}$ , más aún la norma ponderada de un vector  $\mathbf{x} \in \mathbb{R}^n$  y una matriz  $A \in \mathbb{R}^{n \times n}$  se denota y define como  $\|\mathbf{x}\|_A^2 := \mathbf{x}^T A \mathbf{x}$ . También,  $(\cdot)_{n \times n}$  denota una matriz de dimensiones  $n \times n$ , con  $I_{n \times n}$  como la matriz identidad y  $0_{n \times n}$  como una matriz de ceros. Mientras que  $\mathbf{0}_n \in \mathbb{R}^n$  es un vector de ceros de dimensión  $n \times 1$ ,  $\nabla_{(\cdot)} = (\partial/\partial(\cdot))$ ,  $\det[A]$  expresa el determinante de una matriz cuadrada  $A$ , y  $\text{diag}\{a_1, a_2, \dots, a_n\}$  representa una matriz diagonal de dimensión  $n \times n$ , cuyos  $n$  elementos diagonales son  $a_1, a_2, \dots, a_n$ .

El resto del artículo está organizado de la siguiente manera. En la Sección II se presenta el estimador de parámetros propuesto así como la nueva parametrización basada en la ecuación de balance de potencia y una nueva extensión para incorporar fuerzas disipativas. La Sección III presenta un nuevo controlador adaptable compuesto que combina el estimador de parámetros propuesto y un controlador adaptable directo con realimentación de salida, se presenta además el análisis de estabilidad de un nuevo sistema en lazo cerrado y se determinan las condiciones en las ganancias del controlador que garantizan la estabilidad. En la Sección IV se presentan resultados de simulación en un brazo manipulador para demostrar el desempeño del controlador propuesto. Finalmente en la Sección V se detallan los comentarios finales sobre los resultados obtenidos.

## II. ESTIMADOR DE PARÁMETROS

En esta sección presentamos el estimador de parámetros seleccionado y la parametrización propuesta. Con este fin, basados en la propuesta de [18], partimos de una propiedad de los sistemas mecánicos denominada ecuación de balance de potencia. La construcción del regresor puede verse como una operación de filtrado de la ecuación de energía mecánica.

### A. Dinámica de los sistemas Euler-Lagrange (EL)

La obtención del regresor y la ERL respectiva se centra en sistemas EL totalmente actuados con  $n$  grados de libertad, donde las coordenadas generalizadas se representan como  $\mathbf{q} \in \mathbb{R}^n$  y la entrada de control como  $\boldsymbol{\tau} \in \mathbb{R}^n$ . La dinámica de estos sistemas se puede describir mediante las conocidas ecuaciones dadas en forma compacta por:

$$\frac{d}{dt} \left[ \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}}) \right] - \frac{\partial \mathcal{L}}{\partial \mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} - \frac{\partial \mathcal{F}(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}}, \quad (1)$$

donde  $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  define el lagrangiano como

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) := \underbrace{\frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}}}_{\kappa(\mathbf{q}, \dot{\mathbf{q}})} - \mathcal{U}(\mathbf{q}), \quad (2)$$

con  $M : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ ,  $M(\mathbf{q}) > 0$  como la matriz de inercia,  $\mathcal{K} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  es la función de energía cinética,  $\mathcal{U} : \mathbb{R}^n \rightarrow \mathbb{R}$  la función de energía potencial y  $\mathcal{F}(\dot{\mathbf{q}}) : \mathbb{R}^n \rightarrow \mathbb{R}$  es una función que representa la disipación de energía la cual puede representar los efectos de fricción. Particularmente considerando fricción viscosa en las articulaciones del sistema, la función  $\mathcal{F}$  se puede modelar como una función cuadrática, conocida como función de disipación de Rayleigh, dada por

$$\mathcal{F}(\dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathcal{R} \dot{\mathbf{q}}, \quad (3)$$

donde  $\mathcal{R} = \text{diag}\{f_{v_1}, f_{v_2}, \dots, f_{v_n}\}$  contiene los coeficientes de fricción viscosa ( $f_{v_i} \geq 0$ ) de cada una de las articulaciones. La matriz diagonal  $\mathcal{R}$  define cómo la fricción viscosa está presente o no en el robot como a continuación se describe:  $\mathcal{R} = 0_{n \times n}$  (no existe fricción viscosa en ninguna articulación del robot);  $\mathcal{R} \geq 0$  (fricción viscosa está presente parcialmente en el robot, es decir, la fricción viscosa está ausente al

menos en una articulación pero presente en las restantes); y  $\mathcal{R} > 0$  (La fricción viscosa está presente en todas las  $n$  articulaciones).

Claramente, desarrollando (1) con (2) y (3) se sigue que

$$M(\mathbf{q})\ddot{\mathbf{q}} + \dot{M}(\mathbf{q})\dot{\mathbf{q}} - \frac{1}{2} \frac{\partial}{\partial \mathbf{q}} [\dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}}] + \frac{\partial \mathcal{U}(\mathbf{q})}{\partial \mathbf{q}} + \mathcal{R}\dot{\mathbf{q}} = \boldsymbol{\tau}, \quad (4)$$

la cual puede ser expresada en la notación  $\langle M, C, g \rangle$  como:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathcal{R}\dot{\mathbf{q}} = \boldsymbol{\tau}. \quad (5)$$

**B. Derivación de una ecuación de regresión lineal basada en conceptos de energía**

Para construir la ERL sugerida se definen las siguientes parametrizaciones de la matriz de inercia  $M(\mathbf{q})$  y la energía potencial  $\mathcal{U}(\mathbf{q})$ :

$$M(\mathbf{q}) = \sum_{i=1}^l M_i(\mathbf{q})\theta_i^M, \quad \mathcal{U}(\mathbf{q}) = \sum_{j=1}^r \mathcal{U}_j(\mathbf{q})\theta_j^U, \quad (6)$$

con matrices  $M_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$  y funciones vectoriales  $\mathcal{U}_j : \mathbb{R}^n \rightarrow \mathbb{R}$ . Para incorporar directamente los efectos de las fuerzas de disipación, no necesariamente como en la forma (3), introducimos la siguiente parametrización:

$$\dot{\mathbf{q}}^T \frac{\partial \mathcal{F}(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} = \sum_{k=1}^s f_k(\dot{\mathbf{q}})\theta_k^F, \quad (7)$$

donde  $f_j : \mathbb{R}^n \rightarrow \mathbb{R}$ . Luego definiendo  $w := l + r + s \in \mathbb{R}$ , los parámetros constantes desconocidos del sistema  $\theta_i^M, \theta_j^U$  y  $\theta_k^F$  se pueden agrupar en un solo vector dado por:

$$\boldsymbol{\theta} := \text{col}(\theta_1^M, \dots, \theta_l^M, \theta_1^U, \dots, \theta_r^U, \theta_1^F, \dots, \theta_s^F) \in \mathbb{R}^w. \quad (8)$$

A continuación, presentamos una extensión de la parametrización de la ecuación de balance de potencia de [18], con el fin de incorporar términos de funciones de energía disipativos que nos permitan estimar parámetros desconocidos del sistema relacionados con la función  $\mathcal{F}(\dot{\mathbf{q}})$  de (1).

*Proposición 1:* El sistema (1) satisface la ERL:

$$\mathbf{y} = \boldsymbol{\Omega}^T \boldsymbol{\theta}, \quad (9)$$

con  $\boldsymbol{\theta}$  definida en (8),

$$\dot{\mathbf{y}} = -\lambda \mathbf{y} + \dot{\mathbf{q}}^T \boldsymbol{\tau}, \quad (10)$$

cuya realización de estados es:

$$\mathbf{y} = \left[ \frac{1}{p + \lambda} \right] \dot{\mathbf{q}}^T \boldsymbol{\tau}, \quad (11)$$

con  $\lambda > 0$  un parámetro de diseño, y un regresor dado por:

$$\dot{\boldsymbol{\Omega}} = -\lambda \boldsymbol{\Omega} + \begin{bmatrix} p \frac{1}{2} \dot{\mathbf{q}}^T M_1(\mathbf{q}) \dot{\mathbf{q}} \\ \vdots \\ p \frac{1}{2} \dot{\mathbf{q}}^T M_l(\mathbf{q}) \dot{\mathbf{q}} \\ p \mathcal{U}_1(\mathbf{q}) \\ \vdots \\ p \mathcal{U}_r(\mathbf{q}) \\ f_1(\dot{\mathbf{q}}) \\ \vdots \\ f_s(\dot{\mathbf{q}}) \end{bmatrix} \in \mathbb{R}^w, \quad (12)$$

donde  $p := \frac{d}{dt}$ .

*Prueba 1:* De [19], el sistema (1) define un operador pasivo  $\boldsymbol{\tau} \mapsto \dot{\mathbf{q}}$  con una función de almacenamiento  $\mathcal{E}(\mathbf{q}, \dot{\mathbf{q}}) := \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{U}(\mathbf{q})$ , cuya derivada temporal es:

$$\dot{\mathcal{E}}(\mathbf{q}, \dot{\mathbf{q}}) = p [\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{U}(\mathbf{q})] = p \left[ \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} + \mathcal{U}(\mathbf{q}) \right]. \quad (13)$$

Además, satisface la ecuación del balance de potencia:

$$\dot{\mathcal{E}} = \dot{\mathbf{q}}^T [\boldsymbol{\tau} - \frac{\partial \mathcal{F}(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}}], \quad (14)$$

que se puede reescribir convenientemente como

$$\dot{\mathcal{E}} + \dot{\mathbf{q}}^T \frac{\partial \mathcal{F}(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} = \dot{\mathbf{q}}^T \boldsymbol{\tau}. \quad (15)$$

A continuación, aplicando a ambos lados de (15) un filtro definido por  $H(p) := \frac{1}{p + \lambda}$ , se obtiene

$$H(p) [\dot{\mathcal{E}} + \dot{\mathbf{q}}^T \frac{\partial \mathcal{F}(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}}] = \left[ \frac{1}{p + \lambda} \right] \dot{\mathbf{q}}^T \boldsymbol{\tau} = \mathbf{y}, \quad (16)$$

donde el lado derecho de (16) es la realización de estados de (10), es decir (11). A continuación, usando la parametrización (6)-(7) y la ecuación (13) el lado izquierdo de (16) se puede escribir como:

$$\begin{aligned} H(p) \dot{\mathcal{E}} + H(p) \dot{\mathbf{q}}^T \frac{\partial \mathcal{F}(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} &= H(p) p \left[ \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} + \mathcal{U}(\mathbf{q}) \right] \\ &\quad + H(p) \left[ \dot{\mathbf{q}}^T \frac{\partial \mathcal{F}(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right], \\ &= \frac{1}{2} \sum_{i=1}^l p H(p) \dot{\mathbf{q}}^T M_i(\mathbf{q}) \dot{\mathbf{q}} \theta_i^M \\ &\quad + \sum_{j=1}^r p H(p) \mathcal{U}_j(\mathbf{q}) \theta_j^U \\ &\quad + \sum_{k=1}^s H(p) f_k(\dot{\mathbf{q}}) \theta_k^F. \end{aligned} \quad (17)$$

Igualando (17) con el lado derecho de (16) se llega a  $\mathbf{y} = \boldsymbol{\Omega}^T \boldsymbol{\theta}$ , es decir (9), donde

$$\boldsymbol{\Omega} = \begin{bmatrix} \frac{1}{2} p H(p) \dot{\mathbf{q}}^T M_1(\mathbf{q}) \dot{\mathbf{q}} \\ \vdots \\ \frac{1}{2} p H(p) \dot{\mathbf{q}}^T M_l(\mathbf{q}) \dot{\mathbf{q}} \\ p H(p) \mathcal{U}_1(\mathbf{q}) \\ \vdots \\ p H(p) \mathcal{U}_r(\mathbf{q}) \\ H(p) f_1(\dot{\mathbf{q}}) \\ \vdots \\ H(p) f_s(\dot{\mathbf{q}}) \end{bmatrix}, \quad (18)$$

lo cual conduce a:

$$p\Omega + \lambda\Omega = \begin{bmatrix} p\frac{1}{2}\dot{\mathbf{q}}^T M_1(\mathbf{q})\dot{\mathbf{q}} \\ \vdots \\ p\frac{1}{2}\dot{\mathbf{q}}^T M_l(\mathbf{q})\dot{\mathbf{q}} \\ p\mathcal{U}_l(\mathbf{q}) \\ \vdots \\ p\mathcal{U}_r(\mathbf{q}) \\ f_1(\dot{\mathbf{q}}) \\ \vdots \\ f_s(\dot{\mathbf{q}}) \end{bmatrix}, \quad (19)$$

que es equivalente a (12). Lo anterior completa la prueba de la Proposición 1.

### C. Estimador de parámetros MC+EMDR

El estimador de parámetros propuesto es una metodología basada en una etapa de mínimos cuadrados con factor de olvido (MC) más la aplicación de la técnica de extensión y mezcla dinámica del regresor (EMDR) y fue originalmente presentado en [15]. Para el caso de sistemas descritos por una ERL de la forma (9), el estimador se define mediante la siguiente extensión dinámica:

$$\begin{aligned} \dot{\hat{\boldsymbol{\mu}}} &= \alpha P\Omega(y - \Omega^T \hat{\boldsymbol{\mu}}), \\ \dot{P} &= -\alpha P\Omega\Omega^T P + \beta P, \\ \dot{z} &= -\beta z, \\ \beta &= \beta_0 \left(1 - \frac{\|P\|}{\rho}\right), \end{aligned}$$

fijando las condiciones iniciales  $\hat{\boldsymbol{\mu}}(0) = \boldsymbol{\mu}_0 \in \mathbb{R}^w$ ,  $P(0) = \frac{1}{p_0} I_{w \times w}$ ,  $z(0) = 1$ , con las constantes  $\alpha > 0$ ,  $p_0 > 0$ ,  $\beta_0 > 0$  y  $\rho \geq \frac{1}{p_0}$  definen nuevas ERLs escalares:

$$y_i = \Delta\theta_i, \quad i = \{1, \dots, w\}, \quad (20)$$

donde  $\Delta := \det\{I_{w \times w} - zp_0 P\}$ ,  $y := \text{adj}\{I_{w \times w} - zp_0 P\} \times [\hat{\boldsymbol{\mu}} - zp_0 P\boldsymbol{\mu}_0] \in \mathbb{R}^w$  con  $y = \text{col}\{y_1, \dots, y_w\}$ . Como se indica en [15], si  $\Omega(t)$  es EI, es decir satisface (45) mostrado en el Apéndice, entonces existe un  $\Delta_{\min} > 0$  tal que  $\Delta(\tau) \geq \Delta_{\min}$ ,  $\forall t \geq \tau$ .

### III. UN NUEVO CONTROLADOR ADAPTABLE COMPUESTO

En esta sección presentamos la combinación del estimador de parámetros MC+EMDR con el controlador adaptable directo con realimentación de salida presentado en [16] y [17]. El fin de incorporar señales del estimador de parámetros en la dinámica de adaptación, más específicamente el error de identificación, es para garantizar la convergencia paramétrica reemplazando la condición de EP (véase Apéndice ecuación (44)) en el regresor original por una condición más relajada denominada EI (véase Apéndice ecuación (45)) en un nuevo regresor obtenido mediante la ecuación de balance de potencia y el estimador MC+EMDR presentado en las secciones anteriores. El nuevo controlador obtenido mediante la combinación de señales en la ley de adaptación y la misma ley de control se denomina controlador adaptable compuesto [11].

### A. Objetivo de control

Formalmente, el objetivo del control de seguimiento de trayectorias consiste en diseñar una ley de control  $\tau$  de (1) junto con una ley de adaptación para garantizar que

$$\lim_{t \rightarrow \infty} \tilde{\mathbf{q}}(t) = \lim_{t \rightarrow \infty} [\mathbf{q}_d(t) - \mathbf{q}(t)] = \mathbf{0}_n, \quad (21)$$

$$\lim_{t \rightarrow \infty} \dot{\tilde{\mathbf{q}}}(t) = \lim_{t \rightarrow \infty} [\dot{\mathbf{q}}_d(t) - \dot{\mathbf{q}}(t)] = \mathbf{0}_n, \quad (22)$$

con  $\tilde{\mathbf{q}}(t) := \mathbf{q}_d(t) - \mathbf{q}(t)$ , para toda condición inicial  $[\mathbf{q}(0)^T \ \dot{\mathbf{q}}(0)^T]^T$  y donde  $\mathbf{q}_d(t) \in \mathbb{R}^n$  es el vector de trayectorias deseadas el cual se asume variante en el tiempo, al menos dos veces diferenciable y acotado de la forma  $\|\dot{\mathbf{q}}_d(t)\| \leq \kappa_1$ ,  $\|\ddot{\mathbf{q}}_d(t)\| \leq \kappa_2$ , donde  $\kappa_1$  y  $\kappa_2$  son constantes conocidas.

### B. Propiedades del modelo dinámico

El sistema EL descrito en (5) puede ser linealmente parametrizable [20] como:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathcal{R}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = Y(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta}, \quad (23)$$

para todo  $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ , donde  $Y(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{n \times w}$  es una matriz de regresión y  $\boldsymbol{\theta}$  son los parámetros desconocidos del modelo previamente definidos en (8).

*Definición 1:* La llamada dinámica residual [20] se define por:

$$\begin{aligned} \mathbf{h}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) &= [M(\mathbf{q}_d) - M(\mathbf{q}_d - \tilde{\mathbf{q}})]\ddot{\tilde{\mathbf{q}}} \\ &\quad + [C(\mathbf{q}_d, \dot{\mathbf{q}}_d) - C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\mathbf{q}}_d - \dot{\tilde{\mathbf{q}}})]\dot{\tilde{\mathbf{q}}}_d \\ &\quad + \mathbf{g}(\mathbf{q}_d) - \mathbf{g}(\mathbf{q}_d - \tilde{\mathbf{q}}), \end{aligned} \quad (24)$$

y satisface la desigualdad  $\|\mathbf{h}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}})\| \leq k_{c_1} \kappa_1 \|\dot{\tilde{\mathbf{q}}}\| + \frac{s_2 s_1}{\tanh(s_2 \sigma)} \|\tanh(\sigma \tilde{\mathbf{q}})\|$  con  $\sigma > 0$  y las siguientes constantes:

$$\begin{aligned} s_1 &= k_g + k_M \kappa_2 + k_{c_2} \kappa_1^2, \\ s_2 &= 2 \frac{k_1 + k_2 \kappa_2 + k_{c_1} \kappa_1^2}{s_1}, \end{aligned} \quad (25)$$

donde  $k_M \geq n^2 [\max_{i,j,k,\mathbf{q}} \|\frac{\partial M_{ij}(\mathbf{q})}{\partial q_k}\|]$ ,  $k_{c_1} \geq n^2 [\max_{i,j,k,\mathbf{q}} |C_{kij}(\mathbf{q})|]$ ,  $k_{c_2} \geq n^3 [\max_{i,j,k,\mathbf{q}} \|\frac{\partial c_{ijk}(\mathbf{q})}{\partial q_k}\|]$ ,  $k_1 \geq \sup_{\mathbf{q} \in \mathbb{R}^n} \|\mathbf{g}(\mathbf{q})\|$  y  $k_2 \geq \lambda_{\max}\{M(\mathbf{q})\}$  para todo  $\mathbf{q} \in \mathbb{R}^n$  donde  $M_{ij}(\mathbf{q})$  es el  $ij$ -ésimo elemento de la matrix  $M(\mathbf{q})$  y  $c_{ijk}(\mathbf{q})$  es el  $ijk$ -ésimo símbolo de Christoffel [20].

### C. Controlador adaptable compuesto con realimentación de salida

El resultado principal de este manuscrito se presenta en la siguiente proposición.

*Proposición 2:* Considere el sistema (5), satisfaciendo (9) y (20). Suponga que la fricción en el sistema es tal que  $\lambda_{\min}\{\mathcal{R}\} > k_{c_1} \kappa_1$ . Con una ley de control dada por

$$\begin{aligned} \tau &= Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)\hat{\boldsymbol{\theta}} + K_v \tanh(\tilde{\boldsymbol{\vartheta}}) \\ &\quad + K_p \tanh(\sigma \tilde{\mathbf{q}}), \end{aligned} \quad (26)$$

$$\begin{aligned} \dot{\hat{\boldsymbol{\theta}}} &= \Gamma Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)^T [\dot{\tilde{\mathbf{q}}} + \epsilon \tanh(\sigma \tilde{\mathbf{q}})] \\ &\quad + \Gamma \Delta (y - \Delta \hat{\boldsymbol{\theta}}), \end{aligned} \quad (27)$$

$$\dot{z} = -A \tanh(\tilde{\boldsymbol{\vartheta}}), \quad (28)$$

$$\dot{\tilde{\boldsymbol{\vartheta}}} = z + B \tilde{\mathbf{q}}, \quad (29)$$

con  $A = \text{diag}\{a_1, a_2, \dots, a_n\}$ ,  $a_i > 0$ ,  $B = \text{diag}\{b_1, b_2, \dots, b_n\}$ ,  $b_i > 0$ ,  $K_p = \text{diag}\{k_{p1}, k_{p2}, \dots, k_{pn}\}$ ,  $k_{pi} > 0$ ,  $K_v = \text{diag}\{k_{v1}, k_{v2}, \dots, k_{vn}\}$  y  $k_{vi} > 0$ . Suponga que  $\Omega(t)$  verifica (45) mostrado en el Apéndice y que  $K_p$  se elige tal que  $\lambda_{\min}\{K_p\} > \kappa_1$ . Definiendo  $\tilde{\theta} := \theta - \theta$ , entonces el origen del sistema en lazo cerrado obtenido  $[\tilde{q}^T \ \dot{\tilde{q}}^T \ \tilde{\vartheta}^T \ \dot{\tilde{\theta}}^T]^T = [\mathbf{0}_n^T \ \mathbf{0}_n^T \ \mathbf{0}_n^T \ \mathbf{0}_w^T]^T$  es un equilibrio globalmente uniformemente asintóticamente estable si  $\epsilon$  verifica

$$0 < \epsilon_1 < \epsilon < \min\{\epsilon_2, \epsilon_3, \epsilon_4\}, \quad (30)$$

con

$$\epsilon_1 = \frac{v_1^2}{[1 - \xi][\lambda_{\min}\{K_p\} - v_1]\lambda_{\min}\{\mathcal{R}\}}, \quad (31)$$

$$\epsilon_2 = \frac{[\lambda_{\min}\{K_p\} - v_1][\xi\lambda_{\min}\{\mathcal{R}\} - k_{c1}\kappa_1]}{[\lambda_{\min}\{K_p\} - v_1]v_3 + v_2^2}, \quad (32)$$

$$\epsilon_3 = \frac{2[\lambda_{\min}\{K_p\} - v_1]\lambda_{\min}\{K_v B^{-1}A\}}{\lambda_{\max}^2\{K_v\}}, \quad (33)$$

$$\epsilon_4 = \frac{\sqrt{\frac{1}{\sigma}\lambda_{\min}\{K_p\}\lambda_{\min}\{M(\mathbf{q})\}}}{\lambda_{\max}\{M(\mathbf{q})\}}, \quad (34)$$

donde  $\xi \in (0, 1)$ , y las siguientes constantes:

$$\begin{aligned} v_1 &= \frac{s_1 s_2}{\tanh(s_2 \sigma)}, \\ v_2 &= 2k_{c1}\kappa_1 + \lambda_{\max}\{\mathcal{R}\}, \\ v_3 &= k_{c1}\sqrt{n} + \sigma\lambda_{\max}\{M(\mathbf{q})\}. \end{aligned}$$

*Prueba 2:* El sistema en lazo cerrado obtenido de (5) sustituyendo la ley de control (26)-(29) está dado por:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \\ \tilde{\vartheta} \\ \dot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ M(\mathbf{q})^{-1}[-C(\mathbf{q}, \dot{\mathbf{q}})\dot{\tilde{q}} - \mathcal{R}\dot{\tilde{q}} - K_v \mathbf{tanh}(\tilde{\vartheta}) - K_p \mathbf{tanh}(\sigma\tilde{q}) - \mathbf{h}(\tilde{q}, \dot{\tilde{q}}) + Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)\tilde{\theta}] \\ -A \mathbf{tanh}(\tilde{\vartheta}) + B\dot{\tilde{q}} \\ -\Gamma Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)[\dot{\tilde{q}} + \epsilon \mathbf{tanh}(\sigma\tilde{q})] \\ -\Gamma \Delta \dot{\tilde{\theta}} \end{bmatrix}. \quad (35)$$

El origen de (35), es decir  $[\tilde{q}^T \ \dot{\tilde{q}}^T \ \tilde{\vartheta}^T \ \dot{\tilde{\theta}}^T]^T = [\mathbf{0}_n^T \ \mathbf{0}_n^T \ \mathbf{0}_n^T \ \mathbf{0}_w^T]^T$  es un único punto de equilibrio [16].

#### D. Análisis de estabilidad del sistema en lazo cerrado

Considerando la siguiente función de Lyapunov:

$$\begin{aligned} V &= \frac{1}{2}\dot{\tilde{q}}^T M(\mathbf{q})\dot{\tilde{q}} + \sum_{i=1}^n b_i^{-1} \ln(\cosh(\tilde{\vartheta}_i)) \\ &+ \sum_{i=1}^n k_{pi} \sigma^{-1} \ln(\cosh(\sigma\tilde{q}_i)) \\ &+ \epsilon \mathbf{tanh}(\sigma\tilde{q})^T M(\mathbf{q})\dot{\tilde{q}} + \frac{1}{2}\tilde{\theta}^T \Gamma^{-1} \tilde{\theta}, \end{aligned} \quad (36)$$

la cual es definida positiva globalmente y radialmente desacotada [16]. Su derivada temporal a través de las trayectorias del sistema (35) está dada por:

$$\begin{aligned} \dot{V} &= \epsilon \mathbf{tanh}(\sigma\tilde{q})^T [-\mathcal{R}\dot{\tilde{q}} - K_v \mathbf{tanh}(\tilde{\vartheta}) \\ &- K_p \mathbf{tanh}(\sigma\tilde{q}) + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\tilde{q}} - \mathbf{h}(\tilde{q}, \dot{\tilde{q}})] \\ &+ \epsilon \sigma \dot{\tilde{q}}^T M(\mathbf{q}) \text{Sech}^2(\sigma\tilde{q})\dot{\tilde{q}} - \dot{\tilde{q}}^T \mathbf{h}(\tilde{q}, \dot{\tilde{q}}) - \dot{\tilde{q}}^T \mathcal{R}\dot{\tilde{q}} \\ &- \mathbf{tanh}(\tilde{\vartheta})^T K_v B^{-1} A \mathbf{tanh}(\tilde{\vartheta}) - \tilde{\theta}^T \Delta^2 \dot{\tilde{\theta}}. \end{aligned} \quad (37)$$

La derivada temporal (37) satisface la cota superior:

$$\dot{V} \leq -U, \quad (38)$$

donde

$$\begin{aligned} U &= \begin{bmatrix} \|\mathbf{tanh}(\sigma\tilde{q})\| \\ \|\dot{\tilde{q}}\| \\ \|\tilde{\theta}\| \end{bmatrix}^T Q_1 \begin{bmatrix} \|\mathbf{tanh}(\sigma\tilde{q})\| \\ \|\dot{\tilde{q}}\| \\ \|\tilde{\theta}\| \end{bmatrix} \\ &+ \begin{bmatrix} \|\mathbf{tanh}(\sigma\tilde{q})\| \\ \|\dot{\tilde{q}}\| \\ \|\tilde{\theta}\| \end{bmatrix}^T Q_2 \begin{bmatrix} \|\mathbf{tanh}(\sigma\tilde{q})\| \\ \|\dot{\tilde{q}}\| \\ \|\tilde{\theta}\| \end{bmatrix}, \end{aligned} \quad (39)$$

siendo

$$\begin{aligned} Q_1 &= \begin{bmatrix} \frac{\epsilon}{2}[\lambda_{\min}\{K_p\} - v_1] & -\frac{1}{2}v_1 - \frac{1}{2}\epsilon v_2 & 0 \\ -\frac{1}{2}v_1 - \frac{1}{2}\epsilon v_2 & \lambda_{\min}\{\mathcal{R}\} - k_{c1}\mu_1 - \epsilon v_3 & 0 \\ 0 & 0 & \frac{1}{2}\Delta^2 \end{bmatrix}, \\ Q_2 &= \begin{bmatrix} \frac{\epsilon}{2}[\lambda_{\min}\{K_p\} - v_1] & -\frac{\epsilon}{2}\lambda_{\max}\{K_v\} & 0 \\ -\frac{\epsilon}{2}\lambda_{\max}\{K_v\} & \lambda_{\min}\{K_v B^{-1}A\} & 0 \\ 0 & 0 & \frac{1}{2}\Delta^2 \end{bmatrix}. \end{aligned}$$

Seleccionando  $\epsilon$  adecuadamente para satisfacer (30) entonces  $Q_1$  y  $Q_2$  son matrices definidas positivas, por lo que  $\dot{V}$  es definida negativa. De acuerdo a lo anterior se concluye que el origen de (35), es decir  $[\tilde{q}^T \ \dot{\tilde{q}}^T \ \tilde{\vartheta}^T \ \dot{\tilde{\theta}}^T]^T = [\mathbf{0}_n^T \ \mathbf{0}_n^T \ \mathbf{0}_n^T \ \mathbf{0}_w^T]^T$  es un equilibrio globalmente uniformemente asintóticamente estable. Más aún, concluimos que los objetivos de control (21)-(22) se verifican, lo que completa la prueba de la Proposición 2.

#### IV. RESULTADOS DE SIMULACIÓN

En esta sección presentamos los resultados de simulaciones numéricas utilizando el controlador adaptable compuesto (26)-(29) y la versión del controlador adaptable directo presentado en [16] y [17], utilizando como planta un robot manipulador totalmente actuado de dos grados de libertad. Las simulaciones se realizaron en el software Matlab Simulink, con el método de integración ODE23t, con un error de tolerancia de  $1 \times 10^{-6}$ . En la Fig. 1 se muestra un esquema del robot propuesto, donde  $q_1$  y  $q_2$  son las posiciones articulares. Los parámetros del modelo dinámico del robot son los siguientes  $l_1 = 0.45[\text{m}]$ ,  $l_2 = 0.45[\text{m}]$ ,  $l_{c1} = 0.091[\text{m}]$ ,  $l_{c2} = 0.048[\text{m}]$ ,  $m_1 = 23.902[\text{Kg}]$ ,  $m_2 = 3.88[\text{Kg}]$ ,  $I_1 = 1.266[\frac{\text{Kg} \cdot \text{m}^2}{\text{rad}^2}]$ ,  $I_2 = 0.093[\frac{\text{Kg} \cdot \text{m}^2}{\text{rad}^2}]$ ,  $g = 9.81[\frac{\text{m}}{\text{s}^2}]$ . El modelo dinámico (5) se puede describir a partir de la matriz de inercia:

$$M(\mathbf{q}) = \begin{bmatrix} \theta_1 + 2\theta_2 \cos(q_2) & \theta_3 + \theta_2 \cos(q_2) \\ \theta_3 + \theta_2 \cos(q_2) & \theta_3 \end{bmatrix}, \quad (40)$$

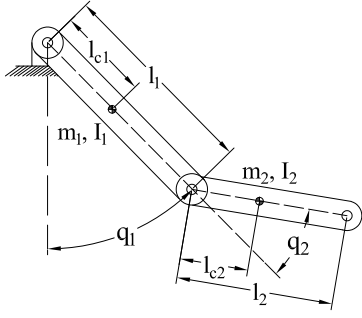


Fig. 1: Esquema del robot de 2 grados de libertad.

y la función de energía potencial dada por:

$$\mathcal{U}(\mathbf{q}) = -g \cos(q_1)\theta_4 - g \cos(q_1 + q_2)\theta_5. \quad (41)$$

La fricción se modela como (3), siendo  $\mathcal{R} = \text{diag}\{f_{v_1}, f_{v_2}\} = \text{diag}\{\theta_6, \theta_7\}$ , donde los parámetros constantes desconocidos están definidos por:  $\theta_1 = m_1 l_{c_1}^2 + m_2 l_1^2 + m_2 l_{c_2}^2 + I_1 + I_2$ ,  $\theta_2 = l_1 m_2 l_{c_2}$ ,  $\theta_3 = m_2 l_{c_2}^2 + I_2$ ,  $\theta_4 = l_{c_1} m_1 + m_2 l_1$ ,  $\theta_5 = m_2 l_{c_2}$ ,  $\theta_6 = f_{v_1}$ ,  $\theta_7 = f_{v_2}$ , con

$$\boldsymbol{\theta} = \text{col}\{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7\}. \quad (42)$$

Los valores nominales de los parámetros dinámicos considerando las definiciones anteriores son  $\theta_1 = 2.351[\frac{\text{Kg}\cdot\text{m}^2}{\text{rad}^2}]$ ,  $\theta_2 = 0.083[\frac{\text{Kg}\cdot\text{m}^2}{\text{rad}^2}]$ ,  $\theta_3 = 0.101[\frac{\text{Kg}\cdot\text{m}^2}{\text{rad}^2}]$ ,  $\theta_4 = 3.921[\text{Kg}\cdot\text{m}]$ ,  $\theta_5 = 0.186[\text{Kg}\cdot\text{m}]$ ,  $\theta_6 = 8[\text{Nm}\frac{\text{s}}{\text{rad}}]$ ,  $\theta_7 = 8[\text{Nm}\frac{\text{s}}{\text{rad}}]$ .

De acuerdo con la Sección II-B la parametrización se define considerando:

$$M_1 := \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, M_2(q_2) := \cos(q_2) \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}, M_3 := \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\mathcal{U}_1(\mathbf{q}) := -g \cos(q_1 + q_2), \quad \mathcal{U}_2(\mathbf{q}) := -g \cos(q_1 + q_2),$$

$$f_1 := \dot{q}_1^2, \quad f_2 := \dot{q}_2^2.$$

La ERL de (9) se verifica de  $y = H(p)[\dot{\mathbf{q}}^T \boldsymbol{\tau}]$ , y el vector regresor  $\boldsymbol{\Omega} = H(p)\text{col}\{p[\frac{1}{2}\dot{q}_1^2], p[(\dot{q}_1^2 + \dot{q}_1\dot{q}_2)\cos(q_2)], p[\frac{1}{2}\dot{q}_2^2 + \dot{q}_1\dot{q}_2], p[-g\cos(q_1)], p[-g\cos(q_1 + q_2)], \dot{q}_1^2, \dot{q}_2^2\}$ .

La sintonización de las ganancias del controlador adaptable propuesto para esta simulación se realizó de la siguiente forma. Primero se ajustan las ganancias del estimador de parámetros MC+EMDR descrito en la sección II-C mediante una simulación con una entrada  $\boldsymbol{\tau}$  de (5) constante. Una vez obtenido un desempeño adecuado del estimador y verificando la convergencia paramétrica, entonces se continua con la sintonización de las ganancias del controlador adaptable compuesto dado en (26)-(29), utilizando las ganancias para el estimador MC+EMDR obtenidas del paso anterior y verificando las restricciones para la ganancia  $\epsilon$  dadas en (30). Para la sintonización de las ganancias del controlador adaptable directo de [16] y [17] se procede directamente al segundo paso del procedimiento anterior. La selección de la ganancia  $\epsilon$  para ambos controladores se hizo considerando las constantes del modelo dinámico reportadas en [21] y dadas por  $\lambda_{\max}\{M\} =$

$5.03[\text{Kg}\cdot\text{m}^2]$ ,  $k_M = 0.672[\text{Kg}\cdot\text{m}^2]$ ,  $k_{c_1} = 0.336[\text{Kg}\cdot\text{m}^2]$ ,  $k_{c_2} = 0.672[\text{Kg}\cdot\text{m}^2]$ ,  $k_g = 80.578[\text{Kg}\cdot\text{m}^2/\text{s}^2]$ ,  $k_1 = 40.2931[\text{Kg}\cdot\text{m}^2]$  y  $k_2 = 5.03[\text{Kg}\cdot\text{m}^2]$ . Las ganancias del controlador adaptable compuesto y directo utilizados durante la simulación fueron fijados como  $\lambda = 1, \alpha = 1, f_0 = 1, \beta_0 = 1, \rho = 2, \mu_0 = \text{col}\{0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$ ,  $k_{p_1} = 4900, k_{p_2} = 4900, k_{v_1} = 10, k_{v_2} = 5, \sigma = 0.65, \xi = 0.45, a_1 = 100, a_2 = 100, b_1 = 100, b_2 = 100, \epsilon = 0.95, \Gamma = \text{diag}\{25, 5, 5, 25, 5, 25, 5\}$ .

Para todas las simulaciones numéricas, las posiciones y velocidades iniciales se fijaron como  $\mathbf{q}(0) = [1 \ 1]^T$  [rad] y  $\dot{\mathbf{q}}(0) = [0 \ 0]^T$  [rad/s] respectivamente. Además se seleccionó el siguiente vector de trayectorias deseadas  $\mathbf{q}_d(t) = \text{col}\{\frac{\pi}{4}e^{-0.1t}, \frac{\pi}{4}e^{-0.01t}\}$  [rad], el cual genera un regresor  $Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$  que no es de excitación persistente. Lo anterior se puede verificar dado que de acuerdo a la definición de la trayectoria deseada propuesta  $\mathbf{q}_d(t) \rightarrow 0$  conforme  $t \rightarrow \infty \Rightarrow \dot{\mathbf{q}}_d(t) \rightarrow 0 \Rightarrow \ddot{\mathbf{q}}_d(t) \rightarrow 0$  conforme  $t \rightarrow \infty$ , por lo que todos los elementos del regresor  $Y$  que no corresponden al par gravitacional tenderán a 0 conforme  $t \rightarrow \infty$ . Una análisis cuidadoso de la matriz  $Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)^T$  nos muestra que, de acuerdo a lo anterior y considerando los elementos debidos al par gravitacional que no tienden a cero conforme  $t \rightarrow \infty$ , el determinante de ésta matriz de dimensiones  $2 \times 2$  tiende a 0 conforme  $t \rightarrow \infty$ . Más aún, el  $\det\{\int_t^{t+T} Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)^T d\tau\} \rightarrow 0$  conforme  $t \rightarrow \infty$ , por lo que claramente no verifica EP, y por lo tanto utilizando la ley de adaptación original de [16] y [17] no se garantiza convergencia paramétrica para la trayectoria deseada  $\mathbf{q}_d(t)$  propuesta.

Se realizaron dos simulaciones, la primera de ellas con el controlador adaptable directo original de [16] y [17] los resultados de simulación se presentan en la Fig. 2. Como se puede apreciar claramente los objetivos de control (21)-(22) se verifican pero la convergencia de los parámetros estimados a los valores reales no. Lo anterior corrobora que el regresor  $Y(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$  no verifica EP para la trayectoria deseada propuesta. Como se observa en Fig. 2, puede verificarse que aún cuando la matriz  $\int_0^t Y Y^T d\tau$  es definida positiva para todo  $t > 0$ , esta no es uniformemente definida positiva, lo cual no verifica EP. Para visualizar la condición anterior, tome de referencia  $t > 200$ [s] de las gráficas en la Fig. 2, si la integración se inicia en este instante el  $\det\{\int_t^{t+T} Y Y^T d\tau\} \rightarrow 0$  y por lo tanto no existe un  $T > 0$  que verifique EP. La siguiente simulación fue utilizando el controlador adaptable compuesto (26)-(29) y los resultados se presentan en la Fig. 3. Como se puede apreciar los objetivos de control (21)-(22) se verifican y además garantizamos la convergencia paramétrica asintótica para todos los parámetros estimados a sus valores reales. Finalmente, note también de la Fig.3 que la suposición EI sobre el regresor  $\boldsymbol{\Omega}(t)$  se verifica, ya que  $\Delta(t) \geq 0$  para todo  $t > 0$ .

## V. CONCLUSIONES

Se ha presentado un controlador adaptable compuesto para el seguimiento de trayectorias de robots manipuladores ac-

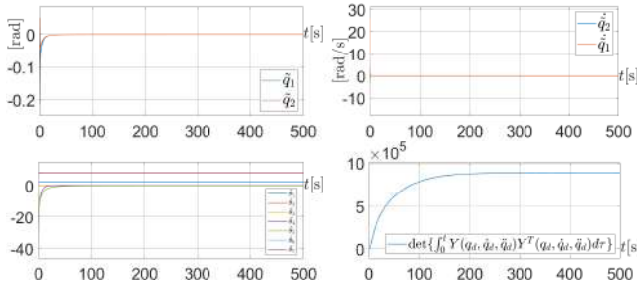


Fig. 2: Comportamiento de las señales del sistema en la simulación utilizando el controlador adaptable directo presentado en [16] y [17].

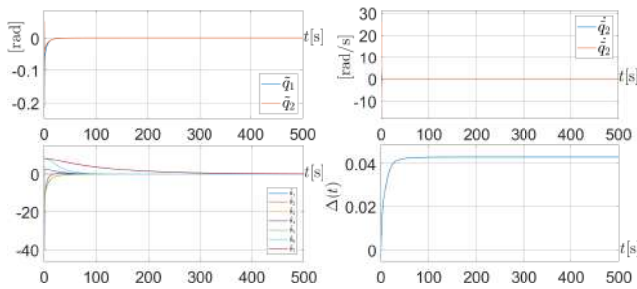


Fig. 3: Comportamiento de las señales del sistema en la simulación utilizando el controlador adaptable compuesto dado por (26)-(29).

cionados por par que garantiza estabilidad global uniforme asintótica del origen del sistema en lazo cerrado resultante. Mediante la técnica de estimación de parámetros MC+EMDR de [15] y una parametrización basada en la ecuación de balance de potencia de sistemas mecánicos, se ha mejorado la propuesta del controlador adaptable directo presentado en [16] y [17], con la principal ventaja de la relajación de condiciones de excitación en un nuevo regresor. Los resultados de simulación validan la propuesta teórica.

#### AGRADECIMIENTOS

Este trabajo fue parcialmente apoyado por las subvenciones CONAHCYT CVU 1106239, Proyecto TecNM Clave 20021.24-P, Proyecto TecNM Clave 20827.24-P y Red Internacional de Control y Cómputo Aplicados (RICCA) del TecNM.

#### APÉNDICE

##### CONDICIONES DE EXCITACIÓN

Considere un ERL escalar dado por

$$x = \varphi\theta, \quad (43)$$

con  $x(t) \in \mathbb{R}$  y  $\varphi(t) \in \mathbb{R}$  señales acotadas y medibles, y  $\theta \in \mathbb{R}$  un parámetro constante desconocido.

**Definición 2:** La función  $\varphi$  de (43) se dice que es de excitación persistente (EP) si existen  $\mu_1 > 0$  y  $T > 0$  tales que

$$\int_t^{t+T} \varphi^2(\tau) d\tau \geq \mu_1, \quad \forall t \in \mathbb{R}_+. \quad (44)$$

**Definición 3:** Se dice que la función  $\varphi$  de (43) es de excitación por intervalo (EI) si verifica

$$\int_0^t \varphi^2(\tau) d\tau \geq \mu_2, \quad (45)$$

para algún  $\mu_2 > 0$  y algún  $t > 0$ .

#### REFERENCIAS

- [1] K. Narendra and A. Annaswamy, *Stable Adaptive Systems*, ser. Dover Books on Electrical Engineering. Dover Publications, 2012.
- [2] J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991.
- [3] K. J. Åström and B. Wittenmark, *Adaptive Control*. Courier Corporation, 2008.
- [4] G. Tao, *Adaptive control design and analysis*. John Wiley & Sons, 2003, vol. 37.
- [5] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*, ser. Dover Books on Electrical Engineering Series. Dover Publications, 2011.
- [6] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive control: algorithms, analysis and applications*. Springer Science & Business Media, 2011.
- [7] M. Prifer, C. Schmidt, and F. Wahl, "Identification of robot dynamics with differential and integral models: a comparison," in *Proceedings of the 1994 IEEE international conference on robotics and automation*. IEEE, 1994, pp. 340–345.
- [8] K. R. Kozlowski and P. Dutkiewicz, "Experimental identification of robot and load dynamics," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 397–402, 1996.
- [9] P. Hsu, M. Bodson, S. Sastry, and B. Paden, "Adaptive identification and control for manipulators without using joint accelerations," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 1987, pp. 1210–1215.
- [10] C. C. De Wit and A. Aubin, "Parameters identification of robots manipulators via sequential hybrid estimation algorithms," *IFAC Proceedings Volumes*, vol. 23, no. 8, pp. 163–168, 1990.
- [11] J.-J. E. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, no. 4, pp. 509–519, 1989.
- [12] P. K. Khosla and T. Kanade, "Parameter identification of robot dynamics," in *1985 24th IEEE conference on decision and control*. IEEE, 1985, pp. 1754–1760.
- [13] J.-J. E. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Transactions on Automatic Control*, vol. 33, no. 11, pp. 995–1003, 1988.
- [14] W. Li and J.-J. E. Slotine, "An indirect adaptive robot controller," *Systems & Control Letters*, vol. 12, no. 3, pp. 259–266, 1989.
- [15] R. Ortega, J. G. Romero, and S. Aranovskiy, "A new least squares parameter estimator for nonlinear regression equations with relaxed excitation conditions and forgetting factor," *Systems & Control Letters*, vol. 169, p. 105377, 2022.
- [16] J. Moreno-Valenzuela, V. Santibanez, E. Orozco-Manriquez, and L. Gonzalez-Hernandez, "Theory and experiments of global adaptive output feedback tracking control of manipulators," *IET control theory & applications*, vol. 4, no. 9, pp. 1639–1654, 2010.
- [17] A. Yarza, V. Santibanez, and J. Moreno-Valenzuela, "An adaptive output feedback motion tracking controller for robot manipulators: Uniform global asymptotic stability and experimentation," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 3, pp. 599–611, 2013.
- [18] J. Romero, R. Ortega, and A. Bobtsov, "Parameter estimation and adaptive control of euler-lagrange systems using the power balance equation parameterisation," *International Journal of Control*, vol. 96, no. 2, pp. 475–487, 2023.
- [19] R. Ortega, A. Loria, P. J. Nicklasson, and H. Sira-Ramírez, *Passivity-based control of euler-lagrange systems: mechanical, electrical and electromechanical applications*. Springer London, 1998.
- [20] R. Kelly, V. Santibáñez, and A. Loria, *Control of robot manipulators in joint space*. Springer London, 2006.
- [21] R. Kelly and V. Santibáñez, *Control de movimiento de robots manipuladores*. Pearson Educación, 2003.

# Seguimiento de trayectoria con modelo cinemático extendido para un robot Ackerman

Jasmanik Torres-López  
*División de estudios de posgrado*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
jasmanik.t.l.96@gmail.com

Hiram Garcia-Lozano  
*División de estudios de posgrado*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
garcia.hiram@gmail.com

Jesús Santiaguillo-Salinas  
*División de estudios de posgrado*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
jsantiaguillo@outlook.com

Jesús Reyes-Rios  
*División de estudios de posgrado*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
j.r09.r05.r96@gmail.com

Luis Castro-Romero  
*División de estudios de posgrado*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
aldaircastro2022@outlook.com

José Alfonso-Jiménez  
*División de estudios de posgrado*  
*Universidad del Papaloapan*  
Loma Bonita, Oaxaca, México  
jluis.01@icloud.com

**Abstract**—En este trabajo se presenta el diseño de una estrategia de control para el seguimiento de trayectoria con un robot móvil en configuración Ackerman. El modelo matemático del robot se construye con base a una extensión cinemática de tercer orden. Se utiliza un control por linealización entrada/salida a fin de garantizar la convergencia del robot a la trayectoria deseada. Los resultados teóricos se evalúan en una simulación y se validan experimentalmente.

**Index Terms**—Robot Ackerman, control por linealización entrada/salida, modelo cinemático, extensión cinemática, seguimiento de trayectorias.

## I. INTRODUCCIÓN

El control de vehículos con sistema de dirección Ackerman, es un área abierta de investigación en tecnologías autónomas y sistemas avanzados de asistencia al conductor (ADAS). El modelo cinemático Ackerman ha sido ampliamente utilizado en la descripción del movimiento de vehículos. Sin embargo, este modelo puede ser insuficiente para capturar el comportamiento real del robot en situaciones desafiantes.

El control y navegación autónoma de vehículos móviles, con configuración Ackerman, ha sido abordado con variedad de enfoques. Por ejemplo, en [2], se presenta el diseño y modelado de una plataforma robótica para la agricultura de precisión RAM (Robot Agrícola Móvil). En este trabajo se compara el desempeño del modelo de RAM con el modelo Ackerman y otros modelos matemáticos, por ejemplo el Ramajani y el Tabile. En [3], se describe la cinemática de un vehículo con configuración Ackerman considerando factores, como el deslizamiento lateral y el frenado, que afectan las restricciones no holónomas del vehículo para el movimiento del robot. En [4], se expone la implementación de un algoritmo de procesamiento de datos de GPS (Global Position System) y sensores inerciales para mejorar la localización de robots móviles, con la finalidad de incrementar la precisión y fiabilidad de la posición estimada. En [5] se utiliza un

sistema de navegación global por satélite en tiempo real, así como una interpolación cúbica spline, para suavizar las trayectorias predeterminadas en un sistema de control de navegación por retroalimentación de estado, basado en un modelo de dirección virtual Ackerman. En [6] se estudia la viabilidad del modelo Ackerman en automóviles, mostrando que el principio de Ackerman es beneficioso en ángulos de dirección grandes a bajas velocidades, ya que reduce el ángulo de deslizamiento y la interacción neumático-carretera, haciendo los giros más suaves y reduciendo el desgaste de los neumáticos. En [7], se presenta una estrategia de control para lograr el seguimiento en formación de un sistema multi-agente heterogéneo, se implementa el modelo de un vehículo en configuración Ackerman y configuración diferencial, permitiendo la coordinación eficiente de múltiples vehículos. En [8], se implementa un modelo cinemático Ackerman para el seguimiento de trayectorias en sistemas con remolque, utilizando datos de velocidad y aceleración recogidos por una unidad de medición inercial (IMU). En [9], se incorpora un modelo del vehículo en configuración Ackerman, implementado para transportar cargas pesadas (vehículo industrial) de forma no tripulada. Se analizan actuadores mecánicos en módulos electrónicos, señales y datos de los servomotores, para mejorar el control en vehículos AWS (All Wheel Steering). En [11] se presenta un análisis, basado en el modelo difuso-Sugeno, del comportamiento no lineal completo de neumáticos, lo cual permite probar vehículos bajo demandas rápidas y agresivas de dirección. Este enfoque se implementa en un vehículo eléctrico con dirección Ackerman, desarrollado mediante un enfoque híbrido que fusiona las ecuaciones convencionales de movimiento dinámico y cinemático. Finalmente, en [12] se propone un controlador de trayectoria y un algoritmo de generación de trayectorias para un vehículo Ackerman, utilizando un modelo dinámico que se fundamenta en las señales de entrada, como el par del motor vinculado al acelerador y el movimiento del volante.

En este trabajo se aborda el problema de seguimiento de trayectorias para un robot móvil en configuración Ackerman, con tracción trasera. Se construye su modelo cinemático a partir de una extensión cinemática, obteniendo un sistema de tercer orden. Se emplea como técnica de diseño de control una linealización entrada/salida considerando como referencia de control el punto medio del eje de las ruedas traseras.

Este trabajo se encuentra organizado en 6 secciones. En la Sección II se describe el modelo cinemático del robot, así como su extensión cinemática. Posteriormente, en la Sección III y IV se realiza el planteamiento del problema y el diseño de la estrategia de control, respectivamente. La Sección V presenta la validación numérica y experimental de los resultados teóricos. Por último, en la Sección VI se muestran las conclusiones y trabajo a futuro.

## II. PRELIMINARES

### A. Modelo cinemático

Se considera un robot móvil con configuración Ackerman, con tracción trasera, que se desplaza a lo largo del plano inercial fijo  $XY$ . La Fig. 1 muestra el esquema del robot móvil, donde  $\xi = [x, y]^T$  representa las coordenadas del punto medio del eje de las ruedas traseras,  $\theta$  es el ángulo de orientación del cuerpo del robot,  $\phi$  es el ángulo de dirección de las ruedas delanteras,  $\ell$  es la distancia entre ejes,  $v$  es la velocidad longitudinal del robot y  $\omega$  es la velocidad angular de dirección

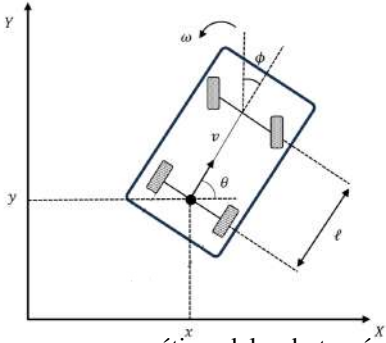


Fig. 1: Diagrama esquemático del robot móvil en configuración Ackerman.

El modelo que describe el movimiento del robot, tomando como referencia de control el punto medio del eje de las ruedas traseras, esta dado por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \tan \phi & 0 \\ \frac{\ell}{0} & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (1)$$

Considerando  $\dot{\xi} = [\dot{x}, \dot{y}]^T$  como la salida del sistema, donde  $\dot{x}$  y  $\dot{y}$  representan las velocidades en las direcciones  $X$  y  $Y$ , respectivamente, tenemos

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = A(\theta) \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (2)$$

con  $A(\theta)$  la matriz de desacoplamiento cinemático.

La estrategia de control seleccionada en este trabajo realiza una linealización por retroalimentación de estados [1]. Esta metodología permite implementar un control lineal basado en terminos del error, facilitando el análisis y diseño del control. En esta estrategia, es indispensable que la matriz de desacoplamiento cinemático  $A(\theta)$  sea no singular, es decir, que exista su inversa. Sin embargo, dado que  $\det(A(\theta)) = 0$ , el modelo (2) no puede ser utilizado. Por lo tanto, se propone hacer uso de la extensión cinemática.

El modelo cinemático extendido se obtiene derivando, en (2), los estados  $\dot{x}$  y  $\dot{y}$  con respecto del tiempo, con lo que se obtiene

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} u_1 \cos \theta - v^2 \frac{\tan \phi}{\ell} \sin \theta \\ u_1 \sin \theta + v^2 \frac{\tan \phi}{\ell} \cos \theta \end{bmatrix}, \quad (3)$$

con  $u_1 = \dot{v}$ . En (3) no se encuentra la variable de control  $\omega$ , por lo tanto, se vuelve a derivar con respecto al tiempo  $\ddot{x}$  y  $\ddot{y}$ , obteniendo

$$\begin{aligned} \begin{bmatrix} \ddot{\ddot{x}} \\ \ddot{\ddot{y}} \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\frac{v^2}{\ell} \sin \theta (1 + \tan^2 \phi) \\ \sin \theta & \frac{v^2}{\ell} \cos \theta (1 + \tan^2 \phi) \end{bmatrix} \begin{bmatrix} u_2 \\ \omega \end{bmatrix} + B \\ &= A(\ell, v, \theta, \phi) \begin{bmatrix} u_2 \\ \omega \end{bmatrix} + B, \end{aligned} \quad (4)$$

con  $u_2 = \dot{u}_1$  y

$$B = \begin{bmatrix} -\frac{3vu_1}{\ell} \sin \theta \tan \phi - \frac{v^3}{\ell^2} \cos \theta \tan^2 \phi \\ \frac{3vu_1}{\ell} \cos \theta \tan \phi - \frac{v^3}{\ell^2} \sin \theta \tan^2 \phi \end{bmatrix}.$$

En el sistema (4) aparecen las dos entradas de control, teniendo una nueva matriz de desacoplamiento  $A(\ell, v, \theta, \phi)$  para la cual  $\det(A(\ell, v, \theta, \phi)) = \frac{v^2}{\ell \cos^2 \phi}$ . Para hacer uso de la estrategia de control propuesta, se deben cumplir las siguientes condiciones  $v \neq 0, \ell \neq 0$  y  $\phi \neq \pm \frac{\pi}{2}$ , a fin de que  $A(\ell, v, \theta, \phi)$  sea invertible.

## III. PLANTEAMIENTO DEL PROBLEMA

Considere una trayectoria continuamente diferenciable preestablecida por el usuario  $m(t) = [x_d(t), y_d(t)]^T$  y sea  $\xi(t) = [x(t), y(t)]^T$ , la posición del punto medio de las ruedas del robot en configuración Ackerman. El objetivo de este trabajo es diseñar una estrategia, tal que se mantenga el seguimiento asintótico de la trayectoria por parte del robot (control de seguimiento), i.e.

$$\lim_{t \rightarrow \infty} (\xi(t) - m(t)) = 0.$$

## IV. ESTRATEGIA DE CONTROL

Para el modelo (4) se propone una estrategia de control diseñada con la técnica de linealización entrada/salida, dada por

$$U = \begin{bmatrix} u_2 \\ \omega \end{bmatrix} = A^{-1}(v, \ell, \theta, \phi) \left( \begin{bmatrix} r_x \\ r_y \end{bmatrix} - B(v, \ell, \theta, \phi) \right) \quad (5)$$

donde  $A^{-1}(v, \ell, \theta, \phi)$  es la inversa de la matriz de desacoplamiento y  $[r_x, r_y]^T$  es el control auxiliar, definido por

$$\begin{bmatrix} r_x \\ r_y \end{bmatrix} = \begin{bmatrix} \ddot{x}_d - k_2(\ddot{x} - \ddot{x}_d) - k_1(\dot{x} - \dot{x}_d) - k_0(x - x_d) \\ \ddot{y}_d - k_2(\ddot{y} - \ddot{y}_d) - k_1(\dot{y} - \dot{y}_d) - k_0(y - y_d) \end{bmatrix} \quad (6)$$

con  $[x_d, y_d]^T$ ,  $[\dot{x}_d, \dot{y}_d]^T$  y  $[\ddot{x}_d, \ddot{y}_d]^T$ , las posiciones, velocidades y aceleraciones de la trayectoria deseada, respectivamente y  $k_0, k_1, k_2$  las ganancias de control.

**Proposición 1.** Considere el sistema (4) y la ley de control (5). Suponga  $k_2, k_1, k_0 > 0$  con  $k_1 > \frac{k_0}{k_2}$ , entonces el sistema en lazo cerrado (4)-(5)-(6) es estable y el error converge asintóticamente a cero.

**Demostración.** A partir de la estrategia de control (5)-(6) y el modelo cinemático (4), el sistema en lazo cerrado, esta dado por

$$\begin{aligned} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} &= A(v, \ell, \theta, \phi) \left[ A^{-1}(v, \ell, \theta, \phi) \left( \begin{bmatrix} r_x \\ r_y \end{bmatrix} - B \right) \right] + B \\ &= \begin{bmatrix} r_x \\ r_y \end{bmatrix} \\ &= \begin{bmatrix} \ddot{x}_d - k_2(\ddot{x} - \ddot{x}_d) - k_1(\dot{x} - \dot{x}_d) - k_0(x - x_d) \\ \ddot{y}_d - k_2(\ddot{y} - \ddot{y}_d) - k_1(\dot{y} - \dot{y}_d) - k_0(y - y_d) \end{bmatrix}. \end{aligned} \quad (7)$$

Por lo tanto, la dinámica del error en  $x$  y  $y$ , esta dada por

$$\begin{aligned} \ddot{e}_x + k_2\dot{e}_x + k_1e_x + k_0e_x &= 0, \\ \ddot{e}_y + k_2\dot{e}_y + k_1e_y + k_0e_y &= 0, \end{aligned} \quad (8)$$

donde los errores de posición, velocidad y aceleración de las componentes  $x$  y  $y$  se definen como

$$\begin{aligned} e_x &= x - x_d, & e_y &= y - y_d, \\ \dot{e}_x &= \dot{x} - \dot{x}_d, & \dot{e}_y &= \dot{y} - \dot{y}_d, \\ \ddot{e}_x &= \ddot{x} - \ddot{x}_d, & \ddot{e}_y &= \ddot{y} - \ddot{y}_d. \end{aligned}$$

Al aplicar la transformada de Laplace en (8), el polinomio característico de la dinámica del error en  $x$  y  $y$  se describe como

$$s^3 + k_2s^2 + k_1s + k_0 = 0. \quad (9)$$

Se hace uso del criterio de Routh para establecer las condiciones en las que se garantice la convergencia del error a cero, es decir, para que los valores  $k_0, k_1$  y  $k_2$  de las raíces del polinomio característico se encuentren en el semiplano izquierdo del plano complejo. La tabla de Routh, para (9), se contruye de la siguiente manera

$$\begin{array}{c|cc} s^3 & 1 & k_1 \\ s^2 & k_2 & k_0 \\ s^1 & b & 0 \\ s^0 & k_0 & 0 \end{array}$$

Tabla 1: Tabla de Routh para el polinomio característico

con  $b = \frac{k_2k_1 - k_0}{k_2}$ . El criterio de Routh establece que, todas las raíces del polinomio en cuestión se encuentran en el semiplano complejo izquierdo, si no existe cambio de signo

en los elementos de la primer columna. Por lo tanto  $k_0 > 0$ ,  $k_2 > 0$  y  $k_1 > \frac{k_0}{k_2}$ .

Para calcular las ganancias  $k_0, k_1$  y  $k_2$ , se propone la forma de (9) como un polinomio de tercer orden con un par de polos dominantes complejos,

$$(s + p)(s^2 + 2\zeta\bar{\omega} + \bar{\omega}^2), \quad (10)$$

donde  $\zeta$  es el factor de amortiguamiento y  $\bar{\omega}$  la frecuencia natural no amortiguada. Igualando (9) con (10), las ganancias de control se definen de la forma

$$\begin{aligned} k_2 &= p + 2\zeta\bar{\omega}, \\ k_1 &= 2p\zeta\bar{\omega} + \bar{\omega}^2, \\ k_0 &= p\bar{\omega}^2. \end{aligned} \quad (11)$$

Por lo tanto, dadas  $k_0 > 0$  y  $k_2 > 0$ , la condición  $k_1 > \frac{k_0}{k_2}$  establece que el error de seguimiento converge asintóticamente a cero cuando

$$k_1 > \frac{p\bar{\omega}^2}{p + 2\zeta\bar{\omega}}. \quad \square$$

## V. VALIDACIÓN NUMÉRICA Y EXPERIMENTAL

La estrategia de control se valida numéricamente mediante simulaciones en el software MATLAB/Simulink y experimentalmente por medio de la plataforma de la Fig. 2. El robot móvil utilizado es un prototipo a escala 1:10 de un automóvil (Autominy 4.0), diseñado por investigadores de la Freie Universität Berlin [10].

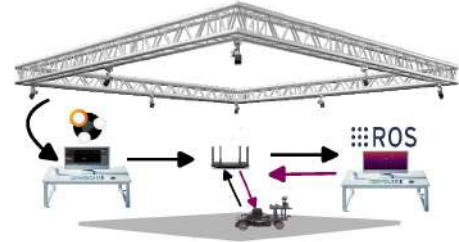


Fig. 2: Plataforma experimental.

La plataforma consta de un sistema de adquisición de datos, una etapa de control y un sistema de comunicación. El sistema de adquisición de datos esta conformado por 7 cámaras infrarrojas OptiTrack, en conjunto con el software Motive. Las cámaras capturan el movimiento del robot en cada instante de tiempo, mientras que el software Motive procesa los datos capturados para obtener la posición y orientación del robot.

La etapa de control recibe los datos del sistema de adquisición. La ley de control, programada en Python, utiliza estos datos para construir las entradas de control, las cuales se envían al robot para el seguimiento de trayectoria.

Por último, la etapa de comunicación esta conformada por el sistema operativo para robots (ROS), así como el uso de un router, el cual permite la comunicación entre los distintos componentes de la plataforma experimental.

La Fig. 3 muestra el diagrama a bloques del sistema en lazo cerrado utilizado para la validación numérica. El bloque trayectoria, contiene las ecuaciones paramétricas que describen la trayectoria deseada.

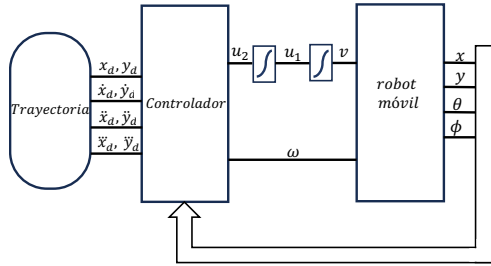


Fig. 3: Diagrama a bloques para la simulación numérica.

Las ganancias de control  $k_0, k_1, k_2$  fueron calculadas, tal que, la respuesta del error en  $x$  y  $y$  sea subamortiguada. Se propone un factor de amortiguamiento  $\zeta = 0.7$ , un tiempo de asentamiento  $t_s = 6$  s. Haciendo uso del criterio del 5%,  $t_s = \frac{0.3}{\zeta \bar{\omega}}$ , se determina la frecuencia natural  $\bar{\omega} = 0.7143 \frac{rad}{s}$ . Estos valores dan como resultado un par de polos complejos conjugados,  $s_{1,2} = -0.5 \pm 0.5101i$ . El polo real se coloca en  $p = 5$ , de tal forma que sea 10 veces mayor a la parte real de los polos complejos. Por lo tanto, las ganancias son las siguientes:  $k_0 = 2.551, k_1 = 5.51, k_2 = 6.0$ , con un tiempo de muestreo de 0.02 s.

#### A. Prueba 1

En la prueba 1 se utiliza una trayectoria circular con radio  $r = 1.3$  m, periodo  $T = 70$  s, y frecuencia  $\omega = \frac{2\pi}{T} = 0.0898$ . La trayectoria deseada  $m(t)$  tiene las siguientes ecuaciones paramétricas

$$\begin{aligned} x_d(t) &= r \cos(\omega t), \\ y_d(t) &= r \sin(\omega t), \end{aligned}$$

El comportamiento del robot, en simulación y en el experimento, se observa en la Fig. 4. La línea continua de color verde representa la trayectoria deseada, mientras que la trayectoria del robot, en la simulación y el experimento, se representa con líneas punteadas roja y azul, respectivamente. Las condiciones iniciales deben ser seleccionadas cerca de la trayectoria debido a que los resultados de este trabajo son locales.

Se puede observar en la Fig. 5 que, en la simulación, el error de seguimiento converge a cero. En color rojo se presentan los errores de posición en la simulación y en color azul los errores en el experimento. La respuesta transitoria inicial corresponde a la diferencia de las condiciones iniciales del robot y del inicio de la trayectoria deseada. Se puede notar un pequeño sobreimpulso debido a las ganancias de control seleccionadas.

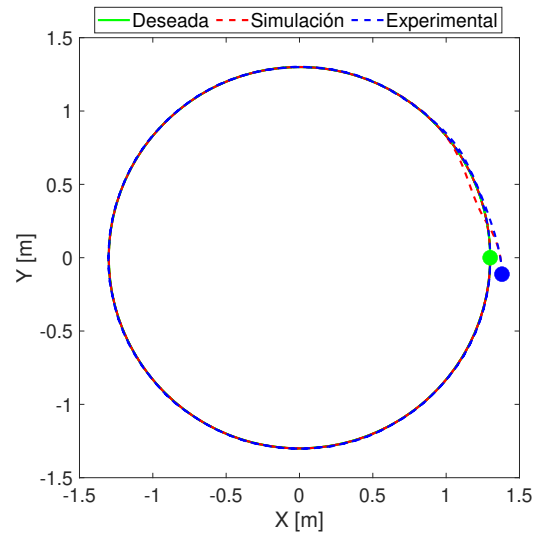


Fig. 4: Seguimiento de trayectoria.

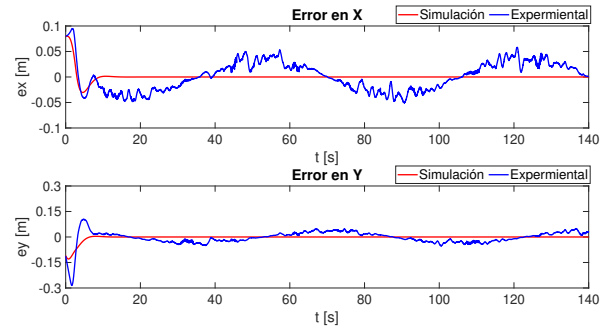


Fig. 5: Errores de posición.

El sistema alcanza el estado estacionario en aproximadamente  $t_s = 8$  s. Cabe resaltar que, en los resultados experimentales, después de alcanzar el estado estacionario se presentan oscilaciones con amplitud de  $\pm 5$  cm.

En la Fig. 6 se muestran las señales de control de velocidad  $v$ , donde la línea roja corresponde a la simulación, mientras que la azul al experimento.

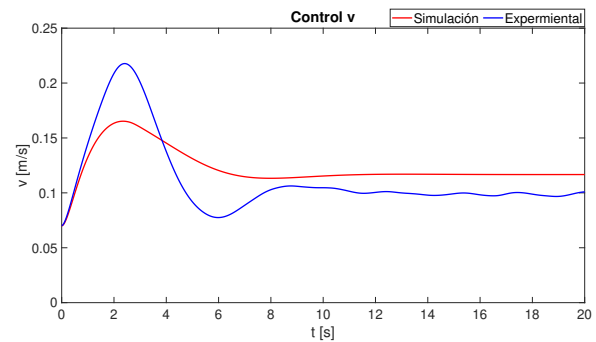


Fig. 6: Entrada de control  $v$ .

En esta figura se puede observar que, en el estado transitorio, el esfuerzo de control alcanza un pico mayor a 0.2 m/s, sin

embargo, una vez alcanzado el estado estacionario se mantiene constante alrededor de  $0.1 \text{ m/s}$ .

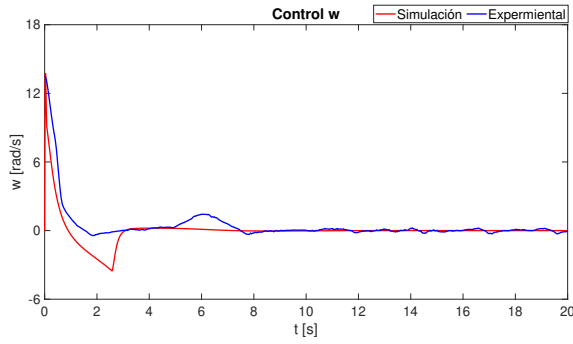


Fig. 7: Entrada de control  $w$ .

La Fig. 7 muestra la señal de control  $w$ . Como se puede observar en esta figura, después de alcanzar el estado estacionario, la velocidad angular se mantiene en una vecindad del cero.

### B. Prueba 2

En la segunda prueba se utilizó como trayectoria deseada  $m(t)$  una lemniscata de Bernoulli, cuyas ecuaciones paramétricas, están dadas por

$$x_d(t) = \frac{a \cos(\omega t)}{1 + \sin^2(\omega t)},$$

$$y_d(t) = \frac{b \sin(\omega t) \cos(\omega t)}{1 + \sin^2(\omega t)},$$

con periodo  $T = 100 \text{ s}$ , frecuencia natural  $\omega = 0.0897 \frac{\text{rad}}{\text{s}}$ , con las mismas ganancias de control usadas en la trayectoria circular. El resultado del seguimiento con esta trayectoria se muestra en la Fig. 8.

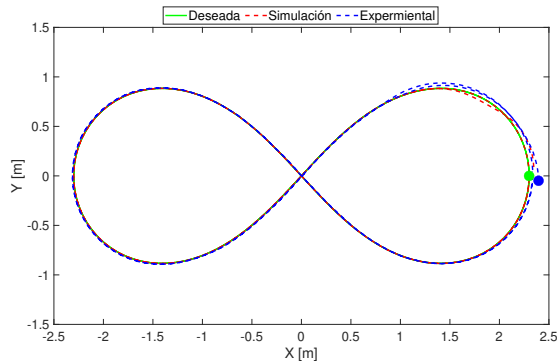


Fig. 8: Seguimiento de trayectoria.

Se utilizaron condiciones iniciales idénticas en la simulación y en el experimento,  $x = 2.34 \text{ m}$ ,  $y = 0.08 \text{ m}$ , orientando el robot en dirección de la trayectoria. Como se puede observar en la Fig. 9, el error en la simulación decae a cero, línea continua roja. Sin embargo, en la experimentación, el error se mantiene en una vecindad del cero, teniendo un error máximo en los extremos de la lemniscata. Son en estas secciones donde

se tienen los máximos esfuerzos de control (Fig. 10 y Fig. 11), lo que satura el ángulo de dirección  $\phi$  del robot, como se puede observar en la Fig. 12. El rango del ángulo de dirección del autominy está limitado a  $-0.36 < \phi < 0.38 \text{ rad}$  debido a restricciones mecánicas.

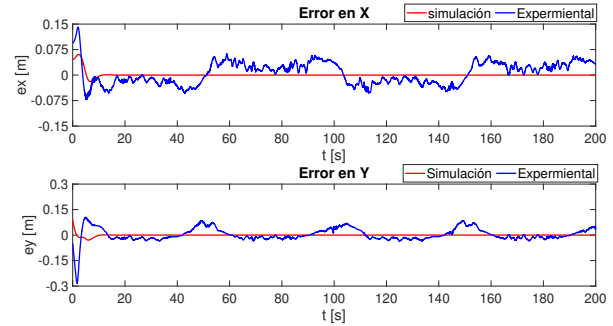


Fig. 9: Error de posición.

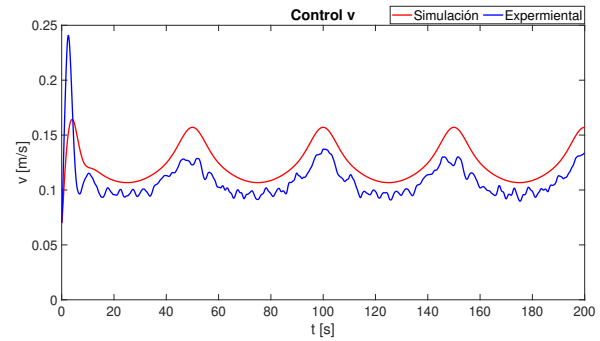


Fig. 10: Entrada de control  $v$ .

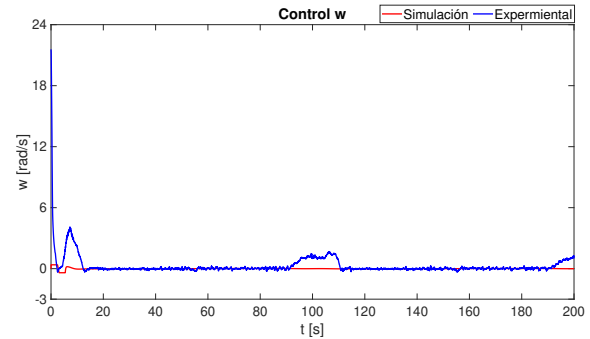


Fig. 11: Entrada de control  $w$ .

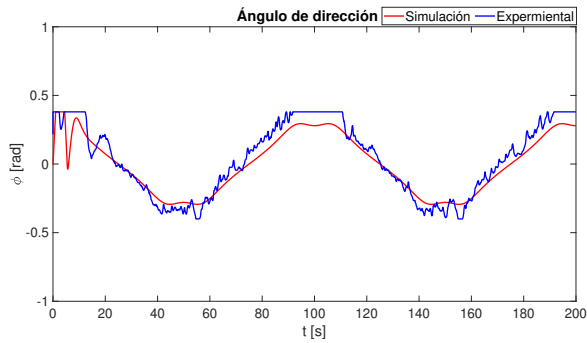


Fig. 12: Ángulo de dirección  $\phi$ .

## VI. CONCLUSIÓN

En este trabajo, se abordó el problema del seguimiento de trayectorias de un robot móvil en configuración Ackerman. Se utilizó un modelo cinemático extendido, representado por una ecuación diferencial no lineal de tercer orden. Para la estrategia de control se utilizó una linealización entrada/salida, lo que redujo el problema de control a la selección de tres ganancias de control. Se validó el modelo y la estrategia de control por medio de un software de cómputo numérico (MATLAB/Simulink) y mediante resultados experimentales.

El modelo del robot, por medio de la extensión cinemática, considera restricciones adicionales a las que se tienen con un modelo de primer orden. Además, al requerir que la matriz de desacoplamiento cinemático sea invertible se debe considerar el condicionamiento numérico de esta matriz, a fin de evitar condiciones iniciales cercanas a los puntos donde es singular, considerando que la precisión de cualquier sistema utilizado en la estrategia de control es finita.

Como trabajos futuros se pretende reducir los errores de posición logrando un seguimiento de trayectoria más preciso. Probar con otros algoritmos de control que puedan ofrecer una mayor precisión en el seguimiento de trayectorias. Realizar más experimentos en diferentes escenarios y condiciones para evaluar la robustez y eficacia del sistema de control desarrollado.

## REFERENCES

- [1] Bruno, S., Lorenzo, S., Luigi, V., Giuseppe, O. (1994). *Robotics: modelling, planning and control*, 2010. Springer London.
- [2] Torres, C. J., Archila, J. F., Tronco, M. L., Becker, M., Porto, A. J. V., & Tiberti, A. J. (2013). Estudio cinemático de una plataforma robótica para agricultura. *Revista Colombiana De Tecnologías De Avanzada (RCTA)*, 2(22), 131-137. doi.org/10.24054/16927257.v22.n22.2013.421
- [3] Ren, X., & Cai, Z. (2010, July). Kinematics model of unmanned driving vehicle. In *2010 8th World Congress on Intelligent Control and Automation* (pp. 5910-5914). IEEE. DOI: 10.1109/WCICA.2010.5554512
- [4] Marín, L., Soriano, Á., Mayans, V., Vallés, M., Valera, Á., & Albertos, P. Localización asistida por GPS para robots móviles en configuración Ackerman de recursos limitados. Conference Name XXXIV Jornadas de Automática
- [5] Zhang, L., Zhang, R., Li, L., Ding, C., Zhang, D., & Chen, L. (2022). Research on virtual Ackerman steering model based navigation system for tracked vehicles. *Computers and Electronics in Agriculture*, 192, 106615. doi.org/10.1016/j.compag.2021.106615

- [6] Sui, D., & Zhang, Y. (2023). Analyzation of the Application Scenarios of Ackerman Geometry based on Vehicle Steering Model. Highlights in Science, *Engineering and Technology*, 46, 71-82. doi.org/10.54097/hset.v46i.7676
- [7] Alfonso-Jiménez, J. L., Santiaguillo-Salinas, J., García-Lozano, H. N., & Pérez-Castro, N. (2024). Seguimiento en formación para un sistema multi-agente heterogéneo. *Pádi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, 12(Especial2), 8-13. doi.org/10.29057/icbi.v12iEspecial2.12073
- [8] Liu, Y., Iacomini, D., Powell, B., Gafford, J., Ball, J., Shi, J., & Davenport, C. (2019). Development, testing, and assessment of a kinematic path-following model for towing vehicle systems. *SAE International Journal of Vehicle Dynamics, Stability, and NVH*, 3(10-03-01-0005), 57-70. doi.org/10.4271/10-03-01-0005
- [9] Choi, M. W., Park, J. S., Lee, B. S., & Lee, M. H. (2008, October). The performance of independent wheels steering vehicle (4WS) applied Ackerman geometry. In *2008 international conference on control, automation and systems* (pp. 197-202). IEEE. DOI: 10.1109/IC-CAS.2008.4694549
- [10] Alomari, K., Mendoza, R. C., Sundermann, S., Goehring, D., & Rojas, R. (2020). Fuzzy Logic-based Adaptive Cruise Control for Autonomous Model Car. In *ROBOVIS* (pp. 121-130).
- [11] Economou, J. T., Luk, P. C. K., Tsourdos, A., & White, B. A. (2003, October). Hybrid modelling of an all-electric front-wheel Ackerman steered vehicle. In *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484)* (Vol. 5, pp. 3294-3298). IEEE. DOI: 10.1109/VETECONF.2003.1286276
- [12] Franch, J., & Rodríguez-Fortun, J. M. (2009, August). Control and trajectory generation of an ackerman vehicle by dynamic linearization. In *2009 European Control Conference (ECC)* (pp. 4937-4942). IEEE. DOI: 10.23919/ECC.2009.7075182

# Control continuo en tiempo finito para robots manipuladores

Lucero De Santiago Reyes  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de la Laguna*  
Torreón, Coahuila, 27000  
lucerosantiago@live.com.mx

Víctor Adrián Santibáñez Dávila  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de la Laguna*  
Torreón, Coahuila, 27000  
vasantibanezd@lalaguna.tecnm.mx

Jorge Alberto Villalobos Chin  
*Tecnológico Nacional de México*  
*Instituto Tecnológico de la Laguna*  
Torreón, Coahuila, 27000  
jorgevillaloboschin@hotmail.com

**Abstract**—En este documento se presenta una ley de control de posición saturada, que preserva las propiedades de convergencia en tiempo finito de controladores continuos para robots manipuladores. Se presenta un análisis formal de estabilidad en tiempo finito y se valida experimentalmente en el robot manipulador CICESE de 2 grados de libertad (GDL) que se mueve en el plano vertical. La estructura de saturación propuesta es más sencilla que otras existentes dado que utiliza operaciones computacionalmente simples y no requiere el uso de condicionales para implementar la ecuación de control.

**Index Terms**—Control continuo en tiempo finito, Robots manipuladores, Estabilidad, Evaluación experimental.

## I. INTRODUCCIÓN

Con la finalidad de desarrollar controladores que incentiven a un sistema a llegar a una posición deseada lo más pronto posible, Haimo en [1] se pregunta si es posible encontrar una ley de control continua y acotada que traslade a un sistema a su punto de equilibrio en un tiempo finito; y empieza a proponer la estructura de estos. En la actualidad, el interés en estos controladores ha aumentado, siendo estudiados y aplicados a sistemas  $n$ -dimensionales, por ejemplo, en sistemas mecánicos (Euler-Lagrange) para evaluar su desempeño en cuanto a robustez, rapidez y nulidad del error en estado estacionario del sistema. Estas son propiedades inherentes a este tipo de controladores [2]. Con la finalidad de proporcionar una estructura para leyes de control continuas que garanticen estabilidad en tiempo finito, Arturo Zavala e Isabelle Fantoni, en [3], proponen el criterio de homogeneidad local.

Convencionalmente, se busca que el equilibrio del sistema implique estabilidad asintótica y/o exponencial. En sistemas dinámicos esto significa que la convergencia de las trayectorias al punto de equilibrio se alcance a lo largo de un tiempo infinito. Sin embargo, en muchas aplicaciones es deseable que las trayectorias del sistema dinámico converjan al equilibrio estable en un tiempo finito [4].

Los teoremas de estabilidad de Lyapunov que se estudian, generalmente involucran sistemas dinámicos con campos vectoriales Lipschitz continuos, lo cual implica unicidad de soluciones en un tiempo pasado y futuro, por lo tanto, la convergencia a un estado de equilibrio es alcanzado en el infinito. Por lo tanto, con la finalidad de lograr la convergencia en tiempo finito, el sistema debe ser NO Lipschitz continuo,

garantizando así, la no unicidad de soluciones en tiempo pasado. Sin embargo, la unicidad de soluciones en tiempo futuro es preservada [4].

Como al inicio del estudio de nuevas áreas de conocimiento, los controladores de tiempo finito fueron estudiados de forma teórica, a través del análisis en trabajos como en [1], [2], [5], [6], [7], se intuía que los controladores de tiempo finito eran superiores a los controladores convencionales, con el fin de validar experimentalmente esta intuición, en [8] se formalizó e implementó un controlador PD con realimentación de estados de tiempo finito en un robot manipulador de 3 grados de libertad, demostrando así, que éste es más robusto que el controlador del mismo tipo pero exponencialmente estable. Existen otros trabajos como en [9], en el que se se introduce teórica y formalmente un controlador difuso de tiempo finito, el cual es implementado en un robot de 2 grados de libertad, mismo que es comparado con un controlador difuso asintótico.

En este trabajo se presenta un controlador continuo de tipo Saturación-Proporcional-Saturación Derivativa con compensación de gravedad (SP-SD+g), el cual garantiza la convergencia en tiempo finito del error de posición para robots manipuladores. Esto se logra al diseñar una ley de control para que el sistema en lazo cerrado sea homogéneo, más un término aditivo desvaneciente, con grado de homogeneidad negativo. Cabe resaltar que para evidenciar la superioridad del controlador de tiempo finito propuesto, este es comparado con el mismo controlador pero en el caso asintótico. Esta comparación se realizó experimentalmente en tiempo real para un robot manipulador de 2 grados de libertad, obteniéndose menor error de posición en estado estacionario.

Adicionalmente, la saturación propuesta, a diferencia de las estructuras presentadas en [7] por ejemplo, puede ser analizada usando la teoría de homogeneidad y aproximaciones homogéneas clásica [10] por lo que el análisis teórico es más sencillo.

En la sección II de este documento, se presenta el modelo dinámico del robot manipulador así como sus propiedades, además, la ley de control continuo con convergencia en tiempo finito es originalmente introducida. En la sección III, se encuentra el análisis de estabilidad del controlador, el cual consta de 2 etapas. La primera muestra la prueba de estabilidad asintótica, mientras que la segunda etapa, constituye

el análisis de estabilidad de tiempo finito. Posteriormente, en la sección IV se encuentran los resultados experimentales obtenidos, y en la sección V, se encuentran las conclusiones.

## II. MÓDELO DINÁMICO Y CONTROLADOR

### A. Modelo dinámico del robot manipulador CICESE

El modelo dinámico de un robot manipulador de  $n$  grados de libertad, es abordado para su análisis como un sistema mecánico articulado, formado mediante eslabones conectados entre sí. Para su análisis, se coloca en un marco de referencia cartesiano para obtener sus parámetros. La dinámica de este robot es descrita en la ecuación (1), misma que es obtenida a través de las ecuaciones de movimiento de Lagrange como aparece en [11].

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \quad (1)$$

donde  $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$  es la matriz de inercias,  $\ddot{\mathbf{q}} \in \mathbb{R}^n$  es el vector de aceleraciones angulares,  $\dot{\mathbf{q}} \in \mathbb{R}^n$  es el vector de velocidades angulares y  $\mathbf{q} \in \mathbb{R}^n$  es el vector de posiciones angulares,  $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$  es la matriz centrífuga y de Coriolis,  $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$  es el vector de pares gravitacionales, y  $\boldsymbol{\tau} \in \mathbb{R}^n$  es el vector de fuerzas externas, o sea, los pares aplicados por los accionadores en las articulaciones del manipulador.

Los elementos del modelo dinámico correspondientes al robot utilizado se muestran a continuación. Estos se obtienen a partir de la Figura 1, la cual muestra el esquema del robot manipulador CICESE de 2 grados de libertad, es decir,  $n = 2$ , puesto que cuenta con un actuador en cada uno de sus eslabones. Además, en la Tabla I, se presentan los parámetros físicos correspondientes.

$$\begin{aligned} M_{11}(\mathbf{q}) &= m_1 l_{c1}^2 + m_2 [l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)] + I_1 + I_2, \\ M_{12}(\mathbf{q}) &= m_2 [l_{c2}^2 + l_1 l_{c2} \cos(q_2)] + I_2, \\ M_{21}(\mathbf{q}) &= m_2 [l_{c2}^2 + l_1 l_{c2} \cos(q_2)] + I_2, \\ M_{22}(\mathbf{q}) &= m_2 l_{c2}^2 + I_2 \\ C_{11}(\mathbf{q}, \dot{\mathbf{q}}) &= -m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2, \\ C_{12}(\mathbf{q}, \dot{\mathbf{q}}) &= -m_2 l_1 l_{c2} \sin(q_2) [\dot{q}_1 + \dot{q}_2], \\ C_{21}(\mathbf{q}, \dot{\mathbf{q}}) &= m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1, \\ C_{22}(\mathbf{q}, \dot{\mathbf{q}}) &= 0 \\ g_1(\mathbf{q}) &= [m_1 l_{c1} + m_2 l_1] g \sin(q_1) \\ &\quad + m_2 l_{c2} g \sin(q_1 + q_2), \\ g_2(\mathbf{q}) &= m_2 l_{c2} g \sin(q_1 + q_2). \end{aligned}$$

### B. Propiedades del modelo dinámico

En seguida se presentan un par de propiedades para robots manipuladores [11]. Los autovalores mínimos y máximos de la matriz  $M$  se denotan por  $\lambda_{\min}\{M\}$  y  $\lambda_{\max}\{M\}$  respectivamente. Además, la norma de  $\mathbf{x} \in \mathbb{R}^n$  se denota por  $\|\mathbf{x}\|$ .

**Propiedad 1.** La matriz de inercias  $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$  es una matriz simétrica y definida positiva. Por lo tanto, por el teorema

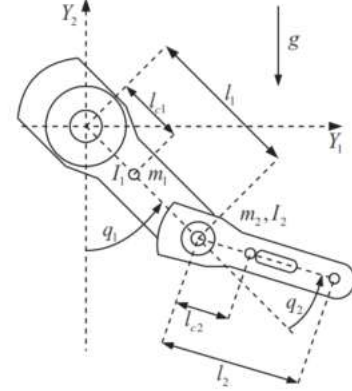


Figura 1. Esquema del robot CICESE

Tabla I  
PARÁMETROS FÍSICOS DEL ROBOT CICESE

Descripción	Notación	Valor	Unidades
Longitud eslabón 1	$l_1$	0.450	m
Longitud eslabón 2	$l_2$	0.450	m
Distancia al centro de masa (eslabón 1)	$l_{c1}$	0.091	m
Distancia al centro de masa (eslabón 2)	$l_{c2}$	0.048	m
Masa eslabón 1	$m_1$	23.902	kg
Masa eslabón 2	$m_2$	3.880	kg
Inercia eslabón 1 respecto al centro de masa	$I_1$	1.266	kg m <sup>2</sup>
Inercia eslabón 2 respecto al centro de masa	$I_2$	0.093	kg m <sup>2</sup>
Aceleración de la gravedad	$g$	9.81	m/s <sup>2</sup>

de Rayleigh-Ritz se establece que para todo  $\mathbf{x} \in \mathbb{R}$ , se cumple que

$$\lambda_{\min}\{M(\mathbf{q})\} \|\mathbf{x}\|^2 \leq \mathbf{x}^T M(\mathbf{q}) \mathbf{x} \leq \lambda_{\max}\{M(\mathbf{q})\} \|\mathbf{x}\|^2. \quad (2)$$

**Propiedad 2.** La matriz

$$\frac{1}{2} \dot{M}(\mathbf{q}) - C(\mathbf{q}, \dot{\mathbf{q}}) \quad (3)$$

es antisimétrica (donde  $C(\mathbf{q}, \dot{\mathbf{q}})$  fue obtenida mediante los coeficientes de Christoffel). También es cierto que

$$\dot{M}(\mathbf{q}) = C(\mathbf{q}, \dot{\mathbf{q}}) + C(\mathbf{q}, \dot{\mathbf{q}})^T. \quad (4)$$

### C. Controlador propuesto SP-SD+g en tiempo finito

La ley de control a utilizar, está expresada en la ecuación

$$\boldsymbol{\tau} = K_p \Phi(\tilde{\mathbf{q}}) \mathbf{x}(\tilde{\mathbf{q}}) - K_v \Gamma(\dot{\mathbf{q}}) \mathbf{y}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}), \quad (5)$$

donde:

$$K_p = \text{diag}\{k_{p1}, k_{p2}, \dots, k_{pn}\}, \quad (6)$$

es una matriz diagonal definida positiva,  $K_p \in \mathbb{R}^{n \times n}$ ,  $K_p > 0$ ,

$$K_v = \text{diag}\{k_{v1}, k_{v2}, \dots, k_{vn}\}, \quad (7)$$

es una matriz diagonal definida positiva,  $K_v \in \mathbb{R}^{n \times n}$ ,  $K_v > 0$ ,

$$\Phi(\tilde{q}) = \text{diag} \left\{ \frac{1}{\varepsilon_1 + |\tilde{q}_1|^{\alpha_1}}, \frac{1}{\varepsilon_2 + |\tilde{q}_2|^{\alpha_2}}, \dots, \frac{1}{\varepsilon_n + |\tilde{q}_n|^{\alpha_n}} \right\}, \quad (8)$$

donde  $\Phi(\tilde{q}) \in \mathbb{R}^{n \times n}$ ,

$$\mathbf{x}(\tilde{q}) = [|\tilde{q}_1|^{\alpha_1} \text{sign}(\tilde{q}_1) \quad |\tilde{q}_2|^{\alpha_2} \text{sign}(\tilde{q}_2) \quad \dots \quad |\tilde{q}_n|^{\alpha_n} \text{sign}(\tilde{q}_n)]^T, \quad (9)$$

con  $\mathbf{x}(\tilde{q}) \in \mathbb{R}^n$ ,

$$\Gamma(\dot{q}) = \text{diag} \left\{ \frac{1}{\zeta_1 + |\dot{q}_1|^{\beta_1}}, \frac{1}{\zeta_2 + |\dot{q}_2|^{\beta_2}}, \dots, \frac{1}{\zeta_n + |\dot{q}_n|^{\beta_n}} \right\}, \quad (10)$$

donde  $\Gamma(\dot{q}) \in \mathbb{R}^{n \times n}$ ,

$$\mathbf{y}(\dot{q}) = [|\dot{q}_1|^{\beta_1} \text{sign}(\dot{q}_1) \quad |\dot{q}_2|^{\beta_2} \text{sign}(\dot{q}_2) \quad \dots \quad |\dot{q}_n|^{\beta_n} \text{sign}(\dot{q}_n)]^T, \quad (11)$$

donde  $\mathbf{y}(\dot{q}) \in \mathbb{R}^n$ ,

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} g([m_1 l_{c1} + m_2 l_1] \text{sen}(q_1) + m_2 l_{c2} \text{sen}(q_1 + q_2)) \\ g \quad m_2 l_{c2} \text{sen}(q_1 + q_2) \end{bmatrix}, \quad (12)$$

donde  $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ .

*D. Ecuación de lazo cerrado.*

La ecuación de lazo cerrado del sistema está dada por la ecuación (13), que aparece a continuación.

$$\begin{bmatrix} \dot{\tilde{q}} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(\mathbf{q})^{-1} [K_p \Phi(\tilde{q}) \mathbf{x}(\tilde{q}) - K_v \Gamma(\dot{q}) \mathbf{y}(\dot{q}) - C(\mathbf{q}, \dot{q}) \dot{q}] \end{bmatrix} \quad (13)$$

### III. ESTABILIDAD GLOBAL EN TIEMPO FINITO

En esta sección, se realiza la prueba de estabilidad para el controlador (5) aplicado en la planta previamente descrita en (1). Esta prueba consta de 2 partes. La primera parte prueba estabilidad global asintótica del origen, mientras que la segunda garantiza estabilidad global en tiempo finito. Primeramente, se presentan los preeliminarios para llevar a cabo dicha prueba.

#### A. Preliminares

Para realizar la prueba de estabilidad en tiempo finito, es necesario comprender previamente una serie de conceptos, que se enunciarán en seguida.

**Definición 1. Estabilidad en tiempo finito.** [4]

Considera el sistema no lineal:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad (14)$$

se dice que su solución trivial  $\mathbf{x} \equiv 0$  es estable en tiempo finito, si existe una vecindad  $N \subset D \subset \mathbb{R}^n$  del origen, y una

función de tiempo de estabilización  $T : N \setminus \{0\} \rightarrow (0, \infty)$ , tal que se cumplan los siguientes atributos:

- Convergencia en tiempo finito.

$$\mathbf{x}(t, x_0) \neq 0, \forall t \in (0, T(x))$$

$$\mathbf{x}(t, x_0) = 0, \forall t \geq T(x)$$

- Estabilidad de Lyapunov.

Además, se dice que el origen es globalmente estable en tiempo finito si  $N = \mathbb{R}^n$ .

A continuación se recuerda el concepto de homogeneidad, el cual es útil para asegurar estabilidad en tiempo finito global para sistemas autónomos [10]. En forma coloquial, homogeneidad es una operación de escala o dilatación de acción multiplicativa de números reales positivos en el espacio de estados. Más formalmente, homogeneidad local es enunciada en términos de una familia de dilataciones  $\delta_\epsilon^r(\mathbf{x})$ , cuyo significado es el siguiente:

Una familia de dilataciones  $\delta_\epsilon^r(\mathbf{x})$ , es definida como:

$$\delta_\epsilon^r(\mathbf{x}) = (\epsilon^{r_1} x_1, \dots, \epsilon^{r_n} x_n)$$

$\forall \mathbf{x} \in \mathbb{R}^n, \forall \epsilon > 0, r_n > 0 \in \mathbb{R}$ , donde  $r_n$  son los llamados coeficientes de dilatación y son positivos diferentes de cero. Estos son agrupados, para propósitos de notación en una familia  $\mathbf{r} = (r_1, \dots, r_n)$ .

#### Definición 2. Homogeneidad.

 [9]

Dado una  $\mathbf{r}$ , una función  $V : \mathbb{R}^n \rightarrow \mathbb{R}$ , un campo vectorial  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , se dice que son  $\mathbf{r}$ -homogéneos de grado  $\alpha$  si satisfacen

$$V(\delta_\epsilon^r(\mathbf{x})) = \epsilon^\alpha V(\mathbf{x}), \quad (15)$$

respectivamente

$$\mathbf{f}(\delta_\epsilon^r(\mathbf{x})) = \epsilon^\alpha \mathbf{f}(\mathbf{x}), \quad (16)$$

para cada  $\mathbf{x} \in \mathbb{R}^n$  y toda  $\epsilon > 0$ .

#### Teorema 1.

 [10]

Sea

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}).$$

Supóngase que  $\mathbf{f}$  es homogéneo de grado  $k$ . Entonces, el origen es estable en tiempo finito si y sólo si es asintóticamente estable y  $k < 0$ .

Cuando existen términos adicionales que hacen imposible la aplicación directa del Teorema 1, se considera el siguiente caso especial:

$$\Sigma = \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \hat{\mathbf{f}}(\mathbf{x}),$$

donde  $\mathbf{f}, \hat{\mathbf{f}} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  son campos vectoriales continuos que cumplen con  $\mathbf{f}(0) = \hat{\mathbf{f}}(0) = 0$ .

#### Lema 1.

 [3]

Asuma que el origen  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$  es un equilibrio asintóticamente estable, entonces el origen de  $\Sigma$  es localmente estable en tiempo finito si se cumple el límite:

$\lim_{\epsilon \rightarrow 0^+} \frac{\hat{f}_j(\delta_r(x))}{\epsilon^{k+r_j}} = 0$  y  $\mathbf{f}(\mathbf{x})$  es homogéneo de grado  $k < 0$ .

Para llevar a cabo la prueba de estabilidad del controlador, se propone la siguiente función candidata de Lyapunov.

$$V(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) = \sum_{i=1}^n \int_0^{\tilde{q}_i} k_{pi} \frac{|\sigma|^{\alpha_i} \text{sign}(\sigma)}{\epsilon_i + |\sigma|^{\alpha_i}} d\sigma + \frac{1}{2} \dot{\tilde{\mathbf{q}}}^T M(\mathbf{q}) \dot{\tilde{\mathbf{q}}}, \quad (17)$$

donde  $0 < \alpha_i < 1$ ,  $\epsilon_i > 0$

La ecuación (17) cumple con que es definida positiva y radialmente desacotada bajo las siguientes consideraciones:

- $V(0, 0) = 0$
- $\frac{1}{2} \dot{\tilde{\mathbf{q}}}^T M(\mathbf{q}) \dot{\tilde{\mathbf{q}}} \geq \frac{1}{2} \lambda_{\min}\{M(\mathbf{q})\} \|\dot{\tilde{\mathbf{q}}}\|^2$
- $\mathbf{x} \mathbf{f}(\mathbf{x}) > 0 \rightarrow \int_0^x \mathbf{f}(\tau) d\tau > 0$ ,  $x \neq 0$  para cada elemento de la sumatoria.

Además, la derivada temporal de (17) a lo largo de las trayectorias del sistema en lazo cerrado (13), resulta en la ecuación (18) que aparece a continuación:

$$\dot{V}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) = -\dot{\tilde{\mathbf{q}}}^T K_v \Gamma(\dot{\tilde{\mathbf{q}}}) \mathbf{y}(\dot{\tilde{\mathbf{q}}}) \quad (18)$$

Finalmente, con el teorema de invarianza de La Salle [11], se concluye estabilidad asintótica global del origen del sistema en lazo cerrado (13).

Como se declaró al inicio de esta sección, esta prueba consta de dos partes, una vez que se ha concluido que el origen es globalmente asintóticamente estable, se procede a realizar la prueba de estabilidad en tiempo finito.

Para utilizar el Lema 1, enunciado previamente en los preliminares, se reescribe al sistema como en (19)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \hat{\mathbf{f}}(\mathbf{x}), \quad (19)$$

donde:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} -\dot{\tilde{\mathbf{q}}} \\ M^{-1}(\mathbf{q}_d) [K_p \Phi(0) \mathbf{x}(\tilde{\mathbf{q}}) - K_v \Gamma(0) \mathbf{y}(\dot{\tilde{\mathbf{q}}})] \end{bmatrix},$$

$$\hat{\mathbf{f}}(\mathbf{x}) = \begin{bmatrix} 0 \\ \underbrace{-M^{-1}(\mathbf{q}_d - \tilde{\mathbf{q}}) C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) \dot{\tilde{\mathbf{q}}}}_1 + \\ \underbrace{M^{-1}(\mathbf{q}_d - \tilde{\mathbf{q}}) [K_p \Phi(\tilde{\mathbf{q}}) \mathbf{x}(\tilde{\mathbf{q}}) - K_v \Gamma(\dot{\tilde{\mathbf{q}}}) \mathbf{y}(\dot{\tilde{\mathbf{q}}})]}_2 \\ \underbrace{-M^{-1}(\mathbf{q}_d) [K_p \Phi(0) \mathbf{x}(\tilde{\mathbf{q}}) - K_v \Gamma(0) \mathbf{y}(\dot{\tilde{\mathbf{q}}})]}_3 \end{bmatrix}$$

Note que el origen del sistema  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$  es asintóticamente estable de manera global ya que  $\mathbf{f}$  tiene la forma del lazo cerrado de un control PD con compensación de gravedad [11]. Para comprobar que  $\mathbf{f}(\mathbf{x})$ , es un campo vectorial homogéneo de grado  $k < 0$ , se aplica la familia de dilataciones dada por  $\delta(\mathbf{x}) = (\epsilon^{r_1} \tilde{\mathbf{q}}, \epsilon^{r_2} \dot{\tilde{\mathbf{q}}})$  obteniendo

$$\mathbf{f}(\delta(\mathbf{x})) = \begin{bmatrix} -(\epsilon^{k+r_1}) \dot{\tilde{\mathbf{q}}} \\ (\epsilon^{k+r_2}) M^{-1}(\mathbf{q}_d) [K_p \Phi(0) \mathbf{x}(\tilde{\mathbf{q}}) - K_v \Gamma(0) \mathbf{y}(\dot{\tilde{\mathbf{q}}})] \end{bmatrix} \quad (20)$$

Concluyendo así que  $\mathbf{f}(\mathbf{x})$  es homogéneo de grado  $k < 0$  si para cada  $i \in \{1, 2, \dots, n\}$  se cumple que:

$$\beta_i = \frac{2\alpha_i}{\alpha_i + 1} \quad (21)$$

Para analizar las condiciones presentadas en el Lema 1 relacionadas a  $\hat{\mathbf{f}}(\mathbf{x})$ , se analizan las expresiones señaladas como 1, 2 y 3 en la ecuación (19):

Para la expresión 1, se tiene que

$$\lim_{\epsilon \rightarrow 0^+} \frac{\|M^{-1}(\mathbf{q}_d - \epsilon^{r_1} \tilde{\mathbf{q}}) C(\mathbf{q}_d - \epsilon^{r_1} \tilde{\mathbf{q}}, \epsilon^{r_2} \dot{\tilde{\mathbf{q}}}) \epsilon^{r_2} \dot{\tilde{\mathbf{q}}}\|}{\epsilon^{k+r_2}} = \|-M^{-1}(\mathbf{q}_d) C(\mathbf{q}_d, 0) \dot{\tilde{\mathbf{q}}}\| \lim_{\epsilon \rightarrow 0^+} \frac{\epsilon^{r_2}}{\epsilon^{k+r_2}} = 0$$

Para las expresiones 2 y 3, utilizando el teorema de compresión se obtiene

$$\lim_{\epsilon \rightarrow 0^+} \frac{\|\boldsymbol{\eta}\| K_p \mathbf{x}(\epsilon^{r_1} \tilde{\mathbf{q}})}{\epsilon^{k+r_2}} \leq \lim_{\epsilon \rightarrow 0^+} \frac{\epsilon^{\alpha r_1}}{\epsilon^{k+r_2}} \lim_{\epsilon \rightarrow 0^+} \|\boldsymbol{\eta}\| \|K_p \mathbf{x}(\tilde{\mathbf{q}})\| \leq 0, \quad (22)$$

$$\lim_{\epsilon \rightarrow 0^+} \frac{\|\boldsymbol{\nu}\| K_v \mathbf{y}(\epsilon^{r_2} \dot{\tilde{\mathbf{q}}})}{\epsilon^{k+r_2}} \leq \lim_{\epsilon \rightarrow 0^+} \frac{\epsilon^{\beta r_2}}{\epsilon^{k+r_2}} \lim_{\epsilon \rightarrow 0^+} \|\boldsymbol{\nu}\| \|K_v \mathbf{y}(\dot{\tilde{\mathbf{q}}})\| \leq 0 \quad (23)$$

donde  $\boldsymbol{\eta} = M^{-1}(\mathbf{q}_d - \epsilon^{r_1} \tilde{\mathbf{q}}) \Phi(\epsilon^{r_1} \tilde{\mathbf{q}}) - M^{-1}(\mathbf{q}_d) \Phi(0)$  y  $\boldsymbol{\nu} = M^{-1}(\mathbf{q}_d - \epsilon^{r_1} \tilde{\mathbf{q}}) \Gamma(\epsilon^{r_2} \dot{\tilde{\mathbf{q}}}) - M^{-1}(\mathbf{q}_d) \Gamma(0)$

La combinación de estabilidad global asintótica y estabilidad en tiempo finito, tiene como resultado *estabilidad global en tiempo finito* del origen de (13) según el Lema 1 [12].

#### IV. RESULTADOS EXPERIMENTALES

En esta sección, se presentan los resultados experimentales en el robot manipulador CICESE de 2 grados de libertad, descrito en la Sección II. Este manipulador se muestra en la Figura 2 y es propiedad del posgrado en ingeniería eléctrica, del área de mecatrónica y control en el Instituto Tecnológico de la Laguna. De acuerdo con el fabricante de los actuadores, el torque suministrado por cada motor es de  $\tau_1 = \pm 150$  Nm y  $\tau_2 = \pm 15$  Nm.

Las posiciones deseadas para cada articulación son  $q_{d1} = 45^\circ$  y  $q_{d2} = 90^\circ$ , y la sintonización de las ganancias proporcional y derivativa para cada articulación reciben los valores que aparecen a continuación; también se muestra el valor de los coeficientes de dilatación  $\alpha_i$  y  $\beta_i$ , y de  $\epsilon_i$  y  $\zeta_i$ . Nótese que el valor de los coeficientes de dilatación son diferentes



Figura 2. Robot CICESE.

para el controlador de tiempo finito (CCTF) y el controlador asintótico (CCA):

$$\begin{aligned}
 k_{p1} &= 85.0 && [\text{N} \cdot \text{m} / \text{grado}] \\
 k_{v1} &= 25.0 && [\text{N} \cdot \text{m} \text{ s/grado}] \\
 k_{p2} &= 10.0 && [\text{N} \cdot \text{m} / \text{grado}] \\
 k_{v2} &= 1.0 && [\text{N} \cdot \text{m} \text{ s/grado}] \\
 \alpha_i &= 0.50 && \text{CCTF} \\
 \beta_i &= 2/3 && \text{CCTF} \\
 \alpha_i &= 1 && \text{CCA} \\
 \beta_i &= 1 && \text{CCA} \\
 \varepsilon_i &= 0.50 \\
 \zeta_i &= 0.50.
 \end{aligned}$$

Enseguida, se presentan las gráficas de los resultados experimentales obtenidos con el controlador **SP-SD+g** asintótico y el control **SP-SD+g** de tiempo finito, los que serán representados como “CCA” y “CCTF”, respectivamente. Se utilizan las mismas ganancias  $k_{pi}$  y  $k_{vi}$  en ambos controladores, pero el valor de las potencias  $\alpha$  y  $\beta$  es modificado.

En la Figura 3 y Figura 4 se observa el comportamiento del error de posición de la articulación 1 y 2, respectivamente; en la Figura 5 y Figura 6 aparecen los pares aplicados en cada uno de los actuadores, y finalmente, en las Figura 7 y 8, aparecen las velocidades angulares de cada articulación respecto al tiempo.

## V. CONCLUSIONES

En la comparación que se realizó, el control continuo SP-SD+g de tiempo finito implementado, demostró ser superior al SP-SD+g asintótico. Cabe resaltar que a pesar de que no

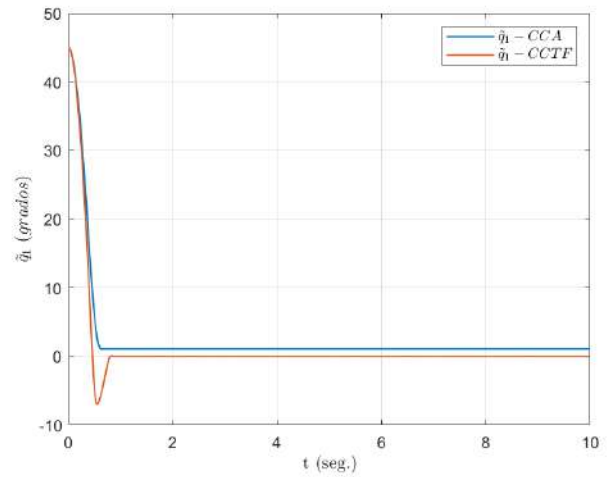


Figura 3. Error de posición de articulación 1.

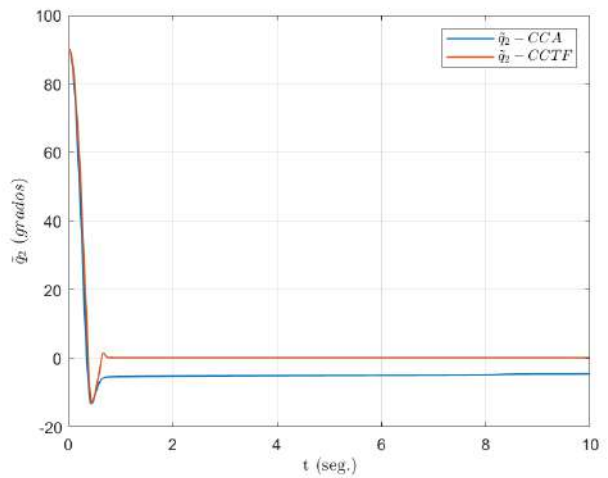


Figura 4. Error de posición de articulación 2.

fue compensada la fricción de los actuadores, el controlador SP-SD+g continuo de tiempo finito logra obtener un error en estado estacionario muy cercano al cero, por lo que se afirma que es robusto, rápido y cumple el objetivo de control.

Para ilustrar estas conclusiones, a continuación se presenta la Tabla II, la cual contiene el valor cuadrático medio del error de posición de cada articulación, y el error en estado estacionario  $\tilde{q}_{1ss}$  y  $\tilde{q}_{2ss}$  (de 1 a 10 seg.).

Tabla II  
VALOR RMS

	$\tilde{q}_1$	$\tilde{q}_2$	$\tilde{q}_{1ss}$	$\tilde{q}_{2ss}$
CCTF	6.8585	12.47	0.0078	0.2222
CCA	7.216	12.7826	1.1115	4.9124

## AGRADECIMIENTOS

Este artículo fue apoyado parcialmente por los proyectos TecNM, por la Red RICCA TecNM y por beca CONACYT.

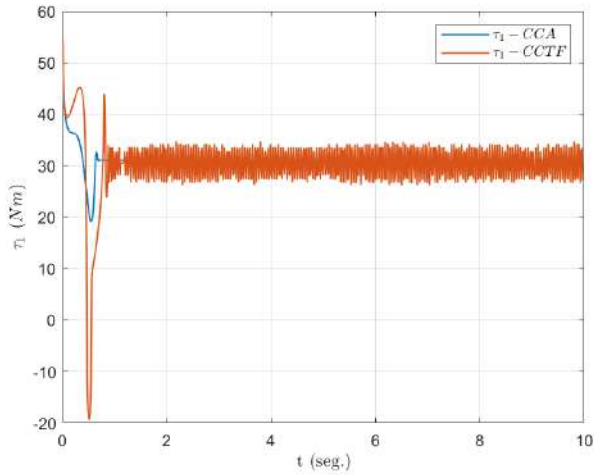


Figura 5. Par aplicado del actuador 1.

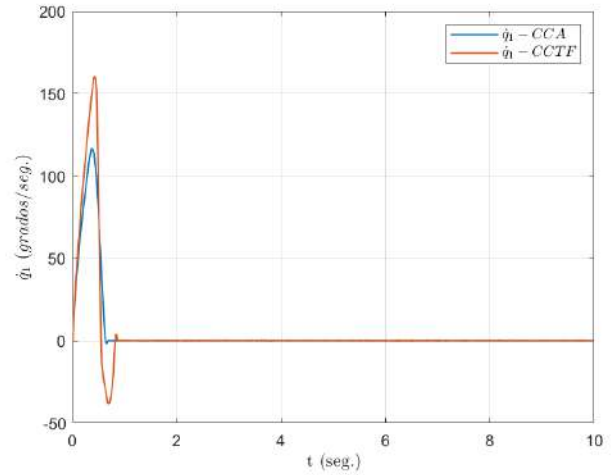


Figura 7. Velocidad angular de articulación 1.

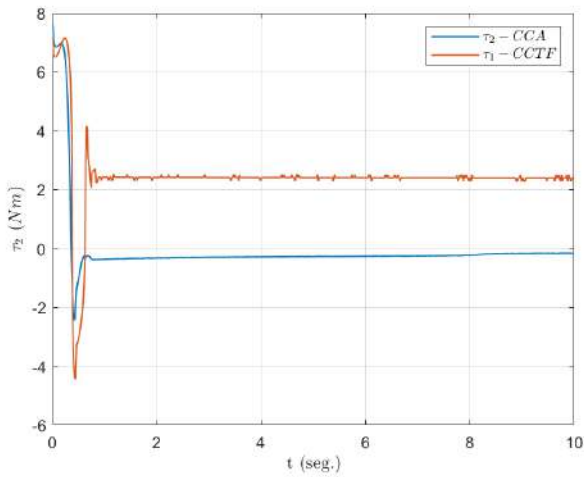


Figura 6. Par aplicado del actuador 2.

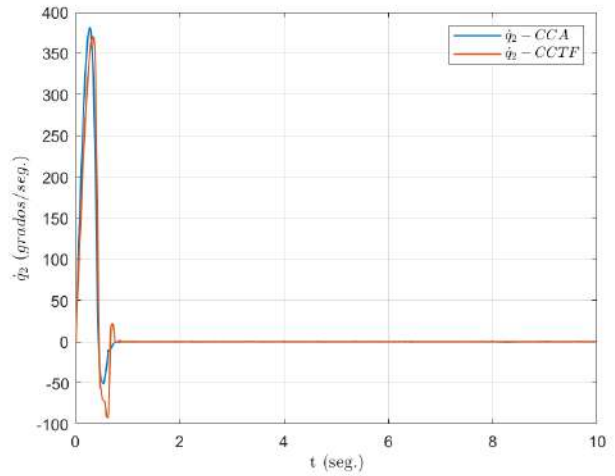



Figura 8. Velocidad angular de articulación 2.

## REFERENCES


- [1] V. T. Haimo, "Finite time controllers," *SIAM Journal on Control and Optimization*, vol. 24, no. 4, pp. 760–770, 1986.
- [2] E. Cruz-Zavala, E. Nuño, and J. Moreno, "Non-homogeneous finite-time controllers for double integrator dynamics," in *Memorias del Congreso Nacional de Control Automático*, 2021, p. 255–260.
- [3] A. Zavala-Río and I. Fantoni, "Global finite-time stability characterized through a local notion of homogeneity," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 471–477, 2013.
- [4] W. M. Haddad and V. Chellaboina, *Nonlinear dynamical systems and control: a Lyapunov-based approach*. Princeton university press, 2008.
- [5] S. Yu, X. Yu, B. Shirinzadeh, and Z. Man, "Continuous finite-time control for robotic manipulators with terminal sliding mode," *Automatica*, vol. 41, no. 11, pp. 1957–1964, 2005.
- [6] E. Cruz-Zavala, T. Sanchez, E. Nuño, and J. Moreno, "Nuevas familias de controladores de tiempo finito para sistemas de segundo orden siso y mimo," in *Memorias del Congreso Nacional de Control Automático*, 2019, pp. 401–406.
- [7] A. Zavala-Río and G. I. Zamora-Gómez, "Local-homogeneity-based global continuous control for mechanical systems with constrained inputs: finite-time and exponential stabilisation," *International Journal of Control*, vol. 90, no. 5, pp. 1037–1051, 2017.
- [8] G. I. Zamora-Gómez, A. Zavala-Río, E. Vázquez-Ramírez, F. Reyes, and V. Santibáñez, "A robustness study of a finite-time/exponential tracking continuous control scheme for constrained-input mechanical systems: analysis and experiments," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 10, pp. 3923–3944, 2020.
- [9] J. Villalobos-Chin, A. Pizarro-Lerma, V. Santibáñez, R. García-Hernández, and A. Zavala-Río, "Continuous finite-time fuzzy control of mechanical systems via non-lipschitz membership functions," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 106804, 2023.
- [10] S. P. Bhat and D. S. Bernstein, "Geometric homogeneity with applications to finite-time stability," *Mathematics of Control, Signals and Systems*, vol. 17, pp. 101–127, 2005.
- [11] R. Kelly, V. Santibáñez *et al.*, "Control de movimiento de robots manipuladores," 2003.
- [12] J. A. V. Chin, "Sobre la teoría de estabilidad y sus aplicaciones al control." Ph.D. dissertation, Instituto Tecnológico de la Laguna, 2023.

# Control gestual de un robot móvil omnidireccional mediante CNN


Carlos-A. Salado-Chávez

*Instituto de Electrónica y Mecatrónica  
Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca, México  
carlos0201@live.com.mx 


Matew Herrera-Perez

*Instituto de Electrónica y Mecatrónica  
Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca, México  
mattewhperez@gmail.com 


Karla I. Salado-Chávez

*División de estudios de Posgrado  
Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca, México  
itzchav3001@gmail.com 

Oscar D. Ramírez-Cárdenas

*División de estudios de Posgrado  
Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca, México  
odramirez@mixteco.utm.mx 

Esther Lugo González

*Instituto de Electrónica y Mecatrónica  
Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca, México  
elugog@mixteco.utm.mx 

**Abstract**—Este trabajo presenta el diseño y la construcción de un robot móvil omnidireccional, controlado mediante el reconocimiento de la lengua de señas en inglés. La dirección del robot se ajusta con señas interpretadas por una red neuronal convolucional.

El robot ha sido diseñado de forma modular, lo que facilita tanto el ensamblaje como la manufactura y permite un acceso sencillo para la reparación y el reemplazo de piezas. La comunicación del robot con una interfaz externa se realiza a través de una conexión Wi-Fi, utilizando un módulo ESP32 para la transmisión de datos. Esta configuración permite la operación del robot basada en la lengua de señas, subrayando su utilidad en la interacción humano-robot y facilitando la comunicación con personas que utilizan este lenguaje.

Además, el robot podría tener aplicaciones en la asistencia doméstica, proporcionando una solución accesible tanto para personas con pérdida auditiva o limitaciones de movilidad como para usuarios sin tales limitaciones, mejorando así la funcionalidad y el apoyo en el entorno doméstico.

**Index Terms**—Robot Omnidireccional, Lengua de señas, Visión por computadora, redes convolucionales.

## I. INTRODUCCIÓN

En los últimos años, la robótica ha experimentado avances significativos que han transformado la capacidad de los robots para interactuar con su entorno y con los humanos. Los progresos en tecnologías de sensores han permitido a los robots reconocer gestos, expresiones faciales y posturas mediante técnicas avanzadas de visión artificial. Estos desarrollos han facilitado la creación de interfaces que mejoran la interacción entre humanos y máquinas. En particular, la robótica de servicio se enfoca en asistir a los seres humanos realizando tareas que son peligrosas, repetitivas o para mejorar la calidad de vida [1]. Este campo ha llevado al desarrollo de robots móviles con diferentes configuraciones de locomoción, como los tipos diferencial, Ackermann y omnidireccional, incluyendo el modelo presentado en este trabajo.

Los robots omnidireccionales son un tipo de robots móviles terrestres que se desplazan en distintas direcciones sin

necesidad de cambiar de orientación por medio de ruedas mecanum. Es decir, efectúan distintos movimientos de rotación o traslación en el plano. Las aplicaciones que se le dan actualmente a este tipo de robots son: tareas de navegación autónoma, seguimiento de objetivos, entretenimiento, entre otras [2]. En [3], Gordón et.al. desarrollan un robot omnidireccional de cuatro ruedas con un manipulador de 5 grados de libertad con pinza de agarre sobre la estructura, el robot es capaz de planear rutas y de reconocer señalamientos de tráfico por medio de algoritmos de Visión Artificial y procesamiento de imágenes en tiempo real. A pesar de que su uso se limita a ambientes controlados y con señalizaciones de tráfico, este es un avance para los vehículos omnidireccionales de mayor capacidad de movimiento. En el 2020, Ramos [4] presenta dos robots omnidireccionales para fines de entretenimiento de 3 y 4 ruedas respectivamente, en este trabajo se implementaron en los robots mecanismos que permiten al vehículo disparar proyectiles con el fin de inmovilizar al oponente. En ese mismo año, Espinoza et.al. [5] propone el uso de un conjunto vehículos móviles trabajando coordinadamente en tareas de transporte a corta distancia resolviendo necesidades en logística industrial. Entre las propuestas de los robots móviles presentan a los omnidireccionales y describe que las ventajas en la utilización de estos es la capacidad de dirigirse en cualquier dirección, además de ser estable, sin embargo, la complejidad de implementación aumenta.

Por otro lado, en el 2023 Sotelo [6] presenta la propuesta de un robot móvil equipado con Visión Artificial para la detección de obstáculos y personas. Este robot utiliza *Python* como lenguaje de programación y aprovecha bibliotecas de código abierto como *OpenCV*. Además, incorpora ROS (Sistema Operativo de Robots), que facilita al móvil diversas funciones, incluyendo mapeo, localización y control, según se detalla en [6]. Pawar et al. [7] presentaron en 2021 un algoritmo de detección y seguimiento de objetos en tiempo real para robots móviles omnidireccionales, basado en redes neuronales

convolucionales. Los autores demostraron la efectividad de su enfoque al aplicar el algoritmo al seguimiento de una pelota de rugby, destacando su potencial para diversas aplicaciones en robótica móvil.

En este trabajo se presenta el diseño de un robot móvil omnidireccional capaz de ajustar su dirección en respuesta a señales visuales, específicamente gestos de mano, capturadas por una cámara. Para la identificación de estos gestos, se emplea un modelo de clasificación basado en una red neuronal convolucional (CNN), entrenada con un conjunto de datos personalizado. Una vez clasificada la señal, se envían las instrucciones correspondientes al robot para modificar su dirección de movimiento. Esta implementación abre nuevas posibilidades en la interacción humano-máquina, ofreciendo un control intuitivo tanto para personas con pérdida auditiva o limitaciones de movilidad como para usuarios sin estas restricciones. La adopción de esta tecnología subraya el potencial de la inteligencia artificial para mejorar la accesibilidad y la funcionalidad en diversas aplicaciones robóticas.

## II. PRELIMINARES

### A. Control PID

Para la sintonización de controladores PID, el modelo de primer orden con tiempo muerto (FOST, por sus siglas en inglés) es un modelo matemático utilizado para describir el comportamiento de sistemas industriales con retrasos significativos [8]. La función de transferencia del proceso controlado se describe como:

$$Y(s) = \frac{K}{Ts + 1} e^{-Ls} \cdot \frac{1}{U(s)} \quad (1)$$

donde  $K$  es la ganancia del proceso,  $T$  es la constante de tiempo,  $L$  es el tiempo muerto, y  $U(s)$  es la entrada del proceso.

La identificación de modelos FOST se puede realizar mediante un método gráfico propuesto por Astrom y Hagglund [9]. La ganancia estática del proceso  $K$  se obtiene a partir de los estados estacionarios de la entrada y salida del proceso. La intersección de la tangente a la respuesta escalón con la pendiente más pronunciada respecto al eje horizontal proporciona el valor de  $L$ . El tiempo  $T$  se determina a partir de la diferencia entre  $L$  y el momento en que la respuesta escalón alcanza el 63% de  $K$ .

La sintonización de controladores PI para modelos FOST es crucial para obtener una respuesta óptima en sistemas industriales. Según Marlin [10], la sintonización de controladores PI para modelos FOST se puede realizar mediante un método de dos puntos. Los tiempos en que la salida del proceso alcanza el 28% de  $K$  y el 63% de  $K$ , respectivamente, se miden y se utilizan para estimar  $T$  y  $L$ . Una vez obtenidas las ganancias iniciales del controlador PI utilizando el método FOST, se puede aplicar la sintonización fina para optimizar aún más el rendimiento del sistema mediante un controlador PID. Para la sintonización de controladores PID utilizando este método, se deben considerar los siguientes puntos clave: aumentar la ganancia proporcional disminuye la estabilidad, el error decae

más rápidamente si se disminuye el tiempo de integración, disminuir el tiempo de integración disminuye la estabilidad, y aumentar el tiempo derivativo mejora la estabilidad [11]. El controlador PID ajustará la entrada del motor para minimizar el error entre la posición deseada y la actual, optimizando así el comportamiento dinámico del motor y logrando un control más preciso.

### B. Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN o ConvNets) son algoritmos de aprendizaje profundo diseñados para el análisis de patrones espaciales. Estas redes están especializadas en identificar características como bordes, esquinas, texturas y formas abstractas, lo que les permite asociar imágenes a una clase [12]. A diferencia de las redes neuronales artificiales tradicionales, las CNN se optimizan mediante el aprendizaje a través de una serie de convoluciones y operaciones de submuestreo, que reducen la dimensión espacial y permiten la generalización de características. La red culmina en capas completamente conectadas (perceptrón multicapa) que producen la salida final correspondiente al número de clases [13].

Cada capa convolucional realiza productos escalares seguidos de funciones no lineales, procesando los vectores de imagen en bruto hasta obtener la puntuación de clase final. La operación de convolución combina la matriz de entrada con un filtro o kernel de tamaño  $n$ , mientras que el submuestreo divide la matriz en subregiones para reducir su dimensión. La última capa incluye funciones de pérdida asociadas con las clases, que guían la optimización del modelo para mejorar su rendimiento [13], [14], el esquema de una CNN se muestra en la Figura 1.

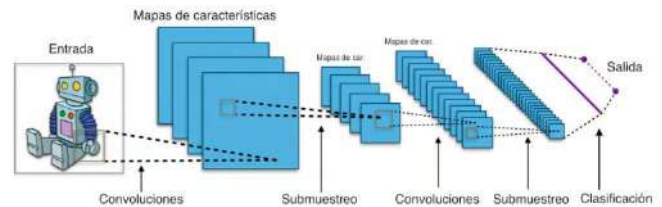


Fig. 1. Esquema de una CNN [13]

### C. Lengua de señas (ASL)

La lengua de señas americano de la Figura 2 (ASL, por sus siglas en inglés) es un lenguaje con su propia gramática.

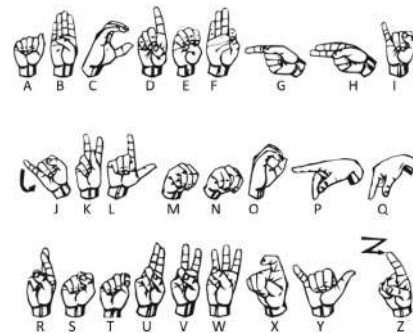


Fig. 2. Alfabeto de la lengua de señas americana. [15]

Esto significa que el ASL no se basa en una representación teatral de movimientos, sino que ofrece un lenguaje visual-gestual con reglas gramaticales propias. A diferencia del inglés, el ASL es un idioma morfológicamente complejo, comparable a los idiomas hablados polisintéticos, que suelen tener palabras con muchas sílabas [16]. En Kaggle se encuentra una base de datos, del ASL con 8442 imágenes [17].

### III. CONSTRUCCIÓN DEL ROBOT OMNIDIRECCIONAL

El proceso de elaboración del robot requiere de varias etapas: diseño, impresión 3D, manufactura, selección de componentes electrónicos y comunicación entre el robot y la computadora.

#### A. Diseño y Manufactura

El diseño del prototipo se basa en la protección de los componentes internos, incluidos las cuatro ruedas mecanum, el microcontrolador y el Puente H, y en la facilidad del proceso de manufactura. Para esto la estructura se hace con impresión 3D por la protección que ofrece a los componentes, además de ser fácil de ensamblar y reparar. La Figura 3 muestra el diseño de la estructura del robot móvil omnidireccional.

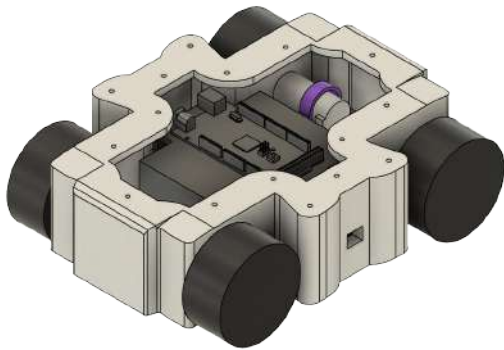


Fig. 3. Diseño CAD

Se utiliza impresión 3D con ácido poliláctico (PLA) para construir el prototipo. Debido a las limitaciones de tamaño de las impresoras 3D estándar, se fabrica la estructura con 7 piezas diferentes: una base y 6 piezas superiores. De estas 6 piezas superiores, 4 están destinadas exclusivamente al montaje de los motores, mientras que las otras 2 proporcionan el espacio necesario para los componentes adicionales. Esta configuración permite un ensamblaje y desensamblaje fácil. Para asegurar la resistencia estructural, se utilizan rieles que unen las 6 piezas principales, y la unión de la parte superior con la base se realiza mediante tornillos. El prototipo ensamblado se muestra en la Figura 4.

#### B. Electrónica

Se implementa un sistema de control de motores DC con encoders utilizando un microcontrolador ESP32 y drivers de motor DRV8833. El ESP32 es un microcontrolador de 32 bits con una frecuencia de reloj de hasta 240 MHz y capacidad de conectividad Wi-Fi y Bluetooth. Los drivers de motor DRV8833 son capaces de manejar corrientes de hasta 1.5 A por canal y tienen una resolución del ancho de pulsos (PWM,



Fig. 4. Robot móvil omnidireccional

por sus siglas en inglés) de 12 bits. Los motores DC utilizados son del tipo GA25-370, con una velocidad de 60 RPM y una resolución de encoder de 1000 pulsos por revolución. Estos motores tienen una transmisión de engranajes estable, resistente al desgaste y con una gran torsión de transmisión. Una fuente de alimentación externa proporciona energía a los motores y drivers donde las conexiones se muestra en la Figura 5. El microcontrolador ESP32 permite la programación y el control del sistema, mientras que los drivers de motor DRV8833 y los motores DC con encoders trabajan juntos para proporcionar un control preciso del movimiento. La retroalimentación de posición de los encoders permite ajustar el control del movimiento en tiempo real, lo que es esencial para la precisión y la eficiencia del robot. El sistema también cuenta con capacidades de conectividad Wi-Fi y Bluetooth para la comunicación remota y el control del robot.

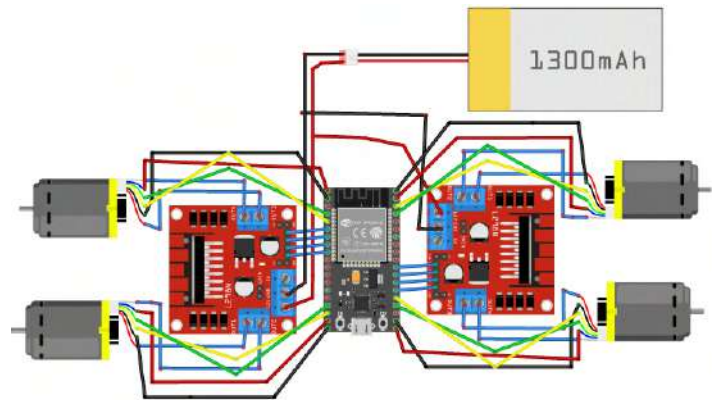


Fig. 5. Esquemático electrónico.

#### C. Comunicación

Se presenta un sistema de comunicación Wi-Fi para controlar el robot omnidireccional utilizando un módulo ESP32 y un programa *Python*. El objetivo es permitir comandos remotos para mover el robot en diferentes direcciones y detener su movimiento. Este sistema es útil en entornos donde se requiere una mayor flexibilidad y movilidad en el control del robot.

La configuración del ESP32 como receptor y el envío de comandos desde *Python* a través de una conexión TCP permiten un control preciso y flexible del robot. La integración de ambos sistemas ofrece una solución robusta y eficiente para

el control remoto de robots móviles. Además, la comunicación Wi-Fi proporciona una mayor movilidad y facilidad de uso, eliminando la necesidad de conexiones físicas y permitiendo el control del robot desde cualquier dispositivo con cámara que tenga acceso a la red. En el diagrama de la Figura 6, se muestran las conexiones del sistema.

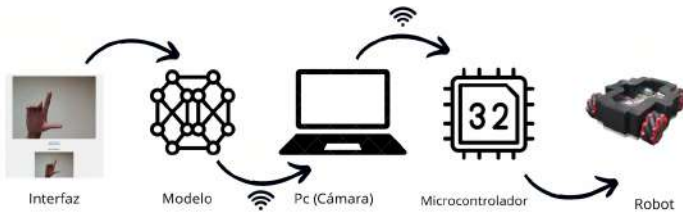


Fig. 6. Diagrama de comunicación

#### IV. RED NEURONAL CONVOLUCIONAL

##### A. Selección de señas

Para la elección de las letras en la lengua de señas (Figura 7), se tomó la letra inicial de cada una de las palabras que se muestran a continuación, debido a que el alfabeto de señas es americano, se optó por elegir las de acuerdo a la primer letra de la palabra en inglés que describe el movimiento.

- *Left*: Para que el vehículo se dirija a la izquierda.
- *Right*: Para que el vehículo se dirija a la derecha.
- *Stop*: Para que el vehículo pare.
- *Go*: Para que el vehículo avance al frente.
- *Turn*: Para que el vehículo realice una vuelta.
- *Diagonal*: Para que el vehículo realice un movimiento en diagonal.

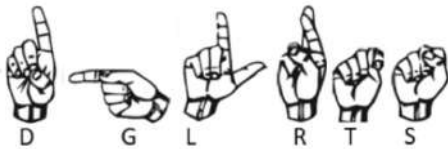


Fig. 7. Señas seleccionadas

##### B. Base de datos

Para el entrenamiento de la red neuronal, se utiliza la base de datos proporcionada por Kaggle [17], la cual contiene 8,442 imágenes. Se seleccionaron seis letras, Figura 7, por lo que la base de datos se redujo a 1,976 imágenes. Para abordar esta limitación, se amplía la base de datos mediante la captura de 4197 fotografías de las 6 señas seleccionadas, las cuales se etiquetan manualmente. En la Tabla I se presenta la cantidad de imágenes de la base de datos, la cual consta de 6,173 imágenes: 5,383 imágenes se destinan al entrenamiento (E), 594 a la validación (V) y 196 a la prueba (P). Estas particiones se establecieron con el objetivo de validar el rendimiento del modelo entrenado.

##### C. Procesamiento de imagen

Previo al entrenamiento, y tomando en cuenta que la red neuronal con la que se iba a obtener el modelo, no es una red preentrenada, se revisó la base de datos obtenida

Seña	Kaggle	Autores	Total	E	V	P	Total
D	379	718	1097	961	106	30	1097
G	345	685	1030	900	99	31	1030
L	346	685	1031	900	99	32	1031
R	291	750	1041	908	100	33	1041
S	314	659	973	847	94	32	973
T	301	700	1001	867	96	38	1001
Total	1976	4197	6173	5383	594	196	6173

TABLE I  
BASE DE DATOS

y se recortaron de forma manual las imágenes en caso de encontrar elementos externos. Posteriormente se procesaron las imágenes, se quitó el fondo, con la librería *remove* en *Python 3.8*, se estableció un tamaño estándar en las imágenes, se convirtieron a escala de grises y a los píxeles que no eran de color negro (con valor de 0) se les sumó un valor de 50, para aumentar el brillo de las imágenes.

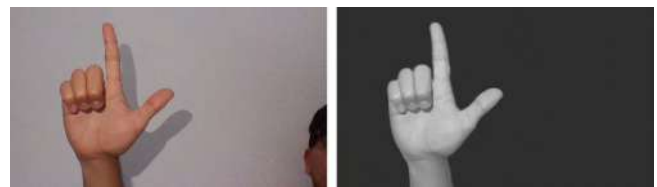


Fig. 8. Comparativa de imagen original e imagen procesada

##### D. Modelo

La red consiste en tres capas convolucionales (*Conv*), cada una seguida de una operación de reducción de tamaño conocida como *Pooling* con un factor de (2,2). Después de las capas convolucionales, los datos se aplanan (*Flatten*) para transformarlos en un vector unidimensional que pueda ser procesado por las capas completamente conectadas (FC). Las capas completamente conectadas incluyen tres capas ocultas con una función de activación de Unidad lineal Rectificada (ReLU). La capa de salida consta de 6 neuronas, correspondiente al número de clases deseadas. Esta capa de salida utiliza una función de activación *softmax* para generar un vector de probabilidades. El valor con mayor probabilidad en este vector determina la clase a la que pertenece la entrada, permitiendo así identificar la letra deseada, como se muestra en la Figura 9.

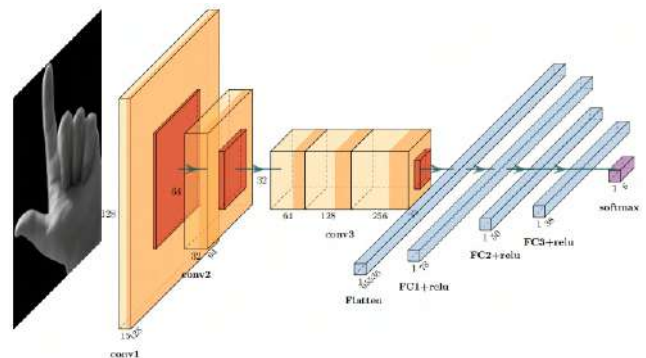


Fig. 9. Red neuronal convolucional.

## V. RESULTADOS

En la Figura 10 la gráfica muestra la evolución de la precisión (accuracy) tanto en el conjunto de entrenamiento como en el conjunto de validación a lo largo de las épocas. La precisión se refiere a la proporción de predicciones correctas realizadas por el modelo. Se observa que la precisión en el conjunto de entrenamiento se estabiliza alrededor de 0.9422, mientras que en el conjunto de validación alcanza aproximadamente 0.9798 en la época 35 lo que indica que tiene una alta precisión. En la Figura 11 se muestra el valor

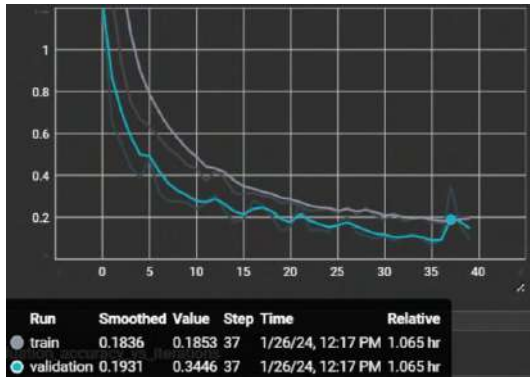


Fig. 10. Gráfica de exactitud

de pérdida durante el entrenamiento, la pérdida en el conjunto de entrenamiento disminuye rápidamente al inicio y luego se estabiliza alrededor de 0.1853 en la época 37. La pérdida en el conjunto de validación también disminuye inicialmente, alcanzando un valor de aproximadamente 0.3446 en la época 37, aunque presenta una ligera variabilidad de 0.01 hacia el final. El valor de pérdida en la última época es de 0.11899 y el

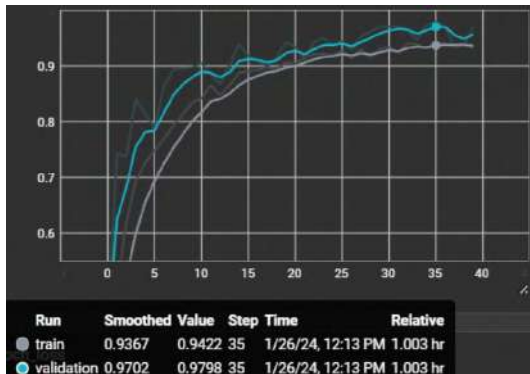


Fig. 11. Gráfica de pérdida

valor de precisión es de 0.94387, como se muestra en la Figura 12, lo que indica una buena capacidad de predicción y un bajo error del modelo. En la Figura 13, las coincidencias entre

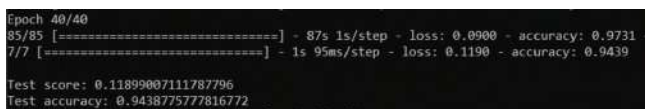


Fig. 12. Valores de pérdida y precisión finales

clases se denotan de acuerdo a la intensidad del color azul. Se

puede observar que los recuadros con mayor intensidad son los de la diagonal, esto indica que la mayoría de las señas en la base de datos coinciden con las imágenes utilizadas para la validación. En la Matriz Confusión (Figura 13) se observa que las letras R y D son en las que el modelo presenta mayor error, ya que tienen cierto parecido al tener un dedo extendido (Figura 7). En la Figura 14 se muestra la evaluación del

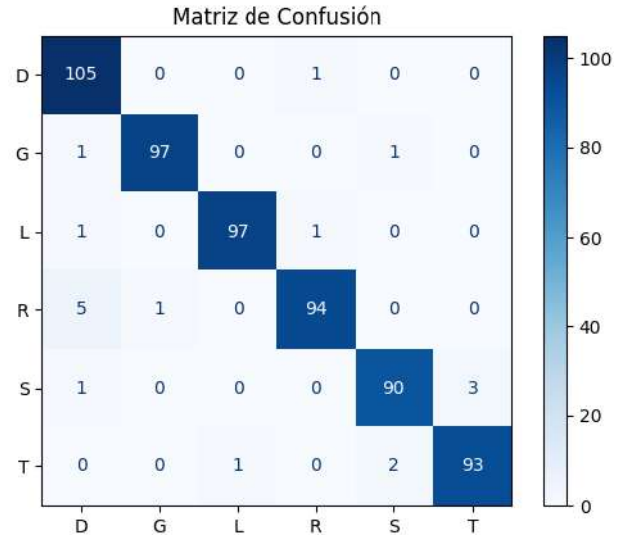


Fig. 13. Evaluación del modelo (validación).

modelo, se tienen 18 errores de 594 imágenes, esto quiere decir que el error es del 3.03030303%. Los resultados obtenidos destacan la eficacia y precisión del modelo. La CNN ha alcanzado una precisión general del 97%, con un rendimiento consistente en todas las clases. Las métricas individuales para cada clase, muestran buenos resultados. Estos muestran la robustez del modelo en la clasificación. Al evaluar con la



Fig. 14. Reporte de validación

carpeta de prueba que se observa en la Figura 16 se obtiene una precisión del 99%. En la matriz de confusión (Figura 15) se observa que hay una sola seña mal clasificada. Lo que demuestra que el modelo es robusto. Una vez obtenido el modelo, se implementa una interfaz con la cual se lee la seña con una cámara, posteriormente procesa la imagen, una vez que el modelo detecta que letra ha sido seleccionada, envía la variable a la ESP32 vía Wi-Fi, como se muestra en el diagrama de la Figura 6. Se modifican las velocidades de los motores con respecto a la dirección deseada para permitir el movimiento

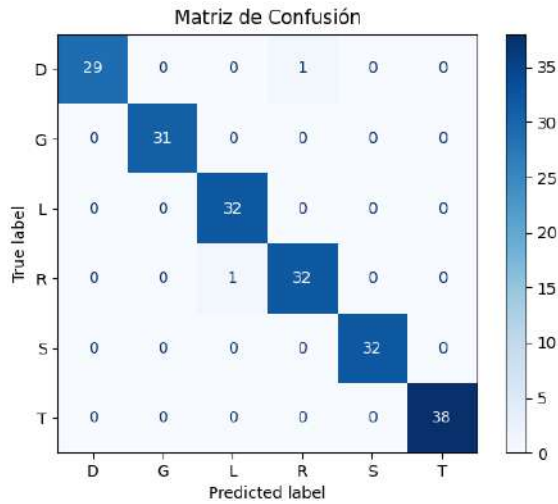


Fig. 15. Evaluación del modelo (prueba).

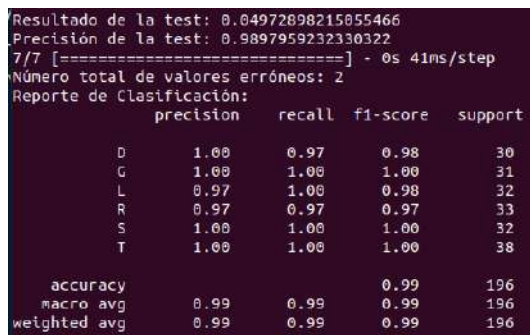


Fig. 16. Reporte de test

necesario en cada llanta omnidireccional, como se indica en la Tabla II.

Seña	Motor 1	Motor 2	Motor 3	Motor 4
D	0.0	-10.0	10.0	0.0
G	10.0	-10.0	10.0	-10.0
L	10.0	10.0	-10.0	-10.0
R	-10.0	-10.0	10.0	10.0
S	0.0	0.0	0.0	0.0
T	10.0	10.0	10.0	10.0

TABLE II

CAMBIOS EN LAS VELOCIDADES DE LOS MOTORES SEGÚN LA DIRECCIÓN DESEADA.

## VI. CONCLUSIONES

El modelo obtenido con la Red Neuronal Convolutiva (CNN) demostró un rendimiento destacado en la clasificación de imágenes, alcanzando una precisión del 96.97%. El modelo ha exhibido una notable exactitud y precisión en un entorno controlado; sin embargo, estos resultados no contemplan la presencia de elementos adicionales, como rostros u objetos inesperados. Una vez que la imagen es capturada por la interfaz, es necesario procesarla para determinar la dirección en la que debe moverse el robot móvil omnidireccional.

El prototipo desarrollado, gracias a su diseño modular, optimiza el ensamblaje y desensamblaje, facilitando tanto su reparación como las adaptaciones futuras. En cuanto a sus funcionalidades, el robot es capaz de recibir y clasificar

correspondencia enviada por los usuarios, permitiendo así un control remoto eficiente. Este sistema no está dirigido exclusivamente a personas con pérdida auditiva o limitaciones de movimiento; también proporciona una solución accesible para usuarios sin restricciones físicas. Este enfoque inclusivo amplía significativamente el rango de aplicaciones del robot, adaptándolo a una amplia variedad de contextos.

Para mejorar el rendimiento y la adaptabilidad del sistema, se recomienda explorar diversas estrategias, como el uso de modelos preentrenados mediante *transfer learning*, la implementación de técnicas de aumento de datos y la integración de sistemas de detección de objetos en tiempo real. Además, para aplicaciones en entornos más complejos o con variaciones imprevistas en las imágenes, es fundamental desarrollar una arquitectura de red neuronal más avanzada, incorporando un mayor número de capas convolucionales. Estos enfoques no solo fortalecerán la robustez del sistema ante variaciones en los datos, sino que también garantizarán una mayor flexibilidad y fiabilidad en diferentes condiciones operativas.

## REFERENCES

- [1] J. Ariño Sustaeta, "Desarrollo de un robot de servicio para la asistencia a personas de la tercera edad," Ph.D. dissertation, Universitat Politècnica de València, 2014.
- [2] A. Campos Siurana, "Diseño e implementación de un vehículo de cuatro ruedas omnidireccionales," Ph.D. dissertation, Universitat Politècnica de València, 2018.
- [3] C. Gordón, P. Encalada, H. Lema, D. León, and C. Peñaherrera, "Autonomous robot kuka youbot navigation based on path planning and traffic signals recognition," in *Proceedings of the Future Technologies Conference (FTC) 2018: Volume 1*. Springer, 2019, pp. 63–78.
- [4] J. Ramos Pérez, "Diseño e implementación de un robot omnidireccional para fines de entretenimiento," 2020.
- [5] F. Espinosa, C. Santos, and J. Sierra-García, "Transporte multi-agv de una carga: estado del arte y propuesta centralizada," *Revista Iberoamericana de Automática e Informática industrial*, vol. 18, no. 1, pp. 82–91, 2020.
- [6] D. Sotelo Castillo, "Programación, control e implementación de un robot móvil de seguimiento reactivo basado en visión artificial y ros," 2023.
- [7] A. V. Pawar, S. J. Rajput, H. K. Soni, and N. R. Joshi, "Effective object detection and tracking for holonomic robot using deep neural architecture," in *2021 IEEE 18th India Council International Conference (INDICON)*, 2021, pp. 1–6.
- [8] Q. Bi, W.-J. Cai, E.-L. Lee, Q.-G. Wang, C.-C. Hang, and Y. Zhang, "Robust identification of first-order plus dead-time model from step response," *Control Engineering Practice*, vol. 7, no. 1, pp. 71–77, 1999.
- [9] K. J. Åström and T. Hägglund, "The future of pid control," *Control engineering practice*, vol. 9, no. 11, pp. 1163–1175, 2001.
- [10] T. E. Marlin and T. Marlin, *Process control: designing processes and control systems for dynamic performance*. McGraw-Hill New York, 1995, vol. 2.
- [11] K. Ogata, *Ingeniería de control moderna*. Pearson educación, 2003.
- [12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [13] J. Martínez Llamas, "Reconocimiento de imágenes mediante redes neuronales convolucionales," 2018.
- [14] K. O'shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.
- [15] A. N. F. Radilla, R. A. B. Flores, E. V. Rodríguez, K. R. P. Daniel, and G. Benitez-Garcia, "Aprendizaje de similitud semántica para el reconocimiento del alfabeto de lengua de señas."
- [16] E. L. Newport and R. P. Meier, "The acquisition of american sign language," in *The crosslinguistic study of language acquisition*. Psychology Press, 2017, pp. 881–938.
- [17] S. Team, "Asl sign language alphabet pictures [minus j, z]," Sep 2020. [Online]. Available: <https://www.kaggle.com/datasets/signnteam/asl-sign-language-pictures-minus-j-z>

# Estimación de la posición para el seguimiento de un robot móvil con ruedas

Servando Encina-Espino  
*TecNM/Instituto Tecnológico de La Laguna*  
Torreón, Coahuila, México  
m.sencinae@correo.itlalaguna.edu.mx

Alejandro Enrique Dzul López  
*TecNM/Instituto Tecnológico de La Laguna*  
Torreón, Coahuila, México  
aedzull@lalaguna.tecnm.mx

Pedro Castillo  
*Sorbonne Universités*  
Université de Technologie de Compiègne  
Compiègne, France  
castillo@hds.utc.fr

Héctor Ríos Barajas  
*TecNM/Instituto Tecnológico de La Laguna*  
Torreón, Coahuila, México  
*CONAHCYT-IxM*  
Ciudad de México, México  
hriosb@lalaguna.tecnm.mx

**Resumen**—Este trabajo presenta el uso del filtro de Kalman como una herramienta matemática para la estimación de la posición de un robot móvil con ruedas (RMR), el cual es seguido por un vehículo aéreo del tipo Quad-Rotor, basado en la estimación de la posición del RMR. La estimación es realizada a través del procesamiento de imágenes capturadas por una cámara montada en el Quad-Rotor. Una vez obtenida la estimación, se utilizan leyes de control implementadas en el Quad-Rotor. Los resultados experimentales obtenidos muestran la eficacia de la propuesta.

**Index Terms**—Filtro de Kalman, control de seguimiento, Quad-Rotor.

## I. INTRODUCCIÓN

La investigación sobre los sistemas robóticos móviles ha estado creciendo día a día, debido principalmente al interés, no solo científico, sino también en el área de negocios, logrando una continua generación de aplicaciones civiles y militares para estos sistemas. Algunas de estas aplicaciones se pueden encontrar en juguetes, en tomas de fotografía o de video para la industria cinematográfica, en transporte de materiales en la industria, como apoyo en situaciones de incendios para su control, detección o supervisión; también como apoyo en búsqueda y rescate, en inspección de estructuras o superficies, en seguimiento de objetos, etc. El interés científico obviamente va de la mano con la generación de aplicaciones, dado que siempre se está en busca de mejoras en los materiales para su construcción (más ligeros, más resistentes y más económicos), en los componentes electrónicos (más pequeños, más precisos, con menor consumo de energía, etc.), así mismo existe un interés en la parte teórica, por ejemplo, buscando cómo representarlos de una manera matemática más completa, y generando algoritmos de control robustos que garanticen su correcto funcionamiento ante cualquier tipo de contratiempo.

Cuando se requiere realizar el estudio del seguimiento de un objeto, primeramente se deberá definir el tipo de objeto

que se seguirá, el método que se utilizará para obtener su posición, el tipo de sistema robótico que tendrá el rol de seguidor, además del tipo y ambiente donde se desenvolverán (terrestre, aéreo, etc.). En este artículo, se presentará el caso de un robot aéreo del tipo Quad-Rotor que sigue a un robot móvil con ruedas.

El auge en la investigación de sistemas robóticos móviles han generado el uso de una amplia variedad de temas relacionados a cada aplicación (filtrado de Kalman, procesamiento de imágenes, diseño de controladores, generación de trayectorias, o alguna combinación de estos). A continuación, se mencionará una pequeña muestra de publicaciones ligados a este tipo de temas. El uso del filtro de Kalman, para la estimación de parámetros, fue utilizado en [1], en donde se estima la posición de una pelota, obteniendo dicha posición mediante el análisis de filtrado y predicción de imágenes de video. En [2], se presenta el diseño de un sistema de visión embebido en un vehículo aéreo con el fin de realizar el seguimiento de un objetivo; en éste se describe el desarrollo de un algoritmo robusto para la detección y estimación de la trayectoria, así como la predicción de la siguiente posición y así, mediante un servomotor, ajustar la cámara y mantener el objetivo en el centro de la imagen en todo momento; uno de los inconvenientes que presenta este trabajo es la necesidad de agregar otro módulo para realizar el manejo y análisis de las imágenes, debido al costo computacional necesario para ejecutar el algoritmo, lo cual no siempre es posible de implementar en otros sistemas con bajos recursos de procesamiento. Para abordar el tema de seguimiento de objetivos, uno de los esquemas comúnmente utilizados es el de Líder-Seguidor, en el cual se conoce la cinemática del líder (objetivo) para obtener los errores de posición del seguidor. En nuestro caso, no se considera la información de la cinemática del objetivo. En [3], los autores presentan el diseño de un controlador por backstepping para la formación Líder-Seguidor de un vehículo aéreo que sigue

a un vehículo terrestre. El trabajo [4] propone un algoritmo, basado en geometría, para la detección y seguimiento de un objeto esférico a través de una cámara y un sistema de medición inercial, ambos montados en un Quad-Rotor; una de las diferencias del trabajo es la utilización de la dinámica del objetivo y una planeación de trayectoria agresiva, considerando la restricción del campo de visión de la cámara. Una de las maneras en las que comenzó el seguimiento de objetivos, mediante el uso de un Quad-Rotor con una cámara, consistió en partir de una posición inicial, para después crear una trayectoria que terminara en un punto dado, como se explica en [5], donde se utilizó un algoritmo para crear una trayectoria que llegara a una posición determinada, calculada mediante una cámara, en el menor tiempo posible y teniendo en cuenta el campo de visión de la cámara. En nuestro caso, la trayectoria se va generando con el seguimiento de un objetivo móvil. En [6], se propone la planeación de trayectorias para un Quad-Rotor al seguir un objetivo móvil, donde la trayectoria está libre de colisiones, y se utiliza información visual para ello. En [7], los autores presentan el desarrollo de un sistema embarcado de visión, en un Quad-Rotor, para el seguimiento de objetivos, mediante la detección de su color, teniendo como limitación la sencillez del fondo y con colores distintos al del objetivo, siendo estos objetivos estacionarios. En [8], se presenta el diseño de un controlador robusto para el seguimiento de un objetivo terrestre móvil, donde la ley de control robusta considera el tiempo de retraso en la transmisión de la posición del objetivo, sin embargo no fue validado experimentalmente.

En el presente trabajo se utiliza el filtro de Kalman como un estimador de la trayectoria descrita por un objetivo, donde la posición es obtenida a partir del procesamiento de imágenes de una cámara instalada sobre un Quad-Rotor. Se considera que el procesamiento de las imágenes entra en funcionamiento cuando el RMR está siendo detectado. La estimación es inyectada a controladores de tipo PD y PID para la posición del Quad-Rotor, mientras que para

su orientación se utilizaron controladores PD. Mediante la validación experimental, se muestran los resultados de la estimación de la posición para el seguimiento del RMR por parte del Quad-Rotor.

La organización del resto del documento es el siguiente: la sección II muestra la descripción del problema a tratar así como algunos conceptos básicos de visión utilizados en este trabajo. En la sección III se explica cómo se realizó la estimación de la posición del robot móvil con ruedas. El control utilizado para el seguimiento es descrito en la sección IV. Los resultados experimentales, obtenidos de la estimación de la trayectoria, son presentados en la sección V. Finalmente, las conclusiones de este artículo están dadas en la sección VI.

## II. DESCRIPCIÓN DEL PROBLEMA Y TRATAMIENTO DE LA IMAGEN

### II-A. Problema a resolver

El sistema a utilizar está compuesto por un robot aéreo, del tipo Quad-Rotor, y un robot móvil con ruedas, del tipo diferencial (ver Figura 1). Con el fin de que el Quad-Rotor realice el seguimiento del RMR, se utiliza una cámara montada en el Quad-Rotor para que ésta detecte un marcador visual, situado sobre el RMR, y así poder obtener sus coordenadas  $X - Y$ , con las cuales se generan las consignas deseadas del Quad-Rotor, tomándose en consideración que la altura se mantendrá fija. Por otro lado, es bien sabido que el procesamiento de las imágenes, así como la transmisión de la posición al controlador, induce un retraso de tiempo entre el momento de adquisición de una imagen y la obtención de las coordenadas  $X - Y$ . Ante ello, se propone la utilización del algoritmo del filtro de Kalman como un estimador en avance de las coordenadas y sus dos primeras derivadas. Esta información es la que finalmente se inyectará a los controladores del Quad-Rotor, buscando obtener un seguimiento óptimo del RMR.

### II-B. Procesamiento de imágenes

Una señal de video está compuesta por una sucesión de imágenes. Por lo tanto, cada imagen es procesada con el propósito de buscar y obtener alguna característica de interés. Para realizar el procesamiento de la imagen, ésta debe estar en un formato digital, es decir, en una representación numérica, que por lo general son arreglos matriciales. Cada elemento de la matriz contiene información sobre el pixel de la imagen. El formato digital de la imagen que se utilizará es el llamado RGB (Red-Green-Blue), ver Figura 2, donde cada casilla posee un valor entero entre el rango de 0 a 255. Basándose en esta información numérica, se pueden entonces realizar diversas operaciones para el procesamiento de las imágenes (filtrado, detección de bordes, etc.). Estas operaciones se realizarán utilizando una librería llamada OpenCV [9]. Para el procesamiento de la imagen, se considera que la cámara obtiene imágenes de la vista superior del RMR, por lo tanto, es requisito que el Quad-Rotor esté sobre el RMR o en su entorno próximo.

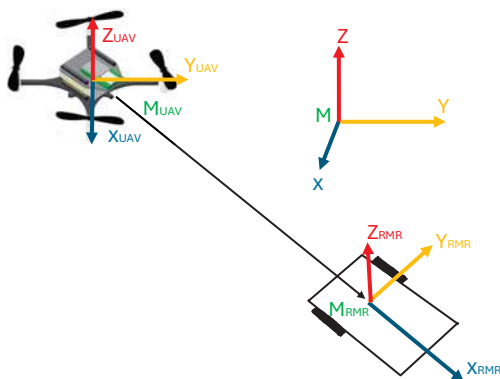


Figura 1. Sistema robótico.

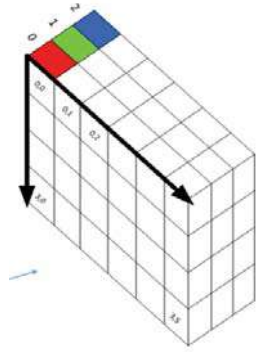


Figura 2. Representación matricial del formato RGB.

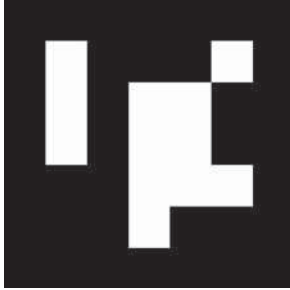


Figura 3. Marcador ArUco seleccionado.

Al RMR se le colocó en la parte superior un marcador binario de referencia, es decir, una imagen específica que nos permite identificarla en la escena. La Figura 3 muestra el marcador utilizado, el cual es un marcador cuadrado compuesto por un borde negro que, en su interior, tiene un patrón binario que evita singularidades al momento de calcular la orientación; gracias a estas características, se puede obtener la orientación y orientación del objeto con un solo marcador, lo que permite que la identificación sea realizada de una forma rápida. El marcador seleccionado se encuentra registrado en una librería, llamada ArUco [10], por lo que su procesamiento consume un tiempo menor. Una vez que se ha detectado el marcador en la imagen, se procede a realizar el cálculo de la posición del objetivo dentro de la imagen, utilizando diferentes algoritmos de la librería OpenCV. Considérese también que la cámara ya se encuentra calibrada y se conocen las dimensiones reales del marcador.

### III. ESTIMACIÓN DE LA POSICIÓN DEL OBJETIVO

La estimación de la posición del RMR se realizará utilizando el procesamiento de las imágenes proporcionadas por una cámara, así como un algoritmo basado en el filtro de Kalman. Para ello, se tratarán a continuación algunos conceptos de dicho filtro.

El filtro de Kalman ha sido ampliamente utilizado, por ejemplo, para la estimación de estados de un sistema, donde una de sus primeras aplicaciones fue en los sistemas de

navegación, buscando obtener un rendimiento óptimo de los sensores; este filtro se basa en modelos estadísticos para realizar la estimación del estado, los cuales van ponderando los valores obtenidos al tomar en cuenta el valor anterior.

El funcionamiento del filtro de Kalman se basa en un ciclo de predicción y corrección, en el cual se obtienen predicciones del estado del sistema, mediante algunas ecuaciones, así como el cálculo de la incertidumbre, para posteriormente realizar la actualización de estos valores con la medición que se obtiene en cada instante, ajustando así el valor de una matriz, llamada de covarianza y un valor conocido como la ganancia de Kalman. El filtro de Kalman está compuesto por 5 ecuaciones.

Para la siguiente explicación, considérese sólo la estimación de la coordenada  $x$ . En la primera etapa, llamada etapa de predicción, se tienen las siguientes ecuaciones:

- Ecuación de extrapolación de estado

$$\hat{\mathbf{x}}_{t_{k+1}, t_k} = \mathbf{F}\hat{\mathbf{x}}_{t_k, t_k} + \mathbf{w}_{t_k}, \quad (1)$$

donde  $\mathbf{x} \in \mathbb{R}^3$  es el vector de estado ligado a la posición  $x$  del RMR, así como su velocidad y su aceleración, quedando de la siguiente manera  $\mathbf{x} = [x \ \dot{x} \ \ddot{x}]^T$ .  $\hat{\mathbf{x}}$  es la estimación del vector de estado, entonces  $\hat{\mathbf{x}}_{t_{k+1}, t_k}$  corresponderá a la predicción del estado en la iteración  $k$ , y  $\hat{\mathbf{x}}_{t_k, t_k}$  es el estimado actual del estado.  $\mathbf{F} \in \mathbb{R}^{3 \times 3}$  es la matriz de transición de estados:

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta_t & \frac{\Delta_t^2}{2} \\ 0 & 1 & \Delta_t \\ 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

con  $\Delta_t \in \mathbb{R}_+$  como el periodo de tiempo entre cada iteración. Por último,  $\mathbf{w}_{t_k} \in \mathbb{R}^3$  denota la perturbación del sistema en el instante  $k$ , dicha perturbación puede ser considerada como el error generado por el retraso en el procesamiento de la imagen y el cálculo de la posición.

- Ecuación de extrapolación de covarianza

$$\mathbf{P}_{t_{k+1}, t_k} = \mathbf{F}\mathbf{P}_{t_k, t_k}\mathbf{F}^T + \mathbf{Q}, \quad (3)$$

donde  $\mathbf{P}_{t_{k+1}, t_k} \in \mathbb{R}^{3 \times 3}$  representa la predicción de la matriz de covarianza de la siguiente estimación del estado,  $\mathbf{P}_{t_k, t_k}$  es la matriz de covarianza calculada en el instante  $k$ , y  $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$  denota la matriz que representa la variación en la perturbación del sistema:

$$\mathbf{Q} = \begin{bmatrix} \frac{\Delta_t^4}{4} & \frac{\Delta_t^3}{2} & \frac{\Delta_t^2}{2} \\ \frac{\Delta_t^3}{2} & \Delta_t^2 & \Delta_t \\ \frac{\Delta_t^2}{2} & \Delta_t & 1 \end{bmatrix}. \quad (4)$$

- Ecuación para la actualización de la ganancia de Kalman

$$\mathbf{k}_{t_k} = \mathbf{P}_{t_k, t_{k-1}}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{t_k, t_{k-1}}\mathbf{H}^T + \mathbf{R})^{-1}, \quad (5)$$

donde  $\mathbf{R} \in \mathbb{R}$  es el ruido de medición de la matriz de covarianza. En este trabajo, al sólo obtener la medición de una variable de estado, se considera como  $\mathbf{R} = 1$ , lo cual implica que se le da el valor total para compensar el

error en la estimación de la medición.  $\mathbf{P}_{t_k, t_{k-1}}$  representa la matriz de covarianza del instante anterior.  $\mathbf{H} \in \mathbb{R}^{1 \times 3}$  representa a la matriz de observación relacionada a la medición obtenida de los estados; en este caso, sólo corresponde al valor de la coordenada medida  $x$ :

$$\mathbf{H} = [1 \ 0 \ 0]. \quad (6)$$

- Ecuación de actualización o corrección

$$\hat{\mathbf{x}}_{t_k, t_k} = \hat{\mathbf{x}}_{t_k, t_{k-1}} + \mathbf{k}_{t_k} (\mathbf{z}_{t_k} - \mathbf{H} \hat{\mathbf{x}}_{t_k, t_{k-1}}), \quad (7)$$

donde  $\mathbf{k}_{t_k} \in \mathbb{R}^3$  denota el vector de ganancias de Kalman,  $\mathbf{z}_{t_k} \in \mathbb{R}^3$  corresponde a un vector con los valores de las mediciones realizadas de los estados en el instante  $k$ .

- Ecuación de actualización de la matriz de covarianza

$$\mathbf{P}_{t_k, t_k} = \mathbf{P}_{t_k, t_{k-1}} - \mathbf{k}_{t_k} \mathbf{H} \mathbf{P}_{t_k, t_{k-1}}. \quad (8)$$

Como puede observarse, las ecuaciones están expresadas de manera discreta, sin embargo, se puede considerar que el cálculo se realiza a una velocidad lo suficientemente rápida, por lo cual se puede suponer que los valores obtenidos de las coordenadas del RMR son de tipo continuo. Asimismo, se considera que el periodo de tiempo entre cada iteración ( $\Delta_t$ ) es constante.

La Figura 4 muestra el esquema utilizado para el cálculo de la estimación de la posición del RMR. Por lo tanto, posteriormente al uso del filtro de Kalman, se obtiene como datos deseados a:  $\mathbf{x} = [x_d \ \dot{x}_d \ \ddot{x}_d]^\top$ , y realizando los mismos pasos para el cálculo de la coordenada  $y$ , entonces se obtiene de igual manera  $\mathbf{y} = [y_d \ \dot{y}_d \ \ddot{y}_d]^\top$ .

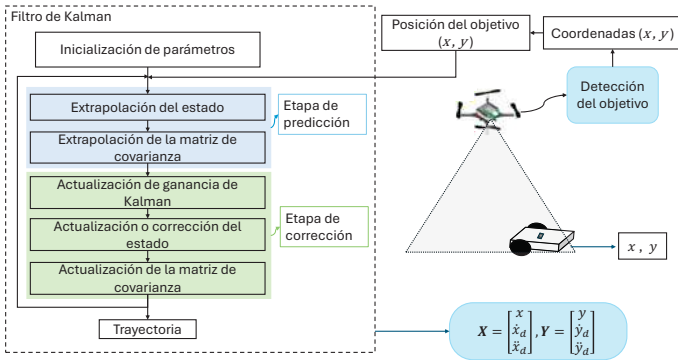


Figura 4. Esquema para la estimación de la posición del RMR.

#### IV. CONTROLADOR DEL QUAD-ROTOR

Controladores del tipo Proporcional-Derivativo y Proporcional-Integral-Derivativo fueron implementados en el Quad-Rotor con el fin de lograr el seguimiento en

posición (el lector es referido a [12] y [13] para una mayor información sobre el modelado y los controladores):

$$\nu_x = K_{0_x} \epsilon_x + K_{2_x} \epsilon_{\dot{x}}, \quad (9)$$

$$\nu_y = K_{0_y} \epsilon_y + K_{2_y} \epsilon_{\dot{y}}, \quad (10)$$

$$\nu_z = K_{0_z} \epsilon_z + K_{1_z} \epsilon_{\dot{z}} + K_{2_z} \epsilon_{\ddot{z}}, \quad (11)$$

donde  $K_{j_x}, K_{j_y}, K_{j_z}$ , con  $j = \overline{0, 2}$ , tienen la siguiente forma:

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ K_{0_p} & K_{1_p} & K_{2_p} \end{pmatrix}, \quad (12)$$

con  $p = x, y, z$ , y son valores asignados tal que cumplan Hurwitz (valores propios con parte real negativa).  $\nu_x, \nu_y, \nu_z$  son controles virtuales.  $\epsilon_p = p_d - p$ ,  $e_p = \int_0^t \epsilon_p(\tau) d\tau$ , y  $\epsilon_p = \dot{e}_p$ . El empuje total del Quad-Rotor está definido como:

$$u = m \sqrt{\nu_x^2 + \nu_y^2 + (\nu_z + g)^2}, \quad (13)$$

donde  $m$  denota la masa total del Quad-Rotor, y  $g$  representa la aceleración gravitacional.

Por otro lado, los momentos angulares  $\tau = [\tau_\phi \ \tau_\theta \ \tau_\psi]^\top$ , poseen una estructura Proporcional-Derivativa:

$$\tau_\phi = K_{p_\phi} \epsilon_\phi + K_{d_\phi} \epsilon_{\dot{\phi}}, \quad (14)$$

$$\tau_\theta = K_{p_\theta} \epsilon_\theta + K_{d_\theta} \epsilon_{\dot{\theta}}, \quad (15)$$

$$\tau_\psi = K_{p_\psi} \epsilon_\psi + K_{d_\psi} \epsilon_{\dot{\psi}}, \quad (16)$$

donde  $k_{p_i}, k_{d_i}$ , con  $i = \overline{\phi, \theta, \psi}$ , tiene la siguiente forma:

$$\begin{pmatrix} 0 & 1 \\ K_{p_i} & K_{d_i} \end{pmatrix} \quad (17)$$

con  $i = \phi, \theta, \psi$  y con valores asignados tal que cumplan Hurwitz.

#### V. RESULTADOS EXPERIMENTALES



Figura 5. Quad-Rotor y RMR utilizados.

Con el fin de validar el uso del filtro de Kalman en el cálculo de las coordenadas  $x - y$  del RMR, se utilizó como Quad-Rotor a un vehículo Parrot AR Drone 2 y como RMR a un

Cuadro I  
VALORES DE GANANCIAS

$p$	$K_{0_p}$	$K_{1_p}$	$K_{2_p}$	$i$	$K_{p_i}$	$K_{d_i}$
$x$	0.2	0	0.2	$\phi$	15	0.1
$y$	0.2	0	0.2	$\theta$	15	0.5
$z$	0.3	0.2	0.1	$\psi$	1	0.1

Parrot minidrone Jumping Sumo, ver Figura 5. El Laboratorio HEUDIASYC, de la Univesité de Technologie de Compiègne, en Francia, cuenta con dichos vehículos, donde se obtuvieron los resultados experimentales. El software de los vehículos fue modificado, instalándose un software llamado FI-AIR de código abierto [11], que se ejecuta con el sistema operativo Linux. El marcador ArUco, de la Figura 3, fue colocado en la parte superior del RMR.

Para la realización de los experimentos, el RMR describe una trayectoria dada por  $x_d = \cos(0.41 \frac{\pi}{11})$ , y  $y_d = \sin(0.82 \frac{\pi}{11})$ . Es importante especificar que el Quad-Rotor se encuentra en una posición cercana al RMR para iniciar el algoritmo de estimación (detección del marcador). El Cuadro I muestra los valores de las ganancias utilizadas en los controladores, los cuales fueron sintonizados heurísticamente.

Los valores iniciales del algoritmo de estimación son:  $\Delta_t = 0.005[s]$ , y

$$P_0^0 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}. \quad (18)$$

Los resultados del experimento se muestran en las Figuras 6-11. El video del experimento puede observarse en: <https://youtu.be/yzC1yd-pvew?si=NtFd3MINRbg5TGiN>. En la Figura 6 se muestran las trayectorias 3D realizadas por el Quad-Rotor y el RMR, observándose que el Quad-Rotor sigue al RMR en su recorrido. El mismo recorrido puede observarse en la Figura 7, pero en plano  $X - Y$ . En ambas figuras se aprecia el arranque del experimento, donde el Quad-Rotor se posiciona cerca del RMR para iniciar el algoritmo una vez que el marcador es detectado.

En la Figura 8, se muestra el comportamiento del Quad-Rotor en la coordenada  $z$ , llegando a estabilizarse cerca de la altura deseada, presentando un pequeño error con respecto a la referencia dada. Por otro lado, los errores de seguimiento, para el caso del plano  $X - Y$ , se muestran en la Figura 9, donde se puede apreciar una variación periódica, debido a los diversos giros cerrados que son realizados por el RMR, modificando la estimación de la posición en ciertos puntos, sin embargo, el Quad-Rotor logra seguir al RMR en su desplazamiento.

Para evaluar el desempeño del seguimiento del Quad-Rotor y el RMR, se utilizó el error cuadrático medio (RMSE), el cual sirve para evaluar la diferencia entre el valor deseado y el valor actual. Su cálculo se realizó utilizando la siguiente formula:  $e_{X_{rms}} = \sqrt{\frac{1}{N} \sum_{i=0}^N (x_{di} - x_i)^2}$ , donde  $N$  es el

número de elementos en  $x_{(i)}$ , y  $x$  representa la posición en la coordenada en el eje  $X$ . En la Figura 10 se presentan los valores RMS para la posición en cada coordenada  $X, Y, Z$ .

Finalmente, en la Figura 11 se observan las señales de control, donde el empuje total, ligado a la coordenada  $z$ , se mantiene más estable debido en parte a que la altura deseada se definió como constante.

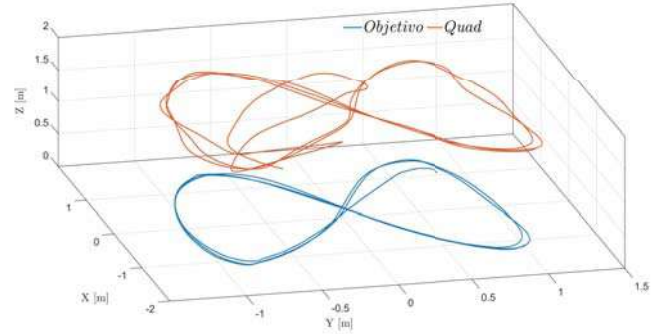


Figura 6. Gráfica 3D de las trayectorias descritas por el Quad-Rotor y el RMR.

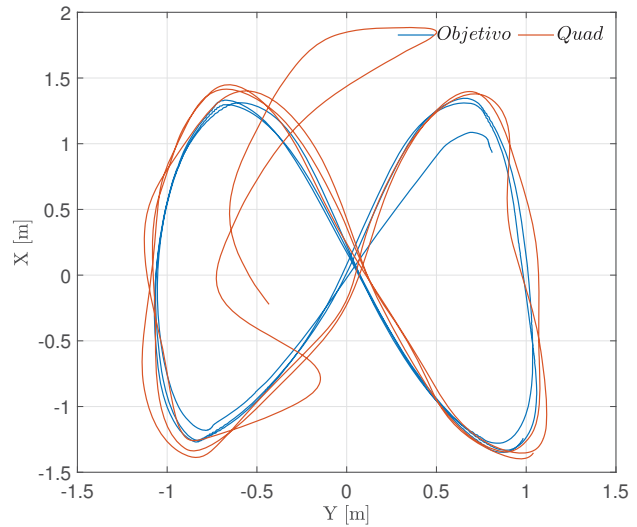


Figura 7. Gráfica en  $X - Y$  de las trayectorias descritas por el Quad-Rotor y el RMR.

## VI. CONCLUSIONES

En este artículo se presenta el uso del filtro de Kalman para la estimación de la posición de un RMR. Para ello, el filtro de Kalman recibe como datos los valores de la posición obtenidas por un procesamiento de imágenes de una cámara montada en un Quad-Rotor. El filtro de Kalman sirve como un estimador si se considera que existe cierto error en la obtención de la posición, así como un error debido al tiempo transcurrido entre el procesamiento de la imagen y el cálculo de la posición.

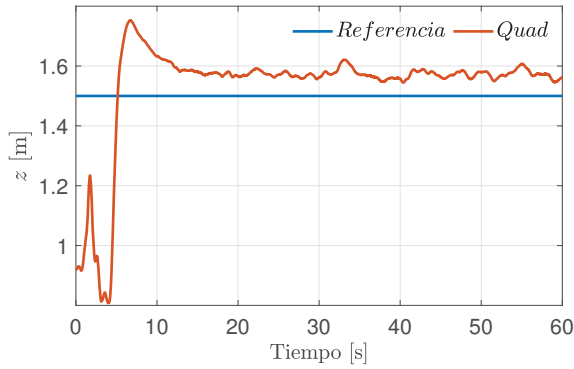


Figura 8. Error de seguimiento en Z.

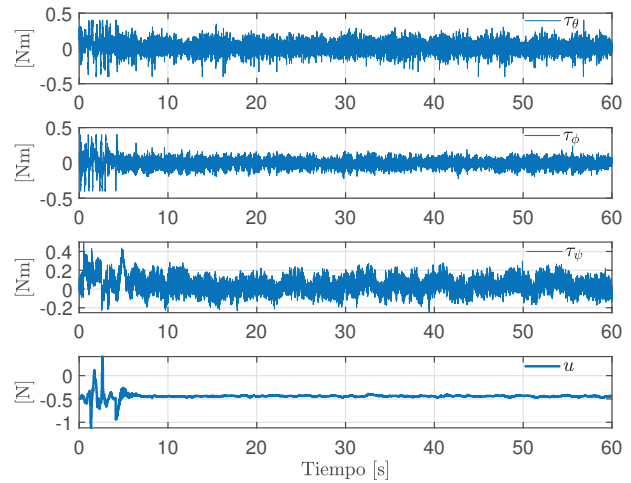


Figura 11. Señales de control.

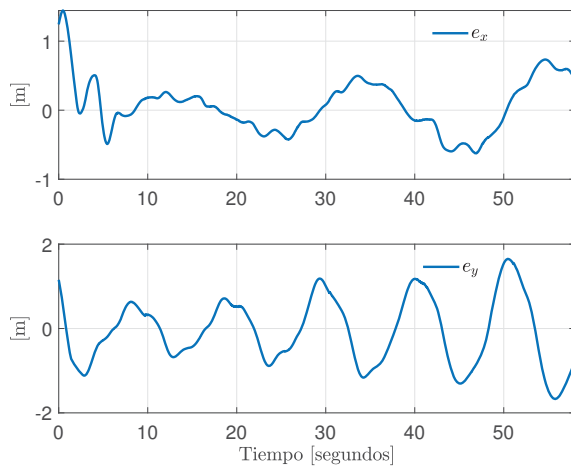


Figura 9. Errores de seguimiento en  $x$  (gráfica superior) y en  $y$  (gráfica inferior).

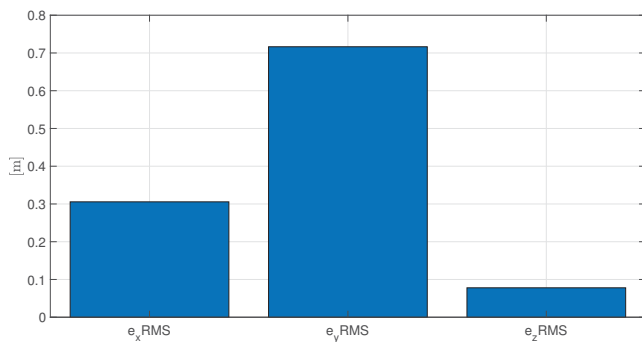


Figura 10. Desempeño de la tarea de seguimiento.

A través de los resultados experimentales, se observó que el Quad-Rotor logra satisfactoriamente el seguimiento del RMR. Como trabajo futuro, se plantea realizar una comparativa del filtro Kalman con otros algoritmos de estimación, así como el análisis con respecto a otros controladores.

## REFERENCIAS

- [1] P. Gunjal, B. Gunjal, H. Shinde, S. Vanam, and S. Aher, "Moving Object Tracking using Kalman Filter", 2018 International Conference On Advances in Communication and Computing Technology (ICACCT), Sangamner, India 08-09 February 2018.
- [2] F. Lin, X. Dong, B.M. Chen, K.Y. Lum, and T. Heng Lee, "A robust real-time embedded vision system on an unmanned rotorcraft for ground target following", IEEE Transactions on Industrial Electronics, Vol. 59, pp. 1038–1049.
- [3] M.A. Vallejo-Alarcón, R. Castro-Linares, and M. Velasco-Villa, "Unicycle-Type Robot & Quadrotor Leader-Follower Formation Backstepping Control", IFAC-PapersOnLine 48-19, pp. 051–056, 2015
- [4] J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar, "Autonomous Flight for Detection, Localization, and Tracking of Moving Targets with a Small Quadrotor", IEEE Robotics and Automation Letters, Vol 2, Issue 3, pp 1762–1769, July 2017.
- [5] B. Penin, R. Spica, P. Giordano, and F. Chaumette, "Vision-Based Minimum-Time Trajectory Generation for a Quadrotor UAV", 2017 IEEE/RSK International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada 24-28 September 2017.
- [6] J. Chen, T. Liu, and S. Shen, "Tracking a moving target in cluttered environments using a quadrotor", 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea 09-14 October 2016.
- [7] A. G. Kendall, N. N. Salvapantuala, and K. A. Stol, "On-board object tracking control of a quadcopter with monocular vision", in Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, US, 27–30 May 2014.
- [8] R. Falcón, H. Ríos, D. Efimov, and A. Dzul, "A Robust Control Strategy for Target Tracking using a Quad-Rotor", 2022 Congreso Nacional de Control Automático (CNCA 2022), pp. 300–305, Tuxtla Gutiérrez, Chiapas, 12-14 de octubre de 2022.
- [9] OpenCV Team, Website: <https://opencv.org/releases/>, agosto 2024.
- [10] OpenCV Team, Website: [https://docs.opencv.org/4.x/d5/dae/tutorial\\_aruco\\_detection.html](https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html), agosto 2024.
- [11] G. Sanahuja, G. Bayard, C. Richard, "FI-AIR - Framework libre AIR: Framework d'aide au développement d'applications pour drones", Website: <https://gitlab.utc.fr/uav-hds/flair/flair-src>, agosto 2024.

- [12] H. Ríos, R. Falcón, O. González, and A. Dzul, "Continuous Sliding-Modes Control Strategies for Quad-Rotor Robust Tracking: Real-Time Application" IEEE Transactions on Industrial Electronics, ISSN: 0278-0046, Vol. 66, Issue: 02, pp. 1264-1272, 2019.
- [13] J-C Rodríguez-Cervantes, A. Dzul, H. Ríos, "Formation Control Design for Multi-Quad-Rotors with Experimental Validation", 2023 Congreso Nacional de Control Automático (CNCA 2023), AMCA, pp. 443-448, ISSN: 2594-2492, Acapulco, Guerrero, Mexico, 25-27 de octubre del 2023.

# Generación de Trayectorias en Línea Para el Control de Caminado a Través del Momento Angular

Emanuel Marquez-Acosta  
Tecnológico Nacional de México  
Instituto Tecnológico de la Laguna  
Torreón, Coahuila, 27000  
m.jemarqueza@correo.itlalaguna.edu.mx

Víctor De-León-Gómez  
Tecnológico Nacional de México  
Instituto Tecnológico de la Laguna  
Torreón, Coahuila, 27000  
vsdeleong@lalaguna.tecnm.mx

Víctor Santibáñez  
Tecnológico Nacional de México  
Instituto Tecnológico de la Laguna  
Torreón, Coahuila, 27000  
vasantibanezd@lalaguna.tecnm.mx

**Abstract**—En este trabajo se presenta un nuevo generador de trayectorias para el centro de masa (CoM) en el caminado de un robot humanoide en línea, donde el objetivo es llevar el robot a un momento angular (sobre el pie de soporte) deseado al final del paso a través de la modificación del punto de momento cero (ZMP) del robot. Se formula un problema de optimización usando la dinámica del modelo de péndulo invertido lineal (LIP) para predecir el momento angular al final del paso. Como realimentación del generador se propone un filtro de Kalman lineal que fusiona información de la central inercial y de los encoders de los motores para estimar la posición y velocidad del CoM. Las simulaciones muestran que el generador presentado genera ciclos límite estables para las trayectorias del CoM. Los resultados incluyen experimentos en el robot NAO para validar el funcionamiento de nuestra estrategia.

**Index Terms**—robot humanoide, locomoción bípeda, NAO, LIP, Momento angular, filtro de Kalman, estimación de estados

## I. INTRODUCCIÓN

En el caminado de robots, la generación de trayectorias de caminado, generalmente descritas por trayectorias del centro de masa (CoM) y del pie libre, deben ser diseñadas para garantizar que el robot recorra una distancia caminando sin sufrir una caída. Estas trayectorias pueden ser generadas fuera de línea usando un modelo dinámico complejo y un criterio de estabilidad (por ejemplo el punto de momento cero o ZMP [1]) que garantice que el robot no caerá durante el caminado [2]. El problema que surge al generar trayectorias fuera de línea es la dificultad de considerar perturbaciones o cambios en el entorno, lo que disminuye su robustez ante la mayoría de aplicaciones de un robot al caminar. Una opción más robusta es la generación de trayectorias de caminado en línea.

Al generar trayectorias en línea se tiene más versatilidad ante factores externos, la restricción de tener un modelo dinámico preciso se relaja y un modelo simplificado que capture la dinámica esencial al caminar puede ser usado. Como desventaja, al generar trayectorias en línea el costo computacional aumenta, lo que puede impedir su implementación práctica.

La mayoría de trabajos de generación de trayectorias tienen como objetivo seguir una trayectoria de referencia del ZMP [3]–[5] o de *capture point* [6], [7], las cuales se definen de

Este proyecto ha sido parcialmente financiado por los proyectos TecNM, CONACyT 134534, Beca CONACyT y Red RICCA.

manera heurística y no siempre son sencillas de definir, en especial si la trayectoria del pie de aterrizaje está en constante cambio. El generador presentado en este trabajo no requiere de una trayectoria definida del ZMP, sólo requiere definir un momento angular deseado al final del paso, lo que es una condición mucho más sencilla de definir.

El generador propuesto obtiene una predicción sobre el momento angular sobre el pie de soporte al final del paso usando el modelo de péndulo invertido lineal (LIP) [8], se propone un proceso de optimización que encuentra el ZMP que reduce el error entre el momento angular al final del paso y un momento angular deseado. El generador produce ciclos límite estables para la posición y velocidad del CoM en el modelo LIP. Además el generador es implementado en un modelo dinámico que considera la dinámica completa (el modelo esencial [9]), dando como resultado de igual forma un ciclo límite estable.

Además, en este artículo se presenta un filtro de Kalman como estimador de la posición y velocidad del CoM del robot. Un trabajo muy importante en la estimación de estados de robots con piernas es el de [10], en donde se presentó un filtro de Kalman Extendido para la estimación de estados de un robot cuadrúpedo fusionando información de la unidad de medición inercial (IMU) y de los encoders de los motores para estimar la posición y orientación del cuerpo del robot. En [11] este mismo estimador fue aplicado a un robot humanoide simplificando el diseño del estimador al incluir las restricciones de orientación del pie con el suelo. Basado en estos trabajos, en [12] se desarrolló un filtro de Kalman lineal al desacoplar la estimación de la orientación del cuerpo del robot de la estimación de la posición y velocidad del CoM. En [13], un estimador muy parecido al presentado en [12] fue implementado en el robot MIT Cheetah agregando la velocidad de las piernas del robot en las mediciones del filtro. En el caso de este artículo se optó por el filtro mostrado en [12] con una modificación al incluir el bias del acelerómetro en el vector de estados estimados.

El presente artículo está estructurado como se describe a continuación. En la Sección 2 se desarrolla el modelo dinámico del modelo LIP en función del momento angular. En la Sección 3 se presenta la función objetivo a minimizar y el algoritmo para la generación de trayectorias en línea.

En la Sección 4 se muestran los resultados de simulación. La Sección 5 describe el estimador de estados usado en este trabajo y en la Sección 6 se muestran los resultados experimentales.

## II. MODELADO

Como modelo dinámico se eligió el conocido modelo LIP en función del momento angular, ya que de acuerdo con [14], el momento angular es una mejor predicción que la velocidad del CoM. El modelo LIP consiste en asumir que la masa total del robot  $m$  se encuentra concentrada en un sólo punto y que la altura del CoM está restringida a moverse sobre una línea recta <sup>1</sup> [8]. La ecuación del modelo está dada por :

$$\ddot{x} = \frac{g}{z}(x - p_x), \quad (1)$$

donde  $\ddot{x}$  y  $x$  son la aceleración y posición del CoM,  $g$  es la constante de gravedad,  $z$  la altura del CoM y  $p_x$  la coordenada en  $x$  del ZMP. El par en el punto de soporte está dado por el

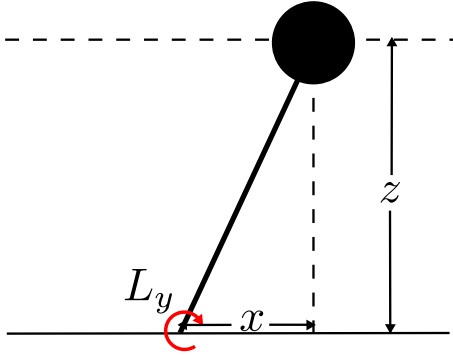


Fig. 1. Modelo Lineal de Péndulo Invertido (LIP)

producto cruz de la posición  $\mathbf{r}$  del CoM y la fuerza  $\mathbf{f}$  presente en el CoM:

$$\boldsymbol{\tau} = \mathbf{r} \times \mathbf{f} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \times \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \end{bmatrix} \quad (2)$$

Analizando la componente del par en el eje de giro:

$$\tau_y = -xm\ddot{z} + zm\ddot{x} = zm\ddot{x}, \quad (3)$$

debido a que la altura es constante,  $\dot{z} = 0$  y  $\ddot{z} = 0$ . Sustituyendo la dinámica del LIP (1) en la ecuación anterior:

$$\tau_y = zm \left( \frac{g}{z}(x - p_x) \right) \quad (4)$$

Por definición, el par  $\tau_y$  es la derivada con respecto al tiempo del momento angular  $L_y$

$$\dot{L}_y = \tau_y = mg(x - p_x) \quad (5)$$

<sup>1</sup>En el caso de este trabajo la altura del CoM siempre se considerará constante.

Por definición el momento angular  $\mathbf{L}$  está dado por el producto cruz del vector de posición  $\mathbf{r}$  del CoM y el vector del momento lineal  $m\mathbf{v}$ :

$$\mathbf{L} = \mathbf{r} \times m\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \times \begin{bmatrix} m\dot{x} \\ m\dot{y} \\ m\dot{z} \end{bmatrix} \quad (6)$$

Entonces

$$L_y = -xm\dot{z} + zm\dot{x} = zm\dot{x} \quad (7)$$

Con lo que obtenemos las ecuaciones del modelo LIP, donde los estados son la posición del centro de masa  $x$  y el momento angular sobre el pie de soporte  $L_y$ :

$$\dot{x} = \frac{L_y}{zm} \quad (8)$$

$$\dot{L}_y = mg(x - p_x) \quad (9)$$

El mismo procedimiento puede usarse para encontrar la dinámica del CoM en el eje  $y$ . Es importante señalar que la dinámica de (8) y (9) es idéntica a la dinámica de (1), la diferencia son las variables que se usan como estados. En forma de espacio de estados:

$$\begin{bmatrix} \dot{x} \\ \dot{L}_y \end{bmatrix} = \begin{bmatrix} 0 & 1/zm \\ mg & 0 \end{bmatrix} \begin{bmatrix} x \\ L_y \end{bmatrix} + \begin{bmatrix} 0 \\ -mg \end{bmatrix} p_x \quad (10)$$

## III. GENERACIÓN DE TRAYECTORIAS EN LÍNEA

Discretizando, se tiene el vector de estados  $\mathbf{x}_k = [x_k \quad L_k]^T$  y la entrada de control  $u_k = p_k$

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}u_k \quad (11)$$

y el vector de salida

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k \quad (12)$$

En este caso la salida es el momento angular, por lo que  $\mathbf{C} = [0 \quad 1]$ . Iterando las ecuaciones (11) y (12) se puede obtener una predicción para el momento angular pasadas  $N$  muestras:

$$\underbrace{\begin{bmatrix} L_{k+1} \\ L_{k+2} \\ \vdots \\ L_{k+N} \end{bmatrix}}_{\hat{\mathbf{L}}_{k+1}} = \underbrace{\begin{bmatrix} \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \vdots \\ \mathbf{C}\mathbf{A}^N \end{bmatrix}}_{\mathbf{P}_x} \mathbf{x}_k + \mathbf{P}_u \underbrace{\begin{bmatrix} p_k \\ p_{k+1} \\ \vdots \\ p_{k+N-1} \end{bmatrix}}_{\mathbf{p}_k}, \quad (13)$$

con

$$\mathbf{P}_u = \begin{bmatrix} \mathbf{C}\mathbf{B} & 0 & \dots & 0 \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{A}^{N-1}\mathbf{B} & \mathbf{C}\mathbf{A}^{N-2}\mathbf{B} & \dots & \mathbf{C}\mathbf{B} \end{bmatrix}, \quad (14)$$

donde  $N$  es el número de muestras sobre las cuales se realiza la predicción del momento angular, en este caso el número de muestras durante un paso del robot, el cual está relacionado con el tiempo de muestreo y la duración del paso del robot. En forma compacta

$$\hat{\mathbf{L}}_{k+1} = \mathbf{P}_x \mathbf{x}_k + \mathbf{P}_u \mathbf{p}_k. \quad (15)$$

A diferencia de otros trabajos [3]–[5], no se busca optimizar sobre una trayectoria con un horizonte fijo. Se busca optimizar para un valor deseado (momento angular) al final del paso.

En cada instante de muestreo se resuelve el problema de optimización:

$$\min_{\mathbf{p}_k(i:N)} \mathbf{J} \quad (16)$$

donde

$$\mathbf{J} = \frac{\alpha}{2}(L_{k+N-i} - L_{des})^2 + \frac{\beta}{2}\mathbf{p}_k(i:N)^2, \quad (17)$$

Donde  $i$  es el instante de muestreo actual,  $L_{k+N-i}$  corresponde a la predicción del momento angular al final del paso realizada en el instante  $i$ , lo que es un renglón de (13),  $\alpha$  y  $\beta$  son pesos para la función de optimización y  $L_{des}$  es el momento angular deseado al final del paso. Es decir cada instante de muestreo se calcula cuál es la trayectoria del ZMP que llevará al momento angular deseado al final del paso. Durante el paso, esta trayectoria será cada vez mas corta debido a que cada vez se está más cerca del final del paso, es decir, el horizonte de predicción será cada vez más pequeño.

Si no existen restricciones, la solución del problema de optimización (16) puede resolverse de forma analítica, derivando (16), igualando a cero y despejando para  $\mathbf{p}_k$ . Si se quieren incluir restricciones, por ejemplo limitar la posición del ZMP, es necesario el uso de un solucionador numérico de problemas de programación cuadrática (QP).

Una vez encontrada la trayectoria del ZMP, solo se aplica la primera muestra de la trayectoria encontrada al sistema, como es normal en el control por modelo predictivo [3].

A continuación se describe el algoritmo del generador de trayectorias

---

**Algoritmo 1** Algoritmo de generación de trayectorias en línea

---

```

i = 0
while i ≠ N do
    p_k(i : N) = arg min_{p_k} J
    u_k = p_k(0)
    x_{k+1} = Ax_k + Bu_k
    i ← i + 1
end while

```

---

**A. Eje y**

Debido a que en el modelo LIP la dinámica del eje  $y$  está dada por la misma ecuación (1). Para lograr trayectorias en el eje  $y$  es el mismo algoritmo. La diferencia es que según sea el pie de soporte el momento angular deseado al final del paso será positivo o negativo. En nuestra convención de marcos, cuando el pie de soporte es el derecho se requiere un momento angular positivo. Cuando el pie de soporte es el izquierdo se requiere un momento angular negativo.

$${}^yL_{des} = \mu L_{des}, \quad (18)$$

donde  $\mu = -1$  si el pie de soporte es el izquierdo, mientras que  $\mu = 1$  cuando el pie de soporte es el derecho.

**IV. RESULTADOS DE SIMULACIÓN**

Para comprobar el funcionamiento del generador de trayectorias presentado en la sección anterior se realiza la siguiente simulación para 6 pasos del robot, el cual inicia en el reposo  $[x, y, {}^xL, {}^yL] = [0, D/2, 0, 0]$  y tiene como momento angular deseado al final del paso  ${}^xL_{des} = 0.43$  y  ${}^yL_{des} = 0.377$ . Un largo de paso  $S = 0.1$  m, un ancho de paso  $D = 0.1$  m y un tiempo de muestreo  $\Delta t = 0.005$  s. El modelo discreto (11) fue encontrando usando el comando *c2d()* de Matlab y se usó la solución analítica al problema de optimización (16).

En la Fig. 2 se muestra el momento angular durante el caminado en los ejes  $x$  e  $y$  respecto al tiempo junto con la referencia del momento angular deseado. Como puede verse la dinámica del sistema tiende a un ciclo límite, inclusive cuando no se inicia en el punto fijo del ciclo límite, lo que es característico de un ciclo límite estable. Puede verse que no se alcanza el momento angular deseado al final del paso pero este no es el objetivo del trabajo, el cual es obtener ciclos límite estables para el movimiento del CoM.

En la Fig. 3 se muestra la trayectoria del CoM en el plano  $x-y$  junto con la trayectoria del ZMP obtenida en el problema de optimización, la cual siempre se encuentra dentro de la zona de soporte.

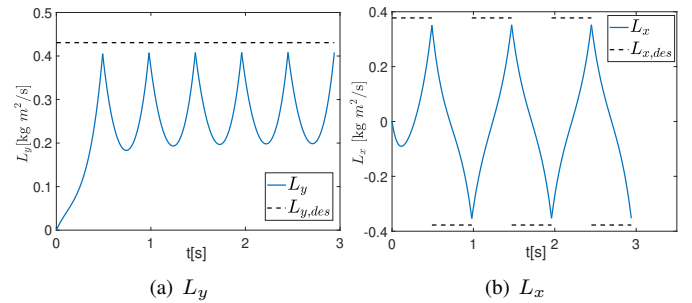


Fig. 2. Momentos angulares durante el caminado y momento angular deseado al final del paso obtenidos en simulaciones

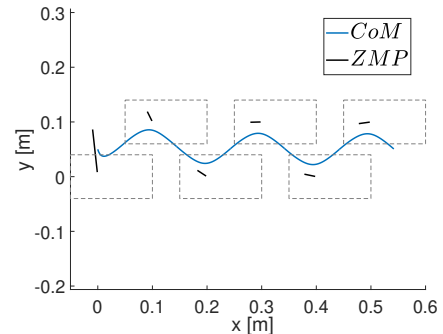


Fig. 3. Trayectorias del CoM y del ZMP durante el caminado obtenidas en simulación

**A. Simulación considerando la dinámica completa**

Para validar el funcionamiento del generador de trayectorias basado en el modelo LIP, las trayectorias generadas del ZMP fueron implementadas en el modelo esencial [9] de un robot

bípido planar de 6 g.d.l, el cual es un modelo dinámico que considera la dinámica completa de un robot.

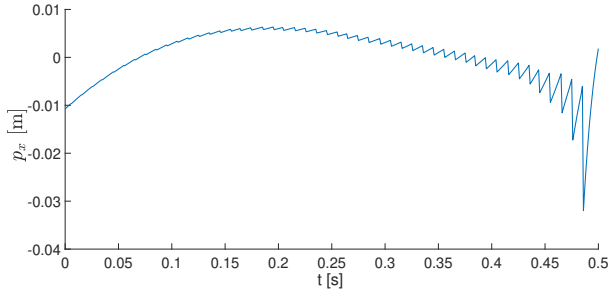


Fig. 4. ZMP de un paso obtenido aplicando el generador en un modelo con la dinámica completa.

Al realizar simulaciones del generador propuesto en el modelo que considera la dinámica completa se obtuvo como resultado la trayectoria del ZMP de la Fig. 4, en la que puede verse una trayectoria con cambios repentinos en el ZMP debido a la diferencia de la dinámica del LIP y la dinámica completa. Para evitar estos cambios del ZMP se agregó un término adicional a la función objetivo (16):

$$\min_{\mathbf{p}_k(i:N)} \mathbf{J} + \mathbf{J}_b \quad (19)$$

donde

$$\mathbf{J}_b = \frac{\gamma}{2} (\mathbf{p}_k(i:N) - \mathbf{p}_{k-1})^2, \quad (20)$$

donde  $\mathbf{p}_{k-1}$  es el ZMP obtenido en el problema de optimización del instante anterior. El objetivo de este nuevo término en la función objetivo es disminuir la diferencia entre el ZMP obtenido en el instante actual y el ZMP obtenido en el instante anterior.

Los resultados de simulación de esta nueva función objetivo pueden verse en la Fig. 5, los cuales muestran un ciclo límite estable para la posición y velocidad del CoM en el robot. En la Fig. 6 se muestra la nueva trayectoria obtenida para el ZMP con la función objetivo (19) con un comportamiento mucho más suave que en la Fig. (4). Destacando que el generador propuesto crea ciclos límite estables en un modelo que considera la dinámica completa del robot (la cual es diferente de la dinámica del LIP).

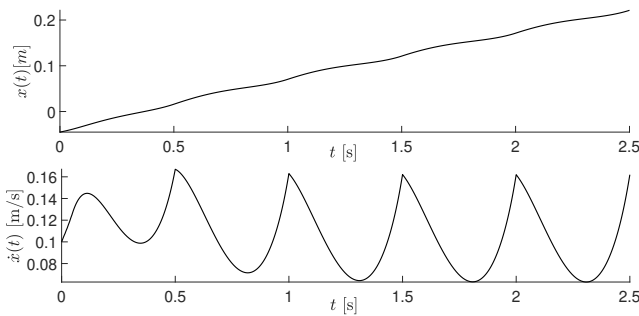


Fig. 5. Posición y velocidad del CoM durante el caminado de un robot planar usando un modelo que considera la dinámica completa del robot.

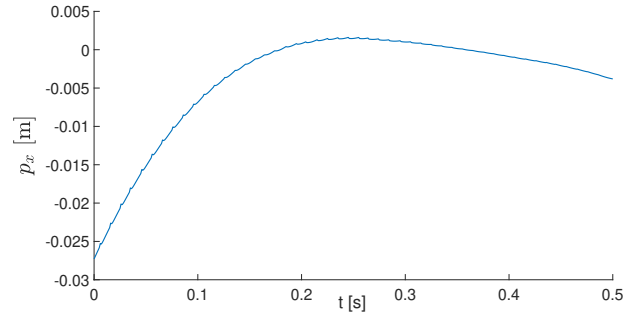


Fig. 6. ZMP obtenido de un paso aplicando la nueva función objetivo (20).

## V. ESTIMADOR DE ESTADOS

Debido a que este controlador recae en la predicción del momento angular al final del paso, la cual depende de la posición y velocidad del CoM actual. Es necesario una estimación de estas dos variables en tiempo real. Para lograr una buena estimación de estos dos estados se usará un filtro de Kalman. El filtro presentado fusiona información de las lecturas del acelerómetro incluidas en el modelo de proceso con información obtenida de los encoders de posición de los motores.

El modelo para la estimación de estados consiste en considerar al CoM como una masa puntual que es afectada por la aceleración obtenida por la IMU sujeto a incertidumbres modeladas como ruido blanco gaussiano. Además de considerar que los pies permanecen estáticos mientras que se encuentran en contacto con el suelo. Se considera que la posición de los pies puede ser modificada debido a deslizamientos con el suelo, los cuales son modelados como ruido blanco gaussiano. A diferencia de [12] se incluye un término de bias  $\mathbf{b}_a$  para modelar el offset presente en las lecturas del acelerómetro:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= {}^0\mathbf{R}_b(\mathbf{a} - \mathbf{b}_a - \boldsymbol{\omega}_a) + \mathbf{g} \\ \dot{\mathbf{p}}_i &= \boldsymbol{\omega}_{p,i} \\ \dot{\mathbf{b}}_a &= \boldsymbol{\omega}_{ba}, \end{aligned}$$

donde  $\mathbf{p}$  y  $\mathbf{v}$  son la posición del CoM,  $\mathbf{a}$  es la aceleración obtenida por la IMU,  $\boldsymbol{\omega}_a$ ,  $\boldsymbol{\omega}_{p,i}$  y  $\boldsymbol{\omega}_{ba}$  son ruidos blancos gaussianos,  $\mathbf{g}$  es el vector de gravedad,  $\mathbf{p}_i$  es la posición de cada uno de los pies,  ${}^0\mathbf{R}_b$  es la matriz de rotación del cuerpo del robot respecto al marco inercial, el cual se encuentra fijo al suelo e inicialmente coincide con el marco del pie de soporte. El robot NAO cuenta con su propio proceso de filtrado del giroscopio y acelerómetro para encontrar los ángulos Roll-Pitch-Yaw del cuerpo respecto al marco inercial. Estos ángulos son usados para encontrar la matriz de rotación  ${}^0\mathbf{R}_b$ .

Las salida del modelo es la posición del pie derecho e izquierdo respecto al centro de masa. Además de la altura de ambos pies respecto al marco inercial:

$$\mathbf{y}_F = [{}^b\mathbf{p}_R \quad {}^b\mathbf{p}_L \quad {}^0p_{R,z} \quad {}^0p_{L,z}]^T \quad (21)$$

El vector de estados estimados es la posición y velocidad del CoM respecto al marco inercial, la posición de los pies

respecto al marco inercial y el bias en el acelerómetro respecto al marco inercial:

$$\mathbf{x}_F = [{}^0\mathbf{p}_b \quad {}^0\mathbf{v}_b \quad {}^0\mathbf{p}_R \quad {}^0\mathbf{p}_L \quad {}^0\mathbf{b}_a]^T, \quad (22)$$

Para obtener el estado estimado óptimo, el filtro de Kalman requiere de dos modelos: el modelo de proceso, que describe como los estados evolucionan en el tiempo y el modelo de medición que observa las señales que pueden ser medidas en el sistema. Ambos se describen a continuación.

#### A. Modelo de proceso

En el modelo de proceso se modela al CoM como una masa puntual, cuya aceleración es obtenida por la IMU. En tiempo discreto con un tiempo de muestreo  $\Delta t$  esta dado por:

$$\bar{\mathbf{x}}_{F,k+1} = \begin{bmatrix} {}^0\mathbf{p}_{b_k} + \Delta t {}^0\mathbf{v}_{b_k} + \frac{\Delta t^2}{2} ({}^0\mathbf{R}_b \mathbf{a} + \mathbf{g} + {}^0\mathbf{b}_{a,k}) \\ {}^0\mathbf{v}_{b_k} + \Delta t ({}^0\mathbf{R}_b \mathbf{a} + \mathbf{g} + {}^b\mathbf{b}_{a,k}) \\ {}^0\mathbf{p}_{R_k} \\ {}^0\mathbf{p}_{L_k} \\ {}^0\mathbf{b}_{a,k} \end{bmatrix}, \quad (23)$$

donde  ${}^0\mathbf{R}_b \mathbf{a} + \mathbf{g}$  es la entrada de control  $\mathbf{u}_F$ . En espacio de estados

$$\bar{\mathbf{x}}_{F,k+1} = \mathbf{F} \hat{\mathbf{x}}_k + \mathbf{G} \mathbf{u}_F, \quad (24)$$

donde

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \Delta t \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \frac{\Delta t^2}{2} \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \Delta t \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}, \quad (25)$$

$$\mathbf{G} = \begin{bmatrix} \frac{\Delta t^2}{2} \mathbf{I}_{3 \times 3} & \Delta t \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}^T. \quad (26)$$

La salida del modelo de proceso es:

$$\bar{\mathbf{y}}_{F,k} = \begin{bmatrix} -{}^0\mathbf{p}_{b_k} + {}^0\mathbf{p}_{R_k} \\ -{}^0\mathbf{p}_{b_k} + {}^0\mathbf{p}_{L_k} \\ {}^0\mathbf{p}_{Rk}(3) \\ {}^0\mathbf{p}_{Lk}(3) \end{bmatrix} = \mathbf{H} \bar{\mathbf{x}}_{F,k}, \quad (27)$$

$$\mathbf{H} = \begin{bmatrix} -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & [0, 0, 1] & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & [0, 0, 1] & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (28)$$

#### B. Modelo de medición

Como modelo de medición se usa el modelo cinemático directo de posición del robot y los encoders para calcular la posición de las piernas del robot respecto al centro de masa y la altura de los pies:

$$\mathbf{y}_{F,k} = \mathbf{dkin}(\mathbf{q}_k) + \mathbf{n}, \quad (29)$$

donde  $\mathbf{q}_k$  son las posiciones articulares y  $\mathbf{n}$  es ruido blanco gaussiano que modela incertidumbres en el modelo cinemático y ruido en las lecturas de los encoders. La función  $\mathbf{dkin}()$  es definida como el modelo cinemático directo de posición de la posición de los pies respecto al CoM, el cual puede ser encontrado por matrices de transformación y parámetros Denavit-Hartenberg.

#### C. Consideración del contacto de los pies con el suelo

Uno de los aspectos mas importantes al implementar un filtro de Kalman en un robot con piernas es el cómo lidiar con los contactos intermitentes con el suelo. Para esto la metodología más usada [10]–[13] es estimar cuál es el pie que se encuentra en contacto con el suelo a partir de los sensores de fuerza de los pies. Si se estima que un pie no se encuentra en contacto con el suelo, su matriz de covarianza en el modelo de proceso y medición  $\mathbf{Q}$  y  $\mathbf{R}$  se aumenta a un valor muy alto para no considerar las mediciones del pie que no se encuentra en contacto con el suelo.

#### D. Ecuaciones del filtro Kalman

Finalmente para encontrar la estimación del estado  $\hat{\mathbf{x}}$  las ecuaciones del filtro de Kalman [15] son aplicadas, las cuales están divididas en dos partes, predicción:

$$\begin{aligned} \bar{\mathbf{x}}_{k+1} &= \mathbf{F} \hat{\mathbf{x}}_k + \mathbf{G} \mathbf{u}_{F,k} \\ \bar{\mathbf{P}}_{k+1} &= \mathbf{F} \mathbf{P}_k \mathbf{F}^T + \mathbf{Q}, \end{aligned}$$

y corrección:

$$\begin{aligned} \mathbf{K} &= \bar{\mathbf{P}}_{k+1} \mathbf{H}^T (\mathbf{H} \bar{\mathbf{P}}_{k+1} \mathbf{H}^T + \mathbf{R}) \\ \hat{\mathbf{x}}_{k+1} &= \bar{\mathbf{x}}_{k+1} + \mathbf{K} (\mathbf{y}_{F,k} - \bar{\mathbf{y}}_{F,k}) \\ \mathbf{P}_{k+1} &= (\mathbf{I} - \mathbf{K} \mathbf{H}) \bar{\mathbf{P}}_{k+1}, \end{aligned}$$

donde  $\mathbf{P}$  es la matriz de covarianza del estado estimado  $\hat{\mathbf{x}}$ ,  $\mathbf{Q}$  es la matriz de covarianza del proceso,  $\mathbf{R}$  es la matriz de covarianza de las mediciones,  $\mathbf{K}$  es la ganancia de Kalman.

## VI. RESULTADOS EXPERIMENTALES

Al realizar pruebas experimentales con el estimador de estados se encontró que las estimaciones presentan un retraso de alrededor de 40 ms respecto a los valores esperados. Este retraso provoca que al realimentar las señales del estimador en el generador, el caminado resultante no funciona correctamente, inclusive algunas veces volviendo al sistema inestable. Siendo el mayor problema la realimentación en el eje  $x$ , la cual contenía ruido en la velocidad, obstaculizando el desempeño del generador.

Debido a estos inconvenientes en las pruebas experimentales, se optó por realizar un caminado en el lugar (*walking in place*), el cual consiste en mantenerse en equilibrio mientras se realiza un intercambio del pie de soporte, permaneciendo en el mismo lugar. Además, en los experimentos se realimenta sólo un porcentaje de la posición y velocidad del CoM en el eje  $y$  estimada por el filtro de Kalman. El resto de la realimentación se obtiene a partir del modelo LIP (10).

El experimento consiste en un caminado sin desplazamiento en el eje  $x$  con un tiempo de paso de  $T_s = 0.8$  s, un tiempo de muestreo  $\Delta t = 0.04$  s y un momento angular deseado al final del paso de  ${}^y L_{des} = 0.4$ . En este experimento se realimentó un 25% de la posición del CoM obtenida por el estimador de estados y un 15% de la velocidad. La posición y velocidad del CoM en el eje  $x$  no fue realimentada por el estimador de estados. Otros detalles de implementación como añadir

una fase de doble soporte y un offset para las articulaciones *HipRoll* del robot fueron aplicados [16].

En la Fig. 7 se muestra una serie de imágenes del robot durante el experimento. Un video del experimento puede encontrarse en <https://www.youtube.com/watch?v=V2thI291jKA>.

En la Fig. 8 se muestra la trayectoria del CoM obtenida por el generador durante el experimento, así como el ZMP que genera esta trayectoria al resolver el problema de optimización (17). Como puede verse la trayectoria para el CoM obtenida por el generador se conserva periódica luego de dar varios pasos y al estar sensando información real del robot, lo que valida a nuestro generador como un generador de trayectorias estables para el CoM.

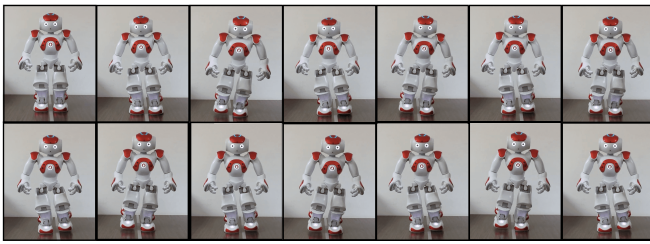


Fig. 7. Resultados experimentales del caminado

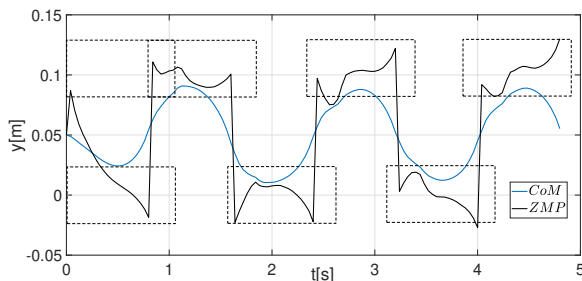


Fig. 8. Trayectoria del ZMP y del CoM del eje  $y$  obtenida en experimentos

El ZMP obtenido por el generador no está limitado a permanecer dentro de una región determinada, como se muestra en la Fig. 8, lo que puede ocasionar una pérdida de equilibrio al no mantenerse siempre dentro del área del pie [1]. Esto puede solucionarse definiendo restricciones en el problema de optimización (17) y usando un solucionador numérico de problemas de programación cuadrática, lo que incrementaría el costo computacional, el cual es limitado en el robot NAO. Es por esto que se decidió no incluir restricciones en los experimentos. Además los resultados mostraron que aunque el ZMP del modelo LIP salga durante pequeños momentos de la zona del pie, el robot real se encuentra en equilibrio.

## VII. CONCLUSIONES

En este artículo se presentó un nuevo generador de trayectorias en línea que genera caminados estables en el modelo LIP a partir de la predicción del momento angular al final del paso y un problema de optimización. Los resultados fueron validados en simulación en un modelo que considera

la dinámica completa del robot y en experimentos en el robot NAO dando como resultados caminados estables.

Otro resultado de este trabajo es la implementación de un filtro de Kalman para la estimación de la posición y velocidad del CoM del robot. El filtro presentado es un filtro de Kalman lineal, lo que facilita su implementación en línea. Además de que modela bias presentes en el acelerómetro. Los resultados experimentales del estimador no fueron los esperados debido al hardware limitado con el que cuenta el robot NAO pero aún así se lograron llevar a cabo experimentos usando el estimador de estados junto con el generador de trayectorias.

Como trabajo a futuro se tiene: mejorar el funcionamiento del estimador de estados para poder realizar un caminado hacia adelante o en una dirección arbitraria, considerar la posibilidad de incluir restricciones en el ZMP y comparar este generador con otros generadores del estado del arte.

## REFERENCIAS

- [1] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life." *I. J. Humanoid Robotics*, vol. 1, pp. 157–173, 03 2004.
- [2] E. Marquez-Acosta, V. De-Leon-Gomez, and V. Santibañez, "A walking gait for robot nao by using the essential model," in *Latin American Congress on Automation and Robotics*, 2021, pp. 62–67.
- [3] P. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 137–142.
- [4] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, 2003, pp. 1620–1626 vol.2.
- [5] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footprint placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010. [Online]. Available: <https://doi.org/10.1163/016918610X493552>
- [6] M. Shafice-Ashtiani, A. Yousefi-Koma, M. Shariat-Panahi, and M. Khadiv, "Push recovery of a humanoid robot based on model predictive control and capture point," in *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, 2016, pp. 433–438.
- [7] M. Krause, J. Engelsberger, P.-B. Wieber, and C. Ott, "Stabilization of the capture point dynamics for bipedal walking based on model predictive control," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 165–171, 2012, 10th IFAC Symposium on Robot Control. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016336059>
- [8] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, 2001, pp. 239–246 vol.1.
- [9] V. De-León-Gómez, Q. Luo, A. Kalouguine, J. A. Pámanes, Y. Aoustin, and C. Chevallereau, "An essential model for generating walking motions for humanoid robots," *Robotics and Autonomous Systems*, vol. 112, pp. 229–243, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889018303373>
- [10] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State Estimation for Legged Robots: Consistent Fusion of Leg Kinematics and IMU," in *Robotics: Science and Systems VIII*. The MIT Press, 07 2013. [Online]. Available: <https://doi.org/10.7551/mitpress/9816.003.0008>
- [11] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, "State estimation for a humanoid robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sep. 2014. [Online]. Available: <http://dx.doi.org/10.1109/IROS.2014.6942674>
- [12] T. Flayols, A. Del Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse, "Experimental evaluation of simple estimators for humanoid robots," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 889–895.

- [13] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2245–2252.
- [14] Y. Gong and J. W. Grizzle, "Angular momentum about the contact point for control of bipedal locomotion: Validation in a lip-based controller," *CoRR*, vol. abs/2008.10763, 2020. [Online]. Available: <https://arxiv.org/abs/2008.10763>
- [15] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.
- [16] E. Marquez-Acosta, V. De-Leon-Gomez, and V. Santibañez, "Real-time implementation of an online controller for the walking of the nao robot," in *Memorias del Congreso Nacional de Control Automático*, 2023, pp. 580–585.

# CAPÍTULO 6

---

## MODELADO CINEMÁTICO

# Pruebas experimentales de tiempo real en el control cinemático de movimiento de un robot cuadrúpedo

M. C. Martínez-Rodríguez  
*Tecnológico Nacional de México*  
*IT de Ensenada*  
Ensenada, B.C.  
alm17760542@ite.edu.mx

A. Y. Aguilar-Bustos  
*Tecnológico Nacional de México*  
*IT de Ensenada*  
Ensenada, B.C.  
aaguilar@ite.edu.mx

E. Bugarin  
*Tecnológico Nacional de México*  
*IT de Ensenada*  
Ensenada, B.C.  
ebugarin@ite.edu.mx

J. L. Muñoz-Torres  
*Tecnológico Nacional de México*  
*IT de Ensenada*  
Ensenada, B.C.  
alm17760517@ite.edu.mx

**Resumen**— En este documento se presentan pruebas experimentales de tiempo real relacionadas con la estrategia de control cinemático para el movimiento de un robot cuadrúpedo. Para ello, se describe el modelo cinemático del mismo y la implementación del algoritmo de control. El robot cuadrúpedo empleado es de bajo costo y cuenta con 12 servomotores Dynamixel AX-12A modificados para trabajar en tiempo real y a relativamente alta velocidad de procesamiento. Para la experimentación fue utilizado un sistema de cómputo con el sistema operativo de tiempo real Linux con el parche PREEMPT\_RT. Finalmente, se detallan los experimentos realizados haciendo énfasis en la periodicidad del lazo de control; obteniéndose un promedio de 0.0065 [s] para el periodo de muestreo y de 0.006972 [s] para el peor caso de tiempo de ejecución (WCET por sus siglas en inglés).

**Palabras clave**— Robot cuadrúpedo, modelo cinemático, control de movimiento, tiempo real, validación experimental.

## I. INTRODUCCIÓN

La locomoción con patas permite a humanos y animales atravesar entornos difíciles con agilidad y fluidez. Debido a esta habilidad, los mismos han servido como fuente de inspiración en el campo de la robótica. Esta clase de robots móviles son llamados robots con patas y presentan diversas ventajas sobre otros robots móviles; como por ejemplo, pueden ayudar a realizar tareas complejas o interactuar en áreas inaccesibles. Ahora bien, para controlar los movimientos de estos robots existen diferentes metodologías, una de ellas se apoya en el espacio operacional donde la tarea deseada es establecida en coordenadas operacionales y después mapeada al espacio articular para conseguir el movimiento de cada eslabón del robot. El mapeo es realizado de distintas formas, por ejemplo, analíticamente donde por el método geométrico son encontrados los valores articulares en función de una posición deseada. Otras metodologías empleadas son propuestas mediante la utilización de los métodos numéricos; por ejemplo, Orbegoso *et al.* en [1] resolvieron la cinemática inversa (CI) de un robot cuadrúpedo utilizando Newton-Raphson.

Por otro lado, Pechev en [2] describe la transformación del espacio operacional al articular en un sistema en lazo cerrado proponiendo una solución para dicho sistema sin la inversión de matrices; evitando así problemas de singularidad. Aristidou y Lasenby en [3] solucionan la CI evitando el uso de ángulos junto

con matrices de rotación; en cambio encuentran la posición de las variables articulares al ubicar puntos sobre rectas. Sen *et al.*, en [4], por otra parte, formulan la CI de un robot cuadrúpedo a partir de su cinemática directa (CD) de forma analítica.

El objetivo principal de este artículo no es la propuesta de un control de movimiento para un robot cuadrúpedo; sino la constatación de la posibilidad de realizar experimentos de tiempo real y relativa alta velocidad de procesamiento en diversas propuestas de control de movimiento del robot cuadrúpedo diseñado en nuestro laboratorio ex profeso para estos propósitos [5]. De esta manera, la metodología utilizada en el presente artículo para el control de movimiento del robot cuadrúpedo se basa en lo descrito por Siciliano *et al.* en [6] donde mediante la cinemática de velocidad se establece un algoritmo para mapear los movimientos articulares del robot. Por cierto, el robot cuadrúpedo diseñado tiene una morfología mamífera, conformándose de 3 articulaciones por cada pata.

Un sistema robótico suele interactuar constantemente con su entorno, por lo que es de interés garantizar los tiempos de respuesta del mismo. Los sistemas de tiempo real (STR) aseguran una interacción continua y oportuna con su entorno [7]; ya que no sólo dependen de los resultados lógicos de los cálculos, sino también del tiempo cuando se producen estos resultados. En caso de que un STR incluya una computadora es necesario emplear un sistema operativo de tiempo real (SOTR). Los sistemas embebidos de tiempo real donde se utilizan los SOTRs pueden ser “duros” o “suaves”; es decir, estrictos o con posibilidad de seguir operando a pesar de ocasionalmente exceder el límite de tiempo requerido, respectivamente [8].

Los robots cuadrúpedos más avanzados funcionan con SOTRs. El Mini Cheetah [9], por ejemplo, opera en Linux con el parche PREEMPT\_RT para tiempo real, alcanzando tiempos de ciclo de 0.001 [s]. Por otro lado, el robot ANYmal [10] es controlado por una red de computadoras que operan de manera coordinada utilizando un parche de baja latencia de Ubuntu, logrando tiempos de ciclo de 0.0025 [s].

Diversos estudios han comparado el desempeño de diferentes SOTRs evaluando la periodicidad mediante una serie de tareas. Por ejemplo, Boltov *et al.* [11] realizaron pruebas experimentales en una SBC (Single Board Computer) para el

procesamiento de imágenes, comparando los SOTRs PREEMPT\_RT y Xenomai; mientras que Muñoz-Torres *et al.* [12] evaluaron el rendimiento de estos mismos SOTRs en un sistema de visión instalado en una computadora NUC (Next Unit of Computing). Ambos estudios encontraron diferencias en la latencia mínima entre estos SOTRs, con una ligera ventaja para PREEMPT\_RT. Lo anterior justifica, para este trabajo, el uso del SOTR Linux con el parche PREEMPT\_RT.

Finalmente, para cumplir con el objetivo principal de este artículo, primero se explica la cinemática de velocidad del robot cuadrúpedo (objeto de estudio); luego se describe el control cinemático para lograr su movimiento; enseguida se detalla la plataforma experimental que constituye al robot cuadrúpedo; a continuación se le imponen 3 escenarios de movimiento para la valoración del mismo; y por último se establecen las conclusiones de este trabajo.

## II. ESTRATEGIA DE MOVIMIENTO

### A. Cinemática de posición

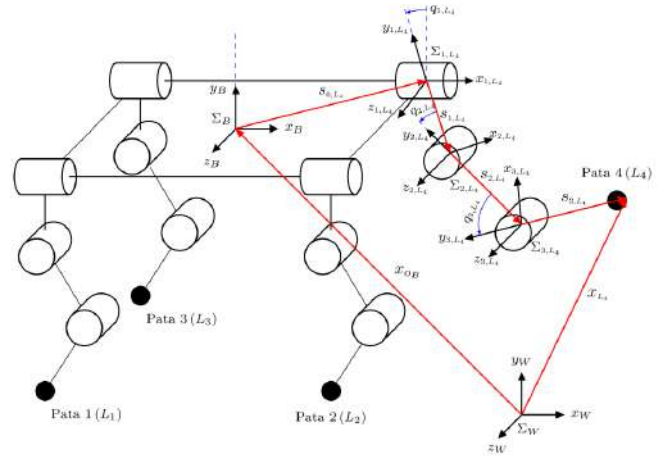
Para el cálculo de la cinemática directa de posición del robot cuadrúpedo bajo estudio se siguió la convención utilizada en [5]. Así pues (ver Fig. 1), fue asignado un marco de coordenadas en cada eslabón de cada pata ( $\Sigma_{j,L_i}$  con  $i = 1, 2, 3, 4$  correspondiendo al número de pata y  $j = 1, 2, 3$  al número de eslabón de la pata en cuestión). Se asignó el marco de la base ( $\Sigma_B$ ) en el centro geométrico de la estructura que sostiene las 4 patas y el marco inercial ( $\Sigma_W$ ) en algún lugar en el espacio de trabajo. De igual forma se establecieron las variables articulares  $q_{j,L_i}$ . Note que  $s_{\ell,L_i}$  (con  $\ell = 0, 1, 2, 3$ ) son los vectores de posición (constantes) entre orígenes y  $x_{L_i}$  es el vector de posición del extremo de la pata  $i$ ; esto es (en lo que sigue el superíndice derecho indica el marco de referencia)

$$\begin{aligned} s_{0,L_i}^W &= R_B^W s_{0,L_i}^B \\ s_{1,L_i}^W &= R_B^W R_{1,L_1}^B s_{1,L_i}^{1,L_1} \\ s_{2,L_i}^W &= R_B^W R_{1,L_1}^B R_{2,L_1}^{1,L_1} s_{2,L_i}^{2,L_1} \\ s_{3,L_i}^W &= R_B^W R_{1,L_1}^B R_{2,L_1}^{1,L_1} R_{3,L_1}^{2,L_1} s_{3,L_i}^{3,L_1} \\ x_{L_i}^W &= x_{oB}^W + s_{0,L_i}^W + s_{1,L_i}^W + s_{2,L_i}^W + s_{3,L_i}^W \end{aligned} \quad (1)$$

donde  $R_B^W$  es la matriz de rotación de la base referenciada a  $\Sigma_W$ ,  $x_{oB}^W$  es el vector de posición del origen de  $\Sigma_W$  al origen de  $\Sigma_B$  y  $R_{j,L_i}^{j-1,L_i}$  las matrices de rotación entre marcos consecutivos.

### B. Cálculo de la posición de la base

Como una aproximación muy preliminar supóngase que el robot desarrolla movimientos de rotación de la base pequeños (tal que en todo momento  $R_B^W = I$ ) y que cuando todas las variables articulares  $q_{j,L_i}$  son iguales a cero  $\Sigma_W$  coincide exactamente con  $\Sigma_B$  (tal que  $x_{L_i}^W$  es conocida), entonces la evolución de  $x_{oB}^W$  puede ser calculada resolviendo (1) desde cualquier pata  $i$ ; esto es



Nota: Las patas 1 y 3 son las patas traseras y las patas 2 y 4 son las delanteras. Los marcos de coordenadas para las patas 1, 2 y 3 se colocan de la misma forma que en la pata 4. Además, observe que  $q_{j,L_i}$  se mide de  $y_{j-1,L_i}$  a  $y_{j,L_i}$ .

Fig. 1. Definición de marcos de coordenadas en el robot cuadrúpedo.

$$x_{oB}^W = x_{L_i}^W - (s_{0,L_i}^W + s_{1,L_i}^W + s_{2,L_i}^W + s_{3,L_i}^W).$$

Sin embargo, como sabemos que es una aproximación muy preliminar (lo cual se considera práctico debido a que de esta manera no es necesario agregar otros sensores, como unidades inerciales); en realidad se considerará el promedio de lo calculado por cada pata, esto es

$$x_{oB}^W = \frac{1}{4} (x_{oB1}^W + x_{oB2}^W + x_{oB3}^W + x_{oB4}^W). \quad (2)$$

### C. Cinemática de velocidad

Realizando un análisis de propagación de velocidades lineales y angulares desde la base del robot cuadrúpedo hasta su pata  $i$  se puede obtener su cinemática de velocidad siguiente:

$$\xi_{L_i}^W = \begin{bmatrix} v_{L_i}^W \\ \omega_{L_i}^W \end{bmatrix} = \begin{bmatrix} a & b & c & I & -D \\ h_{1,L_i}^W & h_{2,L_i}^W & h_{3,L_i}^W & 0 & I \end{bmatrix} \begin{bmatrix} \dot{q}_{L_i} \\ \xi_B^W \end{bmatrix} \quad (3)$$

donde  $\xi_{L_i}$  denota las velocidades del último eslabón de la pata  $L_i$ ,  $v_{L_i}$  es la velocidad lineal en el extremo de la pata  $x_{L_i}$ ,  $\omega_{L_i}$  es la velocidad angular del eslabón donde esta localizado el extremo de la pata  $L_i$ ,  $\dot{q}_{L_i} = [\dot{q}_{1,L_i} \ \dot{q}_{2,L_i} \ \dot{q}_{3,L_i}]$ ,  $\xi_B = [v_{oB} \ \omega_B]$ ,  $v_{oB}$  la velocidad lineal del origen de  $\Sigma_B$ ,  $\omega_B$  la velocidad angular de la base,  $h_{j,L_i}$  son los ejes de giro de  $q_{j,L_i}$ ,  $a = h_{1,L_i}^W \times (s_{1,L_i}^W + s_{2,L_i}^W + s_{3,L_i}^W)$ ,  $b = h_{2,L_i}^W \times (s_{2,L_i}^W + s_{3,L_i}^W)$ ,  $c = h_{3,L_i}^W \times s_{3,L_i}^W$ ,  $\tilde{D}$  es la matriz que representa el producto cruz del vector  $D = (s_{0,L_i}^W + s_{1,L_i}^W + s_{2,L_i}^W + s_{3,L_i}^W)$  con  $\omega_B^W$  (tal que  $-\tilde{D}\omega_B^W = \omega_B^W \times D$ ),  $I \in \mathbb{R}^{3 \times 3}$  representa la matriz identidad y  $0 \in \mathbb{R}^{3 \times 3}$  es una matriz de ceros.

### D. Control cinemático

El caso de estudio a considerar será cuando el cuadrúpedo se encuentra de pie con sus cuatro patas en una posición constante de contacto con el suelo (esto es  $v_{L_i}^W \equiv 0$ ). Ahora, la ecuación (3) puede reescribirse en términos de Jacobianos como

$$\begin{bmatrix} v_{L_i}^W \\ \omega_{L_i}^W \end{bmatrix} = \begin{bmatrix} J_{L_i, \dot{q}_{L_i}, v} & J_{L_i, B, v} \\ J_{L_i, \dot{q}_{L_i}, \omega} & J_{L_i, B, \omega} \end{bmatrix} \begin{bmatrix} \dot{q}_{L_i} \\ \xi_B^W \end{bmatrix} \quad (4)$$

donde  $J_{L_i, \dot{q}_{L_i}, v} = [a \ b \ c] \in \mathbb{R}^{3 \times 3}$ ,  $J_{L_i, B, v} = [I \ -\tilde{D}] \in \mathbb{R}^{3 \times 6}$ ,  $J_{L_i, \dot{q}_{L_i}, \omega} = [h_{1, L_i}^W \ h_{2, L_i}^W \ h_{3, L_i}^W] \in \mathbb{R}^{3 \times 3}$  y  $J_{L_i, B, \omega} = [0 \ I] \in \mathbb{R}^{3 \times 6}$ . A continuación, supóngase que en las propuestas de movimiento deseado para el cuadrúpedo la base del robot no rota (esto es  $R_B^W = I$  y  $\omega_B^W = 0$ ); de tal manera que de la primera ecuación de (4) se tiene que

$$\begin{aligned} 0 &= J_{L_i, \dot{q}_{L_i}, v} \dot{q}_{L_i} + J_{L_i, B, v} \begin{bmatrix} v_{oB}^W \\ 0 \end{bmatrix} \\ &= J_{L_i, \dot{q}_{L_i}, v} \dot{q}_{L_i} + v_{oB}^W; \end{aligned} \quad (5)$$

es decir,

$$v_{oB}^W = -J_{L_i, \dot{q}_{L_i}, v} \dot{q}_{L_i}. \quad (6)$$

El objetivo de control será llevar asintóticamente la posición de la base a una deseada; es decir

$$\lim_{t \rightarrow \infty} x_{oB}^W(t) = x_{oBd}^W(t). \quad (7)$$

De tal manera que el error de posición se calculará como

$$e = x_{oBd}^W - x_{oB}^W. \quad (8)$$

Note que

$$\begin{aligned} \dot{e} &= \dot{x}_{oBd}^W - \dot{x}_{oB}^W \\ &= v_{oBd}^W - v_{oB}^W \\ &= \dot{x}_{oBd}^W + J_{L_i, \dot{q}_{L_i}, v} \dot{q}_{L_i}. \end{aligned} \quad (9)$$

Si se logra que

$$\dot{e} + Ke = 0 \quad (10)$$

con  $K \in \mathbb{R}^{3 \times 3}$  una matriz definida positiva (generalmente diagonal), el sistema es asintóticamente estable [6]. El error tiende a cero con una tasa de convergencia que depende de los valores propios de la matriz  $K$ . En este sentido, al sustituir (9) en (10) se puede resolver para  $\dot{q}_{L_i}$  (al haberse encontrado que  $J_{L_i, \dot{q}_{L_i}, v}$  es una matriz cuadrada. Nótese que esta matriz depende de la medición de las variables articulares  $q_{L_i}$  y que debe tenerse cuidado de que no sea singular); lo cual resulta

$$\dot{q}_{L_i} = -J_{L_i, \dot{q}_{L_i}, v}^{-1} (\dot{x}_{oBd}^W + Ke). \quad (11)$$

Para implementar (11) en modo posición note que, en una primera aproximación

$$\dot{q}_{L_i}(t) = \frac{q_{L_i}(t) - q_{L_i}(t-T)}{T} \quad (12)$$

con  $T$  el periodo de muestreo. Entonces, sustituyendo (12) en (11) y despejando  $q_{L_i}(t)$  se tiene finalmente que

$$q_{L_i}(t) = -T J_{L_i, \dot{q}_{L_i}, v}^{-1} (\dot{x}_{oBd}^W + Ke) + q_{L_i}(t-T). \quad (13)$$

### III. DESCRIPCIÓN DEL ROBOT CUADRÚPEDO

El robot cuadrúpedo utilizado (ver Fig. 2) está equipado con 12 servomotores Dynamixel AX-12A [5]; servomotores que han sido modificados [13] para operar en tiempo real a una frecuencia de muestreo de 1 [kHz] bajo el protocolo de comunicación SPI (Serial Peripheral Interface). Internamente a estos servomotores se les ha programado, en modo voltaje, un controlador PD con ganancias  $K_p = 10$  y  $K_d = 0.1$  para regular

la posición de su eje de carga. El robot cuadrúpedo también cuenta con un sistema de cómputo de tiempo real que será descrito a continuación.

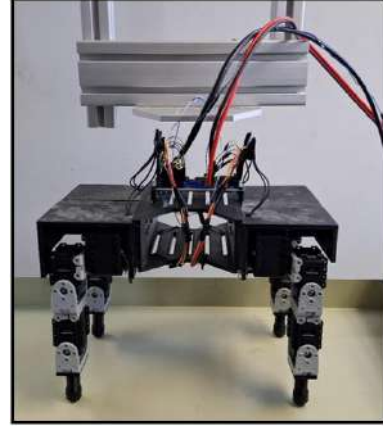


Fig. 2. Robot cuadrúpedo.

En la Fig. 3 se ilustra el esquema general del robot cuadrúpedo. Obsérvese que se cuenta con una computadora NUC (modelo NUC515RYK con el parche de Linux PREEMPT\_RT de tiempo real versión 5.10.90-rt61-rc1) quien realiza las operaciones principales del sistema robótico (cálculo de la ley de control, almacenamiento de datos, etc.) y un Arduino Mega 2560 encargado de la transferencia de datos por SPI entre la NUC y los 12 servomotores. Note que para acceder al protocolo SPI, la NUC utiliza el módulo W5100 quien traduce de UDP (User Datagram Protocol) a SPI y que para la asignación de una IP (Internet Protocol), necesaria para la comunicación, se utiliza el servicio DHCP (Dynamic Host Configuration Protocol) proporcionado por un módem.

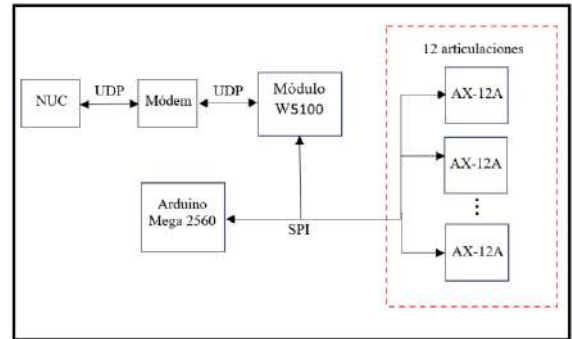


Fig. 3. Diagrama a bloques del robot cuadrúpedo.

El diagrama de flujo programado en la NUC se muestra en la Fig. 4. Y básicamente consiste en contar con ciertos mecanismos de protección y la organización de los tiempos del experimento de forma que de un tiempo inicial ( $t_{inicial}$ ) hasta  $t_{condicion\_inicial}$  no se realice alguna operación, de  $t_{condicion\_inicial}$  a  $t_{inicio\_exp}$  colocar el robot en una condición inicial (que, para evitar singularidades en el Jacobiano, se mueven las articulaciones 3 de cada pata a  $10 \frac{\pi}{180}$  [rad]) y de  $t_{inicio\_exp}$  a  $t_{final\_exp}$  correr propiamente el experimento. Finalmente, en la Fig. 5 se detalla el diagrama de

flujo programado en el Arduino; el cual, como se ha mencionado se encarga de la transferencia de datos por SPI entre la NUC y los servomotores.

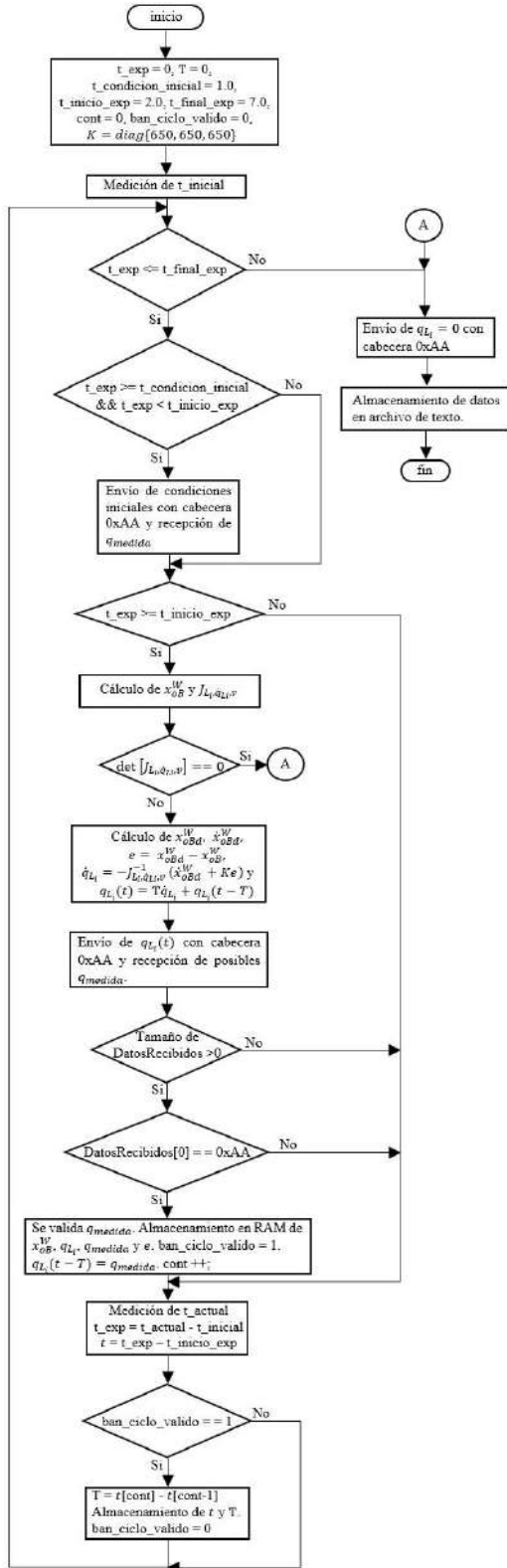


Fig. 4. Diagrama de flujo en la NUC.

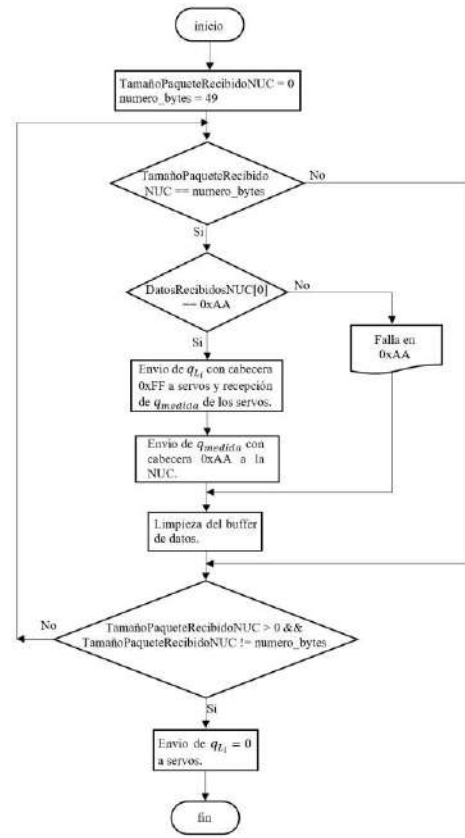


Fig. 5. Diagrama de flujo en el Arduino.

#### IV. EXPERIMENTOS

En la Tabla 1 se presentan los vectores de posición entre los orígenes de los marcos de coordenadas del robot cuadrúpedo bajo estudio, los cuales describen las dimensiones del robot. Para cumplir con el objetivo de este trabajo, se implementaron tres escenarios de control de movimiento. En cada uno de ellos se ejecutó el controlador (13) con  $K = \text{diag}\{650, 650, 650\}$  y se definieron las siguientes condiciones iniciales para cada articulación (buscando evitar singularidades en el Jacobiano al inicio del experimento):

$$q_{Li}(0) = \left[ 0 \quad 0 \quad 10 \frac{\pi}{180} \right]^T \text{ [rad].}$$

TABLA 1. VECTORES DE POSICIÓN ENTRE LOS ORÍGENES DE LOS MARCOS DE COORDENADAS DEL ROBOT CUADRÚPEDO

Pata 1 [m]	Pata 2 [m]	Pata 3 [m]	Pata 4 [m]
$s_{0,L_1}^B = \begin{bmatrix} -0.1254 \\ 0 \\ 0.0694 \end{bmatrix}$	$s_{0,L_2}^B = \begin{bmatrix} 0.1254 \\ 0 \\ 0.0694 \end{bmatrix}$	$s_{0,L_3}^B = \begin{bmatrix} -0.1254 \\ 0 \\ -0.0694 \end{bmatrix}$	$s_{0,L_4}^B = \begin{bmatrix} 0.1254 \\ 0 \\ -0.0694 \end{bmatrix}$
$s_{1,L_1}^1 = \begin{bmatrix} 0 \\ -0.0225 \\ 0 \end{bmatrix}$	$s_{2,L_2}^2 = \begin{bmatrix} 0 \\ -0.0694 \\ 0 \end{bmatrix}$	$s_{3,L_3}^3 = \begin{bmatrix} 0 \\ -0.0696 \\ 0 \end{bmatrix}$	

Nota: Si  $q_{Li} = 0$  entonces

$$\begin{aligned}
 x_{L_1}^W &= [-0.1254 \quad -0.1615 \quad 0.0694]^T \text{ [m]} \\
 x_{L_2}^W &= [0.1254 \quad -0.1615 \quad 0.0694]^T \text{ [m]} \\
 x_{L_3}^W &= [-0.1254 \quad -0.1615 \quad -0.0694]^T \text{ [m]} \\
 x_{L_4}^W &= [0.1254 \quad -0.1615 \quad -0.0694]^T \text{ [m]}.
 \end{aligned}$$

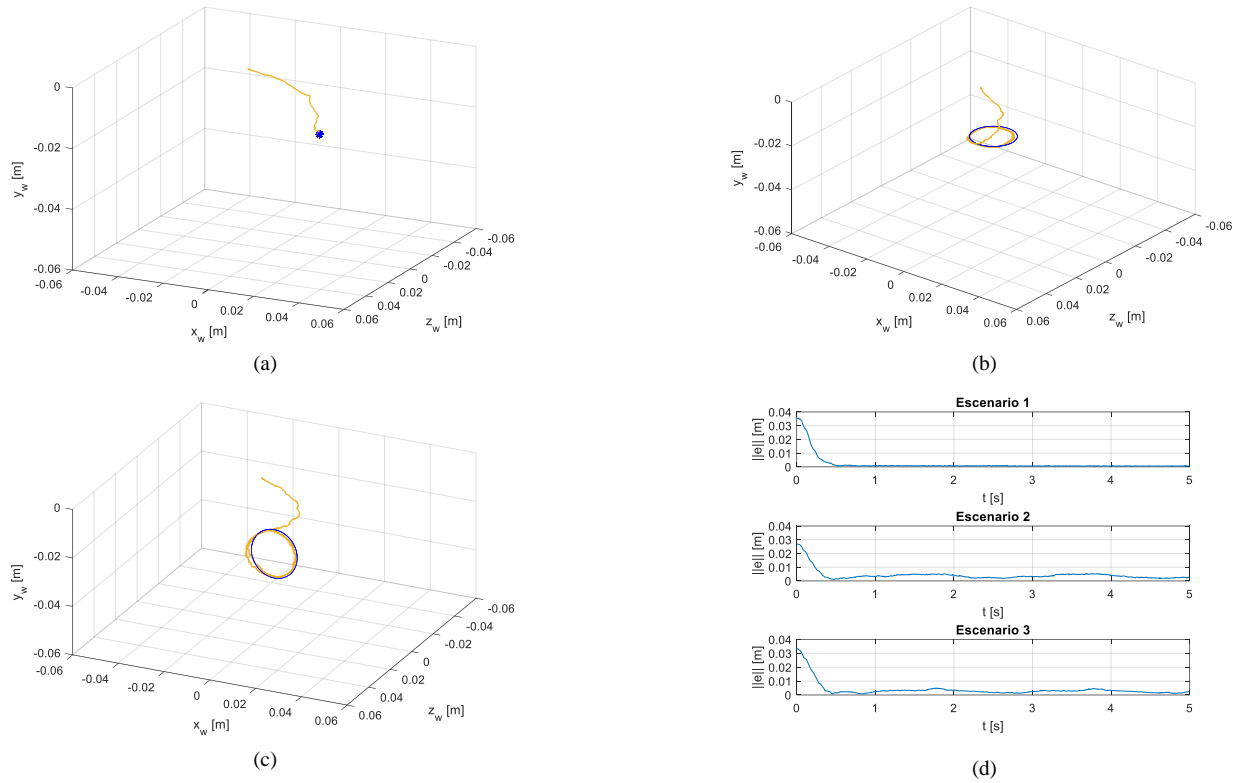


Fig. 6. Trayectoria de  $x_{oB}^W$  en (a) Escenario 1, (b) Escenario 2 y (c) Escenario 3. (d) Gráficas de  $\|e\|$  contra el tiempo para los tres escenarios.

#### A. Escenario 1

El Escenario 1 corresponde con uno de regulación, tal que

$$x_{oBd}^W = \begin{bmatrix} 0.02 \\ -0.02 \\ 0 \end{bmatrix} [\text{m}].$$

En la Fig. 6(a) se observa la trayectoria de  $x_{oB}^W$  para este escenario de experimentación, nótese que la trayectoria tiende a la posición constante deseada marcada con un “\*” azul. En la primera parte de la Fig. 6(d) se muestra la evolución contra el tiempo de la norma del error para este caso; el cual en un tiempo de 5 [s] ya está en aproximadamente 0.001 [m].

#### B. Escenario 2

En el Escenario 2 se estableció una trayectoria circular deseada para  $x_{oB}^W$  en el plano  $x_W$ - $z_W$  (un plano horizontal) a una altura de  $y_W = -0.02$  [m] definida como

$$x_{oBd}^W = \begin{bmatrix} r\cos(2\pi ft) \\ -0.02 \\ r\sin(2\pi ft) \end{bmatrix} [\text{m}]$$

donde  $r = 0.01$  [m] es el radio del círculo deseado y  $f = 0.5$  [Hz] es su frecuencia. En la Fig. 6(b) se presenta la trayectoria de  $x_{oB}^W$  para este escenario experimental, destacándose que

tiende a la trayectoria circular deseada marcada con un trazo azul. En la segunda parte de la Fig. 6(d) se aprecia la evolución contra el tiempo de la norma del error para este caso; que para cuestiones prácticas oscila alrededor de los 0.0038 [m] después de los 2 [s].

#### C. Escenario 3

En el Escenario 3 también se estableció una trayectoria circular deseada para  $x_{oB}^W$  pero ahora en el plano vertical  $x_W$ - $y_W$  y se definió como

$$x_{oBd}^W = \begin{bmatrix} r\cos(2\pi ft) \\ r\sin(2\pi ft) - 0.03 \\ 0 \end{bmatrix} [\text{m}]$$

con  $r = 0.01$  [m] y  $f = 0.5$  [Hz]. En la Fig. 6(c) se presenta el resultado experimental para este escenario notándose que también se tiende a la trayectoria circular deseada (visualizada en un trazo azul) con un error que oscila alrededor de también 0.0038 [m] después de los 2 [s]; según se aprecia en la tercera parte de la Fig. 6(d).

#### D. Análisis de periodicidad de los escenarios

Para el análisis de la periodicidad de los experimentos se exponen las siguientes gráficas correspondientes a los tiempos de ciclo de cada uno de los escenarios realizados. En la Fig. 7 se observan los tiempos medidos de cada ciclo por escenario y en la Fig. 8 sus histogramas.

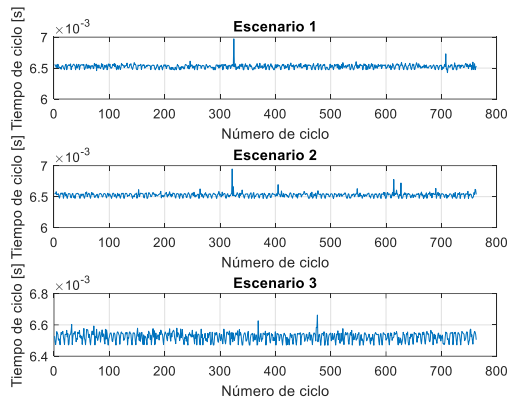


Fig. 7. Gráficas de tiempo de ciclo.

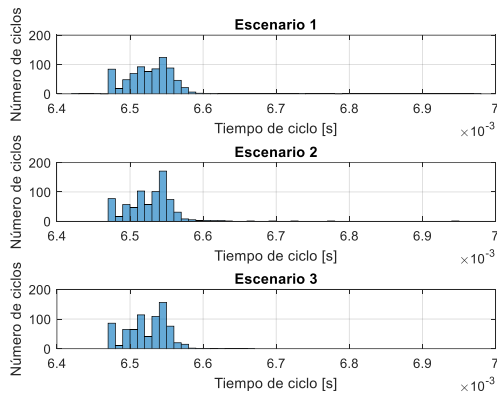


Fig. 8. Histogramas de tiempos de ciclo.

Finalmente, en la Tabla 2 se ilustran los datos estadísticos correspondientes de los tiempos de ciclo para cada escenario. Nótese que en los tres escenarios se tuvo un tiempo promedio de ciclo de 0.0065 [s], que el peor caso de tiempo de ejecución (WCET) fue de 0.006972 [s] ocurrido en el Escenario 1 y que el mejor caso de tiempo de ejecución (BCET) fue de 0.006425 [s] también para el Escenario 1. Lo cual da muestra de que los escenarios experimentados fueron de tiempo real con una relativa alta velocidad de procesamiento.

TABLA 2. DATOS ESTADÍSTICOS DE LOS TIEMPOS DE CICLO

	Escenario 1 [s]	Escenario 2 [s]	Escenario 3 [s]
Tiempo promedio	0.006526	0.006528	0.006524
Peor caso de tiempo de ejecución (WCET)	0.006972	0.006947	0.006663
Mejor caso de tiempo de ejecución (BCET)	0.006425	0.00647	0.00647
Desviación estándar	0.00003393	0.00003495	0.00002805

## V. CONCLUSIONES

En este trabajo se detallan pruebas experimentales de tiempo real para el control de movimiento de un robot cuadrúpedo de bajo costo utilizándose el SOTR Linux con el

parche PREEMPT\_RT. Para este propósito primero fue analizado su modelo cinemático diseñándose una ley de control en este contexto. Luego se experimentaron tres escenarios, uno de ellos de regulación y dos de seguimiento de trayectoria, obteniéndose resultados satisfactorios. Se destaca que en cada escenario experimentado se obtuvo una periodicidad del lazo de control bastante aceptable pues se logró en promedio un periodo de muestreo de 0.0065 [s] y un peor caso de tiempo de ejecución (WCET) de 0.006972 [s]. Lo anterior da muestra de la posibilidad de realizar experimentos de tiempo real y relativa alta velocidad de procesamiento en diversas propuestas de control de movimiento para este robot cuadrúpedo bajo estudio.

## AGRADECIMIENTOS

Se agradece el apoyo financiero del TecNM, del Instituto Tecnológico de Ensenada y del CONAHCYT.

## REFERENCIAS

- [1] L. Orbegoso, C. Rodríguez & E. Valverd. "Control cinemático para un robot cuadrúpedo usando el Método de Newton-Raphson", Revista Elektron, Vol. 5, No. 1, pp. 77-82, 2021.
- [2] A. Pechev. "Inverse Kinematics without matrix inversion", IEEE International Conference on Robotics and Automation, 2008.
- [3] A. Aristidou & J. Lasenby. "FABRIK: A fast, iterative solver for the Inverse Kinematics problem", Elsevier, Graphical Models 73, 243–260, doi:10.1016/j.gmod.2011.05.003, 2011.
- [4] M. A. Sen, V. Bakircioglu & M. Kalyoncu. "Inverse Kinematic Analysis Of A Quadruped Robot", INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 6, ISSUE 09, 2017.
- [5] J. C. Rojas, E. Bugarin & A. Y. Aguilar. "On Hierarchical Quadratic Programming for the Whole body Control of Quadruped Robot", Memorias del XX Congreso Mexicano de Robótica, COMRob, 2018.
- [6] B. Siciliano, L. Sciacivco, L. Villani & G. Oriolo. "Robotics: Modelling, Planning and Control", Springer, 2009.
- [7] R. Oshana. DSP for Embedded and Real-Time systems. Elsevier B.V. 2012.
- [8] A. Burns & A. Wellings, "Sistemas de Tiempo Real y Lenguajes de Programación", 3a ed. Madrid: Addison Wesley, 2002.
- [9] B. Katz, J. D. Carlo & S. Kim. "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control", International Conference on Robotics and Automation (ICRA), 2019.
- [10] M. Hutter, C. Gehring, D. Jud, A. Lauber1, C. D. Bellicoso1, V. Tsounis, J. Hwangbo1, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer & M. Hoepflinger. "ANYmal - A Highly Mobile and Dynamic Quadruped Robot". IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.
- [11] Y. Boltov, I. Skarga-Bandurova, I. Kotsiuba, M. Hrushka, G. Krivoulyya & R. Siriak. "Performance Evaluation of Real-Time System for Vision-Based Navigation of Small Autonomous Mobile Robots". The 10h IEEE International Conference on Dependable Systems, Services and Technologies, 2019.
- [12] J. L. Muñoz-Torres, E. Bugarin, E. Rodríguez-Orozco, J. A. Rojas-Quintero & A. Y. Aguilar-Bustos. "Comparativa de un sistema de visión de tiempo real bajo Xenomai y PREEMPT\_RT". Pádi Boletín Científico De Ciencias Básicas E Ingenierías Del ICBI, 12(Especial2), 74-79. <https://doi.org/10.29057/icbi.v12iEspecial2.12268>, 2024.
- [13] B. Sánchez, E. Bugarin, & A. Y. Aguilar. "Mejoramiento de un servomotor inteligente de bajo costo". Pádi Boletín Científico De Ciencias Básicas E Ingenierías Del ICBI, 11(Especial2), 127-133. <https://doi.org/10.29057/icbi.v11iEspecial2.10852>, 2023.

# Comparative Analysis of Forward Kinematics in Robotic Manipulators: Homogeneous Matrices, Screw Theory, and Quaternions

Jesús A. Medrano-Hermosillo

*Graduate Studies and Research Division  
Tecnológico Nacional de México  
Chihuahua, México  
jesus.mh@chihuahua.tecnm.mx*

Abraham E. Rodríguez-Mata

*Graduate Studies and Research Division  
Tecnológico Nacional de México  
Chihuahua, Mexico  
abraham.rm@chihuahua.tecnm.mx*

Victor A. Gonzalez-Huitron

*Systems and Computing Department  
Tecnológico Nacional de México  
Querétaro, México  
victor.gh@queretaro.tecnm.mx*

Jesús Ricardo López Gutiérrez

*INAOE  
Puebla, México  
jrlopez@inaoep.mx*

Rafael Sandoval Rodríguez

*Tecnológico Nacional de México  
Chihuahua, Mexico  
rafael.sr@chihuahua.tecnm.mx*

Larbi Djilali

*Graduate Studies and Research Division  
Tecnológico Nacional de México  
Chihuahua, México  
larbidjar@chihuahua.tecnm.mx*

**Abstract**—This article seeks to introduce new readers interested in robotics to some tools available in the field. It specifically focuses on the study and comparison of kinematic analysis in robotic systems, with an emphasis on forward kinematics using a range of techniques. These techniques range from traditional methods to advanced ones. In particular, the concepts of forward kinematics are explored using traditional methods such as homogeneous matrices, geometric studies like screw theory, quaternions, and geometric algebra. All this is presented from a theoretical approach accompanied by practical examples, making it ideal for new readers interested in the topic.

**Index Terms**—Kinematics, Screw Theory, Quaternions.

## I. INTRODUCTION

In robotics, kinematics is essential to study the relationship between the coordinates of a robot's joints and its movement in space [1]. Kinematics is divided into forward kinematics, which calculates the position and orientation of the end effector from known joint angles, and inverse kinematics, which determines the joint angles required for a specific position [1], [2]. To calculate forward kinematics, one begins with the assignment of coordinate frames, which can vary according to the D-H convention used by various algorithms detailed in [2]. The parameters D-H, which contain information about rotation and position of each degree of freedom, are then calculated to construct the corresponding homogeneous matrices. This topic has been widely studied, as shown in the following paragraphs with references to traditional works [3].

The following works provide a formal examination of homogeneous matrices. In [4], the authors calculate the forward kinematics for a redundant 4 degrees of freedom robot using the traditional method of D-H matrices and homogeneous matrices. The study includes a numerical simulation to illustrate the results. In [5], the kinematics of a robotic arm

is analyzed using homogeneous matrices, with illustrations of the robot's behavior for different angular configurations. Other studies can be found in [6]–[8]. Forward kinematics have been widely studied, especially with increasing degrees of freedom and associated D-H parameters. There are works on the calculation of forward kinematics without D-H parameters. In [9], a recursive algorithm with Euler angles and rotation matrices is used for a computationally efficient and easy-to-implement solution. In [10], kinematics is analyzed with neural networks using EBPDC, demonstrating its performance on a 7 degrees of freedom robotic arm. Other works with recursive algorithms include [11]–[13]. However, these algorithms can be difficult to program and may require expensive hardware to process.

Quaternions and screw theory are alternative techniques to D-H parameters, recently used in robotics [14]. In [15], the authors apply dual quaternions to calculate the forward kinematics of a 6 degrees of freedom robotic arm, highlighting its shorter calculation time compared to the D-H formalism. Furthermore, the study decomposes the kinematic problem into subproblems, thus reducing its complexity. In [16], the authors highlight that modern robotics approaches emphasize exponential formulations, utilizing dual quaternions to analyze and compare with matrix methods. Their results demonstrate a 30–40 % speedup compared to traditional techniques. Other relevant works using quaternions are seen in [17]–[19]. On the other hand, screw theory is a crucial mathematical tool in modern robotics [20], focused on calculating spatial mechanisms. This approach has recently gained relevance, addressing fundamental concepts such as basic dynamics and linear algebra. Other studies on forward kinematics using screw theory can be found in [21]–[23].

This article aims to help new users understand various methodologies in forward kinematics, including traditional

methods with homogeneous matrices, screw theory, geometric algebra and quaternions. The document is organized as follows: Section II presents the theoretical bases of the different techniques; Section III addresses the various examples; and Section IV describes conclusions and future work.

## II. MATHEMATICAL FUNDAMENTALS

### A. Homogeneous Transformation Matrices

In forward kinematics, the position and orientation of the end-effector are calculated from the robot's joint angles, as described in  $P = \varphi(\theta)$ . Here,  $P \in \mathbb{R}^3$  represents the position of the end-effector,  $\theta$  denotes each degree of freedom, and  $\varphi : \theta \rightarrow p \mid \theta \in \mathbb{R}^n$ , where  $n$  is the number of degrees of freedom of the robot. In practice,  $\varphi$  performs the domain conversion using the Lie group  $SE(3)$ , which is described as [24]:

$$SE(3) := \left\{ \begin{pmatrix} R & P \\ 0 & 1 \end{pmatrix} : R \in SO(3), P \in \mathbb{R}^3 \right\} \quad (1)$$

In the literature, the above expression is formulated as a homogeneous matrix that encompasses the rotations and positions of one coordinate system with respect to another, determined by the Denavit-Hartenberg parameters. An example of a homogeneous matrix is expressed as follows:

$$T_{k-1}^k = \begin{bmatrix} \cos \theta_k & -\sin \theta_k \cos \alpha_k & \sin \theta_k \sin \alpha_k & a_k \cos \theta_k \\ \sin \theta_k & \cos \theta_k \cos \alpha_k & -\cos \theta_k \sin \alpha_k & a_k \sin \theta_k \\ 0 & \sin \alpha_k & \cos \alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where  $\theta_k, \alpha_k, a_k, d_k \in \mathbb{R}$  represent the Denavit-Hartenberg parameters of one coordinate system with respect to another. In other words, there will be a homogeneous matrix for each degree of freedom of the system, and one more coordinate system than each degree of freedom. However, since the main objective of forward kinematics is to find the relationship between the base and the end-effector, it is necessary to perform a multiplication of all the obtained homogeneous matrices to find the base-end-effector relationship. Therefore, for the calculation, the following operation is necessary:

$$T_0^n = \prod_{k=0}^{n-1} T_k^{k+1} \quad (3)$$

### B. Screw Theory

The tangent space is a smooth path ( $\gamma$ ) through the identity ( $e$ ) in a group ( $G$ ):

$$\gamma = \mathbb{R} \rightarrow G \quad (4)$$

where  $\gamma(0) = e$ . In other words, the tangent space can be calculated from the derivatives in some local coordinate system around  $e$ . As indicated above, the tangent space can be used to calculate the elements of the Lie algebra of various Lie groups, where in a rigid body, the Lie algebra describes its motion [24]. In traditional robotics, based on the group  $SE(3)$ , where  $R$

is the rotation matrix and  $P$  is the translation vector at each joint. Taking (1) as a parametrized function of the form:

$$\gamma = t \rightarrow SE(3) \quad (5)$$

Thus, differentiating the group around the identity yields:

$$se(3) := \left\{ \begin{pmatrix} \Omega & v \\ 0 & 0 \end{pmatrix} : \Omega \in so(3), v \in \mathbb{R}^3 \right\} \quad (6)$$

$se(3)$  represents the Lie algebra of the Lie group  $SE(3)$ . In robotics,  $\Omega$  and  $v$  illustrate the angular velocity and linear velocity respectively [20]. The elements of the Lie algebra are usually written as follows:

$$S = \begin{pmatrix} \Omega \\ v \end{pmatrix} \quad (7)$$

where  $S$  is the line constructed by two three-dimensional vectors called a screw. For position control, it is essential to calculate the position of the end-effector with each change in the various angles. Therefore, it is important to evaluate the behavior of the robot in the following way. Let  $X(0) = (x, y, z)^T$  be a point on the end-effector when the robot is in its initial position. Thus, using screw theory, the position of the end-effector of a serial robot with 6 degrees of freedom can be calculated as:

$$X = K(\theta) \begin{pmatrix} X(0) \\ 1 \end{pmatrix}, K(\theta) = e^{\theta_1 S_1} e^{\theta_2 S_2} e^{\theta_3 S_3} e^{\theta_4 S_4} e^{\theta_5 S_5} e^{\theta_6 S_6} \quad (8)$$

$\theta_j$  ( $j = 1, 2, \dots, 6$ ) represents the joint angle. Using a traditional approach, each  $e^{\theta_j S_j}$  would represent the Lie group  $SE(3)$ . Approximately  $SE(3)$ , the group can be described by unifying two sets of matrices. One is  $SO(3)$ , which represents the rotation of the rigid body about a given axis, and the other is the translation matrix. The group  $SO(3)$  is straightforward to calculate using the rotation matrices of the traditional approach. However, the translation matrix can be determined through the following conjugation:

$$\begin{pmatrix} R & P \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} I_3 & P(0) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I_3 & -P(0) \\ 0 & 1 \end{pmatrix} \quad (9)$$

where  $I_3$  is the  $3 \times 3$  identity matrix and  $P(0)$  is a starting point on the rotation axis. Thus, the translation is given by the vector:

$$P = (I_3 - R)P(0) \quad (10)$$

Finally, with  $SE(3)$  and (8), the position of the end-effector can be calculated as:

$$X = \prod_{i=1}^n R_i X(0) + \sum_{i=1}^n \prod_{j=0}^{i-1} R_j P_i \quad (11)$$

In the previous equation,  $R_0 = I_3$ .

TABLE I  
BASIC OPERATIONS WITH QUATERNIONS

Operation	Representation
Multiplication by a constant	$\lambda \mathbf{q}_1 = [\lambda w_1, \lambda \mathbf{v}_1]$
Addition	$\mathbf{q}_1 + \mathbf{q}_2 = [w_1 + w_2, \mathbf{v}_1 + \mathbf{v}_2]$
Quaternion multiplication	$\mathbf{q}_1 \odot \mathbf{q}_2 = [w_1 w_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, w_1 \mathbf{v}_2 + w_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2]$
Conjugate	$\bar{\mathbf{q}} = [w, -\mathbf{v}]$
Norm	$\ \mathbf{q}\  = \sqrt{\mathbf{q}\bar{\mathbf{q}}}$

### C. Quaternions

Quaternions are four-dimensional vectors with distinct algebraic operations, introduced by William R. Hamilton. They have applications in fields such as quantum mechanics, computer graphics, electrodynamics, robotics, etc [25]. A quaternion is defined as:

$$\mathbf{q} = w + xi + yj + zk, \quad (12)$$

where  $w, x, y$ , and  $z$  are real numbers, and  $i, j$ , and  $k$  are imaginary units, where  $i^2 = j^2 = k^2 = -1$ . It is important not to confuse these elements with other sections that are used as iterative variables [26]. Alternatively, a quaternion can be represented in a two-component form consisting of a scalar part and a vector part:

$$\mathbf{q} = [w, \mathbf{v}], \quad \mathbf{v} = [x, y, z], \quad (13)$$

here  $w$  is the scalar part and  $\mathbf{v}$  is the vector part. This representation defines the vector  $\mathbf{v}$  with coordinates  $x, y, z$  along the unit vectors  $i, j, k$ , respectively [26]. Some of the most important operations with quaternions can be seen in Table I. However, although quaternions are excellent for representing rotations, they have certain limitations for translations, which is why they were replaced by dual quaternions. Dual quaternions can be considered as an extension of quaternions, as they represent both rotations and translations of a rigid body.

Let  $\mathbf{q}_R = \cos \frac{\theta}{2} + \hat{u} \sin \frac{\theta}{2}$  be a quaternion representing a rotation around the unit vector  $\hat{u}$  through an angle  $\theta$ . Its conjugate is  $\mathbf{q}_R^* = \cos \frac{\theta}{2} - \hat{u} \sin \frac{\theta}{2}$  such that  $\mathbf{q}_R \mathbf{q}_R^* = \mathbf{q}_R^* \mathbf{q}_R = 1$ . This represents the equivalent of  $SO(3)$ . On the other hand, let  $t = (t_1, t_2, t_3)$  be a translation. A point  $v$  under the rotation  $R$  followed by the translation  $t$  becomes the point  $Rv + t$  (which represents  $SE(3)$ ). The above sequence of transformation  $R, t$  can be compactly represented by a dual quaternion. The translation vector  $t$  is a pure quaternion. We combine it with the rotation quaternion  $\mathbf{q}_R$  into the following dual quaternion:

$$\hat{\mathbf{q}} = \mathbf{q}_R + \frac{\epsilon}{2} t \mathbf{q}_R = \cos \frac{\theta}{2} + \hat{u} \sin \frac{\theta}{2} + \frac{\epsilon}{2} \left( -\sin \frac{\theta}{2} (t \cdot \hat{u}) + \cos \frac{\theta}{2} t + \sin \frac{\theta}{2} t \times \hat{u} \right) \quad (14)$$

where,  $\epsilon \neq 0$  and  $\epsilon^2 = 0$  is the dual operator. If the transformation is a pure rotation, i.e.,  $t = 0$ , we end up with  $\hat{\mathbf{q}} = \mathbf{q}_R$ . If the transformation is a pure translation, i.e.,  $\theta = 0$ , we end up with  $\hat{\mathbf{q}} = 1 + \frac{\epsilon}{2} t$ . To obtain the relation of one

coordinate system with respect to another, a point  $\hat{\mathbf{p}}$  is assigned where the moved point  $\hat{\mathbf{p}}'$  is represented by:

$$\hat{\mathbf{p}}' = \hat{\mathbf{q}} \otimes \hat{\mathbf{p}} \otimes \bar{\hat{\mathbf{q}}} \quad (15)$$

### D. Geometric algebra

Geometric Algebra (GA) is an algebra that combines the advantages of quaternions and vector algebra (also called the unified and universal language of natural sciences). GA is a linear space of dimension  $2^n$  denoted by  $\mathbb{G}_n$ , where  $n = p + q + r$ . It is important to note that  $p, q$ , and  $r$  are the number of bases that square to  $+1, -1$ , and  $0$ , respectively. Unlike vector multiplication, GA has its own multiplication called the *geometric or Clifford product* [27]. The geometric product is a non-commutative operation that is associative and distributive over addition. For two vectors, the Clifford product is written as follows:

$$ab = a \cdot b + a \wedge b \quad (16)$$

The right-hand side of (16) illustrates two elements; the first is the dot product, which is common in vector calculus. The second is the wedge product (antisymmetric part), where the wedge product (also called the exterior product) is an associative, distributive, and anticommutative operation ( $a \wedge b = -b \wedge a$ ). The components calculated by the Clifford product of  $k$  linearly independent vectors span the  $k$ -vector space. Each element of this space is called a  $k$ -vector and is represented as  $\langle A \rangle_k$ , where  $k$  indicates the grade [28]. The combination of  $k$ -vectors is called a multivector and is illustrated as follows:

$$A = \langle A \rangle_1 + \langle A \rangle_2 + \dots + \langle A \rangle_n \quad (17)$$

In addition to (16), the Clifford product of two multivectors  $A_t$  and  $B_s$ , where  $t$  and  $s$  represent the grade of each multivector, is:

$$A_t B_s = \langle AB \rangle_{t+s} + \langle AB \rangle_{t+s-2} + \dots + \langle AB \rangle_{|t-s|} \quad (18)$$

where  $\langle AB \rangle_k$  denotes the  $k$ -vector of the multivector  $A_t B_s$  [27], [28]. In GA,  $e_i$  is used to represent the various basis vectors, where  $1 \leq i \leq n$ . The Clifford product of two basis vectors is calculated as:

$$\{1; e_i; e_i \wedge e_j; e_i \wedge e_j \wedge e_k; \dots; I = e_1 \wedge \dots \wedge e_n\} \quad (19)$$

$I$  is known as the pseudoscalar, where this variable is the identity of the algebra. On the other hand, using a quaternionic approach, the various bases could be written as:  $i = e_1, j = e_2, k = e_3$ . The aforementioned basis vectors lead to the multivector for the entire GA:

$$e_i e_j = \begin{cases} 1 & \text{for } i = j \in 1, \dots, p \\ -1 & \text{for } i = j \in p+1, \dots, p+q \\ 0 & \text{for } i = j \in p+q+1, \dots, p+q+r \\ e_i \wedge e_j & \text{for } i \neq j \end{cases} \quad (20)$$

Some important definitions that are useful in GA are included below. For more details, see [28].

*Definition 1:* Suppose a multivector of grade  $t$ , i.e.,  $A_t \in \mathbb{G}_n$ . Then, the reverse of  $A_t$ , denoted as  $\widetilde{A}_t$ , is defined by:

$$\widetilde{A}_t = \sum_{i=0}^t (-1)^{\frac{i(i-1)}{2}} \langle A_t \rangle_i \quad (21)$$

*Definition 2:* Suppose a multivector of grade  $t$   $A_t \in \mathbb{G}_n$ . Then, the rotation of  $A_t$ , denoted as  $A'_t$ , is defined by the following versor product:

$$A'_t = R_\theta A_t \widetilde{R}_\theta = e^{-\frac{\theta}{2}L} A_t e^{\frac{\theta}{2}L} \quad (22)$$

where  $R_\theta$  is the rotor,  $\theta$  is the angle of rotation, and  $L$  is the Lie algebra generator for the rotor. The bivector (or 2-vectors)  $L$  is simply the representation of rotation by quaternions in a different form, taking  $e_2 e_3 \rightarrow \mathbf{i}$ ,  $e_3 e_1 \rightarrow \mathbf{j}$ , and  $e_1 e_2 \rightarrow \mathbf{k}$ .

*Definition 3:* Based on (11), the position of the tool tip can be represented using geometric algebra as:

$$X = \prod_{i=1}^n R_{\theta_i} X(0) \prod_{i=1}^n \widetilde{R}_{\theta_{n-i+1}} + \sum_{i=1}^n \prod_{j=0}^{i-1} R_{\theta_j} P_i \prod_{j=0}^{i-1} \widetilde{R}_{\theta_{i-j-1}} \quad (23)$$

where  $R_{\theta_0} = 1$  and  $X(0), P \in \mathbb{G}_3$ . In GA, the translation is calculated as  $P = P(0) - R_\theta P(0) \widetilde{R}_\theta$  [29].

### III. RESULTS

#### A. Example using homogeneous matrices

Consider the robot in Figure 1, where the Denavit-Hartenberg (DH) parameters of the two degrees of freedom (DOF) robot are: For the first joint ( $n = 1$ ), the parameter  $\theta$  is  $\theta_1$ ,  $d$  is 0m,  $\alpha$  is  $0^\circ$ , and  $a$  is  $l_1$ . For the second joint ( $n = 2$ ), the parameter  $\theta$  is  $\theta_2$ ,  $d$  is 0m,  $\alpha$  is  $0^\circ$ , and  $a$  is  $l_2$ . This structured representation of the DH parameters facilitates the analysis and understanding of the robot's kinematics [1]. Substituting the parameters into (2) yields the following results:

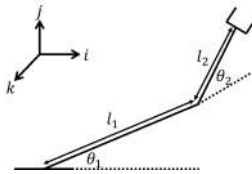


Fig. 1. Two degrees of freedom robot

$$T_0^1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & l_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & l_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

$$T_1^2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

Now, applying (3), we proceed with the calculation of the matrix that defines the position and rotation of the end-effector with respect to the base. The result is as follows:

$$T_0^2 = \begin{bmatrix} \cdot & \cdot & \cdot & l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ \cdot & \cdot & \cdot & l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \\ \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

In the previous matrix, the fourth column represents the relative position between the base and the end-effector. For space reasons, the part describing the rotational relationship was omitted; however, it can be easily obtained through the mentioned matrix multiplication. When talking about forward kinematics using homogeneous matrices, the analysis directly relies on implementing the D-H parameters in the matrices and performing the necessary multiplications. While this widely used technique is popular, it can increase computational costs due to unnecessary multiplicative parameters, such as perspective and scale vectors. Additionally, the matrix multiplication process incurs significant computational costs.

#### B. Example using screw theory

Consider the 2-DOF robot in Figure 1. Now, using (11), the position of the end-effector is calculated as follows [29]:

$$X = R_1 R_2 X(0) + P_1 + R_1 P_2 \quad (27)$$

where the rotation matrices are as follows:

$$R_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 \\ \sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \mid i = 1, 2 \quad (28)$$

taking  $P_1(0) = [0 \ 0 \ 0]^T$  as the initial base position, the first translation vector with (10) is:

$$P_1 = (I_3 - R_1)P_1(0) = [0 \ 0 \ 0]^T \quad (29)$$

and the second translation vector with the initial joint position  $P_2(0) = [l_1 \ 0 \ 0]^T$  is:

$$P_2 = [l_1(1 - \cos \theta_2) \ -l_1 \sin \theta_2 \ 0]^T \quad (30)$$

Finally, since the initial position of the end-effector is  $X(0) = [l_1 + l_2 \ 0 \ 0]^T$ , its position is determined by:

$$X = \begin{pmatrix} l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \\ 0 \end{pmatrix} \quad (31)$$

As observed, screw theory offers a straightforward solution to forward kinematics problems, and its ease of application is commendable. However, it's important to recognize that, similar to using homogeneous coordinates, the  $SO(3)$  group is essential in this context.

### C. Example using dual quaternions

Consider the 2-DoF robot in Figure 1. First, it is necessary to obtain the dual quaternions for each link. Therefore, we proceed to obtain the rotation quaternions first. The rotation quaternions are:

$$\mathbf{q}_{R1} = \cos \frac{\theta_1}{2} + \mathbf{k} \sin \frac{\theta_1}{2} \quad (32)$$

$$\mathbf{q}_{R2} = \cos \frac{\theta_2}{2} + \mathbf{k} \sin \frac{\theta_2}{2} \quad (33)$$

While the translation quaternions are:

$$t\mathbf{q}_{R1} = 0 + \frac{1}{2}(l_1\mathbf{k}) \quad (34)$$

$$t\mathbf{q}_{R2} = 0 + \frac{1}{2}(l_2\mathbf{k}) \quad (35)$$

With the above, the dual quaternions are expressed as follows:

$$\hat{\mathbf{q}}_1 = \mathbf{q}_{R1} + \epsilon t\mathbf{q}_{R1} \quad (36)$$

$$\hat{\mathbf{q}}_2 = \mathbf{q}_{R2} + \epsilon t\mathbf{q}_{R2} \quad (37)$$

Thus, the final position of the end of the second link, taking as a base  $p = (l_1 + l_2)i$ , is calculated as:

$$\hat{\mathbf{p}}' = \hat{\mathbf{q}}_1 \otimes (\hat{\mathbf{q}}_2 \otimes \hat{\mathbf{p}} \otimes \bar{\hat{\mathbf{q}}}_2) \otimes \bar{\hat{\mathbf{q}}}_1 \quad (38)$$

Where, when expanded, we get the following result:

$$p' = \begin{pmatrix} l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \\ 0 \end{pmatrix} \quad (39)$$

The use of dual quaternions can effectively reduce computational costs compared to the homogeneous matrix method; however, it requires a solid understanding of quaternions and their operations. Nonetheless, it remains a powerful tool for solving forward kinematics.

### D. Example using geometric algebra

Consider the 2-DoF robot in Figure 1. Using Definition 3, the position of the end-effector is written as:

$$X = R_{\theta_1} R_{\theta_2} X(0) \tilde{R}_{\theta_2} \tilde{R}_{\theta_1} + P_1 + R_{\theta_1} P_2 \tilde{R}_{\theta_1} \quad (40)$$

Since the axis of rotation is in  $\mathbf{k}$ , the Lie algebra generator is  $e_1 e_2$ . Thus, the rotors are:

$$\begin{aligned} R_{\theta_i} &= e^{-\frac{\theta_i}{2} e_1 e_2} = \cos \frac{\theta_i}{2} - e_1 e_2 \sin \frac{\theta_i}{2} \quad | \quad i = 1, 2 \\ \tilde{R}_{\theta_i} &= e^{\frac{\theta_i}{2} e_1 e_2} = \cos \frac{\theta_i}{2} + e_1 e_2 \sin \frac{\theta_i}{2} \quad | \quad i = 1, 2 \end{aligned} \quad (41)$$

taking  $P_1(0) = 0$  as the initial base position, the first translation vector is:

$$P_1 = P_1(0) - R_{\theta_1} P_1(0) \tilde{R}_{\theta_1} = 0 \quad (42)$$

and the second with  $P_2(0) = l_1 e_1$  is:

$$P_2 = l_1 e_1 - \left( \cos \frac{\theta_2}{2} - e_1 e_2 \sin \frac{\theta_2}{2} \right) l_1 e_1 \left( \cos \frac{\theta_2}{2} + e_1 e_2 \sin \frac{\theta_2}{2} \right) \quad (43)$$

multiplying the previous equation and using (20) yields:

$$P_2 = l_1 e_1 - \left( l_1 \cos \frac{\theta_2}{2} e_1 + l_1 \sin \frac{\theta_2}{2} e_2 \right) \left( \cos \frac{\theta_2}{2} + e_1 e_2 \sin \frac{\theta_2}{2} \right) \quad (44)$$

doing some algebra and applying some trigonometric identities:

$$P_2 = l_1 (1 - \cos \theta_2) e_1 - l_1 \sin \theta_2 e_2 \quad (45)$$

Finally, since the initial position of the end-effector is  $X(0) = (l_1 + l_2) e_1$ . Now, using (40), the position of the end-effector is:

$$\begin{aligned} X &= (l_1 + l_2) (\cos(\theta_1 + \theta_2) e_1 + \sin(\theta_1 + \theta_2) e_2) \\ &\quad + l_1 (\cos \theta_1 e_1 - \cos(\theta_1 + \theta_2) e_1) \\ &\quad + l_1 (\sin \theta_1 e_2 - \sin(\theta_1 + \theta_2) e_2) \end{aligned} \quad (46)$$

or,

$$\begin{aligned} X &= l_1 \cos \theta_1 e_1 + l_2 \cos(\theta_1 + \theta_2) e_1 \\ &\quad + l_1 \sin \theta_1 e_2 + l_2 \sin(\theta_1 + \theta_2) e_2 \end{aligned} \quad (47)$$

Using geometric algebra for forward kinematics involves solving through an iterative equation, much like in screw theory. This approach avoids the use of D-H parameters. In this context, the  $e_1$  axis corresponds to the first row of the homogeneous matrix or the x-axis, while the  $e_2$  axis represents the y-axis, or the second row of the matrix. However, employing this technique requires familiarity with Clifford algebra, as the standard rules of matrix space do not apply.

## IV. COMPARISON

Based on the results obtained, it is clear that the forward kinematics yield consistent outcomes across the various tools employed. In other words, the position of the end-effector can be computed as follows: along the x-axis, the expression is  $l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2)$ , and along the y-axis, it is  $l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)$ . The choice or comparison based on computational cost or other metrics is beyond the scope of this work, which focuses on providing tools and examples for new readers interested in forward kinematics. However, to use this equation, it is essential to know the angular values of each joint, specifically  $\theta_1$  and  $\theta_2$ . With this information, the equation yields the Cartesian coordinates of the robot's end-effector. A graphical example of using this equation is shown in Figure 2, which illustrates the positions of the joints at different angular values and  $l_1 = l_2 = 1m$ .

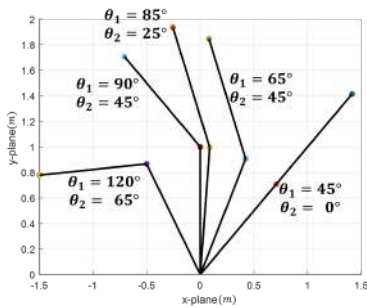


Fig. 2. Comparison of the robot's behavior under varying angular values

## V. CONCLUSIONS

This work investigated various methods for obtaining forward kinematics in mechatronic systems through different techniques. We began with the traditional approach using homogeneous matrices, examining their derivation and behavior. However, this method involved multiple matrix multiplications, leading to increased computational costs. We also explored alternative techniques, such as dual quaternions, which allowed for iterative application. While effective, transitioning to dynamics posed challenges, as dual quaternions can be complex for those new to robotics. Additionally, we introduced geometric methods, including screw theory, which provided a compelling alternative by bypassing the D-H algorithm and utilizing iterative equations for forward kinematics. Geometric algebra was also considered, which avoided D-H parameters and employed an iterative framework. Although its applications were less common in the literature due to its non-matrix formulation, it offered significant advantages, such as facilitating the computation of robot speeds through dot products. Overall, this exploration highlighted the variety of methods available for forward kinematics, each with its own advantages and limitations. By examining the techniques discussed, new readers gained a clearer understanding of the tools available, enabling them to select those that best suited their specific needs. Furthermore, the examples provided empowered readers to analyze and replicate these tools for their applications.

Future research should analyze the computational cost of these techniques to identify the most suitable for specific applications. Additionally, applying these methodologies to parallel robots remains an unexplored area in need of investigation.

## REFERENCES

- [1] B. Siciliano and O. Khatib, eds., *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer, 2008.
- [2] J. Craig, *Introduction to Robotics, eBook, Global Edition*. Pearson Education, 2021.
- [3] R. Singh, V. Kukshal, and V. S. Yadav, "A review on forward and inverse kinematics of classical serial manipulators," *Advances in Engineering Design: Select Proceedings of ICOIED 2020*, pp. 417–428, 2021.
- [4] Y. Li and L. Wang, "Kinematic model and redundant space analysis of 4-dof redundant robot," *Mathematics*, vol. 10, no. 4, 2022.
- [5] F. A. Alamsyah, "The kinematics analysis of robotic arm manipulators cylindrical robot RPP type for FFF 3d print using scilab," *IOP Conference Series: Materials Science and Engineering*, vol. 494, p. 012100, mar 2019.

- [6] D. Constantin, M. Lupoae, C. Baciuc, and B. Ilie, "Forward kinematic analysis of an industrial robot," 03 2015.
- [7] D. Constantin, "Kinematic analysis of the lynx6 robotic arm," vol. XXV, pp. 65–76, 03 2015.
- [8] Wang, Wenli, Wang, Peng, Zhao, Yongguo, Zhu, Yunhai, and Zhao, Jie, "Kinematics and workspace analysis of a robotic arm for medical delivery robots," *E3S Web Conf.*, vol. 257, p. 01067, 2021.
- [9] F. Gonçalves, T. Ribeiro, A. F. Ribeiro, G. Lopes, and P. Flores, "A recursive algorithm for the forward kinematic analysis of robotic systems using euler angles," *Robotics*, vol. 11, no. 1, 2022.
- [10] R. F. Reinhart and J. J. Steil, "Recurrent neural associative learning of forward and inverse kinematics for movement generation of the redundant pa-10 robot," in *2008 ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems (LAB-RS)*, pp. 35–40, 2008.
- [11] S. T. Kanneganti, J. Pei, and D. F. Hougen, "Developing interpretable machine learning for forward kinematics of robotic arms," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 01–09, 2021.
- [12] F. Tavassolian, H. Khotanlou, and P. Varshovi-Jaghargh, "Forward kinematics analysis of a 3-prr planer parallel robot using a combined method based on the neural network," in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 320–325, 2018.
- [13] R. J. Chandran, I. J. Raglend, and M. D. Anand, "Forward kinematic solution for a five joint robot using artificial neural network," in *2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]*, pp. 1704–1709, 2014.
- [14] J. G. Farias, E. De Pieri, and D. Martins, "A review on the applications of dual quaternions," *Machines*, vol. 12, no. 6, p. 402, 2024.
- [15] W. Ge, L. Chen, X. Wang, E. Xing, and T. Zielinska, "Kinematics modeling and analysis of manipulator using the dual quaternion," in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 750–755, 2019.
- [16] N. T. Dantam, "Robust and efficient forward, differential, and inverse kinematics using dual quaternions," *The International Journal of Robotics Research*, vol. 40, no. 10–11, pp. 1087–1105, 2021.
- [17] D. Pavlo, D. Grangier, and M. Auli, "Quaternion: A quaternion-based recurrent model for human motion," *arXiv preprint arXiv:1805.06485*, 2018.
- [18] N. Zivkovic, J. Vidakovic, S. Mitrovic, and M. Lazarevic, "Implementation of dual quaternion-based forward kinematics algorithm in ros," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4, IEEE, 2022.
- [19] A. Valverde and P. Tsiotras, "Spacecraft robot kinematics using dual quaternions," *Robotics*, vol. 7, no. 4, p. 64, 2018.
- [20] K. Lynch and F. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.
- [21] M. Cardona and C. G. Cena, "Direct kinematics and jacobian analysis of exoskeleton robots using screw theory and simscape multibody™," in *2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX)*, pp. 1–6, 2019.
- [22] Y. Li, Y. Zhao, T. Zhang, and T. Li, "Forward kinematics analysis and experiment of hybrid high-altitude board installation robot based on screw theory," *Advances in Mechanical Engineering*, vol. 11, no. 4, p. 1687814019846266, 2019.
- [23] C. hua MAN, X. FAN, C. rong LI, and Z. hui ZHAO, "Kinematics analysis based on screw theory of a humanoid robot," *Journal of China University of Mining and Technology*, vol. 17, no. 1, pp. 49–52, 2007.
- [24] J. M. Selig, *Geometric Fundamentals of Robotics (Monographs in Computer Science)*. SpringerVerlag, 2004.
- [25] E. Özgür and Y. Mezouar, "Kinematic modeling and control of a robot arm using unit dual quaternions," *Robotics and Autonomous Systems*, vol. 77, pp. 66–73, 2016.
- [26] N. Zivković, J. Vidaković, and M. Lazarević, "Forward kinematics algorithm in dual quaternion space based on denavit-hartenberg convention," *Applied Engineering Letters*, vol. 8, no. 2, pp. 52–59, 2023.
- [27] D. Hestenes, *New Foundations for Classical Mechanics*. Fundamental Theories of Physics, Springer Netherlands, 1999.
- [28] E. Bayro-Corrochano, *Geometric Computing - for Wavelet Transforms, Robot Vision, Learning, Control and Action*. 01 2010.
- [29] J. A. Medrano-Hermosillo, R. Lozoya-Ponce, J. Ramírez-Quintana, and R. Baray-Arana, "Forward kinematics analysis of 6-dof articulated robot using screw theory and geometric algebra," in *2022 XXIV Robotics Mexican Congress (COMRob)*, pp. 1–6, IEEE, 2022.

# Análisis de Aceleración de un Manipulador Paralelo 6 UPUR

Fredy Uriel Nava-Andrés  
*Instituto de Electrónica y Mecatrónica*  
*Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca, México  
naaf990628@gs.utm.mx

Manuel Arias-Montiel  
*Instituto de Electrónica y Mecatrónica*  
*Universidad Tecnológica de la Mixteca*  
Huajuapán de León, Oaxaca, México  
mam@mixteco.utm.mx

**Resumen**—Este trabajo complementa el análisis de posición y de velocidad de un manipulador paralelo con estructura 6 UPUR, desarrollando el análisis de aceleración mediante la teoría de tornillos. Se toma como punto de partida el análisis de velocidad previamente reportado en la literatura para redefinir los tornillos que permiten extender el análisis a aceleración, asegurando las condiciones necesarias para que dicho análisis esté completo. Las ecuaciones de aceleración obtenidas son validadas mediante simulaciones numéricas realizadas en software especializado para el análisis de sistemas multi-cuerpos.

**Palabras Clave**—análisis de aceleración, manipulador paralelo, teoría de tornillos

## I. INTRODUCCIÓN

La dificultad de obtener el análisis cinemático de manipuladores paralelos, es una de las principales desventajas de este tipo de robots, como se ha reportado ampliamente en la literatura especializada [1]–[3]. El análisis directo de posición es el primer problema que se presenta porque las ecuaciones de cerradura de lazo normalmente conllevan a un sistema de ecuaciones no lineales cuya solución no es única. El problema se complica aún más para los análisis de velocidad y de aceleración, pues el sistema de ecuaciones no lineales se tiene que derivar con respecto al tiempo y la manipulación algebraica resulta excesivamente laboriosa, más aún si lo que se desea es una representación entrada-salida en donde no se vean involucradas las variables cinemáticas de las juntas pasivas.

En los últimos años, se ha utilizado de forma exitosa la teoría de tornillos infinitesimales para resolver este tipo de problemas. Esta teoría es atribuida a Robert Ball a finales del siglo XIX, e inicialmente fue aplicada a la dinámica de cuerpos rígidos. No obstante, diversos autores la han utilizado para obtener el análisis cinemático de los manipuladores paralelos debido a que gracias a ella se omite el uso de derivadas y un gran número de operaciones matemáticas. La teoría de tornillos es útil para describir la movilidad de los manipuladores paralelos de manera sistemática. El análisis cinemático de los manipuladores se puede escalar al siguiente orden (velocidad) con facilidad gracias al uso del álgebra de Lie. En contraste, el análisis de aceleración no fue nada fácil de hallar, de hecho, antes de los trabajos de Brand, Mises, Sugimoto y Duffy donde definen y aplican tanto el álgebra como el cálculo motor

en los manipuladores, la teoría de tornillos era considerada únicamente para los análisis cinemáticos de primer orden. De tal modo que gracias a ellos, entre muchos otros, la teoría de tornillos pudo alcanzar los análisis cinemáticos de orden superior, tales como el análisis de aceleración, pulso e hiperpulsos de cualquier manipulador [3].

En [4] el análisis cinemático de velocidad y de aceleración del robot Delta, es desarrollado usando teoría de tornillos. Las ecuaciones de entrada-salida de velocidad y de aceleración son obtenidas sistemáticamente utilizando el concepto de tornillo recíproco. Además, el análisis de singularidades del robot se realiza a partir de las ecuaciones de entrada-salida de velocidad. Gallardo-Alvarado y García-Murillo [5] presentan el análisis cinemático de un robot paralelo generador del movimiento de Schönflies por medio de la teoría de tornillos. Las ecuaciones entrada-salida de velocidad y de aceleración se obtienen sistemáticamente recurriendo a la teoría de tornillos recíprocos. Para ello, el robot se modela como si fuese un manipulador paralelo de seis grados de libertad gracias a la incorporación de pares cinemáticos ficticios que conectan las extremidades con la plataforma fija y una cadena cinemática virtual con la finalidad de aplicar sin restricciones el álgebra de Lie. El análisis de singularidades se aborda con base en la ecuación entrada-salida de velocidad. La teoría de tornillo también ha sido usada para desarrollar los modelos cinemáticos de manipuladores con arquitectura híbrida, como en [6], en donde el análisis cinemático de velocidad y de aceleración es obtenido desacoplando los movimientos lineales y angulares y usando la teoría de los tornillos recíprocos. Con el propósito de demostrar la versatilidad que posee la teoría de tornillos para desarrollar el análisis cinemático de diferentes tipos de manipuladores, en [7] se emplea esta teoría para realizar el análisis cinemático de velocidad de un manipulador paralelo cuya plataforma móvil es configurable, dando la posibilidad de poseer dos efectores finales. Los resultados obtenidos se comparan con los resultados de simulaciones numéricas realizadas con el software ADAMS VIEW, demostrando la exactitud de la teoría de tornillos para diferentes tipos de manipuladores.

Este trabajo complementa el análisis de posición y de velocidad de un manipulador paralelo con estructura 6 UPUR reportado en [8], desarrollando el análisis de aceleración

mediante la teoría de tornillos. Las ecuaciones de aceleración obtenidas son validadas mediante simulaciones numéricas realizadas en software especializado para el análisis de sistemas multi-cuerpos.

## II. CASO DE ESTUDIO

### A. Descripción del Manipulador Paralelo

El manipulador paralelo considerado en este trabajo consiste en una plataforma fija que se une a una plataforma móvil a través de seis cadenas cinemáticas las cuales constan de tres tipos de pares cinemáticos inferiores, dos juntas universales (U), una junta prismática (P) y una junta de revoluta (R) que une a la junta universal con la plataforma móvil, razón por la cual se denomina 6-UPUR. Es importante aclarar que la letra subrayada denota la junta actuada dentro de cada cadena cinemática. En la Figura 1, estas juntas son representadas por cilindros, dos cilindros pequeños de color plata para el caso de las juntas universales y un cilindro color verde para el caso de las juntas prismáticas. En cuanto a las juntas de revolutas, estas se encuentran en la parte superior de la plataforma móvil en forma de rodamientos que le permiten girar a la cadena cinemática alrededor de un eje perpendicular a la plataforma móvil.

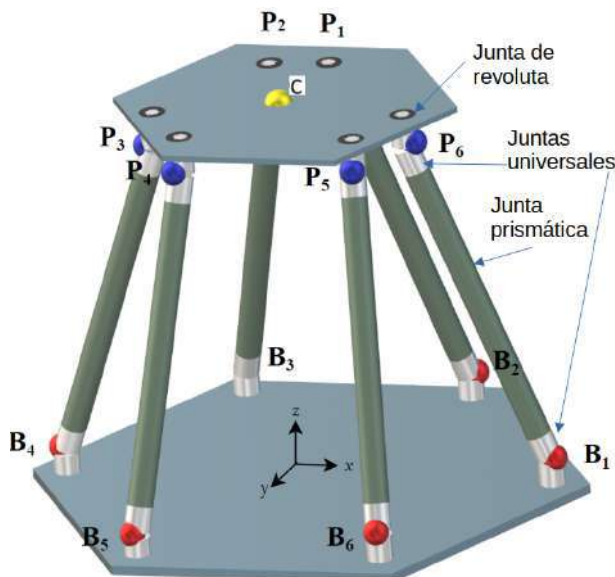


Fig. 1. Manipulador paralelo 6-UPUR.

### B. Puntos de Interés

Los puntos de interés para el análisis cinemático del manipulador paralelo 6-UPUR son: el punto de origen, donde se encuentra el marco de referencia fijo; el punto sobre el cual se obtendrá la cinemática, conocido como el polo de referencia (punto  $C$ ); los puntos que ubican las juntas universales inferiores (puntos  $B_i$ ) y los puntos que ubican las juntas universales superiores (puntos  $P_i$ ). Las coordenadas de estos puntos con respecto al marco fijo de referencia ubicado en el centro de la plataforma fija se muestran en la Tabla 1.

TABLA I  
COORDENADAS INICIALES DE LOS PUNTOS DE INTERÉS (EN MM)

Puntos	[x	y	z]
$B_1$	[200	0	21.5]
$B_2$	[100	173.2	21.5]
$B_3$	[-100	173.2	21.5]
$B_4$	[-200	0	21.5]
$B_5$	[-100	-173.2	21.5]
$B_6$	[100	173.2	21.5]
$P_1$	[14.01	114.14	275.5]
$P_2$	[-14.01	114.4	275.5]
$P_3$	[-105.85	-44.93	275.5]
$P_4$	[-91.84	-69.2	275.5]
$P_5$	[91.84	-69.2	275.5]
$P_6$	[105.85	-44.93	275.5]
$C$	[0	0	299.82]

### C. Definición de los Tornillos

Como se menciona en [8], un tornillo consiste en un par de vectores concatenados con una parte primaria y una parte dual. La parte primaria corresponde a un vector unitario a lo largo del eje de giro del tornillo, asociado a la velocidad angular del cuerpo, y la parte dual es un segundo vector que representa el momento producido por la parte primaria alrededor de un punto de referencia fijo (polo de referencia), que se asocia al componente de velocidad lineal.

La definición de los tornillos de cualquier manipulador se torna elemental antes de desarrollar los análisis cinemáticos, pues hay que definir la dirección, la posición y el polo de referencia de cada tornillo para que estos describan la cinemática correctamente. Por un lado, encontrar la posición y la dirección de los tornillos resulta sencillo, solo requiere de la representación en forma de tornillo de los pares cinemáticos inferiores que describen la configuración bajo la que se encuentre el manipulador. Por otro lado, el polo de referencia debe encontrarse en el cuerpo de interés y ser definido de acuerdo a la geometría del manipulador durante todo el análisis cinemático, haciendo de esto una tarea meticulosa.

En la Figura 2 se presenta el esquema de una de las seis idénticas cadenas cinemáticas que unen la base fija con la plataforma móvil del manipulador 6 UPUR. Esta representación toma como base los pares cinemáticos inferiores y las restricciones obtenidas en [8] para el análisis cinemático de velocidad. La cadena cinemática mostrada contiene los tornillos numerados del uno al seis, donde cada superíndice representa un cuerpo rígido que va desde la plataforma fija, cuerpo 0, hasta la plataforma móvil, cuerpo 6.

Para obtener las expresiones que definan a los tornillos, primero se requiere definir un marco de referencia fijo. En este análisis se usa el marco de referencia cuyo origen (punto  $O$ ) se encuentra en el punto central sobre la superficie superior de la plataforma fija, como se ilustra en la Figura 3.

Con el marco de referencia establecido, resta definir el polo de referencia. Dado que el análisis cinemático del manipulador 6-UPUR debe de obtener el movimiento de la plataforma móvil con respecto a la plataforma fija, entonces el polo de referencia para el análisis será el centro de la plataforma

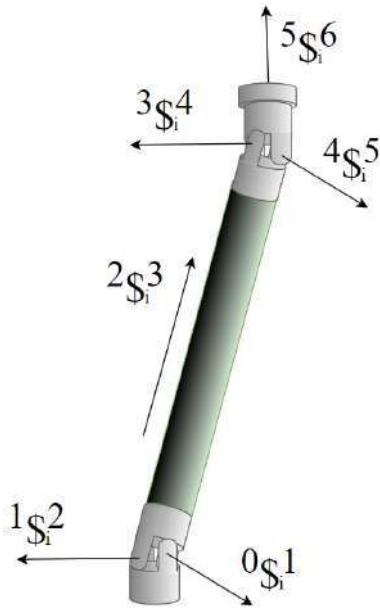


Fig. 2. Definición de tornillos para el manipulador 6-UPUR.

móvil (punto  $C$ ). Para encontrar los vectores correspondientes a la parte dual de cada tornillo, se requiere conocer las características geométricas del manipulador, específicamente los vectores de posición de los puntos  $B_i$  y  $P_i$  del manipulador en las seis cadenas cinemáticas.

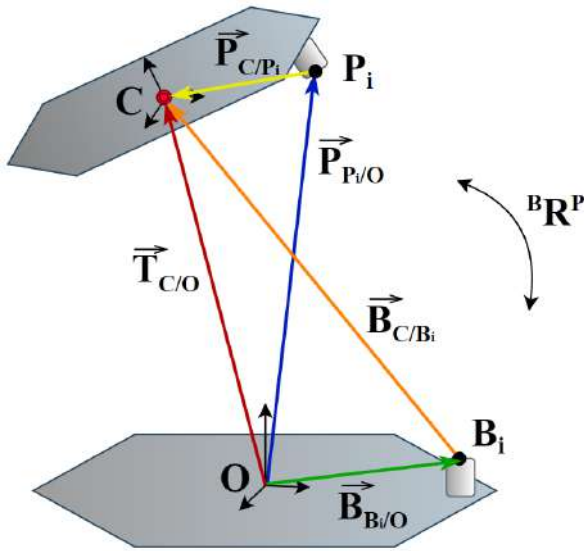


Fig. 3. Vectores de posición para la definición de tornillos.

En la Figura 3 se observan las características geométricas del manipulador bajo análisis, en ella los puntos  $B_i$  y  $P_i$  son representados con respecto al marco de referencia fijo por los vectores  $\vec{B}_{B_i/O}$  y  $\vec{P}_{P_i/O}$ , respectivamente, mientras que la posición de la plataforma móvil se define mediante el vector  $\vec{T}_{C/O}$ . Con esta figura, se muestra que los vectores  $\vec{B}_{C/B_i}$  y  $\vec{P}_{C/P_i}$  son los necesarios para definir la parte dual de cada

tornillo. El vector  $\vec{B}_{C/B_i}$  se obtiene de la diferencia entre los vectores  $\vec{T}_{C/O}$  y  $\vec{B}_{B_i/O}$ , mientras que la diferencia entre los vectores  $\vec{T}_{C/O}$  y  $\vec{P}_{P_i/O}$  tiene como resultado al vector  $\vec{P}_{C/P_i}$ .

Los tornillos quedan definidos por las expresiones dadas en (1).

$$\begin{aligned} {}^0\mathcal{S}_i^1 &= \begin{bmatrix} {}^0S_i^1 \\ {}^0S_i^1 \times (T_{C/O} - B_{B_i/O}) \end{bmatrix} & {}^2\mathcal{S}_i^3 &= \begin{bmatrix} 0 \\ {}^2S_i^3 \end{bmatrix} \\ {}^1\mathcal{S}_i^2 &= \begin{bmatrix} {}^1S_i^2 \\ {}^1S_i^2 \times (T_{C/O} - B_{B_i/O}) \end{bmatrix} & {}^3\mathcal{S}_i^4 &= \begin{bmatrix} {}^3S_i^4 \\ {}^3S_i^4 \times (T_{C/O} - P_{P_i/O}) \end{bmatrix} \\ {}^5\mathcal{S}_i^6 &= \begin{bmatrix} {}^5S_i^6 \\ {}^5S_i^6 \times (T_{C/O} - P_{P_i/O}) \end{bmatrix} & {}^4\mathcal{S}_i^5 &= \begin{bmatrix} {}^4S_i^5 \\ {}^4S_i^5 \times (T_{C/O} - P_{P_i/O}) \end{bmatrix} \end{aligned} \quad (1)$$

donde  ${}^jS_i^k$  es el vector unitario que define la dirección de cada tornillo de la cadena cinemática de acuerdo con la Figura 2.

Para eliminar las variables cinemáticas de las juntas pasivas del análisis y obtener una representación entrada-salida de velocidad y de aceleración, se define el tornillo recíproco siguiendo las condiciones establecidas en [9] como

$${}^2\mathcal{S}_{ri}^3 = \begin{bmatrix} {}^2S_{ri}^3 \\ {}^2S_{ri}^3 \times ({}^O_B T_{OP} - P_i) \end{bmatrix} \quad (2)$$

### III. ANÁLISIS DE VELOCIDAD

El análisis directo de velocidad consiste en encontrar el estado de velocidad de la plataforma móvil a partir de las velocidades de cada una de las juntas actuadas, siendo uno de los principales problemas eliminar las velocidades articulares pasivas del manipulador. Esta complejidad aumenta con el número de cadenas cinemáticas que el manipulador presente. A través del álgebra de Lie, la teoría de tornillos se encarga de resolver este tipo de inconvenientes, pues logra que el análisis cinemático del manipulador dependa únicamente de las contribuciones de los movimientos articulares actuados, dejando de lado el conocimiento de los movimientos articulares pasivos.

Para el análisis de velocidad del manipulador paralelo 6 UPUR mediante la teoría de tornillos, se considera que el estado velocidad puede representarse como una combinación lineal de los tornillos presentes en cualquiera de sus cadenas cinemáticas [3]–[8]. De tal forma que el estado de velocidad de la plataforma móvil puede representarse con la sumatoria de los tornillos mostrados en la Figura 2, tal y como se muestra en (3).

$$\begin{aligned} {}^B V_o^P &= {}_0\omega_1^i {}^0\mathcal{S}_i^1 + {}_1\omega_2^i {}^1\mathcal{S}_i^2 + {}_2\omega_3^i {}^2\mathcal{S}_i^3 + {}_3\omega_4^i {}^3\mathcal{S}_i^4 \\ &+ {}_4\omega_5^i {}^4\mathcal{S}_i^5 + {}_5\omega_6^i {}^5\mathcal{S}_i^6 \end{aligned} \quad (3)$$

Obsérvese que en (3) se encuentran las velocidades de todas las juntas del manipulador, incluyendo las velocidades

articulares pasivas. No obstante, al aplicar la forma de Klein con la base de tornillos recíprocos definida en (2), el estado de velocidad se expresa como

$$\begin{aligned} \{^B V_o^P, {}^2 \mathbb{S}_{ri}^3\} &= \{\cancel{0\dot{\omega}_1^i} \cancel{0\mathbb{S}_i^1, {}^2 \mathbb{S}_{ri}^3}\} + \{\cancel{1\dot{\omega}_2^i} \cancel{1\mathbb{S}_i^2, {}^2 \mathbb{S}_{ri}^3}\} \\ &+ \{2\dot{\omega}_3^i \ 2\mathbb{S}_i^3, {}^2 \mathbb{S}_{ri}^3\} + \{\cancel{3\dot{\omega}_4^i} \cancel{3\mathbb{S}_i^4, {}^2 \mathbb{S}_{ri}^3}\} \\ &+ \{\cancel{4\dot{\omega}_5^i} \cancel{4\mathbb{S}_i^5, {}^2 \mathbb{S}_{ri}^3}\} + \{\cancel{5\dot{\omega}_6^i} \cancel{5\mathbb{S}_i^6, {}^2 \mathbb{S}_{ri}^3}\} \end{aligned} \quad (4)$$

que puede reducirse a

$$\{^B V_o^P, {}^2 \mathbb{S}_{ri}^3\} = 2\dot{\omega}_3^i \quad (5)$$

El estado de velocidad puede despejarse al desarrollar la forma de Klein. Sin embargo, una forma sistemática y compacta de obtenerlo es representar la ecuación (5) en su forma matricial, ya que la forma de Klein guarda similitudes con el producto entre matrices. Para realizar esto, hay que acomodar las matrices de manera que el producto entre estas pueda efectuarse, además de anexas el operador de polaridad [3], lo que conduce a:

$$J^T \Delta^B V_o^P = \dot{Q} \quad (6)$$

A (6) se le conoce como la ecuación de velocidad entrada-salida, donde  $^B V_o^P$  es el estado de velocidad de la plataforma móvil del manipulador,  $J$  es la matriz jacobiana directa asociada a la base de tornillos recíprocos del manipulador que se representa como:

$$J = \begin{bmatrix} {}^2 S_{r1}^3 & {}^2 S_{r2}^3 & {}^2 S_{r3}^3 & {}^2 S_{r4}^3 & {}^2 S_{r5}^3 & {}^2 S_{r6}^3 \\ {}^2 S_{o r1}^3 & {}^2 S_{o r2}^3 & {}^2 S_{o r3}^3 & {}^2 S_{o r4}^3 & {}^2 S_{o r5}^3 & {}^2 S_{o r6}^3 \end{bmatrix} \quad (7)$$

$\dot{Q}$  es el vector columna de las velocidades articulares generalizadas dado por:

$$\dot{Q} = \begin{bmatrix} {}^2 \dot{q}_1^3 & {}^2 \dot{q}_2^3 & {}^2 \dot{q}_3^3 & {}^2 \dot{q}_4^3 & {}^2 \dot{q}_5^3 & {}^2 \dot{q}_6^3 \end{bmatrix}^T \quad (8)$$

y  $\Delta$  es el operador de polaridad, definido en (9) como una matriz de  $6 \times 6$  [3].

$$\Delta = \begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix} \quad (9)$$

donde  $I_3$  es la matriz identidad de  $3 \times 3$ .

#### IV. ANÁLISIS DE ACELERACIÓN

El análisis directo de aceleración presenta los mismos inconvenientes del análisis de velocidad, pues se requiere obtener la aceleración tanto angular como lineal del cuerpo de interés a partir de las aceleraciones articulares en cada una de las juntas en sus extremidades, incluyendo las aceleraciones articulares pasivas. Por tal motivo, al definir el estado de aceleración como una combinación lineal de tornillos, el concepto de tornillo recíproco puede aplicarse para eliminar la influencia de las aceleraciones articulares pasivas.

Los tornillos que se usarán para este fin serán los mismos que se muestran en la Figura 2, por lo tanto, el estado de aceleración reducido del manipulador bajo análisis en forma de tornillo satisface:

$$\begin{aligned} {}^B A_o^P &= 0\dot{\omega}_1^i \ 0\mathbb{S}_i^1 + 1\dot{\omega}_2^i \ 1\mathbb{S}_i^2 + 2\dot{\omega}_3^i \ 2\mathbb{S}_i^3 + 3\dot{\omega}_4^i \ 3\mathbb{S}_i^4 \\ &+ 4\dot{\omega}_5^i \ 4\mathbb{S}_i^5 + 5\dot{\omega}_6^i \ 5\mathbb{S}_i^6 + \mathcal{A}_i \end{aligned} \quad (10)$$

donde  $\mathcal{A}_i$  es definido por:

$$\begin{aligned} \mathcal{A}_i &= [0\dot{\omega}_1^i \ 0\mathbb{S}_i^1 \ 1\dot{\omega}_2^i \ 1\mathbb{S}_i^2 + 2\dot{\omega}_3^i \ 2\mathbb{S}_i^3 + 3\dot{\omega}_4^i \ 3\mathbb{S}_i^4 + 4\dot{\omega}_5^i \ 4\mathbb{S}_i^5 + 5\dot{\omega}_6^i \ 5\mathbb{S}_i^6] \\ &+ [1\dot{\omega}_2^i \ 1\mathbb{S}_i^2 \ 2\dot{\omega}_3^i \ 2\mathbb{S}_i^3 + 3\dot{\omega}_4^i \ 3\mathbb{S}_i^4 + 4\dot{\omega}_5^i \ 4\mathbb{S}_i^5 + 5\dot{\omega}_6^i \ 5\mathbb{S}_i^6] \\ &+ [2\dot{\omega}_3^i \ 2\mathbb{S}_i^3 \ 3\dot{\omega}_4^i \ 3\mathbb{S}_i^4 + 4\dot{\omega}_5^i \ 4\mathbb{S}_i^5 + 5\dot{\omega}_6^i \ 5\mathbb{S}_i^6] \\ &+ [3\dot{\omega}_4^i \ 3\mathbb{S}_i^4 \ 4\dot{\omega}_5^i \ 4\mathbb{S}_i^5 + 5\dot{\omega}_6^i \ 5\mathbb{S}_i^6] \\ &+ [4\dot{\omega}_5^i \ 4\mathbb{S}_i^5 \ 5\dot{\omega}_6^i \ 5\mathbb{S}_i^6] \end{aligned} \quad (11)$$

Obsérvese que (11) contiene elementos encontrados durante el análisis de velocidad, de modo que las únicas variables desconocidas en (10) son las aceleraciones articulares pasivas, y al igual que en el análisis de velocidad, el análisis de aceleración puede desarrollarse a partir de las aceleraciones articulares generalizadas aplicando la forma de Klein al estado de aceleración reducido.

Considerando que (10) es la combinación lineal de los tornillos que se utilizaron en el análisis de velocidad, entonces la base de tornillos recíprocos es la misma que está dada por (2). Por lo tanto, aplicando la forma de Klein con la base de tornillos recíprocos a (10) se obtiene:

$$\begin{aligned} \{^B A_o^P, {}^2 \mathbb{S}_{ri}^3\} &= \{\cancel{0\dot{\omega}_1^i} \cancel{0\mathbb{S}_i^1, {}^2 \mathbb{S}_{ri}^3}\} + \{\cancel{1\dot{\omega}_2^i} \cancel{1\mathbb{S}_i^2, {}^2 \mathbb{S}_{ri}^3}\} \\ &+ \{2\dot{\omega}_3^i \ 2\mathbb{S}_i^3, {}^2 \mathbb{S}_{ri}^3\} + \{\cancel{3\dot{\omega}_4^i} \cancel{3\mathbb{S}_i^4, {}^2 \mathbb{S}_{ri}^3}\} \\ &+ \{\cancel{4\dot{\omega}_5^i} \cancel{4\mathbb{S}_i^5, {}^2 \mathbb{S}_{ri}^3}\} + \{\cancel{5\dot{\omega}_6^i} \cancel{5\mathbb{S}_i^6, {}^2 \mathbb{S}_{ri}^3}\} \\ &+ \{\mathcal{A}_i, {}^2 \mathbb{S}_{ri}^3\} \end{aligned} \quad (12)$$

reduciendo términos se tiene que

$$\{^B A_o^P, {}^2 \mathbb{S}_{ri}^3\} = 2\dot{\omega}_3^i + \{\mathcal{A}_i, {}^2 \mathbb{S}_{ri}^3\} \quad (13)$$

y pasando (13) a forma matricial resulta:

$$J \Delta^B A_o^P = \ddot{S} + C \quad (14)$$

donde  $J$  es la misma matriz jacobiana directa dada en (7),  $\Delta$  el operador de polaridad que se representa en (9),  $\ddot{S}$  es el vector columna de aceleraciones articulares generalizadas representado por

$$\ddot{S} = \begin{bmatrix} \ddot{S}_1 & \ddot{S}_2 & \ddot{S}_3 & \ddot{S}_4 & \ddot{S}_5 & \ddot{S}_6 \end{bmatrix}^T \quad (15)$$

y  $C$  es el vector columna de aceleración complementaria que se define como en (16).

$$C = \begin{bmatrix} \{ \mathcal{A}_{1,}^2, \mathcal{S}_{r1}^3 \} \\ \{ \mathcal{A}_{2,}^2, \mathcal{S}_{r2}^3 \} \\ \{ \mathcal{A}_{3,}^2, \mathcal{S}_{r3}^3 \} \\ \{ \mathcal{A}_{4,}^2, \mathcal{S}_{r4}^3 \} \\ \{ \mathcal{A}_{5,}^2, \mathcal{S}_{r5}^3 \} \\ \{ \mathcal{A}_{6,}^2, \mathcal{S}_{r6}^3 \} \end{bmatrix} \quad (16)$$

La ecuación (14) es la denominada ecuación entrada-salida de aceleración, con la cual el análisis de aceleración directo se concluye, únicamente hay que despejar  ${}^B A_o^P$  si se conocen con las aceleraciones articulares generalizadas o despejar  $\ddot{S}$  si ya se cuenta con algún estado de aceleración, siempre cuidando que el manipulador no se encuentre en una posición de singularidad que evite que la matriz  $J$  sea invertible.

## V. RESULTADOS NUMÉRICOS

Para validar los modelos analíticos obtenidos, se realizaron simulaciones numéricas en el software ADAMS View. Se propusieron funciones polinomiales de tercer orden para las trayectorias de los seis grados de libertad de la plataforma móvil del manipulador, mediante la cinemática inversa se obtuvieron las trayectorias de los seis actuadores lineales correspondientes a las trayectorias deseadas de la plataforma móvil, y finalmente las trayectorias de los seis actuadores fueron introducidas tanto al análisis cinemático directo por teoría de tornillos como a los actuadores del prototipo virtual en ADAMS View en forma de funciones de movimiento. Se obtuvieron mediciones de las aceleraciones angulares y lineales de la plataforma móvil del manipulador, las cuales fueron comparadas con las obtenidas a través de la teoría de tornillos para la misma trayectoria de entrada.

Las trayectorias consideradas son de la forma

$$x(t) = a_0 t^3 + a_1 t^2 + a_2 t + a_3 \quad (17)$$

donde los coeficientes  $a_i$  son obtenidos a partir de las condiciones iniciales y finales de posición y de velocidad como en [10].

En la Figura 4 se presenta el modelo tridimensional del manipulador paralelo 6 UPUR en la que se muestran las restricciones de movimiento dadas por las diferentes juntas que constituyen sus cadenas cinemáticas, los marcos de referencia y las entradas de movimiento asignadas a las juntas prismáticas.

Se considera una trayectoria para la plataforma móvil con condiciones iniciales cero tanto en posiciones lineales como angulares, que inicien y terminen en reposo (velocidades iniciales y finales iguales a cero) y con las siguientes posiciones finales después de 5 segundos:

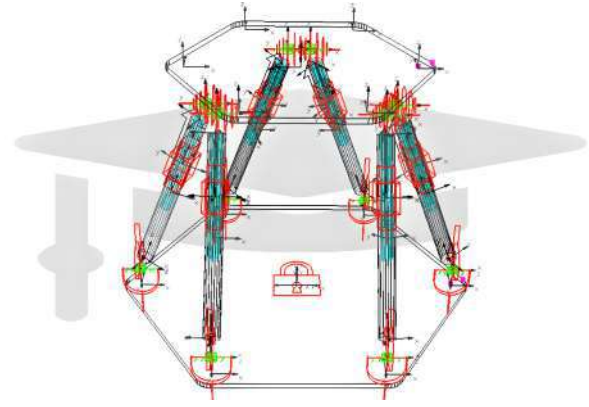


Fig. 4. Modelo del manipulador paralelo 6-UPUR en ADAMS View.

$$\begin{aligned} x &= 15 \text{ mm} \\ y &= 18 \text{ mm} \\ z &= 38 \text{ mm} \\ \alpha &= 10^\circ \\ \beta &= 5^\circ \\ \gamma &= 5^\circ \end{aligned} \quad (18)$$

En la Figura 5 se presenta la comparación de los resultados obtenidos para la aceleración angular de la plataforma móvil por teoría de tornillos (T. T.) y mediante el software de simulación multicuerpos ADAMS View (SMC) y en la Figura 6 se muestra el error absoluto que representa la diferencia entre los valores de aceleración angular obtenidos mediante la simulación en ADAMS View y los calculados mediante teoría de tornillos. Se puede apreciar que dichos errores están en el orden de  $10^{-5} \frac{rad}{s^2}$ .

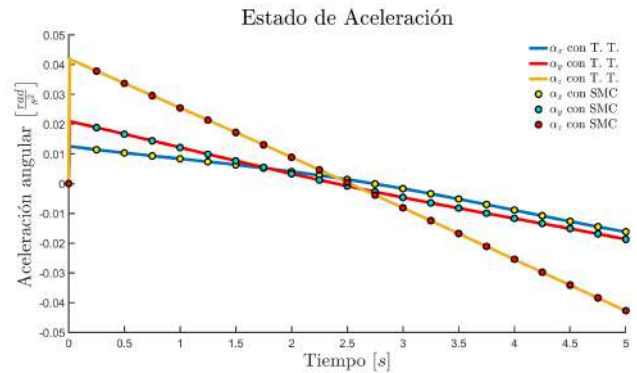


Fig. 5. Comparación de la aceleración angular de la plataforma móvil del manipulador paralelo 6-UPUR.

En las Figuras 7 y 8 se pueden apreciar los resultados correspondientes a las aceleraciones lineales del punto central de la plataforma móvil y sus correspondientes errores absolutos, respectivamente. El valor máximo el error absoluto de aceleración lineal se presenta en el movimiento a lo largo del eje  $y$  y tiene un valor de  $0.03 \frac{mm}{s^2}$ .

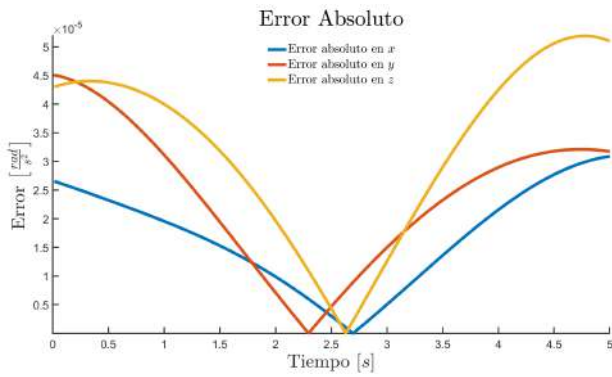


Fig. 6. Error absoluto de aceleración angular.

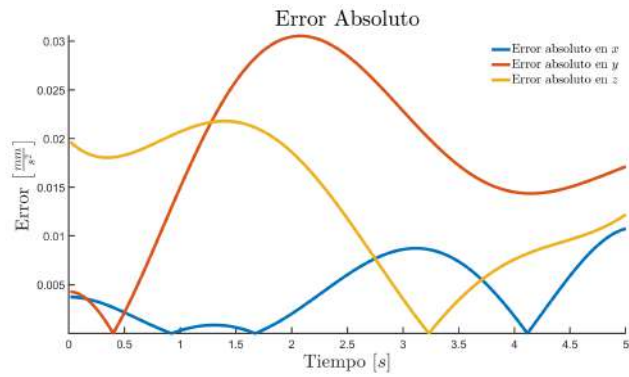


Fig. 8. Error absoluto de aceleración lineal.

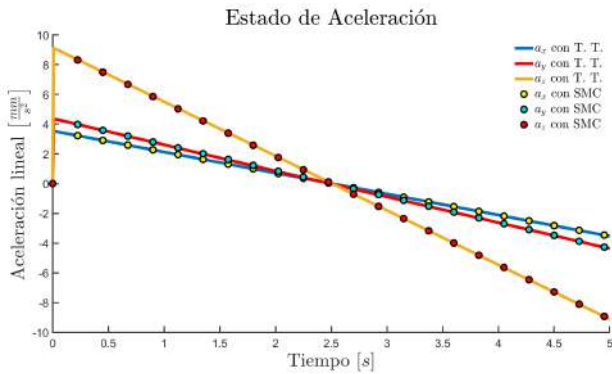


Fig. 7. Comparación de la aceleración lineal de la plataforma móvil del manipulador paralelo 6-UPUR.

## VI. CONCLUSIONES

En este trabajo se desarrolló el modelo cinemático de aceleración de un manipulador paralelo con configuración 6UPUR mediante la teoría de tornillos, para lo cual fue necesario completar previamente el análisis de velocidad. Se usó el concepto de tornillo recíproco para eliminar las aceleraciones de las juntas pasivas del análisis de forma metódica y obtener un modelo de aceleración entrada-salida, mostrando así las ventajas de usar teoría de tornillos para obtener los modelos cinemáticos de manipuladores paralelos sobre los métodos tradicionales que implican el uso de cálculo y álgebra y con los cuales resulta sumamente tedioso y complicado obtener representaciones de entrada-salida de velocidades y aceleraciones. Los modelos obtenidos con teoría de tornillos se compararon con resultados numéricos obtenidos mediante software especializado para el análisis de sistemas multi-cuerpos obteniendo errores absolutos del orden de 0.125% para aceleraciones angulares y de 0.71% para aceleraciones lineales, demostrando así la validez de los modelos cinemáticos obtenidos y la posibilidad de extender el análisis a variables como la derivada de la aceleración (pulso) y la segunda derivada de la aceleración (hiper pulso). Las representaciones entrada-salida de las variables cinemáticas son útiles además del análisis en sistemas robóticos, en el desarrollo de controladores de movimiento para este tipo de

sistemas.

## REFERENCIAS

- [1] H. D. Taghirad, *Parallel Robots: Mechanics and Control*, 1st ed. CRC Press: Boca Raton, FL, 2013.
- [2] J. P. Merlet, *Parallel Robots*, 2nd ed. Springer: Dordrecht, The Netherlands, 2006.
- [3] J. Gallardo-Alvarado, *Kinematic Analysis of Parallel Manipulators by Algebraic Screw Theory*, 1st ed. Springer: Switzerland, 2016.
- [4] J. Gallardo-Alvarado, A. L. Balmaceda-Santama, and E. Castillo-Castaneda, "An application of screw theory to the kinematic analysis of a Delta-type robot," *J. Mech. Sci. Technol.*, vol. 28(9), pp. 3785–3792, 2014.
- [5] J. Gallardo-Alvarado, and M. García-Murillo, "Análisis cinemático del manipulador paralelo 4-PRUR mediante la teoría de tornillos," *Rev. Iberoam. Autom. In.*, vol. 14(3), pp. 299–306, 2017.
- [6] J. Gallardo-Alvarado, "Kinematics of a hybrid manipulator by means of screw theory," *Multibody Syst. Dyn.*, vol. 14, pp. 345–366, 2005.
- [7] J. Gallardo-Alvarado, J. H. Tinajero-Campos, and A. Sánchez-Rodríguez, "Cinemática de un manipulador configurable por medio de la teoría de tornillos," *Rev. Iberoam. Autom. In.*, vol. 18(1), pp. 58–67, 2020.
- [8] H. J. Cortés-Ruiz, and M. Arias-Montiel, "Análisis cinemático de un robot paralelo 6-UPUR mediante teoría de tornillos," *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, vol. 12(Especial 2), pp. 108–114, 2024.
- [9] J. Zhao, B. Li, X. Yang, and H. Yu, "Geometrical method to determine the reciprocal screws and applications to parallel manipulators," *Robotica*, vol. 27(6), pp. 929–940, 2009.
- [10] R. N. Jazar, *Theory of Applied Robotics. Kinematics, Dynamics, and Control*, 2nd ed. Springer: New York, NY, 2010.

# CAPÍTULO 7

---

## ESTIMACIÓN

# Estimación de postura en un robot autobalanceable usando visión y redes neuronales supervisadas

Armando Saenz  
Posgrado e Investigación  
FICA-UJED  
Gómez Palacio, Durango  
jsaenz@ujed.mx

Issac Gandarilla  
Posgrado e Investigación  
FICA-UJED  
Gómez Palacio, Durango  
isaac.gandarilla@ujed.mx

Arturo Tadeo Espinoza Fraire  
Posgrado e Investigación  
FICA-UJED  
7 Gómez Palacio, Durango  
atespinoza@ujed.mx

Cristina Saenz  
Coordinación de ITC  
FICA-UJED  
Gómez Palacio, Durango  
cristina.saenz@ujed.mx

**Resumen**—El presente artículo describe una metodología para aplicar una red neuronal multicapa perceptron para medir la postura de un robot. Se describe el funcionamiento de aplicación llamada DataGenCV que es un producto de este trabajo. La aplicación tiene la finalidad de crear bases de datos mediante visión para redes neuronales supervisadas. Posteriormente, la base de datos obtenida con la aplicación será utilizada para entrenar una red neuronal. La red neuronal cuenta con una capa de entrada de cuatro datos correspondientes a las coordenadas de las dos características, dos capas ocultas con 64 y 32 neuronas respectivamente, y una capa de salida con dos neuronas, donde la primera salida corresponde al desplazamiento lineal y la segunda al ángulo del péndulo. Se utiliza la función de activación ReLU. Se realizan pruebas experimentales para contrastar la medición de la red neuronal con sensores propioceptivos de un robot.

**Index Terms**—Procesamiento de imágenes, redes neuronales, vision artificial, medico, MLP

## I. INTRODUCTION

El avance en el procesamiento de imágenes y las redes neuronales ha permitido desarrollar soluciones innovadoras en el ámbito de la robótica. Este artículo presenta un software desarrollado en Visual C# 2022 que facilita la aplicación de filtros de color y la selección de características específicas en un robot autobalanceable. La generación de una base de datos precisa de las coordenadas de estas características permite mejorar significativamente la calibración y el control del robot, optimizando su desplazamiento y el ángulo del péndulo. La red neuronal, entrenada con estos datos, proporciona un método eficaz para medir y ajustar estos parámetros en tiempo real.

Existen investigaciones dedicadas a explorar el uso de redes neuronales y técnicas de procesamiento de imágenes para la localización y control de robots móviles. Por ejemplo, en [1] desarrollaron una arquitectura mediante redes neuronales convolucionales en tiempo real que utiliza sensores de bajo costo y una cámara monocular para la relocalización precisa de robots en entornos sin GPS. Lee y Oh [2] propusieron un sistema basado en una red neuronal para estimar la posición y orientación del robot a partir de imágenes de una cámara, logrando alta precisión con cálculos simples. En [3] presenta una metodología para poder estimar con bastante precisión la postura de un robot móvil usando la calibración propuesta en [4], mejorando un control de posición. Estas investigaciones resaltan la efectividad de las redes neuronales y la visión

por computadora para mejorar la precisión y eficiencia en la estimación de poses y el control de robots.

Las redes neuronales de perceptrón multicapa [5] (MLP, por sus siglas en inglés) han demostrado ser efectivas para una variedad de tareas de clasificación y regresión. Presenta una gran capacidad para modelar relaciones altamente no lineales. Las aplicaciones que se han desarrollado en la robótica son la estimación de la postura, la planificación de movimientos y el control de sistemas. Algunas investigaciones recientes han demostrado que las MLP son excelentes opciones respecto a los métodos tradicionales en precisión y velocidad, sobre todo cuando se cuentan con grandes conjuntos de datos que pueden ser utilizados para su entrenamiento [6] [7]. Una sugerencia importante es utilizar la función de activación ReLU, en particular, ha mejorado significativamente el rendimiento de las redes MLP al mitigar el problema del desvanecimiento del gradiente, permitiendo entrenar redes más profundas y eficientes [8].

Este artículo se enmarca en este contexto, proponiendo una solución que combina técnicas de procesamiento de imágenes y redes neuronales para mejorar la precisión en la medición del desplazamiento y el ángulo de un robot carro-péndulo, eliminando la necesidad de calibración manual y optimizando su desempeño en tiempo real. La aportación principal de este desarrollo es la aplicación DataGenCV que es para generar una base de datos mediante información visual. La aplicación cuenta con una licencia MIT y es de código libre. Se puede descargar el código fuente desde un repositorio en Github con la dirección <https://github.com/ArmandoSaenz/DataGenCV> y en la misma dirección se puede descargar un instalador para correr directamente la aplicación.

El artículo se estructura de la siguiente manera: en la primera sección se presenta el planteamiento del problema, describiendo los desafíos actuales en la calibración de cámaras para robots móviles. La segunda sección detalla la aplicación desarrollada en Visual C# 2022, llamada DataGenCV, para generar la base de datos, explicando su funcionamiento y características principales. En la tercera sección, se discute el diseño y entrenamiento de la red neuronal, incluyendo la arquitectura utilizada y los parámetros de entrenamiento. La cuarta sección presenta los resultados obtenidos, acompañados de un análisis y discusión sobre la efectividad del enfoque

propuesto. Finalmente, la última sección concluye con un resumen de los hallazgos principales y una discusión sobre posibles futuras mejoras y aplicaciones del sistema.

## II. PLANTEAMIENTO DEL PROBLEMA

La calibración precisa de cámaras es un desafío crítico en la robótica móvil, especialmente en aplicaciones que requieren alta precisión en la medición del desplazamiento y el ángulo de componentes móviles, como es el caso de los robots carro-péndulo. La calibración manual de las cámaras es un proceso laborioso y propenso a errores, lo que puede afectar negativamente el rendimiento del sistema de control del robot. Además, las variaciones en las condiciones del entorno y el desgaste del hardware pueden provocar la necesidad de recalibraciones frecuentes, incrementando el tiempo de inactividad y los costos operativos.

## III. ROBOT AUTOBALANCEABLE

En particular, los robots autobalanceable enfrentan desafíos adicionales debido a la necesidad de medir tanto el desplazamiento lineal como el ángulo del péndulo con alta precisión. La falta de una calibración adecuada puede llevar a errores significativos en estas mediciones, comprometiendo la estabilidad y el control del robot. Este problema se agrava cuando se utiliza una cámara monocular, que, aunque es más económica y fácil de integrar que las cámaras estéreo, presenta mayores desafíos en términos de precisión y exactitud en la estimación de coordenadas en el mundo real.

Las consideraciones del robot se muestran en la Figura 11. Se tiene un marco del mundo llamado  $\Sigma_W$  con los ejes  $x_W$  y  $z_W$ . El avance del robot se hace en dirección del eje  $x_W$  y el ángulo del péndulo llamado  $\alpha$  será medido desde el eje vertical (i.e.  $y_W$ ). La postura del robot será representada por  $\xi$  y es definida de la siguiente forma:

$$\xi = [x \quad \alpha]^T. \quad (1)$$

El ángulo  $\varphi$  representa el giro de la rueda y  $r$  es el radio de la rueda. Por lo tanto, si se quiere conocer la posición del robot respecto a  $\Sigma_W$  se puede hacer mediante el ángulo actual de la rueda y la condición inicial por lo que  $x$  podría ser obtenida mediante la siguiente expresión:

$$x = r\varphi + x_0 \quad (2)$$

donde  $x_0$  es la condición inicial (i.e. distancia inicial) del robot.

El setup utilizado en este estudio incluye una cámara monocular que no está alineada con el plano de movimiento del robot, lo cual añade complejidad al proceso de calibración y estimación. Además, la cámara presenta distorsión en los extremos de la imagen, lo que puede introducir errores adicionales en la detección de las características. El carro-péndulo cuenta con dos marcadores visibles que se utilizan para ubicar al robot en el espacio. Estos marcadores son esenciales para el proceso de correlación de coordenadas y permiten generar una base de datos precisa que relaciona las posiciones de los marcadores con el desplazamiento y el ángulo del péndulo.

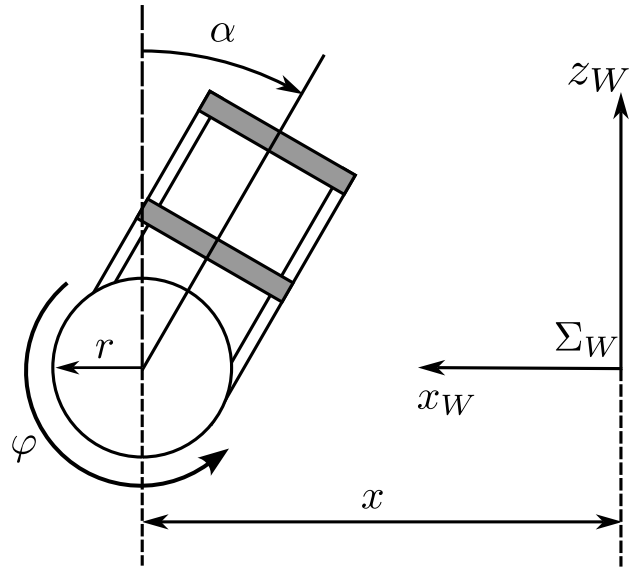


Figura 1. Aplicación DataGenCV

### III-A. Modelo de la cámara

En el mercado es posible encontrar distintos tipos de cámaras digitales cuyas características pueden variar desde el tipo de lente que se utiliza hasta el sensor fotosensible. La selección del lente es muy importante ya que dependiendo de la calidad del lente pueden existir o no deformaciones en la imagen. Existen básicamente cuatro modelos matemáticos para representar una cámara los cuales son: pinhole [9], lente delgada [10], lente ancha y omnidireccional [11]. En robótica se utiliza comúnmente el modelo pinhole que los lentes son de poca distancia focal a comparación con los de lente delgada y lente ancha. Dependiendo de la aplicación y la distancia entre el objeto observado y la cámara o que tan grande sea el espacio que se desea observar es la selección de la lente.

El modelo pinhole es una formulación relativamente sen-

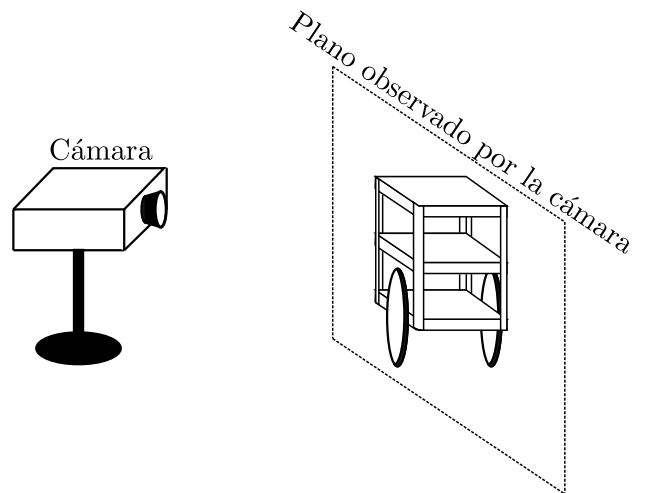


Figura 2. Ubicación de la cámara respecto al movimiento del robot

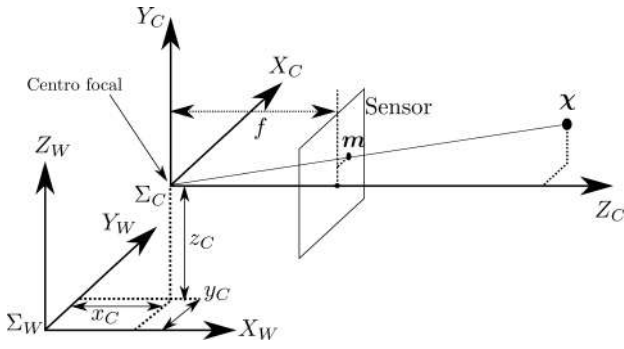


Figura 3. Punto proyectado sobre el sensor de la cámara

Parámetros	Descripción	Unidad
$f$	Distancia focal	metros
$k_u$	Constante de conversión de metros a píxeles en el eje $u$	$\frac{\text{metros}}{\text{píxeles}}$
$k_v$	Constante de conversión de metros a píxeles en el eje $v$	$\frac{\text{metros}}{\text{píxeles}}$
$\phi$	Ángulos existente entre los ejes $u$ y $v$	$\frac{\text{metros}}{\text{píxeles}}$
$u_0$	Distancia desde el centro focal hasta el origen del marco $\Sigma_f$ a lo largo del eje $u$	$\frac{\text{metros}}{\text{píxeles}}$
$v_0$	Distancia desde el centro focal hasta el origen del marco $\Sigma_f$ a lo largo del eje $v$	$\frac{\text{metros}}{\text{píxeles}}$

Tabla I

TABLA DE PARÁMETROS INTRÍNSECOS

cilla de como el lente de una cámara concentra los rayos de luz haciendolos pasar por un pequeño orificio para que sean proyectados sobre el sensor fotosensible de la cámara. Este modelo considera parámetros extrínsecos y parámetros intrínsecos. Los extrínsecos son la posición y orientación de la cámara respecto al marco del mundo. Los parámetros intrínsecos son los datos internos de la cámara correspondiente a la distancia focal, tamaño y resolución del sensor fotosensible, así como también la orientación del sensor respecto al lente. La disposición de la cámara a utilizar es la mostrada en la Figura 2.

$$\chi^C = \begin{bmatrix} y^C \\ z^C \\ x^C \end{bmatrix} = R_W^C \chi - t^C, \quad (3)$$

con

$$R_W^C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad t^C = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix}.$$

III-A1. *Espacio de la imagen:* Para convertir un punto en coordenadas del mundo (i.e.  $\chi^C$ ) a coordenadas en el espacio de la imagen es necesario conocer los parámetros

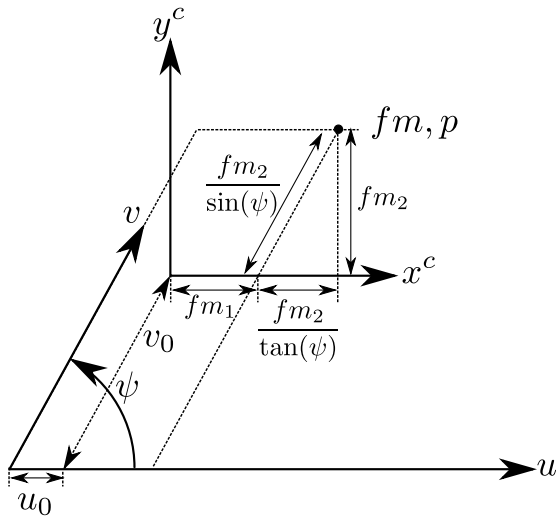


Figura 4. Parámetros intrínsecos de la cámara

intrínsecos de la cámara. Estos parámetros varían en cada cámara, incluso aunque sean del mismo modelo. De acuerdo a lo reportando en [12] pueden variar en el tiempo, aunque de una manera muy lenta por lo que pueden considerarse constantes en un periodo de tiempo relativamente corto. Los parámetros pueden observarse en las Figuras 3 y 4. En la Tabla I se menciona cuales son las variables que se utilizan para definirlos y una descripción de cada uno. La conversión de un punto en coordenadas del mundo a coordenadas en el espacio de la imagen se realiza de la siguiente manera. Sea  $m \in \mathbb{R}^3$  un vector tal que

$$m = \begin{bmatrix} m_1 \\ m_2 \\ 1 \end{bmatrix} = \frac{1}{z^C} \chi^C = \frac{1}{z^C} \begin{bmatrix} x^C \\ y^C \\ z^C \end{bmatrix}. \quad (4)$$

El vector  $m$  son las coordenadas sobre un plano donde  $z^C$  es la unidad. Para encontrar la proyección sobre el sensor de la cámara y pasarlo al espacio de la imagen se realiza la siguiente transformación.

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K m \quad (5)$$

$$K = \begin{bmatrix} f k_u & \frac{f k_u}{\tan(\phi)} & u_0 \\ 0 & \frac{f k_v}{\sin(\phi)} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Por lo tanto, existe una necesidad de desarrollar métodos automáticos y robustos para la calibración de cámaras monoculares, que sean capaces de generar datos precisos y consistentes para la medición del desplazamiento y el ángulo del péndulo. Este artículo propone una solución para no requerir del conocimiento de los parámetros intrínsecos y extrínsecos de la cámara.

#### IV. APLICACIÓN DATAGENCv

El software DataGenCV, es una aplicación desarrollada para este trabajo. Esta programado en Visual C# 2022, es una herramienta poderosa y versátil diseñada para generar bases de datos precisas mediante la aplicación de filtros de color y la selección de características específicas en las imágenes

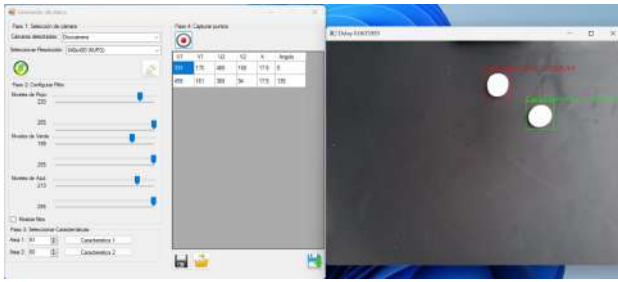


Figura 5. Aplicación DataGenCV

capturadas por la cámara del robot carro-péndulo. Este software se caracteriza por su facilidad de uso y su capacidad para automatizar la calibración de cámaras, optimizando la medición del desplazamiento y el ángulo del péndulo.

#### IV-A. Uso de la aplicación

Al iniciar DataGenCV (véase Figura 5), el software detecta automáticamente las cámaras compatibles conectadas al sistema y muestra sus nombres en una lista desplegable. El usuario debe seleccionar la cámara deseada, tras lo cual se cargan las posibles resoluciones y opciones de compresión de imágenes disponibles para esa cámara. Una vez seleccionada la resolución y la compresión adecuada, se debe hacer clic en el botón de conectar para que la aplicación comience a capturar imágenes, abriendo una ventana con el video en tiempo real.

En la interfaz principal de DataGenCV, se encuentran seis barras deslizantes (trackbars) que permiten ajustar los valores mínimos y máximos del componente rojo del filtro de color. Además, hay una casilla de verificación (checkbox) que permite alternar la vista entre la imagen con el filtro aplicado y la imagen original, facilitando así el ajuste preciso de los valores del filtro de color.

Una vez configurado el filtro, el usuario puede identificar dos características específicas en la imagen. Para ello, se utilizan dos botones etiquetados como *Característica 1* y *Característica 2*. Al presionar uno de estos botones, el usuario

debe hacer clic en la imagen para indicar la ubicación de la característica deseada. Junto a estos botones, hay dos cuadros de texto donde se puede especificar el tamaño de la ventana de recorte, centrada en el punto donde se hizo clic en la imagen.

Después de seleccionar ambas características, aparecerán en la imagen dos recuadros que marcan visualmente las características seleccionadas. El usuario puede entonces presionar la tecla Enter (asegurándose de que la ventana de la imagen tenga el foco) o hacer clic en el botón con el símbolo de grabar (un círculo rojo). Al hacerlo, el video se detendrá en una imagen fija y se solicitará al usuario que ingrese la posición e inclinación actuales del robot.

Este proceso puede repetirse tantas veces como sea necesario, y cada registro se mostrará en una tabla a la derecha de la interfaz. Una vez alcanzado el número deseado de registros, el usuario puede guardar los datos utilizando dos botones de guardado ubicados en la parte inferior de la interfaz. El botón de la izquierda guarda los valores del filtro y los tamaños de la ventana, mientras que el botón de la derecha guarda los registros mostrados en la tabla. Ambos archivos se guardan en formato JSON, un estándar que puede ser leído por cualquier lenguaje de programación. El instalador de la

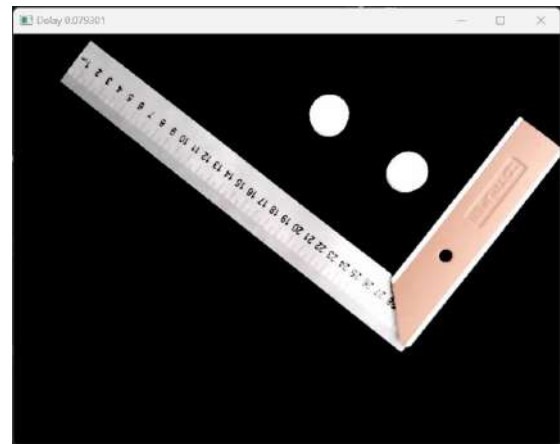


Figura 7. Imagen con Filtro



Figura 6. Imagen sin filtro



Figura 8. Identificación de las características

aplicación y código fuente del mismo están disponibles en la liga <https://github.com/ArmandoSaenz/DataGenCV>

## V. RED NEURONAL

La red neuronal utilizada en este estudio ha sido desarrollada en Python y está diseñada para interpretar las coordenadas de las dos características seleccionadas y proporcionar estimaciones precisas del desplazamiento lineal y el ángulo del péndulo del robot. A continuación, se detalla su arquitectura, el proceso de entrenamiento y su aplicación.

### V-A. Arquitectura de la Red Neuronal

La red neuronal empleada es un perceptrón multicapa (MLP) que es mostrada en la Figura 10 y cuya estructura se describe en los siguientes puntos:

- Capa de entrada: Consta de 4 neuronas, correspondientes a las coordenadas  $(u_i, v_i)$  con  $i = 1, 2$  de las dos características seleccionadas. El vector entrada para esta capa será representado por el vector  $m$  definido como:

$$q = [u_1 \quad v_1 \quad u_2 \quad v_2]^T \quad (7)$$

- Primera capa oculta: Contiene 64 neuronas, utiliza la función de activación ReLU (Rectified Linear Unit) para introducir no linealidad y mejorar la capacidad de la red para aprender relaciones complejas en los datos. La función ReLU se define a continuación:

$$\text{ReLU}(\beta) = \begin{cases} 0 & \forall \beta \leq 0 \\ \beta & \forall \beta > 0 \end{cases} \quad (8)$$

- Segunda capa oculta: Contiene 32 neuronas, también utilizando la función de activación ReLU.
- Capa de salida: Tiene 2 neuronas, donde la primera neurona proporciona la estimación del desplazamiento lineal del robot y la segunda neurona estima el ángulo del péndulo. La función de activación en esta capa es lineal, adecuada para tareas de regresión.

La expresión que representaría la red neuronal en forma compacta es:

$$\begin{bmatrix} x \\ \alpha \end{bmatrix} = W_3 \text{ReLU}(W_2 \text{ReLU}(W_1 q)) \quad (9)$$

donde  $W_1 \in \mathbb{R}^{64 \times 4}$  son los pesos de interconexión entre la capa de entrada a la primera capa oculta,  $W_2 \in \mathbb{R}^{32 \times 64}$  son

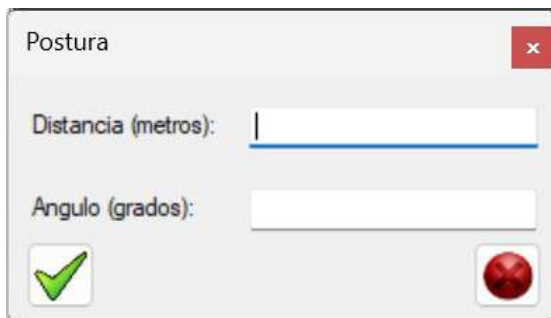


Figura 9. Ventana para capturar la postura

los pesos de la primera capa oculta a la segunda capa oculta y  $W_3 \in \mathbb{R}^{2 \times 32}$  son los pesos de la segunda capa oculta a la capa de salida.

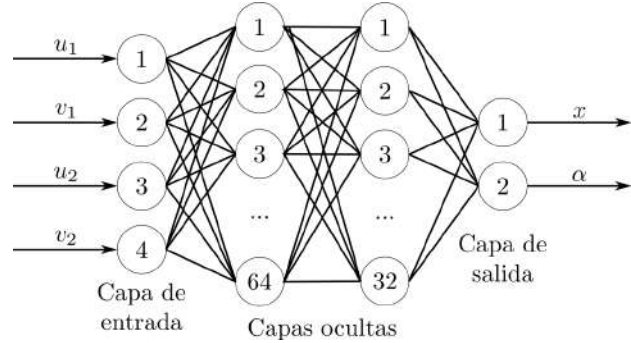


Figura 10. Red Neuronal MLP

### V-B. Proceso de Entrenamiento

El entrenamiento de la red neuronal se realiza utilizando la base de datos generada con DataGenCV. Esta base de datos contiene múltiples registros de las coordenadas de las características y las correspondientes mediciones de desplazamiento y ángulo. El proceso de entrenamiento sigue los siguientes pasos:

- Preprocesamiento de datos: Los datos se normalizan para asegurar que todas las entradas tengan una escala similar, lo que ayuda a acelerar el proceso de entrenamiento y a mejorar la estabilidad de la red. División de los datos: La base de datos se divide en conjuntos de entrenamiento y validación en una proporción de 80/20. La cantidad de registros en la base de datos es de 30.
- Entrenamiento: La red se entrena utilizando un algoritmo de optimización Adam [13], que ajusta los pesos de las conexiones neuronales para minimizar el error entre las salidas predichas y los valores reales. Se utiliza el error cuadrático medio (MSE) como función de pérdida. La cantidad de épocas indicadas es de 600.
- Validación: Durante el entrenamiento, el rendimiento de la red se evalúa periódicamente utilizando el conjunto de validación para evitar el sobreajuste.

### V-C. Aplicación

Una vez entrenada, la red neuronal puede utilizarse en tiempo real para estimar el desplazamiento y el ángulo del péndulo del robot. Las coordenadas de las características seleccionadas se alimentan a la red, que produce las estimaciones correspondientes. Esto permite un control preciso del robot sin necesidad de recalibraciones manuales constantes.

## VI. RESULTADOS

En esta sección se presentan los resultados obtenidos mediante el uso de la aplicación DataGenCV y la red neuronal entrenada. Los resultados se analizan en términos de precisión en la estimación del desplazamiento lineal y el ángulo del



Figura 11. Robot autobalancable utilizado

péndulo del robot, así como la eficacia del enfoque propuesto en la reducción de errores de calibración.

El robot utilizado se muestra en la Figura 11 esta programado con un control por redes neuronales que puede ser consultado en [14]. El objetivo de dicho control es mantener el péndulo en posición horizontal sin ninguna posición en específico. Al robot se le añadieron dos cubos de color rojo de 1 cm de lado y un peso de 2g para no afectar la dinámica del robot. El peso del robot es de 1.54 kg.

El tiempo total de experimentación fue de 30 segundos. En las Figuras 12 y 13 se muestra la posición y orientación del robot medidas con encoders e IMU, respectivamente. El tiempo de muestreo para estas mediciones fue de 1 milisegundo. En contraste, las Figuras 14 y 15 muestran la posición y orientación medidas con la cámara y la red neuronal. La cámara utilizada presenta una velocidad de captura de 30 cuadros por segundo, lo cual se traduce en un tiempo de muestreo de aproximadamente 33 milisegundos.

La diferencia entre las Figuras 12 y 14 puede apreciarse como pequeña; ambas muestran curvas con formas similares, y los picos observados en los segundos 7, 12 y 23 se presentan en ambas gráficas. La mayor diferencia notable es un desplazamiento de 20 cm entre ellas, lo cual se debe a que el origen de la cámara es el mismo en cualquier experimento, mientras que el origen del robot se toma en la ubicación donde se enciende.

Por otro lado, la diferencia entre las Figuras 13 y 15 es mucho más notable, principalmente porque la medición de la IMU presenta un ruido de alta frecuencia y amplitud considerable, mientras que en la medición con la cámara no se observa este ruido. Sin embargo, en la zona donde se presenta menos ruido, que es de los 10 a los 20 segundos aproximadamente, las dos gráficas son semejantes, lo que permite deducir que, al menos en movimientos lentos, las mediciones de ambos dispositivos coinciden.

## VII. CONCLUSIONES

El desarrollo del software DataGenCV y la implementación de una red neuronal en Python han demostrado ser una solución efectiva para la medición del desplazamiento y el

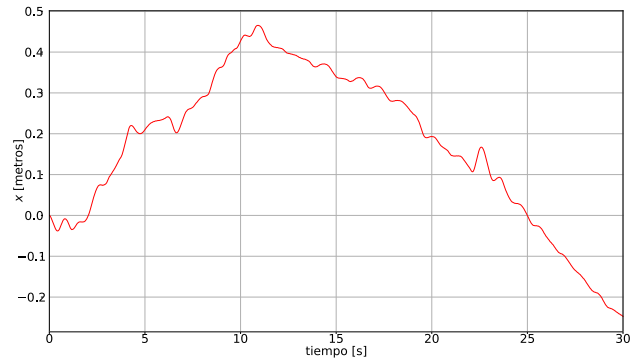


Figura 12. Desplazamiento medido por los encoders del robot

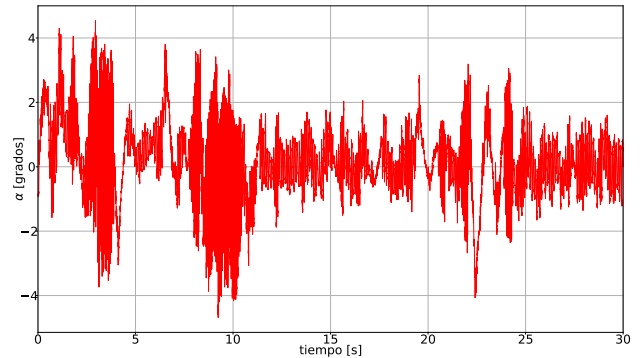


Figura 13. Ángulo medido desde la central inercial

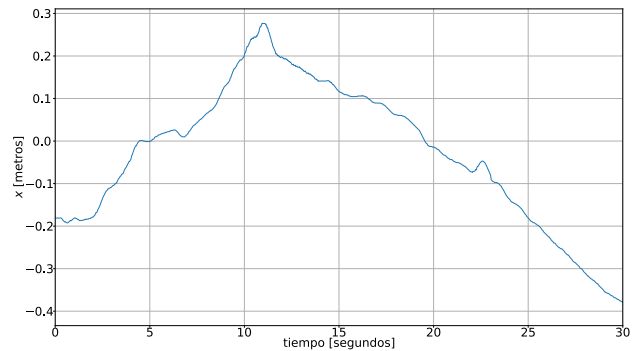


Figura 14. Ángulo del péndulo medido desde la cámara

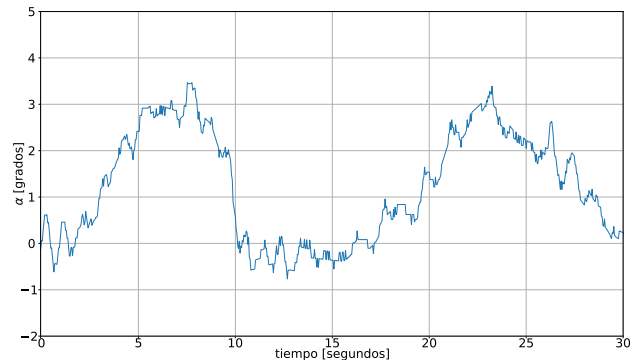


Figura 15. Ángulo del péndulo medido desde la cámara

ángulo del robot carro-péndulo. Los resultados obtenidos a lo largo de las pruebas y experimentaciones validan la eficacia del enfoque propuesto en varios aspectos clave:

La utilización de DataGenCV permite automatizar el proceso de calibración de la cámara, eliminando la necesidad de calibraciones manuales frecuentes. Esto no solo ahorra tiempo, sino que también reduce el margen de error humano, mejorando la consistencia y fiabilidad del sistema.

La red neuronal entrenada con la base de datos generada por DataGen ha mostrado una alta precisión en la estimación del desplazamiento y el ángulo del péndulo. Las comparaciones con los valores medidos por encoders e IMU revelan una correlación significativa, destacando la capacidad del sistema para replicar las mediciones reales con un mínimo error.

En comparación con las mediciones de la IMU, que presentan un ruido de alta frecuencia y amplitud considerable, las mediciones obtenidas mediante la cámara y la red neuronal son mucho más limpias y estables. Esto es especialmente evidente en los intervalos de tiempo con movimientos lentos, donde las gráficas de ambas mediciones coinciden estrechamente.

## REFERENCIAS

- [1] H. Sinha, J. Patrikar, E. G. Dhekane, G. Pandey, and M. Kothari, "Convolutional neural network based sensors for mobile robot relocation," in *International Conference on Methods, Models in Automation and Robotics (MMAR)*, 2018, pp. 774–779.
- [2] J.-H. Lee and S.-Y. Oh, "An absolute robot pose estimation system based on a ceiling camera image using a neural network," in *2006 IEEE International Conference on Systems, Man and Cybernetics*, 2006, pp. 3839–3843.
- [3] R. Nurlaili, I. A. Sulistijono, and A. Risnumawan, "Mobile robot position control using computer vision," in *International Electronics Symposium (IES)*, 2019, pp. 382–386.
- [4] Z. Zhang, "A flexible new technique for camera calibration," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 22, 2000.
- [5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 10 1986.
- [6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84–90, 6 2017.
- [8] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines vinod nair," in *Proceedings of ICML*, vol. 27, 7 2010, pp. 807–814.
- [9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd ed. Cambridge University Press, 2004.
- [10] C. Ricolfe-Viala, A.-J. Sanchez-Salmeron, and A. Valera, "Efficient lens distortion correction for decoupling in calibration of wide angle lens cameras," *IEEE Sensors Journal*, vol. 13, pp. 854–863, 2 2013.
- [11] S. Urban, J. Leitloff, and S. Hinz, "Improved wide-angle, fisheye and omnidirectional camera calibration," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 108, pp. 72–79, 10 2015.
- [12] E. Malis, "Visual servoing invariant to changes in camera-intrinsic parameters," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 72–81, 2004.
- [13] I. Jais, A. R. Ismail, and S. Q. Nisa, "Adam optimization algorithm for wide and deep neural network," *Knowledge Engineering and Data Science*, vol. 2, pp. 41–46, 2019. [Online]. Available: <https://doi.org/10.17977/um017v2i12019p41-462019KnowledgeEngineeringandDataScience>  
W:<http://journal2.um.ac.id/index.php/keds>|E:keds.journal@um.ac.idThisisanopenaccessarticleundertheCCBY-SAlicense
- [14] I. Gandarilla, J. Montoya-Cháirez, V. Santibáñez, C. Aguilar-Avelar, and J. Moreno-Valenzuela, "Trajectory tracking control of a self-balancing robot via adaptive neural networks," *Engineering Science and Technology, an International Journal*, vol. 35, 11 2022.

# Estimación y Compensación de Perturbaciones en una Formación en Cadena Sobre Ejes Móviles

J. Eduardo Ruiz Ruiz

Centro de Investigación y de Estudios Avanzados del IPN  
Sección de Mecatrónica  
Ciudad de México, México  
Email: jeduardo.ruiz@cinvestav.mx

Julio A. Báez Hernández

Centro de Investigación y de Estudios Avanzados del IPN  
Sección de Mecatrónica  
Ciudad de México, México  
Email:julio.baez@cinvestav.mx

Martin Velasco Villa

Centro de Investigación y de Estudios Avanzados del IPN  
Sección de Mecatrónica  
Ciudad de México, México  
Email:velasco@cinvestav.mx

Alejandro Rodriguez Angeles

Centro de Investigación y de Estudios Avanzados del IPN  
Sección de Mecatrónica  
Ciudad de México, México  
Email: aangeles@cinvestav.mx

**Resumen**—En este trabajo se presenta el control sobre ejes móviles de una formación en cadena sujeta a perturbaciones de deslizamiento y patinado en algún miembro de la formación. Se considera una formación basada en tiempos de separación entre dos robots consecutivos en la formación, esto es, se desea que el robot  $R_{i+1}$  siga la trayectoria retrasada  $\tau$  unidades de tiempo del robot  $R_i$ . Se muestra como mediante mediciones de distancia y orientación tomadas a partir de robot  $R_{i+1}$  es posible estimar las posibles perturbaciones externas desconocidas actuando sobre el robot  $R_i$ . Las perturbaciones estimadas son usadas para recalcular la trayectoria que el robot  $R_{i+1}$  tiene que seguir y de esta forma compensar las perturbaciones afectando la formación.

**Index Terms**—Robótica móvil, formación en cadena, líder seguidor, perturbaciones de patinado y deslizamiento

## I. INTRODUCCIÓN

Actualmente, la navegación autónoma ha experimentado un notable avance gracias a sistemas de formación más sofisticados y versátiles, estos sistemas permiten la coordinación y cooperación entre múltiples unidades. Gracias a esto es posible dar una solución a diversas problemáticas modernas que se presentan en las grandes urbes, ya sea en el ámbito relacionado a los transportes civiles o aquel relacionado a la industria. [1], [2], [3], [4].

Indudablemente, el control de la separación entre vehículos es un aspecto crucial en la navegación autónoma y la cooperación entre unidades que conforman la formación. Por tal motivo diversos autores abordan el tema de control mediante distintos enfoques entre los cuales se pueden mencionar, la ley de control para la separación inter-vehicular constante [5], el Control Crucero Adaptativo (ACC) el cual utiliza sensores, como radares y cámaras, para detectar la presencia y velocidad de los vehículos precedentes [6] o una técnica más compleja como lo es el Cooperative Adaptive Cruise Control (CACC)

El primer autor agradece a Conahcyt por el apoyo recibido. No. CVU: 1240278.

la cual agrega comunicación directa entre vehículos para una cooperación más avanzada. [7], [8], [9].

Otra manera de abordar el tema es mediante, el análisis del modelo cinemático para un vehículo, donde se han desarrollado esquemas para robots diferenciales basados en tiempos de separación constante o variante [10], [11]. Aunque una de las principales limitantes de dichos trabajos es que todo el desarrollo se realiza mediante los ejes inerciales, lo cual, limita la capacidad de solución unicamente a entornos controlados de laboratorio. Siendo una alternativa lo visto en [14], donde se presenta una estrategia de formación en cadena para un conjunto de  $n$  robots móviles de tracción diferencial. La estrategia se basa en marcos móviles ubicados en los vehículos, de modo que la posición de un robot  $R_i$  se determina con respecto al marco ubicado en el robot  $R_{i+1}$ . A su vez para corregir las limitantes relacionadas a las coordenadas globales se implementan sistemas de visión artificial los cuales permiten llevar este problema fuera de un laboratorio [12]. Sin embargo, las estrategias anteriormente mencionadas siguen contando con un problema, el cual es la falta de compensación ante perturbaciones que pudiesen presentarse en uno de los robots.

Por ende este trabajo se enfoca en la obtención de un modelo cinemático para un pelotón en formación Líder-Seguidor, considerando las perturbaciones de deslizamiento o patinado que pueda sufrir el robot líder y como dicha perturbación puede ser compensada la perturbación para realizar una corrección eficiente de la trayectoria del robot seguidor.

La estructura del documento es la siguiente, en la sección 2, se presenta el modelo cinemático del robot móvil en un marco de referencia global. La sección 3 menciona el esquema de formación a seguir así como la dinámica resultante entre el robot líder y el seguidor. En la sección 4 se aborda el análisis de las perturbaciones en el movimiento que el robot líder llegase a experimentar. La sección 5 muestra una estimación de las perturbaciones del robot líder. En la sección 6 se implementa

la estimación de las perturbaciones al esquema Líder-Seguidor planteado. En la sección 7 se observan los resultados obtenidos en simulaciones numéricas. Y finalmente en la sección 8, se presentan de manera breve algunas conclusiones.

## II. ESQUEMA DE FORMACION LÍDER-SEGUIDOR

Considere el modelo cinemático de un robot móvil diferencial como el mostrado en la figura 1, obtenido siguiendo la metodología presentada en [15], [16],

$$\begin{aligned}\dot{x}_i &= v_i \cos \theta_i \\ \dot{y}_i &= v_i \sin \theta_i \\ \dot{\theta}_i &= \omega_i\end{aligned}\quad (1)$$

donde  $x_i$  e  $y_i$  corresponden a la posición cartesiana del “ $i$  – esimo robot” en el plano X-Y,  $\theta_i$  corresponde a la orientación del robot con respecto al eje X y  $v_i, \omega_i$  corresponden a las velocidades lineales y angulares que representan las señales de control. Se asume que el robot se desplaza en una superficie plana libre de patinados o deslizamientos, por lo que se considera un contacto puntual de las ruedas y la superficie de trabajo.

Bajo estas consideraciones el robot móvil, dado por (1) satisface la restricción no-holónoma,

$$\dot{x}_i \sin \theta_i - \dot{y}_i \cos \theta_i = 0. \quad (2)$$

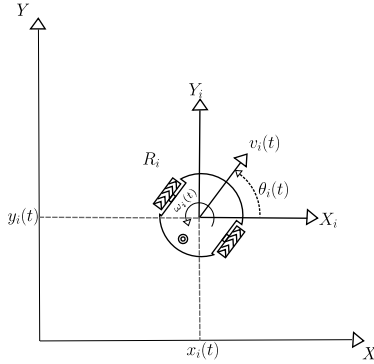


Figura 1. Diagrama del robot móvil diferencial.

En este trabajo se analiza el problema de formación en cadena de un conjunto de  $n$  robots diferenciales. Numerando el conjunto de robots de 1 a  $n$ , el análisis que se llevara a cabo, involucra un par de robot consecutivos en la formación, esto es, un robot líder  $R_i$  y un robot seguidor  $R_{i+1}$ . Para este par de robots se plantea a continuación el objetivo principal del trabajo, el cual es realizar una estimación y su posterior compensación de las perturbaciones que puedan presentarse en algún miembro de la formación.

## III. PROBLEMA DE FORMACIÓN EN TRAYECTORIA RETARDADA

Comenzando el análisis de la formación en cadena, donde para cada par de robots consecutivos se tiene la formación Líder-Seguidor considerando que el robot líder  $R_i$  describe

una trayectoria en el plano X-Y generada por velocidades lineal y angular acotadas en la forma,

$$\sup_{t \geq 0} v_i(t) = \bar{v}_i \quad \sup_{t \geq 0} \omega_i(t) = \bar{\omega}_i. \quad (3)$$

Bajo estas condiciones, para un par de robots, se desea que el robot seguidor  $R_{i+1}$ , siga la trayectoria del robot líder  $R_i$  retrasada  $\tau$  unidades de tiempo. Esta formación basada en tiempo de separación se describe en la Figura 2.

*Observación 1:* Notese que si la estrategia descrita se aplica a cada par de robots consecutivos en la formación, se obtendrá la formación de toda la cadena de robots, donde el robot seguidor  $R_{i+1}$  convergerá a la trayectoria retrasada de su robot líder  $R_i$ .

Para abordar el problema se toma en cuenta lo siguiente:

*Suposición 1:* En el plano  $X - Y$ , no se cuenta con la posición, ni orientación de los robots involucrados, y se usaran los dispositivos disponibles en el robot como lo son una cámara o lidar, capaces de medir la distancia entre los dos robots.

Nótese que bajo estas consideraciones el problema de formación puede resolverse en ambientes no sujetos a condiciones de laboratorio, sino que puede abordarse de igual manera en exteriores.

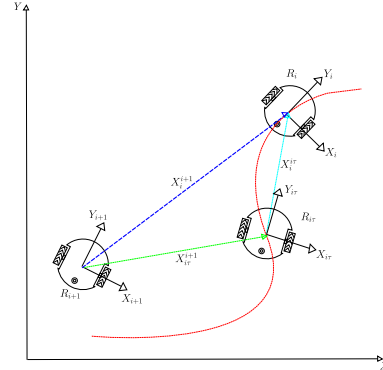


Figura 2. Formación líder-seguidor basada en tiempo de retardo.

La dinámica relativa en el esquema de formación Líder-Seguidor para dos robots consecutivos descrito anteriormente se muestra en el plano cartesiano en la Figura 3, donde  $l_i^{i+1}(t)$  y  $\alpha_i^{i+1}(t)$  representa la posición y orientación relativa del robot líder con respecto al robot seguidor medidos con respecto a los ejes móviles  $X_{i+1} - Y_{i+1}$  montados sobre el robot seguidor.

A partir de la Figura 3, es posible ver que,

$$\begin{aligned}x_i^{i+1} &= (x_i - x_{i+1}) \cos \theta_{i+1} + (y_i - y_{i+1}) \sin \theta_{i+1} \\ y_i^{i+1} &= -(x_i - x_{i+1}) \sin \theta_{i+1} + (y_i - y_{i+1}) \cos \theta_{i+1}\end{aligned}\quad (4)$$

donde  $x_i^{i+1}$  y  $y_i^{i+1}$  corresponden a las componentes del vector  $l_i^{i+1}$  con respecto a los ejes móviles  $X_{i+1} - Y_{i+1}$ .

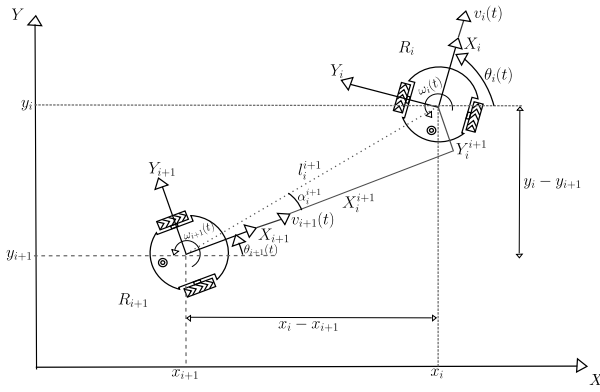


Figura 3. Formación Líder-Seguidor en el plano cartesiano.

La dinámica de las distancias relativas  $x_i^{i+1}, y_i^{i+1}$  puede obtenerse al considerar su derivada con respecto al tiempo, esto es,

$$\begin{aligned} \dot{x}_i^{i+1} &= -(x_i - x_{i+1})\dot{\theta}_{i+1} \sin \theta_{i+1} + (\dot{x}_i - \dot{x}_{i+1}) \cos \theta_{i+1} \\ &\quad + (y_i - y_{i+1})\dot{\theta}_{i+1} \cos \theta_{i+1} + (\dot{y}_i - \dot{y}_{i+1}) \sin \theta_{i+1} \\ &= -(x_i - x_{i+1})\omega_{i+1} \sin \theta_{i+1} \\ &\quad + (v_i \cos \theta_i - v_{i+1} \cos \theta_{i+1}) \cos \theta_{i+1} \\ &\quad + (y_i - y_{i+1})\omega_{i+1} \cos \theta_{i+1} \\ &\quad + (v_i \sin \theta_i - v_{i+1} \sin \theta_{i+1}) \sin \theta_{i+1} \\ &= -(x_i - x_{i+1}) \sin \theta_{i+1} + (y_i - y_{i+1}) \cos \theta_{i+1} \omega_{i+1} \\ &\quad + v_i (\cos \theta_i \cos \theta_{i+1} + \sin \theta_i \sin \theta_{i+1}) \\ &\quad - v_{i+1} (\cos^2 \theta_{i+1} + \sin^2 \theta_{i+1}) \\ &= y_i^{i+1} \omega_{i+1} + v_i \cos(\theta_i - \theta_{i+1}) - v_{i+1}. \end{aligned} \quad (5)$$

$$\begin{aligned} \dot{y}_i^{i+1} &= -(x_i - x_{i+1}) \cos \theta_{i+1} \omega_{i+1} - (\dot{x}_i - \dot{x}_{i+1}) \sin \theta_{i+1} \\ &\quad - (y_i - y_{i+1}) \omega_{i+1} \sin \theta_{i+1} + (\dot{y}_i - \dot{y}_{i+1}) \cos \theta_{i+1} \\ &= -[(x_i - x_{i+1}) \cos \theta_{i+1} + (y_i - y_{i+1}) \sin \theta_{i+1}] \omega_{i+1} \\ &\quad - (v_i \cos \theta_i - v_{i+1} \cos \theta_{i+1}) \sin \theta_{i+1} \\ &\quad + (v_i \cos \theta_i - v_{i+1} \sin \theta_{i+1}) \cos \theta_{i+1} \\ &= -x_i^{i+1} \omega_{i+1} + v_i (-\cos \theta_i \sin \theta_{i+1} + \sin \theta_i \cos \theta_{i+1}) \\ &\quad + v_{i+1} (\cos \theta_{i+1} \sin \theta_{i+1} - \cos \theta_{i+1} \sin \theta_{i+1}) \\ &= -x_i^{i+1} \omega_{i+1} + v_i \sin(\theta_i - \theta_{i+1}). \end{aligned} \quad (6)$$

Además, definiendo la orientación relativa entre los robots como,

$$\alpha_i^{i+1} = \theta_i - \theta_{i+1} \quad (7)$$

se obtiene,

$$\dot{\alpha}_i^{i+1} = \omega_i - \omega_{i+1}.$$

Por lo tanto, la dinámica relativa entre los robots  $R_i$  y  $R_{i+1}$  con respecto a los ejes móviles  $X_{i+1} - Y_{i+1}$  resulta,

$$\begin{aligned} \dot{x}_i^{i+1} &= y_i^{i+1} \omega_{i+1} + v_i \cos \alpha_i^{i+1} - v_{i+1} \\ \dot{y}_i^{i+1} &= x_i^{i+1} \omega_{i+1} + v_i \sin \alpha_i^{i+1} \\ \dot{\alpha}_i^{i+1} &= \omega_i - \omega_{i+1}. \end{aligned} \quad (8)$$

*Observación 2:* Nótese que considerando el robot líder como un robot generador de una trayectoria deseada, el problema de seguimiento de trayectoria puede resolverse al considerar una retroalimentación de la forma propuesta por [17], [12],

$$\begin{aligned} v_2 &= k_1 e_{l_x} + v_d \cos(e_\theta) \\ \omega_2 &= \omega_d + k_2 v_d \frac{\sin(e_\theta)}{e_\theta} e_{l_y} - k_3 e_\theta \end{aligned} \quad (9)$$

Donde  $e_{l_x}$  y  $e_{l_y}$  serán los errores de medición entre la posición del robot líder con el seguidor así como  $v_d$  y  $\omega_d$  las velocidades del robot líder.

#### IV. PERTURBACIONES EN EL ROBOT LÍDER.

Para determinar el posible patinado y deslizamiento del robot líder  $R_i$  y de esta manera poder compensar la perturbación, considere ahora la configuración líder-seguidor con el robot líder sometido a perturbaciones externas, esto es, el robot líder toma la forma,

$$\begin{aligned} \dot{x}_i &= v_i \cos(\theta_i) + \dot{\gamma}_{i_x} \\ \dot{y}_i &= v_i \sin(\theta_i) + \dot{\gamma}_{i_y} \\ \dot{\theta}_i &= \omega_i + \dot{\gamma}_{i_\theta} \end{aligned} \quad (10)$$

mientras que el robot seguidor evoluciona conforme a,

$$\begin{aligned} \dot{x}_{i+1} &= v_{i+1} \cos \theta_{i+1} \\ \dot{y}_{i+1} &= v_{i+1} \sin \theta_{i+1} \\ \dot{\theta}_{i+1} &= \omega_{i+1}. \end{aligned} \quad (11)$$

Tomando en cuenta las distancias relativas dadas en (4), se tiene entonces que,

$$\begin{aligned} \dot{x}_i^{i+1} &= -(x_i - x_{i+1})\dot{\theta}_{i+1} \sin \theta_{i+1} + (\dot{x}_i - \dot{x}_{i+1}) \cos \theta_{i+1} \\ &\quad + (y_i - y_{i+1})\dot{\theta}_{i+1} \cos \theta_{i+1} + (\dot{y}_i - \dot{y}_{i+1}) \sin \theta_{i+1} \\ &= -(x_i - x_{i+1})\omega_{i+1} \sin \theta_{i+1} + (v_i \cos(\theta_i) + \dot{\gamma}_{i_x}) \\ &\quad - v_{i+1} \cos \theta_{i+1} \cos \theta_{i+1} + (y_i - y_{i+1})\omega_{i+1} \cos \theta_{i+1} \\ &\quad + (v_i \sin(\theta_i) + \dot{\gamma}_{i_y} - v_{i+1} \sin \theta_{i+1}) \sin \theta_{i+1} \\ &= -(x_i - x_{i+1}) \sin \theta_{i+1} + (y_i - y_{i+1}) \cos \theta_{i+1} \omega_{i+1} \\ &\quad + v_i (\cos(\theta_i) \cos \theta_{i+1} + \sin(\theta_i) \sin \theta_{i+1}) \\ &\quad - v_i (\cos^2 \theta_{i+1} + \sin^2 \theta_{i+1}) + \dot{\gamma}_{i_x} \cos \theta_{i+1} + \dot{\gamma}_{i_y} \sin \theta_{i+1} \\ &= y_i^{i+1} \omega_{i+1} - v_{i+1} + v_i [\cos(\theta_i) + \sin(\theta_i) \sin \theta_{i+1}] \\ &\quad \dot{\gamma}_{i_x} \cos \theta_{i+1} + \dot{\gamma}_{i_y} \sin \theta_{i+1} \\ &= y_{i+1} \omega_{i+1} - v_{i+1} + v_i [\cos(\theta_i - \theta_{i+1}) \cos \gamma_{i_\theta}] \\ &\quad \dot{\gamma}_{i_x} \cos \theta_{i+1} + \dot{\gamma}_{i_y} \sin \theta_{i+1} \\ &= y_i^{i+1} \omega_{i+1} - v_{i+1} + v_i \cos(\theta_i - \theta_{i+1}) \cos \gamma_{i_\theta} \\ &\quad \dot{\gamma}_{i_x} \cos \theta_{i+1} + \dot{\gamma}_{i_y} \sin \theta_{i+1}. \end{aligned} \quad (12)$$

Considerando además las derivadas de  $y_i^{i+1}$  y  $\alpha_i^{i+1}$  se obtiene,

$$\begin{aligned} \dot{x}_i^{i+1} &= y_i^{i+1} \omega_{i+1} - v_{i+1} + v_i \cos(\theta_i - \theta_{i+1}) \cos \gamma_{i_\theta} + \dot{\eta}_{i_x} \\ \dot{y}_i^{i+1} &= -x_i^{i+1} \omega_{i+1} + v_i \sin(\theta_i - \theta_{i+1}) + \dot{\eta}_{i_y} \\ \dot{\alpha}_i^{i+1} &= \omega_i - \omega_{i+1} + \dot{\eta}_{i_\theta} \end{aligned} \quad (13)$$

con,

$$\begin{aligned}\eta_{i_x} &= \dot{\gamma}_{i_x} \cos \theta_{i+1} + \dot{\gamma}_{i_y} \sin \theta_{i+1} \\ \eta_{i_y} &= -\dot{\gamma}_{i_x} \sin \theta_{i+1} + \dot{\gamma}_{i_y} \cos \theta_{i+1} \\ \eta_{i_\theta} &= \dot{\gamma}_{i_\theta}\end{aligned}\quad (14)$$

De (14) es evidente que cuando no exista perturbación,  $\eta_{i_x}$ ,  $\eta_{i_y}$  y  $\eta_{i_\theta}$  serán nulos.

La relación entre las perturbaciones en ejes fijos  $\dot{\gamma}_i$  y las presentes en ejes móviles  $\dot{\eta}_i$ , definidas como,

$$\dot{\gamma}_i = [\dot{\gamma}_{i_x} \quad \dot{\gamma}_{i_y} \quad \dot{\gamma}_{i_\theta}]^T \quad \dot{\eta}_i = [\dot{\eta}_{i_x} \quad \dot{\eta}_{i_y} \quad \dot{\eta}_{i_\theta}]^T$$

toman la forma,

$$\begin{bmatrix} \dot{\eta}_x \\ \dot{\eta}_y \\ \dot{\eta}_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 \\ -\sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_{1_x} \\ \dot{\gamma}_{1_y} \\ \dot{\gamma}_{1_\theta} \end{bmatrix}. \quad (15)$$

## V. ESTIMACIÓN DE LOS TÉRMINOS DE PERTURBACIÓN

Nótese que a partir de las mediciones  $x_i^{i+1}$ ,  $y_i^{i+1}$ ,  $\alpha_i^{i+1}$ , tomados a partir del robot seguidor, es posible reconstruir  $\dot{x}_i^{i+1}$ ,  $\dot{y}_i^{i+1}$ ,  $\dot{\alpha}_i^{i+1}$  bajo el conocimiento de las velocidades lineales y angulares de ambos robots, esta re-construcción puede realizarse a partir del sistema (13), por lo tanto la dinámica perturbada se observa de la forma,

$$\begin{aligned}\dot{x}_i^{i+1} &= y_i^{i+1} \omega_{i+1} + v_i \cos \alpha_i^{i+1} - v_{i+1} + \dot{\eta}_{i_x} \\ \dot{y}_i^{i+1} &= -x_i^{i+1} \omega_{i+1} + v_i \sin \alpha_i^{i+1} + \dot{\eta}_{i_y} \\ \dot{\alpha}_i^{i+1} &= \omega_i - \omega_{i+1} + \dot{\eta}_{i_\theta}.\end{aligned}\quad (16)$$

La dirección de los vectores de perturbación  $\dot{\eta}_i$  en los ejes móviles  $X_i^{i+1} - Y_i^{i+1}$  pueden verse en la Figura 4.

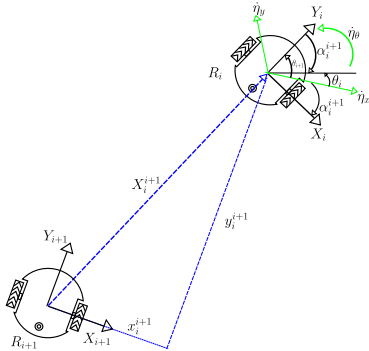


Figura 4. Vectores de perturbación  $\dot{\eta}_{i_x}$ ,  $\dot{\eta}_{i_y}$ ,  $\dot{\eta}_{i_\theta}$  en ejes móviles.

Considere que a partir del robot  $R_{i+1}$  se miden las distancias  $x_i^{i+1}$ ,  $y_i^{i+1}$  y la orientación  $\alpha_i^{i+1}$  y por medio de una derivada aproximada se obtiene,  $\dot{x}_i^{i+1}$ ,  $\dot{y}_i^{i+1}$ ,  $\dot{\alpha}_i^{i+1}$ .

Ante la aparición de perturbaciones en el robot  $R_i$ , se modifica su desplazamiento, pero no cambian directamente  $v_i$  y  $\omega_i$ , por lo tanto, a partir de (16), las perturbaciones estimadas resultan,

$$\begin{aligned}\dot{\eta}_x &= \dot{x}_i^{i+1} - y_i^{i+1} \omega_{i+1} - v_i \cos \alpha_i^{i+1} + v_{i+1} \\ \dot{\eta}_y &= \dot{y}_i^{i+1} + x_i^{i+1} \omega_{i+1} - v_i \sin \alpha_i^{i+1} \\ \dot{\eta}_\theta &= \dot{\alpha}_i^{i+1} - \omega_i + \omega_{i+1}.\end{aligned}\quad (17)$$

Nótese que las perturbaciones se estiman en función de los mediciones  $x_i^{i+1}$ ,  $y_i^{i+1}$ ,  $\alpha_i^{i+1}$ , su derivada aproximada y las velocidades de control de los robots.

## VI. PROBLEMA DE FORMACIÓN LÍDER-SEGUIDOR CON TIEMPO DE SEPARACIÓN

Para resolver el problema de formación retardada, considerando ahora el diagrama mostrado en la Figura 2 donde además del robot líder  $R_i$  y el robot seguidor  $R_{i+1}$  se muestra un robot virtual que  $R_{i\tau}$  representa la imagen retardada  $\tau$  unidades de tiempo del robot líder.

A partir de la Figura 2, puede describirse la estrategia de formación:

- i Considerando la posición del robot líder  $R_i(t)$  y la del robot retardado  $R_{i\tau}(t) = R_i(t-\tau)$ , obtener la dinámica relativa entre los dos robots.
- ii Considerando la relación  $X_i^{i+1}$  y la estimación  $X_{i\tau}^{i\tau}$  obtener la distancia  $X_{i\tau}^{i\tau}$ .
- iii Proponer una retroalimentación basada en  $X_{i\tau}^{i\tau}$  para lograr que  $R_{i+1}$  converja a la posición de  $R_{i\tau}$

Siguiendo los desarrollos presentados anteriormente la dinámica entre el robot  $R_i$  y su dinámica retrasada  $R_{i\tau}$  en el caso ideal libre de perturbaciones toma la forma,

$$\begin{aligned}\dot{x}_i^{i\tau} &= -v_i(\bar{\tau}) + v_i(t) \cos \alpha_i^{i\tau}(t) + \omega_i(\bar{\tau}) y_i^{i\tau}(t) \\ \dot{y}_i^{i\tau} &= v_i(t) \sin \alpha_i^{i\tau}(t) - \omega_i(\bar{\tau}) x_i^{i\tau}(t) \\ \dot{\alpha}_i^{i\tau} &= \omega_i(t) - \omega_i(\bar{\tau})\end{aligned}\quad (18)$$

donde,

$$\bar{\tau} = t - \tau \quad y \quad \alpha_i^{i\tau} = \theta_i - \theta_{i\tau}. \quad (19)$$

Nótese que la estimación de las distancias  $x_i^{i\tau}$ ,  $y_i^{i\tau}$  y la orientación relativa, esta basada en las velocidades aplicadas actuales  $v(t)$ ,  $\omega(t)$  y sus valores retrasados  $v(t-\tau)$ ,  $\omega(t-\tau)$ . Así como los valores  $\dot{\gamma}_{i_x}$ ,  $\dot{\gamma}_{i_y}$  y  $\dot{\gamma}_{i_\theta}$  son las perturbaciones presentes en el robot  $R_{i\tau}$

En el caso de considerar la dinámica perturbada del robot líder  $R_i$  dada en (16) se obtiene la dinámica relativa modificada,

$$\begin{aligned}\dot{x}_i^{i\tau} &= y_i^{i\tau} \bar{\omega}_{i\tau} + \cos \alpha_i^{i\tau} v_i - \bar{v}_{i\tau} + (\dot{\gamma}_{i_x} - \bar{\gamma}_{i_x}) \cos \bar{\theta}_{i\tau} \\ &\quad + (\dot{\gamma}_{i_y} - \bar{\gamma}_{i_y}) \sin \bar{\theta}_{i\tau} \\ \dot{y}_i^{i\tau} &= -x_i^{i\tau} \bar{\omega}_{i\tau} + \sin \alpha_i^{i\tau} v_i - (\dot{\gamma}_{i_x} - \bar{\gamma}_{i_x}) \sin \bar{\theta}_{i\tau} \\ &\quad + (\dot{\gamma}_{i_y} - \bar{\gamma}_{i_y}) \cos \bar{\theta}_{i\tau} \\ \dot{\alpha}_i^{i\tau} &= \omega_i - \omega_i(\bar{\tau}) - \dot{\gamma}_{i_\theta}\end{aligned}\quad (20)$$

la cual incluye el efecto de las perturbaciones modificando la trayectoria descrita en el sistema (20) con respecto al original (18).

Para lograr que  $R_{i+1}$  tienda a  $R_{i\tau}$  se requiere ahora que,

$$\lim_{t \rightarrow 0} x_{i\tau}^{i+1} = 0 \quad \lim_{t \rightarrow 0} y_{i\tau}^{i+1} = 0 \quad \lim_{t \rightarrow 0} \alpha_{i\tau}^{i+1} = 0. \quad (21)$$

Lo anterior es posible al considerar la dinámica relativa,

$$\begin{aligned}\dot{x}_{i\tau}^{i+1} &= y_{i\tau}^{i+1} \omega_{i+1} - v_{i+1} + \bar{v}_{i\tau} \cos \alpha_{i\tau}^{i+1} \\ \dot{y}_{i\tau}^{i+1} &= -x_{i\tau}^{i+1} \omega_{i+1} + \bar{v}_{i\tau} \sin \alpha_{i\tau}^{i+1} \\ \dot{\alpha}_{i\tau}^{i+1} &= \bar{\omega}_{i\tau} - \omega_{i+1}\end{aligned}\quad (22)$$

Considere ahora la función candidata de Lyapunov,

$$V = \frac{k_3}{2} [(x_{i\tau}^{i+1})^2 + (y_{i\tau}^{i+1})^2] + \frac{1}{2} (\alpha_{i\tau}^{i+1})^2. \quad (23)$$

Tomando la derivada con respecto al tiempo se obtiene,

$$\begin{aligned} \dot{V} &= k_3(x_{i\tau}^{i+1}\dot{x}_{i\tau}^{i+1} + y_{i\tau}^{i+1}\dot{y}_{i\tau}^{i+1}) + \alpha_{i\tau}^{i+1}\dot{\alpha}_{i\tau}^{i+1} \\ &= k_3x_{i\tau}^{i+1}(-v_{i+1} + \bar{v}_{i\tau} \cos \alpha_{i\tau}^{i+1}) + k_3y_{i\tau}^{i+1}(\bar{v}_{i\tau}) \sin \alpha_{i\tau}^{i+1} \\ &\quad + \alpha_{i\tau}^{i+1}(\bar{\omega}_{i\tau} - \omega_{i+1}). \end{aligned} \quad (24)$$

Defínase entonces la retroalimentación,

$$v_{i+1} = \bar{v}_{i\tau} \cos \alpha_{i\tau}^{i+1} + k_1x_{i\tau}^{i+1}. \quad (25)$$

Se obtiene entonces,

$$\begin{aligned} \dot{V} &= -k_1k_3(x_{i\tau}^{i+1})^2 + k_3y_{i\tau}^{i+1}(\bar{v}_{i\tau} \sin \alpha_{i\tau}^{i+1}) \\ &\quad + \alpha_{i\tau}^{i+1}(\bar{\omega}_{i\tau} - \omega_{i+1}) \end{aligned} \quad (26)$$

Por otra parte, considere,

$$\omega_{i+1} = k_2\alpha_{i\tau}^{i+1} + k_3y_{i\tau}^{i+1}(\bar{v}_{i\tau} \frac{\sin \alpha_{i\tau}^{i+1}}{\alpha_{i\tau}^{i+1}}) + \bar{\omega}_{i\tau} \quad (27)$$

con lo cual se obtiene,

$$\begin{aligned} \dot{V} &= -k_1k_3(x_{i\tau}^{i+1})^2 + k_3y_{i\tau}^{i+1}(\bar{v}_{i\tau} \sin \alpha_{i\tau}^{i+1}) \\ &\quad + \alpha_{i\tau}^{i+1}(\bar{\omega}_{i\tau} - k_2\alpha_{i\tau}^{i+1} - k_3y_{i\tau}^{i+1}(\bar{v}_{i\tau} \sin \alpha_{i\tau}^{i+1}) - \bar{\omega}_{i\tau}) \\ &= -k_1k_3(x_{i\tau}^{i+1})^2 - k_2(\alpha_{i\tau}^{i+1})^2 \end{aligned} \quad (28)$$

Como  $\dot{V}$  es semidefinida negativa, es posible aplicar el Lema de Barbalat, el cual muestra la convergencia buscada.

## VII. EXPERIMENTOS EN SIMULACIÓN NUMÉRICA

Para realizar una evaluación de la estrategia de formación basada en tiempos de separación sujeta a posibles perturbaciones de patinado y deslizamiento se considerará un conjunto de tres robots móviles del tipo diferencial. Se asumirá que el robot líder de la formación  $R_1$  describe cualquier trayectoria generada por velocidades  $v_1(t)$ ,  $w_1(t)$  acotadas. Se considerará que el robot líder, en el caso ideal sigue una trayectoria tipo lemniscata descritas por:

$$\begin{aligned} x_d(t) &= a \cos(pt) \\ y_d(t) &= b \sin(2pt) \end{aligned} \quad (29)$$

con  $a = 0,8$ ,  $b = 0,6$  and  $p = \frac{2\pi}{50}$ . Lo que produce las velocidades,

$$\begin{aligned} v_1(t) &= \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \\ \omega_1(t) &= \frac{\dot{y}_d(t)\dot{x}_d(t) - \dot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \end{aligned} \quad (30)$$

En la formación en cadena, se desea que el robot  $R_2$  siga la trayectoria retardada del robot  $R_1$  y que el robot  $R_3$  siga la trayectoria retrasada del robot  $R_2$ , en ambos casos con un retardo de  $\tau = 3$  segundos.

Para mostrar la estimación y compensación de las perturbaciones externas desconocidas, se asumirá que en  $t = 10$  ocurren perturbaciones afectando al robot  $R_1$  de la forma,

$$\begin{aligned} \dot{\gamma}_{1x}(t) &= \frac{a_{1x}}{\sqrt{2\pi\alpha_{1x}^2}} e^{-\frac{t^2}{2\alpha_{1x}^2}} \\ \dot{\gamma}_{1y}(t) &= \frac{a_{1y}}{\sqrt{2\pi\alpha_{1y}^2}} e^{-\frac{t^2}{2\alpha_{1y}^2}} \\ \dot{\gamma}_{1\theta}(t) &= 0. \end{aligned} \quad (31)$$

La evolución temporal de las perturbaciones desconocidas  $\dot{\gamma}_{1i}(t)$  en (31), la cual es tipo Gauss, tratando de asemejarse a un movimiento repentino, que altere la posición del robot afectado, el cual puede observarse en la Figura 5. La estimación de las perturbaciones actuando en el robot  $R_1$  calculadas a partir de mediciones sobre el eje móvil  $X_2 - Y_2$  colocado sobre el robot  $R_2$  se muestran también en la Figura 5.

Las evolución de las señales obtenidas por los estimadores de las trayectorias deseadas retardadas  $X_1^{1\tau}(t)$  para la dinámica entre el robot  $R_1$  y su imagen retardada  $R_{1\tau}$  y la dinámica relativa entre el robot  $R_2$  y su imagen retardada  $R_{2\tau}$  dada por  $X_2^{2\tau}(t)$ , obtenidas con respecto a los ejes  $X_1 - Y_1$  y  $X_2 - Y_2$  respectivamente, se muestran en la Figura 6.

Los errores de seguimiento del robot  $R_2$  a la imagen retardada del Robot  $R_1$ ,  $X_{1\tau}^2(t)$  y del robot  $R_3$  a la imagen retardada del robot  $R_2$  dada por  $X_{2\tau}^3(t)$  pueden verse en la Figura 7. Nótese la que el error  $X_{1\tau}^2(t)$  es medido a partir del eje móvil  $X_2 - Y_2$  y el error  $X_{2\tau}^3(t)$  es medido sobre el eje  $X_3 - Y_3$ . La convergencia de estos errores es clara para los periodos donde no existen perturbaciones, mientras que el error  $X_{1\tau}^2(t)$  permanece acotado cuando estas actúan. El error  $X_{2\tau}^3(t)$  nunca es afectado por las perturbaciones.

Aún cuando con la estrategia móvil presentada, no es posible tener el comportamiento de los robots sobre los ejes inerciales  $X - Y$ , tomando los datos necesarios a partir de la simulación, es posible mostrar esta evolución. La Figura 8 muestra el efecto de la perturbación sin aplicar la estrategia de compensación de este trabajo, mientras que en la Figura 9 se muestra la evolución de los robots en el plano  $X - Y$  cuando se aplica la compensación de las perturbaciones mediante su estimación propuesta en este trabajo. Es claro como el Robot  $R_2$  converge adecuadamente a la trayectoria retardada del robot  $R_1$ .

## VIII. CONCLUSIONES

En este trabajo se presentó una estrategia de estimación y compensación del efecto de perturbaciones de deslizamiento y patinado en una formación en cadena de robots móviles diferenciales. Se muestra como es posible estimar la perturbaciones a partir de mediciones de distancias y ángulos relativos entre dos robots consecutivos  $R_i$ ,  $R_{i+1}$ , efectuadas a partir del robot  $R_{i+1}$  y como estas mediciones pueden ser usadas para modificar la trayectoria deseada del robot  $R_{i+1}$ . Se muestra como el robot  $R_{i+1}$  puede seguir la trayectoria retrasada,  $\tau$  unidades de tiempo, del robot  $R_i$  aún en los periodos de tiempo en los cuales las perturbaciones están

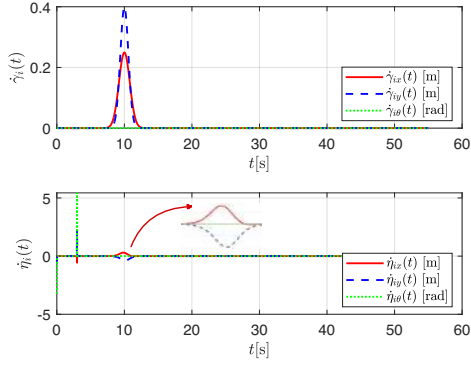


Figura 5. Perturbaciones desconocidas  $\hat{\gamma}_i(t)$  en los ejes fijos  $X - Y$  y su estimación  $\hat{\eta}_i(t)$  en los ejes móviles  $X_2 - Y_2$ .

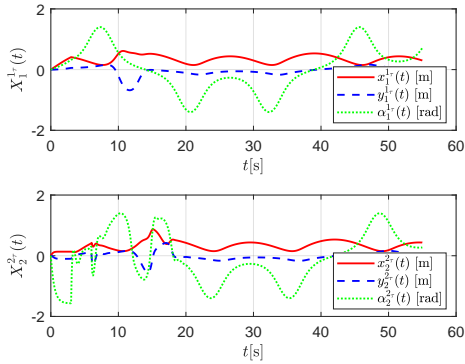


Figura 6. Evolución de las señales estimadas retardadas  $X_1^{1\tau}(t)$  y  $X_2^{2\tau}(t)$ .

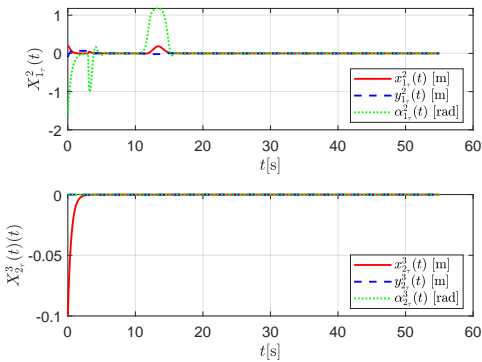


Figura 7. Evolución de los errores de seguimiento móvil  $X_{1\tau}^2(t)$  y  $X_{2\tau}^3(t)$ .

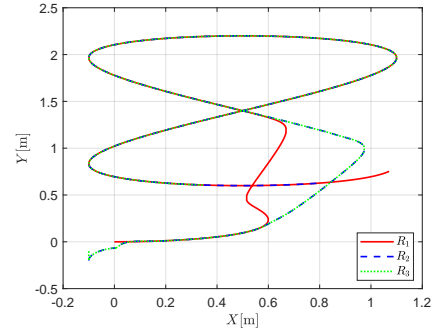


Figura 8. Evolución de los robots en el plano Cartesiano sin compensación de perturbaciones.

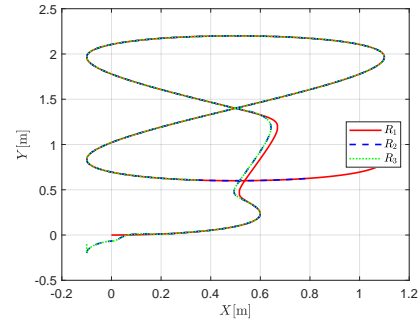


Figura 9. Evolución de los robots en el plano Cartesiano aplicando la estrategia de compensación de perturbaciones propuesta.

afectando la formación. La convergencia del robot  $R_{i+1}$  a la trayectoria retrasada del robot  $R_i$  se muestra mediante técnicas de Lyapunov.

## REFERENCIAS

- [1] Michael Lammert, Jeffrey Gonder, Kenneth Kelly, Kambiz Salari, and Jason Ortega. Class 8 tractor trailer platooning: Effects, impacts, and improvements, 2016.
- [2] Eric Guizzo. Three engineers, hundreds of robots, one warehouse. *IEEE Spectrum*, 45(7):26–34, 2008.
- [3] Vicente Milanés, Steven Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, and Masahide Nakamura. Cooperative adaptive cruise control in real traffic situations. *Intelligent Transportation Systems, IEEE Transactions on*, 15:296–305, 02 2014.
- [4] Derek Caveney. Cooperative vehicular safety applications. *IEEE Control Systems Magazine*, 30:38 – 53, 09 2010.
- [5] Sonja Stüdli, M.M. Seron, and R.H. Middleton. Vehicular platoons in cyclic interconnections with constant inter-vehicle spacing. *IFAC-PapersOnLine*, 50:2511–2516, 07 2017.
- [6] Yinglong He, Biagio Ciuffo, Quan Zhou, Michail Makridis, Konstantinos Mattas, Ji Li, Ziyang Li, Fuwu Yan, and Hongming Xu. Adaptive cruise control strategies implemented on experimental vehicles: A review. *IFAC-PapersOnLine*, 52(5):21–27, 2019. 9th IFAC Symposium on Advances in Automotive Control AAC 2019.
- [7] Muhammad Mohd Ariffin, Mohd Azizi Abdul Rahman, and Hairi Zamzuri. Effect of leader information broadcasted throughout vehicle platoon in a constant spacing policy. pages 132–137, 10 2015.
- [8] J. Ploeg, E. Semsar-Kazerouni, G. Lijster, N. Wouw, van de, and H. Nijmeijer. Graceful degradation of cooperative adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):488–497, 2015.

- [9] Steven Shladover. Automated vehicles for highway operations (automated highway systems). *Proceedings of The Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering - PROC INST MECH ENG I-J SYST C*, 219:53–75, 02 2005.
- [10] A. Rodríguez-Angeles M. Velasco-Villa. C. A. Domínguez-Ortega, R. D. Cruz-Morales. Control de un convoy de robots móviles con tiempo variable de separación. *Memorias del Congreso Nacional de Control Automático*, pages 140–145, 2018.
- [11] Raúl Cruz-Morales, Martín Velasco-Villa, and Alejandro Rodríguez-Ángeles. Chain formation control for a platoon of robots using time-gap separation. *International Journal of Advanced Robotic Systems*, 15:172988141877085, 04 2018.
- [12] A. Rodríguez-Angeles. I. Z. Maruri-López, M. Velasco-Villa. Leader-follower formation of mobile robots on a mobile axis system. *Memorias del Congreso Nacional de Control Automático*, pages 1–6, 2020.
- [13] Hesheng Wang, Dejun Guo, Xinwu Liang, Weidong Chen, Guoqiang Hu, and Kam K. Leang. Adaptive vision-based leader-follower formation control of mobile robots. *IEEE Transactions on Industrial Electronics*, 64(4):2893–2902, 2017.
- [14] I. Z. Maruri-Lopez J. A. Baez-Hernandez R. D. Cruz-Morales M. Velasco-Villa, A. Rodríguez-Angeles. Leader-follower formation control based on non-inertial frames for non-holonomic mobile robots. *Plos One*, 19, January 2024.
- [15] G. Campion, G. Bastin, and B. Dandrea-Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1):47–62, 1996.
- [16] Carlos Canudas de Wit, Georges Bastin, and Bruno Siciliano. *Theory of Robot Control*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 1996.
- [17] A. Rodríguez-Angeles-M. Velasco-Villa M. Infante-Jacobo, R. D. Cruz-Morales. Formación de vehículos mediante tiempos de separación utilizando mediciones relativas. *Memorias del Congreso Nacional de Control Automático*, pages 45–50, 2021.

# Implementación de un observador de tipo Luenberger para el sistema caótico de Kapitaniak con el uso de tarjetas de adquisición de datos

Tonalli C. Galicia López  
*Postgraduate Program in Engineering Sciences, Dynamics Systems and Control*  
*Tecnológico Nacional de México*  
*Texnológico de Tijuana*  
Tijuana B.C., México  
tonalli.galicia@tectijuana.edu.mx

Paul J. Campos Hernández  
*Postgraduate Program in Engineering Sciences, Dynamics Systems and Control*  
*Tecnológico Nacional de México*  
*Texnológico de Tijuana*  
Tijuana B.C., México  
paul.campos@tectijuana.edu.mx

Diana Gamboa Loaiza  
*Postgraduate Program in Engineering Sciences, Dynamics Systems and Control*  
*Tecnológico Nacional de México*  
*Texnológico de Tijuana*  
Tijuana B.C., México  
diana.gamboa@tectijuana.edu.mx

Luis N. Coria de L.R.  
*Postgraduate Program in Engineering Sciences, Dynamics Systems and Control*  
*Tecnológico Nacional de México*  
*Texnológico de Tijuana*  
Tijuana B.C., México  
luis.coria@tectijuana.edu.mx

**Abstract**—En este trabajo se presenta la implementación del sistema caótico de Kapitaniak utilizando las tarjetas de adquisición de datos PCI-6221 (con módulo de comunicación SCB-68A) y USB-6211, desarrolladas por National Instruments, con el uso de MATLAB® como interfaz. La implementación se realiza mediante un esquema independiente y asíncrono para emular la dinámica maestro-esclavo entre el sistema y el observador. Se emplea el método numérico Runge-Kutta de 4to orden para la estimación de las variables de estado en tiempo real, los cuales se actualizan en ventanas temporales de 50 ms siendo visualizados en osciloscopio. Los resultados hacen factible utilizar MATLAB® en conjunto a tarjetas DAQ's para la adquisición y procesamiento de datos que describan un esquema de observador, permitiendo el monitoreo en tiempo real de la dinámica del sistema.

**Palabras clave**—Observador, MATLAB®, DAQ-6221, SCB68A, DAQ-USB6211.

## I. INTRODUCCIÓN

Los sistemas caóticos son sistemas dinámicos caracterizados por su alta sensibilidad a condiciones iniciales, órbitas periódicas densas y comportamientos topológicos mixtos [1], lo que los hace ideales para aplicaciones en áreas como criptografía, control, procesamiento de señales, redes neuronales, entre otras [2]. Dentro de dichas aplicaciones se suele requerir de un esquema maestro-esclavo lo que implica

la sincronización de estos a través de la observación de uno o varios estados del sistema maestro [3].

En el contexto de los sistemas de control, los observadores son algoritmos utilizados para proporcionar estimaciones precisas de los estados internos de un sistema, que permiten su monitorización así como la detección de fallos y la optimización del control [4,5]. Estos sistemas son cruciales en escenarios donde no se pueden medir todas las variables de estado directamente, debido a restricciones físicas o económicas; además, su implementación se facilita significativamente con entornos computacionales como MATLAB [6], el cual permite el desarrollo de modelos y simulaciones sofisticadas.

Estudios recientes confirman la eficacia de MATLAB en sistemas de control basados en observadores y en la adquisición de datos. En particular, se ha explorado el uso de observadores de Luenberger adaptados para sistemas no lineales, utilizando redes neuronales para mejorar la estimación de estados en entornos con condiciones iniciales variadas [8].

Otro enfoque significativo incluye el diseño basado en datos para la predicción y control utilizando el método predictivo de modelos (MPC, por sus siglas en inglés) en MATLAB. Este método se ha utilizado para formular problemas de optimización y manejo de restricciones en sistemas dinámicos, demostrando su aplicabilidad en sistemas de control industrial y robótico [9]. Además, se ha desarrollado un sistema de adquisición de datos embebido para el análisis de la

Los autores agradecen al TecNM con los proyectos PRODEP ITTIJ-CA-10, 20136.24-P y 20229.24-P. Este trabajo fue desarrollado en el marco de la "Red Internacional de Control y Cómputo Aplicados" soportada por el TecNM.

dinámica vertical de vehículos [10]. Este sistema emplea un microcontrolador para la recolección de datos en tiempo real, los cuales son analizados mediante MATLAB para evaluar parámetros de la suspensión del vehículo.

Estos desarrollos subrayan la importancia de herramientas como MATLAB para la implementación y análisis en tiempo real de sistemas complejos, consolidando su papel en la ingeniería moderna y la investigación aplicada.

El objetivo de este trabajo es explorar la implementación de esquemas de sincronización entre maestro-observador dentro del entorno de MATLAB usando tarjetas DAQ para la comunicación. Se destaca el hecho que la metodología propuesta no es excluyente de implementar otros tipos de sistemas de ecuaciones diferenciales ordinarias (EDO's), sean o no caóticos, para su generación y/o sincronización.

Este artículo se divide en las siguientes secciones: en la sección II se describe el sistema caótico de Kapitaniak y el observador a implementar, en la sección III se presenta la discretización del sistema y el observador. En la sección IV se dan las especificaciones de hardware y software mientras que en la sección V se describe el proceso de implementación, seguido de la sección VI donde se exponen los resultados. Por último, en la sección VII se presentan las conclusiones obtenidas.

## II. SISTEMA CAÓTICO DE KAPITANIAK

El sistema caótico multiestable de Kapitaniak es un sistema caótico tridimensional con un atractor de punto fijo y un atractor extraño oculto el cual se presenta cuando las condiciones iniciales son  $\mathbf{X}(0) = [5.4, -1.8, 3.3]$  [11]. El modelo matemático del sistema se describe por:

$$\frac{dx_1}{dt} = x_3, \quad (1)$$

$$\frac{dx_2}{dt} = -x_2 - x_3, \quad (2)$$

$$\frac{dx_3}{dt} = 0.1x_1 + 5x_2 - x_3 + x_1x_2 - 0.3x_1x_3 + a; \quad (3)$$

donde el parámetro  $a = 1$ . La Figura 1 muestra los planos de fase de la dinámica del sistema.

### A. Observador

En [12] se propone un observador de tipo Luenberger, el cual se describe por las ecuaciones (4)-(6). Este observador considera disponible el estado  $x_1$  del sistema (1)-(3), así como el conjunto  $\Omega$  (7) donde se desarrolla la dinámica del sistema por un periodo de  $10^6$  s.

$$\frac{dz_1}{dt} = z_3 + L_1(x_1 - z_1), \quad (4)$$

$$\frac{dz_2}{dt} = -z_2 - z_3 + L_2(x_1 - z_1), \quad (5)$$

$$\frac{dz_3}{dt} = 0.1z_1 + 5z_2 - z_3 + z_1z_2 - 0.3z_1z_3 + 1 + L_3(x_1 - z_1); \quad (6)$$

$$\Omega = \{(x_1, x_2, x_3) \mid x_1 \in [-4.9150, 10.8287], \\ x_2 \in [-5.3136, 13.7184], x_3 \in [-14.6249, 22.3794]\}; \quad (7)$$

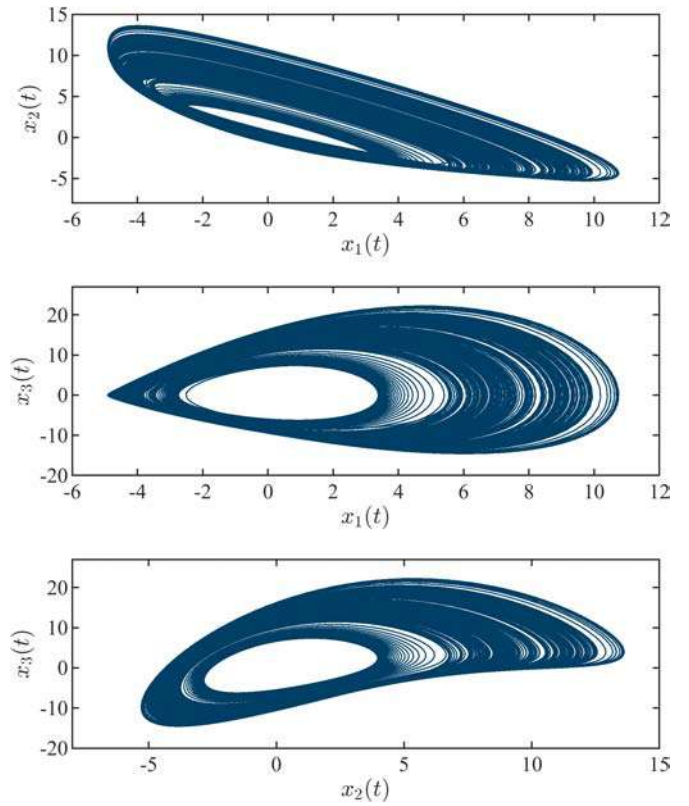


Figura 1. Planos de fase  $(x_1, x_2)$ ,  $(x_1, x_3)$  y  $(x_2, x_3)$  del sistema de Kapitaniak para una simulación con duración de 1000 s.

donde  $L_1 = 24.4844$ ,  $L_2 = -1.0968$  y  $L_3 = -0.913$ . En la Figura 2 se representa el comportamiento del sistema y el observador para el estado  $x_2$ , de forma similar para el estado  $x_3$  en la Figura 3, bajo condiciones iniciales  $\mathbf{X}(0) = [5.4, -1.8, 3.3]$  y  $\mathbf{Z}(0) = [0, 0, 0]$ , iniciando la sincronización en  $t = 0$  s. La Figura 4 ilustra el comportamiento del error para los tres estados, mostrando una sincronización en los primeros 8 s.

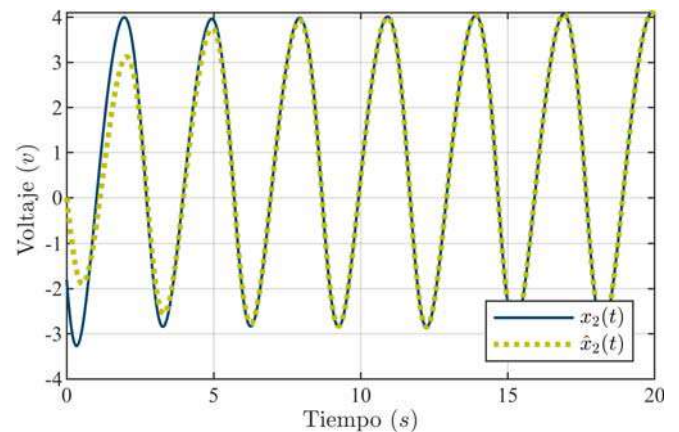


Figura 2. Comparación entre sistema y observador para  $x_2(t)$ .

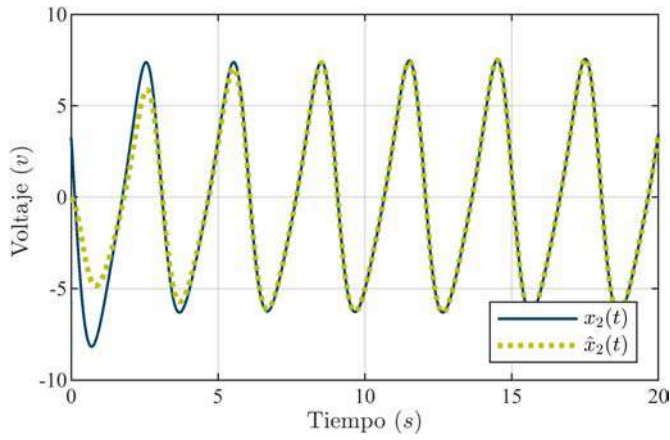


Figura 3. Comparación entre sistema y observador para  $x_3(t)$ .

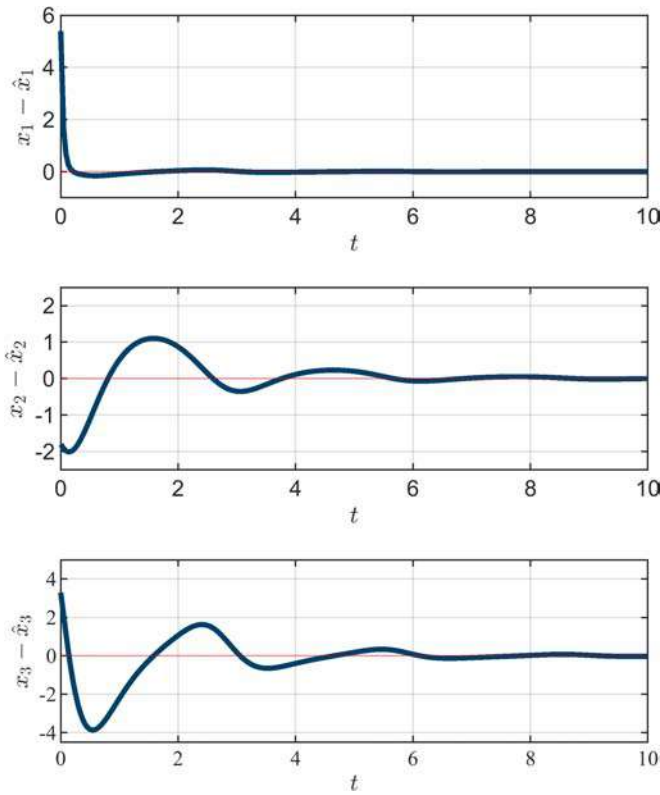


Figura 4. Dinámica del error para el observador de Kapitanik propuesto.

### III. DISCRETIZACIÓN

El sistema (1)-(3) discretizado se representa como

$$x_1[n+1] = x_3[n], \quad (8)$$

$$x_2[n+1] = -x_2[n] - x_3[n], \quad (9)$$

$$x_3[n+1] = 0.1x_1[n] + 5x_2[n] - x_3[n] + x_1[n]x_2[n] - 0.3x_1[n]x_3[n] + a; \quad (10)$$

a su vez, el observador (4)-(6) se representa como

$$z_1[m+1] = z_3[m] + L_1(x_1[n] - z_1[m]), \quad (11)$$

$$z_2[m+1] = -z_2[m] - z_3[m] + L_2(x_1[n] - z_1[m]), \quad (12)$$

$$z_3[m+1] = 0.1z_1[m] + 5z_2[m] - z_3[m] + z_1[m]z_2[m] + 1 + L_3(x_1[n] - z_1[m]); \quad (13)$$

donde los estados  $\mathbf{X}[n+1]$  y  $\mathbf{Z}[m+1]$  son calculados de forma asíncrona, ambos a través del método de Runge-Kutta de 4to orden. En la metodología de este trabajo, los valores de las muestras  $[n-1]$  y  $[m-1]$  no son almacenados, toda vez que los mismos son reemplazados por la muestra posterior lo que permite eficientizar el uso de memoria y ejecutar el código en periodos indefinidos de prueba.

### IV. ESPECIFICACIONES DE HARDWARE

A continuación se enlistan las características del hardware utilizado para la implementación.

#### • Computadora - Sistema

- **Modelo:** Dell OptiPlex 3020,
- **Procesador:** Intel Core i5-4590 @ 3.30GHz, 3301 MHz - 4 núcleos,
- **Memoria RAM:** 12 GB DDR4,
- **Almacenamiento:** SSD 500 GB,
- **Sistema Operativo:** Windows 10 Pro.

#### • Computadora - Observador

- **Modelo:** Dell Precision T1700,
- **Procesador:** Intel Xeon E3-1246 @ 3.50GHz, 3501 MHz - 4 núcleos,
- **Memoria RAM:** 32 GB DDR4,
- **Almacenamiento:** HDD 1 TB,
- **Sistema Operativo:** Windows 10 Pro.

#### • Software

- **MATLAB R2023a,**
- \* **Paquetes utilizados:** Data Acquisition Toolbox,
- **NI MAX 2024 versión Q2.**

#### • Otros equipos

- **Osciloscopio:** Tektronix TDS2012B,
- **DAQ USB-6211 (Sistema),**
- \* **Tipo:** Dispositivo de adquisición de datos USB,
- \* **Entradas analógicas:** 16 entradas analógicas,
- \* **Salidas analógicas:** 2 salidas analógicas,
- \* **Entradas digitales:** 24 entradas digitales,
- \* **Salidas digitales:** 24 salidas digitales,
- \* **Resolución:** 16 bits,
- \* **Velocidad de muestreo:** 250 kS/s.
- **DAQ PCI-6221 con módulo SCB-68A (Observador)**
- \* **Tipo:** Caja de terminales (Terminal Block) conectada por puerto PCI,
- \* **Conector:** Conector de 68 pines para conectar con dispositivos DAQ,
- \* **Entradas Analógicas:** 16 entradas analógicas,
- \* **Salidas Analógicas:** 2 salidas analógicas,

- \* **Entradas Digitales:** 24 líneas de entrada digital,
- \* **Salidas Digitales:** 24 líneas de salida digital,
- \* **Contadores/Timers:** 2 contadores de 32 bits.

## V. METODOLOGÍA

Para el diseño del esquema del observador se emplean dos algoritmos, cada uno ejecutándose de forma independiente y asíncrona en los equipos, ver Figura 5. Ambos algoritmos permiten realizar escalado en tiempo y resultan ser flexibles en caso de modificar el hardware empleado, ya sea para la generación de la dinámica del sistema así como la implementación del observador. De manera general se tienen como limitaciones las velocidades de actualización entre la lectura y la generación de señales análogas (inherentes a MATLAB), así como la velocidad de procesamiento de los equipos de cómputo para realizar las estimaciones en tiempo real. Se tiene que el primer algoritmo realiza el cálculo de la dinámica del sistema, así como el envío de la señal hacia el observador mediante una salida analógica, además de permitir la visualización de cada uno de los estados del sistema; el diagrama de flujo se muestra en la Figura 6; . El segundo algoritmo recibe la información enviada por el sistema mediante una entrada analógica y se calcula la dinámica del observador, para luego enviarse para su visualización mediante salida(s) analógica(s) según se especifique en el algoritmo, ver Figura 7.

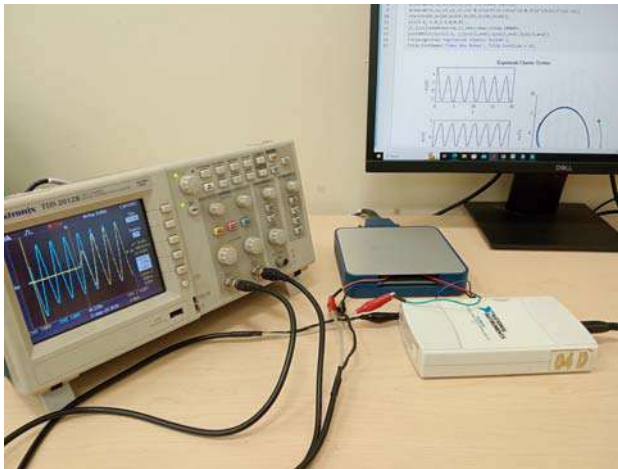


Figura 5. Interconexión de DAQ's. Los equipos que los controlan son independientes y asíncronos entre sí.

Para el primer algoritmo, se emplea la DAQ USB-6211 teniendo como salida analógica en AO0 los valores del estado  $x_1$  (el cual es enviado al observador); mientras que, la salida AO1 es destinada a la escritura de los estados  $x_2$  o  $x_3$ , según se requiera. El observador hace uso de la DAQ PCI-6221 a través del módulo SCB68A realizando la lectura del estado  $x_1$  en el puerto AI0, y visualizando los estados del observador  $z_2$  o  $z_3$  a través de AO1.

Los valores tanto del sistema maestro como del observador son actualizados cada 50 ms, lapso en el cual se calculan los nuevos valores a enviar en el siguiente periodo a través del

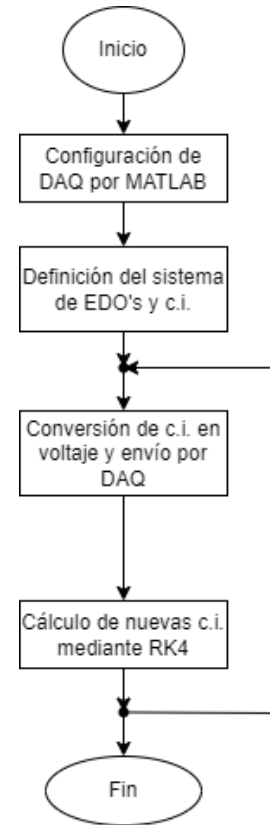


Figura 6. Diagrama de flujo para la generación de la dinámica del sistema.

método de Runge-Kutta de 4to orden con un paso de 0.0004 s; este cálculo se realiza en un tiempo aproximado de 10 ms con lo que los valores nuevos se obtienen aproximadamente 40 ms antes de ser requeridos.

Dado que las DAQ's se limitan a  $\pm 10$  v, para la lectura y escritura de señales analógicas, es necesario hacer un escalamiento en la salida del sistema maestro y su compensación en los cálculos del observador, con lo que el valor del estado  $x_1$  enviado del maestro es escalado por un factor de 0.9, el cual numéricamente se compensa con su recíproco en el observador. La visualización en osciloscopio de los estados  $x_2$  con  $z_2$ , y  $x_3$  con  $z_3$  también escalados en factores de 0.7 y 0.4 respectivamente, esto adecuándose al conjunto  $\Omega$  en (7). Es importante enfatizar el hecho de que estos escalamientos solamente son aplicados para la visualización de los estados y no para la resolución del sistema.

## VI. RESULTADOS

En la Figura 8 se ilustra el resultado de simular la sincronización del estado  $x_2$  con  $z_2$  en MATLAB, sin realizar escalamiento en la magnitud de los valores de los estados y con un retardo en la activación del observador de 10 s, teniendo una sincronización aparente a partir de los 7 s posteriores a que el observador es activado. De manera análoga, la Figura 9 muestra el comportamiento para la sincronización de  $x_3$  con

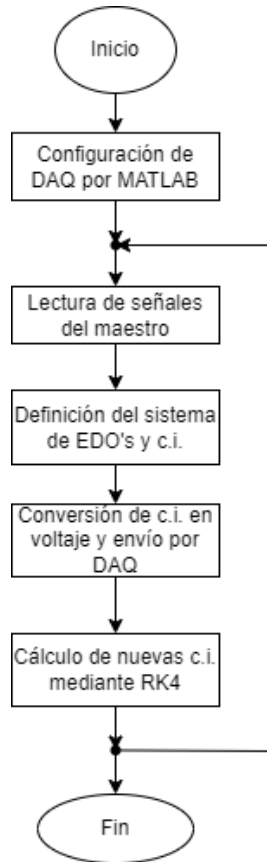


Figura 7. Diagrama de flujo para el observador.

$z_3$ , notándose que la sincronización es aparente alrededor de los 5 s después de la activación del observador. Las Figuras 10 y 11 son el resultado de la implementación del observador: la Figura 10 compara las dinámicas de los estados  $x_2$  con  $z_2$  buscando las mismas condiciones en que fueron simuladas en MATLAB, obteniendo un resultado semejante en el tiempo de sincronización; del mismo modo la Figura 11 ofrece un resultado positivo al comparar el tiempo de sincronización para  $x_3$  con  $z_3$  con el esperado en la simulación por MATLAB. Cabe añadir que la implementación fue sometida a un lapso de 24 horas manteniéndose la sincronización de ambos sistemas.

## VII. CONCLUSIONES

La implementación de este observador a través de DAQ's con el uso de MATLAB permite la sincronización entre maestro-observador, esto dentro de los tiempos esperados por simulaciones independientes y bajo un esquema donde maestro y observador trabajan de forma independiente y asíncrona. El hecho de que la sincronización se mantiene después de un periodo de prueba de 24 horas, así como la versatilidad de adecuar los valores de las señales para evitar superar las capacidades de lectura y escritura de voltaje, fomenta la exploración de estas herramientas para emular esquemas de sincronización a largo plazo, enriqueciendo las alternativas para la sincronización de sistemas. Es importante mencionar que la metodología empleada no es exclusiva para las tarjetas

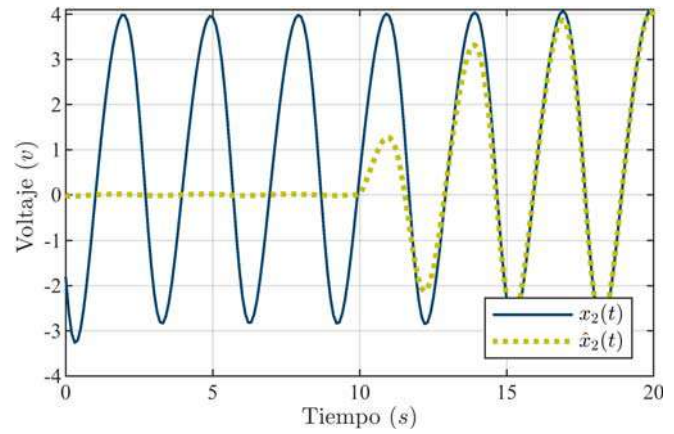


Figura 8. Comparación entre sistema y observador para  $x_2(t)$  con un retraso en el observador de 10 s.

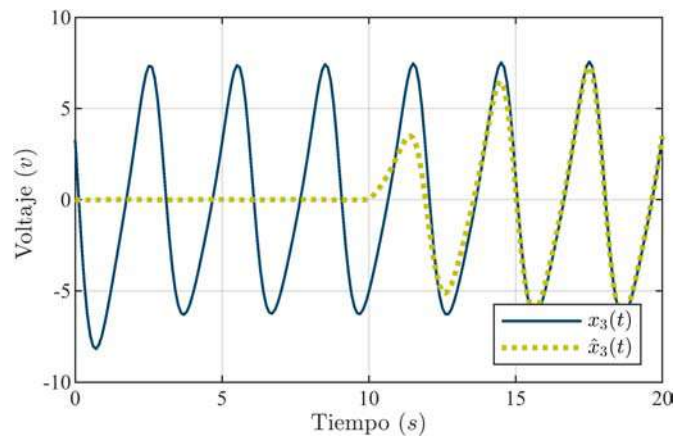


Figura 9. Comparación entre sistema y observador para  $x_3(t)$  con un retraso en el observador de 10 s.

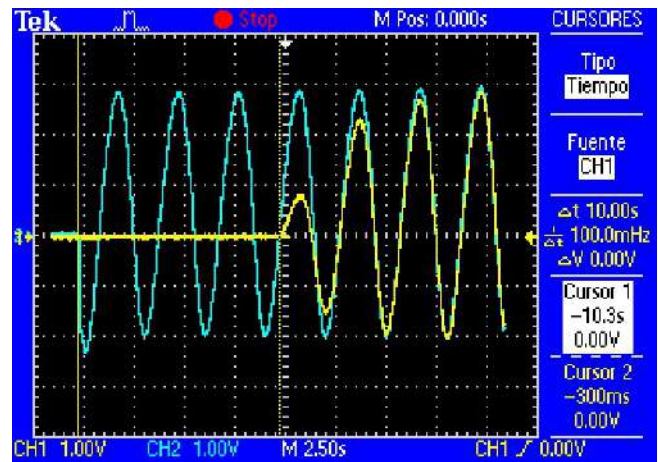


Figura 10. Comparación en osciloscopio entre sistema (azul) y observador (amarillo) para  $x_2(t)$  con un retraso en este último de 10 s.

de adquisición de datos aquí presentadas, y en caso de utilizar hardware distinto entonces el código se modifica ligeramente

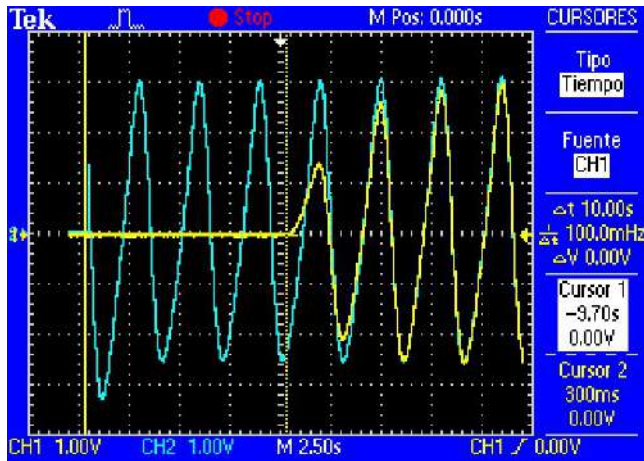


Figura 11. Comparación en osciloscopio entre sistema (azul) y observador (amarillo) para  $x_3(t)$  con un retraso en este último de 10 s.

solo para la configuración de los puertos de entrada y salida del dispositivo seleccionado, esto gracias a que el programa está en lenguaje m y a la versatilidad de MATLAB de poderse comunicar con diversos dispositivos externos.

Dados los resultados obtenidos, se estudiará la factibilidad de implementar un controlador en un esquema semejante. Además, se trabajará en la optimización del algoritmo del método numérico para reducir los tiempos de cálculo y generar ventanas de tiempo menores entre actualizaciones de valores de estados. Se realizarán etapas de adecuación de señal entre la comunicación de los dispositivos para la comparación entre los tiempos de convergencia con y sin el uso de esta.

#### REFERENCIAS

- [1] J. H. Pérez-Cruz, E. A. Portilla-Flores, P. A. Niño-Suárez, and R. Rivera-Blas, "Design of a nonlinear controller and its intelligent optimization for exponential synchronization of a new chaotic system", *Optik*, vol. 130, pp. 201-212, 2017.
- [2] G. Chen and T. Ueta, "Chaos in Circuits and Systems", vol. 11. World Scientific, 2002.
- [3] V. Lynnyk and S. Čelikovský, "Generalized synchronization of chaotic systems in a master-slave configuration," *2021 23rd International Conference on Process Control (PC)*, 2021, pp. 37-42. IEEE.
- [4] J. Zhang, A. K. Swain, and S. K. Nguang, "Robust Observer-Based Fault Diagnosis for Nonlinear Systems Using MATLAB", Springer, 2016.
- [5] A. Khan, W. Xie, B. Zhang, and L.-W. Liu, "A survey of interval observers design methods and implementation for uncertain systems", *Journal of the Franklin Institute*, vol. 358, no. 6, pp. 3077-3126, 2021.
- [6] MathWorks, "Data Acquisition Toolbox Documentation," MathWorks, 2024. [Online]. Available: <https://www.mathworks.com/help/daq/index.html>. [Accessed: Jul. 23, 2024].
- [7] MathWorks, "Observers - MATLAB and Simulink," MathWorks, 2024. [Online]. Available: <https://www.mathworks.com/help/control/ug/what-is-an-observer.html>. [Accessed: Jul. 23, 2024].
- [8] Niazi, M. U. B., Cao, J., Sun, X., Das, A., and Johansson, K. H., "Learning-based design of Luenberger observers for autonomous nonlinear systems," in *2023 American Control Conference (ACC)*, IEEE, 2023, pp. 3048-3055.
- [9] Augustine, M. T., "Model Predictive Control using MATLAB," *Papers With Code*, 2023. [Online]. Available: <https://math.paperswithcode.com/paper/model-predictive-control-using-matlab>. [Accessed: Jul. 23, 2024].

- [10] J. I. Venceslau de Souto, Á. Barbosa da Rocha, R. N. C. Duarte, and E. de Moura Fernandes, "Design and implementation of an embedded data acquisition system for vehicle vertical dynamics analysis", *Sensors*, vol. 23, no. 23, p. 9491, 2023.
- [11] T. Kapitaniak, S. A. Mohammadi, S. Mekhilef, F. E. Alsaadi, T. Hayat, and V.-T. Pham, "A new chaotic system with stable equilibrium: Entropy analysis, parameter estimation, and circuit design", *Entropy*, vol. 20, no. 9, p. 670, 2018.
- [12] J. H. Pérez-Cruz, J. M. Allende Peña, C. Nwachiona, J. de J. Rubio, J. Pacheco, J. A. Meda-Campaña, D. Ávila-González, O. Guevara Galindo, I. A. Romero, and S. I. Belmonte Jiménez, "A Luenberger-Like Observer for Multistable Kapitaniak Chaotic System," *Complexity*, vol. 2020, no. 1, pp. 9531431, 2020. doi: <https://doi.org/10.1155/2020/9531431>.
- [13] X. Chen, F. Zhao, Y. Liu, H. Liu, T. Huang, y J. Qiu, "Reduced-order observer-based preassigned finite-time control of nonlinear systems and its applications," *IEEE Trans. on Systems, Man, and Cybernetics: Systems\**, vol. 53, no. 7, pp. 4205-4215, Jul. 2023.

# CAPÍTULO 8

---

## ROBÓTICA HUMANOIDE

# Sistema Multimodal de Reconocimiento de Emociones Empleando la Interacción con el Robot Humanoide NAO

Vanessa Cristell Torres López  
División de Estudios de Posgrado  
Universidad Tecnológica de la Mixteca  
Huajuapán de León, Oaxaca, México  
cristelltl@gmail.com

José Anibal Arias Aguilar  
División de Estudios de Posgrado  
Universidad Tecnológica de la Mixteca  
Huajuapán de León, Oaxaca, México  
anibal@mixteco.utm.mx

Eduardo Sánchez Soto  
Instituto de Computación  
Universidad Tecnológica de la Mixteca  
Huajuapán de León, Oaxaca, México  
esanchez@mixteco.utm.mx

**Abstract**— Emotion recognition is an emerging area of research that is increasingly significant for achieving natural Human-Robot Interaction (HRI). This study presents the development of two unimodal and one bimodal emotion recognition systems using image and voice input signals. Initially, Convolutional Neural Networks (CNNs) were implemented for each input signal using existing databases for training, validation, and testing to optimize model performance. Subsequently, the unimodal systems' results were integrated to form a bimodal system that combined both input signals. Finally, the CNNs were retrained using a proprietary database generated through interactions with the NAO humanoid robot, aiming to capture more natural emotional data by minimizing the overacting of emotions in the robot's presence.

**Keywords**—CNN, NAO, Human-Robot Interaction, Automatic Emotion Recognition.

## I. INTRODUCCIÓN

El reconocimiento de emociones implica la identificación y clasificación de estados emocionales a partir de datos sensoriales. Las emociones se pueden manifestar de diversas formas, incluyendo expresiones faciales, tono de voz, gestos corporales y fisiología. Tradicionalmente este reconocimiento se ha llevado a cabo mediante técnicas de procesamiento de señales y aprendizaje automático que, aunque efectivas, enfrentan limitaciones en términos de escalabilidad y exactitud. El reconocimiento de emociones se efectúa a través de diferentes medios, tales como la voz y el reconocimiento de expresiones faciales [1]. Al reconocer la emoción presente en un usuario, los sistemas computacionales y robóticos pueden ajustar su funcionamiento.

Los métodos de reconocimiento de emociones basados en uso de Redes Neuronales Convolucionales (CNN), aprenden características directamente a partir de los datos sin necesidad de ingeniería manual. Esto no solo mejora la exactitud, sino que también permite una mejor escalabilidad y adaptación a nuevos datos. Las CNN pueden capturar patrones jerárquicos y complejos en los datos de entrada, lo que las hace especialmente

adecuadas para el reconocimiento de emociones en imágenes y señales de audio [2].

En [3], Obaid & Alrammahi presentan un sistema de reconocimiento de expresiones faciales que clasifica siete emociones: felicidad, tristeza, miedo, sorpresa, disgusto, ira y neutralidad utilizando una CNN híbrida. Este modelo combina una red CNN para procesar imágenes faciales estáticas y una red de creencias profundas (DBN) para integrar características espaciales y temporales. El modelo alcanzó un alto rendimiento de reconocimiento en las bases de datos JAFFE, KDEF y RaFD, con 98%, 95% y 98% de exactitud, respectivamente.

Muhammad [4], presenta un modelo CNN 3D con fusión de funciones múltiples al incorporar tres técnicas de extracción de características de audio distintas, como son los Coeficientes Cepstrales de Frecuencia Mel (MFCC), Chroma Shift y un espectrograma Mel. El modelo se probó y entrenó utilizando las bases de datos SUBESCO, con 8 emociones (aburrimiento, fatiga, confusión, frustración, alegría leve, tristeza leve, ansiedad leve y desprecio leve) y RAVDESS obteniendo exactitudes del 96 y 89%, respectivamente.

Pan [5], en su investigación, analiza el habla, expresiones faciales y electroencefalogramas (EEG) a través de un Reconocimiento Multimodal de Emociones (MER) basado en el aprendizaje llamado Deep-Emotion. El modelo propuesto contiene una rama por cada señal de entrada empleando el modelo GhostNet para la rama de las expresiones faciales, un algoritmo Lightweight Fully Convolutional Neural Network (LFCNN) para la señal del habla y un modelo Long Short-Term Memory (LSTM) en el caso de los EEG. Pan reporta una exactitud promedio de 96% en el modelo propuesto al utilizar las bases de datos CK+, Emo-DB y MAHNOB-HCI.

Así como las arquitecturas que emplean más de un algoritmo de Deep Learning para la clasificación de emociones en un tipo de señal de entrada muestran desempeños superiores a aquellos modelos que contemplan únicamente CNN, el MER supone una opción de mejora en la exactitud para la clasificación de emociones; es por ello que se propone el desarrollo de tres sistemas de reconocimiento de emociones basados en CNN que

consisten en el análisis de dos señales: voz e imagen, a fin de evaluar su desempeño y determinar si un mayor número de señales incrementa la exactitud del sistema.

## II. GENERACIÓN DE LA BASE DE DATOS EMO-MX-NAO CON LA INTERACCIÓN DEL ROBOT HUMANOIDE NAO

### A. Robot Humanoide NAO

El robot NAO, mostrado en la Fig. 1, desarrollado por la empresa francesa Aldebaran Robotics (ahora SoftBank Robotics), es un robot humanoide que ha sido ampliamente utilizado en diversos campos, incluyendo la educación, la investigación y el entretenimiento. Desde su lanzamiento en 2006, NAO ha destacado por su capacidad de interacción y su versatilidad, convirtiéndose en una herramienta clave para el estudio y desarrollo de la robótica social [6]



Fig. 1. Robot NAO (Tomado de [6])

### B. Club de teatro de la Universidad Tecnológica de la Mixteca

Para la generación de la base de datos emocional, se contó con el apoyo de los miembros del club de teatro, conformado por 9 integrantes: 5 hombres y 4 mujeres, con edades de 18 a 25 años, alumnos de la Universidad Tecnológica de la Mixteca que asisten a los talleres impartidos en el Centro de Actividades Culturales (CAC) de la misma institución educativa.

La creación del *dataset*, resultado de 12 horas de grabación segmentadas en tres sesiones de cuatro horas cada una dado el tiempo de descanso necesario para el correcto funcionamiento del robot, se dividió en dos partes, donde la primera contiene las expresiones faciales que los actores interpretan en la interacción con el robot NAO. Se obtuvo un conjunto de datos para expresiones faciales nombrado EMO-MX-EF-NAO creado utilizando la cámara superior del robot NAO y una cámara externa al mismo. EMO-MX-SP-NAO es el conjunto de datos de emociones en voz capturadas por un micrófono externo al robot.

### C. EMO-MX-EF-NAO

La generación de la base de datos EMO-MX-EF-NAO se realizó bajo el siguiente flujo de trabajo:

- Presentación del robot con el actor: NAO comienza la interacción presentando y describiendo la actividad.
- Emisión de frase por NAO: Por cada emoción básica NAO pronunció un total de 10 frases. El orden de representación de las emociones fue: tristeza, miedo, aversión, ira, sorpresa y alegría.

- Al final de cada frase, fueron tomadas cuatro fotografías consecutivas: dos con la cámara superior del robot y dos con una cámara externa.

La generación de la base de datos se efectuó bajo un ambiente controlado y con luz artificial, como se observa en la Fig. 2, a fin de minimizar las variables que puedan afectar el aprendizaje del sistema de reconocimiento de expresiones faciales.



Fig. 2. Interacción NAO-Actor

La base de datos EMO-MX-EF-NAO contiene un total de 1236 imágenes de expresiones faciales para las seis emociones básicas en blanco y negro con un tamaño 192 x 192 píxeles, donde se ha recortado únicamente el rostro a fin de eliminar información no relevante. La Tabla 1 muestra la distribución de la base de datos.

TABLA 1. DISTRIBUCIÓN DE IMÁGENES EN LA BASE DE DATOS EMO-MX-EF-NAO

Emoción	Entrenamiento	Validación	Pruebas
01- Enojo	145	31	30
02- Aversión	145	31	30
03- Miedo	145	31	30
04- Alegría	145	31	30
05- Tristeza	145	31	30
06- Sorpresa	145	31	30

La Fig. 3 ilustra algunas imágenes contenidas en la base de datos EMO-MX-EF-NAO bajo diferentes emociones.



Fig. 3. Imágenes contenidas en EMO-MX-EF-NAO

#### D. EMO-MX-SP-NAO

EMO-MX-SP-NAO representa la base de datos de audio generada a partir de la interacción con el robot NAO. A continuación, se enlistan las doce frases empleadas para esta base de datos. El conjunto de audios incluye cada frase grabada seis veces, una por cada emoción básica:

- Son las once en punto.
- Eso es exactamente lo que pasó.
- Estoy de camino a la reunión.
- Me pregunto de qué se trata esto.
- El avión está casi lleno.
- Quizás mañana haga frío.
- Compraré un nuevo despertador.
- Tengo una cita con el médico.
- No olvides una chaqueta.
- No ha llovido en varios días.
- La superficie está resbaladiza.
- Pararemos en un par de minutos.

La base de datos contiene en total 546 archivos. Los audios fueron grabados utilizando un micrófono externo al robot, bajo una frecuencia de muestreo de 48kHz, con una duración de 5 segundos cada uno.

La Tabla 2 muestra la distribución de los audios en la base de datos EMO-MX-SP-NAO.

TABLA 2. DISTRIBUCIÓN DE IMÁGENES EN LA BASE DE DATOS EMO-MX-SP-NAO

Emoción	Entrenamiento	Validación	Pruebas
01- Enojo	68	12	11
02- Aversión	68	12	11
03- Miedo	68	12	11
04- Alegría	68	12	11
05- Tristeza	68	12	11
06- Sorpresa	68	12	11

### III. METODOLOGÍA

Se presenta el desarrollo de los sistemas monomodales y multimodal a partir de una base de datos existente y la base de datos propia EMO-MX-NAO.

#### A. Experimento 1. Sistema Monomodal de Reconocimiento de Expresiones Faciales entrenado con una base de datos preexistente.

La Fig. 4 muestra las arquitecturas utilizadas para el reconocimiento de expresiones faciales: una red de tres vs una red de cinco capas convolucionales.

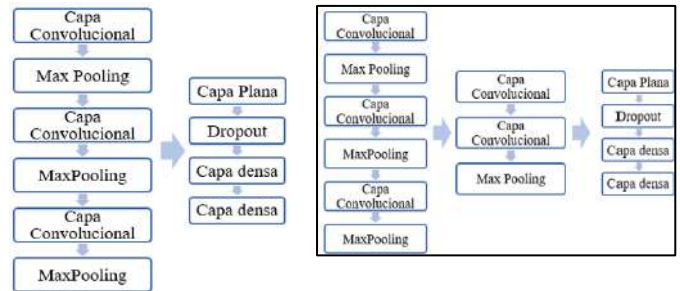


Fig. 4. Arquitecturas de: a) tres capas convolucionales y b) cinco capas convolucionales para el Reconocimiento de Expresiones Faciales.

Se fusionaron las bases de datos CK+ y FER2013 con un tamaño de imágenes de 48x48 pixeles, descartando la emoción desprecio, puesto que es una emoción compuesta por ira y aversión, para la clasificación de las seis emociones básicas: emoción, aversión, miedo, felicidad, tristeza y sorpresa [7][8]. La base de datos se revisó cuidadosamente a fin de descartar imágenes no correspondientes a rostros humanos o fuera del rango de edad (18, 25) años, dando como resultado la distribución de imágenes de la Tabla 3.

TABLA 3. DISTRIBUCIÓN DE IMÁGENES EN LA BASE DE DATOS CK+ Y FER2013

Emoción	Entrenamiento	Validación	Prueba
01- Enojo	400	100	100
02- Aversión	400	100	100
03- Miedo	400	100	100
04- Alegría	400	100	100
05- Tristeza	400	100	100
06- Sorpresa	400	100	100

Al entrenar la red neuronal de tres capas convolucionales con tamaño de lote de 16 imágenes se obtuvieron exactitudes de 59% en entrenamiento, 56% para validación y 59% en los datos de prueba. Por otro lado, el modelo CNN de cinco capas convolucionales fue entrenado con un tamaño de lote de 256 imágenes obteniendo resultados de 67% en entrenamiento, 64% en validación y 66% en imágenes de prueba. Las matrices de confusión de los modelos descritos, Fig. 5, muestran que ambos modelos clasifican mucho mejor las emociones felicidad y sorpresa que las clases miedo y tristeza.

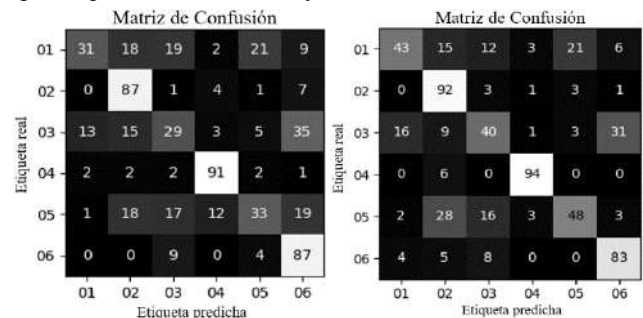


Fig. 5. Matriz de confusión para: a) Modelo de tres capas convolucionales b) Modelo de cinco capas convolucionales entrenado con la fusión de las bases de datos CK+ y FER2013

**B. Experimento 2. Sistema monomodal de Reconocimiento de Expresiones Faciales entrenado con la base de datos EMO-MX-EF-NAO**

Se entrenaron y evaluaron las dos arquitecturas CNN para la clasificación de expresiones faciales descritas en el experimento 1 a fin de comparar su desempeño con la base de datos EMO-MX-EF-NAO.

En el caso de la arquitectura de tres capas convolucionales con un tamaño de lote de 64 imágenes, se obtuvo una exactitud de 93%, 87% y 87% en entrenamiento, validación y prueba, respectivamente; mientras que para el modelo de cinco capas convolucionales, bajo un tamaño de lote de 128 imágenes las exactitudes obtenidas son de 88%, 84% y 84% en cada fase, respectivamente.

La Fig. 6 muestra las matrices de confusión de los modelos descritos. Se observa mejor balance en la clasificación en comparación con los resultados obtenidos con la base de datos CK+ y FER2013.

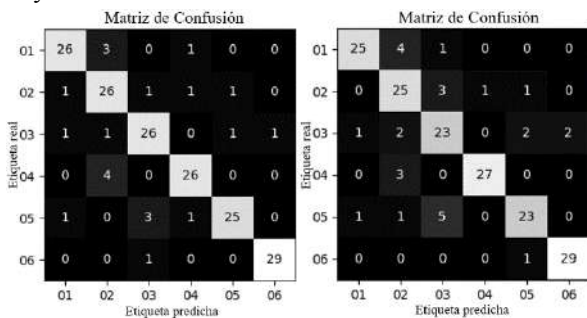


Fig. 6. Matrices de confusión para: a) Modelo de tres capas convolucionales y b) Modelo de cinco capas convolucionales entrenados con la base de datos EMO-MX-EF-NAO

**C. Sistema Monomodal de Reconocimiento de Emociones en Voz entrenado con la base de datos RAVDESS.**

Para la clasificación de las emociones mediante la voz, también se han implementado dos redes CNN, con las arquitecturas que se muestran en la Fig. 7.

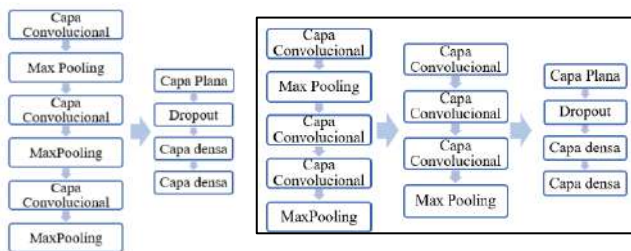


Fig. 7. Arquitecturas de CNN con: a) tres capas convolucionales y b) seis capas convolucionales para la clasificación de emociones en habla.

Para entrenar y evaluar el modelo, se utilizó el subconjunto *emotional speech* de la base de datos RAVDESS que cuenta con 1146 audios. Los datos fueron distribuidos tal como se muestra en la Tabla 4 de acuerdo a la regla 70-15-15.

TABLA 4. DISTRIBUCIÓN DE AUDIOS EN LA BASE DE DATOS RAVDESS

Emoción	Audios de entrenamiento	Audios de validación	Audios de prueba
01- Enojo	131	30	30
02- Aversión	131	30	30
03- Miedo	131	30	30
04- Alegría	131	30	30
05- Tristeza	131	30	30
06- Sorpresa	131	30	30

Cada archivo de audio pasó por una transformación a espectrogramas, utilizando funciones de la librería *librosa* para ingresar imágenes a la CNN. Los espectrogramas fueron recortados y redimensionados a 48x48 pixeles para conservar únicamente las zonas de color que puedan proporcionar información útil, por lo que se eliminó la descripción de color, el título de la figura y la descripción de los ejes.

La Fig. 8 muestra algunas de las imágenes que ingresan a la CNN de clasificación de emociones en la voz.

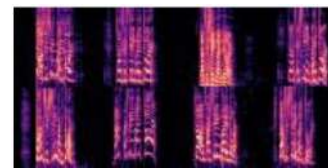


Fig. 8. Muestra de imágenes que ingresan a la CNN de reconocimiento en voz

Al entrenar la CNN de tres capas convolucionales con un tamaño de lote de 64 imágenes se obtuvieron exactitudes de 76%, 66% y 61% en entrenamiento, validación y prueba, respectivamente. El entrenamiento para el modelo de seis capas convolucionales para un tamaño de lote de 64 imágenes mostró exactitudes de 84%, 64% y 63% en cada fase. Las matrices de confusión de los modelos descritos, mostradas en la Fig. 9, indican una mejor clasificación de las emociones: enojo, aversión y miedo.

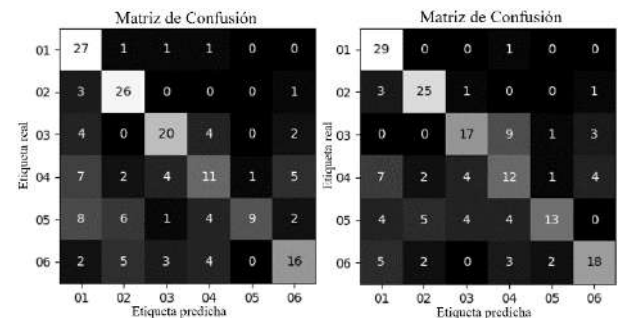


Fig. 9. Matriz de confusión para el Sistema Monomodal de Reconocimiento de Emociones en Voz con: a) tres capas convolucionales y b) seis capas convolucionales para la base de datos RAVDESS.

**D. Sistema Monomodal de Reconocimiento de Emociones en Voz entrenado con la base de datos EMO-MX-SP-NAO**

Se evaluaron a su vez, las dos arquitecturas CNN para reconocimiento de emociones en habla con la base de datos EMO-MX-SP-NAO.

Para la red CNN de tres capas convolucionales entrenada con un tamaño de lote de 32 imágenes se obtuvieron exactitudes de 58%, 54% y 48% en entrenamiento, validación y prueba respectivamente, mientras que para la red de seis capas convolucionales con 16 imágenes por lote en entrenamiento se obtuvieron desempeños de 77%, 52% y 56% en cada fase. Las matrices de confusión de los modelos descritos, mostradas en la Fig. 10, muestran que el modelo de tres capas convolucionales tiende a confundir las emociones: felicidad, enojo y sorpresa, mientras que el modelo de seis capas convolucionales confunde las clases: aversión y tristeza.

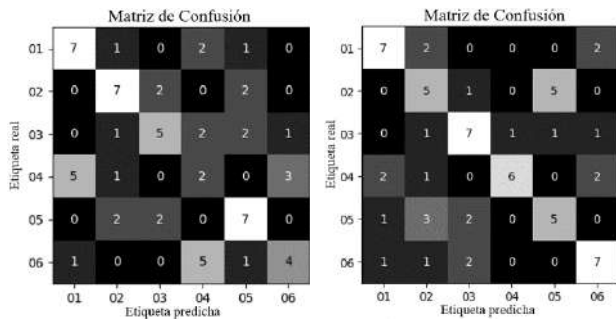


Fig. 10. Matriz de confusión para el Sistema Monomodal de Reconocimiento de Emociones en Voz con: a) tres capas convolucionales y b) seis capas convolucionales para la base de datos EMO-MX-SP-NAO.

### E. Sistema Multimodal de Reconocimiento de Emociones: Fusión en Bajo Nivel

Para la fusión en bajo nivel se optó por utilizar la misma arquitectura de CNN de tres capas convolucionales utilizada en el reconocimiento de expresiones faciales.

La fusión en bajo nivel implica combinar todas las características existentes antes de ingresar a la red neuronal [9] por lo que se unió cada espectrograma con una expresión facial de la misma clase. La Fig. 11 muestra la combinación de características a bajo nivel para cada una de las bases de datos utilizadas.

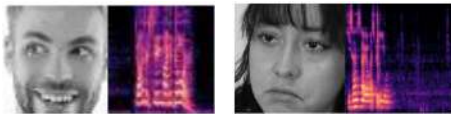


Fig. 11. Fusión de características a bajo nivel en a) CK+ FER2013 y RAVDESS y b) EMO-MX-NAO

La fusión de características a bajo nivel en la base de datos CK+ FER2013 y RAVDESS dio como resultado un *dataset* de 1146 imágenes de 96x48 píxeles, mientras que la fusión en EMO-MX-NAO resultó en un *dataset* de 546 imágenes de 384x192 píxeles.

El entrenamiento realizado bajo un tamaño de lote de 16 imágenes en el primer conjunto de datos dio como resultado 75% 72% y 66% de exactitud en entrenamiento, validación y prueba.

Para el segundo conjunto de datos, se obtuvieron exactitudes de 87%, 73% y 72% en cada fase, respectivamente.

Las matrices de confusión de los modelos, mostradas en la Fig. 12, muestran una clasificación más balanceada en la base de datos EMO-MX-NAO en comparación con la base de datos preexistente.

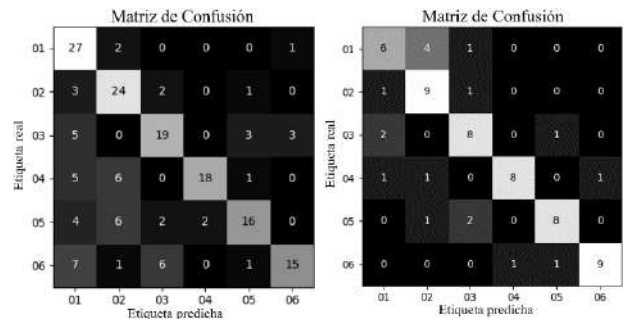


Fig. 12. Matrices de confusión para fusión en bajo nivel en: a) CK+ FER2013 y RAVDESS y b) EMO-MX-NAO

### F. Sistema multimodal de reconocimiento de emociones: Fusión en Nivel Medio

La Fig. 13 ilustra la arquitectura utilizada para la fusión de nivel medio en el sistema multimodal de reconocimiento de emociones. Para cada rama se define una sub-red de tres capas convolucionales y una capa densa cuyas salidas se concatenan y se convierten en la entrada a una tercera capa densa para finalmente pasar a la capa de clasificación.

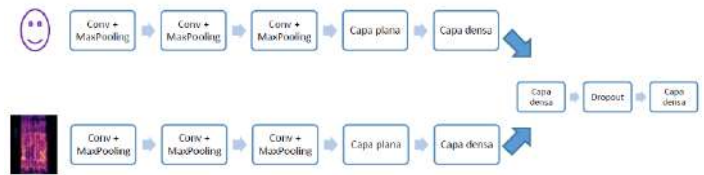


Fig. 13. Arquitectura del sistema multimodal de reconocimiento de emociones con fusión media

Las entradas de cada rama corresponden a las imágenes de las expresiones faciales y los espectrogramas utilizados en fusión de bajo nivel, pero sin combinar para poder ingresar correctamente a cada sub-red.

El entrenamiento realizado bajo un tamaño de lote de 64 imágenes en el conjunto de datos CK+ FER2013 y RAVDESS dio como resultado 78% 58% y 60% de exactitud en entrenamiento, validación y prueba. Mientras que el entrenamiento realizado en EMO-MX-NAO con tamaño de lote de 16 mostró resultados de 99% 83% y 82% en cada fase, respectivamente.

El entrenamiento realizado con la base de datos EMO-MX-NAO mostró mejor clasificación de emociones en comparación con las bases de datos preexistentes, tal como se muestra en las matrices de confusión de la Fig. 14.

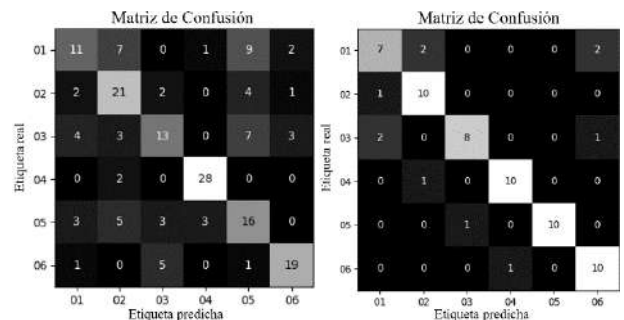


Fig. 14. Matrices de confusión para fusión en nivel medio en: a) CK+ FER2013 y RAVDESS y b) EMO-MX-NAO

G. Sistema Multimodal de Reconocimiento de Emociones:  
Fusión en Alto Nivel

La Fig. 15 ilustra la arquitectura seleccionada para el sistema multimodal de reconocimiento de emociones en fusión de alto nivel. Las constantes  $\alpha$  y  $\beta$  representan los porcentajes asignados a cada rama para la clasificación de emociones.



Fig. 15. Arquitectura de fusión de alto nivel

La Tabla 5 muestra los resultados obtenidos para fusión en alto nivel bajo diferentes valores de los coeficientes  $\alpha$  y  $\beta$ .

TABLA 5. EXACTITUD EN IMÁGENES DE PRUEBA PARA CONJUNTO PREEXISTENTE VS EMO-MX-NAO

$\alpha$	$\beta$	Exactitud de prueba conjunto preexistente	Exactitud de prueba conjunto EMO-MX-NAO
0.05	0.95	0.66	0.55
0.10	0.90	0.67	0.58
0.15	0.85	0.67	0.59
0.20	0.80	0.66	0.61
0.25	0.75	0.68	0.64
0.30	0.70	0.71	0.70
0.35	0.65	0.72	0.71
0.40	0.60	0.74	0.80
0.45	0.55	0.79	0.83
0.50	0.50	0.81	0.88
0.55	0.45	0.81	0.91
0.60	0.40	0.81	0.91
0.65	0.35	0.80	0.90
0.70	0.35	0.78	0.91
0.75	0.25	0.78	0.88
0.80	0.20	0.76	0.88
0.85	0.15	0.76	0.86
0.90	0.10	0.75	0.86
0.95	0.05	0.73	0.85

IV. CONCLUSIONES

El desarrollo de esta investigación permitió demostrar que las emociones pueden clasificarse con porcentajes de exactitud altos, proporcionales al número de señales analizadas para la clasificación y la calidad de las muestras contenidas en la base de datos.

Las expresiones faciales mostraron un porcentaje de clasificación superior al utilizar un modelo entrenado con la base de datos generada a partir de la interacción con el robot humanoide NAO, lo que es atribuible al entorno controlado donde se tomaron las fotografías, el ambiente de confianza que generó el robot y la dimensionalidad de las imágenes en comparación con las bases de datos preexistentes.

En contraste, los modelos de clasificación de emociones en voz muestran mejores resultados con la base de datos RAVDESS en comparación con la base de datos EMO-MX-SP-NAO, sin embargo, no puede realizarse una comparación válida que indique los parámetros que influyen en la exactitud obtenida con la base de datos generada debido a los idiomas diferentes bajo los que fueron creados los *datasets*.

De forma general, la clasificación de emociones muestra mejores resultados cuando se realiza sobre las expresiones faciales en comparación con los análisis realizados para las emociones en el habla.

El desempeño de los modelos bajo los tres niveles de fusión es más alto al ser entrenado y evaluado con la base de datos EMO-MX-NAO, lo cual indica una alta calidad en el *dataset* generado.

El nivel de fusión que presenta mejores resultados en exactitud en el conjunto de prueba es el nivel alto, donde se encuentra la exactitud más alta obtenida durante toda la investigación: 91%.

Como trabajo a futuro se plantea la incorporación de señales fisiológicas, ritmo cardíaco, entre otras, en la clasificación de emociones y la incorporación del modelo bimodal a la interacción humano robot.

REFERENCIAS

- [1] C. Breazeal, *Designing Sociable Robots*. MIT Press, 2004.
- [2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015, doi: 10.1038/nature14539.
- [3] A. J. Obaid and H. K. Alramahi, "An Intelligent Facial Expression Recognition System Using a Hybrid Deep Convolutional Neural Network for Multimedia Applications," *Applied Sciences*, vol. 13, no. 21, p. 12049, 2023, doi: 10.3390/app132112049.
- [4] M. Rahaman, "A 3D CNN Model with Multi-Feature Fusion for Enhancing Human Emotion Recognition from Speech," 2024.
- [5] J. Pan, W. Fang, Z. Zhang, B. Chen, Z. Zhang, and S. Wang, "Multimodal Emotion Recognition based on Facial Expressions, Speech, and EEG," *IEEE Open Journal of Engineering in Medicine and Biology*, advance online publication, 2023, doi: 10.1109/OJEMB.2023.3240280.
- [6] SoftBank Robotics, "NAO the Humanoid and Programmable Robot," SoftBank Robotics, n.d. [Online]. Available: <https://www.softbankrobotics.com/emea/en/nao>. [Accessed: Jun. 30, 2023].
- [7] Asociación Española contra el Cáncer, *Las emociones, comprenderlas para vivir mejor*. Guías y protocolos. Madrid, España: Asociación Española contra el Cáncer, 2010.
- [8] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *Journal of Personality and Social Psychology*, vol. 17, no. 2, pp. 124-129, 1971, doi: 10.1037/h0030377.
- [9] X. Wang and A. Gupta, "Videos as space-time region graphs," presented at the European Conference on Computer Vision (ECCV), 2018.

# Órtesis para rehabilitación de la mano impresa en 3D

Valentina Islas-Gonzalez  
*Instituto Politécnico Nacional*  
*ESIME Unidad Azcapotzalco*  
Ciudad de México, México  
vislasg2100@alumno.ipn.mx

Gonzalo Solano-Ruiz  
*Instituto Politécnico Nacional*  
*ESIME Unidad Zacatenco*  
Ciudad de México, México  
gsolanor2100@alumno.ipn.mx

Angel D. Hernández-Lecourtois  
*Instituto Politécnico Nacional*  
*UPIBI*  
Ciudad de México, México  
ahernandezl2105@alumno.ipn.mx

Axel Cid-Trejo  
*Instituto Politécnico Nacional*  
*ESIME Unidad Azcapotzalco*  
Ciudad de México, México  
acidt2100@alumno.ipn.mx

Noemi Hernández-Oliva  
*Instituto Politécnico Nacional*  
*CECyT 2 Miguel Bernard*  
Ciudad de México, México  
nhernandezo@ipn.mx

Marisol Alejandre-Flores  
*Instituto Politécnico Nacional*  
*CECyT 2 Miguel Bernard*  
Ciudad de México, México  
malejandref@ipn.mx

**Resumen**—En este trabajo se presenta el diseño y construcción de una órtesis para rehabilitación de la mano, basada en el registro de señales electromiográficas (EMG) y la tecnología de imitación mediante un guante con sensores flex. La órtesis, fabricada con PETG mediante manufactura aditiva, replica los movimientos de una mano sana en la otra, con lo que se espera que el paciente tenga una rehabilitación más rápida y menos agotadora. Durante su uso, se registran las señales EMG generadas por la actividad muscular de la mano, con el fin de evaluar el desempeño del sujeto al utilizar la órtesis, recavando información relevante para un posible tratamiento. El diseño modular y de fácil mantenimiento permite que el dispositivo sea accesible y efectivo. Como trabajo a futuro, se iniciara la siguiente fase del proyecto, mejorando la precisión del sistema y la comodidad de los pacientes.

**Index Terms**—órtesis, rehabilitación, imitación, impresión 3D, PETG

## I. INTRODUCCIÓN

Desde hace algunos años, se han desarrollado técnicas para mejorar las patologías de pacientes con lesiones en cualquier área del cuerpo, las cuales han sido combinadas con el uso de la tecnología con el propósito de diseñar dispositivos para la rehabilitación o la sustitución de algún miembro. En ese sentido, algunos trabajos reportados, como [1] que presenta un estudio de diez años del uso de la bio-robótica en el área. En [2] se enfoca en los clasificadores de control microeléctrico utilizados en manos protésicas. En [3], se expone el estado actual del arte de una mano protésica mioeléctrica.

Por otro lado, también se han elaborado prótesis, órtesis y exoesqueletos que, combinados con diversas técnicas, presentan una amplia gama de desarrollos utilizados en la rehabilitación o como dispositivos de apoyo en pacientes

que perdieron algún miembro. En lo que respecta a órtesis, los trabajos siguientes fueron de interés para la propuesta diseñada. En [4] se presentan una órtesis que cumple dos funciones: la de dispositivo de asistencia y la de rehabilitación. En [5], el propósito es transportar una amplia gama de objetos y reproducir el movimiento de la extremidad superior de la humana. En [6] se presenta un guante de asistencia portátil para pacientes que padecen accidentes cerebro vasculares o con graves discapacidades en las manos. En [7] se desarrolló un guante robótico suave que puede ser utilizado en entornos clínicos y domésticos en un estudio piloto. En [8] elaboraron un guante de agarre denominado Hand Extension Robot Orthosis, HERO, con el propósito de satisfacer las solicitudes de terapeutas y pacientes que han sufrido accidentes y padecen graves deficiencias en sus manos. En [9] llevan a cabo una revisión de órtesis activas de la mano, enfocándose en el diseño, funcionalidad y aplicaciones clínicas. Para [10] se presenta el desarrollo de una órtesis activa de muñeca para pacientes con Parkinson. Finalmente, en [11] se presenta el diseño y construcción de un dispositivo para rehabilitación tipo órtesis.

Se diseñó y construyó un prototipo de órtesis destinado a rehabilitar el miembro superior mediante la lectura de señales electromiográficas (EMG) para este trabajo, el cual incluye un guante con sensores flex y el uso de la terapia de imitación, a bajo costo y modular. La organización del documento es la siguiente: sección 2 presenta con detalle las características del prototipo impreso en 3D, la Sección 3 presenta la implementación de la lectura de las señales EMG durante la terapia de rehabilitación; la Sección 4 presenta las pruebas realizadas con el prototipo realizadas en cinco sujetos, y, por último, en la Sección 5 se presentan las conclusiones y trabajos a futuro.

## II. PROTOTIPO IMPRESO EN 3D

Con el fin de llevar a cabo el diseño de la órtesis que integra el prototipo, se optó por una estructura que involucre adecuadamente una mano lesionada, sin comprometer la integridad del usuario. Por consiguiente, es un requisito indispensable un buen ajuste, de lo contrario, el prototipo debe ser adaptable a posibles modificaciones, como las que se centran en la sustitución de componentes, la reubicación de los servomotores o la fuente de alimentación; así como el ajuste del posicionamiento de los electrodos para la obtención de las señales electromiográficas (EMG). Como resultado, el diseño se enfoca en la capacidad de soportar esos cambios, así como los movimientos bruscos provocados inicialmente por los servomotores, la tensión y fricción de los hilos que producen el movimiento, así como el desgaste propio por el uso continuo de la órtesis por el usuario.

### II-A. Diseño mecánico de la órtesis

Una vez que se identificaron los requerimientos para el diseño de la órtesis, se optó por uno de tipo exoesqueleto en forma de manopla que conecta los puntos de articulación de los dedos mediante la unión de protuberancias y huecos posicionados estratégicamente para asegurar un ajuste perfecto. Con el fin de alcanzar esta meta, se empleó la manufactura aditiva de impresión en 3D, ya que el diseño presentaba una complejidad de maquinari. Dicho diseño final se exhibe en la Fig. 1, el cual fue modelado en el software Solid Edge 2023 versión estudiantil.



Figura 1. Diseño de la órtesis realizado en Solid Edge 2023

La manufactura aditiva se realizó utilizando material PETG, debido a sus propiedades principales, tales como alta resistencia a impactos y durabilidad. Este material es más resistente que otros en el mercado, como el PLA, y es más flexible que el ABS, el cual es otro material utilizado en impresión. Esto lo convierte en una opción ideal para piezas funcionales sometidas a esfuerzos, como la falange del dedo mostrada en la Fig. 2.



Figura 2. Diseño de la falange del dedo realizado en Solid Edge 2023

Otra ventaja significativa del PETG en comparación con otros materiales, es la flexibilidad y dureza, lo cual reduce el

tiempo de producción a únicamente la velocidad a la que funciona la impresora 3D utilizada en el proceso de manufactura. Además, es un material con buena tolerancia a la humedad, lo que hace que no sea necesario aplicarle algún tratamiento químico adicional. Por otro lado, el mantenimiento de la órtesis se limita a dos aspectos fundamentales: la limpieza y la revisión de los componentes y piezas. La limpieza es primordial debido a la transpiración y fluidos del usuario durante su uso, principalmente en la punta del dedo, ya que es una pieza cerrada como se indica en Fig. 3.



Figura 3. Diseño de la yema del dedo realizado en Solid Edge 2023

A pesar de ser un material resistente, el PETG es un polímero, por lo que las piezas delgadas como el nudillo de la órtesis modelada (ver Fig. 4) pueden llegar a sufrir fracturas si se utiliza de manera inadecuada por un tiempo prolongado. No obstante, debido a su estructura modular, si se requiere reemplazar una pieza dañada, es sustituida por una nueva, lo cual es factible debido al diseño modular de la órtesis, dado que se pueden armar y desarmar las piezas en los puntos de articulación.



Figura 4. Diseño del nudillo realizado en Solid Edge 2023

### II-B. Electrónica del prototipo

El sistema electrónico de la órtesis está compuesto por 3 tarjetas ESP32 que trabajan de manera conjunta para procesar tanto los movimientos de la mano con correcta movilidad como las señales electromiográficas (EMG) del paciente; además de controlar los servomotores que mueven la órtesis. Cada tarjeta tiene una función específica, y su interacción es clave para el funcionamiento de la órtesis.

*II-B1. Primera ESP32:* La primera tarjeta se encarga de registrar las señales provenientes de cuatro sensores flex ubicados en un guante de imitación que se coloca en la mano con correcta movilidad del paciente. Estos sensores miden la flexión de los dedos y generan señales eléctricas que reflejan los movimientos; luego estos datos son enviados a la segunda tarjeta ESP32 por medio de una conexión Wifi, como se muestra en la Fig. 5 a continuación.

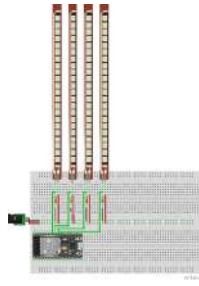


Figura 5. Primer circuito realizado en Fritzing 2023

*II-B2. Segunda ESP32:* La segunda tarjeta es responsable de procesar los datos recibidos de la primera tarjeta y controlar los servomotores MG996R conectados a la órtesis. A partir de las lecturas de los sensores flex, esta tarjeta genera señales que provocan el movimiento de los servomotores, permitiendo con esto que los movimientos de la mano con correcta movilidad se repliquen en la mano lesionada, que lleva la órtesis, ilustrado en la Fig. 6.

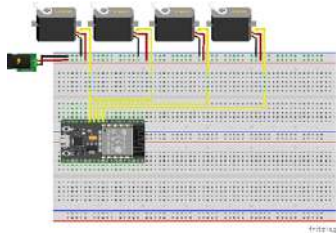


Figura 6. Segundo circuito realizado en Fritzing 2023

*II-B3. Tercera ESP32:* La tercera tarjeta está conectada al sensor EMG, que emplea tres electrodos colocados en el brazo del paciente para captar las señales electromiográficas. Estas señales reflejan la actividad muscular generada cuando el paciente intenta mover su mano. Posteriormente, la tarjeta procesa dichas señales y las proyecta de forma gráfica en una pantalla TFT 7789, lo que permite a un especialista visualizar la respuesta muscular del paciente durante su proceso de rehabilitación, el circuito electrónico se muestra en la Fig. 7.

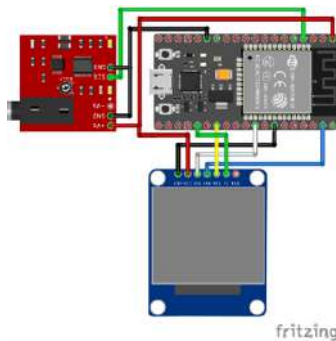


Figura 7. Tercer circuito realizado en Fritzing 2023

### II-C. Descripción de los actuadores

En el diseño de la órtesis se implementaron cuatro servomotores de alto torque modelo MG996R, seleccionados por su

habilidad para proporcionar la fuerza y precisión necesarias para mover las articulaciones de la mano durante la rehabilitación. Estos servomotores están equipados con engranajes metálicos, lo que proporciona una mayor durabilidad y resistencia al desgaste, en la Tabla I se describen sus características físicas. Debido a estas características, resultan idóneas para su implementación en el prototipo, ya que permiten controlar las fuerzas, movimientos y rotaciones necesarias para replicar los movimientos naturales de la mano, tales como la flexión y extensión de los dedos.

Tabla I  
CARACTERÍSTICAS DE LOS ACTUADORES

Característica	Descripción
Modelo	MG996R
Torque máximo	11kgfcm a 6V
Velocidad de operación	0.14s/60° a 6V
Rango de rotación	120° (60° en cada dirección)
Voltaje de operación	4.8V - 7.2V
Corriente de funcionamiento	500mA a 6V
Corriente de parada	2.5A a 6V
Engranajes	Metálicos
Peso	55g
Dimensiones	40.7x19.7x42.9mm

### III. IMPLEMENTACIÓN DE LA LECTURA DE LAS SEÑALES EMG

Las señales electromiográficas (EMG por sus siglas en inglés); son representaciones gráficas de la actividad eléctrica producida durante la contracción o relajación de los músculos esqueléticos. Estas señales se producen cuando las neuronas del cerebro transmiten impulsos eléctricos a las fibras musculares, lo que provoca su contracción. Las fibras musculares se encuentran conectadas entre sí y con el sistema nervioso mediante señales eléctricas. Cuando una neurona motora produce un impulso a un músculo, este lo recibe y lo traduce en una contracción muscular. La electromiografía permite registrar estas señales eléctricas, lo que proporciona una representación clara y detallada de la actividad muscular que se está produciendo. Esto permite evaluar la función y coordinación de los músculos, [12].

El desarrollo de la órtesis de rehabilitación se inicia mediante la colocación del guante de imitación en la mano sana del paciente, el cual cuenta con sensores flex que capturan los movimientos naturales de la mano. En la otra mano se ubica la órtesis y los electrodos necesarios para el registro de las señales EMG. Una vez colocado el prototipo, el paciente realiza movimientos con su mano sana; transmitiendo datos desde el guante a la órtesis. Los movimientos del guante se replican en la mano lesionada del paciente, para ello los servomotores se encargan de mover la órtesis como una respuesta a los datos enviados por el guante de imitación. Además, durante el movimiento replicado, se toman las señales EMG con los electrodos colocados en la mano lesionada, para medir la actividad eléctrica de los músculos. Esta información se requiere para el seguimiento del proceso de rehabilitación del paciente, como se observa en la Fig. 8 siguiente.

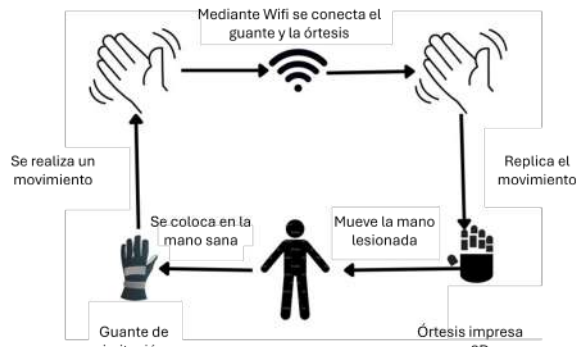


Figura 8. Diagrama del funcionamiento de la órtesis

**Proceso de obtención de las señales:** Se utilizan electrodos superficiales que se pueden colocar sobre la piel (electrodos de superficie) o colocados dentro del músculo (electrodos de aguja) para obtener las señales EMG. Los dos permiten registrar la actividad eléctrica generada por las contracciones musculares y convertirlas en una señal que pueda ser evaluada posteriormente por un especialista. La electromiografía de superficie es la técnica empleada en la órtesis debido a que los electrodos de superficie no son invasivos y no requieren ser colocados por un especialista. El proceso de registro de las señales requiere diversos pasos importantes que son los siguientes:

- **Preparación de la piel:** La piel debe ser limpiada adecuadamente para eliminar cualquier impureza que pueda interferir con la calidad de la señal.
- **Colocación de los electrodos:** Para el registro de las señales se utilizan tres electrodos. Dos se colocan en los puntos motores, cubital anterior y palmar menor. El tercer electrodo, el de referencia o electrodo de tierra, debe colocarse lo más lejos posible de los otros dos, por lo general sobre un hueso palpable. En el caso de la órtesis, este se coloca sobre el codo, como se aprecia en la Fig. 9.

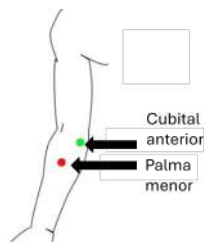


Figura 9. Puntos motores utilizados

- **Registro de las señales:** Una vez que los electrodos se han colocado correctamente, se registran las señales EMG para que posteriormente sean analizadas por un especialista.

**Análisis de las señales** El análisis adecuado de las señales puede proporcionar una gran cantidad de información sobre la actividad muscular. Los parámetros más habituales de las

señales EMG son la amplitud, que indica la intensidad de activación muscular, y la frecuencia que nos proporciona la información sobre la fatiga muscular y el tipo de fibras activadas.

### III-A. Pruebas de validación

Para la validación del prototipo, se realizó una prueba a cinco personas con diferentes características antropométricas, estas muestras no son exclusivas de pacientes con alguna lesión dado que es la primera evaluación del prototipo; sin embargo, si fue posible evaluar la adaptabilidad de la órtesis en cada uno de los sujetos. La prueba consistió en cerrar los dedos de la mano, en la secuencia de meñique, anular, medio e índice, uno por uno y finalizar con el cierre simultáneo de todos los dedos. El objetivo en esta etapa, consiste en evaluar la adaptación de la órtesis a diferentes tipos de manos de acuerdo con el género, tamaño y edad; así como verificar la medición de las señales EMG en los participantes. Los sujetos participantes son los siguientes:

- El sujeto uno, hombre de 50 años, refiere que la órtesis ajustaba bien, ver Fig. 10. Las señales EMG mostraron que su mano está sana, ya que los picos son estables.



Figura 10. Registro EMG en hombre de 50 años de edad.

- Sujeto dos, mujer de 48 años, refiere buen ajuste de la órtesis, ver Fig. 11. En las señales EMG, se observó estabilidad, indicando una mano sana.



Figura 11. Registro EMG en mujer de 48 años de edad.

- Sujeto tres, niña de 14 años, refiere que el tamaño no es el adecuado para su tamaño de mano, ver Fig. 12. En las señales EMG, los picos son muy pequeños, mostrando una inestabilidad en comparación con las otras personas. Esto podría deberse a que la órtesis no se ajusta bien a su mano.



Figura 12. Registro EMG en niña de 14 años de edad.

- Sujeto cuatro, mujer de 53 años, refiere que la órtesis le quedaba bien, ver Fig. 13. Las señales EMG mostraron una gráfica con picos estables, lo que indica que su mano se encuentra sana.



Figura 13. Registro EMG en mujer de 53 años de edad.

- El sujeto cinco, hombre de 68 años, refiere que la órtesis le quedaba bien, ver Fig. 14. Las señales EMG mostraron picos no débiles, lo que indica que su mano no es tan sana en comparación con las personas anteriores.



Figura 14. Registro EMG en hombre de 68 años de edad.

### III-B. Análisis de resultados

La prueba realizada a cinco personas de diferentes edades reveló observaciones importantes sobre la adaptación de la órtesis y la medición de las señales EMG:

1. Ajuste de la órtesis: La órtesis mostró variaciones en el ajuste según la edad y el tamaño de la mano de los participantes. Esto sugiere la necesidad de personalizar la órtesis para diferentes tamaños de mano.
2. Señales EMG: Las señales EMG variaron entre los participantes. Las personas de 50 (ver Fig. 15) y 53 (ver Fig. 18) años mostraron estabilidad en las señales, indicando manos sanas. Mientras que, en la niña de 14 años (ver Fig. 17), se presentó inestabilidad, ocasionada por la falta de ajuste. Con la señora de 48 años (16), se observaron señales menos consistentes, lo que sugiere que el ajuste de la órtesis podría no haber sido el adecuado para ella y el señor de 68 años (ver Fig. 19), mostró picos bajos y altos así como inestabilidad, evidenciando el deterioro que ha sufrido su mano.

3. Impacto de la Edad y el tamaño de la Mano: La edad y el tamaño de la mano influyen en la adaptación de la órtesis y en la calidad de las señales EMG. La persona más joven y los mayores tuvieron más dificultades con el ajuste, lo que indica la necesidad de considerar estos factores en el diseño de la órtesis.

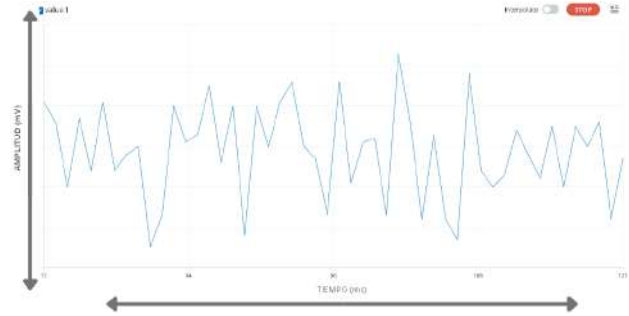


Figura 15. Registro EMG en hombre de 50 años de edad.

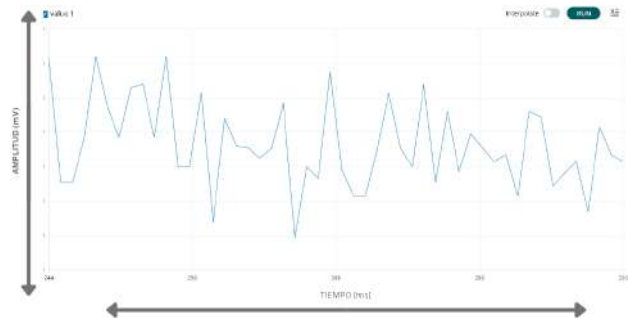


Figura 16. Registro EMG en mujer de 48 años de edad.

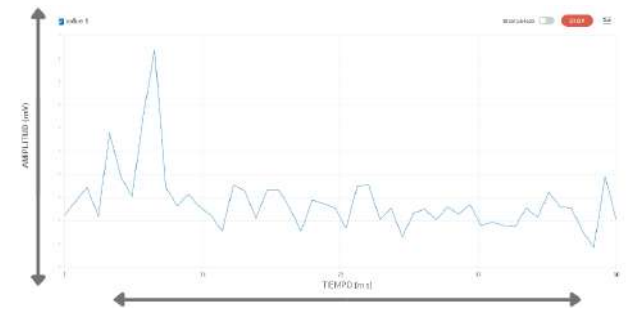


Figura 17. Registro EMG en niña de 14 años de edad.

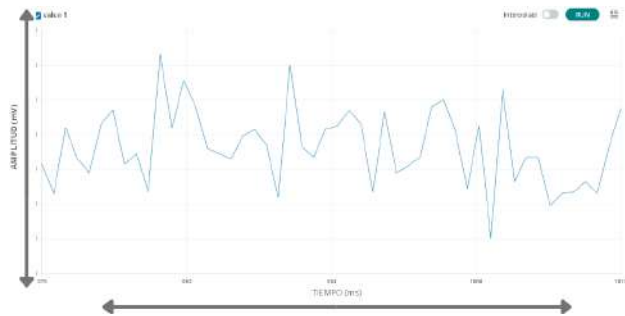


Figura 18. Registro EMG en mujer de 53 años de edad.

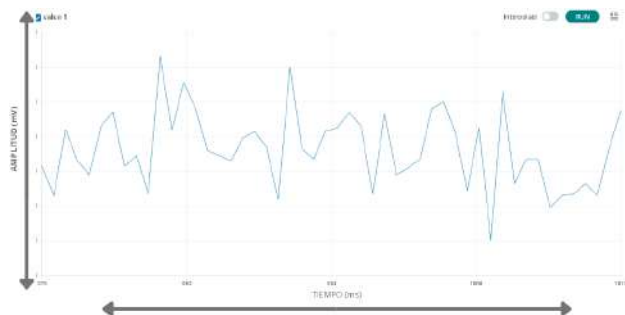


Figura 19. Registro EMG en hombre de 68 años de edad.

En resumen, la prueba destaca la importancia de ajustar la órtesis de manera personalizada para cada individuo, considerando factores como la edad y el tamaño de la mano, para asegurar tanto un buen ajuste como una medición precisa de las señales EMG, para así poder determinar un resultado, que indique que la mano se encuentra en un estado bueno regular o malo, con respecto a su salud.

#### IV. CONCLUSIONES

Se ha llevado a cabo la creación de una órtesis diseñada con el propósito de facilitar la rehabilitación de pacientes con movilidad articular reducida en la mano como consecuencia de una lesión o accidente, mediante la técnica de imitación. Esto implica que una mano con movimiento libre y fluido pueda efectuar movimientos para que la mano que sufre una lesión pueda realizar una terapia de rehabilitación y observar su progreso, apoyándose en el monitoreo de las señales electromiográficas (EMG), con el fin de identificar el progreso de recuperación que el paciente experimenta en el proceso de rehabilitación.

La propuesta final de la órtesis se realizó una amplia investigación en diversos artículos científicos, esto permitió comprender más a fondo las necesidades que se tienen dentro de la rehabilitación.

El proyecto se encuentra en proceso de desarrollo, por lo tanto, se busca mejorar el diseño de la órtesis mediante ajustes en su forma y tamaño que resulten más confortables para cualquier paciente. Asimismo, se pretende que la visualización de las señales sea más práctica para el especialista y el monitoreo pueda guardarse como un archivo. Además, es

posible utilizar otro tipo de material para la manufactura de la órtesis. En general, las adaptaciones se enfocan en brindar mayor comodidad y mejorar la calidad de vida del paciente durante el uso de la órtesis de rehabilitación.

#### AGRADECIMIENTOS

Todos los autores agradecen al Instituto Politécnico Nacional el apoyo por medio de los proyectos SIP20242197 y SIP20242569. El primero, segundo y cuarto autor agradecen la beca BEIFI otorgada.

#### REFERENCIAS

- [1] N. Jiang, C. Chen, J. He, J. Meng, L. Pan, S. Su, X. Zhu, "Bio-robotics research for non-invasive myoelectric neural interfaces for upper-limb prosthetic control: a 10-year perspective review," *National Science Review*, Volume 10, Issue 5, May 2023, nwd048, <https://doi.org/10.1093/nsr/nwad048>
- [2] C. Ziming, H. Min, D. Wang, Z. Xia, F. Sun, and B. Fang. "A Review of Myoelectric Control for Prosthetic Hand Manipulation." *Biomimetics* 8, no. 3:328. <https://doi.org/10.3390/biomimetics8030328>
- [3] C. Meeker, S. Park, L. Bishop, J. Stein and M. Ciocarlie, "EMG pattern classification to control a hand orthosis for functional grasp assistance after stroke," 2017 International Conference on Rehabilitation Robotics (ICORR), London, UK, 2017, pp. 1203-1210, doi: 10.1109/ICORR.2017.8009413.
- [4] P. Geethanjali. "Myoelectric control of prosthetic hands: state-of-the-art review," *Med Devices (Auckl)*. 2016 Jul 27;9:247-55. doi: 10.2147/MDER.S91102. PMID: 27555799; PMCID: PMC4968852.
- [5] L. He, C. Xiong, K. Zhang, "Mechatronic Design of an Upper Limb Prosthesis with a Hand," In: Zhang, X., Liu, H., Chen, Z., Wang, N. (eds) *Intelligent Robotics and Applications*. ICIRA 2014. *Lecture Notes in Computer Science()*, vol. 8917. Springer, Cham. [https://doi.org/10.1007/978-3-319-13966-1\\_6](https://doi.org/10.1007/978-3-319-13966-1_6)
- [6] H. C. Fischer, k. Triandafilou, K. Thielbar, J. Ochoa, E. Lazzaro, "Use of a Portable Assistive Glove to Facilitate Rehabilitation in Stroke Survivors With Severe Hand Impairment," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 3, pp. 344-351, March 2016, doi: 10.1109/TNSRE.2015.2513675.
- [7] L. Cappello, J. T. Meyer, K. C. Galloway, J. D. Peisner, R. Granberry, D. A. Wagner, S. Engelhardt, S. Paganoni, C. J. Walsh, "Assisting hand function after spinal cord injury with a fabric-based soft robotic glove," *J Neuroeng Rehabil*. 2018 Jun 28;15(1):59. doi: 10.1186/s12984-018-0391-x. PMID: 29954401; PMCID: PMC6022347.
- [8] A. Yurkewich, I. Kozak, D. Hebert, R. Wang, A. Mihailidis, "Hand Extension Robot Orthosis (HERO) Grip Glove: enabling independence amongst persons with severe hand impairments after stroke," *J Neuroeng Rehabil*. 2020 Feb 26;17(1):33. doi: 10.1186/s12984-020-00659-5. PMID: 32102668; PMCID: PMC7045638.
- [9] Asadi Dereshgi H.i, Demir D, Yilmaz S., Abderrahmane A, Abderrahmane B "Advancements in Active Dynamic Orthoses: A Comprehensive Review of Hand Muscle Rehabilitation Strategies", *Join SSR*, 4(2), pp. 177-186. doi: 10.58769/joinssr.1391623.
- [10] Andrade A. O., Bourget J., Costa S., Pereira A., Okereke M.I., Vieira M.F., "Ergonomic Assessment of an Active Orthosis for the Rehabilitation of Flexion and Extension of Wrist". *World Congress on Medical Physics and Biomedical Engineering 2018*. IFMBE Proceedings, 68(2). Springer, Singapore.
- [11] Vaca Benitez L. M., Tabie M., Will N., Schmidt S., Jordan M., "Exoskeleton Technology in Rehabilitation: Towards an EMG-Based Orthosis System for Upper Limb Neuromotor Rehabilitation". *Journal of Robotics*, 2013, pág. 13. doi: 10.1155/2013/610589
- [12] L. Gilal, A. Malanda, I. Rodríguez Carreño, J. Rodríguez Falces, J. Navallas "Electromyographic signal processing and analysis methods", *Libro SUPLEMENTO*. 2009. vol. 3, no. 32, 2009, pp. 27-43

# CAPÍTULO 9

---

## MONITOREO Y ADQUISICIÓN DE DATOS

# Monitoreo IoT mediante Raspberry PI

Rene Sanchez Linares  
Sección de Estudios de Posgrado  
e Investigación de ESCOM  
Instituto Politecnico Nacional  
Ciudad de México, México

Romero-Herrera Rodolfo  
Sección de Estudios de Posgrado  
e Investigación ESCOM  
Instituto Politecnico Nacional  
Ciudad de México, Mexico

*Abstract*— Este trabajo detalla un sistema de monitoreo de habitaciones utilizando el Internet de las Cosas (IoT). Su finalidad es asistir al usuario a gestionar su habitación de manera automática e inalámbrica mediante una aplicación móvil. Este trabajo emplea una Raspberry Pi 4 para conectarse con la plataforma en la nube cloud firestore y la aplicación móvil diseñada en Visual Studio Code. El sistema recopila la temperatura, la humedad, la luz, si hay movimiento en la habitación, y ejecuta la acción que se desea, mientras que los datos de los sensores se guardan en la base de datos. En la aplicación móvil, se visualizan los datos de los sensores en tiempo real con fecha y hora. El sistema hará sonar el buzzer cuando la temperatura sea alta, uno de los leds se encenderá en caso de que la luz en la sala sea insuficiente, y el otro led se activara al detectar movimiento humano. El usuario puede actualizar la información de los sensores y valores que se muestran en la interfaz de la aplicación móvil al presionar el botón que aparece en la parte inferior izquierda.

*Keywords*— *Raspberry pi, IoT, sensor, monitoreo de habitaciones*

## I. INTRODUCTION

El Internet de las Cosas, IoT (Internet of things), ha recibido varias designaciones, pero se puede definir como una red mundial de dispositivos inteligentes que pueden “percibir” e interactuar con su entorno utilizando la red para su comunicación e interacción con usuarios y otros sistemas. El concepto del Internet de las Cosas sugiere que, en vez de tener un pequeño grupo de dispositivos de computación muy eficientes como computadoras portátiles, teléfonos avanzados, tabletas, etc., se podría tener un gran grupo de aparatos u objetos que sean “menos eficientes”, objetos como un par de zapatillas, un reflejo, un paraguas, un brazalete, una butaca, etc., que vigilen, registren y procesen datos relacionados de lo que ocurre a su alrededor, y su interacción con el entorno, etc. Esta información puede tener un mayor efecto en nuestras vidas que el que normalmente se obtendría con los dispositivos habituales mencionados previamente [1].

Lo que distingue a los dispositivos IoT de aquellos con sensores convencionales es principalmente la capacidad para

comunicarse de manera directa o indirecta con la red. Los sensores producen una gran cantidad de información que requiere ser administrada de alguna manera. En términos generales, la capacidad de almacenamiento interna es bastante restringida, por lo que se recurre a alternativas como guardar datos en tarjetas de almacenamiento, o en ordenadores en situaciones en las que los sensores estén directamente conectados a ellos. Dado que los sensores pueden ser incorporados a dispositivos con mejores habilidades de conectividad, la información podría ser guardada en la nube [1].

En este trabajo se diseña un sistema de monitorización de habitaciones IoT. Se desarrolla para observar el funcionamiento de los dispositivos en ciertos entornos. Así, los objetivos de este trabajo son implementar y desarrollar un sistema para monitoreo de habitaciones. En este proyecto se emplea una aplicación móvil que muestra el estado de la habitación de manera inalámbrica mediante una base de datos en Arduino Cloud.

El sistema desarrollado debe supervisar los dispositivos según las circunstancias identificadas, por ejemplo, activar la luz cuando se perciba movimiento en el cuarto o emitir un zumbador cuando la temperatura sea muy alta en la habitación. Se implementa mediante una Raspberry Pi 4. De manera opcional se puede incluir una cámara de video, dependiendo de la aplicación.

## II. TRABAJOS RELACIONADOS

En [2] se diseña, analiza e implementan un sistema automatizado doméstico en una maqueta de casa, que posibilita efectuar labores de supervisión y control de aparatos a distancia [2]. En el Estudio de la plataforma doméstica Home Assistant e integración en Raspberry Pi, se propone un análisis del programa Home Assistant, que se incorpora en una Raspberry Pi, y a través de la adquisición de varios aparatos, se establece un sistema de automatización del hogar por medio de la unión de los mismos[3]. En el Estudio de sistemas de control IoT de bajo coste con Aplicación a variables medioambientales: Proponen un sistema de IoT de tipo bajo coste para la

monitorización de parámetros ambientales en una habitación del hogar, que puede ser extensible a aulas, laboratorios u oficinas [4]. El Prototipo de monitoreo ambiental aplicando el Internet de las cosas con Arduino y Cloud Computing, Diseña un modelo electrónico para la supervisión de los parámetros ambientales: calor, humedad, sonido y nivel de CO, desde una perspectiva centrada en el concepto de “Internet de las Cosas” [1]. También se puede revisar el Sistema de monitorización basado en plataformas arduino, android y cloud en donde implementan un sistema de supervisión que posibilita integrar las plataformas Arduino, Android y la Nube para observar en el dispositivo portátil Android los datos recopilados por dos detectores (calor y luz) conectados a la plataforma mediante un enlace dentro de la misma red WiFi[5]. En [6] se diseñó e implemento de un sistema de control y monitorización remoto de bajo coste mediante Raspberry Pi y gestión en la nube con Firebase; aquí Diseñan un proyecto de instalación digital de bajo presupuesto para gestionar y supervisar remotamente procedimientos fabriles, diseñados para negocios pequeños.

El IoT: Internet de las Cosas (IoT,) se ha establecido como una nueva moda para el progreso de varias aplicaciones conectando una amplia gama de aparatos. Estos pueden ser de diferentes tipos o productores, situación que causa que el avance de estas aplicaciones sea difícil de manejar. Esto lleva a que en cada proyecto se debará revisar las estructuras de construcción. Para esto se analizan varios estilos de programas importantes, y se proponen que factores debieran ser considerados al enfrentar un desarrollo que utilice IoT [7].

Algunas aplicaciones se vinculan con el ámbito educativo así tenemos la Integración de una red de sensores con una plataforma IoT para control inteligente de aulas: Los más recientes progresos en las tecnologías de conexión y detección representan un entorno adecuado para las soluciones basadas en IoT. En la actualidad, hay una diversidad de herramientas que asisten en aprovechar el potencial de estas soluciones, incluyendo las Plataformas IoT, acompañadas por niveles de procesamiento próximos a los márgenes de la red. En [8] se explica la instalación de un sistema para supervisión y control inteligente de características de interés en un edificio académico fundamentado en una plataforma IoT, y se presentan los resultados iniciales de la funcionalidad implementada y las evaluaciones de conectividad realizadas entre los sensores y el servidor.

Recientemente, la aplicación de la tecnología Idc a sistemas de eficiencia energética ha suscitado interés, especialmente para el monitoreo de la eficiencia de sistemas en tiempo real. Este artículo [9] expone el diseño y evolución de ChuchusMOTE, un sistema de vigilancia energética y control doméstica a través del uso de una red de detectores y activadores inalámbricos, que emplean el protocolo de transmisión asíncrona MQTT (Cola de Mensajes de Telemetría de Transporte) para el envío de datos; y permite su visualización en tiempo real. El sistema ChuchusMOTE fue desplegado para vigilar en tiempo real magnitudes energéticas de paneles

solares (tensión, intensidad, potencia y energía generada), el consumo de energía eléctrica y estimar el consumo energético dentro del “Laboratorio de Energías Sustentables” de la Universidad Privada Boliviana [9].

En [10] se muestra el diseño y realización de una malla de sensores administrada mediante la tecnología IoT (Internet de las Cosas). La malla se emplea en una aplicación simulada de Automatización del Hogar. Se combinan diversas tecnologías como la tarjeta de desarrollo photon de particle que coordina la comunicación con los distintos sensores utilizados; Se usó la herramienta Node Red de IBM para programar la interfaz gráfica de usuario a la cual se accede mediante una instancia en un servidor desplegado en IBM Cloud, permitiendo así la opción de revisar en todo momento y lugar por medio de una computadora o un teléfono móvil inteligente el estado de operación de los sensores y el valor de las variables como temperatura, concentración de gases, estado de encendido/apagado de luminarias y detección de movimiento en recintos. Se detallan aspectos sobre la estructura del sistema y los logros alcanzados [10].

### III. METODOLOGÍA

#### A. Componentes generales del sistema

Para el desarrollo del Proyecto es necesario cuatro componentes generales: Una **Raspberry Pi** que puede ser cualquier modelo, pero los modelos más recientes (como Raspberry Pi 4 y 5) ofrecen más potencia y funcionalidades [11] [12]; Los **Sensores**: Dependiendo de lo que se quiera monitorear, en el presente caso sensores de temperatura, humedad, movimiento, luz; pero pueden emplearse otros; Una **Conexión a Internet**: a través de Ethernet o Wi-Fi [13]; y **Software** que Incluye el sistema operativo (como Rasbian), software de programación (Python)[14] y herramientas para la visualización de datos (Node-RED, cloud firestone) [15].

#### B. Implementación

Para la implementación se siguieron seis pasos:

##### 1. Configuración de Raspberry Pi:

- Instalar Raspberry Pi OS en una tarjeta SD y configúrala.
- Conectar a Internet y asegurarse de que esté actualizada.

##### 2. Instalación de Sensores:

- Conectar los sensores a los pines GPIO de la Raspberry Pi.
- Instalar las bibliotecas necesarias para interactuar con los sensores.

##### 3. Recopilación de Datos:

- Escribir scripts en Python para leer los datos de los sensores.
- Asegurarse de que los datos se recojan a intervalos regulares y se almacenen de forma adecuada.

##### 4. Almacenamiento de Datos:

- Se puede almacenar los datos localmente en la Raspberry Pi o enviarlos a una base de datos en la nube.
- Usar bases de datos como SQLite para almacenamiento local o servicios en la nube como AWS, Azure, o Google Cloud.

#### 5. Visualización de Datos:

- Herramientas como Node-RED permiten crear flujos de datos y visualizaciones interactivas.
- Grafana es excelente para crear dashboards detallados.

#### 6. Alertas y Automatización:

- Configura alertas que se activen cuando los datos de los sensores superen ciertos umbrales.
- Automatiza acciones en respuesta a estos datos, como encender un ventilador si la temperatura es demasiado alta.

### C. Resumen del diseño

Este trabajo está compuesto por tres etapas que son: el monitoreo de la habitación, los datos de la nube en cloud firestore, y la aplicación móvil. El modelo de vigilancia de la habitación está diseñado con diversos sensores conectados al microprocesador para vigilar la habitación. El microprocesador recibe la información de los sensores y ejecuta las órdenes correspondientes. La plataforma cloud firestore en la red es el repositorio que guarda la información de los sensores para que el usuario pueda acceder a ellos en cualquier momento y lugar. La conexión a la red o red inalámbrica del microprocesador es indispensable para permitir el acceso a la red. Además, la aplicación móvil ayuda al usuario a monitorear de forma remota el aparato en la estancia. La Figura 1 muestra el esquema de bloques del sistema.

El propósito de este modelo elaborado es asistir al usuario en la supervisión del aparato de su habitación. Las características principales de este sistema son la supervisión de los parámetros de la sala, la seguridad y los dispositivos. Los parámetros de la habitación considerados en este proyecto incluyen la temperatura, la humedad y la luz. Estos parámetros son detectados por varios sensores que están conectados al microcontrolador y envían los datos a la plataforma en la nube a través de Internet. Para construir todas estas características, se consideran varios sensores y microcontroladores en función de sus especificaciones y tamaños. Se utiliza una Raspberry Pi 4 como microcontrolador Arm debido a su bajo costo y accesibilidad a Internet. El lenguaje Python se usa para escribir el código del programa en el microcontrolador para ejecutar las órdenes.

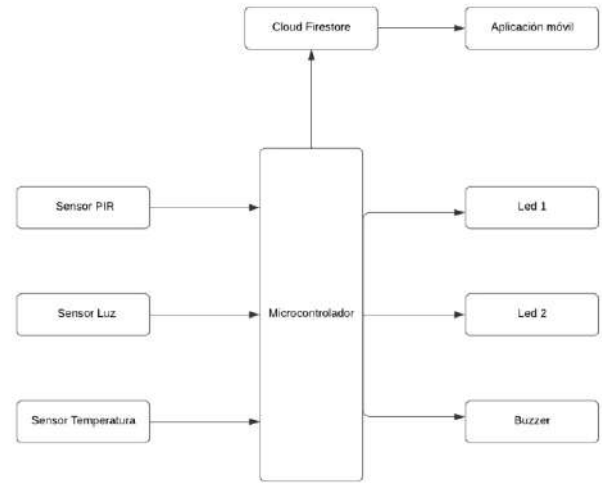


Fig. 1. Esquema de bloques del sistema.

### D. Configuración de aplicaciones móviles

La aplicación móvil se desarrolla para interactuar con los microcontroladores. Se diseña en Visual Studio Code, a través del framework de desarrollo de aplicaciones de código abierto Flutter, utilizando Dart, un lenguaje de programación optimizado para el desarrollo de interfaces de usuario. Las funcionalidades de la aplicación móvil elaborada son mostrar los datos más recientes de los sensores con la hora actual. Cuando hay nuevos datos guardados en Cloud Firestore la aplicación móvil los muestra en la pantalla al presionar el botón de actualizar en la parte inferior derecha. Durante la elaboración de la aplicación móvil, se prueba ejecutando el emulador antes de instalarla en el dispositivo real, así se asegura la protección del teléfono inteligente del usuario contra posibles daños u otros inconvenientes.

### E. Cámara de Video

Agregar una cámara a un sistema IoT con Raspberry Pi puede ofrecer múltiples ventajas, pero también algunas desventajas:

#### Ventajas

#### 1. Monitoreo visual en tiempo real:

Con una cámara, se tiene una vista en tiempo real del entorno, lo cual es especialmente útil para seguridad, vigilancia o monitoreo remoto. Se pueden observar áreas de interés.

#### 2. Detección avanzada de eventos:

Las cámaras permiten implementar **detección de movimiento** o reconocimiento facial, lo que puede desencadenar acciones automáticas, como activar alarmas o enviar notificaciones cuando se detecta una intrusión.

### 3. Captura de datos más completos:

Las imágenes o videos capturados pueden complementar los datos de sensores tradicionales (como temperatura o humedad), proporcionando una capa adicional de información.

### 4. Documentación de eventos:

Con la cámara se pueden grabar videos o capturar imágenes en momentos clave, lo que puede ser útil para revisiones posteriores o análisis. Esto es útil en sistemas de automatización o vigilancia para auditar eventos específicos.

### 5. Flexibilidad en la automatización:

Las cámaras pueden integrarse con algoritmos de visión por computadora para tareas complejas, como contar objetos, detectar fallos en líneas de producción o analizar patrones de comportamiento en áreas controladas.

## Desventajas

### 1. Consumo de recursos:

Procesar imágenes y videos en tiempo real requiere más potencia de procesamiento y memoria. Esto puede sobrecargar una Raspberry Pi, especialmente si se intenta procesar video en alta resolución o ejecutar algoritmos complejos de visión por computadora sin optimizaciones. También se reduce la velocidad de respuesta de otros sensores en el sistema.

### 2. Almacenamiento:

Los archivos de imagen y video ocupan mucho espacio. Si se graban videos largos o se capturan imágenes de forma continua, el almacenamiento local en la Raspberry Pi puede agotarse rápidamente. Además, si se utiliza almacenamiento en la nube, los costos pueden aumentar.

### 3. Consumo de energía:

Las cámaras suelen aumentar el consumo de energía del sistema IoT, lo cual puede ser un problema en aplicaciones donde la Raspberry Pi depende de baterías o fuentes de energía limitadas. Esto es especialmente crítico en soluciones IoT de monitoreo remoto donde la eficiencia energética es clave.

### 4. Privacidad y seguridad:

El uso de cámaras en proyectos de IoT puede generar preocupaciones de privacidad, especialmente en entornos públicos o áreas sensibles. Además, las cámaras conectadas a internet pueden ser vulnerables a ataques de ciberseguridad si no se implementan adecuadamente medidas de protección, como el cifrado de datos o la autenticación robusta.

### 5. Costo adicional:

Agregar una cámara implica un costo adicional tanto por el hardware de la cámara como por posibles actualizaciones necesarias (mejoras en almacenamiento, batería, etc.). Si se requiere una cámara de alta calidad o con características avanzadas (como visión nocturna), los costos pueden incrementarse notablemente.

### 6. Latencia en la transmisión:

La transmisión en tiempo real de video o imágenes a través de una red puede causar retrasos (latencia), especialmente si la conexión de internet no es lo suficientemente rápida o estable. Esto podría afectar la capacidad de respuesta del sistema en tiempo real, particularmente en aplicaciones de seguridad o control de acceso.

## F. Confiabilidad

Para asegurar la **confiabilidad del monitoreo IoT mediante Raspberry Pi**, es fundamental:

- Seleccionar hardware robusto.
- Implementar redundancia en almacenamiento y energía.
- Utilizar software bien diseñado, con manejo de excepciones y optimización.
- Configurar mecanismos de monitoreo continuo y alertas automáticas.
- Mantener una infraestructura segura y actualizada.
- Realizar pruebas y simulaciones para verificar la capacidad de respuesta.

Con estas estrategias, se puede minimizar los riesgos de fallos y asegurar que el sistema de monitoreo IoT funcione de manera confiable y segura.

## IV. RESULTADOS Y ANALISIS

### A. Prototipo de hardware

La Figura 2 muestra el prototipo de desarrollo en hardware para la habitación. A la vez que el sistema se ha incorporado satisfactoriamente y tienen la capacidad de comunicarse entre sí, todas las salidas se pueden visualizar en la consola de la Raspberry pi 4. Una vez que el sistema se enciende, el modo de monitoreo del dispositivo de la habitación es leído por el sistema desde Cloud Firestore periódicamente. Los sensores de temperatura, luz y movimiento monitorean automáticamente los tres tipos de datos de la habitación: temperatura, humedad, luz y si hay movimiento en la sala. Estos datos se almacenan dependiendo de cómo este configurado el microprocesador. La temperatura, humedad y luz pueden devolver salidas consistentes y detectar los cambios de entrada surgidos mientras el sistema está en funcionamiento.



Fig. 2. Prototipo de sistema de monitorización de habitaciones  
Fuente: Elaboración propia

### B. Aplicación móvil

Una vez que el sistema está en funcionamiento la aplicación móvil monitorea y muestra los valores correspondientes. Los sensores funcionan de la siguiente manera: el sensor PIR se encarga de detectar movimiento humano en la habitación, si detecta movimiento uno de los led se enciende mostrando valor de “Encendido” en la aplicación móvil, y se muestra en “Apagado” en caso de que no se detecte movimiento en la sala, el sensor de temperatura y humedad muestra los valores en la aplicación, la temperatura en grados Celsius y la humedad en porcentaje, mientras que si la temperatura excede los 30 grados Celsius hace sonar un buzzer y este muestra valor de “Encendido” en la aplicación móvil o de “Apagado” si la temperatura no supera los 30 grados; por último, el sensor de Luz mientras que no haya suficiente luz en la sala se va a mantener en “Encendido”, si la luz es suficiente en la habitación va a mantener apagado el led y va a visualizar el valor de “Apagado” en la aplicación móvil. La Figura 3 muestra la interfaz de la aplicación móvil para controlar el dispositivo de la sala en el dispositivo real.

El sistema de monitorización IoT desarrollado mediante Raspberry Pi 4 presenta una solución eficaz para supervisar diversos parámetros ambientales en una habitación. Al utilizar

sensores de temperatura, humedad y movimiento, junto con una conexión a la plataforma Cloud Firestore para almacenar y gestionar datos, y una aplicación móvil desarrollada en Flutter para la interfaz de usuario, se logra una integración efectiva de hardware y software. La implementación muestra la capacidad de la Raspberry Pi como microcontrolador accesible y económico, aprovechando Python para la programación. Esta solución no solo demuestra viabilidad técnica, sino también la escalabilidad y versatilidad del IoT en aplicaciones domésticas, ofreciendo a los usuarios una herramienta práctica para monitorear y controlar su entorno de manera remota y eficiente. El proyecto se puede mejorar conectando una cámara que transmita información en vivo al usuario para que este pueda supervisar la sala constantemente. Sin embargo, esto dependerá del escenario como:

- Vigilancia de seguridad.
- Control de calidad en manufactura.
- Monitoreo de tráfico o personas.
- Detección de anomalías visuales (por ejemplo, detección de incendios o fugas de agua).

Sin embargo, es importante evaluar si la infraestructura y recursos disponibles son suficientes para soportar la cámara sin comprometer el rendimiento general del sistema IoT. De esta manera su uso es opcional.



Fig. 3. Interfaz de la aplicación móvil Fuente.

### C. Pruebas de confiabilidad

Para asegurar la **confiabilidad del monitoreo IoT mediante Raspberry Pi**, se siguieron los puntos:

- Seleccionar hardware robusto.

- Implementar redundancia en almacenamiento y energía.
- Utilizar software bien diseñado, con manejo de excepciones y optimización.
- Configurar mecanismos de monitoreo continuo y alertas automáticas.
- Mantener una infraestructura segura y actualizada.
- Realizar pruebas y simulaciones para verificar la capacidad de respuesta.

Con estas estrategias, se minimizo los riesgos de fallos y aseguro que el sistema de monitoreo IoT funcione de manera confiable y segura, dando como resultado la tabla 1.

TABLE I

LA TABLA I MUESTRA LAS PRUEBAS DE EVALUACION DE SENSORES

Pruebas	Temperatura	Humedad	Movimiento	Luz
Funcionales	91.3 %	96.23 %	80 %	100 %
Integración	Si	Si	Si	Si
Rendimiento	Optimo	Optimo	Optimo	Optimo
Confiabilidad	Estable	Estable	Estable	Estable
Mantenimiento	NA	NA	Recalibrable	NA

En la tabla I, se muestra los resultados de las pruebas realizadas; el sensor de temperatura demostró una precisión del 91.3% en pruebas funcionales, está completamente integrado en el sistema, funciona de manera óptima, y es confiable y estable en su rendimiento. No requiere mantenimiento adicional. El sensor de humedad alcanzo una precisión del 96.23% en las pruebas funcionales, también está integrado completamente, muestra un rendimiento óptimo y es confiable y estable en su operación, sin necesidad de mantenimiento adicional. El sensor de movimiento registro una precisión del 90% en las pruebas funcionales y, al igual que los anteriores, está completamente integrado, funciona óptimamente, es confiable y estable, aunque puede requerir recalibración periódica. Finalmente, el sensor de luz obtuvo una precisión del 100% en las pruebas funcionales, está completamente integrado, opera de manera óptima, y es confiable y estable sin necesidad de mantenimiento adicional.

Comparando el sistema de la referencia [16], aunque posee mas sensores y actuadores, lo que agrega mas especificaciones, de igual forma por la arquitectura del sistema propuesto pueden agregarse sin dificultad los mismos sensores y más; solo que, al emplearse una Raspberry de mayores prestaciones, se pueden ampliar sin perder confiabilidad, ya que la arquitectura lo permite.

### CONCLUSIONES

El monitoreo IoT mediante Raspberry Pi es una aplicación poderosa y versátil que combina hardware accesible y software flexible para recoger, procesar y analizar datos de sensores en tiempo real. Esta tecnología encuentra aplicaciones en una amplia gama de campos, desde la automatización del hogar

hasta la gestión industrial. La Raspberry Pi, debido a su bajo costo, tamaño compacto y capacidad para ejecutar un sistema operativo completo, se ha convertido en una herramienta ideal para proyectos de IoT. Al combinarla con una variedad de sensores y software de procesamiento de datos, es posible construir sistemas de monitoreo robustos y personalizados. En resumen, el monitoreo IoT con Raspberry Pi no solo es factible y práctico, sino que también abre un abanico de posibilidades para innovar y optimizar diversos procesos. Aprovechar los fundamentos matemáticos y técnicos garantiza que los sistemas construidos sean precisos, confiables y eficientes, cumpliendo así con las exigencias del entorno moderno de la IoT.

### ACKNOWLEDGMENT

Los autores reconocen el apoyo recibo del Instituto Politécnico Nacional a través de la Escuela Superior de Cómputo para la realización del Proyecto presentado.

### REFERENCES

- [1] F. D. Salgado Castillo and D. S. Coello Moncayo, "Prototipo de monitoreo ambiental aplicando el internet de las cosas con arduino y cloud computing," B.S. thesis, Universidad del Azuay, 2015.
- [2] H. S. Senra, "Diseno e implementaci on de un sistema dom' otico basado' en raspberry pi," Trabajo fin de grado, Departamento de Informatica, Universidad Carlos III de Madrid, Madrid, Espana, 2017.
- [3] S. Romero Cabrera, "Estudio de la plataforma domotica home assistant' e integracion en raspberry pi," 2019.'
- [4] L. J. Puente Lam *et al.*, "Estudio de sistemas de control iot de bajo coste. aplicacion a variables medioambientales," 2022.'
- [5] R. Rodr'iguez Ordo'nez' *et al.*, "Sistema de monitorizacion basado en plataformas arduino, android y cloud," 2017.
- [6] D. Martínez Mart'inez, "Diseno e implementaci on de un sistema de control y monitorizacion remoto de bajo coste mediante raspberry' pi y gestion en la nube con firebase," Ph.D. dissertation, Universitat Politecnica de Val' encia, 2022.'
- [7] G. M. Barrera, "Estilo arquitectonico para aplicaciones iot," Serie' Documentos de Trabajo, Tech. Rep., 2018.
- [8] L. Gomez D'Orazio, S. Medina, and D. M. Montezanti, "Integracion de' una red de sensores con una plataforma iot para control inteligente de aulas," in XXVIII Congreso Argentino de Ciencias de la Computacion' (CACIC)(La Rioja, 3 al 6 de octubre de 2022), 2023.
- [9] E. Escobar Gallardo and A. Villazon, "Sistema de monitoreo energ' etico' y control domotico basado en tecnolog' ia internet de las cosas," Investigacion & Desarrollo, vol. 18, no. 1, pp. 103–116, 2018.
- [10] M. A. P. Gonzalez, D. A. V. Romero, H. R. C. P'erez, E. H. Bar'on, E. V.' Laureano, and J. S. Baylon, "Diseno e implementaci on de una red de' sensores gestionada por iot para aplicaciones de domotica," INVURNUS, vol. 18, no. 1, 2023.
- [11] Monk, Simon. Raspberry pi cookbook. " O'Reilly Media, Inc.", 2022.
- [12] Maksimović, Mirjana, et al. "Raspberry Pi as Internet of things hardware: performances and constraints." design issues 3.8 (2014): 1-6.
- [13] Waher, Peter. Learning internet of things. Packt publishing, 2015.
- [14] Monk, Simon. Programming the Raspberry Pi: getting started with Python. Mcgraw-hill, 2013.
- [15] Waher, Peter. Learning internet of things. Packt publishing, 2015.
- [16] Hanah, Aznor, et al, IoT Room Control And Monitoring System Using Rasberry Pi, 4th International Conference and Workshop on Recent Advances and Innovations in Engineering, IEEE, 2019

# Sistema DAQ para el monitoreo y análisis de sistemas metalúrgicos durante el ciclo térmico de solidificación

Gonzalo Hedain López-Mera  
Departamento de Ingeniería  
Facultad de Estudios Superiores  
Cuautitlán – UNAM  
Cuautitlán Izcalli, México  
<https://orcid.org/0000-0002-5843-7438>

Diego Sánchez-García  
Departamento de Ingeniería  
Facultad de Estudios Superiores  
Cuautitlán – UNAM  
Cuautitlán Izcalli, México  
<https://orcid.org/0009-0001-1821-5953>

Raúl Dalí Cruz-Morales  
Departamento de Ingeniería  
Facultad de Estudios Superiores  
Cuautitlán – UNAM  
Cuautitlán Izcalli, México  
<https://orcid.org/0000-0002-8754-8196>

David Tinoco-Varela  
Departamento de Ingeniería  
Facultad de Estudios Superiores  
Cuautitlán – UNAM  
Cuautitlán Izcalli, México  
<https://orcid.org/0000-0002-1919-7451>

Alma Alejandra Luna-Gómez  
Departamento de Ingeniería  
Facultad de Estudios Superiores  
Cuautitlán – UNAM  
Cuautitlán Izcalli, México  
<https://orcid.org/0009-0006-1156-5519>

**Resumen—** La adquisición precisa y accesible de datos en experimentos térmicos presenta desafíos significativos debido a la falta de dispositivos especializados de bajo costo y a interfaces de usuario dedicadas al análisis de fenómenos térmicos específicos. En este trabajo se aborda esta problemática mediante el desarrollo de una interfaz gráfica de usuario (GUI) en Visual Studio y de un sistema de adquisición de datos (DAQ) adoptando la filosofía de código abierto, con la propuesta de tener un sistema de medición y monitoreo configurable y eficiente que interprete los resultados obtenidos en un gráfico para su análisis. Por lo anterior, se presenta la demostración experimental de la GUI y el sistema DAQ obteniendo la curva de enfriamiento durante la solidificación de estaño de pureza comercial.

**Palabras clave—** Desarrollo, temperatura, interfaz gráfica de usuario, sistema de adquisición de datos, monitoreo, estaño.

## I. INTRODUCCIÓN

La medición de la temperatura es fundamental en gran parte de componentes, aplicaciones científicas, desarrollo de productos y en sistemas industriales. En este enfoque un sistema DAQ es una herramienta útil ya que permite capturar de forma automática y continua datos térmicos en tiempo real. Estos sistemas no solo proporcionan mediciones instantáneas en intervalos programados, sino que también, almacenan datos a lo largo del tiempo, permitiendo realizar análisis detallados de tendencias y variaciones. Por otro lado, la capacidad de analizar datos históricos de temperatura permite la detección de problemas, la optimización de procesos y la toma de decisiones sustentadas.

En este trabajo, se trata el desarrollo de un sistema DAQ para el monitoreo de temperatura en el método de análisis térmico de curvas de enfriamiento, lo cual, representa una alternativa para la construcción de diagramas de fases. Se elige el método de análisis térmico al ser el método más sencillo y uno de los más

empleados en la construcción de diagramas de fase, ya que permite identificar las transiciones de fase de un material a diferentes temperaturas y presiones como se explica en [1], permitiendo estudiar el comportamiento térmico durante la solidificación de materiales metálicos. Contar con un sistema DAQ en el laboratorio enfocado a tecnología de materiales o ciencia de los materiales permitirá crear prácticas de laboratorio para cubrir temas de transformaciones de fases y diagramas de fases, lo que permite a estudiantes de licenciatura aplicar los conocimientos teóricos de cómo las condiciones de temperatura afectan las fases y propiedades de los materiales.

Dentro de los sistemas de monitoreo que hacen uso de un enfoque similar se tiene a [2], donde se presenta un diseño de un sistema de adquisición, almacenamiento y registro de datos para 5 entradas analógicas, entre estas, la variable de temperatura. Aquí se diseña una Interfaz Hombre Máquina (HMI) diseñada en Visual Studio 2010 usando el lenguaje de programación Visual Basic. Los datos son registrados en la HMI y almacenados en una memoria microSD. En cuanto al enfoque de código abierto y reducción de costos se tiene el trabajo de [3], donde se presenta un sistema de adquisición de datos de código abierto utilizando la tarjeta Arduino UNO, un módulo lector de tarjeta SD y un sensor HMZ-433 A1, el cual es un sensor de temperatura y humedad.

Se hace mención que el sistema se verificó en funcionamiento dentro de una caseta meteorológica. Aquí los autores destacan que es posible construir un sistema DAQ de bajo costo basado en la tarjeta Arduino UNO haciendo uso de módulos compatibles con Arduino, además, al ser un sistema de código abierto se pueden realizar modificaciones al sistema para censar variables de interés.

En el área de la ciencia de los materiales, la investigación realizada en [4] presenta el desarrollo de un instrumento virtual desarrollado en LabView y acoplado a un sistema DAQ

Compact RIO de National Instruments. El objetivo de dicha plataforma es generar perfiles de temperatura en el proceso de cocción en hornos a través de la curva característica del proceso de cocción real. Los autores mencionan que trabajar bajo este enfoque permite tener un sistema de bajo costo, reutilizable, de arquitectura abierta y definido por el usuario. Con estos antecedentes se propone desarrollar un sistema DAQ que integre una GUI en Visual Studio, el almacenamiento de datos tanto en la GUI propuesta, como en una tarjeta microSD, y la utilización de una tarjeta Arduino UNO.

A continuación, se presenta el desarrollo del sistema DAQ propuesto y la GUI utilizada para el monitoreo del comportamiento térmico de sistemas metalúrgicos durante su enfriamiento, iniciando con el material en el punto de fusión hasta su solidificación y la forma en cómo se almacenan los datos obtenidos para realizar el análisis de curvas de enfriamiento. Las partes principales que componen el sistema son: una GUI desarrollada en Visual Studio, un microcontrolador Arduino UNO y el diseño de un sistema de medición de temperatura utilizando termopares tipo K.

La demostración experimental del sistema se realiza mediante la adquisición de datos de temperatura del análisis térmico en la curva de enfriamiento del proceso de calentamiento-solidificación del estaño. En la Sección II se presentan las características de diseño del sistema DAQ. Con base en la propuesta de monitoreo de temperatura en tiempo real, la Sección III describe los componentes utilizados en el diseño de la GUI. En la Sección IV se muestra la parte experimental en donde se realiza la demostración experimental de la GUI y el sistema DAQ propuestos. Los resultados de los experimentos y la adquisición de datos de temperatura se presentan en la Sección V, mostrando que un sistema de bajo costo y arquitectura abierta es eficaz y con variaciones pequeñas de las lecturas experimentales comparadas con los cálculos teóricos, las cuales son aceptables en este tipo de trabajos metalográficos. Finalmente, en la Sección VI se dan a conocer las conclusiones y trabajo futuro.

## II. CARACTERÍSTICAS DEL SISTEMA DAQ

Para la obtención de datos en el proceso de solidificación y la transición de temperaturas es necesario tener un sistema DAQ de temperatura que recabe estos datos a lo largo del tiempo. Como antecedente, en la construcción de diagramas de fases por medio del método de análisis térmico, se cuenta con un sistema DAQ comercial Modelo USB-TC del fabricante JM Industrial, que de acuerdo con [5] soporta 8 canales de entrada de termopar y 8 tipos de termopar, entre los que se encuentra el termopar tipo K, cuenta con un sensor de temperatura ambiente incorporado y 8 bits digitales de Entrada/Salida y tiene su propio software gratuito llamado TracerDAQ 2.0.

Dentro de las desventajas de este sistema DAQ, se tiene que el sistema presenta susceptibilidad a descargas electrostáticas durante su manipulación y el uso recomendado de utilizar solamente termopares aislados, que para el propósito de la medición de temperaturas en los sistemas metalúrgicos en estado líquido resulta inconveniente ya que no hay contacto directo entre el termopar y el sistema. En la obtención de curvas de enfriamiento utilizando el DAC USB-TC con termopares tipo

K con junta expuesta [6], la imprecisión en las mediciones y el alto costo de adquisición del dispositivo, son el factor motivante para proponer en este trabajo el diseño y construcción de un sistema DAQ de temperatura de propósito especial.

La tendencia en el diseño electrónico del sistema DAQ de temperatura desarrollado para la obtención de curvas de enfriamiento tiene como motivación generar un diseño innovador, de bajo costo y el uso de un enfoque de diseño modular. Para la disminución de costos, se propone usar tecnología de software de código abierto y de arquitectura abierta, lo que permite realizar cambios de diseño respecto las necesidades del usuario. Además, el diseño modular permite la escalabilidad y ahorro en el tiempo de desarrollo.

Por tanto, el sistema DAQ de temperatura propuesto en el presente documento, tiene como elemento principal el uso de la tarjeta Arduino UNO, que es una plataforma de código abierto con su propio entorno gráfico de desarrollo basado en el microcontrolador ATmega328P.

En la elección del sensor de temperatura se contempla el uso del termopar tipo K (Cromel-Alumel), que de acuerdo con sus características presentadas en [6] permite trabajar en un rango de temperatura de  $-200\text{ }^{\circ}\text{C}$  a  $1250\text{ }^{\circ}\text{C}$  con un límite de error estándar de  $\pm 0.75\%$  entre los  $293\text{ }^{\circ}\text{C}$  a  $1250\text{ }^{\circ}\text{C}$ .

Respecto al tipo de junta caliente, se elige la configuración de termopar de tipo junta expuesta, como se ve en la Fig. 1, esta configuración es de alambres desprotegidos y es la junta con la respuesta más rápida en comparación a la rapidez de respuesta que puede proporcionar una junta a tierra o una junta aislada. Cabe mencionar que el termopar usado es de fabricación propia.



Fig. 1. Estilos de junta caliente para termopares. Fuente: [6].

Entre los dispositivos utilizados para conectar al termopar con el Arduino UNO considerando las características mostradas en la Tabla 1, se elige el convertidor MAX6675 [7], donde dicho dispositivo ocupa comunicación SPI con resolución de 12 bits, una resolución de temperatura de  $0.25\text{ }^{\circ}\text{C}$  y un rango de temperatura asociado al termopar tipo K de  $0$  a  $800\text{ }^{\circ}\text{C}$ .

Como elemento de almacenamiento de información se utiliza un módulo lector de microSD [8] el cual tiene la capacidad de soportar memorias microSD de hasta 32 GB y el cual utiliza comunicación SPI.

Tabla 1: Módulos para termopar tipo K. Fuente: Elaboración propia.

Módulo	MAX6675	MAX31855	AD595A
Tipo de salida	12 bits resolución Salida SPI	14 bits resolución Salida SPI	Voltaje analógico
Resolución	0.25 °C	0.25 °C	10mV/°C
Rango de temperatura (°C)	0-800	-200-800	-200-1250
Costo (Pesos)	\$125.00	\$300.00	\$722.00
Tipo de circuito	Shield	Shield	Integrado

Para la interfaz visual en la interacción de la información almacenada con el usuario se opta por una pantalla LCD de 16x2, esto es 16 columnas y 2 filas con una interfaz de comunicación de tipo paralelo de 4 y 8 bits. La pantalla LCD se eligió en fondo azul y texto blanco.

Un valor agregado al sistema es la integración del módulo de reloj RTC DS3231 para tener la lectura de tiempo real. Dentro de las características del RTC se tiene que puede mantener un registro de segundos, minutos, horas, día de la semana, fecha, mes y año. La fecha se puede ajustar de forma automática en caso de correcciones para año bisiesto y el módulo utiliza comunicación I2C. Para detallar la configuración del sistema DAQ en cuanto a la integración de sus elementos con la tarjeta Arduino UNO se presenta el esquema eléctrico en la Fig. 2.

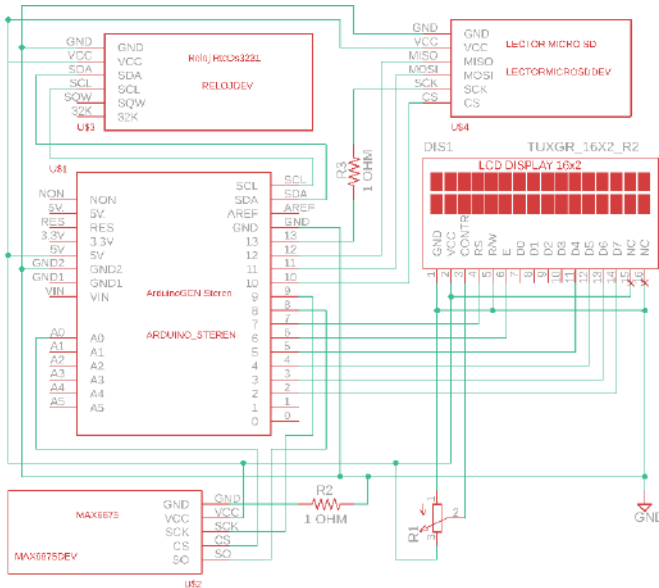


Fig. 2. Esquema eléctrico del sistema DAQ. Fuente: Elaboración propia.

A continuación, se muestra el pseudocódigo que describe de forma simplificada el algoritmo utilizado para la programación de la tarjeta Arduino Uno:

- Inicio;
- Cargar librerías para establecer la comunicación con los módulos;
- Establecer la comunicación con la pantalla LCD;
- Mostrar en la pantalla LCD el mensaje de bienvenida;

- Inicializar el reloj en tiempo real;
- Verificar la actualización de la fecha, en caso contrario, forzar dicha actualización;
- Leer el archivo .txt con el valor numérico en milisegundos del lapso de tiempo ubicado en la microSD contenida en el módulo lector de microSD, que es el intervalo de tiempo de almacenamiento del sistema DAQ ajustado en 1000 (1 s) y con la posibilidad ser modificado por el usuario;
- En caso de que el módulo lector no contenga una microSD el programa se da por terminado, enviando un mensaje de alerta a la pantalla LCD;
- Con los datos iniciales, se crea el archivo .CSV en la memoria, dedicado para almacenar los datos;
- Almacenar los datos de fecha, tiempo transcurrido y el valor de temperatura obtenido del termopar durante el ciclo de lectura en el archivo .CSV;
- Enviar los datos de temperatura y tiempo transcurrido en cada ciclo a la pantalla LCD.

El sistema DAQ de medición de temperatura en su versión de prueba se muestra en la Fig. 3.

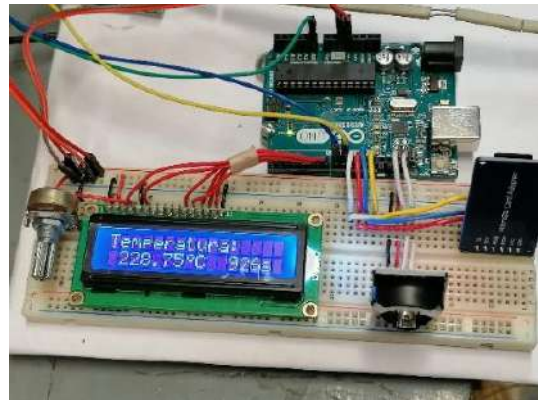


Fig. 3. Sistema DAQ de medición de temperatura en su versión de prueba. Fuente: Elaboración propia.

Para la calibración del sistema DAQ de medición de temperatura se utilizó un multímetro de la marca Fluke modelo 17B+, resultando en una diferencia de valores de 0.25 °C en comparación de mediciones controladas. Una vez completado el circuito en su versión de prueba y verificando el correcto funcionamiento se realiza su versión en PCB, permitiendo obtener una versión compacta y duradera. Para esto, se procede a diseñar el esquema del circuito en el software Eagle de Autodesk como se muestra en la Fig. 4 y manufacturada en una máquina CNC Router 3018.

### III. INTERFAZ GRÁFICA DE USUARIO EN VISUAL STUDIO

La GUI diseñada en este trabajo se desarrolla en la plataforma Visual Studio, usando el lenguaje de programación de Aplicación de Windows Forms (.Net Framework), mismo que utiliza lenguaje C#. El marco utilizado es el .NET Framework 4.8. La elección de esta plataforma se debe a que permite la creación de aplicaciones para Windows de su diseñador visual, por ser una programación orientada a objetos.

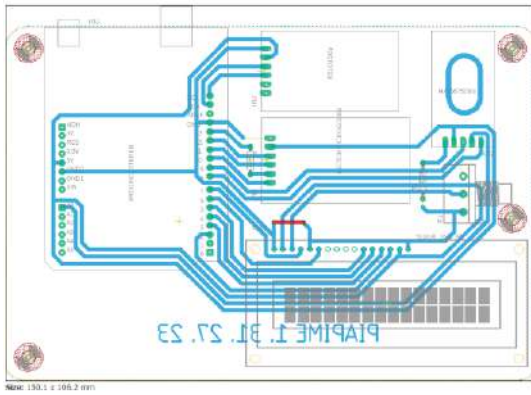


Fig. 4. Estructura del diseño de PCB realizado en Eagle de Autodesk.  
Fuente: Elaboración propia.



Fig. 5. Interfaz gráfica de usuario diseñada en Visual Studio.  
Fuente: Elaboración propia.

Los componentes que se utilizaron en la GUI son:

1. **groupBox**: Representa un control Windows que muestra un marco alrededor de un grupo de controles en un formulario con un título opcional.
2. **label**: Representa una etiqueta estándar de Windows.
3. **comboBox**: Representa un control de cuadro combinado de Windows.
4. **button**: Representa un control de botón de Windows.
5. **pictureBox**: Representa un control de cuadro de imagen de Windows para mostrar una imagen.
6. **panel**: Utilizado para agrupar colecciones de controles.
7. **chart**: El control Chart es un objeto de gráfico que expone eventos.
8. **dataGridView**: Muestra datos en una cuadrícula personalizable.
9. **serialPort**: Representa un recurso de puerto serie.
10. **timer**: Genera un evento después de un intervalo establecido, con la opción de generar eventos recurrentes.

En la interfaz propuesta, se tiene la inicialización del puerto serial, la declaración de animación en la gráfica de temperatura en tiempo real, la declaración del envío de datos para su correcta publicación en la tabla de datos, las acciones para los botones de Actualizar y Conectar, las acciones para las etiquetas de Estado de conexión y las acciones para la etiqueta del valor numérico de la Temperatura Actual, detallados a continuación.

La GUI diseñada para operar con el sistema DAQ de medición de temperatura (presentado en la sección II) utiliza el protocolo de comunicación serial, su configuración inicial se puede ver en la Fig. 5. Esta interfaz permite actualizar los puertos de comunicación mediante el botón amarillo (Actualizar), permitiendo la conexión con el botón verde (Conectar) y cambiando a color amarillo con el texto Desconectar, cuando la interfaz está desconectada.

El historial de valores obtenidos del sistema se mostrará mediante una gráfica de tipo Spline ubicada en la parte central izquierda, que se genera utilizando el método de interpolación para generar un gráfico de línea que utiliza curvas suaves que pasan exactamente por los puntos de entrada. A su vez, la gráfica cuenta con una etiqueta numérica, que se actualizará al valor real actual, de la temperatura obtenida por el termopar.

La Tabla de datos ubicada en la parte central derecha generará un historial numérico de 4 columnas con los datos de Fecha, Hora, Tiempo transcurrido y Temperatura.

El grupo Estado ubicado en la parte inferior izquierda, permite la visualización de los 2 posibles estados de la comunicación serial: Desconectado y Conectado, acompañado de un indicador led verde. Si el estado es Desconectado el led verde permanece apagado y en el estado Conectado el led verde se iluminará. En caso de presentarse el evento de cierre de la GUI, se detiene el timer y se desconecta el puerto serie. Se hace notar que la GUI es nombrada como DAQ Termopar tipo K. Para visualizar el sistema completo de medición de temperatura, la Fig. 6 muestra el diagrama a bloques del proceso descrito,

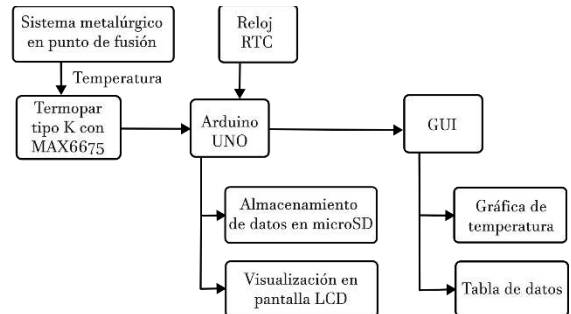


Fig. 6. Diagrama de bloques del sistema completo de medición de temperatura. Fuente: Elaboración propia.

#### IV. VALIDACIÓN EXPERIMENTAL

En esta sección se presenta la validación experimental del sistema DAQ y la GUI diseñada mediante la puesta en marcha del sistema, para la obtención de la curva de enfriamiento del estaño. El tiempo de muestreo dependerá del valor configurado en el sistema DAQ, a través del archivo “veloc.tex”. Para este proyecto se estableció un tiempo de muestreo de 1000 ms, en caso de no existir dicho archivo el sistema tiene el tiempo de muestreo por defecto establecido a 1000 ms. Como se menciona en la Sección II, el primer prototipo fue sometido a diferentes pruebas de medición, calibración y se compararon los resultados obtenidos con un sensor marca Fluke lo que permite asegurar el buen funcionamiento del sistema, la repetitividad y exactitud de sus mediciones, por lo cual se fabrica el sistema DAQ en una

PCB con carcasa manufacturada en impresión 3D de material PLA como se muestra en la Fig. 7.

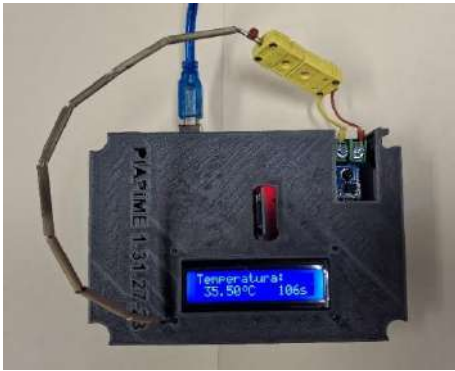


Fig. 7. Sistema DAC en su versión actual. Fuente: Elaboración propia.

La demostración del sistema DAQ se lleva a cabo en un ambiente controlado para el cual fue diseñado, donde se tiene sobre un soporte universal conectado a una computadora por medio de un cable USB tipo A a USB tipo B utilizando comunicación serial para establecer la conexión con la GUI diseñada en Visual Studio. Así también, se tiene el termopar tipo K sumergido en el estaño en su punto de fusión contenido en un crisol de grafito con la capacidad de 70 g como se muestra en la Fig. 8.

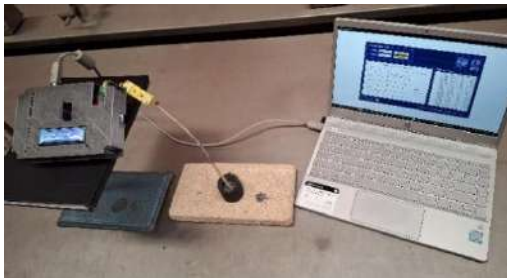


Fig. 8. Demostración de funcionamiento del sistema de medición de temperatura. Fuente: Elaboración propia.

Una vez conectado el sistema DAQ, este inicia su proceso de configuración mostrando mensajes de bienvenida y seguimiento de la configuración de los módulos a través de la pantalla LCD de 16x2. Si el sistema DAQ llega a una pantalla de texto como la mostrada en la Fig. 9, se puede iniciar con la interacción de la GUI en la computadora.



Fig. 9. Pantalla LCD con información de temperatura y tiempo obtenida del sistema DAQ. Fuente: Elaboración propia.

La GUI se inicializa con el grupo de control “Comunicación serial”, donde aparecen los controles de “Puerto” y “Baudios” habilitados con el botón “Conectar” en color verde antes de iniciar una comunicación serial. Una vez que se elige el puerto a utilizar, o presionar el botón de Actualizar para visualizar el

puerto COM y elegir la velocidad de conexión (9600 baudios), se presiona el botón “Conectar” para establecer la comunicación serial. Durante la conexión del sistema DAQ y la GUI, este panel de control de “Comunicación serial” deshabilita la elección de los puertos y velocidad, el botón “Actualizar” y el botón “Conectar” cambia de color verde a amarillo con el texto “Desconectar”, ver Fig. 10.

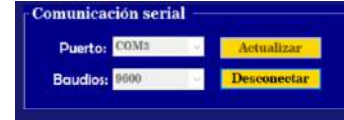


Fig. 10. Estado de Comunicación serial durante la conexión. Fuente: Elaboración propia.

Con el sistema DAQ conectado a la GUI; tanto la gráfica, como la tabla de datos de la temperatura del termopar tipo K inicia mostrando los datos adquiridos de forma gráfica y numérica como se ve en la Fig. 11. El objetivo de tener una gráfica en la GUI es permitir al usuario una interacción gráfica con los datos adquiridos en tiempo real y así enfocarse a la adquisición de datos dentro de la región de la temperatura de solidificación de la curva de enfriamiento del material, o en su caso solo adquirir datos en la región local de interés del sistema medido. Teniendo también la opción de procesar los datos guardados en otro programa.

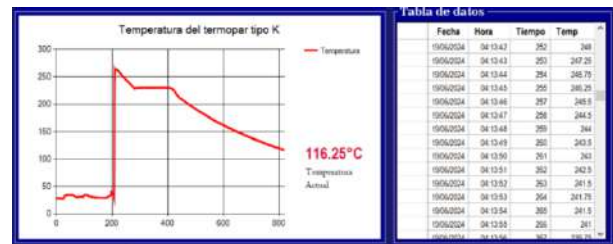


Fig. 11. Verificación de funcionamiento de la gráfica y tabla de datos. Fuente: Elaboración propia.

Para la fundición del material se utiliza el horno de alta temperatura 1400 Furnace de la marca Barnstead Thermolyne, donde se lleva el material a una temperatura de 400°C. El sistema de medición de temperatura desarrollado en este documento puede trabajar de forma integral con la GUI y el sistema DAQ, o en caso de no tener computadora el sistema DAQ puede adquirir datos de forma individual con la conexión a una fuente de voltaje y mantener comunicación con el usuario por medio de la pantalla LCD, con la opción de procesar los datos y obtener las gráficas fuera de línea en el momento que el usuario lo desee.

## V. RESULTADOS

Con los datos obtenidos del sistema de medición de temperatura para la curva de enfriamiento del estaño se grafican los valores que muestran la forma de meseta correspondiente a la temperatura de solidificación en Matlab, acotando los valores de tiempo y temperatura de acuerdo con la gráfica obtenida de la GUI. A su vez se gráfica la temperatura teórica del punto de fusión del estaño que es 231.9°C según [10] a modo de

comparación con los valores obtenidos experimentalmente, como se puede ver en la Fig. 13. La moda de los datos obtenidos en la región de solidificación es de 230°C.

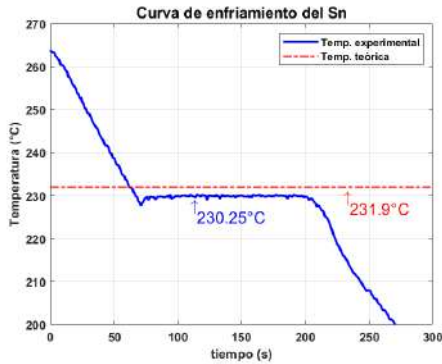


Fig. 12. Curva de enfriamiento del estaño en comparación con la temperatura de su punto de fusión. Fuente: Elaboración propia.

La obtención de los datos de temperatura por medio del sistema DAQ permite visualizar el comportamiento en la curva de enfriamiento del fenómeno de subfusión y recalentamiento, como se puede ver en la Fig. 14, en donde para esta prueba se obtiene un valor de temperatura mínimo de 227.75°C.

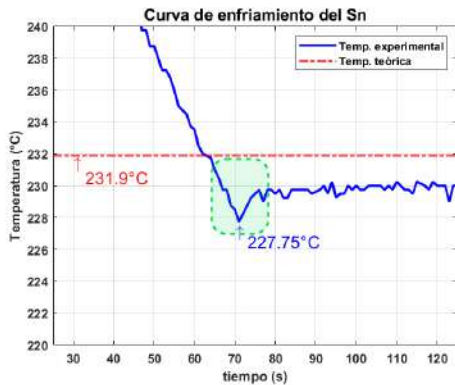


Fig. 13. Identificación del fenómeno de subfusión para el estaño. Fuente: Elaboración propia.

Un comparativo de temperaturas teóricas y temperaturas experimentales realizadas con el sistema desarrollado para el punto de fusión en materiales puros y aleaciones Sn-Zn, se presenta en la Tabla II.

Tabla II. Comparativo de temperaturas de solidificación. Fuente: Elaboración propia.

Material	Temperatura teórica (°C)	Temperatura experimental (°C)
Sn	231.9	230
Zn	419.5	417
Sn - 40% Zn	198.5	192.5
Sn - 15% Zn	198.5	192.25

## VI. CONCLUSIONES

Se desarrolló un sistema DAQ de termopar tipo K que se sumerge en un sistema metalúrgico en estado líquido para la obtención de curvas de enfriamiento que servirán para la construcción de diagramas de fases. La GUI y el sistema DAQ enlazados mediante comunicación serial permiten la obtención de una gráfica, un valor de temperatura en tiempo real y un

histórico de datos en la GUI, y un archivo de datos .CSV en la memoria microSD.

Se logra que el sistema opere en dos modos: el primero es utilizar el sistema DAQ e interactuando con el usuario a través de la GUI, en donde se muestra la gráfica en tiempo real en la computadora del usuario, el segundo modo es únicamente utilizar el sistema DAQ conectado a una fuente de voltaje de corriente directa e interactuar con el usuario mediante la pantalla LCD, esta ventaja permite procesar los datos tanto en línea; como fuera de línea con otro programa.

El sistema desarrollado es considerado funcional y competitivo en comparación de sistemas DAQ de temperatura de tipo comercial, como el USB-TC del fabricante JM Industrial, debido a que se logra tener un sistema de hasta 10 veces menor costo que los comerciales y un sistema de código abierto, con la misma precisión y confianza. Como trabajo futuro, se contempla que el sistema integre la tecnología de comunicación IoT para tener acceso al sistema desde la computadora de forma remota.

### AGRADECIMIENTO

Se agradece a la Universidad Nacional Autónoma de México por su financiamiento parcial a este proyecto, con el PAPIME PE104224 y PAPIIT IA102323.

### AGRADECIMIENTO

- Casals, Ma. Concepción Merino. "Diagramas y transformaciones de fase. 8. Métodos experimentales para la determinación de diagramas de equilibrio." REDUCA 4.3 (2012).
- Ospino, Reinaldo Almario, Bladimir Azdrubal Ramón Valencia, and Jacipit Alexander Ramón Valencia. "Sistema de adquisición de datos para el monitoreo de la calidad del agua a través de las variables de pH, conductividad, temperatura y oxígeno disuelto." Revista Colombiana de Tecnologías de Avanzada (RCTA) 1.17 (2013).
- Azúa-Barrón, Martín, et al. "Sistema de adquisición de datos de bajo costo con la plataforma Arduino." Revista mexicana de ciencias agrícolas 8.1 (2017): 1-12.
- Marruego-Carreazo, Daniel Ernesto, Gustavo Guerrero-Gómez, and July Andrea Gómez-Camperos. "Desarrollo de instrumento virtual enfocado en la adquisición de datos para generar perfiles de temperaturas en hornos." Revista Ingenio 8.1 (2015): 47-58.
- "Tarjeta de adquisición de datos de temperatura USB-TC - Logicbus". Logicbus: Tu Sitio De Automatizacion. Accedido el 21 de junio de 2024. [En línea]. Disponible: <https://www.logicbus.com.mx/tarjeta-daq-temperatura-usb-tc.php>
- "Tipos de termopares" Termokew S. A. de C. V. Accedido el 21 de junio de 2024. [En línea]. Disponible: <https://termokew.mx/images/termopares/Tipos-de-termopares.pdf>
- Accedido el 21 de junio de 2024. [En línea]. Disponible: <https://www.geekfactory.mx/producto/max6675-modulo-interfaz-para-termopar-tipo-k/>
- "shield micro sd". WJ Componentes Eletrónicos. Accedido el 21 de junio de 2024. [En línea]. Disponible: <https://www.wjcomponentes.com.br/shield-micro-sd>
- "DS3231 módulo reloj en tiempo real RTC - Geek Factory". Geek Factory. Accedido el 21 de junio de 2024. [En línea]. Disponible: <https://www.geekfactory.mx/producto/ds3231-modulo-reloj-en-tiempo-real-rtc/>
- "Tin - Element information, properties and uses | Periodic Table". The Royal Society of Chemistry. Accedido el 21 de junio de 2024. [En línea]. Disponible: <https://www.rsc.org/periodic-table/element/50/tin>