ISBN: 978-607-9394-23-3

9 786079 394233

DISEÑO COLABORATIVO EN MECATRÓNICA

Juan Manuel Ramos Arreguín
José Emilio Vargas Soto

Diseño Colaborativo en Mecatrónica

1ª Edición

Ramos Arreguín J.M.
y Vargas Soto J. E.

“Diseño Colaborativo en Mecatrónica”
Ramos Arreguín Juan Manuel
Vargas Soto José Emilio

© Asociación Mexicana de Mecatrónica A.C.
Prolongación Corregidora Norte No. 168, interior E
Col. El Cerrito, C.P. 76160, Querétaro, Qro.
México.

Internet: http://www.mecamex.net

1ª Edición, 07 de junio, 2021.

ISBN: 978-607-9394-23-3

Derechos reservados.
Esta obra es propiedad intelectual de sus autores y los derechos de publicación han sido legalmente transferidos a la editorial. Las opiniones y la información que se muestran en los capítulos del libro son exclusivas de los autores de cada capítulo y no representan la postura de la Asociación Mexicana Mecatrónica A.C. Prohibida su reproducción total o parcial por cualquier medio sin la autorización por escrito del propietario de los derechos del copyright.

Impreso en México – Printed in Mexico

Diseño Colaborativo en Mecatrónica, Asociación Mexicana de Mecatrónica A.C., 2021.
ISBN: 978-607-9394-23-3.

Índice

Capítulo 1 — 1
Control de un sistema electrónico de iluminación para invernadero
Heredia Castillo Jorge, Michua Antonio Camarillo, Gutiérrez Arias J. E. Moisés y Flores Mena J. Eladio.

Capítulo 2 — 15
Control difuso por rangos de operación de un convertidor buck, para alimentación de alambres musculares
Mier Quiroga Luis Antonio, Maldonado Onofre Daniel, Rodríguez Nava Benito y Coutiño Moreno Elvis.

Capítulo 3 — 28
Control de seguimiento articular en un exoesqueleto de 4 grados de libertad para miembros inferiores
M. Zamora, Angel Rodriguez-Liñan, Griselda Quiroz-Compean y Luis M. Torres-Treviño.

Capítulo 4 — 36
Diseño y simulación de un exoesqueleto para la articulación del codo
Centeno Barreda Daniel, Palomino Merino Amparo Dora y López Gutiérrez Jesús Ricardo.

Capítulo 5 — 52
Mapeo y Localización Simultánea (SLAM) con un robot tipo esfera mediante un enfoque minimalista utilizando ROS
Olalde Mendoza Omero Nicolás, Nava Balanzar Luciano, Guillen Soto José María, Jiménez Oronia Francisco y Rodríguez Olivares Noé Amir.

Diseño Colaborativo en Mecatrónica, Asociación Mexicana de Mecatrónica A.C., 2021.
ISBN: 978-607-9394-23-3.

Índice

Capítulo 6 68
Implementación de la cinemática inversa del robot Bioloid Premium
Ortega Palacios Miguel Angel, Palomino Merino Amparo Dora y Reyes Cortés José Fernando.

Capítulo 7 88
Aplicación de un controlador difuso y un PID clásico a la estación de nivel ETLFIE I
Charre Ibarra Saida Miriam, Gudiño Lau Jorge, Alcalá Rodríguez Janeth Aurelia, Duran Fonseca Miguel Ángel y Chávez Montejano Fidel.

Capítulo 8 102
Control de un robot de transporte personal Segway con retroalimentación aproximada del vector de estados.
González Pineda Carlos, Diaz Téllez Juan, García Ramírez Rubén Senén, Estévez Carreón Jaime y Pérez Pérez Jairo.

Capítulo 9 113
Diseño de un sistema inteligente para la recomendación de terapias de rehabilitación de fractura de tobillo
Cotri-Melecez Berenice, Quiroz-Compean Griselda, Torres-Treviño Luis y Rodriguez-Liñan Angel.

Capítulo 10 126
Manejo de LED’s WS2812: Un caso de estudio para integrar código ensamblador en Arduino
Santiago Espinosa Felipe y Barahona Ávalos Jorge Luis.

Diseño Colaborativo en Mecatrónica, Asociación Mexicana de Mecatrónica A.C., 2021.
ISBN: 978-607-9394-23-3.

Índice

Capítulo 11 141
Detección de movimiento mediante procesamiento de imágenes aplicada a un sistema de seguridad
Aldo-Francisco Muñoz-Vargas, Juan-Manuel Ramos-Arreguín, Saúl-Tovar Arriaga, Marco-Antonio Aceves-Fernández, Jesús-Carlos Pedraza-Ortega y Edgar Alejandro Rivas Araiza.

Capítulo 12 154
Análisis de la marcha humana implementando la cinemática de las articulaciones de la cadera y la rodilla
Lezama-Quintero Luis Arturo, Gómez-Pavón Luz del Carmen y Arriaga-Arriaga Cesar Augusto.

Capítulo 13 167
Una comparación cuantitativa entre el diseño clásico y automático (AutoML) de modelos de aprendizaje profundo en tareas de clasificación
Ángel de Jesús Osorio Guevara y Sebastián Salazar Colores.

Capítulo 14 185
Protección contra Vibraciones Fuera de Rango en Estructuras de 1 gdl
Candia García Filiberto, Ambrosio Lázaro Roberto A., Vivaldo de la Cruz Israel, Flores Méndez Javier y Benítez Cruz Gerardo A.

Capítulo 15 203
Sistema para la medición de nivel sin contacto basados en ultrasonido
Guillen Soto José María, Jiménez Oronia Francisco, Olalde Mendoza Omero Nicolas y Rodríguez Olivares Noe Amir.

Diseño Colaborativo en Mecatrónica, Asociación Mexicana de Mecatrónica A.C., 2021.
ISBN: 978-607-9394-23-3.

Índice

Capítulo 16 216
Propuesta de prototipo de ventilador emergente a partir de una bolsa de reanimación
Rivera Leyva Jossie E., Guzmán-Quezada Erick E., Napoles Medran José M., Schiavone Valdez Sergio D. y Curiel Anaya Jorge R.

Capítulo 17 227
Diseño de un péndulo invertido sobre dos ruedas como prototipo de prácticas para Control Automático
Araguz-Lara Paulina, Flores-Gómez Amalia, Garza-Martínez Mateo y Orrante-Sakanassi Jorge Alberto.

Capítulo 18 246
GIGA5 Vaccine: Un software para la vacunación masiva en todas las etapas de la vida
Navarro Álvarez Thiago Andrés y Navarro Parra Sandra Liliana.

Diseño Colaborativo en Mecatrónica, Capítulo 1, pp. 1 - 14
ISBN: 978-607-9394-23-3, 2021.

Control de un sistema electrónico de iluminación para invernadero

Heredia Castillo Jorge✉, Michua Antonio Camarillo, Gutiérrez Arias J. E. Moisés, Flores Mena J. Eladio

Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla, México.
✉herediacastillojorge@gmail.com

Resumen

En este documento se muestra una propuesta para la construcción de un sistema electrónico para controlar la intensidad de luz suplementaria de 10 lámparas LED comerciales de crecimiento. Este sistema se diseñó para aumentar el crecimiento de biomasa seca de raíz y biomasa seca de dosel de un cultivo de jitomate (Solanum lycopersicum L.) en una superficie de 10 m^2. El sistema consta de un dispositivo electrónico compuesto por un módulo de potencia especialmente diseñado para el sistema y un módulo de control. El módulo de control se implementó con una placa de desarrollo Arduino mega programada con un control clásico proporcional. Se demuestra el funcionamiento del sistema para el estado de Puebla, México, en un invernadero de 32 m^2 con otras condiciones físicas necesarias para el crecimiento invariables y una cantidad variable de iluminación, durante un periodo de 80 días mediante una simulación en Matlab utilizando 3 ecuaciones del modelo matemático TOMGRO.

.

Palabras clave: Tomate, Luz suplementaria, Invernadero, Control proporcional, Lámpara led.

1. Introducción

Las plantas son una importante fuente de nutrición para los seres humanos. Los cambios climáticos, plagas y enfermedades en las plantas han causado una disminución en el rendimiento de los cultivos, junto con esto, el uso excesivo de pesticidas ha motivado la elección de plantar en ambientes controlados.

El concepto de agricultura en interiores y sistemas de producción de plantas en entornos controlados ha surgido como un medio alternativo fiable y sostenible de producción de cultivos. La iluminación adecuada es esencial para el crecimiento óptimo de las plantas en interiores [1].

Actualmente, hay docenas de lugares de producción de cultivos interiores, en algunos países, por ejemplo: Japón, China y los Países Bajos, etc. [2]. Una técnica ampliamente utilizada para aumentar la producción de cultivos es la iluminación suplementaria de invernaderos [3], [4], que consiste en suministrar la cantidad necesaria de luz a los cultivos, ya que tanto exceder como caer por debajo de los niveles recomendados de luminosidad alteran la fotosíntesis [5]. Por esta razón, se muestra una propuesta de un circuito dispositivo electrónico para aumentar el rendimiento del cultivo de tomate mediante la manipulación de la intensidad de la luz de una bombilla con tecnología LED comercial de espectro completo.

El modelo matemático TOMGRO reducido [6] se utilizaron como simulador de los resultados del sistema en la cosecha de un invernadero durante un período de 80 días. Este modelo fue validado por el Colegio de Postgraduados del estado de Puebla mediante la construcción de una estación meteorológica ubicada en la Sierra Norte de Puebla.

Los principales parámetros del modelo se midieron durante un período de cosecha, obteniendo su comportamiento estadístico. Posteriormente se llevó a cabo la simulación del modelo matemático del cultivo con el modelo reducido TOMGRO de 3 ecuaciones diferenciales. Al comparar los datos simulados con los estadísticos se obtuvo una coincidencia entre los datos con un error pequeño. Después, se calibraron los parámetros del modelo para reducir el error de coincidencia aún más. Los valores de los parámetros calibrados fueron proporcionados a este trabajo gracias a la colaboración entre el Colegio de Postgraduados del Estado de Puebla y la Benemérita Universidad Autónoma de Puebla.

2. Diseño y construcción del sistema de iluminación

Se diseñó un circuito que modifica el ángulo de fase para controlar la intensidad de la luz de una lámpara. La etapa de entrada que controla el ángulo de fase se implementó con un TRIAC MAC12NG y un dispositivo MOC3010. El pulso de disparo es generado por un microcontrolador PIC16F628A a partir de un circuito genérico de recepción de cruce por cero y un programa de retraso y disparo. En la figura 1 y 2 se muestra el circuito de detección de cruce por cero y el circuito de potencia; los marcadores P2 y P1 está conectados en pines de entrada/salida del microcontrolador.

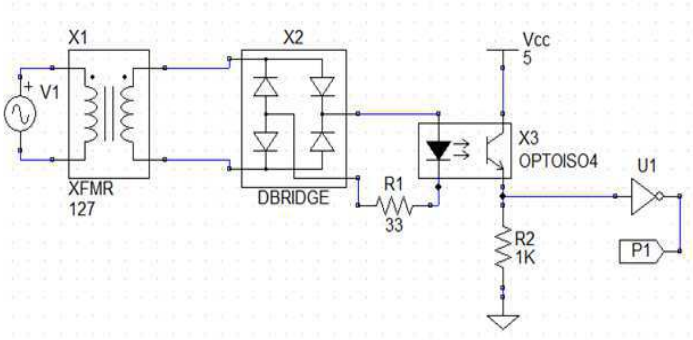


Figura 1. Circuito detector de cruce por cero.

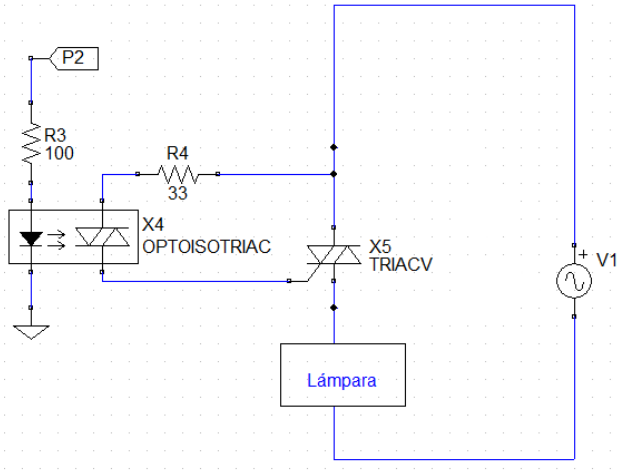


Figura 2. Circuito de potencia.

Desde un ordenador personal, el usuario envía el valor de intensidad de luz deseado. Una placa de desarrollo Arduino mega recibe los datos, los convierte al rango de 0 a 255 y los envía al circuito de control de fase. Se lee el puerto serie y se espera un cruce por cero. A continuación, se calcula un tiempo de retardo para la señal de salida de 127 v. Después del tiempo de retardo, se genera un pulso de 500 s que activa el TRIAC. En la figura 3 se muestra un diagrama de flujo que muestra el programa.

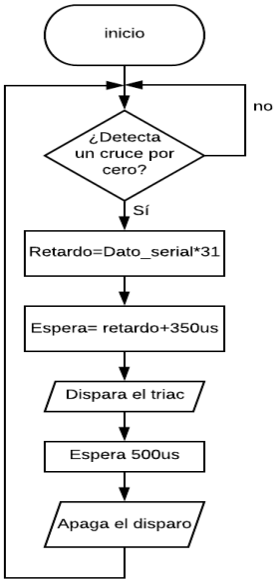


Figura 3. Diagrama de flujo del programa del microcontrolador.

2.1 Lámpara

La calidad de la luz y el crecimiento del cultivo guardan una relación con la iluminación en el crecimiento de la planta [7], siendo las lámparas tipo led, una buena opción de compra. Por ello, para la iluminación del invernadero, se utilizó una lámpara de crecimiento de 45 w de la marca Osunby para plantas de interior.

La lámpara tiene las siguientes características: Espectro completo de luz. De infrarrojos (730-735 nanómetros) a ultravioleta (390-395 nanómetros), un gran número de horas de vida útil (50000 horas), una fuente de alimentación convencional (127 V / 60 Hz).

.

El sensor TSL2591 se utilizó para detectar la iluminación de la lámpara. La lámpara se caracterizó usando el circuito de potencia, el sistema de control y el sensor de iluminación. La caracterización de la lámpara se hizo en un armario cerrado reflectante de 0.609 x 0.609 x 1.2 m. La lámpara se colocó a una altura de 0.6096 m del sensor. Se tomó el valor del sensor de iluminancia contra el valor del ángulo de disparo del sistema de potencia. En la figura 4, se puede ver el resultado de la caracterización. En la misma figura 4, se puede concluir que es posible controlar la lámpara de 2 a 3 radianes de fase.

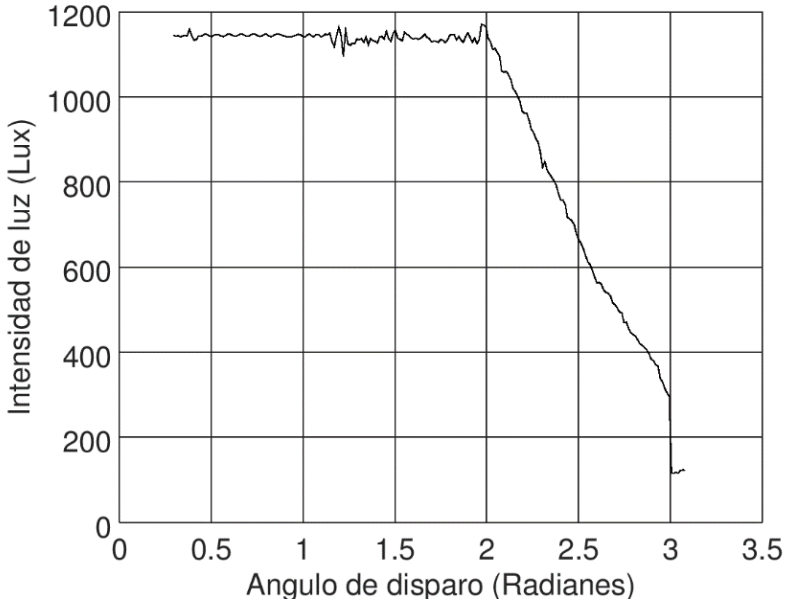


Figura 4. Caracterización de la lámpara (Gráfica ángulo de disparo/ luminosidad).

2.2 Control

La misma placa de desarrollo Arduino mega se utilizó como controlador proporcional. La figura 5 muestra el diagrama de bloques del sistema de control. La ganancia de control proporcional se ajustó por prueba y error hasta que el valor de 0,4. El algoritmo de programación se muestra en la figura 6.

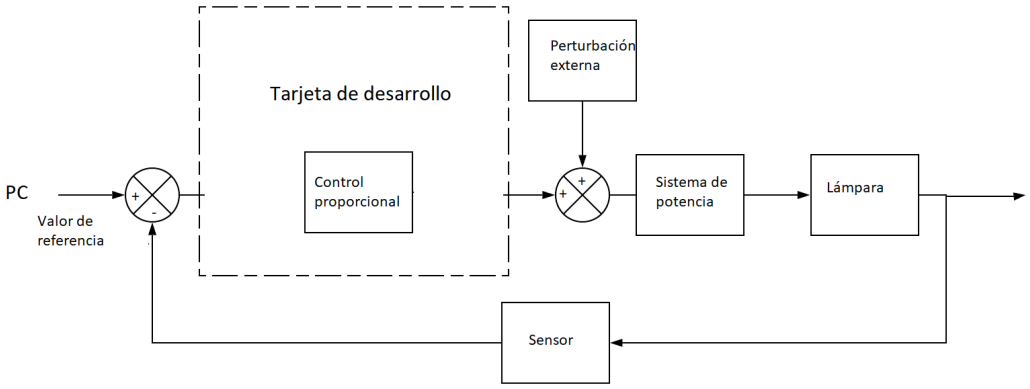


Figura 5. Diagrama de bloques del sistema de control.

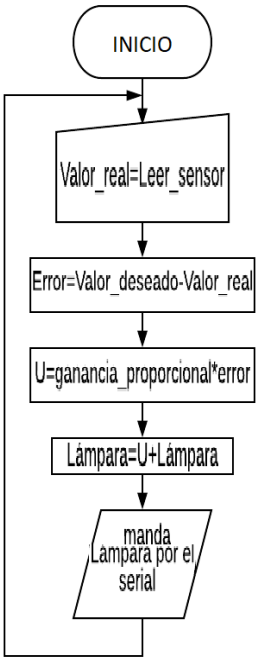


Figura 6. Diagrama de flujo del sistema de control.

El control alcanzó el punto deseado y se mantuvo con una variación aproximada de 15 lux. La respuesta del sistema ante un valor de referencia se muestra en la Figura 7.

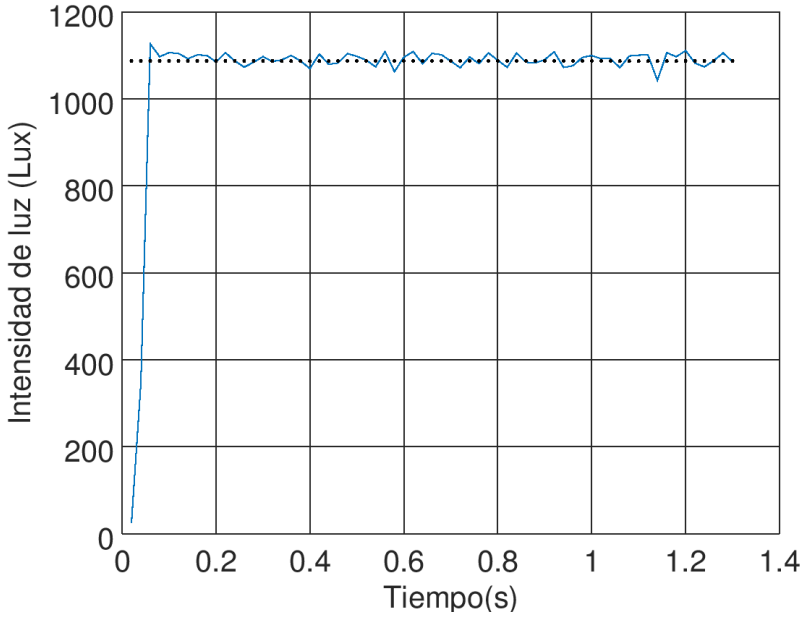


Figura 7. Gráfico del comportamiento del sistema de control.

Finalmente, se añadió un módulo de reloj en tiempo real DS3231 a la placa de desarrollo Arduino para encender automáticamente el sistema durante el día y apagarlo por la noche.

3. Implementación

El sistema se implementó en un invernadero de 8 metros de largo por 4 metros de ancho. El invernadero se ubica en la periferia sur de la ciudad de Puebla, en el estado de Puebla de México. El invernadero tiene un revestimiento de plástico y está apoyado por tubulares de acero comercial cuadrado con un revestimiento de pintura blanca. Los tubulares están unidos por tornillos, además el revestimiento se unió a los tubulares de la misma forma. Las 10 lámparas se instalaron en dos filas de 5 lámparas a lo largo del invernadero para iluminar 10 m^2 con iluminancia controlada de los 32 m^2 totales del invernadero. El sensor y el circuito electrónico fueron colocados en un extremo de este; justo debajo de una de las lámparas. Esto último fue posible gracias a que se realizó una conexión en paralelo de las 10 lámparas a la salida del único sistema de control. Es necesario aclarar que cuando la planta brote es necesario colocar el sensor en una hoja del tope del dosel.

Una imagen de la instalación dentro del invernadero de prueba se muestra en la figura 8. Dicha fotografía fue tomada en la tarde, aproximadamente entre 6 y 8 pm, donde la luz solar es casi nula.

Figura 8. Fotografía que muestra las conexiones de las lámparas dentro del invernadero de prueba.

4. Simulador de cosecha

Para hacer una simulación de la cosecha con los efectos del sistema de iluminación, se utilizó un modelo dinámico de 3 ecuaciones de estado derivado del modelo matemático TOMGRO. A continuación, se describen los elementos y términos del modelo usado.

4.1 Número de nodos

La velocidad de desarrollo del nodo se modela como una tasa de aparición máxima de nodos por día (Nm) multiplicada por una función que reduce el desarrollo vegetativo a temperaturas no óptimas cada día. La expresión que modela dicho fenómeno se muestra en 1.

$$\frac{d}{dt}N = N_m f_m(T_g) \tag{1}$$

En la tabla 1 se dan a conocer las unidades de las variables de la expresión de nodos. Las demás definiciones también serán acompañadas por una tabla que condensa las unidades correspondientes de las variables involucradas.

Tabla 1. Unidades de la expresión de nodos.

Símbolo	Unidades
N	Adimensional
Nm	Nodos d^{-1}
T_g	°C

, donde $f_{m(T_g)}$, Es la reducción del desarrollo vegetativo bajo temperaturas no óptimas. Y se describe por medio de la definición 2.

$$f_{m(T_g)} = \begin{cases} 0, \ Tg \leq 8°C \\ 0.55, \ Tg = 12°C \\ 1, \ Tg = 30°C \\ 0, \ Tg \geq 50°C \end{cases} \tag{2}$$

4.2 Materia seca del dosel

La materia seca del dosel se modela por medio de la expresión 3.

$$\frac{d}{dt}W_c = E\left(P_g - R_m(W_r + W_c)\right) f_c \tag{3}$$

Tabla 3. Unidades de la expresión de materia seca.

Símbolo	Unidades
W_c	$g\ tejido\ m^{-1}$
W_r	$g\ tejido\ m^{-1}$
R_m	(gCH_2O) $/(g\ tejido\ d)$
E	$(g\ tejido)$ $/(gCH_2O)$
P_g	$g\ tejido\ m^{-1}d^{-1}$
f_c	Adimensional

, donde el término P_g se refiere a velocidad de la fotosíntesis bruta en el dosel.

$$P_g = D\frac{\tau Cp(T_g)}{k} ln\left|\frac{\alpha k I_0 + (1-m)\tau C}{\alpha k I_0 e^{-kLAI} + (1-m)\tau C}\right| \quad (4)$$

Tabla 3. Unidades de la expresión 4.

Símbolo	Unidades
D	$(\mu mol\ CO_2)$ $/(m^2 s)$
τ	$(\mu mol\ CO_2)$ $/(m^2\ hoja\ s)$
C	$(\mu mol\ CO_2)$ $/(mol\ aire)$
$p(T_g)$	Adimensional
α	$(\mu mol\ CO_2)$ $/(\mu mol\ fotón)$
k	Adimensional
I_0	$(\mu mol\ fotón)$ $/(m^2 s)$
m	Adimensional
LAI	$(m^2\ hoja)$ $/(m^2 s)$

Por otro lado $p(T_g)$ se refiere a un factor de reducción de fotosíntesis, y está definido como:

$$p(T_g) = \left[1 - \left(\frac{\Phi_h - T_g}{\Phi_h - \Phi_1}\right)^2\right] \quad (5)$$

Tabla 4. Unidades de la definición 5.

Símbolo	Unidades
Φ_h	°C
Φ_1	°C

Además R_m es la velocidad de respiración de mantenimiento.

$$R_m = K_m e^{0.0693(T_g - 25)} \quad (6)$$

Tabla 5. Unidades de la expresión 6.

Símbolo	Unidades
K_m	°C

También I_0 es la densidad del flujo de luz en el tope del dosel.

$$I_0 = \begin{cases} I_m \sin(2\pi(t_d - 6)), & I_0 > 0 \\ 0, & I_0 \leq 0 \end{cases} \quad (7)$$

Tabla 6. Unidades de la definición 7.

Símbolo	Unidades
I_m	$(\mu mol\ fotón)/(m^2 s)$
t_d	d

Finalmente, el índice de área foliar es:

$$LAI = \frac{\rho\delta}{\beta}\ln\left(1 + e^{\beta(N-nb)}\right) \tag{8}$$

Tabla 7. Unidades de la expresión 8.

Símbolo	Unidades
ρ	$numero/m^2$
δ	$m^2 hoja/nodo$
β	$nodo^{-1}$
nb	$nodo$

4.3 Materia seca de raíz

La ecuación de materia seca de raíz es como la ecuación del dosel. La función que modela el crecimiento se presenta en la expresión 3.

$$\frac{d}{dt}W_c = E\left(P_g - R_m(W_r + W_c)\right)(1 - f_c) \tag{9}$$

4.4 Sistema dinámico integrado

El sistema dinámico completo del invernadero se muestra a continuación:

$$\begin{aligned} \frac{d}{dt}N &= N_m f_m(T_g) \\ \frac{d}{dt}W_c &= E\left(P_g - R_m(W_r + W_c)\right) f_c \\ \frac{d}{dt}W_c &= E\left(P_g - R_m(W_r + W_c)\right)(1 - f_c) \end{aligned} \tag{10}$$

Estas 3 variables de estado se eligieron del modelo TOMGRO original, que puede contener más de 109 variables. Para simular el sistema los valores numéricos del modelo TOMGRO se pueden encontrar en la literatura, sin embargo, para esta obra algunos de ellos fueron ligeramente modificados. Los parámetros calibrados a la región geográfica de ubicación del invernadero de prueba fueron proporcionados por el Colegio de Postgraduados del Estado de Puebla (COLPOS). Estos valores se presentan en la tabla 8.

Tabla 8. Parámetros.

Parámetro	Cantidad	Parámetro	Cantidad
N_m	0.021	E	0.7
D	0.108	τ	0.0664
C	350	α	0.056
m	0.1	Φ_h	30
I_m	800	ρ	4
Φ_1	5	β	0.38
δ	0.074	k	0.58
nb	0.38	-	-

4.5 Simulación

La simulación se realizó en MATLAB. La temperatura (T_g) se mantuvo en un valor constante de 18 °C por la noche y 23 °C durante el día, según lo recomendado por los especialistas [8].

$$T_g = \begin{cases} 18\,°C, & I_0 \leq 0 \\ 23°C, & I_0 > 0 \end{cases} \qquad (12)$$

La expresión 12 modela la temperatura ideal que debe de tener el invernadero, esto significa que cuando la densidad del flujo de luz en el tope del dosel (I_0) toma valores positivos; indica que es de día. Por lo tanto, la temperatura debe de ajustarse a 18 ° centígrados. Por otro lado, cuando I_0 no es positiva, entonces es de noche y la temperatura debe de ajustarse a 23° centígrados. Esto en la práctica se puede lograr mediante un control de temperatura. Para propósitos de esta simulación, se considera que la temperatura es ideal naturalmente.

Por otro lado, para modelar la luz se propone que la densidad fotosintéticamente activa resultante en el tope del dosel de la planta dentro del invernadero se rija por la expresión 13.

$$I_0 + I_{in} = \begin{cases} I_{in} & I_0 > 0 \\ 0 & I_0 \leq 0 \end{cases} \qquad (13)$$

, donde I_{in} representa la entrada de intensidad de luz del sistema de iluminación propuesto.

El significado de esta expresión indica que cuando la densidad de luz natural en el tope del dosel (I_0) es menor o igual a cero es de noche y el sistema está apagado. Por otro lado, cuando la densidad de flujo de luz natural (I_0) es mayor a cero, entonces eso significa que es de día y el sistema proporciona la densidad deseada (I_{in}). Esta se regula con el sistema de control para evitar dañar las hojas por exceso de luz, o bien si es que no hay suficiente, aumentar la intensidad de las lámparas para cumplir con la densidad deseada por el especialista a cargo del cultivo.

Para realizar el cálculo de la iluminación fotosintéticamente activa que produce el sistema de iluminación se debe tomar en cuenta que el máximo valor que puede entregar una lámpara a una altura de 0.6096 m es 1170 lux; como se muestra en la figura 4, por lo que cualquier valor de I_{in} podrá ser mayor a dicho valor convertido en unidades de radiación fotosintéticamente activas propias de las plantas. Esto último gracias a que la planta crece, y al crecer, disminuye la altura entre la superficie de la hoja y la lámpara, aumentando así, la densidad de flujo de luz de manera cuadrática. Además, la luz del sol aumenta dicho valor. Sin embargo, es justo mencionar que podría existir un espacio en donde

no se llegue al valor de densidad deseado. Por ejemplo, cuando el valor deseado sea elevado y la altura de la plántula no sea suficientemente grande para que el sensor detecte un valor de densidad igual o superior a la referencia. Si se presenta este problema, se propone que la lámpara se coloque inicialmente casi tocando la tierra y que se vaya subiendo la altura con respecto al suelo proporcionalmente a la altura del brote mientras la planta va creciendo hasta la altura deseada. Una vez llegada a la altura deseada se puede fijar la posición de la lámpara. Esta solución es posible gracias a que las lámparas de tecnología led son frías, es decir, a diferencia de otros sistemas de iluminación tradicionales no producen un aumento de temperatura que pueda quemar la hoja por cercanía. De hecho, esta última es la razón por la cual la tecnología led se proyectaba, desde 2008, como tendencia en lo referente a la iluminación de invernaderos [9].

La función MATLAB ode45 (Runge-Kutta 4/5) se utilizó para resolver el sistema de ecuaciones diferenciales número 10. Los valores iniciales fueron 0 para las variables nodos, materia seca del dosel y materia seca de la raíz. El tiempo total para la simulación fue de un periodo de cosecha normal, es decir 80 días, con puntos generados en la gráfica de salida cada 30 minutos. El valor deseado para fines demostrativos de la simulación se eligió de $I_{in} = 800\ \mu mol\ fotón/m^2 s$. El tiempo transcurrido entre el inicio de la integración numérica del sistema con iluminación natural fue de 3.9 s, el tiempo transcurrido entre el inicio y el final de la integración numérica del sistema con iluminación suplementaria fue de 1.83 segundos, la duración total de la simulación fue de 5.8 segundos. Como evidencia de la simulación se presenta en parte del código principal en lenguaje propio de Matlab.

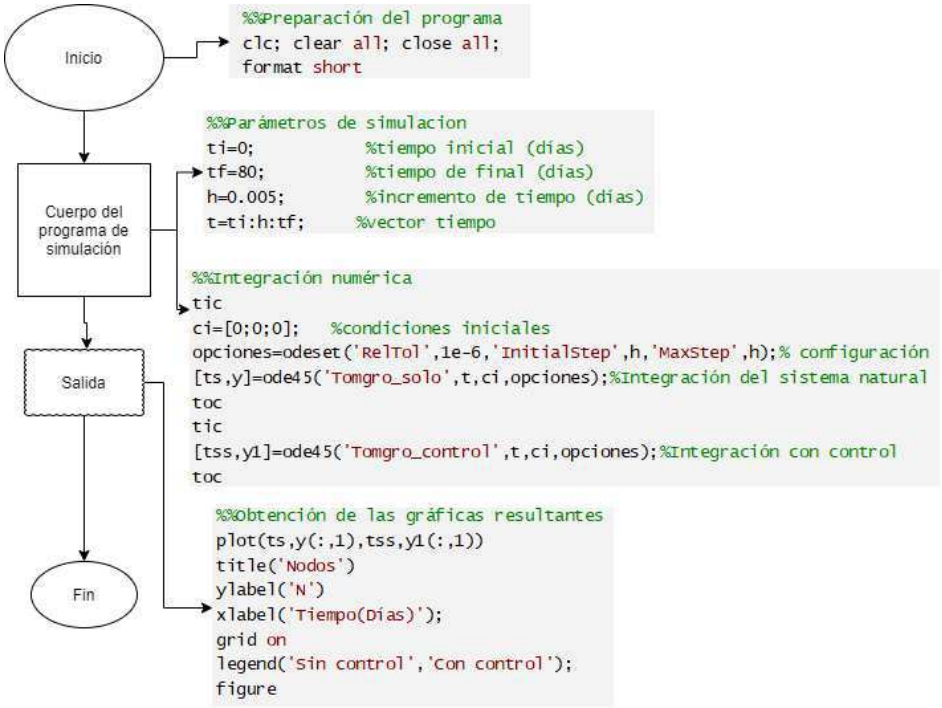


Figura 9. Parte del código principal en Matlab para realizar la simulación.

Se utilizaron dos archivos de función de apoyo para realizar la integración numérica, uno de ellos contenía la ecuación diferencial correspondiente al sistema dinámico del invernadero sin control de iluminación, el otro llevaba el sistema dinámico correspondiente al propuesto en este trabajo. Dentro de estas funciones de apoyo se utilizaron otras funciones de apoyo no lineales como la ecuación de desarrollo vegetativo por condiciones no óptimas. Esta se aproximó por interpolación. Otras funciones por ejemplo la expresión 12 y 13 se implementaron con ayuda de estructuras lógicas IF-ELSE.

5. Resultados

El crecimiento de la planta con luz natural se comparó con el crecimiento con el sistema de luz suplementario con el máximo valor posible. Los resultados del crecimiento máximo con el sistema propuesto se muestran:

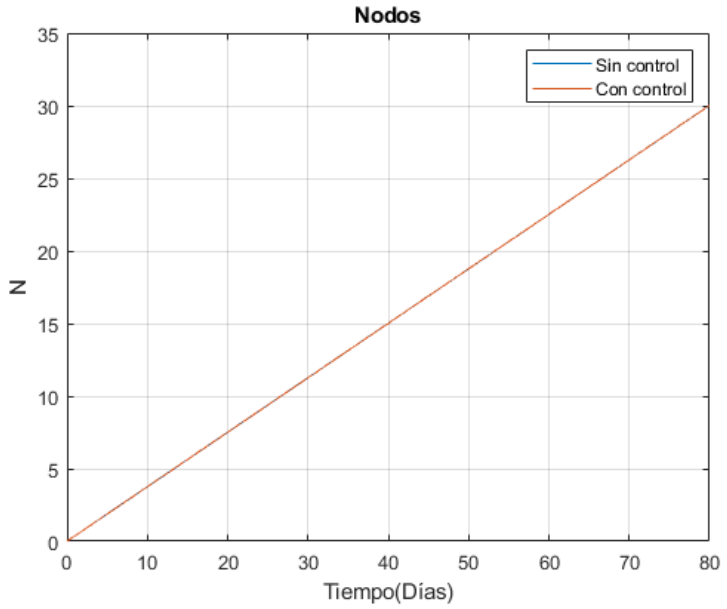


Figura 10. Comparación del incremento de nodos de un cultivo de invernadero frente a un cultivo con un sistema de luz suplementario.

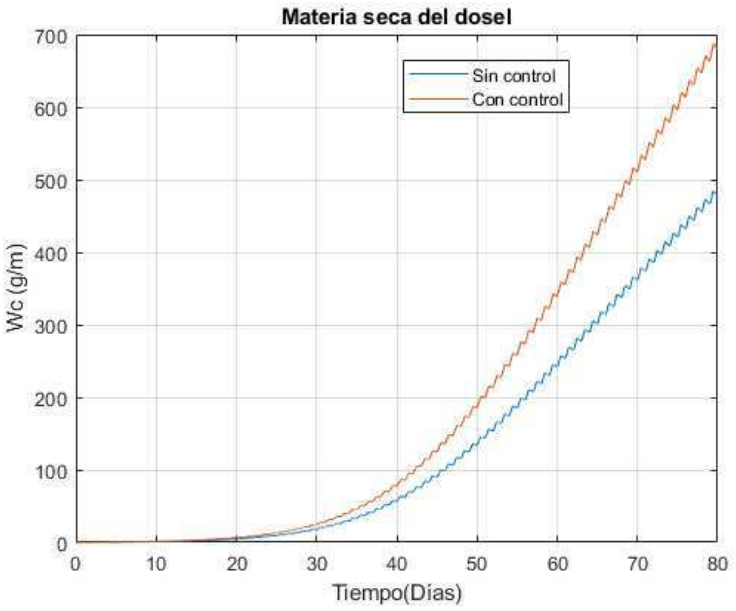


Figura 11. Comparación del aumento de la materia seca del dosel de un cultivo de invernadero, con respecto a un invernadero con un sistema de luz suplementario.

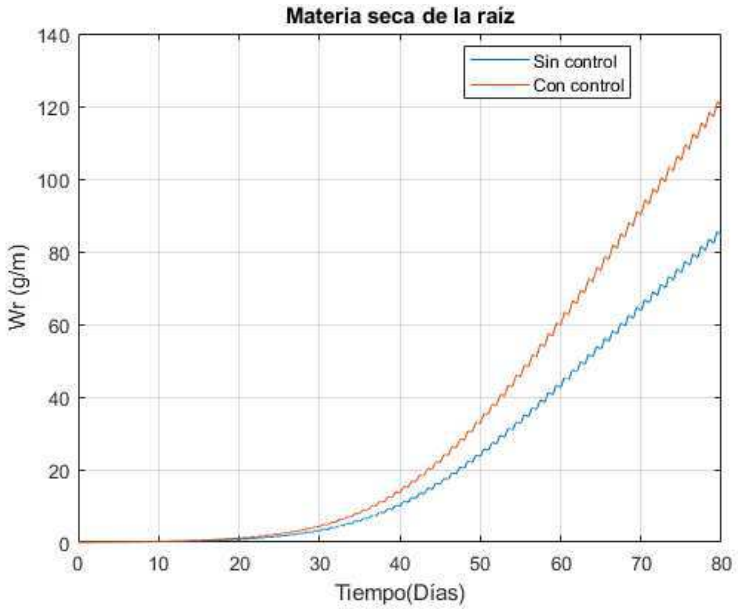


Figura 12. Comparación de materia seca de la raíz de un cultivo de invernadero, con respecto a un invernadero con un sistema de luz suplementario.

Después de obtener las gráficas se demostró el resultado del uso de un sistema de iluminación complementario como el propuesto en este escrito para la región centro sur del estado de Puebla con todas las otras condiciones físicas propicias para el crecimiento, durante un periodo de 80 días. Se puede ver que no existe un aumento en el número de nodos, pero existe un crecimiento de 35.51 $\frac{g}{m}$ en la biomasa seca de raíz y 201.432 $\frac{g}{m}$ en la biomasa seca del dosel. Lo que implica un aumento de más del 30 % en la producción de biomasa por cosecha. Aunque el resultado es una simulación, da una idea general del beneficio que puede tener adoptar un sistema de iluminación suplementaria. En la práctica, se debe de controlar la temperatura, la humedad, la cantidad de nutrientes, el porcentaje de CO_2, etc., con las mismas condiciones de la simulación para obtener resultados parecidos a los obtenidos en este trabajo.

Ahora bien, si se incrementa la biomasa del dosel y la biomasa de la raíz significa que aumenta el peso seco de la capa superior de hojas y el peso seco de la raíz de cada planta de jitomate por metro lineal de altura. Como es bien sabido, la biomasa de los frutos está directamente relacionados con el número de nodos [10]. Por lo tanto, al no existir un aumento en el número de nodos se puede asegurar que no se obtendrá un aumento en el número de frutos o en la biomasa de los frutos.

6. Conclusiones

Con el uso de un sistema de iluminación suplementaria en un invernadero sometido a las mismas condiciones del trabajo planteado se produce un aumento en la biomasa seca del dosel y en la biomasa seca de la raíz del cultivo de jitomate con respecto a una cosecha con las mismas condiciones, pero solo iluminada naturalmente. Este crecimiento está en función, y es directamente proporcional al nivel de referencia de radiación PPFD deseado. Esta conclusión es lógica, debido a que la ecuación de materia seca de raíz y materia seca del dosel depende implícitamente de la radiación fotosintéticamente activa.

Debido a que la ecuación de nodos no depende de la radiación fotosintéticamente activa, no es posible obtener un incremento en el número de nodos al utilizar un sistema de iluminación suplementaria. Al no existir un aumento en el número de nodos no se espera un aumento en el número o en la biomasa de los frutos al usar este sistema de iluminación.

.

Referencias

[1] Avinash A, Dutta S. "*Impacto de la emisión de luz diodos (leds) y su potencial en el crecimiento de la planta y desarrollo en la producción de plantas de ambiente controlado''*, Current Biotechnology, Bentham Science Publishers, Vol 5, No. 1, 1-16, 2016.

[2] Liu, W. *"Light Environmental Management for Artificial Protected Horticulture*". Agrotechnol, 1, 1-4, 2012.

[3] Li, Y., Niu, P., y Su, Z. "*Diseño del sistema de monitoreo y control de invernaderos basado en iluminación led.*", 12º Foro Internacional de China sobre Iluminación de Estado Sólido, SSLCHINA, 123-126, China, 2015.

[4] Yanhui, W. y Xiaofei, J. "*El diseño de sistema de control de iluminación de invernadero*". En la 27ª Conferencia de Control y Decisión, CCDC 2015, 2613-2617, China, 2015.

[5] Murchie E. H, Niyogi K. K, "*Manipulation of Photoprotection to Improve Plant Photosynthesis*", Plant Physiol, Vol 155, 1, 86-92, 2010.

[6] Córdoba G. E.J, Gutiérrez A. J.M. "*Control óptimo para el control de temperatura en el cultivo de tomate en invernadero basado en un sistema dinámico*". Pistas educativas., Vol 39, 125, 83-95, México, 2017.

[7]. Olle, M. y Vir_sile, A. "*Los efectos de la iluminación de diodos emisores de luz sobre el crecimiento y la calidad de las plantas de invernadero*". Ciencias Agrícolas y Alimentarias, 22(2), 2013

[8]. Velasco H. E, Nieto Á. R, Navarro E., "*Cultivo del tomate en hidroponía e invernadero*". bba, México, 1, 2010.

[9]. MASSA, G. et al. "*Plant productivity in response to LED lighting. HortScience"*, vol. 43, no 7, p. 1951-1956, 2008.

[10]. Hernández, F, et al. "*Simulación del crecimiento y desarrollo de pimiento (Capsicum annumm L.) bajo condiciones de invernadero"*. Revista mexicana de ciencias agrícolas, 2011, vol. 2, no 3, p. 385-397.

Diseño Colaborativo en Mecatrónica, Capítulo 2, pp. 15 - 27
ISBN: 978-607-9394-23-3, 2021

Control difuso por rangos de operación de un convertidor buck, para alimentación de alambres musculares

Mier Quiroga Luis Antonio, Maldonado Onofre Daniel, Rodríguez Nava Benito, Coutiño Moreno Elvis

Tecnológico de Estudios Superiores de Jocotitlán, Estado de México, México.
luis.mier@tesjo.edu.mx

Resumen

El presente trabajo muestra la propuesta de un sistema de alimentación para actuadores basados en materiales con memoria de forma, en especial los conocidos como alambres musculares. El constante avance tecnológico exige en que ciertas aplicaciones, como del ámbito automotriz, aeroespacial, industria de extracción de gas y petróleo, sistemas micro electromecánicos, robótica, medicina, entre otras, los actuadores sean del menor tamaño posible, por lo que los alambres musculares son idóneos para dichos sistemas. Asimismo, los alambres musculares tienes grandes ventajas, entre las que destacan que son confiables, reducidos en peso y volumen, cuentan con gran relación fuerza-peso y su operación es silenciosa, comparados con actuadores convencionales para el mismo propósito. Para que un alambre se contraiga debe ser calentado por cualquier medio, la forma mas sencilla de hacerlo es hacer pasar una corriente a través de él. Sin embargo, si dicha corriente no es suficiente el alambre no ejercerá la fuerza requerida, pero si es mayor a la magnitud recomendada por el fabricante, el alambre se sobrecalentará, deteriorándolo e incluso podría llegar a la destrucción total. El sistema que se detalla en este trabajo, consta de un convertidor buck acompañado de un controlador difuso por rangos de operación, que permiten el flujo de corriente adecuado en el alambre muscular, asegurando su correcto funcionamiento y garantizando la vida útil indicada por el fabricante.

Palabras clave: Alambre muscular, memoria de forma, control difuso, buck.

1. Introducción

El desarrollo tecnológico está forzando el avance hacia el diseño de sistemas con funciones y características inteligentes. Gran parte de ellos incluyen, además de microcontroladores y capacidades de comunicación inalámbrica, sensores y actuadores. Existen ciertas aplicaciones, donde por restricciones técnicas y comerciales, tanto el peso como el tamaño del actuador no deben exceder cierto valor, por lo que los alambres musculares o materiales con memoria de forma se han convertido en una excelente solución [1], [2].

Los alambres musculares son aleaciones de metales que tienen la capacidad de memorizar o mantener su forma previa cuando son sometidos a ciertos estímulos, principalmente variaciones magnéticas, térmicas o termomagnéticas [2 - 4]. Los materiales con memoria de forma fueron descubiertos en 1932 por Arne Ölander y desde entonces se han desarrollado diversos trabajos de investigación, así como patentes, donde se usan como actuadores. Sus aplicaciones abarcan diversas áreas, destacando aquellas en el ramo automotriz, aeroespacial [1], en la industria de la extracción del gas y petróleo [5], sistemas micro electromecánicos o MEMS y robótica [6 - 9], así como en medicina [10], entre otras.

Los alambres musculares son simples, confiables, reducidos en peso y volumen, cuentan con gran relación fuerza-peso, operación silenciosa y tiempo de vida relativamente amplio, comparados con actuadores convencionales para el mismo propósito [10 - 12]. En especial, su vida útil está ampliamente

relacionada con su buen manejo. Es posible deformar y contraer el alambre muscular térmicamente mediante el efecto Joule, pasando una corriente a través de él. Lo anterior implica que, aparte de que se debe tener una fuente de energía con gran capacidad de corriente, se debe controlar su magnitud a través del alambre muscular [13]. De tal forma que, si la magnitud de la corriente no es suficiente el alambre no se contraerá y si excede la máxima recomendada por el fabricante, se puede ver reducida su vida útil drásticamente, incluso puede destruirse [1], [4], [8], [14].

Gran parte de los trabajos publicados sobre alambres musculares no proporcionan información sobre las características de la fuente de alimentación utilizada. Asimismo, a excepción de que la investigación se enfoque en controlar su posición o fuerza, no indican de qué forma aseguran que la magnitud de la corriente que fluye a través del alambre no sea mayor a la máxima recomendada por el fabricante. Con el propósito de mantener su vida útil, en este trabajo se utiliza un convertidor buck como fuente de alimentación. Para asegurar que la corriente que pasa a través del alambre muscular sea adecuada, se establece un control difuso para su control. La particularidad de este control es que trabaja con tres conjuntos de funciones de pertenencia de la variable de salida. De acuerdo al valor máximo de la corriente que soporte el alambre y se indique al controlador, éste elegirá que conjunto utilizar en el proceso de desdifusificación para establecer el valor adecuado del ciclo de trabajo del convertidor buck.

Un controlador de tipo difuso basa su funcionamiento en los sistemas basados en reglas para la toma de decisiones y en la lógica difusa para la evaluación de ellas. Es una alternativa para gran variedad de retos en el control aplicado. Es un método práctico que permite realizar controladores no lineales mediante el uso de información obtenida por la experiencia [15].

La organización del presente trabajo es la siguiente; inicialmente se presenta la teoría básica de los alambres musculares, el convertidor buck y del control difuso. Enseguida se detalla el sistema de alimentación propuesto. Posteriormente, se muestran diversos casos de análisis y estudio del sistema propuesto, utilizando la herramienta Simulink, de Matlab. Finalmente, se discuten los resultados obtenidos y se exponen las conclusiones.

2. Alambres musculares

Los científicos conocían los materiales con memoria desde 1932 e incluso habían experimentado con ellos. La primera vez que se utilizó el término “memoria de forma” fue en 1941 [4]. Su importancia fue reconocida hasta 1962, cuando William Beuhler y Frederick Wang presentaron sus reportes de trabajo con aleaciones de niquel y titanio. El equipo científico intentaba desarrollar una aleación resistente al calor y a la corrosión. En el proceso de obtenerla, crearon una aleación con memoria de la forma, hecha con 55% de níquel y 45% de titanio. El equipo bautizó a su nueva aleación como Nitinol, que representa los componentes y el lugar de origen. Las letras "NI" y "TI" son los símbolos atómicos para el níquel y el titanio, "NOL" se refiere al laboratorio naval en donde fue descubierto (Naval Ordinance Laboratory) [1]. Actualmente, la mezcla de níquel y titanio está hecha de partes casi iguales de uno y otro [4].

Es posible hacer que la aleación recuerde una forma en particular. Una vez que recuerda esa forma, a la aleación se le puede hacer volver a la forma original calentándola por encima de una temperatura "de transición", generando una fuerza considerable. Por ejemplo, una pulgada cuadrada de material de nitinol genera una fuerza de retorno a la forma de 30 PSI [11]. Esta característica física se basa en su estructura cristalina dinámica y sensible al calor. Cuando el nitinol está deformado, la estructura cristalina no se destruye, sino que se transforma, cambiando a una singular disposición cristalina. Cuando el material se calienta vuelve a su estructura recordada, sujeta a una menor tensión. Por otro lado, en los metales normales, las deformaciones hacen que la estructura molecular se deshaga, dejando a los átomos en nuevas posiciones cristalinas. A causa de esto, el cristal no puede conservar una memoria de dónde estaban los átomos antes de moverse. El movimiento físico del nitinol se debe a la reestructuración interna de las moléculas, generando que sea muy potente [4].

El alambre Flexinol, como el ejemplo que se muestra en la figura 1, es una marca registrada. Es una aleación de níquel y titanio que se ha entrenado para que funcione como actuador. Si se le utiliza apropiadamente, el alambre puede rendir 1,000,000 ciclos. Se puede hacer contraer el alambre hasta reducir su longitud, en algunos modelos en un porcentaje de hasta 10% de su tamaño total. Cuando se reduce la longitud de un alambre, el volumen absoluto de metal continúa siendo el mismo. Por esta razón, a menor longitud, crecerá el diámetro. La contracción y relajación depende de la temperatura del alambre de aleación de nitinol. El alambre tiene una resistencia eléctrica, al fluir corriente eléctrica a través de ella, produce con rapidez un calor suficiente como para llevar al alambre a su temperatura de transición. Cuando se permite que el material se enfríe, el alambre se puede estirar con facilidad hasta su longitud original [4]. El alambre usualmente se utiliza aplicando una fuerza de tensión opuesta a la dirección de su contracción. Esta fuerza repone al alambre a su longitud original en la fase de baja temperatura. A esta fuerza aplicada se le llama fuerza de extensión. Si se lleva un alambre a su temperatura de transición sin una fuerza de extensión, se contraerá, pero cuando se enfríe no retornará a su longitud original. En consecuencia, ante la ausencia de una fuerza de extensión, cuando el alambre se caliente de nuevo no habrá una nueva contracción [6].

Figura 1. Presentación y datos de un fabricante de alambres musculares.

El tiempo de reacción de los alambres musculares puede ser reducido, en el orden de los milisegundos utilizando pulsos de corriente elevada. Cuando se hace esto, se debe tener en cuenta la masa y la velocidad del material que se va a mover. Cuanto más rápido se mueva una determinada masa, más grande será la inercia a vencer. Si la inercia es mayor a la resistencia a la tracción del alambre, éste se cortará [16].

3. Convertidor buck

Un convertidor buck produce un voltaje promedio de salida con una magnitud menor a la del voltaje de cd de entrada. Asimismo, en la misma razón que reduce el voltaje a la salida, incrementa la magnitud de la corriente disponible en la resistencia de carga *R*. Su circuito básico es como el que se muestra en la figura 2.

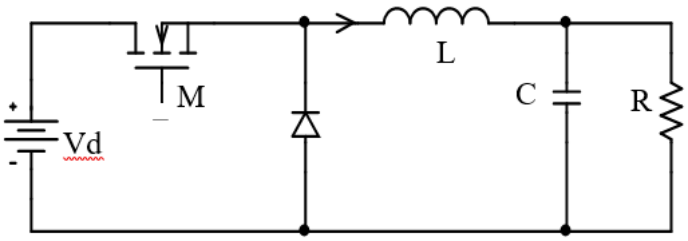


Figura 2. Convertidor buck.

Cuando el interruptor *M* se encuentra cerrado durante un tiempo t_{on}, el voltaje de entrada *Vd* aparece en los extremos del diodo y por lo tanto en un extremo del inductor. Cuando el interruptor es abierto, durante un tiempo t_{off}, el voltaje a través del diodo puede considerarse cero ya que este se encuentra polarizado directamente debido a que la corriente almacenada por el inductor fluye a través de éste, teniendo una forma de onda de la corriente en el inductor como se muestra en la figura 3 [17].

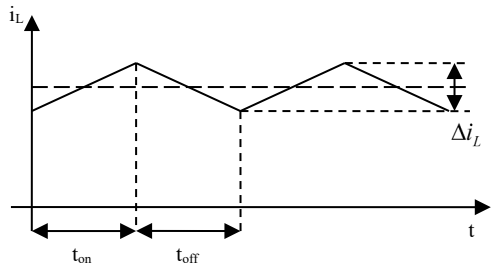


Figura 3. Forma de onda de la corriente en el inductor.

El voltaje de salida promedio *V*o, puede ser determinado de la siguiente forma:

$$\mathrm{Vo} = \frac{1}{\mathrm{Ts}}\int_0^{\mathrm{Ts}} \mathrm{Vo(t)dt} = \frac{1}{\mathrm{Ts}}\int_0^{\mathrm{t_{on}}} \mathrm{Vddt} = \frac{\mathrm{Vd}}{\mathrm{Ts}}\mathrm{t_{on}} \tag{1}$$

Si se define al ciclo de trabajo,

$$\mathrm{D} = \frac{\mathrm{t_{on}}}{\mathrm{Ts}} \tag{2}$$

entonces, la ecuación 1 queda de la siguiente forma:

$$Vo = DVd \qquad \mathrm{Vo} = \mathrm{DVd} \tag{3}$$

donde *Ts* es el periodo de conmutación y está definido como:

$$Ts = t_{on} + t_{off}\ \mathrm{t_{on}} \tag{4}$$

La ecuación 3, indica que el voltaje de salida para el modo continuo depende solamente del ciclo de trabajo [17].

El convertidor puede operar en modo continuo o discontinuo, dependiendo del comportamiento de la corriente a través del inductor [17]. Para el análisis del convertidor se asumen dos estados por periodo de conmutación *Ts*. En el estado de encendido (t_{on}), el transistor conduce la corriente de la fuente de entrada a través del inductor hasta la carga y en el estado de apagado (t_{off}), el transistor deja de conducir, por lo que la corriente del inductor fluye a través del diodo.

4. Control difuso

El Profesor Lotfi Zadeh, de la Universidad de California en Berkeley, a mediados de la década de 1960 introdujo su teoría de los conjuntos difusos. En ella, la idea de pertenencia o de no pertenencia de un elemento a un determinado conjunto fue sustituida por el grado de pertenencia de un elemento a uno o varios conjuntos. Basándose sobre esta idea, se desarrolló toda la teoría de conjuntos, redefiniendo operaciones y propiedades [18].

En el diseño de un controlador, se establecen un conjunto de reglas sobre el control del proceso. Las reglas pueden ser establecidas mediante las decisiones tomadas y acciones realizadas por el operador que controla el proceso. También, la base de reglas puede ser construida a partir de la experiencia de un ingeniero en control que ha realizado el modelo matemático, un amplio análisis y algoritmos de control del proceso a controlar. Tal conocimiento es cargado en el controlador difuso para automatizar el proceso en cuestión (Figura 4). El control difuso proporciona un método para representar e implementar las ideas sobre cómo lograr un control de alto desempeño [19].

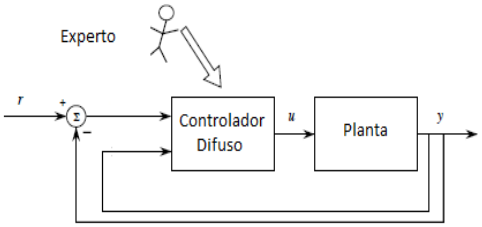


Figura 4. Principio de diseño de un sistema de control difuso.

El control difuso es una de las técnicas más utilizadas cuando no se tiene el conocimiento exacto del sistema que se desea controlar. Al igual que los “controladores clásicos”, toman los valores de las variables de entrada, realizan algún procedimiento con ellas y deciden como modificar las variables de salida, afectando estas últimas a la planta. La diferencia esencial es que no procesan ecuaciones ni tablas, procesan reglas para decidir como cambiar las salidas [15].

Un sistema de control difuso está compuesto de los siguientes elementos [15].

- Base de reglas (conjunto de reglas *If-Then*). Contiene una cuantificación lógica difusa de la descripción lingüística del operador experto sobre cómo lograr un buen control.
- Mecanismo de inferencia (también llamado máquina de inferencia o módulo de inferencia difusa). Emula la decisión del experto. Interpreta y aplica el conocimiento adquirido sobre cómo realizar el mejor control de la planta.
- Interface de difusificación. Convierte las entradas del controlador en información que el mecanismo de inferencia puede usar fácilmente para activar y aplicar reglas.
- Interface de desdifusificación. Convierte las conclusiones del mecanismo de inferencia en las correspondientes entradas del proceso.

5. Desarrollo

5.1 Sistema propuesto

El sistema de alimentación, como se puede ver en la figura 5, se basa en un convertidor buck trabajando en modo continuo. La resistencia de carga del convertidor es el alambre muscular y consta de su propio sistema de accionamiento mediante un transistor Mosfet, activado por un primer sistema de modulación de ancho de pulso, PWM. Para este sistema de activación del alambre, se debe indicar el ancho del pulso D_{sma}, acorde a las necesidades de trabajo del actuador donde está integrado el alambre. Como se puede observar en la mima figura, este ciclo de trabajo es una de las entradas evaluadas por el controlador difuso.

Existe un segundo sistema de modulación de ancho de pulso, que genera las señales de voltaje que intervienen en la conmutación del interruptor propio del convertidor buck. El ciclo de trabajo de esta segunda señal PWM, es la salida del controlador difuso *Dbuck*, de tal forma que al modificar su valor se cambia la magnitud del voltaje de salida del convertidor. En consecuencia, la magnitud de la corriente que circula en el alambre también será modificada, y gracias al algoritmo de control, se asegura que su

magnitud no sea mayor a la recomendada por el fabricante. Asimismo, el controlador difuso bebe permitir el flujo de corriente que se demande para el correcto funcionamiento del alambre, cuando no es exigido al máximo.

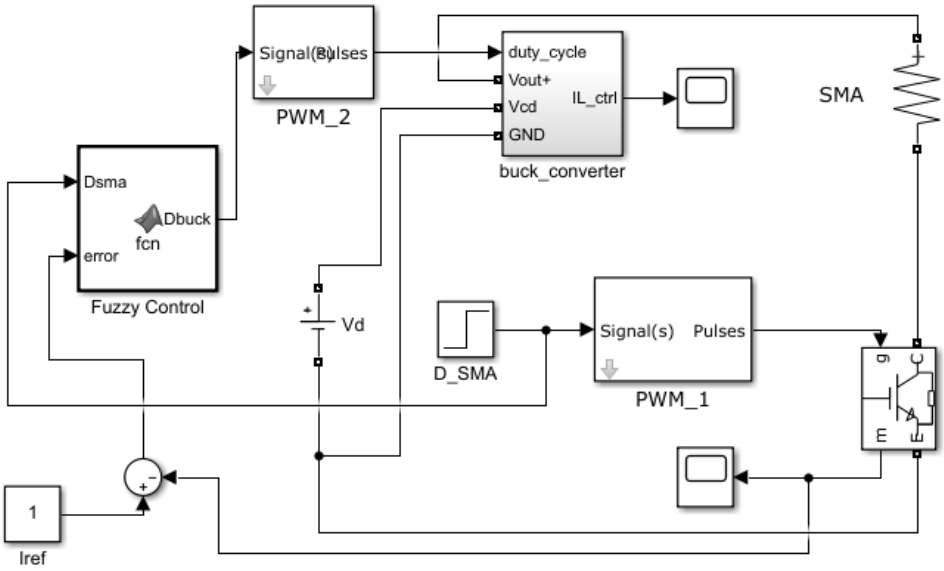
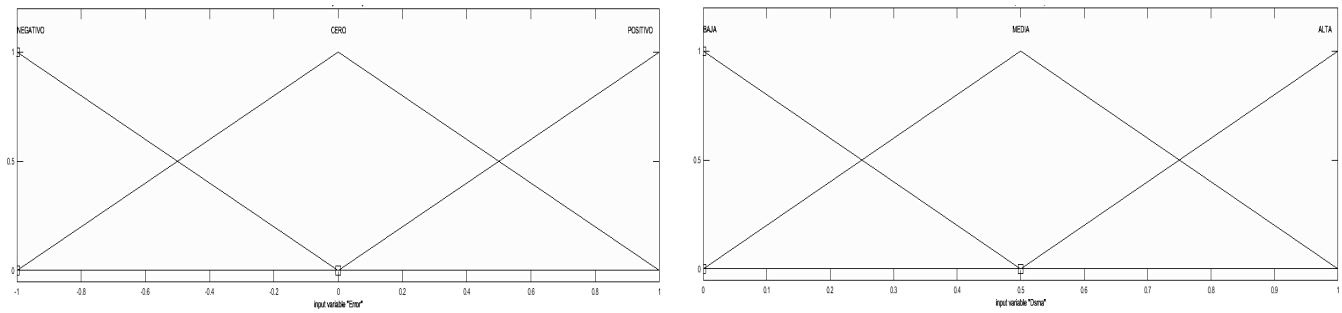


Figura 5. Sistema de alimentación para alambres musculares.

En la figura anterior también es posible observar que el controlador difuso considera el valor de una segunda entrada llamada *error*, que corresponde a la diferencia que existe entre la corriente de referencia *Iref* (valor recomendado por el fabricante del alambre) y la que circula a través del alambre.

5.2 El controlador difuso

En la figura 6 se muestran las funciones de pertenencia de las variables de entrada del control difuso, que corresponden al valor del ciclo de trabajo con el que es conmutado el alambre muscular y el valor del error en corriente.



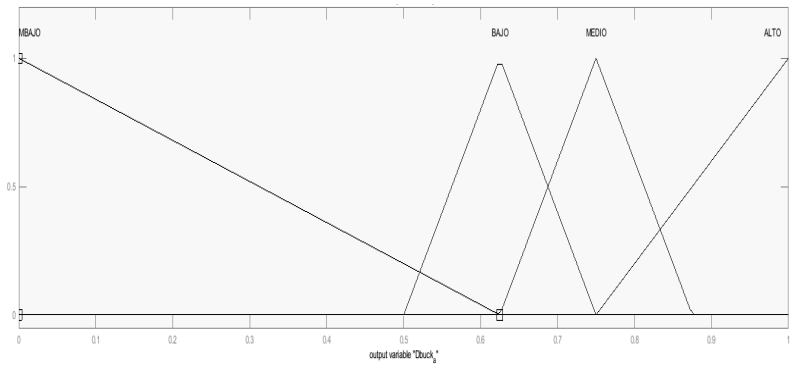
a) Error b) Dsma

Figura 6. Funciones de pertenencia de las variables de entrada.

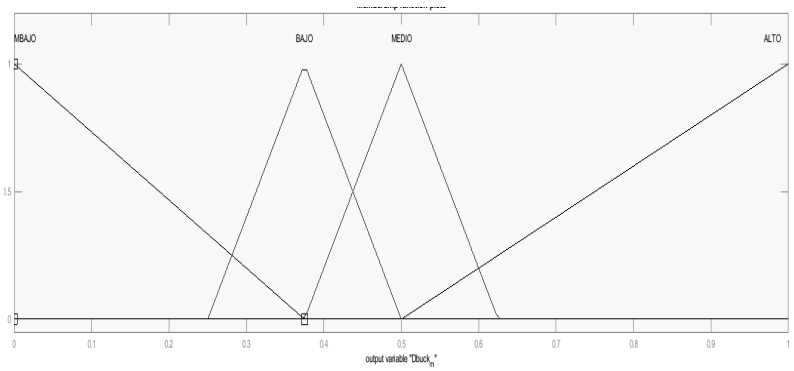
La figura 7 muestra las funciones de pertenencia utilizadas por el controlador para la operación de desdifusificación y determinar el valor de su salida, *Dbuck*. Cabe señalar que, comúnmente un controlador difuso clásico trabaja con un conjunto de funciones de pertenencia, por cada variable de

salida. Es este sistema propuesto, el controlador difuso cuenta con información de tres conjuntos para determinar la variable de salida.

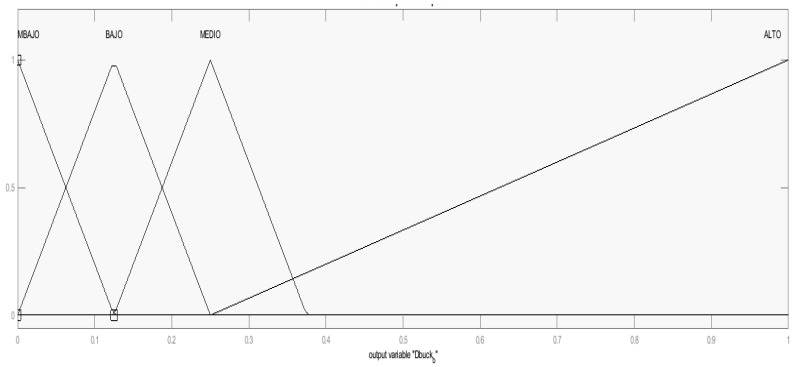
De acuerdo a la magnitud de la corriente de referencia, el controlador determina el conjunto que va considerar en los cálculos de *Dbuck*. Si *Iref* tiene un valor entre 10 mA y 1.5 A, utiliza las funciones de pertenencia de $Dbuck_{bajo}$. Cuando el valor de *Iref* se encuentre en el rango de 1.6 A a 3 A considera las funciones de $Dbuck_{medio}$. Finalmente, si la corriente de referencia se establece entre 3.1 A y 4 A, el proceso de desdifusificación será considerando la información de $Dbuck_{alto}$.



a) Función de pertenencia Dbuckalto



b) Función de pertenencia Dbuckmedio



c) Función de pertenencia Dbuckbajo

Figura 7. Conjunto de funciones de pertenencia de la variable de salida del control difuso por rangos.

En la tabla 1 se muestra la base de reglas que debe considerar el controlador difuso y que se proponen acorde a las acciones que deberá de tomar en distintos casos presentes en las variables de entrada.

Tabla 1. Base de reglas del controlador difuso.

Dbuck		***Dsma***		
		Baja	**Media**	**Alta**
Error	**Negativo**	Mbajo	Mbajo	Bajo
	Cero	Bajo	Medio	Alto
	Positivo	Bajo	Medio	Alto

6. Resultados

El sistema simulado se muestra en la figura 8. Se consideraron y analizaron distintos casos, tanto para asegurar que la corriente a través del alambre muscular no excede la recomendada por el fabricante, como aquellos en los que el sistema debe proveer la corriente requerida en donde el alambre no opera con un ciclo de trabajo del 100%.

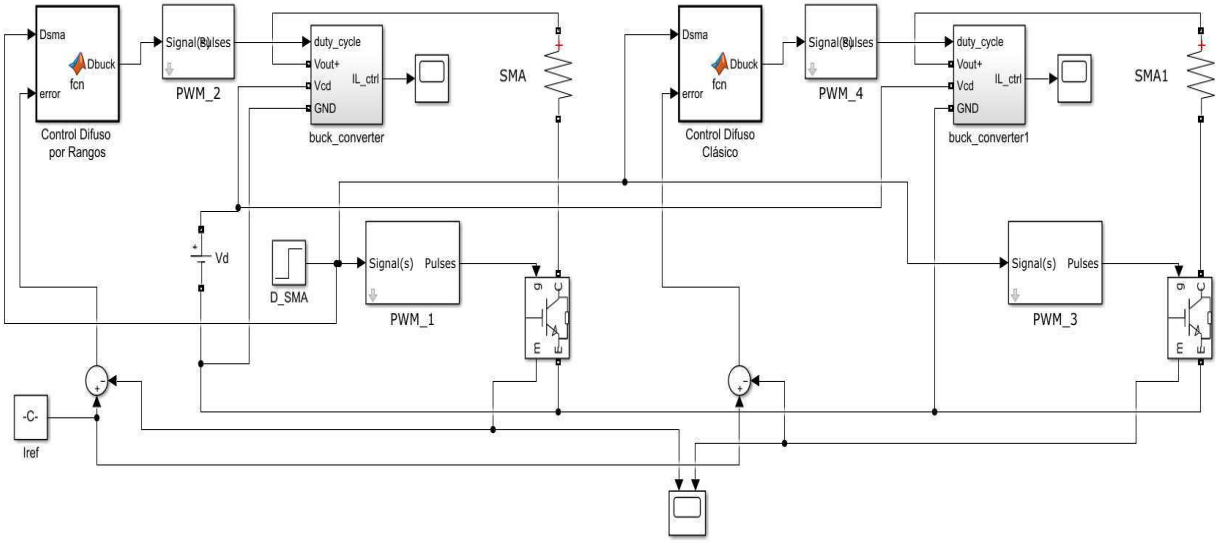


Figura 8. Sistema simulado.

Como se puede observar en la figura 8, en todos los casos de simulación y análisis, el desempeño del sistema propuesto es comparado con el desempeño de un sistema de control difuso clásico. Este último controlador sólo considera el conjunto de funciones de pertenencia de la variable de salida mostrado en la figura 9, para la etapa de desdifusificación.

También se observa en la figura 8 que el voltaje de alimentación, corriente de referencia y ciclo de trabajo del alambre muscular son iguales en ambos sistemas, para asegurar que las condiciones de operación son las mismas. Cabe señalar, que las funciones de pertenencia de las variables de entrada y las reglas de operación son iguales. La única diferencia que existente entre un sistema de control y otro, es la información que utiliza en los cálculos de la etapa de desdifusificación.

En los casos simulados se consideran los datos de alambres comerciales de la marca flexinol. Para las primeras pruebas, inicialmente el alambre se encuentra operando con un ciclo de trabajo de 0 y después, se incrementa hasta 1. Posteriormente, a los 0.9 s de simulación existe una perturbación, incrementando del 15% del voltaje de alimentación del convertidor buck.

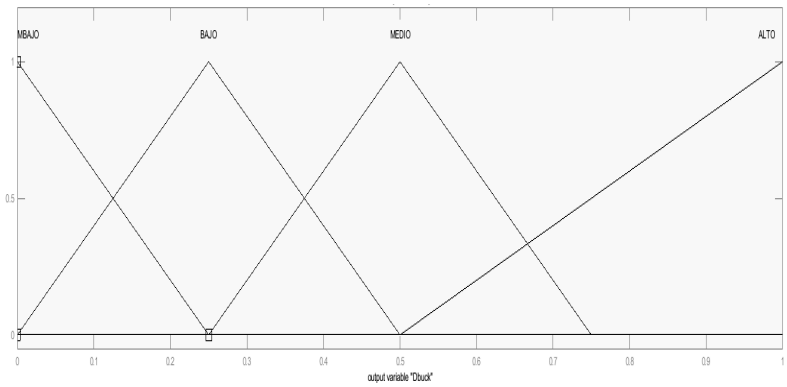


Figura 8. Funciones de pertenencia de la variable de salida del controlador difuso clásico.

En las siguientes tres figuras se muestra el comportamiento de la corriente en el alambre muscular, la señal de color negro corresponde a la del sistema de control propuesto y la señal en color rojo es la corriente obtenida con el sistema de control difuso clásico. Para la respuesta de la figura 9 se trabaja con un alambre muscular que tiene diámetro de 0.25 mm, resistencia lineal de 18.5 Ω/m y el fabricante recomienda una corriente de operación de 1.050 A, por un lapso de tiempo de 1 s, que es el tiempo requerido para su total contracción [20].

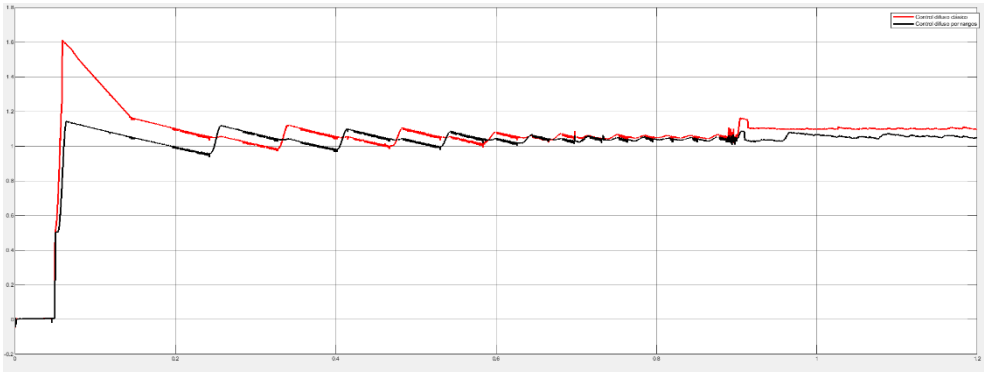


Figura 9. Respuesta en corriente de un alambre muscular de 0.25 mm de diámetro.

En la figura anterior es posible observar que con ambos sistemas de control la corriente logra una estabilización en el máximo valor recomendado por el fabricante. Sin embargo, cuando el ciclo de trabajo de operación del alambre es máximo, existe un mayor sobre impulso en la corriente del sistema con el controlador clásico. También es posible ver que en el momento que existe una perturbación en la magnitud del voltaje de alimentación, con el controlador clásico la corriente no logra regresar al valor de referencia. Por otro lado, con el sistema de control difuso por rangos, se mantiene en dicho valor.

En la figura 10 se muestra la respuesta de corriente ahora con un alambre comercial de 0.38 mm de diámetro, resistencia lineal de 8.3 Ω/m y una corriente recomendada por el fabricante de 2.250 A [20].

Se puede observar en la figura 10 que la respuesta de la corriente es similar al caso mostrado en la figura 9. Cuando el ciclo de trabajo del alambre tiene el valor de 1, existe un mayor sobreimpulso con el controlador difuso clásico y el tiempo de estabilización es mayor, que la respuesta obtenida con el controlador difuso por rangos. Cuando se presenta la perturbación, la corriente del alambre tiene mayor sobreimpulso y no regresa totalmente al valor de referencia cuando se utiliza el control difuso clásico. Con el controlador propuesto en este trabajo, es menor el sobreimpulso y se estabiliza en el valor de referencia.

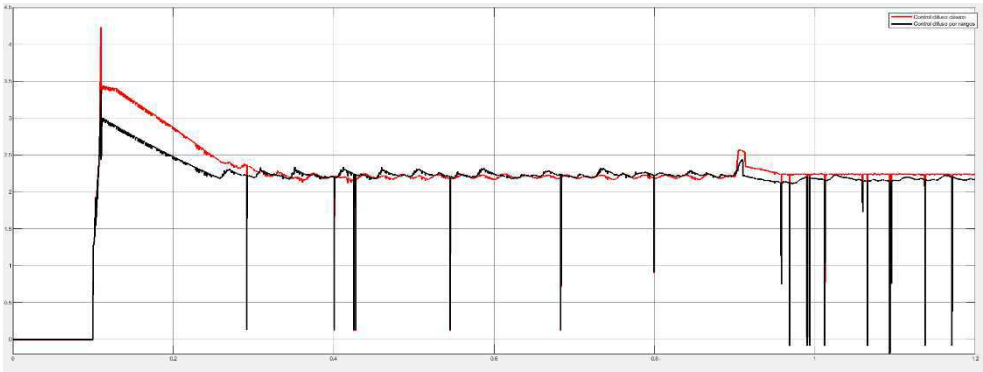


Figura 10. Respuesta en corriente de un alambre muscular de 0.38 mm de diámetro.

La figura 11 muestra la forma en que responde la corriente que circula en un alambre muscular con un diámetro de 0.51 mm, 4.3 Ω/m de resistencia lineal y una corriente de operación recomendada de 4 A [20]. En este caso se puede observar que se mantiene el comportamiento de la corriente de forma similar a los anteriores. Tanto en el momento de que el ciclo de trabajo del alambre es 1 y cuando llega la perturbación, el sobreimpulso y tiempo de estabilización es mayor con el controlador difuso clásico, comparado con el controlador difuso propuesto en este trabajo.

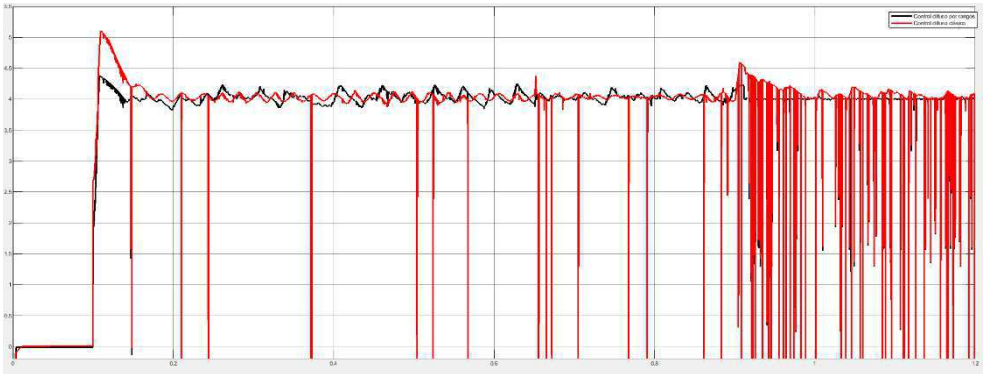


Figura 11. Respuesta en corriente de un alambre muscular de 0.51 mm de diámetro.

En la tabla 2 se muestran los valores de los parámetros de desempeño (máximo sobreimpulso, tiempo de establecimiento y error en estado estable), obtenidos en la respuesta en corriente de cada caso mostrado, tanto para el control difuso clásico, como con el controlador difuso por rangos.

Tabla 2. Valores de parámetros de desempeño

Tipo de controlador	Alambre de 0.25 mm de diámetro			Alambre de 0.38 mm de diámetro			Alambre de 0.51 mm de diámetro		
	Mp	*Ts*	*Ess*	*Mp*	*Ts*	*Ess*	*Mp*	*Ts*	*Ess*
Difuso clásico	52 %	0.7 s	0.15 %	88 %	0.3 s	0.1 %	27.5 %	0.12 s	0 %
Difuso por rangos	9 %	0.7 s	0.1 %	44 %	0.2 s	0 %	8.75 %	0.05 s	0 %

Los valores mostrados en la tabla 2 permiten asegurar que la diferencia entre ambos controladores difusos radica en que, cuando el sistema de alimentación trabaja con el de tipo clásico, permite que la corriente que circule a través del alambre muscular sea mayor.

Las siguientes pruebas permitieron conocer el comportamiento de la corriente en el alambre con el controlador difuso clásico y por rangos, en momentos en que el alambre muscular no trabaja con el máximo ciclo de trabajo. En las siguientes simulaciones, el alambre muscular comienza trabajando con un ciclo de trabajo de 0.5 y después de medio segundo es llevado al máximo. De nuevo se consideraron los datos del fabricante de los alambres de las pruebas anteriores. La figura 12 muestra la respuesta en corriente del alambre de 0.25 mm de diámetro, en la figura 13 se puede observar la respuesta para el alambre de 0.38 mm de diámetro y para el alambre de 0.51 mm de diámetro, su respuesta se poder ver en la figura 14. En todas las figuras, las señales de color rojo corresponden a la respuesta del sistema de alimentación con el controlador difuso clásico. Las señales de color negro, corresponden a la respuesta en corriente cuando el sistema trabaja con el sistema de control difuso por rangos de operación.

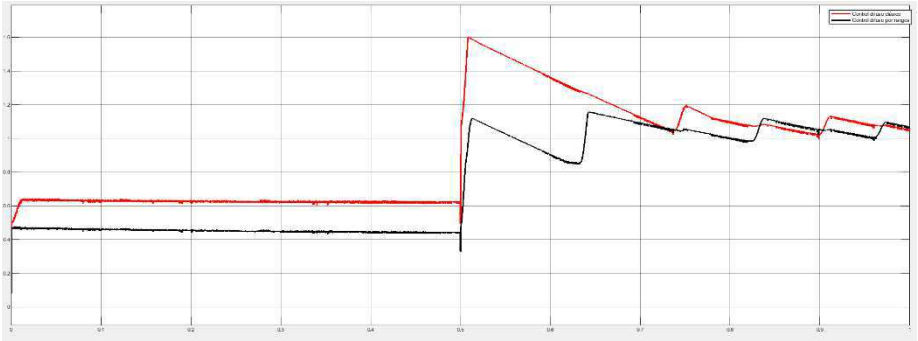


Figura 12. Respuesta en corriente de un alambre muscular de 0.25 mm de diámetro.

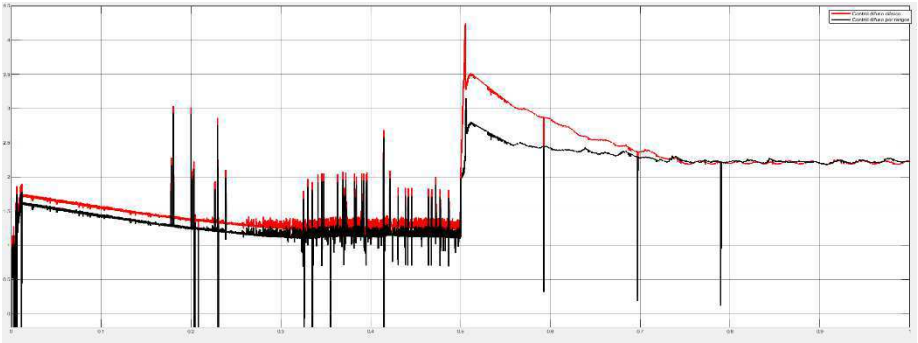


Figura 13. Respuesta en corriente de un alambre muscular de 0.38 mm de diámetro.

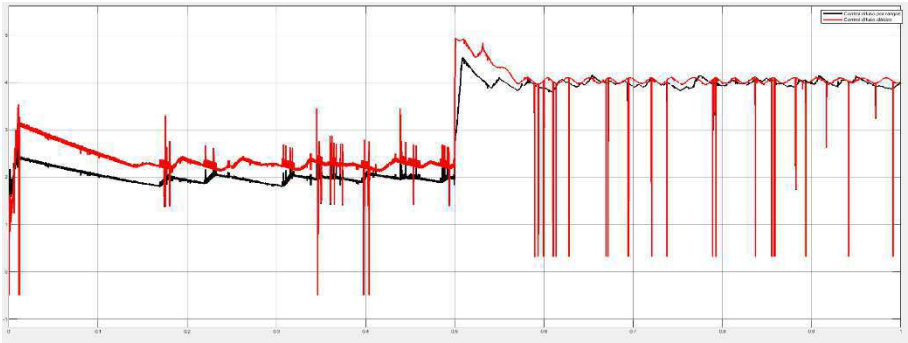


Figura 14. Respuesta en corriente de un alambre muscular de 0.51 mm de diámetro.

En las respuestas anteriores es posible observar que, en general, cuando el sistema de alimentación opera con el controlador clásico, el alambre muscular recibe más corriente de la requerida. Cuando se tiene un ciclo de trabajo de 0.5, el sistema de control difuso por rangos asegura que la corriente que circule, en el caso de que se exceda a la recomendada por el fabricante, sea por menor tiempo, comparado con el sistema con el controlador clásico. Cuando el ciclo de trabajo se lleva al valor de 1, las respuestas son similares a las obtenidas en las pruebas anteriores.

7. Conclusiones

Se presenta la propuesta de un sistema de alimentación para actuadores basados en alambres musculares. El sistema de alimentación para un actuador de este tipo debe tener la capacidad de entregar la corriente requerida en el alambre cuando no se opera al máximo, pero no debe de entregar una magnitud mayor a la requerida. De igual forma, al trabajar el alambre a su máxima capacidad (ciclo de trabajo igual a 1), es importante que la respuesta en corriente sea rápida, pero sin exceder la magnitud recomendada por el fabricante. De otra forma, si no se asegura dicho comportamiento de la corriente, el alambre se va degradando y no cumplirá los ciclos de elongación – contracción indicados por el fabricante e incluso puede llegar a su destrucción total.

El sistema de alimentación propuesto incluye un sistema de control difuso por rangos de operación para el convertidor buck, etapa en la que se basa la propuesta. Las respuestas obtenidas nos permiten concluir que, en efecto, el sistema presentado cumple los requisitos respecto al comportamiento de la corriente. Tanto en el caso de que el alambre opera con un ciclo de trabajo de 0.5, como en el que opera con uno igual a 1, la magnitud de la corriente es adecuada y menor a la que permite el sistema, al operar con un controlador difuso clásico. Lo anterior permite garantizar la correcta operación del alambre, manteniendo la vida útil especificada por su fabricante.

El diseño de un controlador difuso por rangos de operación permite un mejor desempeño del sistema controlado, comparado con el desempeño obtenido con un controlador difuso clásico. Al contar con una función de pertenencia de la variable de salida, por cada rango de operación que se determine, el controlador realiza ajustes más precisos.

Como trabajo futuro se piensa implementar el sistema de alimentación y analizar su desempeño, respecto a la corriente que circula a través del alambre muscular. De igual forma, se buscará implementar en el controlador difuso con sistema de adaptación que ajuste las funciones de pertenencia de la variable de salida, acorde al punto de operación del controlador buck. Una de las opciones que se ha visto que tiene resultados favorables es la técnica conocida como “selección booleana de un grupo de funciones de pertenencia de salida, basado en las regiones de operación de un sistema”, reportada por [21].

Referencias

[1] Mohd Jani J., Leary M., Subic A., Gibson M. A. “*A review of shape memory alloy research, applications and opportunities*”, Materials and Design. Elsevier, vol. 56, pp. 1980 – 2015, 2014.
[2] Naresh C., Bose P. S. C., Rao C. S. P. “*Shape memory alloys: a state of art review*”, Materials Science and Engineering. IOP Publishing, vol. 149, 2016.
[3] Clithy E. “*Application of Shape Memory Alloy*”, Sci Insigt. SSN, vol. 33, nó, 3, pp. 167 – 174, 2020.
[4] Jani J. M., Leary M., Subic A. *“Designing shape memory alloy linear actuators: A review”*. Journal of Intelligent Material Systems and Structures. SAGE, vol. 28, nó. 13, pp. 1699 – 1718, 2016.
[5] Patil D., Song G. “*A Review of Shape Memory Material’s Applications in the Offshore Oil and Gas Industry”,* Smart Materials and Structures. IOP Publishing, vol. 26, 2017.
[6] Tanaka D., Uchiumi Y., Kawamura S., Takato M. “*Four-leg independent mechanism for MEMS microrobot”,* Artif Life Robotics. Springer, vol. 22, pp. 380 – 384, 2017.
[7] Modabberifar M., Spenko M. “*A Shape Memory Alloy-Actuated Gecko-Inspired Robotic Gripper*”, Sensors and Actuators A: Physical. Elsevier, vol. 276, pp. 76 – 82, 2018.

[8] Roshan U., Amarasinghe R., Dayananda, N.W.N. *"Design and Development of a Shape Memory Alloy Spring Actuated Gripper for Minimally Invasive Surgeries",* Proceedings of International Conference on Artificial Life, vol. 23, pp. 566 – 569, Japón, 2018.
[9] Motzki P., Kunze J., York A., Seelecke S. "*Energy-efficient SMA Vacuum Gripper System*", 15th International Conference on New Actuators, Alemania, 2016.
[10] Sohn J. W., Kim G., Choi S. *"A State-of-the-Art Review on Robots and Medical Devices Using Smart Fluids and Shape Memory Alloys*", Applied Sciences. MDPI, vol. 8, nó. 10, 2018.
[11] Gilberston R. G. *"Muscle Wires Project Book"*. Mondo Tronics Inc. 3rd Edition. San Rafael CA, EUA. 2005. ISBN:1-879896-13-3.
[12] Kim H., Han Y., Lee D. Y., Ha J. I., Cho K. J. *"Sensorless displacement estimation of a shape memory alloy coil spring actuator using inductance"*. Smart Materials and Structures. vol. 22, nó. 2. 2013.
[13] Yuan H., Fauroux J., Chapelle F., Balandraud X. "*A review of rotary actuators based on shape memory alloys*", Journal of Intelligent Materials Systems and Structures. SAGE, vol. 28, nó. 14, pp. 1863 – 1885, 2017.
[14] Shintake J., Cacucciolo V., Floreano D., Shea H. "*Soft Robotic Grippers*", Advanced Materials. WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, vol. 30, 2018.
[15[Passino K. M., Yurkovich S. "*Fuzzy control"*. Addison-Wesley. Menlo Park, CA, EUA. 1998. ISBN: 020118074X, 9780201180749.
[16] Chaitanya S. K., Dhanalakshmi K. *"Control of shape memory alloy actuated gripper using Sliding Mode Control"*, International Conference on Control Applications. IEEE. pp. 1247-1252. 2013.
[17] Mohan H., Undeland T. M., Robbins W. P. *"Power electronics: converters, applications, and design"*. John Wiley & Sons Inc. 3rd Edition. EUA. 2002. ISBN: 978-0471226932.
[18] Ross T. J. *"Fuzzy logic with engineering applications"*. John Wiley & Sons Inc. 3rd Edition. EUA. 2010. ISBN: 978-0-470-74376-8.
[19] Jantzen J. *"Foundations of Fuzzy Control"*. John Wiley & Sons Inc. EUA. 2007. ISBN 978-0-470-02963-3.
[20] Dinalloy. "*Technical Characteristics of Flexinol*", Donalloy Inc. 2018.
[21] Vélez Díaz D., Benítez Read J. S. "*Fuzzy System to Control the Neutron Power with Different Sets of Output Membership Functions*", Proc. 7th IFSA World Congress. vol. 4, pp. 132-136, 1997.

Diseño Colaborativo en Mecatrónica, Capítulo 3, pp. 28 - 35
ISBN: 978-607-9394-23-3, 2021

Control de seguimiento articular en un exoesqueleto de 4 grados de libertad para miembros inferiores

M. Zamora, Angel Rodriguez-Liñan,
Griselda Quiroz-Compean, Luis M. Torres-Treviño

Universidad Autónoma de Nuevo León, Facultad de Ingeniería Mecánica y Eléctrica,
Ave. Universidad S/N, Cd. Universitaria, San Nicolás de los Garza, N.L., CP 66455 México.
e-mail: angel.rodriguezln@uanl.edu.mx

Resumen

En este trabajo se presenta un modelo de un exoesqueleto de cuatro grados de libertad, el cual incluye las dinámicas del mecanismo y del actuador. El objetivo de control es el seguimiento de las trayectorias articulares en el plano sagital cuando se emula el ciclo de la marcha humana. Para ilustrar el desempeño del controlador par calculado, la implementación del esquema de control en lazo cerrado con los modelos del exoesqueleto y sus actuadores fue realizada en simulaciones computacionales. Los resultados indican un buen desempeño en el seguimiento de las trayectorias articulares.

Palabras clave: Exoesqueleto de miembro inferior, dinámica con actuadores, control par calculado, filtro extendido de Kalman.

1. Introducción

Se estima que más de mil millones de personas en el mundo viven con alguna discapacidad, donde la más común está asociada a los miembros inferiores [1]. Dado lo anterior, surge la necesidad de desarrollar nuevas tecnologías para asistir la realización de las tareas motrices básicas, así como una mejora en la autonomía del usuario. El uso de dispositivos de asistencia es una alternativa real para la rehabilitación de pacientes con problemas de neuromotricidad. Una de estas alternativas es el uso de exoesqueletos actuados y de funcionamiento automático.

Los exoesqueletos mecánicos [2] son dispositivos que se acoplan al cuerpo del usuario y trabajan en conjunto con sus movimientos. A lo largo de los años, estos mecanismos han sido desarrollados para asistir a las actividades que involucran la locomoción humana, así fue como comenzó la investigación y el desarrollo de órtesis activas y de exoesqueletos a finales de 1960 [3]. La implementación de los exoesqueletos puede ser clasificada de acuerdo con sus habilidades o capacidades en 3 grandes grupos; amplificadores de fuerza, con interacción háptica y rehabilitación [4]. El presente trabajo se centra en el tercer grupo, algunos ejemplos de este grupo son el exoesqueleto HAL (mostrado en la figura 1), Indego y ReWalk, además de algunos otros prototipos recientes descritos en [2] y [5]. En este trabajo un controlador en lazo cerrado es implementado en un modelo de exoesqueleto de cuatro grados de libertad por pierna.

De trabajos como [2] y [5], sabemos que algunos de los exoesqueletos utilizan servomotores como actuadores, por lo que es de interés modelar la dinámica del actuador. En [6] se presenta una metodología para obtener el modelo de un servomotor, así como la identificación paramétrica del mismo. Puesto que es posible obtener el modelo tanto del mecanismo como el de sus actuadores, en este trabajo se plantea la adición de la dinámica correspondiente a los servomotores con la dinámica del mecanismo en un solo modelo, con la finalidad de aplicar un controlador basado en las dinámicas modeladas. El buen desempeño del control en lazo cerrado depende en gran medida de la calidad y precisión de las mediciones disponibles. Cuando no es posible contar con la medición de algunas de

las señales de interés, es necesario realizar la estimación de dichas señales empleando técnicas como el filtro extendido de Kalman (EKF por sus siglas en inglés), el cual es una de las adaptaciones más sencillas del filtro de Kalman para sistemas no lineales, permitiendo la estimación de las variables no medibles [7].

Figura 1: Exoesqueleto HAL.

El objetivo de control es lograr el seguimiento de las posiciones articulares al reproducir el ciclo de la caminata humana en fase de balanceo. Por lo que, para resolverlo, se propone utilizar una ley de control basada en linealización por retroalimentación de estados, conocida como par calculado [8] [9], el cual es un tipo de controlador no lineal utilizado en robótica, basado en el modelo dinámico del mecanismo [10].

El resto del documento se organiza como sigue: En la sección 2 se describe la obtención del modelo del mecanismo. En la sección 3 se incluye el modelo de los actuadores a la dinámica del mecanismo. En la sección 4, se describe el método conocido como filtro extendido de Kalman. La sección 4 corresponde a la ley de control seleccionada, la sección 5 abarca los resultados de las simulaciones. En la sección 6 se dan las conclusiones y discusiones.

2. Modelo del mecanismo

Considérese un mecanismo multiarticualdo completamente actuado de cuatro grados de libertad correspondiente a un exoesqueleto para extremidades inferiores, considerado para rehabilitación en lo que se conoce como fase de balanceo, en la cual se realiza movimiento mientras está suspendido en el aire. Los grados de libertad del mecanismo corresponden con los movimientos en el plano sagital de la cadera (q_1), la rodilla (q_2), el tobillo (q_3) y el pie (q_4), cada eslabón posee un actuador que le permite realizar movimientos de manera independiente al resto de los eslabones. El análisis cinemático se llevó a cabo utilizando la metodología de Denavit-Hartenberg (D-H), donde se asocia un sistema coordinado a cada eslabón y usando transformaciones homogéneas es posible representar la posición y orientación de todos los eslabones del mecanismo. Los sistemas coordenados colocados se pueden ver en la figura 2 y los parámetros D-H en la tabla 1.

Es conveniente conocer la dinámica del mecanismo para utilizar un esquema de control que permita que el mecanismo logre el objetivo de control a pesar de fenómenos mecánicos como la inercia, la gravedad, fuerzas centrífugas, etcétera. La formulación de Lagrange permite obtener las ecuaciones dinámicas a partir de una función llamada Lagrangiano, definida como la diferencia entre la energía cinética $K(q,\dot{q})$ y la energía potencial $U(q)$ de un sistema mecánico, es decir $L(q,\dot{q}) = K(q,\dot{q}) - U(q)$, siendo la ecuación de Euler-Lagrange

$$\tau = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \tag{1}$$

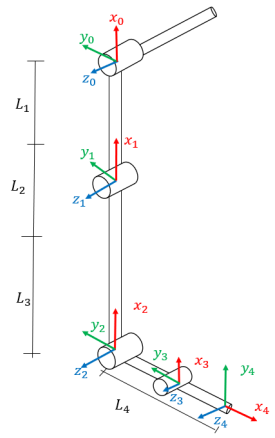


Figura 2: Mecanismo de cuatro grados de libertad.

Tabla 1: Parámetros D-H del mecanismo de cuatro grados de libertad.

i	θ_i	d_i	a_i	α_i
1	$q_1 + 90^\circ$	0	$-L_1$	-90°
2	$q_2 - 90^\circ$	0	$-L_2$	0
3	$q_3 + 90^\circ$	0	$-L_3$	0
4	q_4	0	L_4	$+90^\circ$

donde $\tau \in \mathbb{R}^n$ es el vector de fuerzas o pares de torsión en las articulaciones, $q \in \mathbb{R}^n$ es el vector de posiciones articulares y $\dot{q} \in \mathbb{R}^n$ es el vector de velocidades articulares, n es el número de grados de libertad. El modelo dinámico (1) se puede reescribir la forma compacta

$$M(q)\ddot{q} + H(q,\dot{q}) + G(q) = \tau \tag{2}$$

donde $M(q) \in \mathbb{R}^{n\times n}$ es una matriz definida positiva llamada matriz de inercia, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ es una matriz de fuerzas centrífugas y de Coriolis, $G(q) \in \mathbb{R}^n$ es el vector de efectos gravitacionales, el vector $\ddot{q} \in \mathbb{R}^n$ es el vector de aceleraciones articulares.

3. Modelo del actuador

Para poder aplicar una estrategia de control por linealización del modelo, puesto que en este esquema el control es la señal eléctrica que entra en los actuadores, se propone la inclusión del modelo del actuador al modelo dinámico del mecanismo de la siguiente forma. En este caso particular, se supone que los actuadores con los que cuenta el mecanismo son servomotores RC, de acuerdo con [6] cada uno puede representarse como

$$c_{1i}\ddot{q}_i + c_{2i}\dot{q}_i + q_i = c_{3i}w_i - c_{4i}\tau_i - c_{5i} \tag{3}$$

donde w_i es la señal aplicada por modulación de ancho de pulso (PWM por sus siglas en inglés), q_i es la posición angular, $\dot{q}_i$ es la velocidad angular, $\ddot{q}_i$ es la acelaración angular, τ_i es el par de torsión asociado a la carga; c_{1i}, c_{2i}, c_{3i}, c_{4i} y c_{5i} son coeficientes constantes que dependen de los parámetros eléctricos y mecánicos de cada servomotor RC.
Dado que este problema corresponde a un mecanismo de $n = 4$, los coeficientes de (3) se pueden agrupar tal que $C_1 = diag\{c_{11}, c_{12}, \ldots, c_{14}\}$, $C_2 = diag\{c_{21}, c_{22}, \ldots, c_{24}\}$, $C_3 = diag\{c_{31}, c_{32}, \ldots, c_{34}\}$, $C_4 =$

$diag\{c_{41}, c_{42}, \ldots, c_{44}\}$ y $C_5 = diag\{c_{51}, c_{52}, \ldots, c_{54}\}$ reescribiendo simultáneamente la dinámica de los cuatro servomotores RC (descritos cada uno por (3)) en la forma compacta

$$C_1\ddot{q} + C_2\dot{q} + q = C_3 w - C_4\tau - C_5 \tag{4}$$

Sustituyendo (2) en (4) el modelo de toda la planta incluyendo, los actuadores, los sensores y el mecanismo viene dado por

$$M_c(q)\ddot{q} + H_c(q,\dot{q}) + G_c(q) = C_3 w \tag{5}$$

donde $M_c = (C_1 + C_4 M(q))\ddot{q}$, $H_c = C_2\dot{q} + C_4 H(q,\dot{q})$ y $G_c = q + C_4 G(q) + C_5$. Si se definen las variables $X_1 = q$ y $X_2 = \dot{q}$, se puede reescribir (5) como el modelo en espacio de estados dado por

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= M_c^{-1}(X_1)[C_3 w - H_c(X_1, X_2) - G_c(X_1)] \end{aligned} \tag{6}$$

En la figura 3 se presenta el diagrama en lazo cerrado del mecanismo, suponiendo conocidos los modelos tanto del actuador como del mecanismo. Para implementar el controlador (7) se requiere el valor de las variables q y $\dot{q}$. Sin embargo, en este trabajo se supone que la medición de las velocidades articulares no está disponible, además la información disponible del desplazamiento q puede tener ruido de medición. Por lo que se propone utilizar un filtro de Kalman extendido con la finalidad de estimar la velocidad y eliminar el ruido que el sensor pueda añadir.

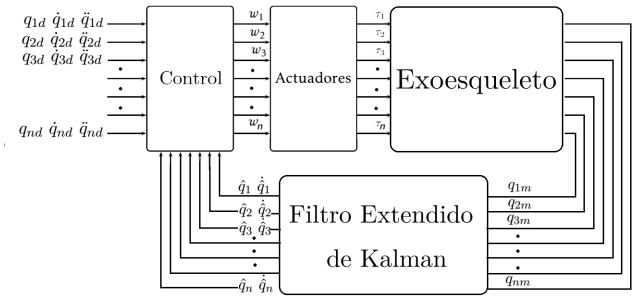


Figura 3: Representación esquemática del sistema en lazo cerrado.

4. Control de seguimiento articular

La dinámica del exoesqueleto que se describe en (2) es no lineal en términos de las variables de estado (posiciones y velocidades articulares). En [10] se describe un controlador, el cual es también no lineal en las variables de estado, con lo cual se consigue describir el sistema de control en lazo cerrado mediante una ecuación diferencial lineal.

Un diagrama a bloques de la implementación de este controlador puede verse en la figura 4, el cual es capaz de satisfacer el objetivo de control de movimiento en forma global mediante una correcta selección de sus ganancias. Dicho controlador recibe el nombre de par-calculado y está descrito por

$$\tau = \mathrm{M(q)}\left[\ \ddot{q}_d + K_v\dot{\tilde{q}} + K_p\tilde{q}\right] + C(q,\dot{q})\dot{q} + g(q) \tag{7}$$

donde $K_v\ y\ K_p$ son matrices definidas positivas, $e_1 = q_d - q$ denota el error de posición y $e_2 = \dot{q}_d - \dot{q}$ es el error de la velocidad. Debido a la similitud en la forma de las ecuaciones (2) y (5), puede proponerse un controlador similar a (7) para el sistema actuado (5) que resulta en

$$w = {C_3}^{-1}(M_c(q)[\ddot{q}_d + K_1 e_1 + K_2 e_2] + H_c(q,\dot{q}) + G_c(q)) \tag{8}$$

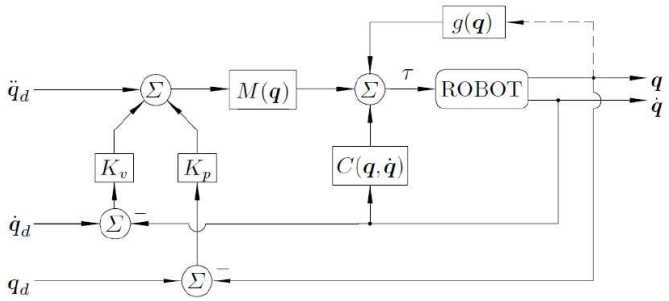


Figura 4: Diagrama de control por par-calculado (imagen obtenida de [10]**).**

donde K_1 y K_2 son matrices definidas positivas denominadas ganancias del control, tal que sustituyendo (8) en (5), se obtiene la ecuación de lazo cerrado

$$\dot{e}_2 + K_1 e_1 + K_2 e_2 = 0 \tag{9}$$

donde $e_1 = q_d - q$, $e_2 = \dot{q}_d - \dot{q}$. La dinámica del error (9) puede reescribirse como el modelo en espacio de estados

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_1 & -K_2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{10}$$

donde I es la matriz identidad. Para cumplir el objetivo de control de seguimiento (con soluciones de (10) que converjen a cero), se pueden elegir las ganancias K_1 y K_2 como matrices diagonales tales que los n polinomios característicos desacoplados

$$s^2 I + K_{2,i} s + K_{1,i} = 0 \tag{11}$$

sean polinomios Hurwitz estables, donde $K_{1,i}$ y $K_{2,i}$ son las *i*-ésimas componentes de K_1 y K_2.

5. Filtro extendido de Kalman

El filtro de Kalman es un método altamente conocido y utilizado para la estimación, éste se basa en la estimación cuadrática lineal (LQE por sus siglas en inglés) el cual tiene su mejor funcionamiento cuando el sistema y las incertidumbres son lineales. Cuando el sistema es no lineal, se recurre al filtro extendido de Kalman (EKF) el cual utiliza una aproximación lineal a la dinámica no lineal del sistema y a las mediciones aprovechando así la teoría del filtro de Kalman. La idea principal detrás del EKF es que el estado predicho se calcula a partir de la estimación anterior utilizando directamente el modelo no lineal del sistema y, de manera similar, la predicción de la medición se calcula a partir del estado predicho utilizando la función de salida no lineal. Considerando que la matriz de covarianza se calcula utilizando la matriz Jacobiana de derivadas parciales del sistema no lineal y funciones de salida evaluadas en cada paso de tiempo con los estados actuales predichos. Considere el sistema no lineal (6) reescrito en la forma

$$\dot{x} = f(x(t), r(t)) + g(w(t))y = h(x(t), v(t)) \tag{12}$$

donde t es el tiempo, $x = [X_1{}^T X_2{}^T]^T \in \mathbb{R}^{2n}$ representa el vector de estados, $w(t) \in \mathbb{R}^n$ es el vector de entrada, $y \in \mathbb{R}^n$ son las mediciones, $r(t)$ y $v(t)$ representan el ruido en el proceso y en la medición cuyas matrices de covarianza $Q \in \mathbb{R}^n$ y $R \in \mathbb{R}^n$, respectivamente. En la práctica, estas matrices podrían variar con el tiempo, sin embargo se supondrá que se mantienen constantes.

Considérese que en el sistema no lineal (12) las matrices A_c, G_c y H_c son las matrices Jacobianas evaluadas en un punto x_0

$$F = \frac{\partial f(\cdot)}{\partial x}|_{x_0}, \quad G = \frac{\partial g(\cdot)}{\partial w}|_{u_0}, \quad H = \frac{\partial h(\cdot)}{\partial x}|_{x_0} \tag{8}$$

donde $f((\cdot), g((\cdot)$ y $h((\cdot)$ representan los campos vectoriales no lineales de (6). Las variables correspondientes a los ruidos de entrada y de la medición se suponen como independientes, además de tener una distribución normal. El algoritmo de predicción del EKF se define como

$$\dot{\hat{x}} = f(\hat{x}_{k|_{k-1}}, u_k)\dot{P} = F_k|_{k-1}P_k|_{k-1}F_k|_{k-1}{}^T + Q_k|_{k-1} \tag{13}$$

Iniciando con $\hat{t}_{k-1} = \hat{x}_{k-1|_{k-1}}$ y $P(t_{k-1}) = P_{k-1|_{k-1}}$ siendo el factor de corrección, donde k es el instante k-ésimo, $\hat{x}_{k|_{k-1}}$ es la estimación apriori, $P_k|_{k-1}$ es la covarianza del error asociada a la estimación apriori , $F_k|_{k-1}$ es la matriz de transición de estados, Q es la matriz de ruido asociado proceso.

$$K_k = P_k|_{k-1}H_k{}^T(H_kP_kH_k{}^T + R^{-1})\hat{x}_k|_k = \hat{x}_k|_{k-1} + K_k(z_k - h(\hat{x}_k|_{k-1}) \tag{14}$$

iniciando $\hat{x}_k|_{k-1} = \hat{x}(t_k)$ y $P_k|_k = P(t_k)$, durante el proceso, las matrices Jacobianas F, G y H son evaluadas en el valor actual de la variable estimada $\hat{x}_k|_{k-1}$, z_k, donde K_k es la gancia de Kalman en el instante k, $\hat{x}_k|_k$ es la estimación a posteriori, $P_k|_k$ es la covarianza del error asociada a la estimación a posteriori, H_k es la matriz que indica la relación entre las mediciones y el vector de estado al momento k, R es la matriz de ruido asociada a la medición, z_k son las mediciones del sistema.

Las variables de desplazamiento filtradas y las de velocidad estimadas por el EKF se utilizan para la implementación del controlador (7), bajo la suposición que converge más rápido la estimación de variables que la dinámica del control. El análisis de estabilidad en lazo cerrado de la planta con el controlador (7) y EKF se deja para un trabajo futuro.

6. Resultados

Para ilustrar la validez y desempeño de la metodología de este trabajo, se muestran resultados de experimentos en simulaciones numéricas, donde se supone que sólo se conoce la medición de la posición articular, la cual contiene ruido.

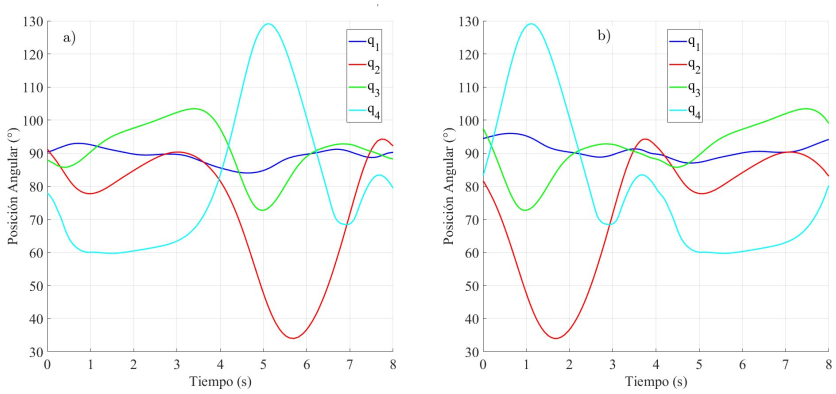


Figura 5: Trayectorias articulares de la cadera, rodilla, tobillo y pie en el plano sagital mientras se realiza la marcha humana. a) corresponde a la pierna derecha en balanceo y b) corresponde a la pierna izquierda en fase de balanceo [11].

En la figura 6 se presenta la medición de la posición angular de cada articulación (línea verde) y la posición articular controlada (línea roja) y la referencia deseada (línea azul). Se puede apreciar que, a pesar dela existencia de ruido en la medición, la estimación que realiza el filtro es lo suficientemente buena para emplearse en el lazo cerrado de control. De la misma manera, en la figura 7 se observa la velocidadangular de cada articulación (línea azul) y la velocidad angular estimada (línea roja), se puede observarque el uso de un filtro extendido de Kalman permite la estimación de velocidades no medibles con un margen de error pequeño.

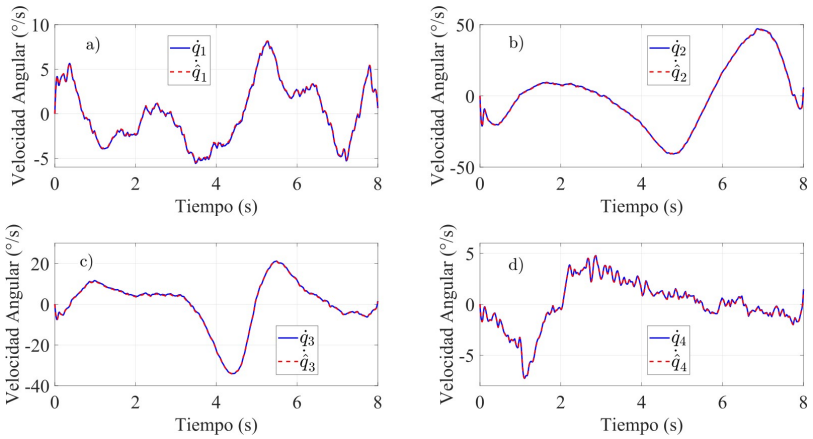


Figura 6: Velocidades articulares obtenidas en simulación (línea azul) y estimadas mediante el filtro de Kalman (línea roja), la estimación de a) corresponde con la articulación de cadera, b) con la rodilla, c) con el tobillo y d) con el pie.

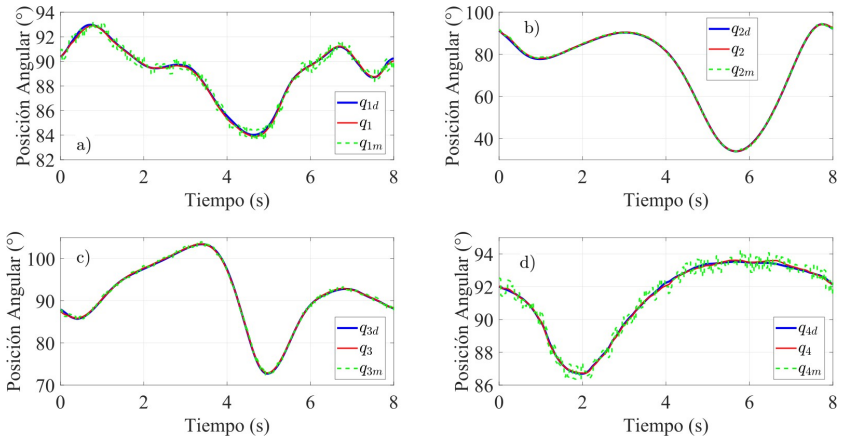


Figura 7: Salida del mecanismo en lazo cerrado retroalimentando los estados estimados (línea roja), la posición articular deseada (línea azul) y la señal leída por el sensor de posición (línea verde), a) corresponde con la articulación de cadera, b) con la rodilla, c) con el tobillo y d) con el pie.

La posición filtrada y la velocidad estimada por el EKF es posible utilizarlas para la implementacióndel controlador. En la figura 6 se observa que, al retroalimentar los estados obtenidos mediante el filtro extendido de Kalman, el esquema de control en lazo cerrado es capaz de lograr el objetivo de seguimiento a lo largo del tiempo de la simulación con buen desempeño, puesto que el error cuadrático medio (ECM) es q_1 = 1.0359°, q_2 = 0.4234°, q_3 = 0.4° y q_4 = 1°. Esto se debe gracias a que el controlador conoce el modelo del mecanismo y de los actuadores, además que las estimaciones de los estados son cercanos a los valores reales, todo esto en conjunto permiten que el esquema de control propuesto tenga un desempeño adecuado.

7. Conclusiones

A lo largo del presente trabajo se discutió sobre el estado actual de los exoesqueletos, los cuales son controlados mediante técnicas convencionales, además de que no se incluye la dinámica de los actuadores,lo cual puede conducir a errores al resolver el problema de seguimiento, por lo que se planteó la adición dela dinámica del actuador a la dinámica del mecanismo, esto con la finalidad de lograr un comportamiento de simulación mucho más real y para ser utilizado en el controlador propuesto. A su vez, se plantea el uso de una herramienta tal como el filtro extendido de Kalman para la estimación de las variables no medibles, aun bajo la existencia de ruido en las mediciones disponibles. El controlador propuesto para resolver el seguimiento de posiciones articulares, es capaz de dar una solución que satisface los criteriosde desempeño para exoesqueletos de rehabilitación que no cuenten con sistemas de visión, puesto que los errores que se obtienen en cada articulación son menores a 5°.

Como parte del trabajo a futuro queda mejorar la metodología propuesta para escenarios que se acerquen más a mecanismos reales, es decir, contemplar la existencia de dinámicas no modeladas, incertidumbre paramétrica, tanto en el modelo del mecanismo como en el modelo del actuador, perturbaciones externas, grados de libertad, etc. Luego probar dichas metodologías más robustas en plataformas experimentales físicas para validar su eficacia. Así como considerar otras etapas de rehabilitación, como fuerza aplicada del paciente, caminata en fase de apoyo, entre otras.

Referencias

[1] Krahn G. Who world report on disability: A review. *Disability and Health Journal*, 4(3):141–142, 2011.

[2] Sierra H. Control de un exoesqueleto para asistir en la bipedestación y la marcha deuna persona. *Tesis para obtener el grado de Doctor en Ciencias. Departamento de Control Automáticode la Universidad Zacatenco*, 2016.

[3] Mizen N. Preliminary design of a full-scale, wearable, exoskeletal structure. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1963.

[4] Anam K., Al-Jumaily A. Active exoskeleton control systems: State of the art. *ProcediaEngineering*, 41:988–994, 2012.

[5] Pachicano G. *Instrumentación de una plataforma experimental para el control de exoesqueletos de miembro inferior*. MSc thesis, 2017.

[6] Pérez-Morales M., Zamora M., Rodriguez-Liñán A. Caracterización paramétrica para un modelo de segundo orden del servomotor RC. *Ingenierías*, 22(82):7, 2019.

[7] Anderle M., Čelikovskỳ S. Sensor fusion for simple walking robot using low-level imple- mentation of extended kalman filter. *IFAC-PapersOnLine*, 51(13):43–48, 2018.

[8] Slotine J., Li W., et al. *Applied nonlinear control*, volume 199. Prentice hallEnglewood Cliffs, NJ, 1991.

[9] Siciliano B., Khatib O. *Springer handbook of robotics*. Springer, 2016.

[10] Kelly R., Santibáñez V., and Loría Perez Julio Antonio. *Control of robot manipulatorsin joint space*. Springer Science & Business Media, 2006.

[11] Tovar-Estrada M., Rodriguez-Liñan A., Quiroz G. Implementation of a scale-lab lower-limb exoskeleton with motion in three anatomical planes. *Cybernetics and Systems*, pages 1–23, 2019.

Diseño Colaborativo en Mecatrónica, Capítulo 4, pp. 36 - 51
ISBN: 978-607-9394-23-3, 2021

Diseño y simulación de un exoesqueleto para la articulación del codo

Centeno Barreda Daniel[1], Palomino Merino Amparo Dora[1], López Gutiérrez Jesús Ricardo[2].

Facultad de Ciencias de la Electrónica. Benemérita Universidad Autónoma de Puebla[1].
Laboratorio Franco Mexicano de Informática y Automática. Centro de investigación y de Estudios Avanzados, IPN[2].
Centeno.201009201@gmail.com

Resumen

Los exoesqueletos son dispositivos que pueden colocarse sobre alguna o varias extremidades del cuerpo humano, estos están diseñados para complementar, sustituir o incrementar las capacidades físicas del usuario. En el presente documento se diseña un exoesqueleto capaz de asistir a un paciente con alguna lesión que afecte la flexión-extensión de la articulación del codo. Este prototipo integra sensores de electromiografía para detectar la intención de movimiento de la articulación del codo, así como una unidad de medición inercial para detectar el movimiento en la articulación del hombro; también se realiza el análisis cinemático y dinámico del prototipo. Por último, se presenta la simulación del control de movimiento aplicando un control PD con compensación adaptable.

Palabras clave: Exoesqueleto, rehabilitación, articulación codo, control adaptable.

1. Introducción

Se estima que el 10% de la población mundial tiene algún tipo de discapacidad. En México, según datos del INEGI entre los años 2014 y 2018, el porcentaje de la población con algún tipo de discapacidad aumentó de 6.0% a 6.3%. Los mismos datos estadísticos indican que el porcentaje de la población cuya discapacidad está relacionada con dificultades para mover su brazo o mano también incrementó de 17.0% a 17.8%.

El uso de exoesqueletos ya sea en terapias de rehabilitación o durante las actividades diarias permite mejorar la calidad de vida de aquellas personas que sufren alguna condición médica que afecta sus habilidades motrices.

Los exoesqueletos son dispositivos diseñados para colocarse sobre las extremidades humanas y cuyo movimiento se encuentra estrechamente relacionado al movimiento anatómico de las articulaciones del cuerpo humano [1]. Algunas de sus aplicaciones son: teleoperación, aumento de fuerza y rehabilitación o asistencia del movimiento en extremidades lesionadas.

Dependiendo de la aplicación y el miembro que cubren, los exoesqueletos ofrecen diversos desafíos en áreas como diseño mecánico, electrónica y control [2],[3],[4]. La característica más importante de los exoesqueletos es la interacción entre el dispositivo y el usuario pues implica un acoplamiento físico entre ambos, por lo que las acciones de los dos elementos deben coordinarse y adaptarse mutuamente, ya que el comportamiento inesperado de uno de ellos durante la interacción puede provocar lesiones graves.

El brazo juega un papel importante en las actividades diarias pues es la extremidad encargada de la manipulación de objetos. Entre los movimientos más importantes que se llevan a cabo en esta extremidad se encuentran aquellos relacionados a la articulación del codo, estos son: flexión-extensión

y pronación-supinación. El primer movimiento tiene la función de acercar o alejar la mano del torso, mientras que el segundo movimiento trabaja en conjunto con la articulación de la muñeca para proporcionar el giro de la mano sobre el eje del antebrazo [5].

Actualmente se han desarrollado una gran variedad de exoesqueletos enfocados a la flexión-extensión de la articulación del codo, cada uno de ellos propone distintas soluciones al control de movimiento. Por ejemplo: el exoesqueleto [6], que es actuado por un motor de corriente directa acoplado a un engranaje helicoidal, utiliza dos sensores de electromiografía y una operación lógica Xor para girar la articulación del codo en la dirección deseada. Por otra parte, el prototipo [7] aplica una estrategia de control que requiere conocer los parámetros tanto del usuario como del actuador. Mientras que en el exoesqueleto del codo para rehabilitación [8], el cual es accionado por cables con memoria de forma, se aplica un control proporcional integral derivativo (PID).

Uno de los retos en el desarrollo de este tipo de exoesqueletos es el diseño e implementación de estrategias de control. El Control proporcional derivativo (PD) más compensación gravitacional es una de las estrategias de control más sencillas de usar en robots manipuladores [9]. Sin embargo, para su implementación en exoesqueletos es necesario conocer parámetros tales como masa y centro de masa de la extremidad sobre la que se coloca el dispositivo.

Para el caso de los exoesqueletos, implementar estrategias de control que se basan el conocimiento de los parámetros del modelo dinámico suele ser complicado debido a la dificultad para obtener el valor numérico de parámetros tales como: masa, centro de masa e inercia de las extremidades del cuerpo humano. Esto se debe a que el cuerpo humano se compone de huesos, músculos y diferentes tejidos que varían en volumen y densidad. Además, estos parámetros son únicos para cada persona.

El presente trabajo consiste en el diseño de un exoesqueleto de asistencia para la flexión-extensión de la articulación del codo y la simulación del control de movimiento aplicando un control PD con compensación adaptable.

El motivo por el que se propone aplicar un control PD con compensación adaptable es que soluciona el problema del control de movimiento aun cuando existe incertidumbre en los parámetros [10]. Además, aunque no garantiza la convergencia de los parámetros hacia los valores reales, si garantiza que el error tienda a cero. Estas características no solo permiten omitir el proceso de medición de los parámetros, sí no que también hacen posible diseñar un exoesqueleto capaz de adaptarse a un amplio rango de usuarios con distintas complexiones.

2. Diseño del prototipo

Para el diseño del prototipo y su posterior análisis cinemático y dinámico solo se consideran 3 grados de libertad [5] los cuales se muestran en las figuras 1 y 2, estos son: flexión-extensión y abducción-aducción de la articulación del hombro; flexión-extensión de la articulación del codo. Cabe destacar que el prototipo solo cuenta con un actuador para la articulación el codo mientras que el usuario tendrá libertad de movimiento en la articulación del hombro.

El dimensionamiento del prototipo se realizó en base a un estudio antropométrico realizado a la población mexicana [11]. De los datos presentados en el estudio, sólo se tomaron en cuenta aquellos relacionados a la extremidad superior, específicamente la distancia entre el hombro, codo y muñeca. En la tabla 1 se muestran las medidas promedio de estatura, peso, longitud del antebrazo y longitud del antebrazo tanto en hombres como en mujeres, el rango de edad es de 18 a 64 años. A partir de estos datos se establecieron las medidas máximas y mínimas para cada uno de los eslabones que componen la estructura del exoesqueleto: longitud del brazo 270-330mm, longitud del antebrazo 220-240mm.

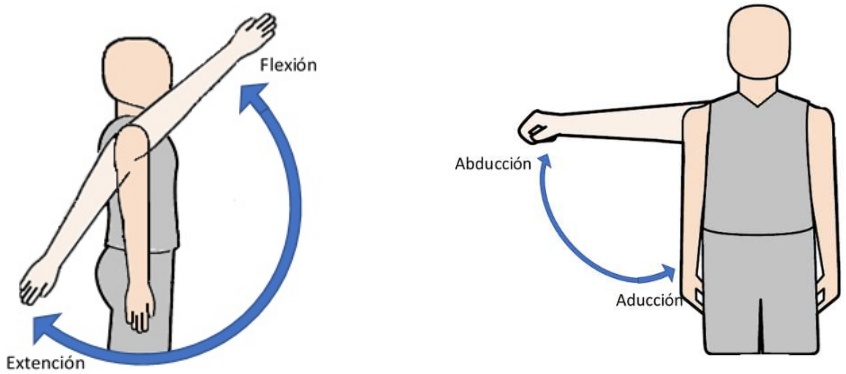


Figura 1. Flexión-extensión y abducción-aducción del hombro

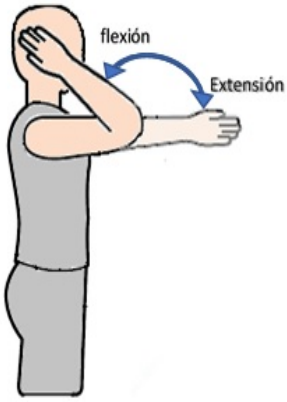


Figura 2. Flexión-extensión del codo

Tabla 1. Medidas antropométricas de la población mexicana.

Descripción	Hombres			Mujeres		
	Percentil 5%	Percentil 50%	Percentil 95%	Percentil 5%	Percentil 50%	Percentil 95%
Peso (kg)	55.31	72.10	97.30	48.00	60.50	88.00
Estatura (mm)	1576	1668	1780	1471	1570	1658
Longitud del brazo (mm)	293	312	335	268	286	300
Longitud antebrazo (mm)	231	243	226	214	228	240

En la figura 3 se muestra el modelo CAD del exoesqueleto, diseñado en la Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla junto con el UMI-LAFMIA del Cinvestav, IPN. Para la articulación del codo se utiliza un servomotor marca Gyems RMD-X8. El movimiento relacionado a la flexión-extensión y abducción-aducción de la articulación del hombro se detecta por medio del sensor MPU6050 el cual es una unidad de medición inercial que contiene un acelerómetro de 3 ejes y un giroscopio de 3 ejes; este sensor fue colocado dentro de una carcasa, a un lado de la articulación del hombro, como se muestra en la figura 4.

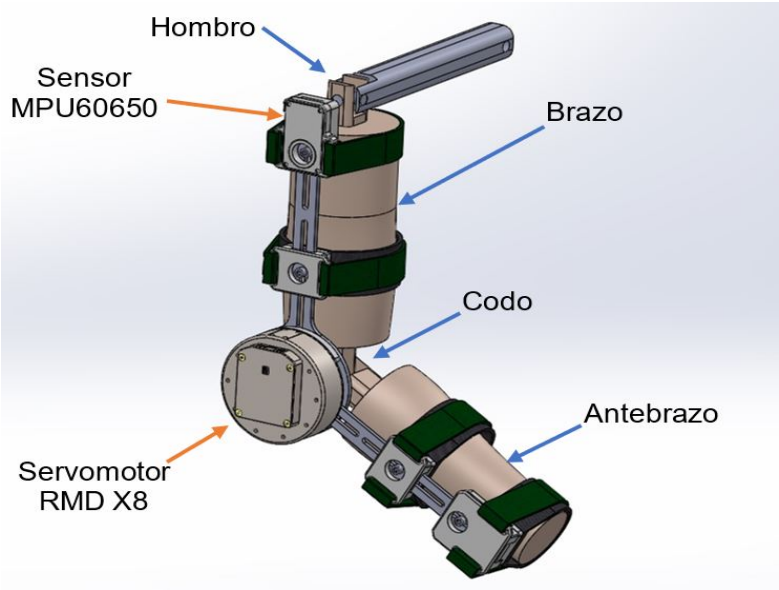


Figura 3. Modelo CAD del exoesqueleto

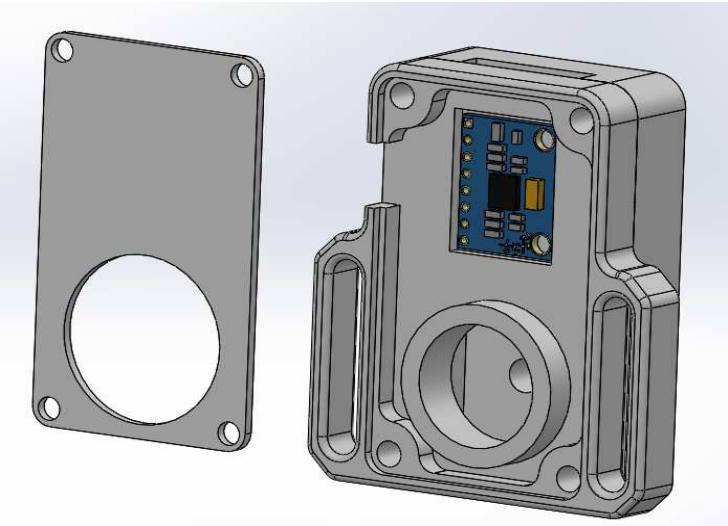


Figura 4. Carcasa del sensor MPU 6050

El prototipo debe estar compuesto por una estructura ligera y capaz de soportar el peso de la extremidad del usuario, así como otras fuerzas externas; considerando estas características se optó por utilizar una aleación de aluminio 6061 para la estructura principal, esta aleación presenta propiedades mecánicas como baja densidad y alta resistencia. Las partes encargadas de sujetar el exoesqueleto al brazo serán manufacturadas usando Nylamid, el cual es un tipo de polímero perteneciente a la familia de las poliamidas.

El peso del prototipo se puede estimar asignando al modelo CAD el tipo de material con el que se fabricará cada pieza. En la tabla 2 se muestra el peso aproximado para cada uno de los eslabones que componen el prototipo. El peso del motor se obtuvo mediante la hoja de datos del fabricante.

Tabla 2. Peso aproximado del prototipo.

Eslabón brazo	peso (gr.)	Eslabón antebrazo	peso (gr)
Barra brazo	79.44	Barra antebrazo	61.55
Sujetador brazo	37.23	Sujetador antebrazo	57.51
Carcaza sensor MPU6050	55.08	Sujetador muñeca	58.86
Motor	630	Soporte Motor	13.23
Total	801.75	Total	191.15
Total	992.9 gr.		

3. Cinemática directa

La cinemática directa contemplando el exoesqueleto junto con la extremidad superior se obtiene de acuerdo con la convención Denavit-Hartenberg (DH) [9]; en la figura 5 se muestra la asignación de los sistemas de referencia. El análisis se realiza sobre el miembro superior derecho, sin embargo, con el fin de hacer un sistema de referencia común para ambos miembros, el sistema de referencia base se ubica sobre el plano sagital entre las articulaciones de los hombros.

Los parámetros DH del sistema se muestran en la tabla 3. Las articulaciones q_1 y q_2 corresponden a la abducción-aducción y a la flexión-extensión del hombro, mientras que q_3 define la flexión-extensión del codo; l_0 representa la longitud media entre ambos hombros, l_2 corresponde a la longitud del brazo medido desde el hombro hasta el codo, l_3 es la longitud del antebrazo medido desde el codo hasta la muñeca.

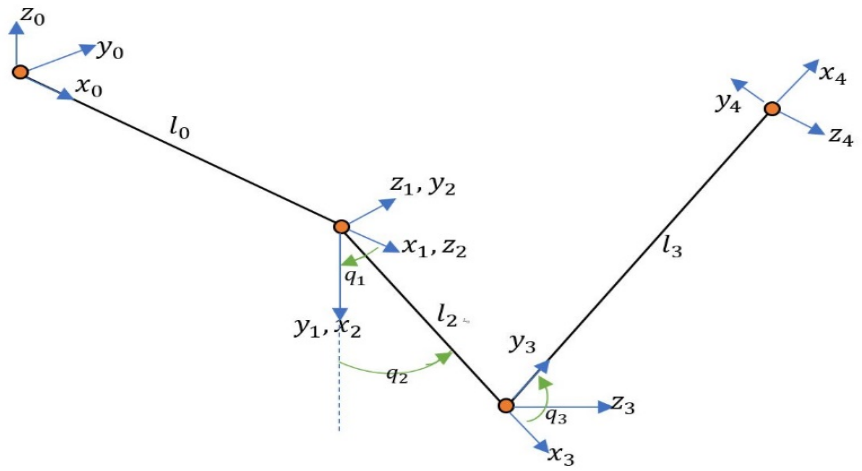


Figura 5. Asignación de los sistemas de coordenadas

Tabla 3. Parámetros DH.

Eslabón	l_i	d_i	α_i	θ_i
0	l_0	0	$-\pi/2$	0
1	0	0	$\pi/2$	q_1
2	l_2	0	0	q_2
3	l_3	0	0	q_3

Las matrices de transformación homogénea para cada articulación, descritas en (1)-(4) se obtienen a partir de la tabla 2.

$$H_0^1 = \begin{bmatrix} 1 & 0 & 0 & l_0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$H_1^2 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$H_2^3 = \begin{bmatrix} C_2 & -S_2 & 0 & l_2C_2 \\ S_2 & C_2 & 0 & l_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$H_3^4 = \begin{bmatrix} C_3 & -S_3 & 0 & l_3C_3 \\ S_3 & C_3 & 0 & l_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

La matriz general de transformación homogénea (5) se obtiene como el producto de las matrices (1)-(4).

$$H_0^4 = H_0^1 H_1^2 H_2^3 H_3^4 = \begin{bmatrix} C_{23}C_1 & -S_{23}C_1 & -S_1 & l_0 + C_1(l_3C_{23} + l_2C_2) \\ S_{23} & C_{23} & 0 & l_3S_{23} + l_2S_2 \\ -C_{23}S_1 & S_{23}S_1 & -C_1 & -S_1(l_3C_{23} + l_2C_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Las funciones senoidales se representan como: $S_1 = \text{sen}\,(q_1)$; $S_2 = \text{sen}\,(q_2)$; $S_{23} = \text{sen}\,(q_2 + q_3)$, mientras que las funciones cosenoidales son: $C_1 = \cos\,(q_1)$; $C_2 = \cos\,(q_2)$; $C_{23} = \cos\,(q_2 + q_3)$.

4. Modelo dinámico

Para obtener el modelo dinámico del exoesqueleto incluyendo la extremidad superior, se emplearon las ecuaciones de movimiento de Euler-Lagrange. Para un robot de n grados de libertad el modelo dinámico se encuentra descrito en forma compacta [9] como:

$$\tau = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \tag{6}$$

Aquí, $M(\boldsymbol{q})$ es la matriz de inercia; $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ es la matriz de fuerzas centrípetas y de Coriolis; $\boldsymbol{g}(\boldsymbol{q})$ es el vector de fuerza y pares gravitacionales; $\ddot{\boldsymbol{q}}$ $\dot{\boldsymbol{q}}$ y $\boldsymbol{q}$ son los vectores de aceleraciones, velocidades y posiciones articulares. La matriz de inercia $M(\boldsymbol{q})$ se puede obtener a partir de [12] como:

$$M(\boldsymbol{q}) = \sum_{i=i}^{n} (m_i J_{vi}^T J_{vi} + J_{wi}^T R_{0i} I_i R_{oi}^T J_{wi}) \tag{7}$$

Donde $\mathrm{m_i}$, J_{vi}, J_{wi}, e I_i son respectivamente la masa, el jacobiano de velocidad lineal, el jacobiano de velocidad angular y el tensor de inercia, todos respectivos al i-ésimo eslabón, mientras que R_{0i} es la matriz de rotación del i-ésimo sistema de referencia con respecto al sistema de referencia fijo. Las longitudes hacia los cetros de masa de los eslabones 2 y 3 medidos respectivamente desde las articulaciones del hombro y codo son l_{c2} y l_{c3}. Para el sistema que se está analizando, lo descrito anteriormente queda definido como:

$$J_{v2} = \begin{bmatrix} -l_{c2}C_2S_1 & -l_{c2}C_1S_1 & 0 \\ 0 & l_{c2}C_2 & 0 \\ -l_{c2}C_1C_2 & l_{c2}S_1 & 0 \end{bmatrix} \tag{8}$$

$$J_{v3} = \begin{bmatrix} -S_1(l_{c3}C_{23} + l_2C_2) & -C_1(l_{c3}S_{23} + l_2S_2) & -l_{c3}S_{23}C_1 \\ 0 & l_{c3}C_{23} + l_2C_2 & l_{c3}C_{23} \\ -C_1(l_{c3}C_{23} + l_2C_2) & s_1(l_{c3}S_{23} + l_2S_2) & l_{c3}S_{23}S_1 \end{bmatrix} \tag{9}$$

$$J_{w2} = \begin{bmatrix} 0 & -S_1 & 0 \\ 1 & 0 & 0 \\ 0 & -C_1 & 0 \end{bmatrix} \tag{10}$$

$$J_{w2} = \begin{bmatrix} 0 & -S_1 & -S_1 \\ 1 & 0 & 0 \\ 0 & -C_1 & -C_1 \end{bmatrix} \tag{11}$$

$$I_2 = \begin{bmatrix} I_{2xx} & 0 & 0 \\ 0 & I_{2yy} & 0 \\ 0 & 0 & I_{2zz} \end{bmatrix} \tag{12}$$

$$I_3 = \begin{bmatrix} I_{3xx} & 0 & 0 \\ 0 & I_{3yy} & 0 \\ 0 & 0 & I_{3zz} \end{bmatrix} \tag{13}$$

$$R_{03} = \begin{bmatrix} C_2C_1 & -S_2C_1 & -S_1 \\ S_2 & C_2 & 0 \\ -C_2S_1 & S_1S_2 & -C_1 \end{bmatrix} \tag{14}$$

$$R_{04} = \begin{bmatrix} C_{23}C_1 & -S_{23}C_1 & -S_1 \\ S_{23} & C_{23} & 0 \\ -C_2S_1 & S_{23}S_1 & -C_1 \end{bmatrix} \tag{15}$$

Por lo tanto, a partir de las ecuaciones (8)-(15) cada elemento de la matriz $M(\boldsymbol{q})$ queda definido como:

$$M(\boldsymbol{q}) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{16}$$

$$m_{11} = (I_{1xx} - l_{c2}^2m_1)s_2^2 + \left(I_{1yy} + l_2^2m_2\right)C_2^2 + \left(I_{2yy} + l_{c3}^2m_2\right)C_{23}^2 + I_{2xx}S_{23}^2 + 2l_2l_{c3}m_2C_2C_{23} + l_{c2}^2m_1$$

$$m_{12} = 0$$

$$m_{13} = 0$$

$$m_{21} = 0$$

$$m_{22} = l_2^2m_2 + 2l_2l_{c3}m_2C_3 + l_{c3}^2m_2 + l_{c2}^2m_1 + I_{1zz} + I_{2zz}$$

$$m_{23} = l_2l_{c3}m_2C_3 + l_{c3}^2m_2 + I_{2zz}$$

$$m_{31} = 0$$

$$m_{32} = l_2l_{c3}m_2C_3 + l_{c3}^2m_2 + I_{2zz}$$

$$m_{33} = l_{c3}^2m_2 + I_{2zz}$$

La matriz $C(\boldsymbol{q},\dot{\boldsymbol{q}})$ se puede obtener mediante los símbolos de Christoffel [13]:

$$c_{ijk}(q) = \frac{1}{2}\left[\frac{\partial M_{kj}(q)}{\partial q_i} + \frac{\partial M_{ki}(q)}{\partial q_j} + \frac{\partial M_{ij}(q)}{\partial q_k}\right] \tag{17}$$

Donde M_{ij} denota el ij-ésimo elemento de la matriz $M(\boldsymbol{q})$. El kj-ésimo elemento de la matriz $C(\boldsymbol{q},\dot{\boldsymbol{q}})$ se obtiene de:

$$c_{kj}(q,\dot{q}) = \begin{bmatrix} c_{1jk}(q) \\ c_{2jk}(q) \\ \cdots \\ c_{njk}(q) \end{bmatrix}^T \dot{\boldsymbol{q}} \tag{18}$$

Entonces a partir de las ecuaciones (16), (17) y (18) la matriz $C(\boldsymbol{q},\dot{\boldsymbol{q}})$ es:

$$C(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{19}$$

$$c_{11} = \left(I_{1yy} - I_{1xx} + l_2^2 m_2 + l_{c2}^2 m_1\right) C_2 S_2 \dot{q}_2 + \left(I_{2yy} - I_{2xx} + l_{c3}^2 m_2\right) C_{23} S_{23} (\dot{q}_2 + \dot{q}_3)$$
$$- l_2 l_{c3} m_2 (C_{23} S_2 + S_{23} C_2) \dot{q}_2 - l_2 l_{c3} m_2 C_2 S_{23} \dot{q}_3$$

$$c_{12} = \left(I_{1yy} - I_{1xx} + l_2^2 m_2 + l_{c2}^2 m_1\right) C_2 S_2 \dot{q}_1 + \left(I_{2yy} - I_{2xx} + l_{c3}^2 m_2\right) C_{23} S_{23} \dot{q}_1 + l_2 l_{c3} m_2 (C_{23} S_2 + S_{23} C_2) \dot{q}_1$$

$$c_{13} = \left(I_{2xx} - I_{2yy} - l_{c3}^2 m_2\right) C_{23} S_{23} \dot{q}_1 - l_2 l_{c3} m_2 C_2 S_{23} \dot{q}_1$$

$$c_{21} = \left(I_{1yy} - I_{1xx} + l_2^2 m_2 + l_{c2}^2 m_1\right) C_2 S_2 \dot{q}_1 + \left(I_{2yy} - I_{2xx} + l_{c3}^2 m_2\right) C_{23} S_{23} \dot{q}_1 + l_2 l_{c3} m_2 (C_{23} S_2 + S_{23} C_2) \dot{q}_1$$

$$c_{22} = -l_2 l_{C3} m_2 S_3 \dot{q}_3$$

$$c_{23} = -l_2 l_{c3} m_2 S_3 (\dot{q}_2 + \dot{q}_3)$$

$$c_{31} = \left(I_{2yy} - I_{2xx} + l_{c3}^2 m_2\right) C_{23} S_{23} \dot{q}_1 + l_2 l_{c3} m_2 C_2 S_{23} \dot{q}_1$$

$$c_{32} = l_2 l_{c3} m_2 S_3 \dot{q}_3$$

$$c_{33} = 0$$

El vector de fuerzas y pares gravitacionales se puede obtener a partir de [8] como:

$$\boldsymbol{g}(\boldsymbol{q}) = \frac{\partial U(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{20}$$

La energía potencial $U(\boldsymbol{q})$ del sistema es:

$$U(\boldsymbol{q}) = m_1 l_{c2} g (1 - C_2 S_1) + m_2 g (l_2 + l_{c3} - S_1 (l_2 C_2 + l_{c3} C_{23})) \tag{21}$$

Entonces aplicando la ecuación (20):

$$\boldsymbol{g}(\boldsymbol{q}) = \begin{bmatrix} -m_2 l_{c3} C_1 C_{23} - (m_2 l_2 + l_{c2} m_1)\, C_1 C_2 \\ m_2 l_{c3} S_1 S_{23} + (m_2 l_2 + l_{c2} m_1) S_1 S_2 \\ l_{c3} m_2 S_{23} S_1 \end{bmatrix} \tag{22}$$

5. Control de movimiento

Una vez que se conoce la cinemática y dinámica el sistema, se procede a proponer una ley de control que permita moverlo siguiendo una trayectoria deseada. Para tal motivo se debe tomar en cuenta que la única articulación en la que se utiliza un actuador es q_3 que corresponde a los movimientos de flexión-extensión del codo. Entonces, la ley de control se diseñará específicamente para que q_3 siga una trayectoria deseada.

Otros aspectos importantes para seleccionar la ley de control adecuada es que el exoesqueleto será utilizado por un rango de personas cuyo peso y estatura varía, además, el usuario podrá manipular objetos, esto conlleva a que exista una incertidumbre en los parámetros del modelo dinámico y por lo tanto se consideran desconocidos. Por otra parte, el usuario podrá mover libremente la articulación del hombro; por lo cual, para detectar las posiciones, velocidades y aceleraciones de las articulaciones q_1 y q_2 referentes a los movimientos de flexión-extensión y aducción-abducción del hombro, se hace uso de una unidad de medición inercial.

En base a lo anterior se propone aplicar un control de tipo PD con compensación adaptable [10], cuya principal característica es que permite resolver el problema de control de movimiento cuando existe incertidumbre en los parámetros del modelo dinámico.

Al utilizar una ley de control adaptable es necesario aplicar la propiedad de linealidad en los parámetros del modelo dinámico [10] de tal forma que este pueda reescribirse como:

$$M(\boldsymbol{q},\boldsymbol{\theta})\boldsymbol{u} + C(\boldsymbol{q},\boldsymbol{\omega},\boldsymbol{\theta})\boldsymbol{v} + g(\boldsymbol{q},\boldsymbol{\theta}) = \Phi\boldsymbol{\theta} \tag{23}$$

Donde θ es un vector de parámetros desconocidos, Φ es una matriz de funciones no lineales, $\boldsymbol{u} = \ddot{\boldsymbol{q}}$, y $\boldsymbol{v} = \boldsymbol{\omega} = \dot{\boldsymbol{q}}$. Para el caso de la articulación que corresponde a la flexión-extensión del codo, la parametrización resultante es:

$$\tau_3 = \Phi\theta = [\Phi_1 \quad \Phi_2 \quad \Phi_3 \quad \Phi_4]\begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\\ \theta_4\end{bmatrix} \tag{24}$$

$$\boldsymbol{\theta} = \begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\\ \theta_4\end{bmatrix} = \begin{bmatrix} I_{2zz} + l_{C3}^2 m_2 \\ l_2 l_{c3} m_2 \\ I_{2yy} - I_{2xx} + l_{c3}^2 m_2 \\ l_{c3} m_2 \end{bmatrix}$$

$$\boldsymbol{\Phi} = \begin{bmatrix}\Phi_1\\ \Phi_2\\ \Phi_3\\ \Phi_4\end{bmatrix}^T = \begin{bmatrix} u_2 + u_3 \\ u_2 C_3 + \omega_1^2 S_{23} C_2 + \omega_2^2 S_3 \\ \omega_1^2 C_{23} S_{23} \\ S_1 S_{23} g \end{bmatrix}^T$$

De forma general, la ley de control PD con compensación adaptable junto con su parametrización se define en las ecuaciones (25) y (26), mientras que el vector de parámetros adaptables $\widehat{\boldsymbol{\theta}}$ se describe en la ecuación (27):

$$\tau = K_p\tilde{\boldsymbol{q}} + K_v\dot{\tilde{\boldsymbol{q}}} + M(\boldsymbol{q},\widehat{\boldsymbol{\theta}})[\ddot{\boldsymbol{q}}_d + \Lambda\dot{\tilde{\boldsymbol{q}}}] + C(\boldsymbol{q},\dot{\boldsymbol{q}},\widehat{\boldsymbol{\theta}})[\dot{\boldsymbol{q}}_d + \Lambda\tilde{\boldsymbol{q}}] + \boldsymbol{g}(\boldsymbol{q},\widehat{\boldsymbol{\theta}}) \tag{25}$$

$$\tau = K_p\tilde{\boldsymbol{q}} + K_v\dot{\tilde{\boldsymbol{q}}} + \Phi\widehat{\boldsymbol{\theta}} \tag{26}$$

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{t}) = \Gamma\int_0^t \boldsymbol{\Phi}^T\left[\dot{\tilde{q}} + \Lambda\tilde{q}\right]ds + \widehat{\boldsymbol{\theta}}(0) \tag{27}$$

Donde K_p, K_v y Γ son matrices definidas positivas y $\Lambda = K_v^{-1}K_p$. Para el caso en el que solo se desea controlar la flexión-extensión de la articulación del codo, la ley de control queda definida como:

$$\tau_3 = k_p\tilde{q}_3 + k_v\dot{\tilde{q}}_3 + \boldsymbol{\Phi}\widehat{\boldsymbol{\theta}} \tag{28}$$

El vector Φ corresponde al obtenido en la ecuación (24); los vectores $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{\omega}$ y $\widehat{\boldsymbol{\theta}}$ quedan definidos como:

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_{d3} + \lambda\tilde{q}_3 \end{bmatrix} \tag{29}$$

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_{d3} + \lambda\tilde{q}_3 \end{bmatrix} \tag{30}$$

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{31}$$

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{t}) = \Gamma\int_0^t \boldsymbol{\Phi}^T\,(\dot{\tilde{q}}_3 + \lambda\tilde{q}_3)ds + \widehat{\boldsymbol{\theta}}(0) \tag{32}$$

Donde $k_p > 0$, $k_v > 0$, $\Gamma \in \mathbb{R}^{4x4} > 0$ y $\lambda = k_p/k_v$. En la figura 6 se muestra el diagrama a bloques del sistema en lazo cerrado.

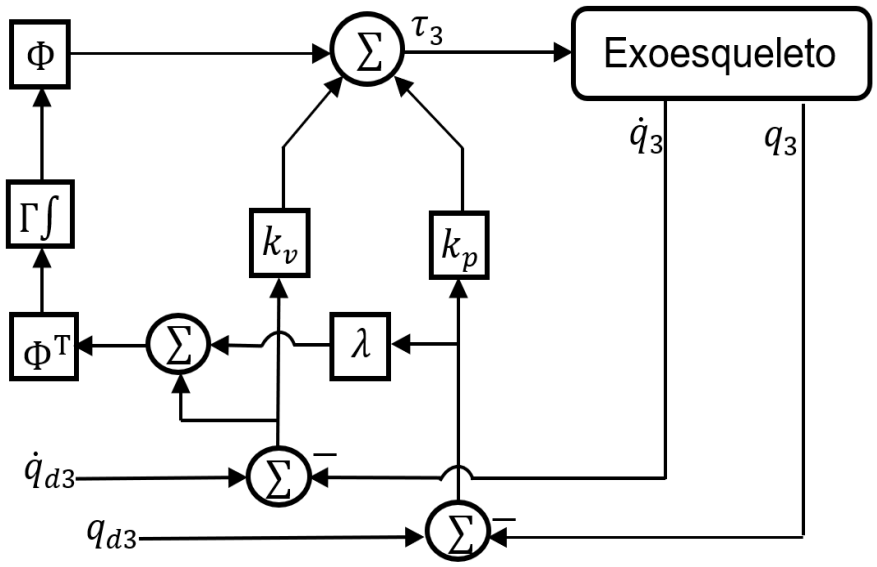


Figura 6. Diagrama a bloques del sistema en lazo cerrado

6. Simulación

Para la etapa de simulación se empleó el entorno Simulink el cual es una herramienta incluida en MATLAB para el modelado, simulación y análisis de sistemas dinámicos mediante diagramas de bloques.

En la figura 7 se muestra el diagrama a bloques que representa al exoesqueleto, la extremidad superior del usuario y la ley de control (28) en lazo cerrado. En el bloque 1 se encuentran las ecuaciones que representan las posiciones, velocidades y aceleraciones para cada articulación, en el bloque 2 se encuentra la ley de control PD con compensación adaptable, en bloque 3 se realiza el cálculo del vector de parámetros adaptables $\widehat{\boldsymbol{\theta}}$, por último, la retroalimentación se obtiene del bloque 4 que es donde encuentra el modelo dinámico del sistema; este bloque fue creado en Simscape.

Simscape es una herramienta para crear modelos de sistemas físicos dentro del entorno de Simulink mediante su representación en diagrama de bloques. Simscape, además permite Importar ensamblajes desde software CAD y añade soporte para la simulación del sistema físico en 3D. En las figuras 8 y 9 se muestra el diagrama a bloques con el que se representa al exoesqueleto junto con la extremidad superior del usuario, así como el modelo 3D resultante, respectivamente.

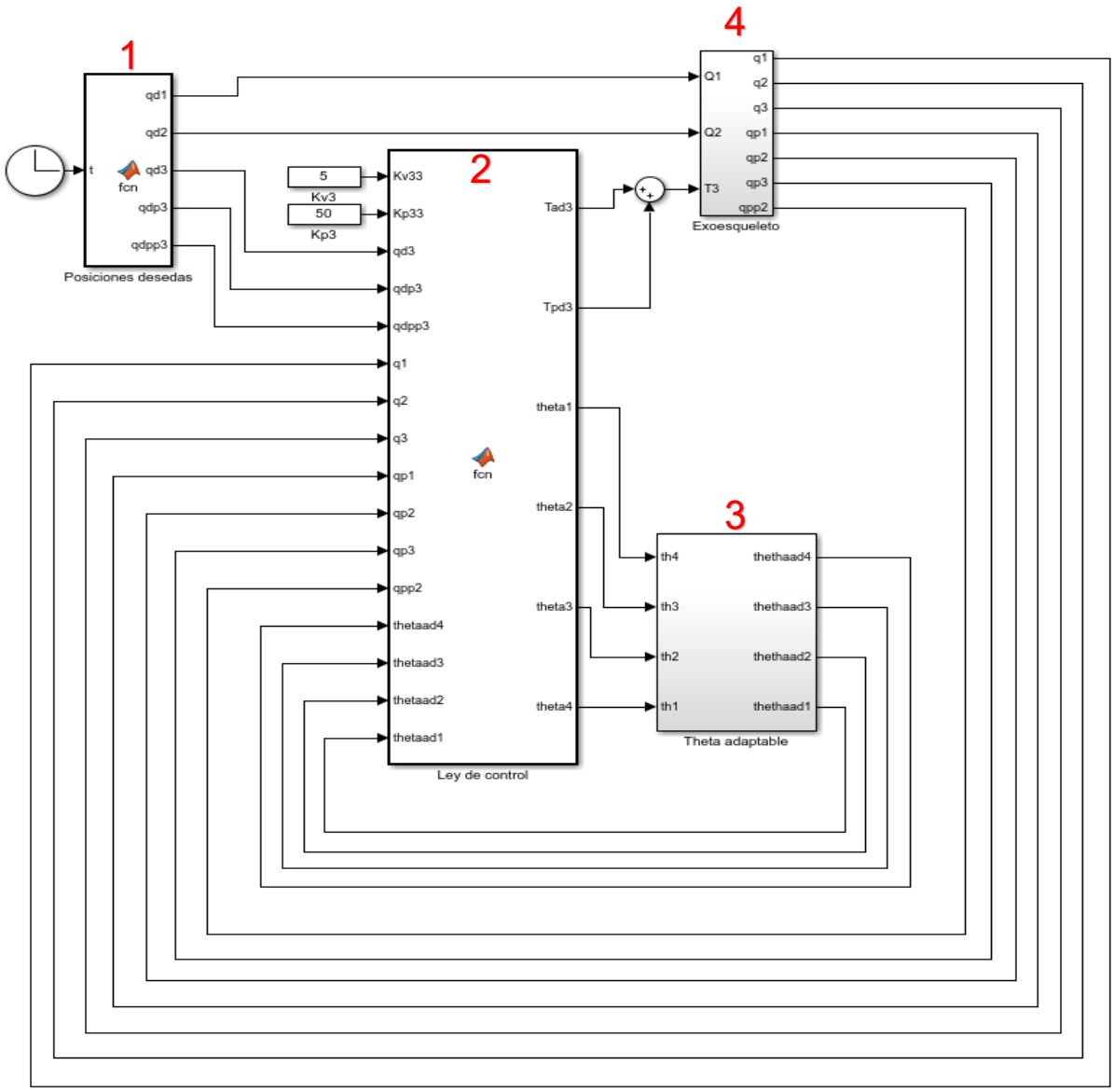


Figura 7. Implementación en Simulink del sistema en lazo cerrado

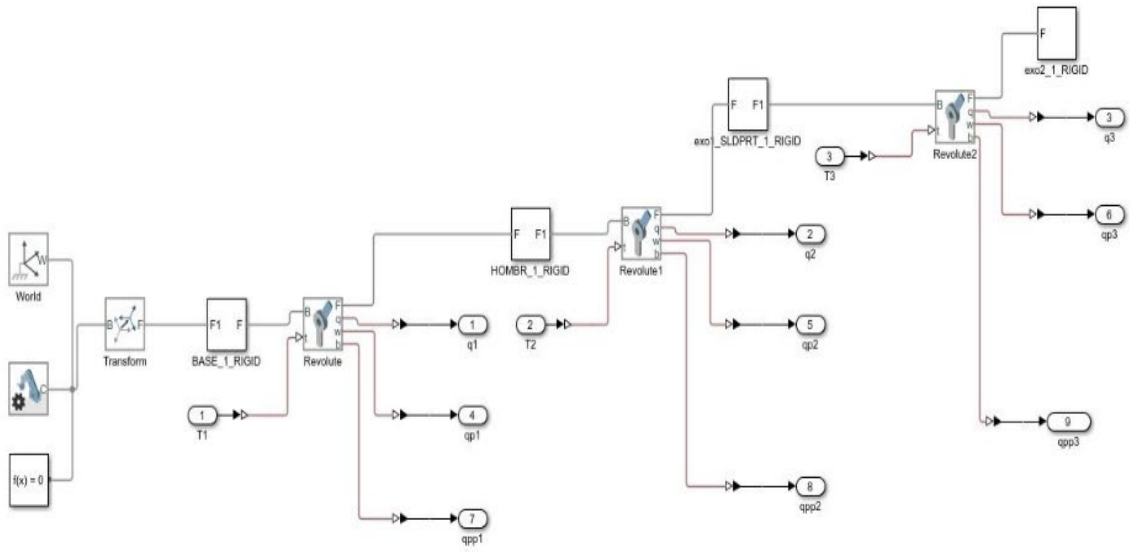


Figura 8. Representación en diagrama a bloques del exoesqueleto junto con la extremidad superior del usuario (desarrollado en Simscape)

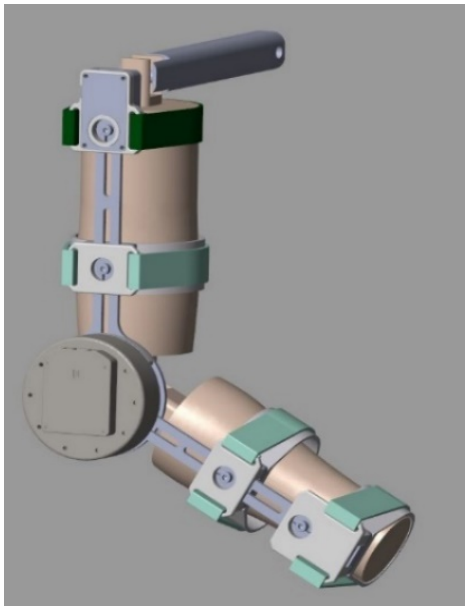


Figura 9. Modelo 3D del diagrama a bloques de la figura 7 (desarrollado en Simscape)

La trayectoria que se emplea en la simulación consiste en una rutina de movimiento de cuatro posiciones que abarcan la mayor parte del rango de movimiento de cada articulación. Esta rutina de movimiento se muestra en la figura 10.

El movimiento de las articulaciones q_1 y q_2 por parte del usuario, se representa por medio de las ecuaciones (33) y (36), estas ecuaciones se ingresan a cada uno de los bloques que representan la flexión-extensión y abducción-aducción del hombro, al hacer esto Simscape calculará automáticamente las velocidades y aceleraciones de dichas articulaciones y entregará estos datos a la salida del bloque 4. Mientras que para q_3 las posiciones velocidades y aceleraciones deseadas quedan definidas en las ecuaciones (38)-(40).

$$q_1 = (55(\cos(t) + 1) - 20)\left(\frac{\pi}{180}\right) \tag{33}$$

$$\dot{q}_1 = -55\sin(t)\left(\frac{\pi}{180}\right) \tag{34}$$

$$\ddot{q}_1 = -55\cos(t)\left(\frac{\pi}{180}\right) \tag{35}$$

$$q_2 = (55\sin(.5t + 1.780\pi) + 35)\left(\frac{\pi}{180}\right) \tag{36}$$

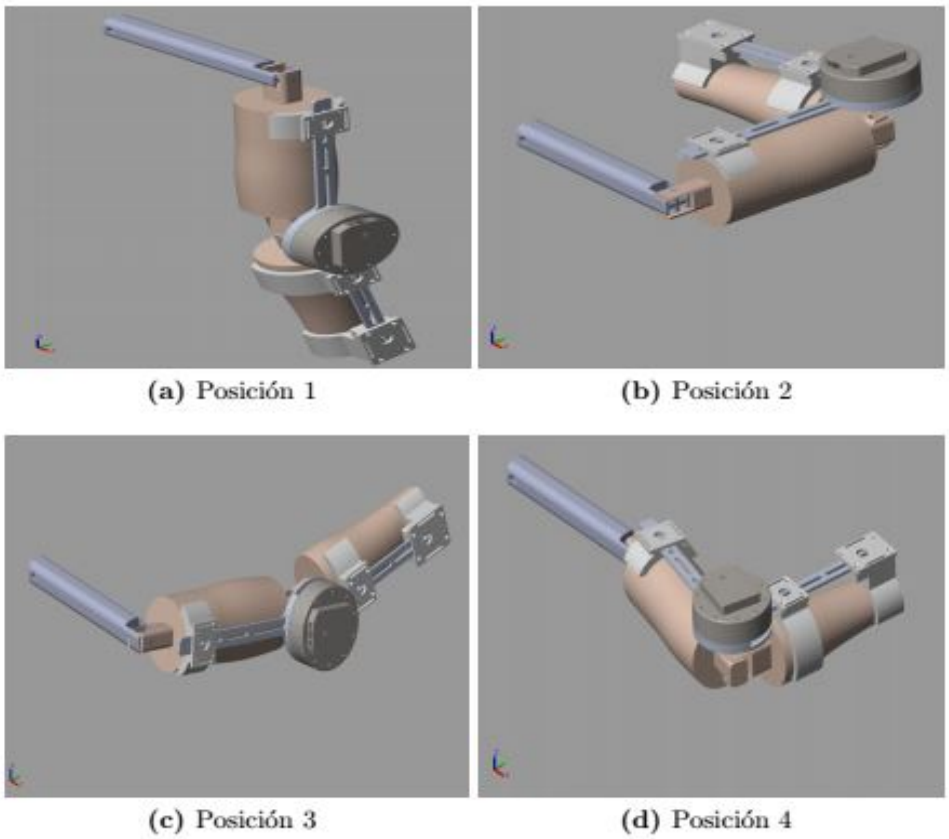
$$\dot{q}_2 = (27.5\cos(.5t + 1.780\pi))\left(\frac{\pi}{180}\right) \tag{37}$$

$$\ddot{q}_2 = (-13.75\sin(.5t + 1.780\pi))\left(\frac{\pi}{180}\right) \tag{38}$$

$$q_{d3} = (50(1 - \cos(t)) + 15)\left(\frac{\pi}{180}\right) \tag{39}$$

$$\dot{q}_{d3} = 50(\sin(t))\left(\frac{\pi}{180}\right) \tag{40}$$

$$\ddot{q}_{d3} = 50(\cos(t))\left(\frac{\pi}{180}\right) \tag{41}$$



(a) Posición 1 (b) Posición 2

(c) Posición 3 (d) Posición 4

Figura 10. Rutina de movimiento

7. Resultados

A continuación, se muestran los resultados de la simulación. Las ganancias seleccionadas son: $k_p = 50,\ k_v = 5,\ \Gamma = diag\{.10\ .24\ 1.3\ .25\}$; estos valores se obtuvieron de forma heurística; en primer lugar, se determinaron las ganancias proporcional y derivativa considerando un lazo de control PD sin compensación, mientras que para la matriz de ganancias adaptables Γ se consideró un valor inicial pequeño [10]. El tiempo de simulación es $t = 10s$. En la figura 11 se muestra la estimación de los parámetros adaptables, en donde se puede notar que los parámetros estimados convergen a los siguientes valores:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \hat{\theta}_3 \\ \hat{\theta}_4 \end{bmatrix} = \begin{bmatrix} I_{2zz} + l_{C3}^2 m_2 \\ l_2 l_{c3} m_2 \\ I_{2yy} - I_{2xx} + l_{c3}^2 m_2 \\ l_{c3} m_2 \end{bmatrix} = \begin{bmatrix} 0.02803 \\ 0.0465 \\ 0.01849 \\ 0.2076 \end{bmatrix} \tag{44}$$

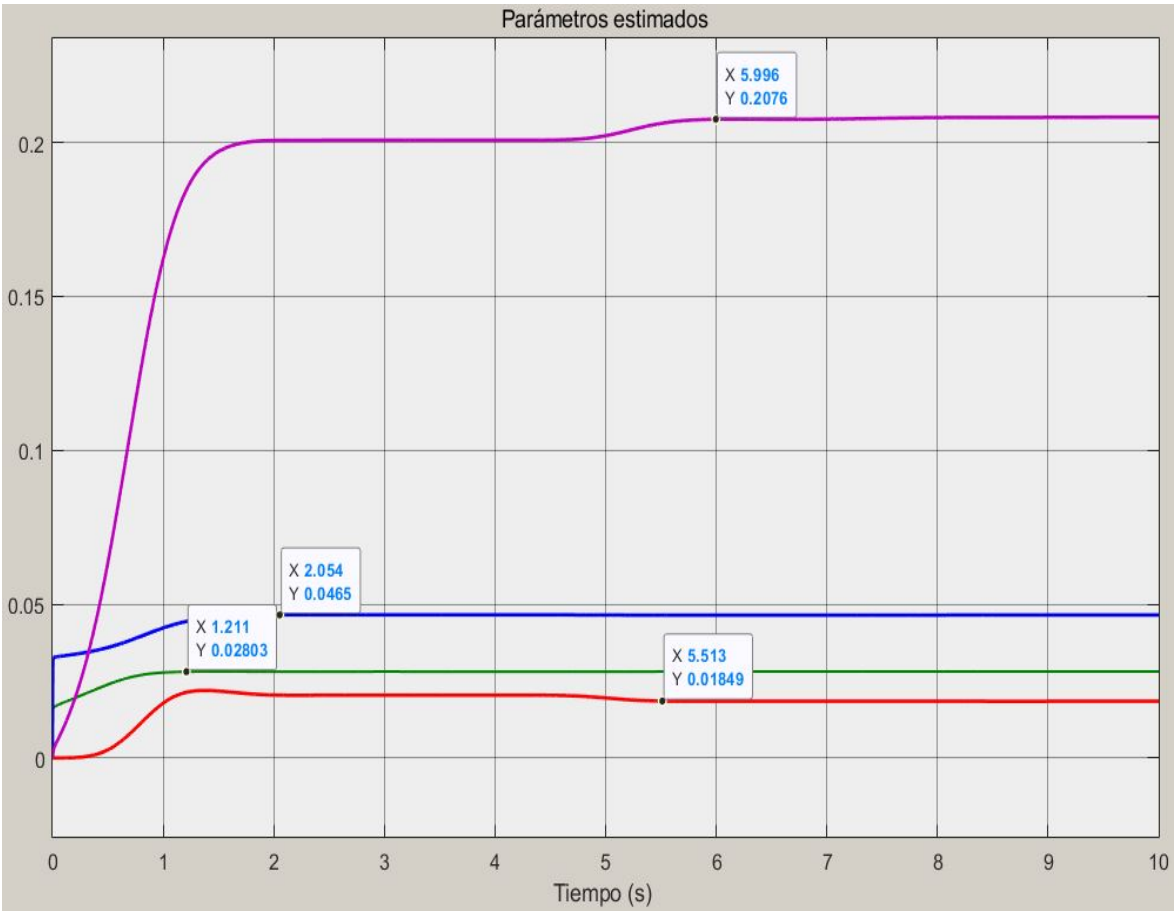


Figura 11. Parámetros estimados

En la figura 12 se muestra error de posición para la articulación q_3, se puede notar que al comienzo de la simulación se presenta un error que alcanza un máximo de 0.83º, sin embargo, conforme el tiempo avanza, el error tiende a cero, si se compara este resultado con la figura 10 se pude notar que el error disminuye conforme la estimación de parámetros converge a un valor.

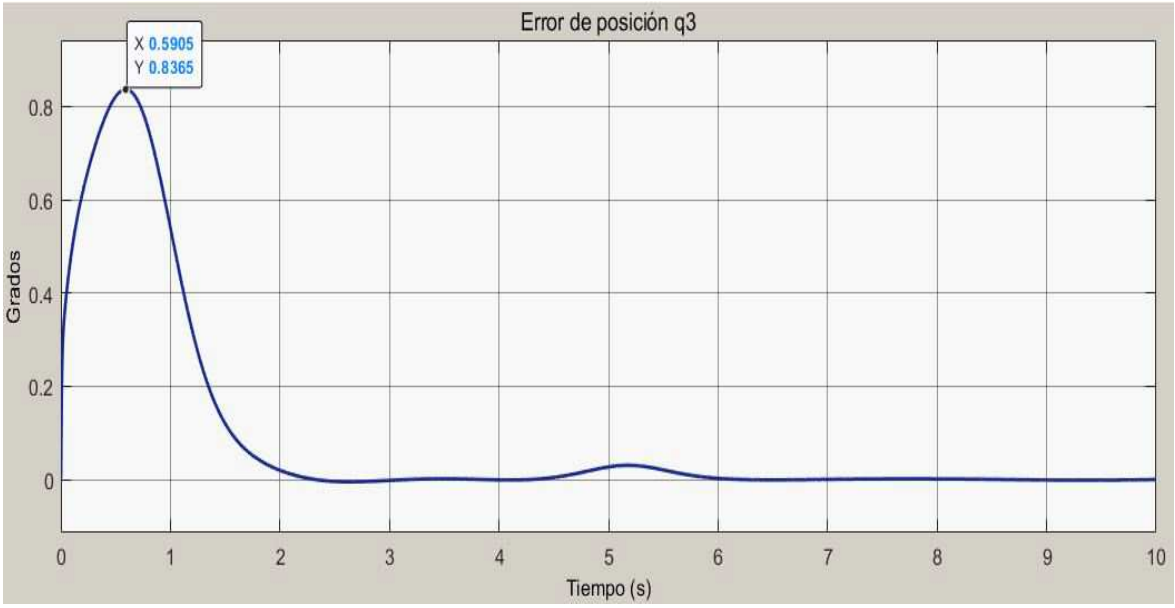


Figura 12. Error de posición para la articulación q_3

En la figura 13 se muestra el torque suministrado a la articulación del codo. Con fines de análisis, el torque se dividió en dos componentes: el torque proporcionado por la parte proporcional derivativa y el torque suministrado por la compensación adaptable. Se puede notar que al igual que con el error, la contribución del torque PD disminuye conforme los valores de los parámetros se estiman quedando únicamente la compensación adaptable.

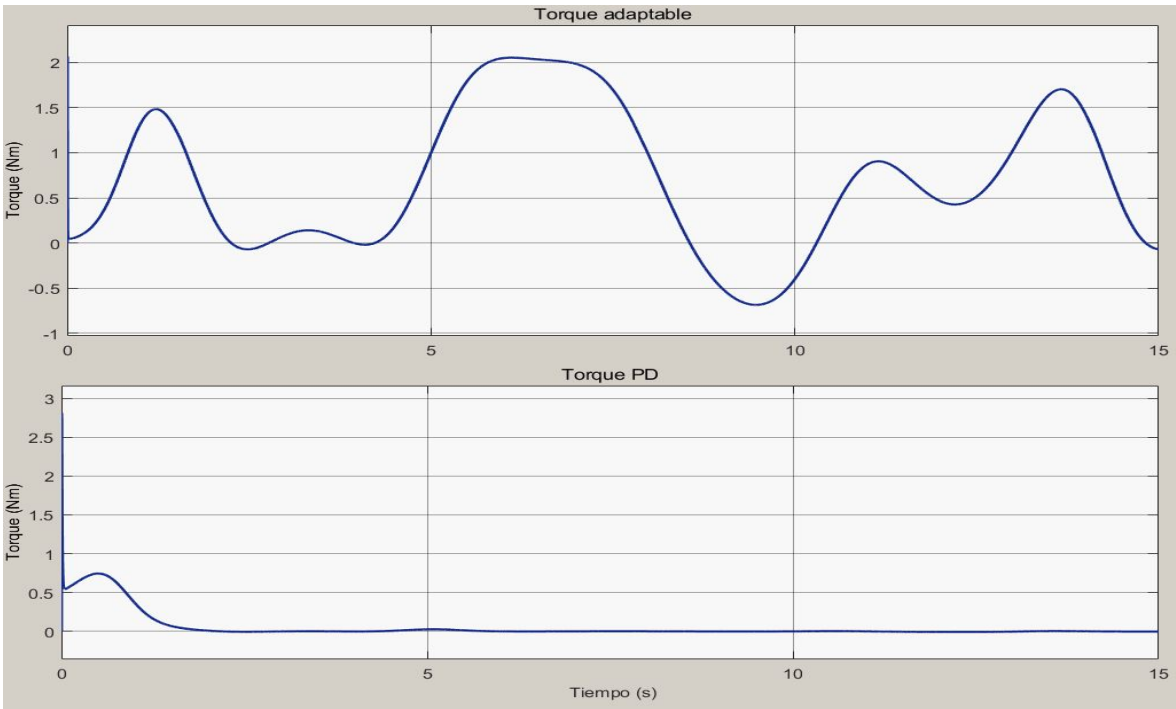


Figura 13. Torque suministrado a la articulación q_3

8. Conclusiones

El prototipo se diseñó con la finalidad de ser una estructura ligera y que pueda ser utilizada en las actividades diarias. La ventaja principal de este prototipo es: que al poseer una estructura que se adapta a un amplio rango de personas con distintos pesos y estaturas se puede omitir el proceso de medición de los parámetros reales; el cual si no se tienen las herramientas adecuadas resulta ser un proceso complicado.

Por medio de la simulación del sistema en lazo cerrado se verificó que, aunque el control PD con compensación adaptable no garantiza la convergencia de los parámetros hacia los valores reales, si garantiza que el error tienda a cero y por lo tanto que el objetivo de control de movimiento se cumpla.

Una vez que se ha comprobado que aplicar un control PD con compensación adaptable para controlar la flexión-extensión de la articulación del codo resulta ser una opción viable, como se ha demostrado aquí, como un trabajo complementario en un futuro cercano se procederá a manufacturar la estructura mecánica del prototipo, también se llevará a cabo caracterización de las señales provenientes del sensor MPU6050 y se integrará al prototipo un sensor de electromiografía que será utilizado para accionar el servomotor. Por último, se realizarán pruebas experimentales y se compararán los resultados obtenidos durante la simulación.

Referencias

[1] Rosen J. “Wearable robotics”, Academic press, USA, Primera edición, 2019.
[2] Hernández J., López R., Salazar S., and Lozano R. *“Standing Mobility Exoskelton Device”*, 16th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE, pp. 1-6, Mexico city, 2019
[3] Yin S., Kuan W., Ching L., Ying Y. and Chao L. *“Design of a Lightweight Forearm Exoskeleton for Fine-Motion Rehabilitation”*, Conference on Advanced Intelligent Mechatronics (AIM), IEEE/ASME, pp. 438-443, Auckland, New Zealand, 2018.
[4] González A., López R., Pérez A., and Salazar S. *“Upper Limb Musculoskeletal Modeling for Human-Exoskeleton Interaction”*, 16th International Conference on Electrical Engineering, Computing Science and Automatic Control, CCE, pp. 1-5, Mexico city, 2019.
[5] Pons J. “Wearable robots: biomechatronic exoskeletons”, John Wiley & Sons, Inglaterra, Primera edición, 2008.
[6] Anand A. and Vigneshwar M. *“Design and Control of an EMG-based Low-cost Exoskeleton forStroke Rehabilitation”*, Fifth Indian Control Conference, ICC, pp 478-483, Delhi, India, 2019.
[7] Borhan B., Mahdi I. and Sohrab N. *“Design and Development of One Degree of Freedom Upper Limb Exoskeleton”*, 3rd RSI International Conference on Robotics and Mechatronics, pp 223-228, Tehran, Iran, 2015
[8] Copaci D., Martín F., Moreno F. y Blanco D. *“SMA Based Elbow Exoskeleton for Rehabilitation Therapy and Patient Evaluation”*, IEEE Access. IEEE, Vol. 7, páginas, 31473-31484, 2019.
[9] Reyes F. *“Robótica. Control de robots manipuladores”,* Alfaomega, México, Primera edición, 2011.
[10] Kelly R., Santibáñez V. y Loria A. “Control of robot manipulators in joint space”, Springer, Inglaterra, primera edición, 2005.
[11] Ávila R., Prado L., González E. *“Dimensiones antropométricas de población latinoamericana”,* Universidad de Guadalajara, México, Segunda edición, 2007.
[12] Kumar S. *“Introducción a la robótica”,* Mc Graw Hill, México, Primera edición, 2009.
[13] Kelly R., Santibáñez V. *“Control de movimiento de robots manipuladores”,* Pearson, Madrid, Primera edición, 2003.

Diseño Colaborativo en Mecatrónica, Capítulo 5, pp. 52 - 67
ISBN: 978-607-9394-23-3, 2021.

Mapeo y Localización Simultánea (SLAM) con un robot tipo esfera mediante un enfoque minimalista utilizando ROS

Olalde Mendoza Omero Nicolás, Nava Balanzar Luciano, Guillen Soto José María, Jiménez Oronia Francisco, Rodríguez Olivares Noé Amir

Dirección de Ingeniería Eléctrica y Electrónica
Centro de Ingeniería y Desarrollo Industrial
nico95olamen@gmail.com

Resumen

La exploración en entornos desconocidos, como lo suelen ser las tareas de exploración, proporciona información útil para arqueología, antropología, entre otras disciplinas. Una tecnología prometedora para la exploración de estos entornos son los robots de tipo esfera; sin embargo, actualmente no son tan utilizados debido a sus limitaciones inherentes. El desplazamiento de un robot de tipo esfera puede llevarse a cabo a través del movimiento de su centro de masa, lo que provoca que el casco del robot ruede sobre la superficie en la que se encuentra, esto complica la implementación de sensores externos que le permiten percibir su entorno, una solución novedosa es el aprovechamiento de los sensores internos. El presente trabajo presenta la implementación de un algoritmo con un enfoque de robótica minimalista, es decir, utilizando la menor cantidad de sensores que permitan tener la percepción del ambiente donde se encuentra el robot tipo esfera, con la finalidad de resolver problemas complejos como lo es el SLAM que por sus siglas en ingles significa Mapeo y Localización Simultáneas, el algoritmo se desarrolló para un robot Sphero$^{M.R}$ el cual es un robot comercial desarrollado por la empresa Orbotix, mediante la utilización de ROS (Robot Operating System). Este estudio está orientado a la reconstrucción de entornos rectilíneos con superficies planas y sin inclinación.

Palabras clave: Detección mínima, Modelos probabilísticos, Robótica minimalista, SLAM, *Sphero*$^{M.R}$

1. Introducción

La robótica móvil ha cobrado una creciente demanda que comenzó a finales del siglo pasado (1980-1990) [1], debido a que la brecha entre el lujo de usar un robot móvil y la necesidad es cada vez más pequeña, gran parte se atribuye a la amplia gama de aplicaciones en las que se puede emplear un robot móvil, tanto en aplicaciones dentro de la industria como en la investigación [2], existe una varíate tan amplia de robots móviles que se pueden clasificar como terrestres, aéreos y acuáticos [3]. Una particularidad que no todos los robots móviles tienen es la autonomía, de ahí el término, robots autónomos, este término, hace referencia a la respuesta del robot ante diversas situaciones sin supervisión humana o de un mapa que pueda seguir [4], a este problema se le denomina SLAM que por sus siglas en ingles significa Mapeo y Localización Simultáneas. El SLAM trata el problema de que un robot que se mueve en un ambiente desconocido debe ser capaz de crear un mapa con base a la información que le entregan los sensores que se encuentran a bordo de este, esta información se encuentra corrompida por el ruido que puede tener el sistema, a su vez el robot debe saber en qué punto de ese mapa se encuentra para no perderse [6][7][8]. En los últimos años el SLAM ha sido un objeto serio de estudio y ha proporcionado diversos algoritmos prácticos para hacerle frente a este problema [4][5], donde la mayoría de estos algoritmos se basan en métodos de estimación Bayesiana [9], debido a que los métodos probabilísticos han demostrado una mejor respuesta, esto se debe a que los métodos probabilísticos consideran las diversas fuentes de incertidumbre, como lo es el ruido de los sensores y actuadores, la deficiencia de cómputo, el error en el modelado del robot o perturbaciones del mismo ambiente [10]. Algunos de los algoritmos más conocidos y trabajados hoy en día son, el FK

(Filtro de Kalman), el FP (Filtro de Partículas), FKE (Filtro de Kalman Extendido) entre otros [11]. El objeto del estudio presentado en esta investigación es el poder reconstruir un ambiente rectilíneo mediante la utilización de un robot Sphero$^{M.R}$ desarrollado por la empresa Orbotix, el cual se cuenta equipado con un sistema sensorial conformado por una unidad de medición inercial (IMU), y un acelerómetro de tres ejes.

En este estudio se discute si las detecciones mínimas como la variación de velocidad, la aceleración y la variación de los ángulos de Euler son suficientes para resolver el problema del SLAM. La locomoción del robot está controlada por un algoritmo de seguimiento de paredes, es decir el robot sigue una pared determinada hasta encontrarse con una esquina, el robot debe determinar si es o no es una esquina y continuar siguiendo la pared, la detección de las esquinas es de suma importancia debido que las esquinas son marcas características del ambiente. La interpretación del ambiente se visualiza en un mapa que se va contrayente mediante el robot avanza. Las características del ambiente se representan mediante puntos de referencia, en los cuales se asume que tienen un error de estimación gaussiano, para determinar la pose se implementó el filtro de Kalman, puesto el sistema cumple con los supuestos que más adelante se exponen.

2. Otras soluciones

Como ya se ha mencionado la robótica móvil fungen un papel importante en las tareas de exploración, esto se ve reflejado en los estudios que se han realizado alrededor de estos temas. Los diversos estudios realizados han expuesto múltiples soluciones utilizando una gama variada de sensores, vehículos y sobre todo diversos algoritmos como lo es estudio presentado en [15], donde realizan la reconstrucción tridimensional de un ambiente interno controlado, mediante la utilización de un vehículo autónoma de 4 ruedas, utilizan una cámara RGB-D como sensor principal para obtener las características del ambiente, para el sistema de percepción visual se implementa un algoritmo de reducción de información 3D a 2D, similar a la información suministrada por sensores de rango y para la estimación de la pose implementan el filtro de partículas. En [16] tiene como objetivo resolver el problema del mapeo y localización simultanea mediante la implementación de una unidad móvil instrumentada, en este estudio se propone la implementación de un vehículo omnidireccional de tres ruedas equipado con una carama RGB-D que le permite obtener la percepción del ambiente en el que se encuentra, cabe mencionar que para este estudio se utilizan ROS en la parte de control he interfaz del robot. En [17] utilizan una pi cámara montada sobre un vehículo móvil para la reconstrucción tridimensional de un ambiente interno controlado. Para el sistema de adquisición, procesamiento y control del proceso hacen uso de ROS, debido al numero de ventajas que ofrece, en este estudio utilizan un algoritmo ya desarrollado para la construcción del mapa y la estimación de la pose del robot.

Como se puede apreciar en los estudios mencionados, el sensor más utilizado son los sensores ópticos, en este caso fueron las cámaras debido a que un solo cuadro tomado se puede obtener información de un área relativamente grande, sin embargo, un factor que predomina en los estudios presentados es el uso de vehículos con ruedas, lo que permite tener una superficie de sujeción para estos sensores. También cabe mencionar que el sistema operativo ROS es una herramienta común de encontrar debido a que ofrece diversas ventajas como algoritmos ya preestablecidos, que es de fuente libre, se puede programar mediante Python lo cual facilita algunos procesos gracias a las librerías, entre otras. Sin embargo, es uno de los objetivos de este estudio, el cual consiste en contribuir con la incorporación de los robots tipo esfera a tareas de exploración mediante el aprovechamiento de los sensores internos con los que cuenta este.

3. Modelo de movimiento

El modelo de movimiento para un robot móvil se puede describir como el modelo de transición de estados, establece que el robot asume la posición después de un comando de control [1]. Para un robot tipo esfera existe dos modelos que describen su movimiento [1][11]. Ambos modelos son algo complementarios en el tipo de información de movimiento que se procesa, el primer modelo llamado

modelo de movimiento de velocidad, asume que los datos de movimiento provienen de los comandos de velocidad (velocidad de rotación y traslación) y de la dirección del robot, el segundo es el modelo de movimiento de odometría, asume que la información de movimiento se proporciona a través de la odometría. Comúnmente los modelos de odometría se suelen aplicar para la estimación de la pose, mientras que los modelos de velocidad se utilizan para la planificación probabilística de movimientos [8]. Para este estudio se implementó el modelo de movimiento de odometría, la cual se obtiene a través de la integración de la aceleración y el giroscopio mediante la imu, este mecanismo de medición se le conoce como medición por estima. El modelo de odometría utiliza la información relativa de la odometría interna del robot, más específicamente, en el intervalo de tiempo (t_{-1}, t), el robot avanza de una pose x_{-1} a una pose x_t. La odometría nos informa un avance de la posición inicial como se muestra en la ecuación 1 a una posición posterior como describe ecuación 2. Para extraer la odometría relativa, se puede ver como una secuencia de tres pasos: rotación, seguida de un movimiento en línea recta (traslación) y otra rotación $\left(\delta_{rot1}\ \delta_{tras1}\ \delta_{rot2}\right)^{\mathrm{T}}$, para ejemplificar de una manera más grafica en la figura 1 se muestra un algoritmo, el cual acepta una entrada de posición inicial x_{t-1} y un control inicial el cual lo describe la ecuación 3, el algoritmo entrega como salida una probabilidad numérica de la pose actual del robot de la forma $p\left(x_t \mid u_t, x_{t-1}\right)$.

$$\overline{x}_{t-1} = \left(\overline{x}\ \overline{y}\ \overline{\theta}\right) \tag{1}$$

$$\overline{x}_{t-1} = \left(\overline{x}'\ \overline{y}'\ \overline{\theta}'\right) \tag{2}$$

$$u_t = \begin{pmatrix} \overline{x}_{t-1} \\ \overline{x}_t \end{pmatrix} \tag{3}$$

1: **Algorithm sample_motion_model_odometry**(u_t, x_{t-1})**:**

2: $\delta_{\mathrm{rot1}} = \mathrm{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$

3: $\delta_{\mathrm{trans}} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^s}$

4: $\delta_{\mathrm{rot2}} = \bar{\theta}' - \bar{\theta} - \delta_{\mathrm{rot1}}$

5: $\hat{\delta}_{\mathrm{rot1}} = \delta_{\mathrm{rot1}} - \mathbf{sample}(\alpha_1 \delta_{\mathrm{rot1}} + \alpha_2 \delta_{\mathrm{trans}})$

6: $\hat{\delta}_{\mathrm{trans}} = \delta_{\mathrm{trans}} - \mathbf{sample}(\alpha_3\ \delta_{\mathrm{trans}} + \alpha_4(\delta_{\mathrm{rot1}} + \delta_{\mathrm{rot2}}))$

7: $\hat{\delta}_{\mathrm{rot2}} = \delta_{\mathrm{rot2}} - \mathbf{sample}(\alpha_1 \delta_{\mathrm{rot2}} + \alpha_2 \delta_{\mathrm{trans}})$

8: $x' = x + \hat{\delta}_{\mathrm{trans}}\ \cos(\theta + \hat{\delta}_{\mathrm{rot1}})$

9: $y' = y + \hat{\delta}_{\mathrm{trans}}\ \sin(\theta + \hat{\delta}_{\mathrm{rot1}})$

10: $\theta' = \theta + \hat{\delta}_{\mathrm{rot1}} + \hat{\delta}_{\mathrm{rot2}}$

11: *return* $x_t = (x', y', \theta')^T$

Figura 1. Algoritmo para la estimación de la pose del robot a través de la implementación de un modelo de movimiento basado en la odometría.

En la figura 1 se muestra el algoritmo para la estimación de la pose del robot que utiliza un modelo de movimiento basado en la odometría, donde de las líneas 2 a la 4 recuperan los parámetros de

movimiento relativo, de la línea 5 a la línea 7 se obtienen las posicione x_t y x_{t-1}, por último, de la line 8 a la 10 se calculan las probabilidades de error para los parámetros de movimiento individuales, retornando la probabilidad de la pose actual del robot. En la figura 2 se ejemplifica este algoritmo mostrando los movimientos de rotación y traslación que se describen en el modelo de movimiento basado en la odometría (dos movimientos de rotación y uno de traslación).

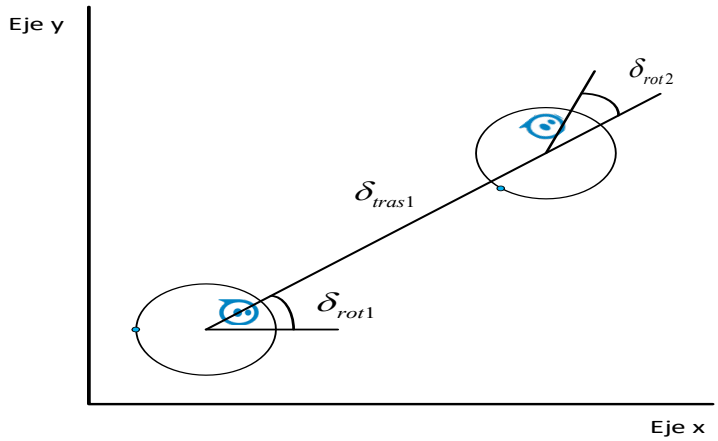


Figura 2. Ejemplo de los movimientos que componen el modelo de movimiento de odometría.

4. Filtro de Kalman

El filtro de Kalman se basa en una representación gaussiana de la incertidumbre [6] [7] [8]. Se usa ampliamente en el caso de que el sistema tenga perturbaciones aleatorias o exista ruido blanco en la fuente de medición, implementa el cálculo de creencias para estados continuos. No es aplicable a espacios de estados discretos o híbridos [12]. El filtro de Kalman representa las creencias mediante la representación de los momentos en el tiempo *t*, la creencia está representada por la media *μt* y la covarianza Σt. Las estimaciones del filtro serán gaussianas si se cumplen los siguientes tres supuestos.

- La siguiente probabilidad de estado $p(x_t \mid u_t, x_{t-1})$ $\mathrm{p(x_t|u_t, x_{t-1})}$ debe ser una función lineal en sus argumentos con ruido gaussiano agregado. Esto se expresa mediante la ecuación 4.

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \tag{4}$$

Donde x_t y x_{t-1} son vectores de estado, u_t es un vector de control, ε_t es el vector de transición de estados A_t es una matriz de transición B_t es una matriz de control.

- La probabilidad de medición $p(z_t \mid x_t)$ $\mathrm{p(z_t|x_t)}$ también debe ser lineal en sus argumentos, con ruido gaussiano añadido. Esto se expresa mediante la ecuación 5.

$$z_t = C_t x_t + \delta_t \tag{5}$$

Aquí z_t es el vector de medición, C_t es una matriz de tamaño k × n, donde k es la dimensión del vector de medición z_t. El vector δ_t describe el ruido de medición. La distribución de δ_t es una Gaussiana multivariante con media cero y covarianza Q_t.

- Finalmente, la creencia inicial bel(x0) debe tener una distribución normal. Se denota la media de esta creencia por μ0 y la covarianza por Σ0:

$$bel(x0) = p(x0) \tag{6}$$

El filtro de Kalman se puede resumir como una secuencia de dos pasos conformados por 5 ecuaciones (predicción del estado del robot y de la incertidumbre, ganancia de Kalman, actualización de la creencia del estado del robot y actualización de la incertidumbre), el primer paso se le denomina predicción, este estado consta en la integración de la medición de la odometría de un estado anterior y del modelo de movimiento del robot, para inferir sobre la nueva pose que asumirá el robot después de un control de movimiento, a la cual va asociada la predicción de la incertidumbre de ese estado futuro, el segundo paso se conoce como corrección o medición, como su nombre lo dice, corrige la creencia predicha en el paso anterior, esto lo hace mediante la diferencia de la creencia y las mediciones de odometría actual, dicha corrección servirá como un estado pasado para la siguiente predicción de pose, esto permite que el algoritmo entregue como salida una creencia de un estado óptimo con distribución gaussiana. En la figura 3 se muestra la interacción de estos dos pasos mediante un diagrama de bloques que expresa las 5 ecuaciones ya mencionadas y la figura 4 muestra la idea de estado óptimo.

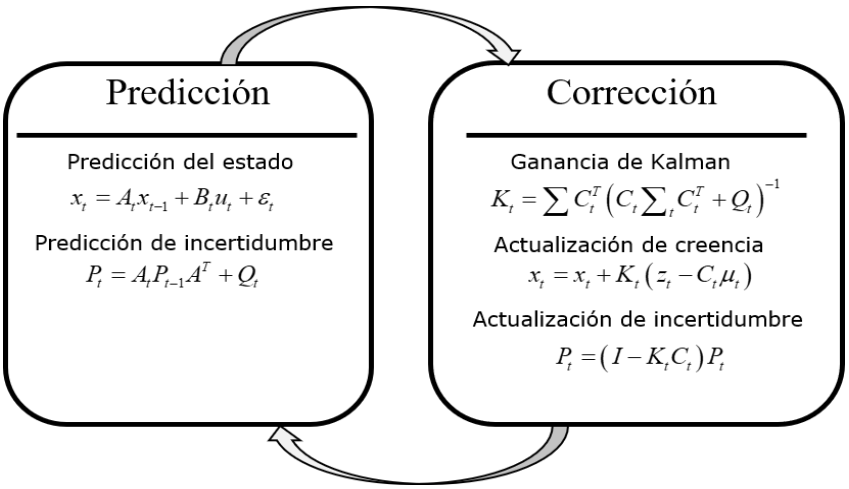


Figura 3. Diagrama de bloques que representa el funcionamiento del filtro de Kalman.

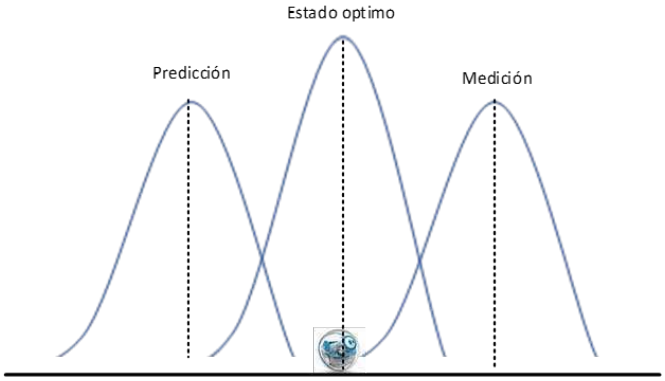


Figura 4. Distribución gaussiana del estado óptimo como consecuencia de la integración de la predicción y de la corrección.

5. Algoritmo de Tremaux

Charles Tremaux fue un ingeniero francés del siglo XIX. Él fue el que inventó este algoritmo para escapar de un laberinto. Actualmente es el más eficiente, y nos asegura la escapatoria de cualquier tipo de laberinto sin importar su forma [14], el algoritmo consiste en ir marcando el recorrido que se esta

haciendo, cuando se llega a un cruce (intersección de dos o más caminos) y este cruce es nuevo no importa que camino se toma, siempre que no se haya pasado antes por ahí. Si un camino viejo lleva a un cruce conocido o a un callejón sin salida se debe de retroceder hasta el cruce anterior, así sucesivamente, si se siguen estos pasos, aunque es posible que lleve horas, este sistema le llevará hasta la salida del laberinto, y si no la hay, de nuevo hasta la entrada, sin embargo, muchas de las veces que se trata de resolver un laberinto el tiempo es un recurso limitado, así que las decisiones de qué camino debe tomar el robot es de vital importancia. Existen técnicas para la toma de decisiones de esta índole, refiriéndose a la dirección que se debe tomar en un cruce, una técnica ampliamente utilizada es la regla de la mano derecha, la cual consiste que cada vez que se debe tomar la decisión de qué camino seguir se debe elegir siempre el lado derecho o bien el lado izquierdo, no importa hacia que lado girar siempre y cuando siempre sea el mismo lado. La regla de la mano derecha aunada al algoritmo de Tremaux se convierten en un algoritmo muy potente y simple que puede resolver cualquier tipo de laberinto no importa que complejidad o tamaño tenga este, un ejemplo de la potencia de la combinación de estos algoritmos se muestra a continuación en la figura 5 donde se resuelven dos laberintos diferentes empleando la misma técnica.

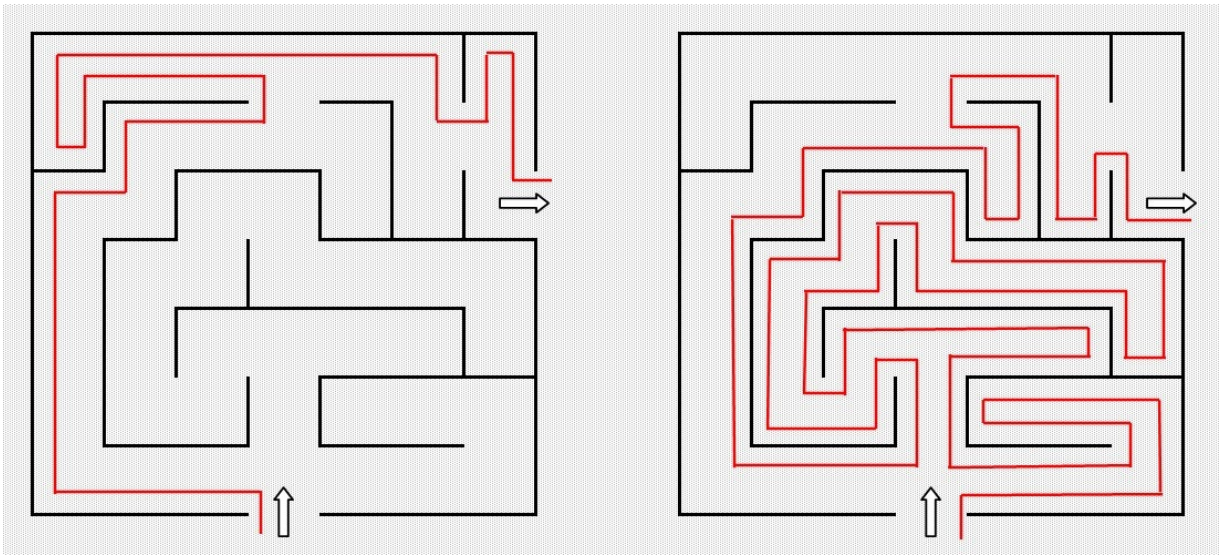


Figura 5. Dos laberintos diferentes resueltos bajo el mismo enfoque utilizando el algoritmo de Tremaux y la regla de la mano derecha [14].

6. Metodología

Para lograr un algoritmo basado en el aprovechamiento de los sensores internos de un robot tipo esfera permita resolver el problema del SLAM, se consideró el trabajo sinérgico entre el algoritmo de Tremaux y la regla de la mano derecha, el cual consiste en que el robot sigue únicamente las paredes del lado derecho hasta encontrar un cruce, es decir la unión de dos o más paredes, haciendo el seguimiento de la nueva pared, esto continua hasta que el robot encuentre una marca de referencia ya conocida, es decir que el robot completo el circuito. En la figura 6 se muestra un diagrama general de la metodología que se siguió. El diagrama que se muestra es la forma más simplificada de ver la solución de este problema, en este capítulo también se explica más detallado como es que el robot detecta las paredes y como es que detecta las esquinas. Cabe mencionar que la motivación de la implementación de este algoritmo se debe a que la experimentación de este estudio se realiza en un ambiente cerrado no cambiante, es decir que las pruebas siempre se realizan bajo las mismas circunstancias, un ejemplo de un ambiente de esta índole puede ser el corredor de una oficina, considerando que el robot no tiene interacción con las personas.

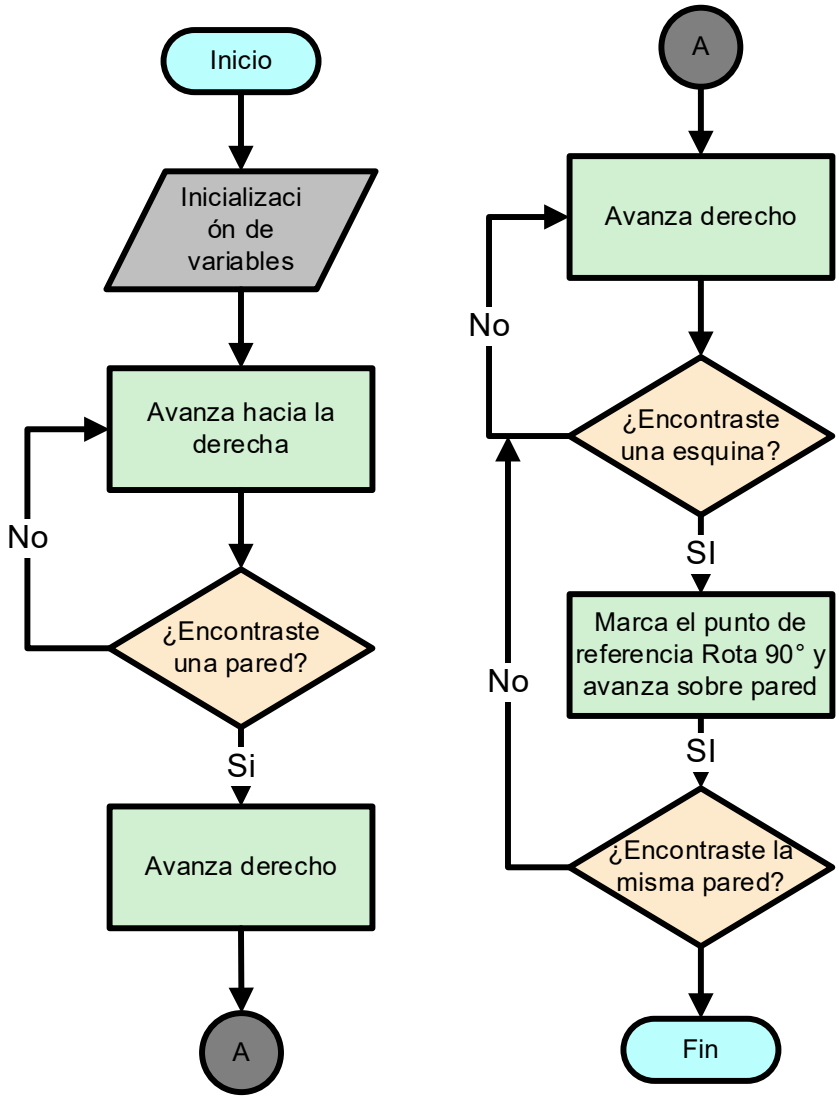
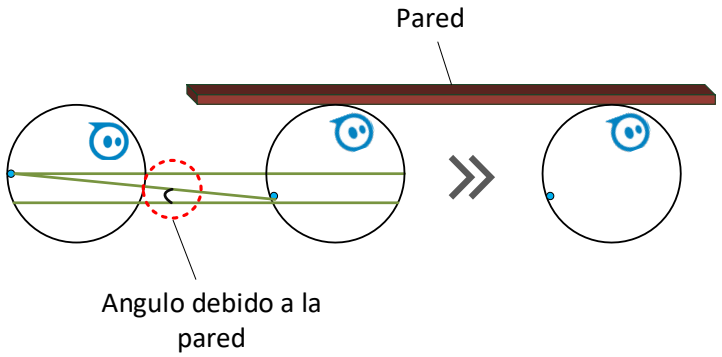


Figura 6. Diagrama de flujo de la metodología implementada en este estudio.

6.1 Seguimiento de paredes

El algoritmo para saber cuándo es que el robot está siguiendo una pared principalmente se basa en los ángulos de Euler (roll, pitch y yaw), ya que al hacer contacto con alguna pared el robot rota sobre el eje y de manera anormal, es decir, diferente a cuando se desplaza únicamente sobre una superficie plana y sin inclinación. Para ejemplificar mejor ver la figura 7, en esta figura se observa que el punto azul que representa el centro del robot al hacer contacto con la pared este cambia de posición y se forma un ángulo entre en centro del robot y el piso. Adicionando que el robot al tener fricción con la pared también varia su velocidad en su componente y, esta variación de la velocidad hace redundante la detección de paredes.

Figura 7. Representación del ángulo que se forma entre el Sphero$^{M.R}$ y el piso cuando se está siguiendo una pared.



6.2 Detección de esquinas

La detección de esquinas se ve reflejada en la disminución de velocidad en sus dos componentes x, y puesto cuando el robot se encuentra en una esquina cóncava, la velocidad de este disminuye drásticamente debido a que las dos paredes limitan los movimientos del robot, al trabajar con una velocidad constante se hace evidente estas variaciones de velocidad, para un mejor entendimiento ver Figura 7.

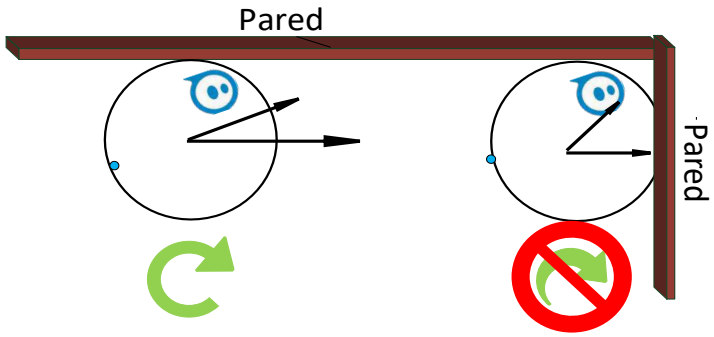


Figura 8. Representación gráfica de la influencia que tiene una esquina en la velocidad del robot.

7. Desarrollo

El desarrollo de este estudio se realizó mediante la utilización de un robot Sphero$^{M.R}$, el cual fue controlado mediante la implementación ROS, como se puede ver en la figura 9 se muestra un diagrama a bloques de la configuración que se utilizó para la conexión entre estas dos plataformas. La comunicación bidireccional entre ROS y el Sphero$^{M.R}$ se realizó por medio de Bluetooth a una frecuencia de 2.4 GHz, con un rango de distancia de operación de 15 metros según el fabricante. Para la lectura y el control del robot se utilizó el paquete Sphero_spkr_one, el cual es un paquete de fuente abierta programado en lenguaje Python, el paquete está diseñado para ser utilizado en Linux con una distribución Ubuntu, este paquete proporciona una serie de temas con los cuales se pude obtener la información de odometría, aceleración, y estado de la batería, también ofrece temas para el control de los motores y control de la iluminación del robot.

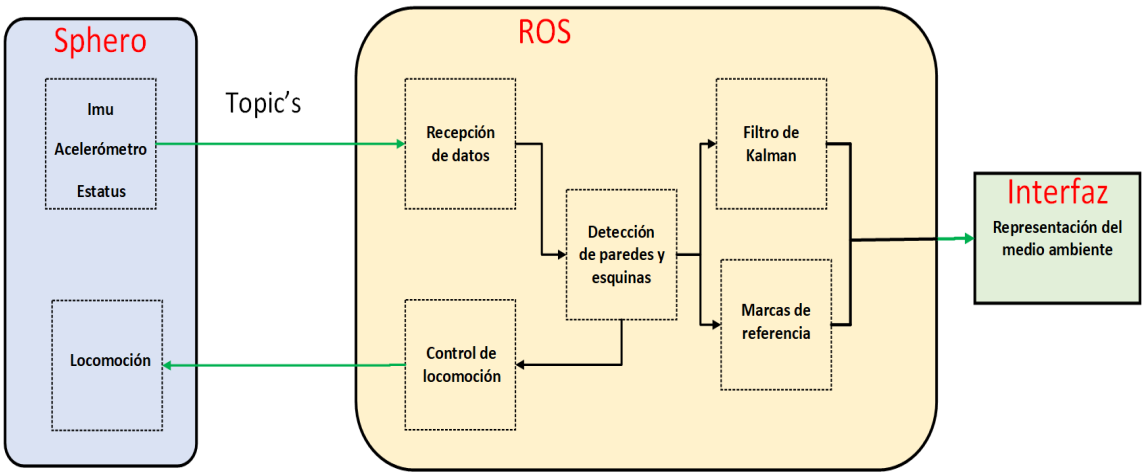


Figura 9. Diagrama de bloques de la conexión y estructuración entre Sphero$^{M.R}$ y ROS.

7.1 Nodo para la recepción de datos

Partiendo de que la comunicación con el robot es exitosa, se comenzó con el desarrollo de un nodo para la recepción de información, el cual se suscribe a los temas /Sphero_0/imu, /Sphero_0/odom y /Sphero_0/status, estos temas publican la información de odometría, imu y el estado de la batería del robot por medio de mensajes de tipo nav_msgs, sensor_msgs y array_msgs respectivamente. Se opta por desarrollar un nodo exclusivo para la recepción de información de manera que el sistema no pierda tiempo en procesamiento de la información, esto permite tener una mejor organización de la estructura de la programación, rastreabilidad de errores y optimización en el código. La recepción de la información trabaja a una frecuencia de 10 Hz que es la frecuencia a la que el Sphero$^{M.R}$ pública, es decir, en un segundo se obtienen 10 lecturas de las variables ya mencionadas (odometría, ángulos de rotación y estado de la batería). Como se puede ver en la figura 10 sección A,B el Sphero$^{M.R}$ publica una gran cantidad de información desordenada, que va desde el promedio y la covarianza de los valores de los sensores, o bien la odometría en el eje z el cual no se utiliza debido a la locomoción del robot, también agrega los ángulos de Euler de los tres ejes, si bien esta información puede ser valiosa en diversas aplicaciones, en este estudio no toda la información es relevante lo cual da pauta para discriminar esta información de manera que fuera conveniente para su interpretación, es decir solo se deja la aceleración en los ejes x,y y el Angulo de Euler del eje y. Una vez discriminada la información se procedió hacer una cadena parametrizada, de tamaño fijo y con un bit característico para el inicio de la trama y un bit para saber el final, de esta manera a la hora de procesar la información siempre se sabe la posición de cada dato, esto lo podemos ver en la figura 10 sección C. La trama parametrizada se publica en el tema \tema_1 de manera que cualquier otro nodo pueda leer lo que se está publicando.

7.2 Nodo para detección de paredes y esquina

La detección de paredes y esquinas es un algoritmo ya presentado en la sección 6 donde se explica el funcionamiento de los algoritmos para determinar cuándo es que se está siguiendo una pared y cuando es que se encontró una esquina, para la implementación de los algoritmos se desarrolló un nodo el cual se suscribe a al tema tema_1, en este nodo se publica una trama de datos que contiene la odometría dirección y cabeceo. Como se puede ver en la Figura 11 se publica la palabra pared cuando el robot se encuentra siguiendo una y publica esquina cuando el algoritmo determina que está en presencia de una, esto se hace con fines ilustrativos puesto lo que realmente importa es la odometría que corresponde al seguimiento de la pared y las coordenadas que corresponden a una esquina, se publican en un tema llamado \tema_2 el cual se utilizara para graficar el recorrido del robot.

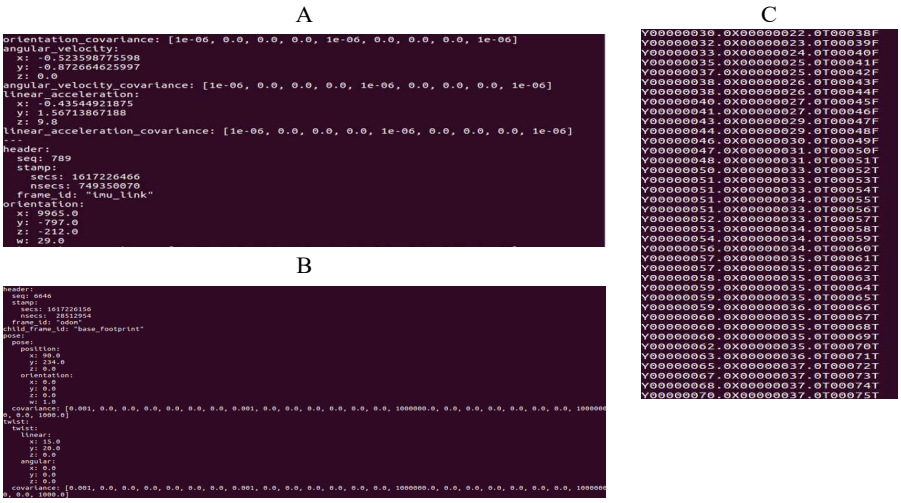


Figura 10. Comparación de la información recibida directamente del Sphero$^{M.R}$ (sección A, B) y la información parametrizada (sección C).

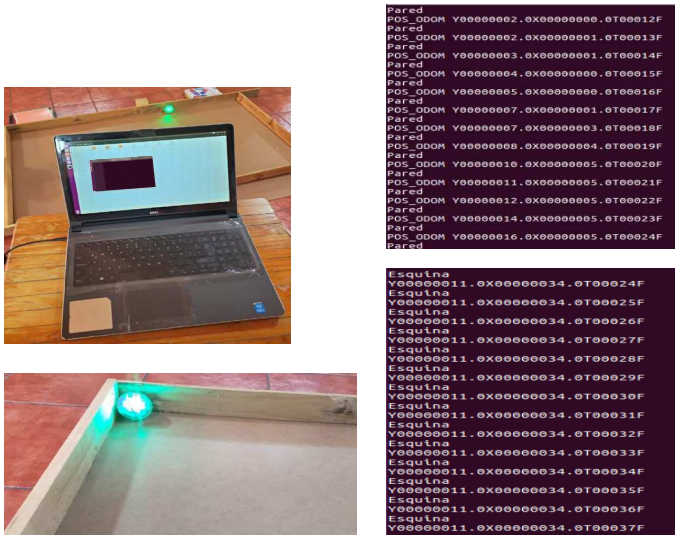


Figura 11. Prueba de detección de esquinas y paredes, al detectar alguna de estas se publicadas en el \tema_2.

7.3 Nodo para la estimación de pose

Para la implementación del Filtro de Kalman también se creó un nodo el cual se suscribió al tema \tema_2 donde se obtuvo la odometría del seguimiento de las paredes. Se siguió un algoritmo que consta de tres pasos que es la medición, actualización y predicción, este algoritmo da como resultado la pose estimada del robot, con una distribución del error gaussiana. Para ejemplificar mejor se puede ver la Figura 12, el cual acepta como entrada un vector de la forma que muestra la ecuación 7.

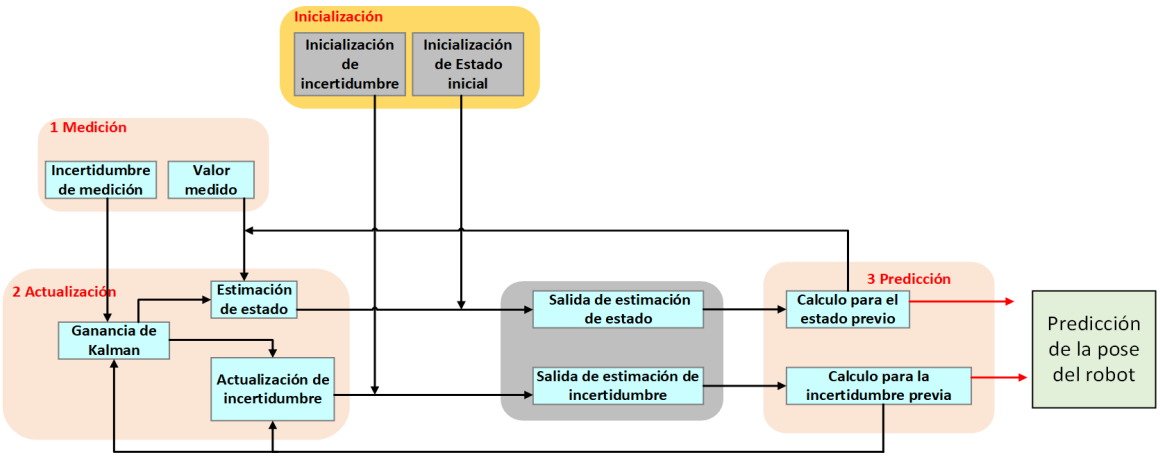


Figura 12. Diagrama de bloques que plantea la formulación del Filtro de Kalman.

$$\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = pose \tag{7}$$

8. Experimentación

Para probar el algoritmo desarrollado para resolver el problema del SLAM se utilizó un banco de pruebas conformado por paredes rectilíneas y una superficie plana sin inclinación, la experimentación consistió en colocar el robot en el ambiente, del cual el robot no tiene previo conocimiento, es decir el robot se encuentra en un ambiente desconocido y su misión consiste en poder hacer una reconstrucción de este, como resultado se espera un mapa construido por el robot, que no varié más del 10% de las medidas del entorno real, en la figura 13, se puede ver el ambiente en el cual se realizaron las pruebas. Las pruebas se hicieron con la configuración del robot que se puede ver en la tabla 1.

Tabla 1. Parámetros para la experimentación.

Parámetro	Valor/ porcentaje
Batería	100 %
Velocidad fija	50 cm/s
Frecuencia de muestreo	10 Hz
Método de detección	Variación de velocidad

La experimentación ya mencionada se realizo 15 veces de manera no continua con la finalidad de que los parámetros del robot siempre fueran los mismos y que no influenciaran en el resultado del algoritmo, en este estudio se presenta la reconstrucción que obtuvo el mejor resultado y un análisis del conjunto de las 15 pruebas realizadas.

9. Resultados

La implementación del algoritmo desarrollado permitió realizar la reconstrucción de un ambiente desconocido rectilíneo con una superficie plana y sin inclinación, como se puede ver en la figura 14, al

finalizar la ejecución del algoritmo se obtuvo un mapa muy cercano a al ambiente donde se encontró el robot.

Figura 13. Banco de pruebas para la reconstrucción de un ambiente rectilíneo.

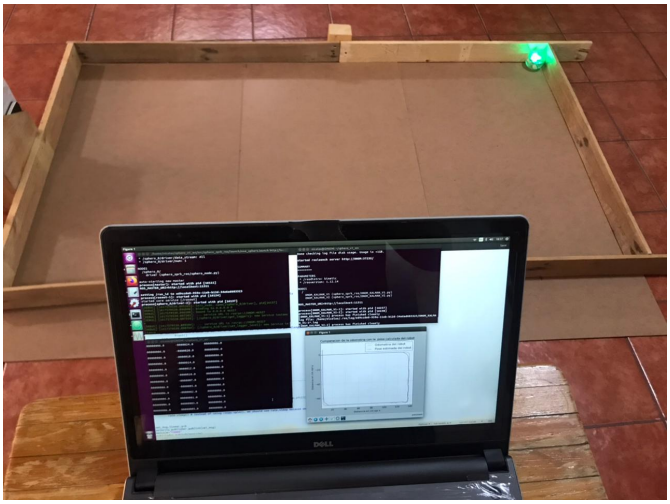


Figura 14. Implementación del algoritmo SLAM para la representación de un ambiente rectilíneo.

Como se puede observar en la figura 15, se representa la trayectoria del robot donde la línea roja es la trayectoria original del robot la cual está corrompida por el ruido y la línea azul son las predicciones de la pose del robot que se obtuvieron por medio del filtro de Kalman, como se puede observare cuando el robot avanza en línea recta la diferencia entre la estimación y la medición es despreciable sin embargo al comenzar y al finalizar las esquinas las diferencia entre las mediciones y las estimaciones se hacen más notorias.

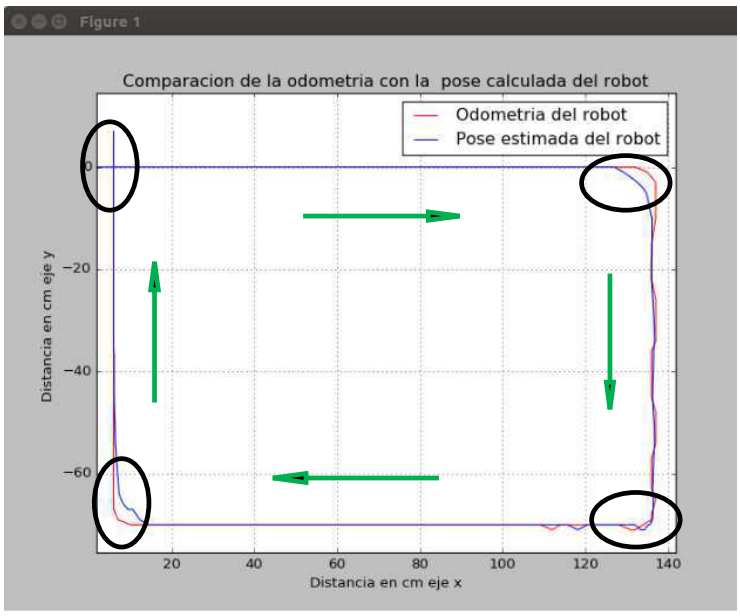


Figura 15. Comparación de la trayectoria del robot y la estimación de la pose del robot.

Para corroborar los resultados del estimador se agarraron las mediciones de la pared más larga, donde el robot recorre una distancia de 150 cm y se comparó la diferencia entre la distancia real y la estimada estos resultados se pueden apreciar en la tabla 2, donde se obtuvo una distribución normal, la cual se muestra en la figura 15.

Tabla 2. Comparación de las medidas reales con las estimadas.

Medida real	Estimación	Error
150 cm	150 cm	0 cm
150 cm	150 cm	0 cm
150 cm	147 cm	3 cm
150 cm	152 cm	-2 cm
150 cm	147 cm	3 cm
150 cm	152 cm	-2 cm
150 cm	148 cm	2 cm
150 cm	147 cm	3 cm
150 cm	146 cm	4 cm
150 cm	152 cm	-2 cm
150 cm	150 cm	0 cm
150 cm	150 cm	0 cm
150 cm	153 cm	-3 cm
150 cm	154 cm	-4 cm
150 cm	148 cm	2 cm

Como se puede apreciar en la Figura 16 la diferencia máxima entre el valor real y la estimación no supera los 4 cm en una distancia de 150 cm por lo cual quiere decir que estamos muy por debajo del 10% de error que se tenía como meta, estos resultados son prometedores para trabajos futuros, además al ver que el error se comporta de manera normal da apertura para continuar con este tema de estudio.

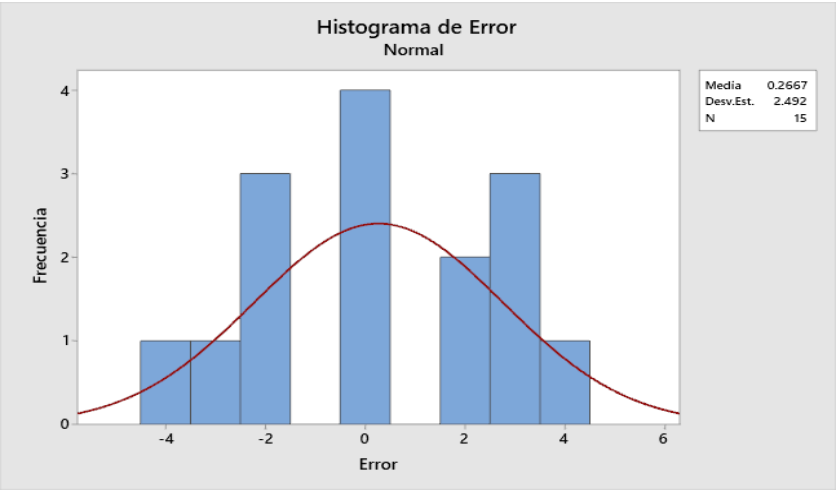


Figura 16. Distribución del error del filtro de Kalman.

Analizando la tabla 2 dejando de lado el porcentaje de margen en el error, se aprecia que realmente el porcentaje de aciertos es muy bajo puesto corresponde a un 14%. Sin embargo, al considerar que el margen de error 10% de la media real podemos notar que el 100% de las pruebas están dentro de este margen, para ejemplificar mejor se puede ver la Figura 17, donde de manera gráfica se ve que la diferencia de las estimaciones está muy por debajo del margen de error.

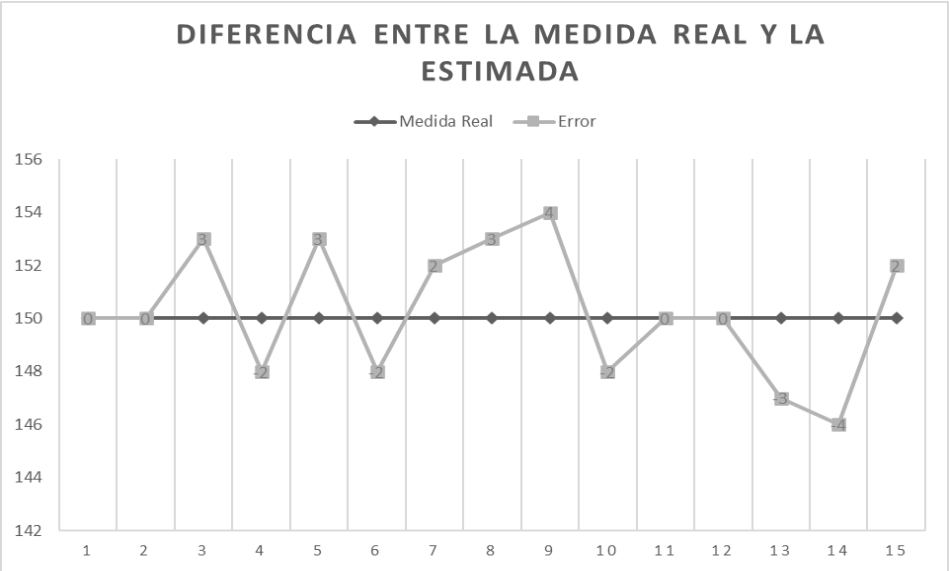


Figura 17. Representación gráfica de la diferencia entre la medida real y las estimaciones.

10. Conclusiones

Actualmente existe una gran cantidad de sensores que son relativamente de bajo costo, los cuales simplifican el trabajo de representación de ambientes, para lograr la solución del Mapeo y Localización Simultanea (SLAM). Algunos de estos son sensores ópticos como cámaras, estero cámaras, o bien sensores reflectivos y ópticos reflectivos como los LIDAR, sensores de rango, entre otros. El tipo de sensores ya mencionados con una sola toma pueden obtener información de las características de un área mayor que si se hiciera mediante el impacto del robot, en primera instancia

pueden parecer la mejor opción para tareas de exploración, sin embargo, al considerar que la sujeción de estos sensores en la morfología como lo es un robot esférico no es tan trivial, puesto un robot esférico avanza mediante la rotación de su carcasa, lo que genera no tener un punto fijo como lo podría ser un vehículo de 4 ruedas, esto hace que este tipo de robots requieran soluciones ingeniosas, como lo es el aprovechamiento de los sensores internos.

Sin embargo, a pesar de los resultados positivos que se obtuvieron en este estudio, una desventaja es que es un método demasiado lento en comparación de otros estudios como los que se presentan en [15][16][17], sin embargo, lo compensan con el bajo costo, energético y de fabricación lo que permite la implementación de técnicas como robótica enjambre. En conclusión, los resultados que se obtuvieron cumplieron el objetivo de demostrar que robots con una morfología tan favorable como lo es la esférica pueden realizar tara de inspección únicamente utilizando los sensores internos, dando pauta para fututas investigaciones.

11. Trabajos futuros

El presente estudio representa la base para trabajos posteriores, puesto la demanda de robots autónomos con alta flexibilidad y maniobrabilidad es latente, este trabajo está diseñado para servir como base para la inspección de tuberías, ya que es una aplicación que actualmente se realiza por robots con diversas morfologías propensas a atascarse en las esquinas o bridas que conforma la red de tuberías, se tiene la hipótesis de que estos problemas se pueden solucionar si la morfología del robot es esférica.

En la literatura se encuentran estudios donde un conjunto de pequeños robots ataca un problema en común de manera que entre ellos comparten información para tener una solución más certera, un problema que se puede encontrar comúnmente es el SLAM, esto nos ha servido de inspiración para realizar el mismo trabajo de reconstrucción, pero implementado varios robots Sphero$^{M.R}$ utilizando técnicas de robótica enjambre y robótica evolutiva.

Referencias

[1] Kandasamy, S. “SLAMming with Spheros: An impact-based approach to Simultaneous Localization and Mapping.” (2015
[2] P. Agarwal, W. Burgard and C. Stachniss, "Survey of Geodetic Mapping Methods: Geodetic Approaches to Mapping and the Relationship to Graph-Based SLAM," in IEEE Robotics & Automation Magazine, vol. 21, no. 3, pp. 63-80, Sept. 2014, doi: 10.1109/MRA.2014.2322282.
[3] REYES, Fernando. “Robótica-control de robots manipuladores”. Alfaomega grupo editor, 2011.
[4] Santos, MC Salvador Ortiz. “Localización y mapeo simultáneo robusto aplicado a la planeación de trayectorias en robot móviles”. Diss. Instituto politécnico nacional, 2019.
[5] ZANNATHA, Juan M. IBARRA, et al. "Desarrollo de un Sistema SLAM VISUAL con reconstrucción 3D monocular de marcas orientadas para un Humanoide.",2009
[6] GREWAL, Mohinder S.; ANDREWS, Angus P. Kalman filtering: Theory and Practice with MATLAB. John Wiley & Sons, 2014.
[7] BARRETO, Armando, et al. Intuitive Understanding of Kalman Filtering with MATLAB®. CRC Press, 2020.
[8] Musoff, Howard; ZARCHAN, Paul. Fundamentals of Kalman filtering: a practical approach. American Institute of Aeronautics and Astronautics, 2009.
[9] Martínez, JOSE YAVHE CASTILLO, et al. Representación Espacial de Ambientes Desconocidos para Robots Móviles con Redes Bayesianas. 2019.
[10] Jefferies, Margaret E.; YEAP, Wai-Kiang (ed.). Robotics and cognitive approaches to spatial mapping. Berlin Heidelberg: Springer, 2008.
[11] Hu, Xiaoxuan. SLAM with SPHEROS. MS thesis. Universitat Politècnica de Catalunya, 2018.
[12] Munuera Raga, Maria Cristina. Filtro de Kalman y sus aplicaciones. 2018.
[13] Bienias, Łukasz, Krzysztof Szczepański, and Piotr Duch. "Maze exploration algorithm for small mobile platforms." IPC. Vol. 21. No. 3. 2016.

[14] Díaz, Andrés Delgado; Esparza, Carlos Humberto. Robot Seguidor De Línea, Modo Solucionador Laberinto. En Ingeniería Electrónica, 2014.
[15] Suárez-Sánchez, Andrés Felipe; Loaiza-Correa, Humberto. Esquema de navegación reactiva con sensores RGB-D. Revista UIS Ingenierías, 2015, vol. 14, no 1, p. 7-19.
[16] LONGINOS GARRIDO, Mauricio, et al. Sistema de localización y mapeo simultáneo para interiores basado en imágenes de profundidad (RGB-D). 2021. Tesis de Maestría.
[17] Obregón, Luz Karime Garzón, Luis Albero Forero Rincón, and Oscar Manuel Duque Suárez. "Diseño e implementación de un sistema de visión artificial usando una técnica de mapeo y localización simultánea (SLAM) sobre una plataforma robótica móvil." Mundo Fesc 8.16 (2018): 8-17.

Diseño Colaborativo en Mecatrónica, Capítulo 6, pp. 68 - 87
ISBN: 978-607-9394-23-3, 2021.

Implementación de la cinemática inversa del robot Bioloid Premium

Ortega Palacios Miguel Angel[1], Palomino Merino Amparo Dora[2], Reyes Cortés José Fernando[2].

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación[1], Facultad de Ciencias de la Electrónica[2]
miguel.ortegap@alumno.buap.mx

Resumen

La implementación de marcha y el control de posición de robots bípedos continúa siendo un desafío para la comunidad científica, debido a que es un problema de gran complejidad, la cual se incrementa si el robot cuenta con un alto número de grados de libertad, El presente trabajo toma como propuesta de investigación el robot Bioloid Premium de 18 grados de libertad. El objetivo es proponer una solución al modelo de su cinemática inversa e implementarlo en el robot para validar los resultados a través de diferentes pruebas experimentales, llevando las articulaciones del robot a diferentes posiciones propuestas. Este modelo servirá como punto de partida para obtener su modelo dinámico en un trabajo posterior, de esta manera se aspira a que la investigación aquí desarrollada pueda ser utilizada para el desarrollo de nuevos métodos de solución de la cinemática inversa en otros robots bípedos.

Palabras clave: Bioloid Premium, robots bípedos, cinemática inversa.

1. Introducción

En la actualidad existen diversos trabajos que emplean a los robots bípedos como objetivo de estudio, como el robot Nao [1], [2], Lola [3], [4] y HRP [5], [6], el objetivo principal de su estudio es la implementación de modelos de marcha controlados. Debido al alto número de grados de libertad de estos robots, se opta por emplear diferentes métodos para controlar sus articulaciones, tales como, redes neuronales [7], algoritmos genéticos [8] y aprendizaje por refuerzo [9]. El robot Bioloid también ha sido usado por la comunidad científica en investigaciones para implementar estrategias de control de marcha bípeda, pero en general, la mayoría los trabajos modelan solo la cinemática de las piernas [10], [11], [12], [13], sin tomar en cuenta el movimiento de los brazos. Debido a estas limitaciones se optó por encontrar la solución de la cinemática inversa y directa del robot completo, utilizando la convención de Denavit-Hartenberg y el método de desacoplamiento cinemático. Esta metodología servirá como punto de partida para desarrollar el modelo dinámico del robot e implementar control de posición en sus articulaciones.

El resto de este documento se estructura de la siguiente forma: en la sección dos se presenta el modelo de cinemática directa, en la tercera sección el modelo de cinemática inversa, la cuarta sección contiene los resultados obtenidos en simulación, empleando el software Matlab, y finalmente en la quinta sección las conclusiones.

2. Cinemática inversa de las piernas

El Kit Bioloid Premium es un robot móvil tipo humanoide de 18 grados de libertad (Figura 1), 6 en las piernas y 3 en cada brazo, emplea una batería tipo LiPo, controlador CM-530, y servomotores Dynamixel AX-12A. La altura del Bioloid es de 39,7 cm. y su peso es de 1,7 Kg; todos sus enlaces son de plástico. Debido a su diseño este robot tiene la capacidad de tomar diferentes formas, donde el tipo hexápodo y humanoide son las que se emplean con mayor frecuencia [14].

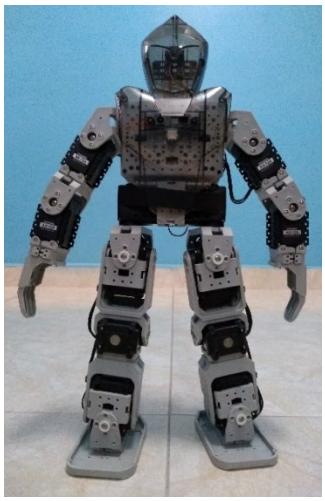


Figura 1. Robot Bioloid Premium.

Empleando el modelo físico del robot Bioloid Premium se propusieron cuatro cadenas cinemáticas abiertas (Figura 10) que corresponden a las piernas y brazos del robot. Para la descripción de la cadena cinemática de las piernas se asignaron sistemas de referencia a cada una de las articulaciones aplicando la convención de Denavit-Hartenberg [15].

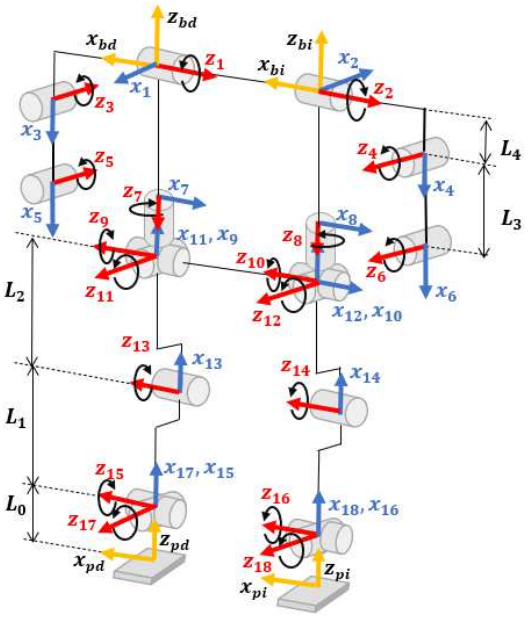


Figura 2. Cadenas cinemáticas asignadas al robot Bioloid Premium.

En la Tabla 1 se proporcionan los parámetros de Denavit-Hartenberg para la cadena cinemática que corresponde a la pierna derecha del robot.

Tabla 1. Parámetros de Denavit-Hartenberg para la cadena cinemática de las piernas del robot.

Servomotor	α_i	l_i	θ_i	d_i
17	90°	L_0	θ_{17}	0
15	-90°	0	θ_{15}	0
13	0°	L_1	θ_{13}	0
11	0°	L_2	θ_{11}	0
9	90°	0	θ_9	0
7	90°	0	$\theta_7 - 45°$	0

Para definir el valor de las variables, se emplearon las medidas reales del robot: $L_0 = 10cm$, $L_1 = L_2 = 7.5\ cm$, además se emplea la notación $sen(\theta_i) = S_i$, $cos(\theta_i) = C_i$ donde i denota el número del servomotor del robot.

En la representación de Denavit-Hartenberg, la matriz de transformación homogénea es una matriz que transforma un vector de posición expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas [6]. La matriz de transformación está definida en la ecuación (1).

$$H_{i-1}^{i} = \begin{pmatrix} cos\ (\theta_i) & -sen(\theta_i)cos\ (\alpha_i) & sen(\theta_i)sen(\alpha_i) & l_i cos(\theta_i) \\ sen(\theta_i) & cos\ (\theta_i)cos\ (\alpha_i) & -cos\ (\theta_i)sen(\alpha_i) & l_i sen(\theta_i) \\ 0 & sen(\alpha_i) & cos\ (\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

Donde el superíndice i representa el número de la articulación actual y el subíndice $i-1$ indica el número de la articulación anterior. Por lo tanto, H_{i-1}^{i} es la matriz de transformación homogénea que representa la rotación y traslación de la articulación i con respecto a la articulación $i-1$.

Utilizando la ecuación (1) y los parámetros de la Tabla 1 se obtienen las siguientes matrices de transformación para las 6 articulaciones de la pierna derecha:

$$H_0^1 = \begin{pmatrix} C_{17} & 0 & S_{17} & L_0C_{17} \\ S_{17} & 0 & -C_{17} & L_0S_{17} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad H_1^2 = \begin{pmatrix} C_{15} & 0 & -S_{15} & 0 \\ S_{15} & 0 & C_{15} & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad H_2^3 = \begin{pmatrix} C_{13} & -S_{13} & 0 & L_1C_{13} \\ S_{13} & C_{13} & 0 & L_1S_{13} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_3^4 = \begin{pmatrix} C_{11} & -S_{11} & 0 & L_2C_{11} \\ S_{11} & C_{11} & 0 & L_2S_{11} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad H_4^5 = \begin{pmatrix} C_9 & 0 & S_9 & 0 \\ S_9 & 0 & -C_9 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad H_5^6 = \begin{pmatrix} C_7 & 0 & S_7 & 0 \\ S_7 & 0 & -C_7 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La matriz de transformación homogénea H_0^6 que corresponde a la pierna derecha es igual al producto de seis matrices H_{i-1}^{i}. Entonces, al multiplicar sucesivamente la ecuación H_0^6 por la matriz inversa de H_{i-1}^{i}, se pueden obtener seis ecuaciones matriciales:

$$H_0^6 = H_0^1 H_1^2 H_2^3 H_3^4 H_4^5 H_5^6 \tag{2}$$

$$(H_0^1)^{-1} H_0^6 = H_1^2 H_2^3 H_3^4 H_4^5 H_5^6 \tag{3}$$

$$(H_1^2)^{-1} (H_0^1)^{-1} H_0^6 = H_2^3 H_3^4 H_4^5 H_5^6 \tag{4}$$

$$(H_2^3)^{-1} (H_1^2)^{-1} (H_0^1)^{-1} H_0^6 = H_3^4 H_4^5 H_5^6 \tag{5}$$

$$(H_3^4)^{-1}(H_2^3)^{-1}(H_1^2)^{-1}(H_0^1)^{-1}H_0^6 = H_4^5H_5^6 \tag{6}$$

$$(H_4^5)^{-1}(H_3^4)^{-1}(H_2^3)^{-1}(H_1^2)^{-1}(H_0^1)^{-1}H_0^6 = H_5^6 \tag{7}$$

Por lo tanto, la matriz de H_{i-1}^i se se define en (8)

$$H_{i-1}^i = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

Donde se puede definir la matriz *noa* como sigue:

$$noa = \begin{pmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{pmatrix}$$

Por lo tanto:

$$\left(H_{i-1}^i\right)^{-1} = \begin{pmatrix} n_x & n_y & n_z & -(n_xp_x + n_yp_y + n_zp_z) \\ o_x & o_y & o_z & -(o_xp_x + o_yp_y + o_zp_z) \\ a_x & a_y & a_z & -(a_xp_x + a_yp_y + a_zp_z) \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{9}$$

Para simplificar el modelo de cinemática inversa de las piernas del robot se empleó el método de desacoplamiento cinemático, presentados en [17], [18] y [19], el cual consiste en la separación de orientación y posición en robots de 6 grados de libertad. Los robots suelen contar con tres grados de libertad adicionales, situados al final de la cadena cinemática y cuyos ejes, generalmente, se cortan en un punto que informalmente se denomina muñeca del robot. Así, dada una posición y orientación final deseada se establece la posición del punto de corte (muñeca del robot) calculando los valores de $\theta_{13}, \theta_{15}\ y\ \theta_{17}$, y a continuación a partir de los datos de orientación y los ya calculados se obtienen los valores del resto de las variables articulares $\theta_7, \theta_9\ y\ \theta_{11}$.

Es posible calcular los ángulos de las tres primeras articulaciones de la pierna tomando en cuenta las matrices H_0^1, H_1^2, H_2^3 y la matriz H_0^3, la cual corresponde a la multiplicación de las 3 anteriores, tal como se muestra en la ecuación matricial (10):

$$(H_0^1)^{-1}H_0^3 = H_1^2H_2^3 \tag{10}$$

$$\begin{pmatrix} C_{17} & S_{17} & 0 & -L_0 \\ 0 & 0 & 1 & 0 \\ S_{17} & -C_{17} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_{15} & 0 & -S_{15} & 0 \\ S_{15} & 0 & C_{15} & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} C_{13} & -S_{13} & 0 & L_1C_{13} \\ S_{13} & C_{13} & 0 & L_1S_{13} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} n_xC_{17} + n_yS_{17} & o_xC_{17} + o_yS_{17} & a_xC_{17} + a_yS_{17} & p_xC_{17} + p_yS_{17} - L_0 \\ n_z & o_z & a_z & p_z \\ n_xS_{17} - n_yC_{17} & o_xS_{17} - o_yC_{17} & a_xS_{17} - a_yC_{17} & p_xS_{17} - p_yC_{17} \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} C_{13}C_{15} & -C_{15}S_{13} & -S_{15} & L_1C_{13}C_{15} \\ C_{13}S_{15} & -S_{13}S_{15} & C_{15} & L_1C_{13}S_{15} \\ -S_{13} & -C_{13} & 0 & -L_1S_{13} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{11}$$

Analizando la ecuación (11) se puede igualar los 16 términos que contiene una matriz, es decir, se pueden plantear 16 ecuaciones y elegir la que más sea amigable para despejar la variable articular. Por lo tanto, a partir de (11) se pueden calcular los ángulos $\theta_{13}, \theta_{15}, \theta_{17}$.

Primero se calcula θ_{13} empleando el término (3,4) de ambos lados de la ecuación:

$$-L_1 sen(\theta_{13}) = P_x sen(\theta_{17}) - P_y cos(\theta_{17})$$

$$\theta_{13} = arcsen\left(\frac{P_y cos(\theta_{17}) - P_x sen(\theta_{17})}{L_1}\right) \tag{12}$$

Después se calcula θ_{15} empleando el cociente de los elementos (2,4) de ambos lados de la ecuación:

$$P_z = L_1 cos(\theta_{13}) sen(\theta_{15})$$

$$\theta_{15} = arcsen\left(\frac{P_z}{L_1 cos(\theta_{13})}\right) \tag{13}$$

Posteriormente se calcula θ_{17} empleando el término (3,3) de ambos lados de la ecuación, como sigue:

$$a_x sen(\theta_{17}) - a_y cos(\theta_{17}) = 0$$

$$\theta_{17} = arctan\left(\frac{a_y}{a_x}\right) \tag{14}$$

La función *arctan(.)* devuelve valores entre $-\pi/2$ y $\pi/2$, y el rango de movimiento de algunas articulaciones estaría limitado, por lo cual, para propósitos de programación, se empleó la función *atan2(.)* para tener un mayor rango de valores. Esta función calcula el arco tangente de y/x, pero los signos de ambos argumentos se utilizan para determinar el cuadrante del resultado, y se permite que x sea cero. El resultado se da en radianes y está en el rango de $-\pi$ a π. Incluso, si tanto *x* como *y* son cero*, atan2(.)* devuelve cero. Por lo tanto:

$$\theta_{17} = atan2(a_y, a_x) \tag{15}$$

$$\theta_{15} = atan2\left(P_z, P_x cos(\theta_{17}) + P_y sen(\theta_{17})\right) \tag{16}$$

El siguiente paso es encontrar las variables articulares θ_7, θ_9 y θ_{11}, empleando la ecuación matricial (18), en la que no es necesario utilizar las matrices de transformación homogénea debido a que no se tiene traslaciones, tan solo se tiene rotaciones, por esta razón se puede utilizar sólo las submatrices de rotación.

La matriz de rotación de 0 a 6 se puede escribir de forma genérica a través de la matriz *noa* que no es más que la matriz de rotación total que se ha realizado con el último sistema de coordenadas que corresponde a la cadera sobre el eje transversal.

Empleando los parámetros las matrices de transformación H_3^4, H_4^5, H_5^6 se puede definir la matriz de rotación de 3 a 6 como se observa en (17):

$$R_3^6 = R_3^4 R_4^5 R_5^6 \tag{17}$$

Donde:

$$R_3^4 = \begin{pmatrix} C_{11} & -S_{11} & 0 \\ S_{11} & C_{11} & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad R_4^5 = \begin{pmatrix} C_9 & 0 & S_9 \\ S_9 & 0 & -C_9 \\ 0 & 1 & 0 \end{pmatrix} \qquad R_5^6 = \begin{pmatrix} C_7 & 0 & S_7 \\ S_7 & 0 & -C_7 \\ 0 & 1 & 0 \end{pmatrix}$$

Por lo tanto:

$$R_3^6 = \begin{pmatrix} C_7(C_9C_{11} - S_9S_{11}) & C_9S_{11} + C_{11}S_9 & S_7(C_9C_{11} - S_9S_{11}) \\ C_7(C_9S_{11} + C_{11}S_9) & S_9S_{11} - C_9C_{11} & S_7(C_9S_{11} + C_{11}S_9) \\ S_7 & 0 & -C_7 \end{pmatrix} =$$

$$\begin{pmatrix} C_{7,9,11} - C_7S_{9,11} & C_9S_{11} + C_{11}S_9 & S_7C_{9,11} - S_{7,9,11} \\ C_{7,9}S_{11} + C_{7,11}S_9 & S_{9,11} - C_{9,11} & S_{7,11}C_9 + S_{7,9}C_{11} \\ S_7 & 0 & -C_7 \end{pmatrix} \tag{18}$$

La matriz de rotación de 0 a 3 se encuentra con los parámetros de las matrices de transformación H_0^1, H_1^2, H_2^3.

$$R_0^3 = R_0^1 R_1^2 R_2^3$$

$$R_0^1 = \begin{pmatrix} C_{17} & 0 & S_{17} \\ S_{17} & 0 & -C_{17} \\ 0 & 1 & 0 \end{pmatrix} \qquad R_1^2 = \begin{pmatrix} C_{15} & 0 & -S_{15} \\ S_{15} & 0 & C_{15} \\ 0 & -1 & 0 \end{pmatrix} \qquad R_2^3 = \begin{pmatrix} C_{13} & -S_{13} & 0 \\ S_{13} & C_{13} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Por lo tanto:

$$R_0^3 = \begin{pmatrix} C_{13}C_{15}C_{17} - C_{13}S_{17} & -C_{13}S_{17} - C_{15}C_{17}S_{13} & -C_{17}S_{15} \\ C_{17}S_{13} + C_{13}C_{15}S_{17} & C_{13}C_{17} - C_{15}S_{13}S_{17} & -S_{15}S_{17} \\ C_{13}S_{15} & -S_{13}S_{15} & C_{15} \end{pmatrix}$$

$$(R_0^3)^{-1} = (R_0^3)^T = \begin{pmatrix} C_{13}C_{15}C_{17} - C_{13}S_{17} & C_{17}S_{13} + C_{13}C_{15}S_{17} & C_{13}S_{15} \\ -C_{13}S_{17} - C_{15}C_{17}S_{13} & C_{13}C_{17} - C_{15}S_{13}S_{17} & -S_{13}S_{15} \\ -C_{17}S_{15} & -S_{15}S_{17} & C_{15} \end{pmatrix} =$$

$$\begin{pmatrix} C_{13,15,17} - C_{13}S_{17} & C_{17}S_{13} + C_{13,15}S_{17} & C_{13}S_{15} \\ -C_{13}S_{17} - C_{15,17}S_{13} & C_{13,17} - C_{15}S_{13,17} & -S_{13,15} \\ -C_{17}S_{15} & -S_{15,17} & C_{15} \end{pmatrix} \tag{19}$$

Reemplazando las ecuaciones (18) y (19) y la matriz *noa* en la ecuación (20) se tiene:

$$R_3^6 = (R_0^3)^T R_0^6 \tag{20}$$

$$\begin{pmatrix} C_{7,9,11} - C_7S_{9,11} & C_9S_{11} + C_{11}S_9 & S_7C_{9,11} - S_{7,9,11} \\ C_{7,9}S_{11} + C_{7,11}S_9 & S_{9,11} - C_{9,11} & S_{7,11}C_9 + S_{7,9}C_{11} \\ S_7 & 0 & -C_7 \end{pmatrix} =$$

$$\begin{pmatrix} C_{13,15,17} - C_{13}S_{17} & C_{17}S_{13} + C_{13,15}S_{17} & C_{13}S_{15} \\ -C_{13}S_{17} - C_{15,17}S_{13} & C_{13,17} - C_{15}S_{13,17} & -S_{13,15} \\ -C_{17}S_{15} & -S_{15,17} & C_{15} \end{pmatrix} \begin{pmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{pmatrix} = \tag{21}$$

A partir de la ecuación (21) se eligen los términos que generan una ecuación amigable para despejar las variables articulares θ_7, θ_9 y θ_{11}. Primero se calcula θ_7 empleando el término (3,3) de ambos lados de la ecuación (21), como sigue:

$$-\cos(\theta_7) = -a_x cos(\theta_{17}) sen(\theta_{15}) - a_y sen(\theta_{15} + \theta_{17}) + a_z cos(\theta_{15})$$

$$\theta_7 = arccos\left(a_x cos(\theta_{17}) sen(\theta_{15}) + a_y sen(\theta_{15} + \theta_{17}) - a_z cos(\theta_{15})\right) \tag{22}$$

Luego se calcula θ_9 empleando el término (2,2) de la siguiente forma:

$$\begin{aligned} sen(\theta_9)\mathrm{sen}(\theta_{11}) - cos(\theta_9)\cos(\theta_{11}) \\ = -o_x\big(cos(\theta_{13}) sen(\theta_{17}) + cos(\theta_{15} + \theta_{17}) sen(\theta_{13})\big) \\ + o_y\big(cos(\theta_{13} + \theta_{17}) - sen(\theta_{13} + \theta_{17}) cos(\theta_{15})\big) - o_z sen(\theta_{13} + \theta_{15}) \end{aligned}$$

$$\begin{aligned} \theta_9 = arccos\big[o_x\big(cos(\theta_{13}) sen(\theta_{17}) + cos(\theta_{15} + \theta_{17}) sen(\theta_{13})\big) \\ - o_y\big(cos(\theta_{13} + \theta_{17}) - sen(\theta_{13} + \theta_{17}) cos(\theta_{15})\big) + o_z sen(\theta_{13} + \theta_{15})\big] - \theta_{11} \end{aligned} \tag{23}$$

Posteriormente se calcula θ_{11} empleando el término (1,2), tal como sigue:

$$\begin{aligned} cos(\theta_9)\mathrm{sen}(\theta_{11}) + cos(\theta_{11})\mathrm{sen}(\theta_9) \\ = o_x\big(cos(\theta_{13} + \theta_{15} + \theta_{17}) - \cos(\theta_{13}) sen(\theta_{17})\big) \\ + o_y\big(cos(\theta_{17}) sen(\theta_{13}) + cos(\theta_{13} + \theta_{15}) sen(\theta_{17})\big) + o_z cos(\theta_{13}) sen(\theta_{15}) \end{aligned}$$

$$\begin{aligned} \theta_{11} = arcsen\big[o_x\big(cos(\theta_{13} + \theta_{15} + \theta_{17}) - sen(\theta_{13} + \theta_{17})\big) \\ + o_y\big(cos(\theta_{17}) sen(\theta_{13}) + cos(\theta_{13} + \theta_{15}) sen(\theta_{17})\big) + o_z cos(\theta_{13}) sen(\theta_{15})\big] \\ - \theta_9 \end{aligned} \tag{24}$$

Es importante mencionar que el proceso anterior es el mismo para calcular el valor de las posiciones articulares θ_8, $\theta_{10}, \theta_{12}, \theta_{14}, \theta_{16}$ y θ_{18} de la pierna izquierda.

3. Cinemática inversa de los brazos

En la Tabla 2 y Tabla 3 se proporcionan los parámetros de Denavit-Hartenberg que corresponden a la cadena cinemática del brazo derecho e izquierdo respectivamente.

Tabla 2. Parámetros de Denavit-Hartenberg para la cadena cinemática del brazo derecho.

Servomotor	α_i	l_i	θ_i	d_i
1	-90°	0	$\theta_1 - 90°$	0
3	-90°	L_5	$\theta_3 + 90°$	0
5	0°	L_4	θ_5	0

Tabla 3. Parámetros de Denavit-Hartenberg para la cadena cinemática del brazo izquierdo.

Servomotor	α_i	l_i	θ_i	d_i
2	90°	0	$\theta_2 + 90°$	0
4	90°	L_5	$\theta_4 - 90°$	0
6	0°	L_4	θ_6	0

Para definir el valor de las variables, se emplearon las medidas reales de las articulaciones de los brazos del robot: $L_3 = 6.8\ cm$, $L_4 = 1.7\ cm$.

Utilizando la ecuación (1) y los parámetros de la Tabla 2 se obtienen las siguientes matrices de transformación para las 3 articulaciones del brazo derecho:

$$Hd_0^1 = \begin{pmatrix} \cos(\theta_1) & 0 & -sen(\theta_1) & 0 \\ sen(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad Hd_1^2 = \begin{pmatrix} \cos(\theta_3) & 0 & -sen(\theta_3) & L_4\cos(\theta_3) \\ sen(\theta_3) & 0 & \cos(\theta_3) & L_4\text{sen}(\theta_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Hd_2^3 = \begin{pmatrix} \cos(\theta_5) & -sen(\theta_5) & 0 & L_3\cos(\theta_5) \\ sen(\theta_5) & \cos(\theta_5) & 0 & L_3\text{sen}(\theta_5) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Y las siguientes matrices de transformación corresponden al brazo izquierdo:

$$Hi_0^1 = \begin{pmatrix} \cos(\theta_2) & 0 & sen(\theta_2) & 0 \\ sen(\theta_2) & 0 & -\cos(\theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad Hi_1^2 = \begin{pmatrix} \cos(\theta_4) & 0 & sen(\theta_4) & L_4\cos(\theta_4) \\ sen(\theta_4) & 0 & -\cos(\theta_4) & L_4\text{sen}(\theta_4) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Hi_2^3 = \begin{pmatrix} \cos(\theta_6) & -sen(\theta_6) & 0 & L_3\cos(\theta_6) \\ sen(\theta_6) & \cos(\theta_6) & 0 & L_3\text{sen}(\theta_6) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Para obtener la cinemática inversa del brazo derecho, se considera la matriz inversa de Hd_0^1, y empleando el mismo método para obtener la cinemática inversa de las piernas del robot se define la ecuación matricial (25):

$$(Hd_0^1)^{-1} Hd_0^3 = Hd_1^2 Hd_2^3 \tag{25}$$

$$\begin{pmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} nd_x & od_x & ad_x & pd_x \\ nd_y & od_y & ad_y & pd_y \\ nd_z & od_z & ad_z & pd_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_3 & 0 & -S_3 & L_4C_3 \\ S_3 & 0 & C_3 & L_4S_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} C_5 & -S_5 & 0 & L_3C_5 \\ S_5 & C_5 & 0 & L_3S_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} nd_xC_1 + nd_yS_1 & od_xC_1 + od_yS_1 & ad_xC_1 + ad_yS_1 & pd_xC_1 + pd_yS_1 \\ -nd_z & -od_z & -ad_z & -pd_z \\ nd_yC_1 - nd_xS_1 & od_yC_1 - od_xS_1 & ad_yC_1 - ad_xS_1 & pd_yC_1 - pd_xS_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} C_3C_5 & -C_3S_5 & -S_3 & L_4C_3 + L_3C_3C_5 \\ C_5S_3 & -S_3S_5 & C_3 & L_4S_3 + L_3C_5S_3 \\ -S_5 & -C_5 & 0 & -L_3S_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{26}$$

Tomando el cociente de los elementos (2,4) y (1,4) del lado izquierdo y derecho de la ecuación (26) se calcula el ángulo θ_3 como sigue:

$$\frac{-pd_z}{pd_xC_1 + pd_yS_1} = \frac{L_4S_3 + L_3C_5S_3}{L_4C_3 + L_3C_3C_5}$$

$$\theta_3 = atan2\left(-pd_z, pd_x cos(\theta_1) + pd_y cen(\theta_1)\right) \tag{27}$$

Considerando el elemento (2,4) de la ecuación (26) se calcula el ángulo θ_5 como sigue:

$$-pd_z = L_4S_3 + L_3C_5S_3$$

$$\theta_5 = arccos\left(\frac{-pd_z - L_4 sen(\theta_3)}{L_3 sen(\theta_3)}\right) \tag{28}$$

Empleando el elemento (3,3) se calcula el ángulo θ_1:

$$ad_yC_1 - ad_xS_1 = 0$$

$$\theta_1 = \text{atan2}\left(ad_y, ad_x\right) \tag{29}$$

Para obtener la cinemática inversa del brazo izquierdo, se considera la matriz inversa de Hi_0^1, y las matrices Hi_1^2, Hi_2^3 y Hi_0^3 para definir la ecuación matricial (30):

$$(Hi_0^1)^{-1}Hi_0^3 = Hi_1^2Hi_2^3 \tag{30}$$

$$\begin{pmatrix} C_2 & S_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ S_2 & -C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} ni_x & oi_x & ai_x & pi_x \\ ni_y & oi_y & ai_y & pi_y \\ ni_z & oi_z & ai_z & pi_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_4 & 0 & S_4 & L_4C_4 \\ S_4 & 0 & -C_4 & L_4S_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} C_6 & -S_6 & 0 & L_3C_6 \\ S_6 & C_6 & 0 & L_3S_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} ni_xC_2 + ni_yS_2 & oi_xC_2 + oi_yS_2 & ai_xC_2 + ai_yS_2 & pi_xC_2 + pi_yS_2 \\ ni_z & oi_z & ai_z & pi_z \\ ni_xS_2 - ni_yC_2 & oi_xS_2 - oi_yC_2 & ai_xS_2 - ai_yC_2 & pi_xS_2 - pi_yC_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} C_4C_6 & -C_4S_6 & S_4 & L_4C_4 + L_3C_4C_6 \\ C_6S_4 & -S_4S_6 & -C_4 & L_4S_4 + L_3C_6S_4 \\ S_6 & C_6 & 0 & L_3S_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{31}$$

Tomando el cociente de los elementos (2,4) y (1,4) de ambos lados de la ecuación (31) se calcula el ángulo θ_4 como sigue:

$$\frac{pi_z}{pi_xC_2 + pi_yS_2} = \frac{L_4S_4 + L_3C_6S_4}{L_4C_4 + L_3C_4C_6}$$

$$\theta_4 = atan2\left(pi_z, pi_xCos(\theta_2) + pi_ySen(\theta_2)\right) \tag{32}$$

Considerando el elemento (2,4) de la ecuación (31) se calcula el ángulo θ_6 como sigue:

$$pi_z = L_4 S_4 + L_3 C_6 S_4$$

$$\theta_6 = arccos\left(\frac{pi_z - L_4 sen(\theta_4)}{L_3 sen(\theta_4)}\right) \tag{33}$$

Empleando el elemento (3,3) se calcula el ángulo θ_2:

$$ai_x S_2 - ai_y C_2 = 0$$

$$\theta_2 = atan2(ai_y, ai_x) \tag{34}$$

4. Resultados

En esta sección se presentan los resultados obtenidos de la simulación e implementación de las ecuaciones de cinemática inversa del robot Bioloid Premium, Se desarrolló un modelo 3D, utilizando "Robotics Toolbox" de Matlab, desarrollado por Corke [6] empleando los parámetros de la Tabla 1. Esta herramienta permite generar el modelo a partir de los parámetros de Denavit-Hartenberg del robot.

Empleando las ecuaciones (12), (15), (16), (22)-(24), (27)-(29), (32)-(34) que se obtuvieron en la secciones de la cinemática inversa del robot, se propusieron algunos parámetros para ubicar las piernas y brazos del robot en diferentes posiciones para validar los resultados del modelo.

Para la primera prueba de cinemática inversa aplicada a los brazos se emplearon los parámetros propuestos en las tablas 4 y 5. En la Figura 3 se muestran los resultados de simulación de las articulaciones del robot ubicadas en la parte superior del robot.

Tabla 4. Parámetros propuestos y resultados de la cinemática inversa para el brazo derecho

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$ad_y = 10$, $ad_x = -40$, $pd_x = 0$, $pd_y = -40$, $pd_z = 50$	$\theta_1 = 75.96$ $\theta_3 = -37.81$ $\theta_5 = 18.32$

Tabla 5. Parámetros propuestos y resultados de la cinemática inversa para el brazo izquierdo

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$ai_y = -10$, $ai_x = -40$, $pi_x = 0$, $pi_y = 40$, $pi_z = 50$	$\theta_2 = -75.96$ $\theta_4 = 37.81$ $\theta_6 = 18.32$

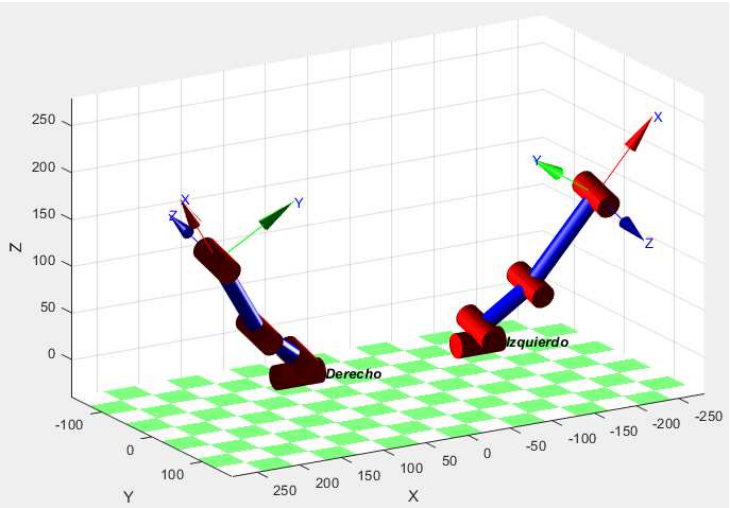


Figura 3. Posición de las articulaciones del robot empleando los valores de las tablas 4 y 5.

Para la primera prueba, se implementaron las ecuaciones que corresponden a la cinemática inversa de los brazos en el robot Bioloid, para posicionar y ubicar las articulaciones superiores con respecto a los parámetros de las tablas 4 y 5, tal como se aprecia en la Figura 4.

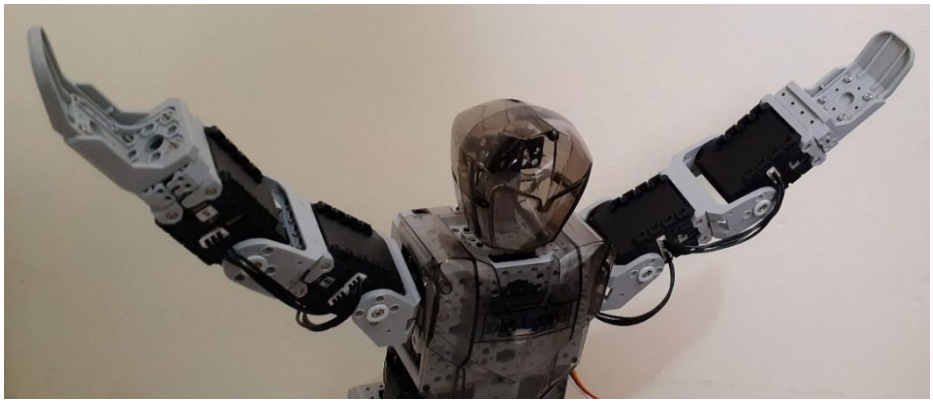


Figura 4. Implementación de los parámetros propuestos en la tabla 4 y 5.

En la segunda prueba se emplearon los parámetros propuestos en las tablas 6 y 7. En la Figura 5 se muestran los resultados de simulación de las articulaciones del robot ubicadas hacia el frente del robot.

Tabla 6. Parámetros propuestos y resultados de la cinemática inversa para el brazo derecho.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$ad_y = 40$, $ad_x = 10$, $pd_x = -50$, $pd_y = 20$, $pd_z = 30$	$\theta_1 = -14.03$ $\theta_3 = -60.65$ $\theta_5 = 75.15$

Tabla 7. Parámetros propuestos y resultados de la cinemática inversa para el brazo izquierdo.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$ai_y = -40$, $ai_x = 10$, $pi_x = -50$, $pi_y = 20$, $pi_z = 30$	$\theta_2 = 14.03$ $\theta_4 = 60.65$ $\theta_6 = 75.15$

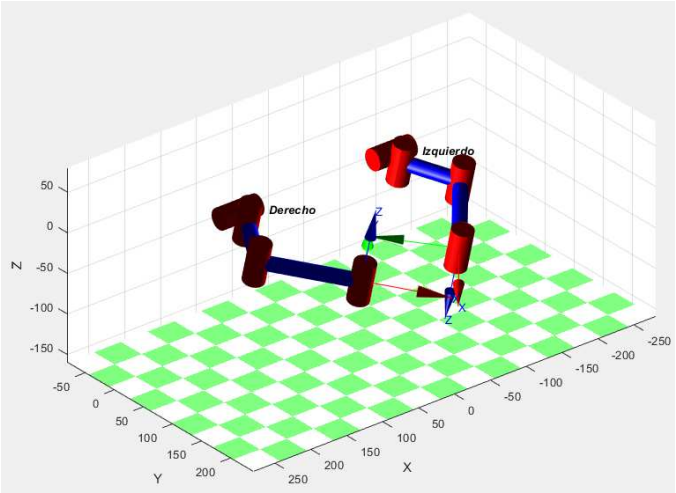


Figura 5. Posición de las articulaciones del robot empleando los valores de las tablas 6 y 7.

También se implementaron las ecuaciones que corresponden a la cinemática inversa de los brazos en el robot Bioloid, para posicionar y ubicar las articulaciones superiores con respecto a los parámetros de las tablas 6 y 7, tal como se aprecia en la Figura 6.

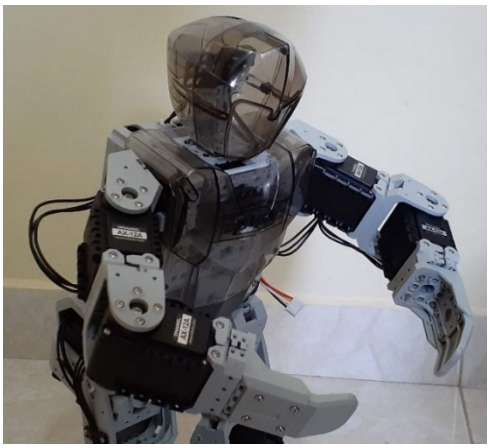


Figura 6. Implementación de los parámetros propuestos en la tabla 6 y 7.

En la tercera prueba se emplearon los parámetros propuestos en las tablas 8 y 9. En la Figura 7 se muestran los resultados de simulación de las articulaciones del robot ubicadas hacia el frente del robot.

Tabla 8. Parámetros propuestos y resultados de la cinemática inversa para el brazo derecho

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$ad_y = 18$, $ad_x = -2$, $pd_x = -10$, $pd_y = 20$, $pd_z = 25$	$\theta_1 = 6.34$ $\theta_3 = -17.18$ $\theta_5 = 5.99$

Tabla 9. Parámetros propuestos y resultados de la cinemática inversa para el brazo izquierdo.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$ai_y = -18$, $ai_x = -2$, $pi_x = -10$, $pi_y = -20$, $pi_z = 25$	$\theta_2 = -6.34$ $\theta_4 = 17.18$ $\theta_6 = 5.99$

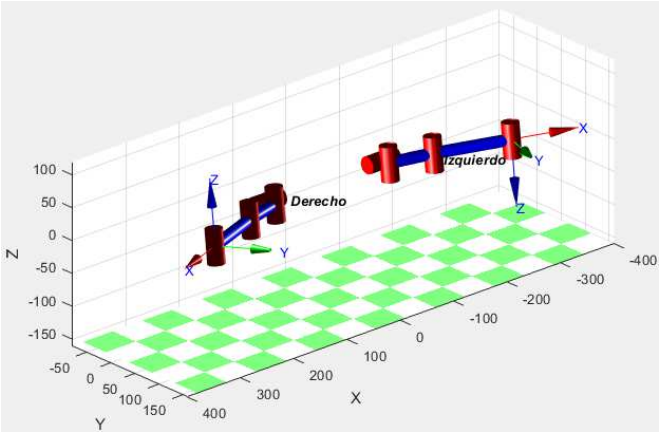


Figura 7. Posición de las articulaciones del robot empleando los valores de las tablas 8 y 9.

Posteriormente se implementaron las ecuaciones que corresponden a la cinemática inversa de los brazos en el robot Bioloid, para posicionar y ubicar las articulaciones superiores con respecto a los parámetros de las tablas 8 y 9, tal como se aprecia en la Figura 8.

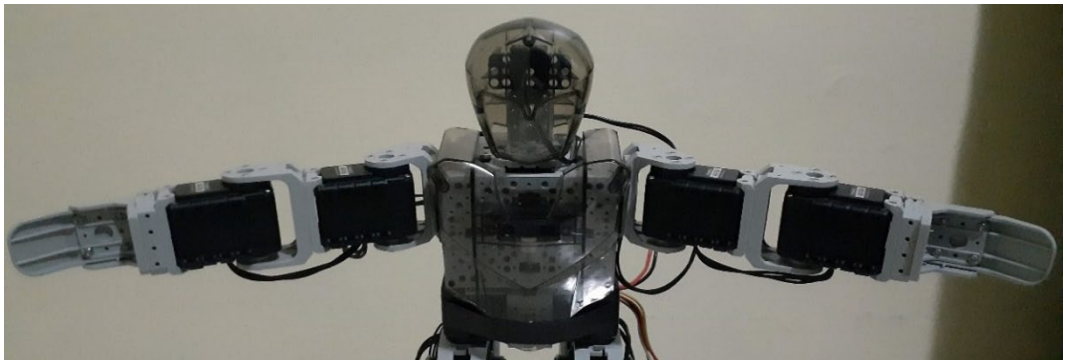


Figura 8. Implementación de los parámetros propuestos en la tabla 8 y 9.

Después se hicieron 4 pruebas diferentes para implementar las ecuaciones que corresponden a la cinemática inversa en las piernas del robot. En la Figura 9 se muestran los resultados de simulación de la primera prueba, empleando los parámetros propuestos en las tablas 10 y 11.

Tabla 10. Parámetros propuestos y resultados de la cinemática inversa para la pierna derecha.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$a_y = 0$, $a_x = 0$, $o_x = 0.5$, $o_y = -1.1$, $o_z = 0$, $p_x = 0$, $p_y = 50$, $p_z = 0$	$\theta_7 = -45$, $\theta_9 = 0$, $\theta_{11} = -60.88$, $\theta_{13} = -41.81$, $\theta_{15} = 0$, $\theta_{17} = 0$

Tabla 11. Parámetros propuestos y resultados de la cinemática inversa para la pierna izquierda.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$a_{yi} = 0$, $a_{xi} = 0$, $o_{xi} = -0.5$, $o_{yi} = -1.1$, $o_{zi} = 0$, $p_{xi} = 0$, $p_{yi} = 50$, $p_{zi} = 0$	$\theta_8 = 45$, $\theta_{10} = 0$, $\theta_{12} = 60.88$, $\theta_{14} = 41.81$, $\theta_{16} = 0$, $\theta_{18} = 0$

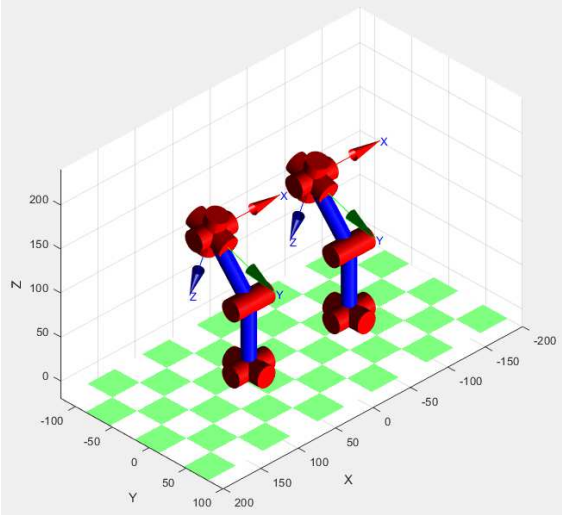


Figura 9. Posición de las articulaciones del robot empleando los valores de las tablas 10 y 11.

Para esta prueba, se implementaron las ecuaciones que corresponden a la cinemática inversa de las piernas del robot Bioloid, para posicionar y ubicar las articulaciones inferiores con respecto a los parámetros de las tablas 10 y 11, tal como se aprecia en la Figura 10.

Figura 10. Implementación de los parámetros propuestos en la tabla 10 y 11.

En la Figura 11 se muestran los resultados de simulación de la segunda prueba, empleando los parámetros propuestos en las tablas 12 y 13, en donde se propuso otra posición y ubicación de las piernas del robot.

Tabla 12. Parámetros propuestos y resultados de la cinemática inversa para la pierna derecha.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$a_y = 0$, $a_x = 0$, $o_x = 0.8$, $o_y = -1.1$, $o_z = 0$, $p_x = 0$, $p_y = 70$, $p_z = 0$	$\theta_7 = -45$, $\theta_9 = 0$, $\theta_{11} = -110.59$, $\theta_{13} = -68.96$, $\theta_{15} = 0$, $\theta_{17} = 0$

Tabla 13. Parámetros propuestos y resultados de la cinemática inversa para la pierna izquierda.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$a_{yi} = 0$, $a_{xi} = 0$, $o_{xi} = -0.8$, $o_{yi} = -1.1$, $o_{zi} = 0$, $p_{xi} = 0$, $p_{yi} = 70$, $p_{zi} = 0$	$\theta_8 = 45$, $\theta_{10} = 0$, $\theta_{12} = 110.59$, $\theta_{14} = 68.96$, $\theta_{16} = 0$, $\theta_{18} = 0$

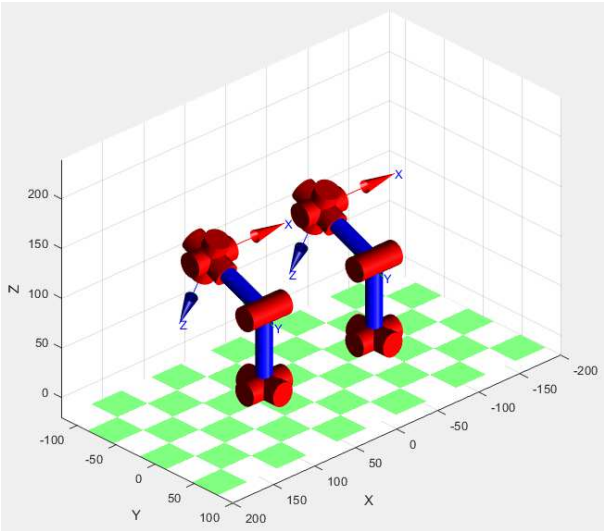


Figura 11. Posición de las articulaciones del robot empleando los valores de las tablas 12 y 13.

Para esta prueba, se implementaron las ecuaciones que corresponden a la cinemática inversa de las piernas del robot Bioloid, para posicionar y ubicar las articulaciones inferiores con respecto a los parámetros de las tablas 12 y 13, tal como se aprecia en la Figura 12.

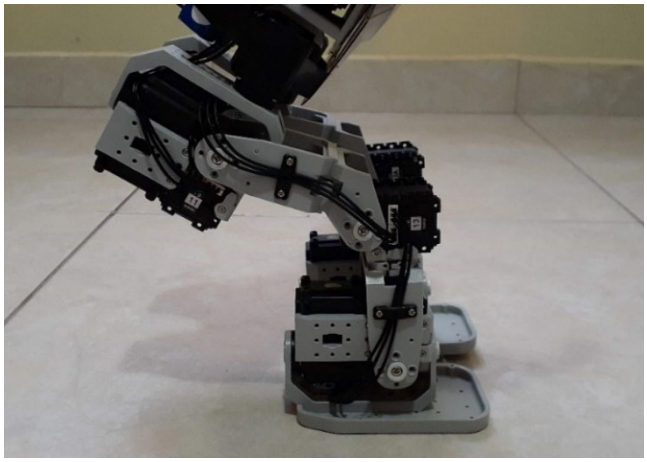


Figura 12. Implementación de los parámetros propuestos en la tabla 12 y 13.

En la Figura 13 se muestran los resultados de simulación de la tercera prueba, empleando los parámetros propuestos en las tablas 14 y 15, en donde se propuso una posición y ubicación diferentes para las piernas del robot.

Tabla 14. Parámetros propuestos y resultados de la cinemática inversa para la pierna derecha.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$a_y = 0$, $a_x = 0$, $o_x = 0$, $o_y = -1.1$, $o_z = 0$, $p_x = 0$, $p_y = 50$, $p_z = -20$	$\theta_7 = -45$, $\theta_9 = 0$, $\theta_{11} = -34.92$, $\theta_{13} = -41.81$, $\theta_{15} = 20.96$, $\theta_{17} = 0$

Tabla 15. Parámetros propuestos y resultados de la cinemática inversa para la pierna izquierda.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$a_{yi} = 0$, $a_{xi} = 0$, $o_{xi} = 0$, $o_{yi} = -1.1$, $o_{zi} = 0$, $p_{xi} = 0$, $p_{yi} = 50$, $p_{zi} = -20$	$\theta_8 = 45$, $\theta_{10} = 0$, $\theta_{12} = 34.92$, $\theta_{14} = 41.81$, $\theta_{16} = -20.96$, $\theta_{18} = 0$

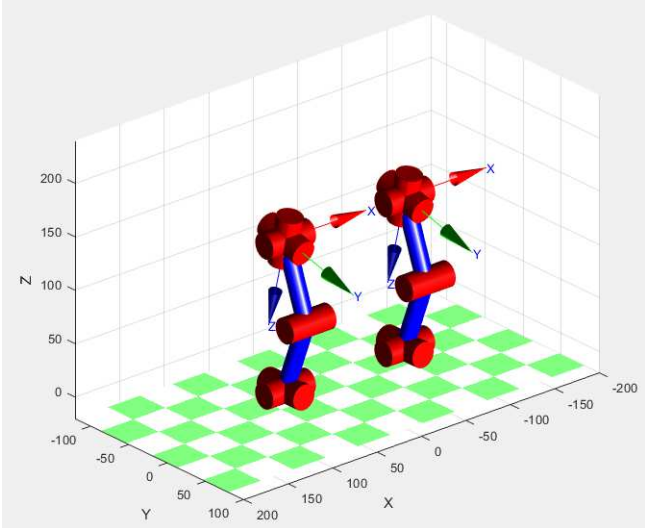


Figura 13. Posición de las articulaciones del robot empleando los valores de las tablas 14 y 15.

Para esta prueba, se implementaron las ecuaciones que corresponden a la cinemática inversa de las piernas del robot Bioloid, para posicionar y ubicar las articulaciones inferiores con respecto a los parámetros de las tablas 14 y 15, tal como se aprecia en la Figura 14.

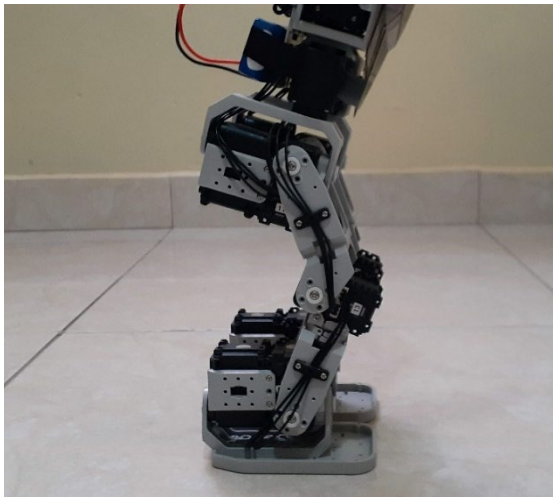


Figura 14. Implementación de los parámetros propuestos en la tabla 14 y 15.

En la Figura 15 se muestran los resultados de simulación de la cuarta prueba, empleando los parámetros propuestos en las tablas 16 y 17, en donde se propuso la útlima posición y ubicación de las piernas del robot.

Tabla 16. Parámetros propuestos y resultados de la cinemática inversa para la pierna derecha.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$a_y = 0$, $a_x = 0$, $o_x = 1.2$, $o_y = -1.1$, $o_z = 0$, $p_x = 0$, $p_y = 50$, $p_z = 20$	$\theta_7 = -45$, $\theta_9 = 0$, $\theta_{11} = -85.82$, $\theta_{13} = -41.81$, $\theta_{15} = -20.96$, $\theta_{17} = 0$

Tabla 17. Parámetros propuestos y resultados de la cinemática inversa para la pierna izquierda.

Parámetros de la cinemática inversa (mm)	Ángulos de los motores (grados)
$a_{yi} = 0$, $a_{xi} = 0$, $o_{xi} = 1.2$, $o_{yi} = -1.1$, $o_{zi} = 0$, $p_{xi} = 0$, $p_{yi} = 50$, $p_{zi} = 20$	$\theta_8 = 45$, $\theta_{10} = 0$, $\theta_{12} = 85.82$, $\theta_{14} = 41.81$, $\theta_{16} = 20.96$, $\theta_{18} = 0$

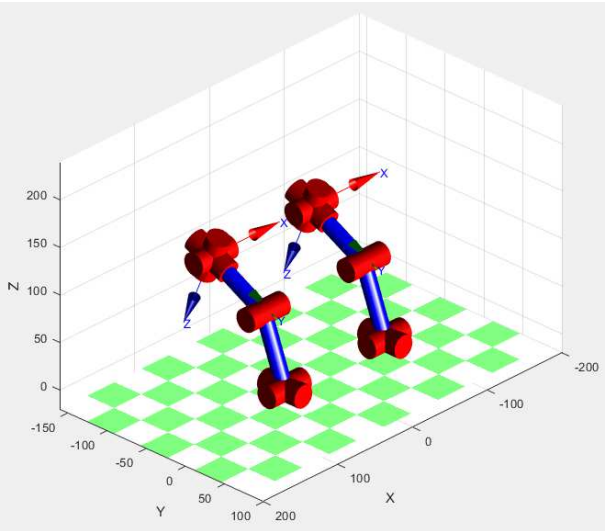


Figura 15. Posición de las articulaciones del robot empleando los valores de las tablas 16 y 17.

Para esta prueba, se implementaron las ecuaciones que corresponden a la cinemática inversa de las piernas del robot Bioloid, para posicionar y ubicar las articulaciones inferiores con respecto a los parámetros de las tablas 16 y 17, tal como se aprecia en la Figura 16.

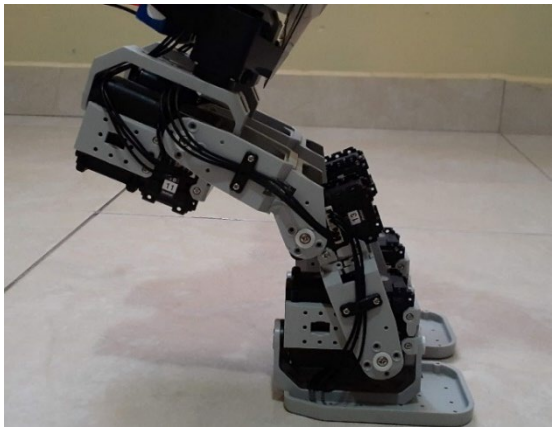


Figura 16. Implementación de los parámetros propuestos en la tabla 16 y 17.

5. Conclusiones

Las ecuaciones obtenidas en el modelo de cinemática directa e inversa fueron validadas correctamente a través de la simulación en Matlab. La función *arctan(.)* devuelve valores entre $-\pi/2$ y $\pi/2$, por lo que el rango de movimiento de algunas articulaciones estaría limitado. Para propósitos de programación, se empleó la función *atan2(.)* para tener un mayor rango de valores (rango de $-\pi$ a π) y poder aplicarlo posteriormente en el robot de manera experimental.

El modelo cinemático inverso tiene algunas restricciones de movimiento, debido a que algunos ángulos se resuelven a través de funciones *arcsen(.)* y *arccos(.)*, pero también a causa de que el robot presenta limitaciones físicas que provocan colisiones entre sus eslabones. Además, se consideraron algunos ángulos fijos para simplificar el cálculo de las ecuaciones, tales como: $\theta_7 = -45°, \theta_8 = 45°, \theta_9 = \theta_{10} = \theta_{15} = \theta_{16} = \theta_{17} = \theta_{18} = 0°$.

Es importante mencionar que los servomotores cuentan con un control de posición de fábrica, por lo cual, en las pruebas experimentales se mantienen estables las posiciones de las articulaciones solo con proporcionar la posición angular de cada una sin necesidad de implementar una ley de control.

Para trabajo a futuro, se propone la obtención del modelo dinámico y proponer una ley de control, tomando en cuenta los aspectos relacionados con los criterios de estabilidad de Lyapunov. Se mejorará el modelo 3D del robot para poder exportarlo a la herramienta Simscape de Matlab, y así tener un modelo virtual del robot que servirá como plataforma experimental para validar el modelo dinámico.

Referencias

[1] Ames, A. D. *Human-inspired control of bipedal walking robots*. IEEE Transactions on Automatic Control, 59(5), 1115-1130, 2014.

[2] Ogura, Y., Shimomura, K., Kondo, H., Morishima, A., Okubo, T., Momoki, S. & Takanishi, *A. Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot*. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3976-3981). IEEE, 2006.

[4] Wahrmann, D., Knopp, T., Wittmann, R., Hildebrandt, A. C., Sygulla, F., Seiwald, P. & Buschmann, T. *Modifying the estimated ground height to mitigate error effects on bipedal robot walking*. In Advanced Intelligent Mechatronics (AIM), 2017 IEEE International Conference on (pp. 1471-1476). IEEE, 2017.

[3] Wittmann, R., Hildebrandt, A. C., Wahrmann, D., Sygulla, F., Rixen, D. & Buschmann, T. *Model-based predictive bipedal walking stabilization*. In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids) (pp. 718-724). IEEE, 2016.

[5] Kajita, S., Benallegue, M., Cisneros, R., Sakaguchi, T., Nakaoka, S. I., Morisawa, M. & Kanehiro, F. *Biped walking pattern generation based on spatially quantized dynamics*. In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids) (pp. 599-605). IEEE, 2017.

[6] Kajita, S., Benallegue, M., Cisneros, R., Sakaguchi, T., Nakaoka, S. I., Morisawa, M. & Kanehiro, F. *Biped gait control based on spatially quantized dynamics*. In 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids) (pp. 75-81). IEEE, 2018.

[7] Hein, D., Hild, M. & Berger, R. *Evolution of biped walking using neural oscillators and physical simulation*. In Robot Soccer World Cup (pp. 433-440). Springer, Berlin, Heidelberg, 2007.

[8] Meggiolaro, M. A., Neto, M. S. & Figueroa, A. L. *Modeling and Optimization with Genetic Algorithms of Quasi-Static Gait Patterns in Planar Biped Robots*. In Congreso Internacional de Ingeniería Mecatrónica y Automatización (CIIMA 2016) (pp. 1-10), 2016.

[9] Tanwani, A. K. *Optimizing walking of a humanoid robot using reinforcement learning (Doctoral dissertation, MS thesis*, Warsaw University of Technology), 2011.

[10] Cerritos-Jasso, J. R., Camarillo-Gómez, K. A., Monsiváis-Medina, J. A., Castillo-Alfaro, G., Pérez-Soto, G. I. & Pámanes-García, J. A. *Kinematic Modeling of a Humanoid Soccer–Player: Applied to BIOLOID Premium Type A Robot.* In FIRA RoboWorld Congress (pp. 49-63). Springer, Berlin, Heidelberg, 2013.

[11] Arias, L. E., Olvera, L. I., Pámanes, J. A. & Núñez, J. V. *Patrón de marcha 3D de tipo cicloidal para humanoides y su aplicación al robot Bioloid*. Revista Iberoamericana de Ingeniería Mecánica, 18(1), 3, 2014.

[12] Chiang, H. D. & Tsai, C. S. *Kinematics Analysis of a Biped Robot*. In Proceeding of 2011 International Conference on Service and Interactive Robots National Chung Hsing University, 2011.

[13] Bravo Montenegro, D. A. *Generación de trayectorias para un robot bípedo basadas en captura de movimiento humano*, 2016.

[14] RobotShop inc. *Bioloid Premium Robot Kit*. Disponible en https://www.robotshop.com/us/es/bioloid-premium-robot-kit.html. 1 de abril de 2021.

[15] Reyes, F. *Robótica - control de robots manipuladores*. Alfaomega grupo editor, pp. 212-218, 2011.

[16] Giraldo, L. F., Delgado, E. & Castellanos, G. *Cinemática inversa de un brazo robot utilizando algoritmos genéticos*. Revista Avances en Sistemas e Informática, 3(1), 29-34, 2006.

[17] Featherstone, R. *Position and velocity transformations between robot end-effector coordinates and joint angles*. The International Journal of Robotics Research, 2(2), 35-45, 1983

[18] Granja Oramas, M. V. *Modelación y análisis de la cinemática directa e inversa del manipulador Stanford de seis grados de libertad*, Bachelor's thesis, Quito, 2014.

[19] Franco, E. H. & Guerrero, R. V. *Diseño Mecánico y Análisis Cinemático del Robot Humanoide AXIS. Pistas Educativas*, 35(108), 2018.

[19] Corke, P. & Robotics, V. control: *Fundamental algorithms in MATLAB*, 2011.

Diseño Colaborativo en Mecatrónica, Capítulo 7, pp. 88 - 101
ISBN: 978-607-9394-23-3, 2021.

Aplicación de un controlador difuso y un PID clásico a la estación de nivel ETLFIE I

Charre Ibarra Saida Miriam, Gudiño Lau Jorge, Alcalá Rodríguez Janeth Aurelia, Duran Fonseca Miguel Ángel, Chávez Montejano Fidel

Facultad de Ingeniería Electromecánica. Universidad de Colima
Km. 20 Carretera Manzanillo-Barra de Navidad.
Manzanillo, Col. México

scharre@ucol.mx

Resumen

En el presente trabajo se describe la aplicación de controladores a una estación didáctica de trabajo en la Facultad de Ingeniería Electromecánica de la Universidad de Colima, la cual fue construida para experimentar con el control de la variable nivel. Se utilizan dos estrategias de control, un controlador difuso y un PID clásico, con la finalidad de analizar las respuestas transitorias de cada uno. El desarrollo de los controladores se realiza utilizando el software de programación gráfica LabVIEW ® y la librería PID and Fuzzy Logic Toolkit, así como un sistema de adquisición de datos, logrando monitorear y establecer el valor deseado en el sistema en tiempo real.

Palabras clave: Variable nivel, LabVIEW, PID, lógica difusa, funciones de membresía, instrumentación virtual.

1. Introducción

Las diferentes revoluciones industriales por las que ha pasado la humanidad han impactado diferentes ámbitos en la sociedad, desde transformaciones en aspectos económicos, sociales, culturares y tecnológicos; la última de estas revoluciones es la cuarta y al igual que las anteriores es un proceso de desarrollo tecnológico e industrial, posee como un elemento clave a las fábricas inteligentes, capaces de mejorar la eficiencia de los recursos y de optimizar producción adaptándose a las necesidades. Esta cuarta revolución promueve la sinergia de distintos desarrollos en áreas como biología sintética, fabricación aditiva, internet de las cosas, inteligencia artificial, entre otras. En entornos industriales es importante la implementación de estas tecnologías para resolver problemas de estandarización, flexibilidad, confiabilidad y costos de actualización tecnológica [1][2][3]. El propósito de este trabajo es mostrar la aplicación de una técnica de inteligencia artificial para sistemas donde se controla el nivel de líquidos y comparar su respuesta con un control clásico.

La Inteligencia Artificial (IA) es una respuesta al deseo de aproximar el comportamiento y el pensamiento humano a diversos sistemas para la solución de determinadas problemáticas, buscan básicamente que algoritmos informáticos emulen la observación, el entendimiento y el razonamiento humano. Los esfuerzos de la IA se enfocan en lograr la comprensión y construcción de entidades inteligentes [3][4].

Los sistemas difusos tienen la capacidad de emular la forma de la inferencia humana y además pueden almacenar la experiencia en forma lingüística. Estos sistemas forman una rama de la IA y le permiten a una computadora analizar información del mundo real en una escala entre lo falso y

verdadero; la lógica difusa puede controlar o describir un sistema usando reglas de sentido común que se refieren a cantidades indefinidas, frecuentemente las reglas son tomadas de expertos que poseen un gran conocimiento adquirido a través de años de observación y experimentación, que solo puede expresarse verbalmente y mediante la lógica difusa se puede representar [4][5].

El uso de la lógica difusa en los sistemas de control ha sido motivado por el deseo de mejorar la robustez, la adaptabilidad y la autonomía, proporcionando un diseño de control simplificado para modelos complejos, sin la necesidad de contar con el modelo de la planta, ha tenido aplicaciones en sistemas dinámicos tales como robots móviles, sistemas didácticos o subactuados entre otros [6][7][8]. Por otra parte, los controladores convencionales generan las señales que requieren los actuadores basándose en el modelo matemático del proceso a controlar.

La ecuación (1), describe a un controlador clásico formado por las tres acciones básicas: Proporcional, Integral y Derivativa, en su forma analógica [9][10]:

$$u(t) = k_p\, E(t) + \frac{K_p}{T_i}\int_0^t E(t)dt + K_p T_d \frac{dE(t)}{dt} \tag{1}$$

donde:

$E(t)$ es el error
K_p es la ganancia proporcional
T_i es el tiempo integral
T_d es el tiempo derivativo

2. Descripción del proceso

La planta o proceso consiste en una estación de trabajo didáctica en arquitectura abierta diseñada y construida en la Facultad de Ingeniería Electromecánica de la Universidad de Colima, la cual se muestra en la figura 1.

Figura 1. Estación didáctica de nivel.

Dentro de los procesos industriales la medición y el control de la variable nivel se hace imprescindible cuando se desea tener una producción continua, cuando un proceso requiere del control de volúmenes de líquidos o bien en el caso más simple, para evitar que un líquido se derrame. Aplicación del control de esta variable se observa en industrias dedicadas al tratamiento de aguas,

a la generación de energía eléctrica, a los hidrocarburos, a la transformación de bienes de consumo, a procesos químicos, entre otras. La estación didáctica construida en la Facultad permite fortalecer la formación de los estudiantes en el área de Instrumentación y Control, al ser diseñada en arquitectura abierta permite experimentar con la variable nivel aplicando diferentes técnicas de control [11][12][13].

El depósito donde se controla el nivel tiene una capacidad de 125 litros de agua y cuenta con un sensor ultrasónico Banner Q45U para obtener el nivel del líquido, éste envía una señal de 4 a 20 mA al controlador y posee como actuador una electroválvula Honeywell modelo M7284A1004, que opera con señales de 0 a 20 mA.

3. Controlador difuso

Una forma de incorporar esquemas de razonamiento típicamente humanos a los sistemas de control es la lógica difusa. Los controladores difusos son las aplicaciones más importantes de la teoría difusa, usan el conocimiento adquirido por la experiencia con un proceso en lugar de ecuaciones diferenciales como los controladores convencionales, convirtiendo el conocimiento experto en estrategias de control automático. Este conocimiento se expresa empleando variables lingüísticas descritas mediante conjuntos difusos.

Las cuatro partes importantes que componen un controlador difuso son: La interfaz de difusificación que convierte los datos de entrada en valores lingüísticos que son las etiquetas de las funciones de pertenencia o conjuntos difusos; la base de conocimientos que contiene toda la información de la aplicación que se va a controlar y consta de una base de datos y reglas lingüísticas; la lógica de decisiones que representa el núcleo del controlador y a partir de la misma se simula la lógica que utilizan las personas para tomar decisiones; y la interfaz de desdifusificación que se encarga de obtener la acción de control nítida a partir de una acción de control difusa, ver figura 2 [4].

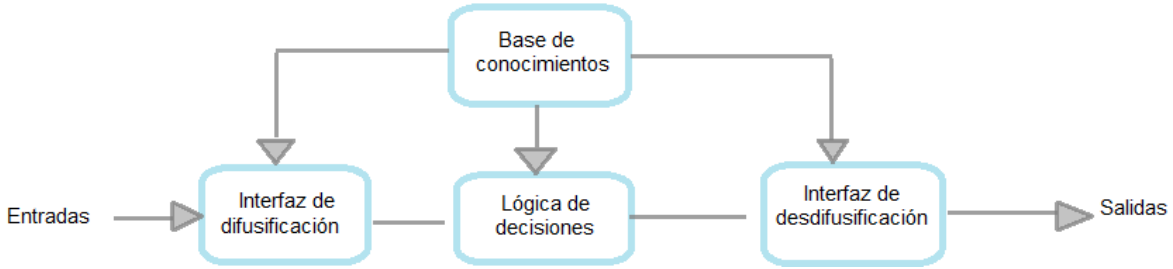


Figura 2. Estructura de un controlador difuso.

La figura 3, muestra el diagrama de bloques del sistema de control PD difuso a desarrollar.

La variable del proceso (nivel) es medida por el sensor, posteriormente esa lectura es restada al punto de ajuste previamente establecido, dando como resultado la señal de error (E).

El controlador tiene como señales de entrada el Error (E) y la Derivada del Error (dE/dt), en base a esas dos señales el controlador, por medio del sistema difuso que se diseña, determina la respuesta más adecuada para corregir el error existente en el proceso y así igualar el valor de la variable del proceso con el punto de ajuste establecido.

La salida del controlador será la acción aplicada al proceso, por lo cual el sistema será MISO (Multiple Input Single Ouput), por contar con dos entradas y una sola salida, figura 4.

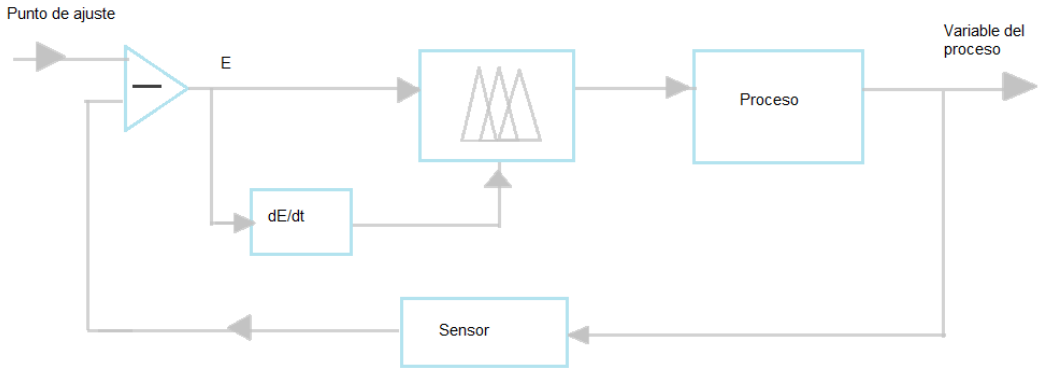


Figura 3. Sistema de control PD difuso en lazo cerrado.

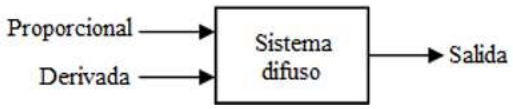


Figura 4. Sistema difuso MISO.

Para las variables lingüísticas E*rror*, D*erivada del Error* y S*alida* del controlador difuso se eligen las etiquetas mostradas en la tabla 1.

Tabla 1. Etiquetas de las variables lingüísticas.

Entradas Error y Derivada del Error	Salida
MP = Muy Positivo	TA = Totalmente Abierto
P = Positivo	AP = Abre Poco
C = Cero	S = Mantener Posición
N = Negativo	CP = Cierra Poco
MN = Muy Negativo	TC = Totalmente Cerrado

El desarrollo del controlador se lleva a cabo aplicando el conocimiento experto y programando el sistema difuso con el software LabVIEW ® usando el toolkit PID and FUZZY Logic. Posteriormente se diseña un instrumento virtual para el control y monitoreo de la variable nivel.

La programación se inicia ejecutando la aplicación *Fuzzy System Designer* con la cual se crean las funciones de membresía y las reglas del sistema difuso, figura 5. Está compuesta por 3 pestañas llamadas *Variables*, *Rules* y *Test System*. En la pestaña *Variables* se crean las variables lingüísticas tanto de las entradas como de la salida del sistema difuso, en la pestaña *Rules* se establecen las reglas del controlador y por último la pestaña *Test System* tiene la función de mostrar por medio de un modelado 3D la respuesta que entrega el controlador una vez que se hayan creado las funciones de membresía y las reglas.

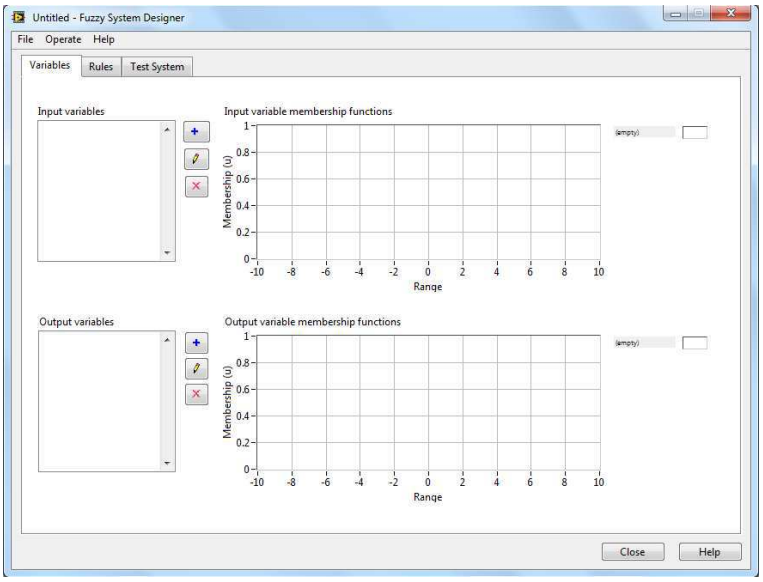


Figura 5. Ventana de la aplicación Fuzzy System Designer [14].

3.1 Variables

Está opción es para definir las variables lingüísticas de entrada y salida. Se divide en dos secciones, una se refiere a las variables de entrada (parte superior) y la otra a las variables de salida (parte inferior). Presenta el espacio para graficar las funciones de membresía, nombrar las variables y colocar el rango de trabajo de cada variable.

El procedimiento es el siguiente, primero se coloca el nombre o etiqueta de la función de membresía (por defecto no name 0), posteriormente se elige el tipo de función que se empleará, presionar el botón ▼ y seleccionar la función más adecuada para el sistema (por defecto viene una función triangular), también se puede modificar el color que tendrá la función de membresía posicionando el puntero en el cuadro de color negro y seleccionando el color para la función de membresía (por defecto viene en color negro), por último colocar los puntos o límites en los que estará posicionada la función de membresía, en este caso por ser una función triangular se deben de colocar los tres puntos que conforman un triángulo (por defecto -10, 0, 10).

En la figura 6 se muestran las funciones de membresía de las variables lingüísticas tanto de las entradas (Error y la Derivada del Error) como de la salida (respuesta del controlador difuso).

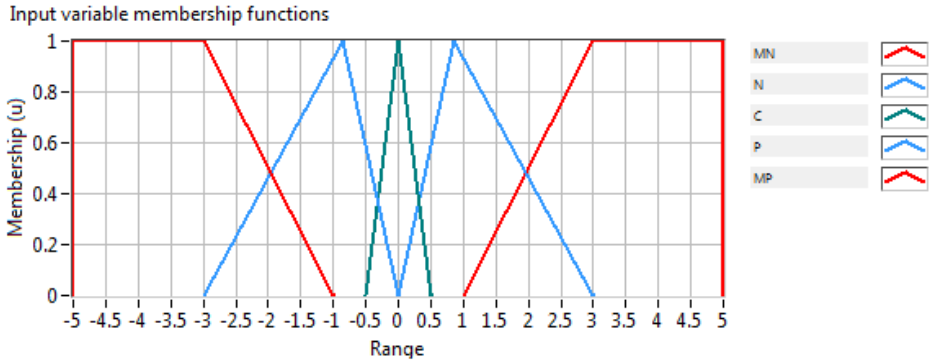
3.1 Reglas

Se llaman reglas difusas al conjunto de proposiciones IF-THEN que modelan el problema que se quiere resolver. Una regla difusa simple tiene la forma:

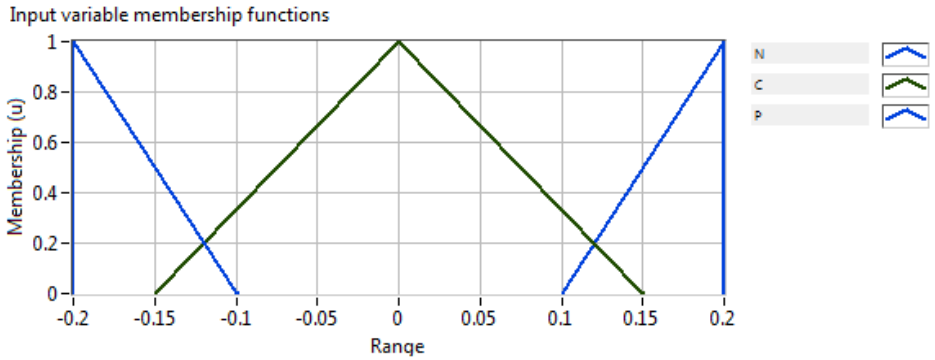
"Si *a* es *U* y *b* es *V*"

donde *U* y *V* son conjuntos difusos definidos en los rangos de *"a"* y *"b"* respectivamente.

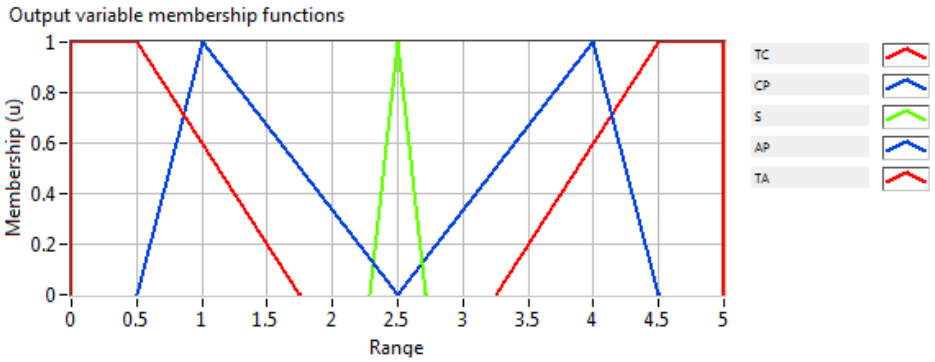
En la lógica difusa una regla se ejecuta si existe un grado de similaridad distinto de cero entre la primera premisa y el antecedente de la regla y el resultado de ejecución de la regla es un consecuente que tiene un grado de similaridad distinto de cero con el consecuente de la regla. A partir de las funciones de membresía para cada una de las entradas y la salida se elaboró la base de reglas mostrada en la tabla 2.



a) Funciones de membresía para el Error



b) Funciones de membresía para la Derivada del Error



c) Funciones de membresía para la Salida

Figura 6. Funciones de membresía del controlador difuso.

Tabla 2. Reglas difusas.

REGLAS		Derivada del Error		
		N	C	P
Error	MP	TA	TA	AP
	P	TA	AP	AP
	C	AP	S	CP
	N	CP	CP	TC
	MN	CP	TC	TC

Para su programación en LabVIEW®, se ingresa a la pestaña *Rules* de la aplicación *Fuzzy System Designer*, figura 7. Para la creación de las reglas se deben tener anteriormente establecidas las funciones de membresía de las variables de entrada y la salida.

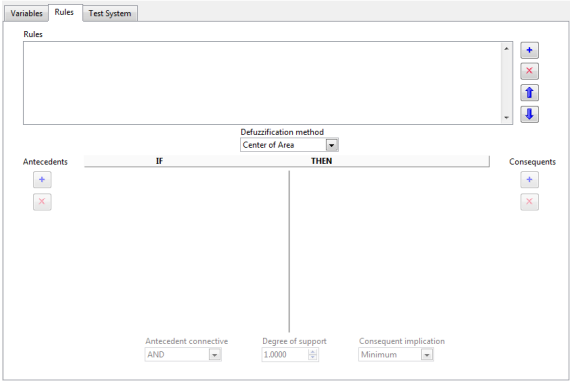


Figura 7. Pestaña Rules de la aplicación Fuzzy System Designer.

La figura 8 muestra las 15 reglas programadas para el sistema difuso.

1. IF 'Error' IS 'MP' AND 'Derivada E' IS 'N' THEN 'Salida' IS 'TA'
2. IF 'Error' IS 'MP' AND 'Derivada E' IS 'C' THEN 'Salida' IS 'TA'
3. IF 'Error' IS 'MP' AND 'Derivada E' IS 'P' THEN 'Salida' IS 'TA'
4. IF 'Error' IS 'P' AND 'Derivada E' IS 'N' THEN 'Salida' IS 'TA'
5. IF 'Error' IS 'P' AND 'Derivada E' IS 'C' THEN 'Salida' IS 'AP'
6. IF 'Error' IS 'P' AND 'Derivada E' IS 'P' THEN 'Salida' IS 'AP'
7. IF 'Error' IS 'C' AND 'Derivada E' IS 'N' THEN 'Salida' IS 'AP'
8. IF 'Error' IS 'C' AND 'Derivada E' IS 'C' THEN 'Salida' IS 'S'
9. IF 'Error' IS 'C' AND 'Derivada E' IS 'P' THEN 'Salida' IS 'CP'
10. IF 'Error' IS 'N' AND 'Derivada E' IS 'N' THEN 'Salida' IS 'CP'
11. IF 'Error' IS 'N' AND 'Derivada E' IS 'C' THEN 'Salida' IS 'CP'
12. IF 'Error' IS 'N' AND 'Derivada E' IS 'P' THEN 'Salida' IS 'TC'
13. IF 'Error' IS 'MN' AND 'Derivada E' IS 'N' THEN 'Salida' IS 'TC'
14. IF 'Error' IS 'MN' AND 'Derivada E' IS 'C' THEN 'Salida' IS 'TC'
15. IF 'Error' IS 'MN' AND 'Derivada E' IS 'P' THEN 'Salida' IS 'TC'

Figura 8. Conjunto de reglas del sistema difuso.

3.2 Prueba del sistema

Por último, la pestaña Test System presenta el comportamiento del sistema difuso, donde se observa la superficie que representa la salida del controlador difuso como función del Error y la Derivada del Error, figura 9.

3.3 Programación de la interfaz virtual

Para el diseño de la interfaz virtual en la plataforma LabVIEW® se toma en cuenta la operación en tiempo real del sistema difuso, mediante el uso de un sistema de adquisición de datos para la lectura de la variable del proceso y la aplicación del controlador, lo que implica el uso del driver *DAQmx – Data Acquisition*. Por otra parte para el controlador difuso es necesaria la utilización de 3 bloques, Current VI´s Path, Strip Path y el FL Load Fuzzy System, figura 10.

El diagrama a bloques del Instrumento Virtual (VI) se presenta en la figura 11.

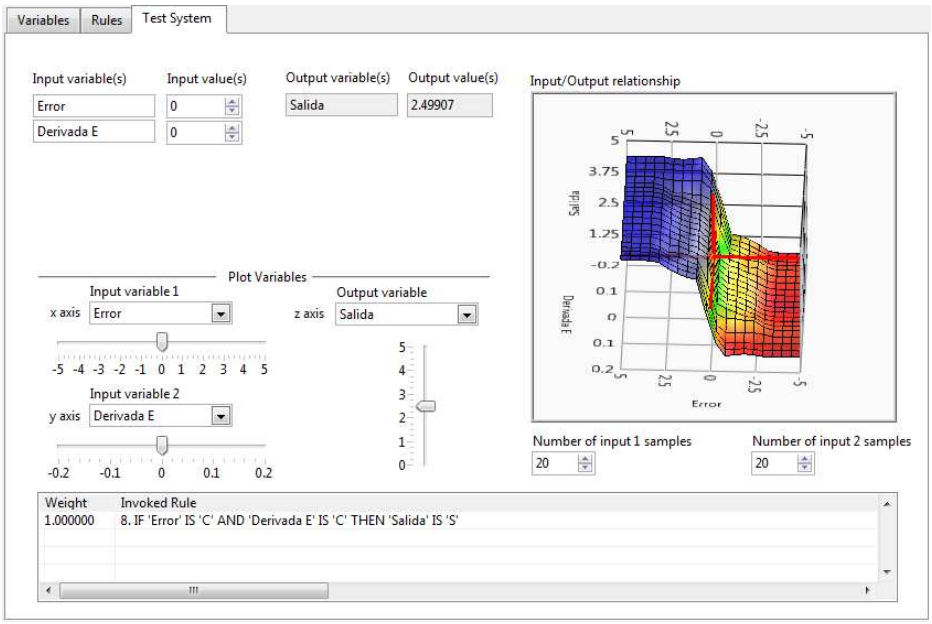


Figura 9. Pestaña Test System de la aplicación Fuzzy System Designer.

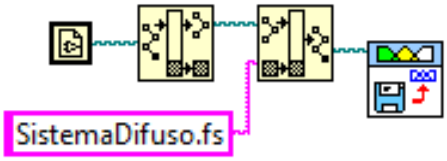


Figura 10. Bloques de programación para el conjunto difuso.

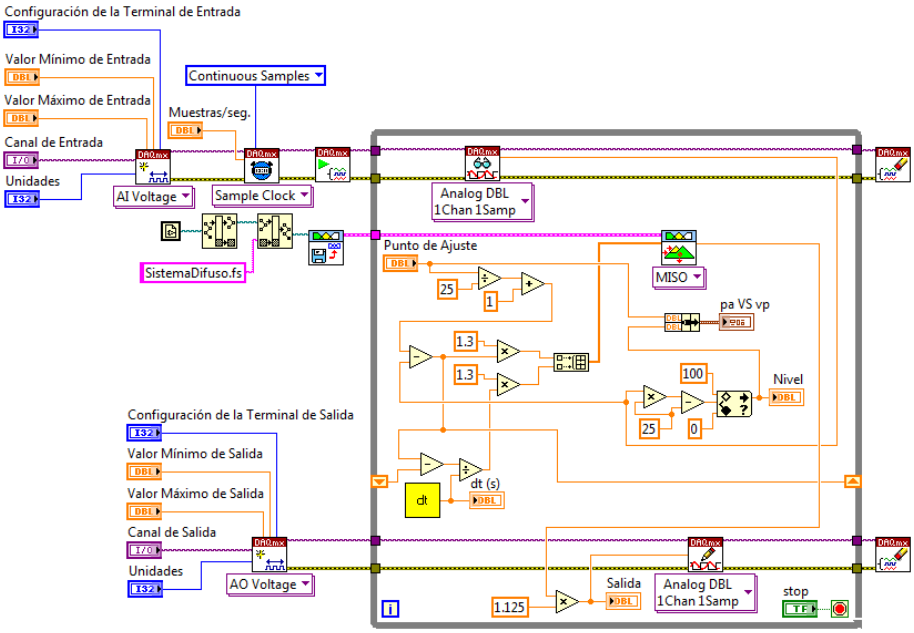


Figura 11. Diagrama a bloques del instrumento virtual del controlador difuso.

La figura 12 muestra el panel frontal terminado, en el cual se pueden seleccionar canales de entrada y salida para la lectura y el envío de las señales del proceso, establecer valores para el punto de ajuste y observar en un indicador tipo tanque el porcentaje de llenado en la estación, así como en registradores las respuestas transitorias del sistema.

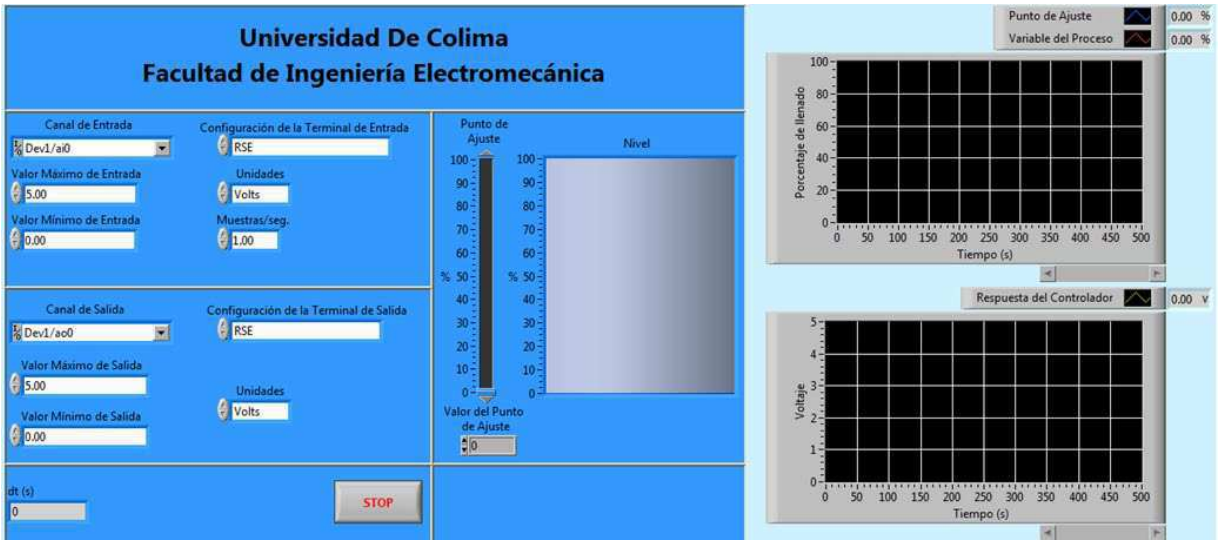


Figura 12. Interfaz virtual del controlador difuso.

4. Controlador PID

Estos corresponden a un modelo clásico conocido como controlador proporcional integral derivativo (PID), ver figura 13; ha sido una estructura aplicada durante décadas en el control de procesos, existiendo una gran cantidad de productos industriales que lo utilizan, una de las desventajas de este tipo de controlador es encontrar los valores de sus ganancias ya que puede haber una infinidad de soluciones de las mismas. Los controladores PID son generalmente usados en el nivel de control más bajo, por debajo de algunos dispositivos de mediano nivel como PLCs, supervisores y sistemas de monitoreo.

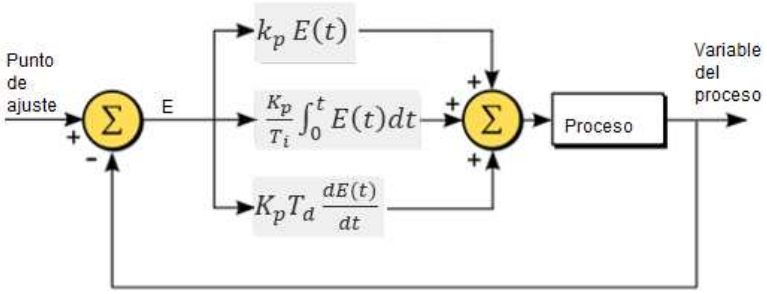


Figura 13. Control PID.

Para la implementación del control PID clásico se utiliza la librería PID Control Toolkit de LabVIEW; así como el driver *DAQmx – Data Acquisition*, con la finalidad de que opere en tiempo real. La figura 14 presenta el diagrama a bloques del programa para el controlador.

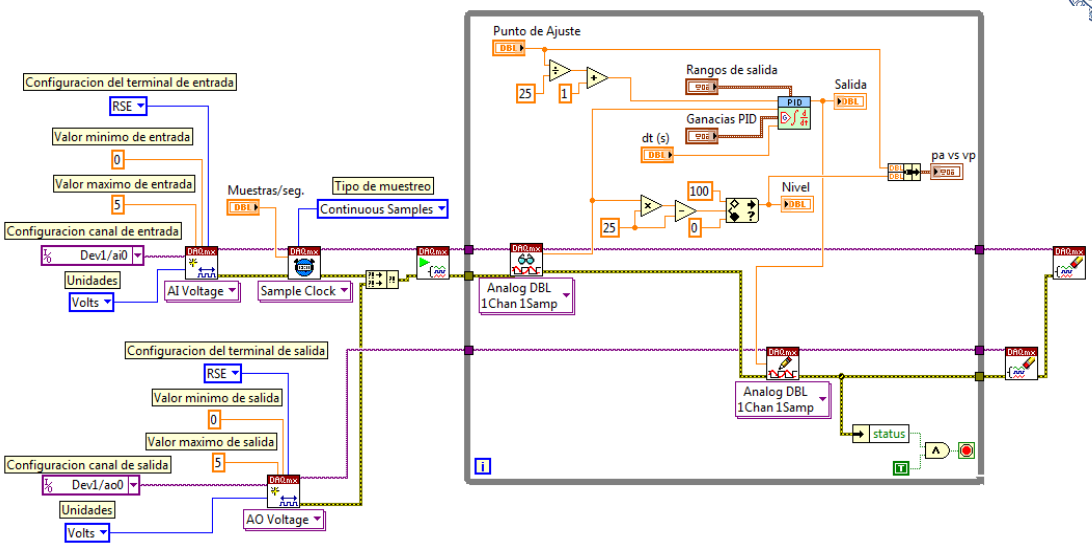


Figura 14. Diagrama a bloques del controlador clásico.

La figura 15 presenta la interfaz virtual para operar el PID programado, donde se establecen las ganancias, el valor del punto de ajuste y se observan los cambios en la variable del proceso y el controlador.

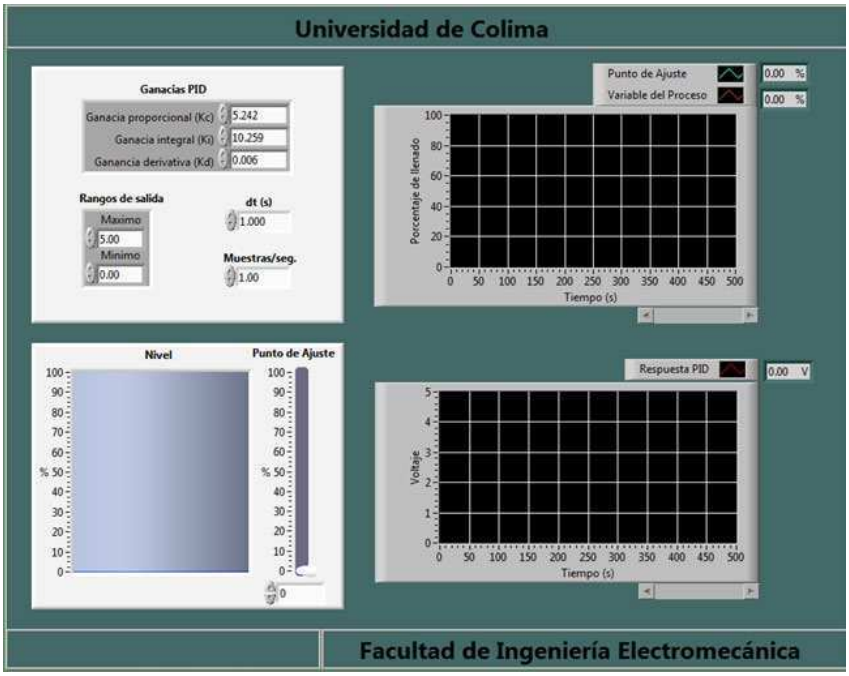


Figura 15. Interfaz virtual del controlador clásico.

5. Sistema de adquisición de datos

Se implementa un sistema de adquisición de datos utilizando la tarjeta NI USB-6009, con la finalidad de aplicar los controladores a la estación de nivel ETLFIE I. Debido a que las entradas y salidas analógicas de la tarjeta de adquisición de datos tienen un rango de 0V a 5V de c.d., es

necesario acondicionar la señal proveniente del sensor ultrasónico (4-20 mA), así como la señal de control aplicada a la válvula (0-20 mA). Se construye un circuito electrónico convertidor de corriente a voltaje y de voltaje a corriente, logrando comunicar la computadora con el proceso, figura 16.

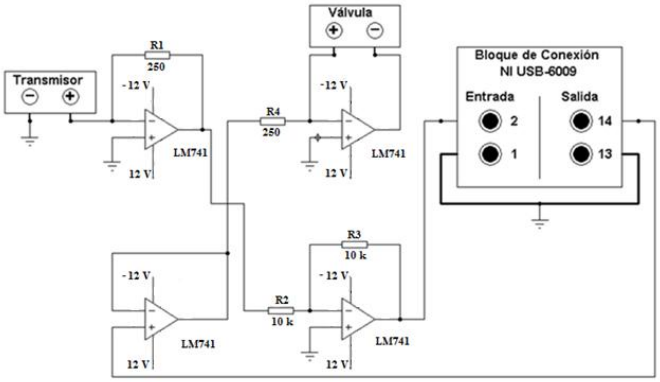


Figura 16. Diagrama del convertidor corriente/voltaje y voltaje/corriente.

6. Resultados

A continuación, se presentan los resultados obtenidos de la aplicación de un controlador difuso tipo PD y un controlador clásico PID, para la variable de nivel.

En la figura 17 y figura 18 se muestran las respuestas transitorias de los controladores, para dos cambios en la variable del proceso. El primero en forma ascendente con un valor inicial correspondiente a 20 % y un punto de ajuste fijado a 30 %, posteriormente en forma descendente iniciando con un 30 % y se fija un punto de ajuste de 20 %.

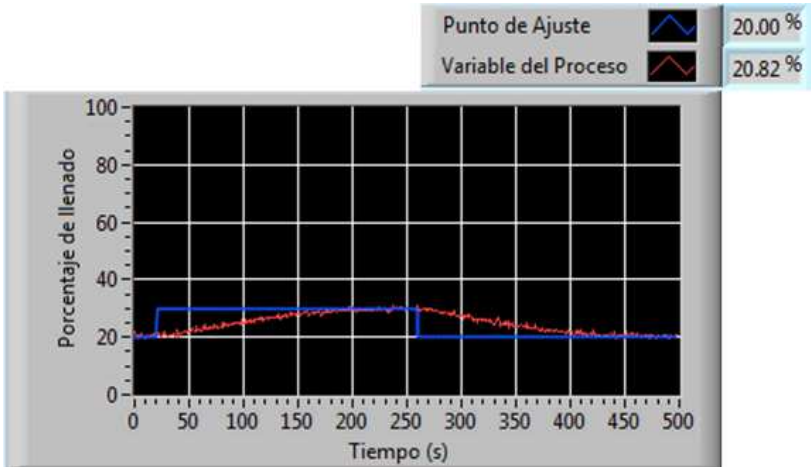


Figura 17. Comportamiento de la variable del proceso con el controlador PD difuso.

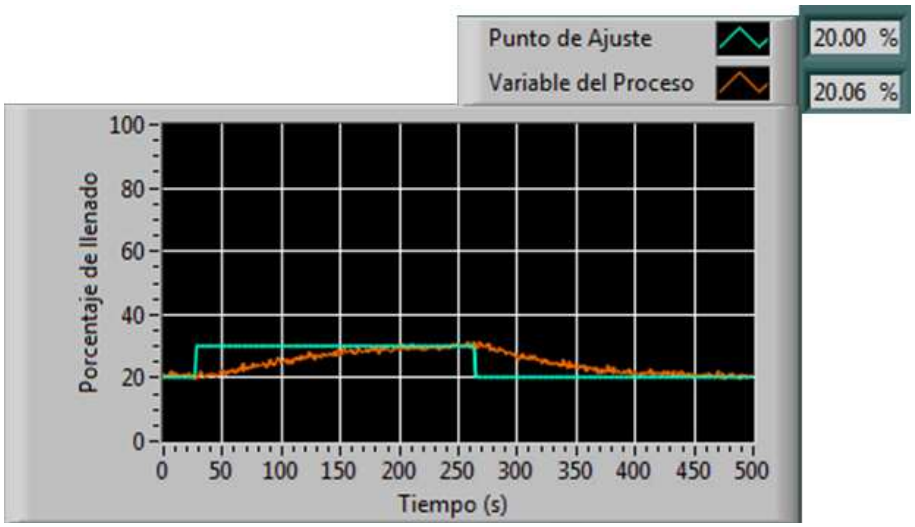


Figura 18. Comportamiento de la variable del proceso con el controlador clásico.

En la figura 17 se observa que el controlador difuso presenta una respuesta transitoria en la cual el tiempo de subida es de 125 s, el tiempo de establecimiento es de 155 s y la respuesta no presenta sobreimpulso. En la misma imagen se logra apreciar un cambio en forma descendente donde los valores de la respuesta transitoria son de 130 s para el tiempo de bajada y un tiempo de 160 s de establecimiento de la señal, tampoco presenta sobreimpulso.

De la figura 18 se puede observar que el controlador clásico presenta una respuesta transitoria con un tiempo de subida es de 125 s, el tiempo de establecimiento es de 175 s y la respuesta no presenta sobreimpulso. En la misma figura se aprecia un cambio en forma descendente donde los valores de la respuesta transitoria son de 105 s para el tiempo de bajada y un tiempo de 165 s de establecimiento de la señal, tampoco presenta sobreimpulso.

En la figura 19 y figura 20 se presentan las respuestas entregadas por el controlador difuso y el controlador clásico respectivamente.

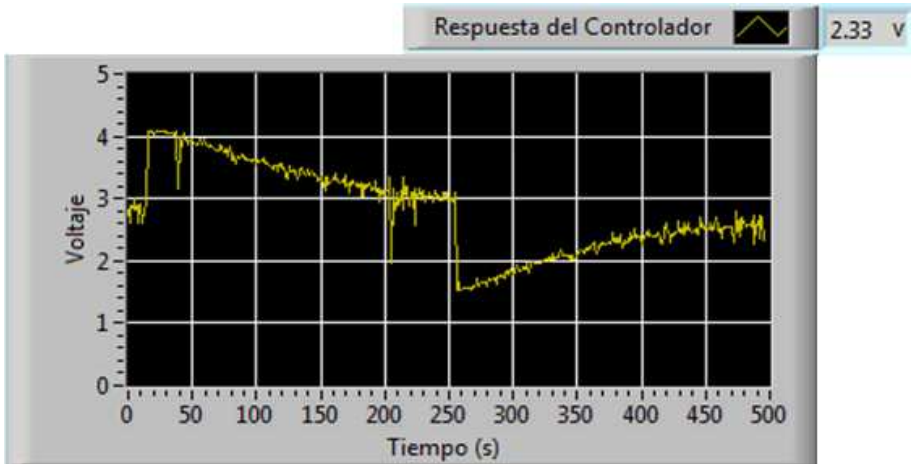


Figura 19. Respuesta del controlador difuso.

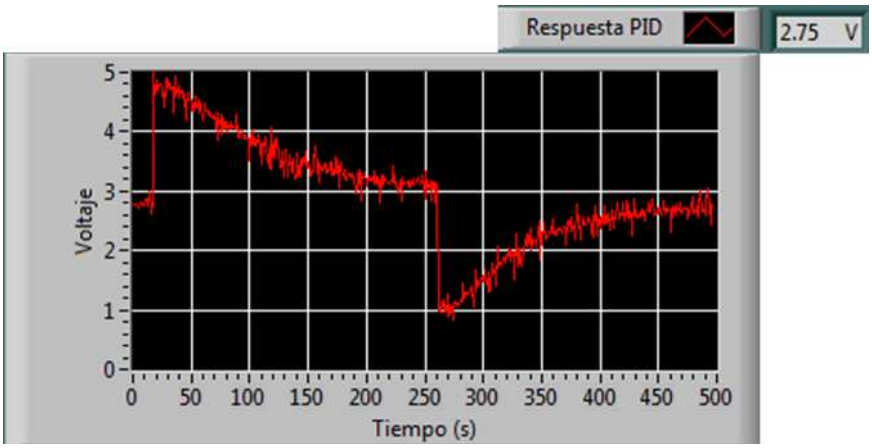


Figura 20. Respuesta del controlador clásico.

7. Conclusiones

En las pruebas realizadas con el controlador de lógica difusa y el controlador PID se puede determinar que los dos controladores presentan una respuesta similar en cuanto a la variable tiempo de establecimiento y tiempo de subida, sin embargo la señal de salida del controlador difuso presenta un menor pico de nivel porcentual, lo cual es deseable para evitar tener señales de control que puedan dañar los dispositivos actuadores presentes en el proceso por las variaciones bruscas. En cuando a los resultados obtenidos se puede concluir que ambos controladores son efectivos cuando se realiza una buena sintonización y un controlador difuso logra convertir el conocimiento experto en estrategias de control automático.

Como trabajo futuro se planea agregar al controlador difuso la parte integral y analizar las respuestas transitorias al aplicarlo en la estación didáctica de nivel ETLFIE I.

Referencias

[1] Schwab K. "*The Fourth Industrial Revolution"*, Crown Business, New York, 2017.
[2] Galindo F., Ruiz S. & Ruiz F. *"Competencias digitales ante la irrupción de la Cuarta Revolución Industrial"*, *Estudos em Comunicação* .1, 25, 1-11, 2017.
[3] Correa C. & Valencia J. *"Configuración de un Control de Temperatura en un Sistema Embebido de Bajo Costo, usando Herramientas de Inteligencia Artificial y el Internet de las Cosas", Revista Ibérica de Sistemas y Tecnologías de Información***.** 34, 78-93, 2019.
[4] Ponce P. *"Inteligencia Artificial con Aplicaciones a la Ingeniería"*. Alfaomega, México, 1ª ed., 2010.
[5] Ponce P., Ramírez F. *"Intelligent Control Systems with LabVIEW",* Springer, London, 2010.
[6] Luna D., Zenteno A., Santiago M., Romero Y., Pérez J., Rubín G. *"Metodología para controlar un robot móvil con lógica difusa"*, *Research in Computing Science*. 149, 11, 261-270 y 2020.
[7] Gómez J., Regino F., Espinel E. *"Diseño de sistema difuso para la detección y ubicación de fugas en tuberías horizontales"*, *Revista Ibérica de Sistemas e Tecnologias de Informação*. 25, 1, 277-290, 2019.
[8] Sifuentes-Mijares J., Camacho-Villalobos F., Lara-Molina M., Heredia-Mendoza K. *"Sistema de control difuso ball&plate mediante PID"*, *Revista Ciencia, Ingeniería y Desarrollo Tec Lerdo*. 1, 6, 81-87, 2020.
[9] Ogata, K. *"Ingeniería de control moderna",* Pearson Educación, España, 5ª ed., 2010.
[10] Anzurez J., Ramírez S., Torres V. *"Simulación y análisis de un control PID clásico y un PID difuso",* Reunión Internacional De Verano, RVPAI/2018, IEEE Sección México, Acapulco, Gro. México, 2018.

[11] Kumar S., Nagpal P. *"Comparative analyisis of P, PI, PID and fuzzy logic controller for tank water level control system"*, International Research Journal of Engineering and Technology (IRJET), 04, 04 , 1174-1176, 2017.

[12] Charre-Ibarra S., Gudiño-Lau J., Alcalá-Rodríguez J., Velez-Díaz D., y Díaz-Parra O. *"Control y monitoreo remoto de la estación de trabajo didáctica para la variable nivel (ETLFIE I)"*, *xikua*, 6, 11, 2018.

[13] Flores E., Reyna P., Jiménez J., Garrido J., Hernández Q. *"Diseño y construcción de prototipo de entrenamiento para prácticas en instrumentación y control"*, *Pistas Educativas*, 39, 128, 2018.

[14] National Instrumets. "*Labview*". Disponible en: https://www.ni.com/es-mx.html

Diseño Colaborativo en Mecatrónica, Capítulo 8, pp. 102 - 112
ISBN: 978-607-9394-23-3, 2021

Control de un robot de transporte personal Segway con retroalimentación aproximada del vector de estados.

González Pineda Carlos, Diaz Téllez Juan, García Ramírez Rubén Senén, Estévez Carreón Jaime, Pérez Pérez Jairo.

Instituto Tecnológico de Puebla
✉M19222685.62@puebla.tecnm.mx

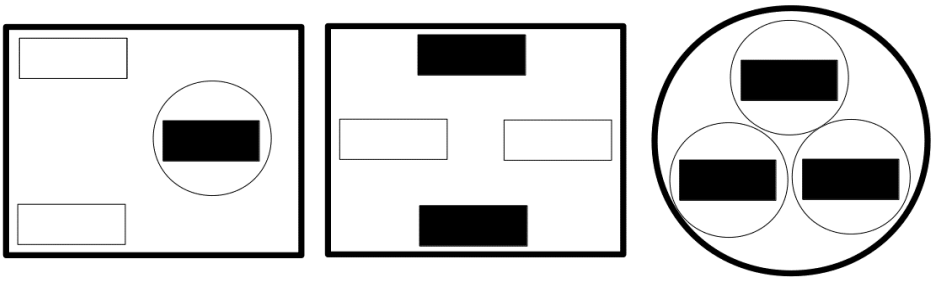
Resumen

este documento presenta el control de la posición de un robot segway a través de una retroalimentación del vector de estado. Demostraremos que utilizando este enfoque lineal simple el transporte de personal segway se puede estabilizar. La aproximación lineal se utiliza en el modelo no lineal para el diseño de la ley de control, ventajas como la fácil implementación en un sistema integrado y el cálculo del bajo costo en los procesadores ha motivado la elección de este enfoque. Para validar el diseño de la ley de control, se han tenido en cuenta tres parámetros: tiempo de estabilización, la capacidad de rechazar perturbaciones externas, y seguimiento de trayectoria.

Palabras clave: Segway, Modelo matemático, Algoritmo de control, Simulación.

1. Introducción

La robótica móvil es una de las ciencias que ha recibido atención en los ámbitos académico, social e industrial. Esto es debido en gran parte al desarrollo de los sistemas de computación integrados, sistema de sensores microelectromecánicos MEMS, dispositivos con sistemas operativos y de almacenamiento de alta energía. Los robots móviles tienen un amplio nicho de aplicaciones: la transferencia de objetos y su seguimiento, la evasión de obstáculos, la cooperación de entornos, análisis e inspección, exploración de áreas remotas por mencionar algunas. En particular, los robots móviles terrestres son el objetivo de investigación en este trabajo. Estos robots se dividen principalmente por los mecanismos que utilizan para moverse, generalmente los dividen en robots con ruedas, rastreados y con piernas. El caso más simple de robots móviles son los robots con ruedas. Los robots con ruedas comprenden una o más ruedas, pueden tener ruedas pasivas o ruedas opcionales y posiblemente ruedas dirigidas. La disposición de las ruedas da diferentes configuraciones de robots móviles terrestres. La figura 1, muestra las configuraciones más utilizadas, sin embargo, la lista no es exhaustiva, sincro-unidad, omnidireccional, balanceo móvil, robot de conducción, etc.



(a) Sincro-unidad (b) Balanceo móvil (c) Omnidireccional

Fig. 1: Configuración de ruedas más común.

Hoy en día, los robots de auto balanceo han ganado popularidad con la introducción del vehículo comercial Segway. Estos robots tienen la capacidad de equilibrarse en sus dos ruedas, haciéndolo un excelente transporte eléctrico personal. Debido a su gran maniobrabilidad, estos robots permiten que las personas sean transportadas en distancias cortas con una velocidad moderada, espacios confinados en interiores y exteriores, dar vueltas en su eje y en esquinas muy pronunciadas, además de pasar por pequeños Desniveles.

El equilibrio físico del robot es un péndulo invertido con dos motores accionados de forma independiente, que nos permiten su equilibrio, así como su conducción recta y giros [1]. Los robots de equilibrio tienen dos motores fijos independientes para conducir las ruedas en cada extremo, a diferencia de los robots diferenciales, el robot de equilibrio carece de ruedas pasivas, también la barra y la persona a bordo hace la estabilización del sistema más complicado [2]. En [3] se diseña el control LQR lineal, se implementa en un sistema integrado basado en FPGA. Los resultados muestran que las pruebas experimentales de sistemas montados integralmente son muy estables. En [4] un control LQR lineal y un ISMC son diseñados, el modelo matemático se basa en el péndulo con una plataforma móvil, la comparación entre ambos algoritmos se realiza en simulación. En [5] está diseñado un control basado en un PID geométrico. Este enfoque garantiza estabilidad exponencial a nivel local casi global del movimiento vertical del Segway. La eficacia de la ley de control se demuestra a través de simulaciones contra las incertidumbres.

En [6] es un prototipo que permite implementar diferentes tipos de controladores, así como cambiar las propiedades físicas del sistema para observar diferencias en la respuesta del sistema, y a partir de los resultados experimentales, poder confirmar la eficacia de los sistemas propuestos. En [7] se enfocaron en un vehículo totalmente operativo con un diseño compacto que se adapta a la absorción de impactos en terrenos e incluye un sistema de control mejorado. Incluye el uso de un amortiguador multidireccional que hace que el vehículo sea más eficaz en los terrenos ligeramente más ásperos. En [8] se incluye un controlador para los motores BDLC, el modelo del sistema y la implementación del control para su estabilización. Integra circuitos propios como el acelerómetro y el controlador para los motores. El análisis de estática y dinámica es en ANSYS. El diseño del control para los motores, el modelo matemático y el control PID los hicieron utilizando el famoso método de las raíces. En [9] se basó en controlar el vehículo Segway con incertidumbre del peso del conductor (con carga y sin carga). Con esta condición, se implementó un regulador cuadrático lineal (LQR) para controlar el vehículo ajustando los parámetros de ganancia del controlador. Se aplica un control robusto adaptativo para equilibrar el Segway haciendo variaciones de peso del conductor.

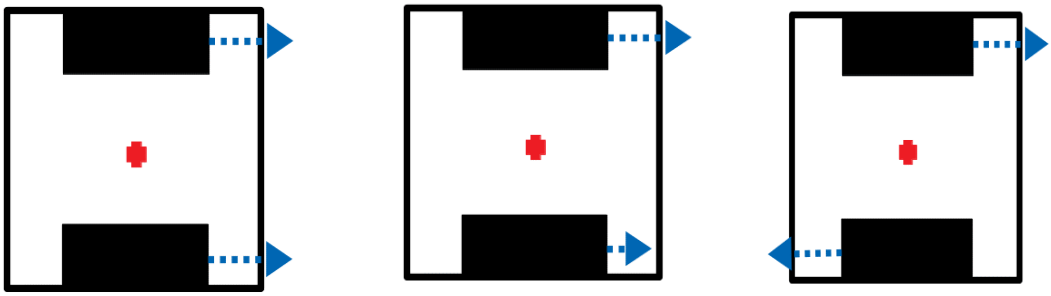
En [10] se presenta un esquema de control robusto. La presencia de entradas desconocidas, como es el par de carga en las flechas de los actuadores, la cual es variante con el tiempo, además de parámetros desconocidos del sistema y la falta de conocimiento de las fricciones estáticas y dinámicas que se adhieren al vehículo, sugiere el diseño de un controlador por rechazo activo de perturbaciones en combinación con un observador de estado extendido. Este control se calcula a través de la propiedad de planitud diferencial del sistema (controlabilidad). De ahí, la salida plana calculada linealiza al sistema entrada salida alrededor del punto de operación del vehículo, la cual se mide a través de un inclinómetro.

El documento se estructura de la siguiente manera. En la sección II algunos preliminares matemáticos utilizados a lo largo del documento son presentados, también la presentación del modelo del Robot Segway. En la sección III se presenta el diseño de la ley de control basado en la retroalimentación de estado. En la sección IV los resultados de la simulación están presentes y muestran la eficacia del algoritmo propuesto. En la sección V se presentan algunas conclusiones y trabajos futuros.

2. Modelo matemático

2.1 Modelo del balanceo del robot

El modelo del robot segway que presentamos se basa en el principio del péndulo invertido. La dinámica del péndulo invertido ha sido la base de los robots bípedos, hélices espaciales y control de actitud de pequeños satélites. La figura 2 muestra los movimientos básicos del robot, conduciendo movimientos hacia atrás, conduciendo hacia los lados el movimiento y girando en su propio eje. El movimiento hacia atrás se logra cuando la velocidad de ambos motores se incrementa o disminuye en la misma cantidad. El movimiento lateral se alcanza cuando se aumenta la velocidad del motor izquierdo, mientras que la velocidad de la derecha se reduce y viceversa. El girar en su propio eje se logra cuando la velocidad de ambos motores se incrementa en la dirección opuesta.



(a) hacia adelante y hacia atrás. (b) hacia los lados (c) movimiento sobre su propio eje.

Fig. 2 Movimiento del robot.

El modelo no lineal del péndulo invertido puede ser modelado como:

$$(m + M)\ddot{x} + (ml\cos\Theta)\ddot{\Theta} = -C_1\dot{x} + (ml\sin\Theta)\ddot{\Theta} + u \quad (1)$$

$$(ml\cos\Theta)\ddot{x} + (J + ml^2)\ddot{\Theta} = -C_2\dot{\Theta} + (mgl\sin\Theta) \quad (2)$$

donde las variables de estado se definen como. X es el desplazamiento del robot, $\dot{x}$ la velocidad, así como el desplazamiento angular de la barra y la persona es Θ y su velocidad angular es $\dot{\Theta}$. La variable u representa la fuerza que empuja el sistema y que es la fuerza por controlar, por lo tanto, estas variables son las señales de los motores. Las constantes J es el momento de inercia con respecto al centro de gravedad del péndulo, l representa la longitud de la barra hasta el centro de masa, M es la masa de la base del Segway, m es la masa de la barra y de la persona, y g representa la gravedad. Las constantes C_1 y C_2 representan los coeficientes de fricción del movimiento rotacional del péndulo y del movimiento lineal de la base. Para el modelo matemático del control, algunas restricciones se han considerado:

- El centro de gravedad de la barra está en su centro geométrico.
- Los coeficientes de fricción del movimiento rotatorio del péndulo y el movimiento lineal de la base son insignificantes.
- El desplazamiento angular Θ es muy pequeño.

El modelo del segway para el diseño del algoritmo de control es:

$$\ddot{x} = \frac{-m^2l^2g\theta + (J + ml^2)u}{\Delta} \quad (3)$$

$$\ddot{\Theta} = \frac{(M+m)(mgl\theta) - mlu}{\Delta} \quad (4)$$

donde $\Delta = (M+m)(J+ml^2) - m^2l^2$. Seleccionando como variables de estado $x_1 = x,\ x_2 = \dot{x},\ x_3 = \theta\ y\ x_4 = \dot{\theta}$. y Considerando el punto de equilibrio de este sistema la posición vertical hacia arriba:

$$[x_1 \quad x_2 \quad x_3 \quad x_4]^T = [x_d \quad 0 \quad 0 \quad 0]^T \tag{5}$$

además, u = 0

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{(J+ml^2)u - m^2l^2gx_3}{\Delta} \\ x_4 \\ \frac{-mlu + (M+m)mglx_3}{\Delta} \end{bmatrix} \tag{6}$$

La linealización aproximada del sistema en el punto de equilibrio es:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-m^2l^2g}{\Delta} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(M+m)mlg}{\Delta} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{J+ml^2}{\Delta} \\ 0 \\ \frac{-ml}{\Delta} \end{bmatrix} u \tag{7}$$

2.2 Modelo del propulsor

El sistema incluye un motor de CD con codificador incremental y su controlador o puente H. En primer lugar, dado un voltaje constante por la batería LIPO, el controlador genera una señal de voltaje promedio que es una función de la modulación de ancho de pulso o señal PWM, lo que hará que el motor alcance una velocidad de estado estable. Esto se puede modelar como:

$$\omega_{ss} = C_R\sigma + \omega_b \tag{8}$$

donde C_R es una constante que depende de la tensión de la batería y el controlador, σ es una señal PWM con un valor entre 1000 y 2000 ciclos de trabajo, y w_b es la velocidad angular que el motor alcanza una vez que el motor se inicializa. Debido a que el motor necesita algún tiempo para lograr la velocidad del estado Ω_{ss}, se genera una segunda ecuación para modelar la respuesta dinámica. Generalmente la dinámica de un motor CD se puede simplificar como una ecuación de primer orden.

$$\omega = \frac{1}{T_m s+1}\omega_{ss} \tag{9}$$

Combinando las ecuaciones 8 y 9 obtenemos el Modelo completo del propulsor

$$\omega = \frac{1}{T_m s+1}(C_R\sigma + \omega_b) \tag{10}$$

3. Algoritmo de control

3.1 Control completo de la retroalimentación del estado

La estrategia por seguir es diseñar una ley de control lineal basado en la retroalimentación de estado para el sistema lineal ecuación (7), por lo que se estabilizan las variables de estado a cero. Una vez que el sistema lineal se ha estabilizado, la ley de control lineal se aplica al sistema no lineal ecuación (1) y ecuación (2). Esto garantizaría que el sistema está operando en su punto de equilibrio.

Verificamos que el sistema lineal sea completamente controlable:

$$\det(C) = \det([B: AB: A^2B: A^3B]) \neq 0 \tag{11}$$

Proponemos la siguiente ley de control:

$$u = -k_1x_1 - k_2x_2 - k_3x_3 - k_4x_4 \tag{12}$$

Se propone que el sistema linealizado en lazo cerrado tenga sus polos en las raíces del siguiente polinomio deseado:

$$p_d(s) = (s + \alpha)^2(s^2 + 2\zeta\omega_n s + \omega_n^2) \tag{13}$$

Donde ζ es el factor de amortiguación y ω_n es la frecuencia natural sin amortiguar correspondiente al par de polos complejos conjugados que generan el factor de segundo grado.

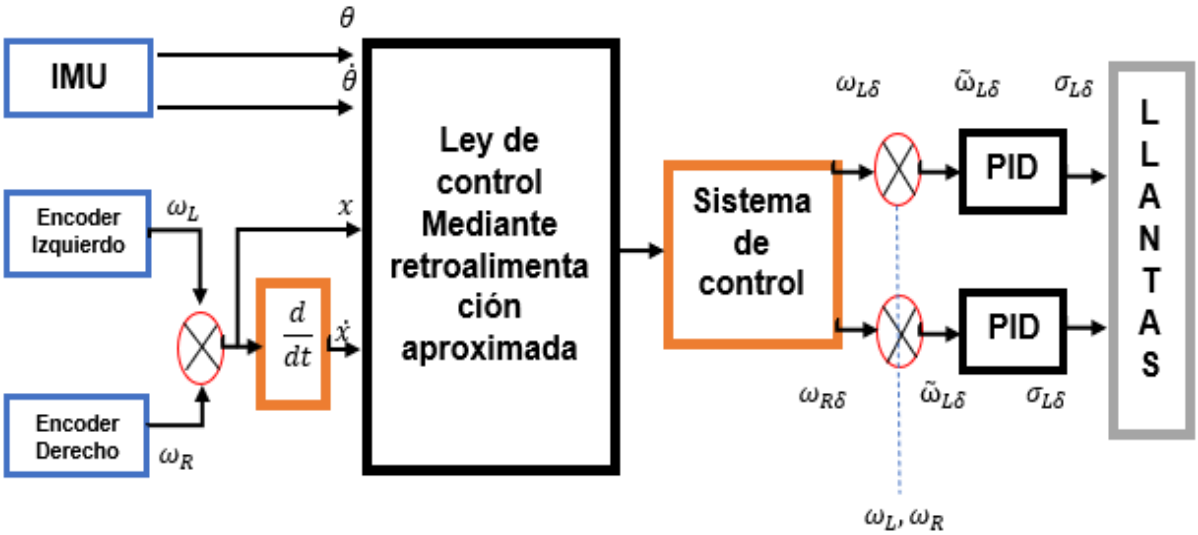


Fig. 3 Diagrama de bloques del sistema de control Segway.

3.2 Control del Motor

Se requiere un control eficiente de la velocidad del motor de CD para el Robot Segway. Una respuesta rápida y suave producirá alto rendimiento en el control de posición. Asumiendo que la velocidad angular del motor de CD es ω, además, estos motores utilizan pulsos de alta frecuencia con modulación (PWM) para controlar la tensión del motor. El objetivo del control es diseñar en cada motor un comando de señal PWM, tal que $lim_{t\to\infty}|\omega_d(t) - \omega(t)| = 0$ Proponemos un control PID para minimizar la señal de error $\tilde{\omega} = \omega_d(t) - \omega(t)$;

$$\sigma = k_p\tilde{\omega} + k_i \int \tilde{\omega} + k_d\tilde{\omega} \tag{14}$$

La velocidad angular de los motores ω puede obtenerse de los codificadores, C_R y ω_b se obtienen a través de la experimentación.

Para sintonizar el PID se necesita la función de transferencia de los motores de CD, es decir su modelo matemático. Esta puede venir en la hoja de datos si no es el caso se tendría que calcular con pruebas experimentales como por ejemplo usando un dinamómetro. Una vez obteniendo de forma

experimental nuestro modelo matemático, las constantes Kp, Ki y Kd se determinan de acuerdo con las siguientes ecuaciones de Zieglev y Nichols.

$$Kp = \frac{1.2\tau}{k*t_m} \tag{15}$$

$$Ki = 2t_m \tag{16}$$

$$Kd = 0.5t_m \tag{17}$$

Donde t_m es el tiempo muerto, τ la constante de tiempo del sistema, y k la ganancia.

Si contamos con la función de transferencia de la hoja de datos del motor proporcionada por el fabricante se ingresa en la ventana de comandos de Matlab y con la herramienta PIDtool se importa para poder así sintonizar nuestra señal con la forma de respuesta deseada para nuestro sistema.

$$G(S) = \frac{0.9841*e^{-0.152s}}{0.316s+1} = \frac{K_0*e^{-s\tau_0}}{(1+s\tau_0)} \tag{18}$$

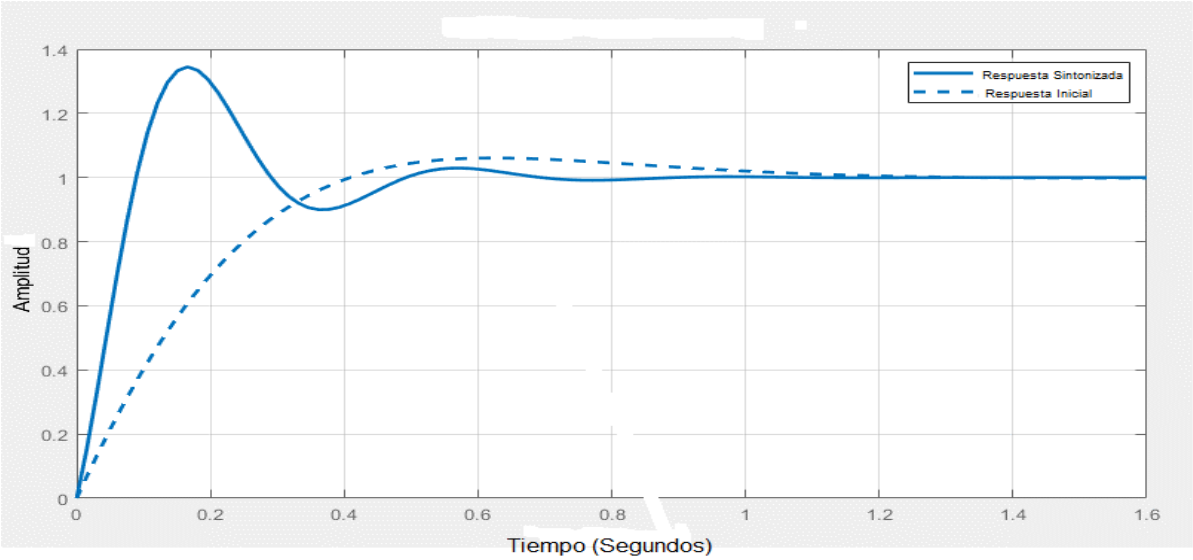


Fig. 4 sintonización de Kp, Ki y Kd del PID en el pidTuner del Toolbox de Matlab.

4. Simulación

La figura 3 muestra el diagrama de bloques del sistema. El sistema tiene dos sensores principales, una Unidad de Medición Inercial (IMU) y un codificador en cada motor BLDC. El sensor IMU estima el ángulo θ así como la velocidad angular $\dot{\theta}$, utilizando un filtro Kalman incorporado. Los codificadores proporcionan información sobre la velocidad angular de la rueda, con la velocidad angular podemos calcular la posición x y la velocidad de $\dot{x}$. Las cuatro variables de estado se retroalimentan al algoritmo de control y es calculada u. El bloque control permite a la señal de control que se convierta en la velocidad angular requerida $\omega_{L\delta}$ para los motores. Por último, es necesario un PID para garantizar la velocidad angular deseada. La elección de ω_n y ζ está hecha para establecer el tiempo de respuesta del sistema. Típicamente ζ se elige como 0.707, con el fin de lograr una respuesta amortiguada, ω_n se elige como $12\frac{rad}{s}$, Las figuras muestran la evolución de las variables de estado del sistema no lineal controlado por la ley de retroalimentación basada en la linealización aproximada en diferentes condiciones. Los parámetros de la simulación del sistema son:

Los valores del vector $K = [k_1 \quad k_2 \quad k_3 \quad k_4]^T$, se obtienen resolviendo el sistema de ecuaciones, que se obtiene mediante la igualación de los coeficientes del término polinómico deseado (ecuación 13).

TABLA I: Parámetros para la simulación

M	Masa de la base	5.2
m	Masa del péndulo	1.9
l	Longitud del péndulo	1.1
J	Momento de inercia	0.7663
g	Gravedad	9.81

$$k_1 = -0.82197167229884664270826534799282 \tag{19}$$

$$k_2 = -3.2878866891953865708330613919713 \tag{20}$$

$$k_3 = -119.18573845782571663159363111853 \tag{21}$$

$$k_4 = -36.868111726020750890932127809401 \tag{22}$$

En las figuras 5, 6, 7 y 8 tratamos de reproducir el comando del avance hacia adelante y hacia atrás con la inclinación de la barra en un $\theta = 35°$, es decir, que se proponen $x_d = [0 \quad 0 \quad 35 \quad 0]^T$ como punto de equilibrio para las siguientes condiciones iniciales $x(0) = [0 \quad 0 \quad 0 \quad 0]^T$ Como podemos ver en las imágenes al inclinarse la barra por un ángulo θ, la base intenta compensar la barra a su posición vertical, por lo tanto, su movimiento es hacia adelante.

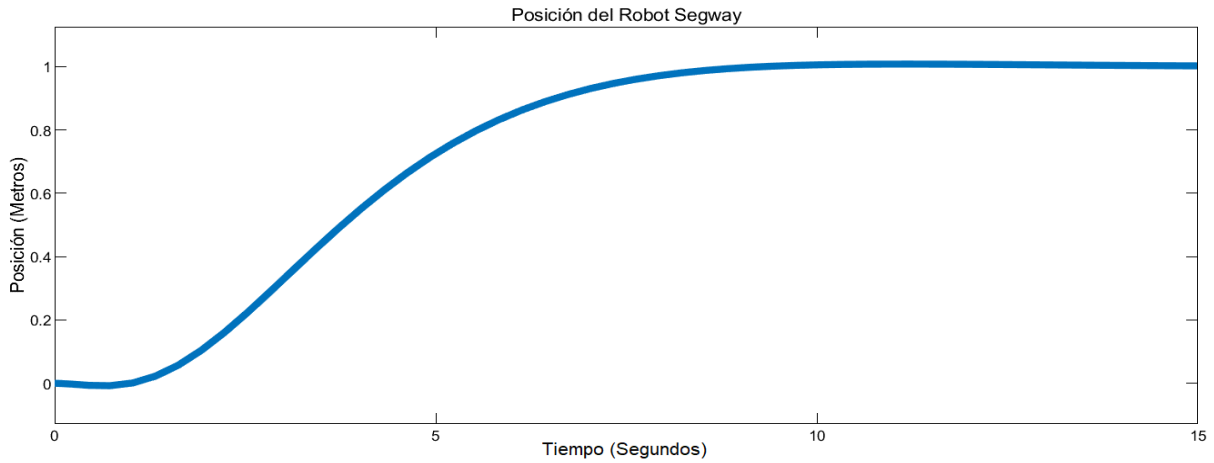


Fig. 5 Movimiento adelante-atrás, evolución de x.

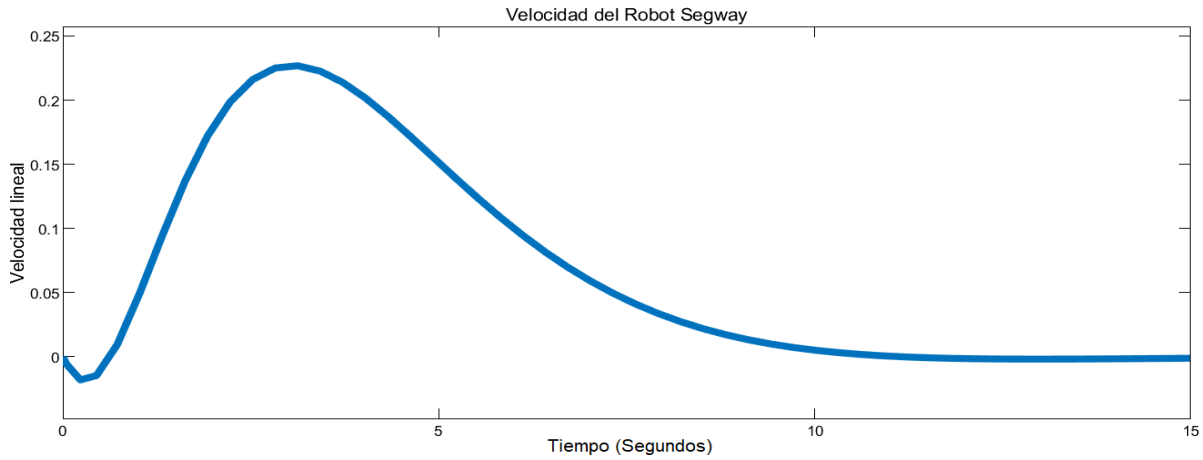


Fig. 6 Movimiento adelante-atrás, evolución de $\dot{x}$

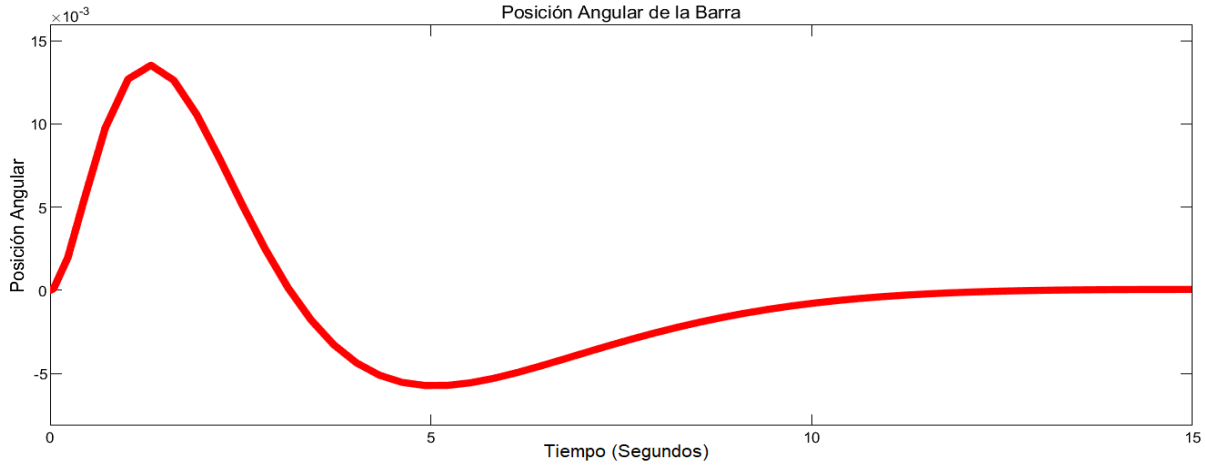


Fig. 7 Movimiento adelante-atrás, evolución de θ.

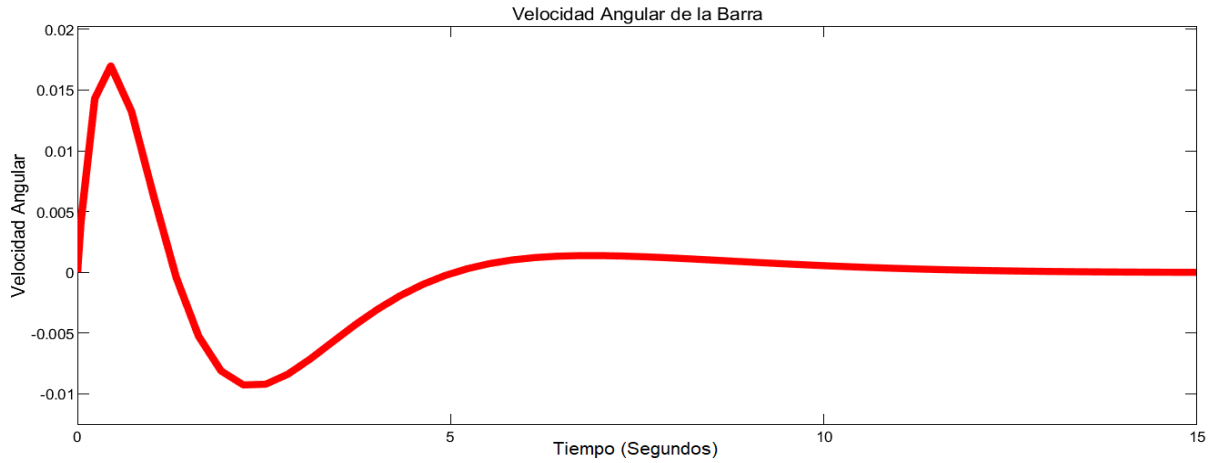


Fig. 8 Movimiento adelante-atrás, evolución de $\dot{\theta}$.

Las figuras 9, 10, 11 y 12 muestran un movimiento flotante mientras la barra del robot está perturbada. Las condiciones iniciales del sistema son $x(0) = [0 \quad 0 \quad 35 \quad 0]^T$.

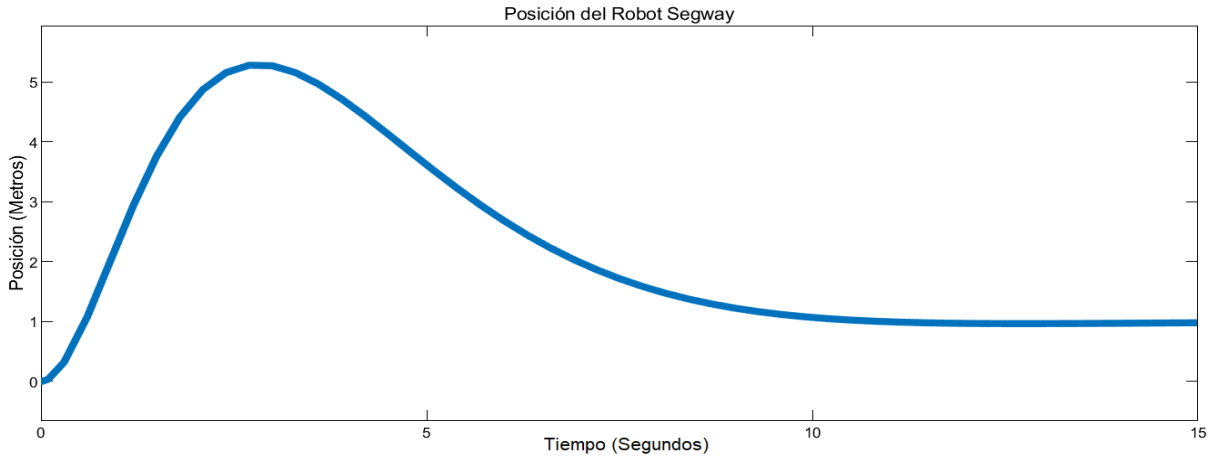


Fig. 9 Movimiento flotante, evolución de x.

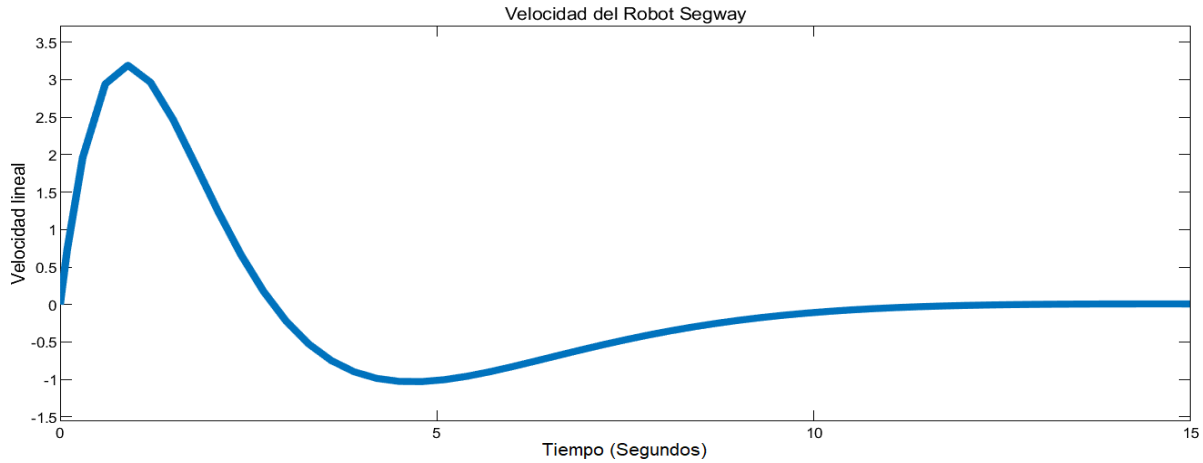


Fig. 10 Movimiento flotante, evolución de $\dot{x}$.

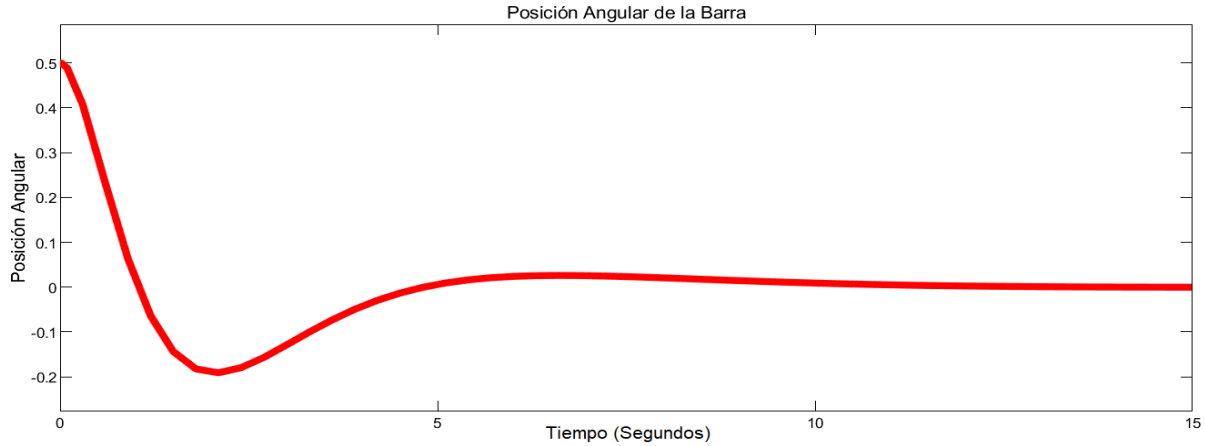


Fig. 11 Movimiento flotante, evolución de θ.

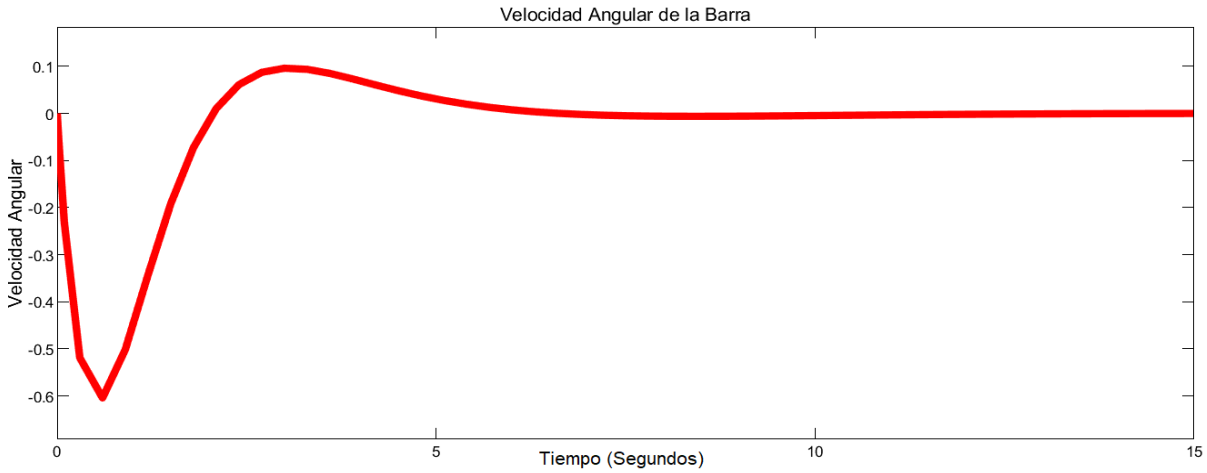


Fig. 12 Movimiento flotante, evolución de $\dot{\theta}$.

La figura 11 muestra la posición de la barra con una condición inicial de 35˚, se puede ver cómo la barra se estabiliza en cero debido al movimiento de la base ver figura 9. El tiempo de estabilización es de aproximadamente 8 segundos, lo que es un excelente tiempo para estos sistemas.

5. Conclusiones

Cabe destacar que el sistema linealizado basado en el sistema de diseño aplicado al sistema no lineal al menos alrededor de su punto de equilibrio no se puede estabilizar el sistema si las perturbaciones se alejan del punto de equilibrio, sin embargo, debido a la naturaleza del robot Segway (pequeños ángulos) la ley de control lineal propuesta por la retroalimentación de estado es lo suficientemente estable para la estabilidad de dicho robot. Es necesario también aclarar que este documento fue realizado pensando solo en la simulación de dicho robot, con expectativas de más adelante diseñarlo, construirlo e implementarlo. El diseño, la instrumentación y la implementación del sistema integrado representará un gran desafío, un microcontrolador Arduino es propuesto para el sistema de control integrado.

Referencias

[1] Braunl, T. (2006). *Mobile Robot Design and Applications with Embedded Systems,* Second Edition. Embedded Robotics: 10.1007/3-540-34319-9

[2] Sira-Ramírez, Hebertt and Márquez, Richard and Rivas, Francklin and Santiago, Orestes (2018), *Linealización aproximada, extendida, exacta:* Control de Sistemas No Lineales: 8420544493, 9788420544496

[3] Pinto L. J. and Kim D. and Lee J. Y. and Han C, *Development of a Segway robot for an intelligent transport system,* 2012 IEEE/SICE International Symposium on System Integration (SII)

[4] Shilpa B. Indu V. and Rajasree S. R. *Design of an underactuated self-balancing robot using linear quadratic regulator and integral sliding mode controller,* 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)

[5] Basnayake I. D. Madhushani T. W. U. and Maithripala D. H. S. *Intrinsic PID controller for a Segway type mobile robot,* 2017 IEEE International Conference on Industrial and Information Systems (ICIIS)

[6] Solís J. Nakadate R. Yoshimura Y. Yuichiro H. Atsuo T. *Development of the two-wheeled inverted pendulum type mobile robot WV-2R for educational purposes*, School of Creative Science and Engineering

[7] Muhammad U. D, Mohsin S. A, Maryam M. Umair E. Umer I. *Segway Electric Vehicle*, 2012 International Conference on Robotics and Artificial Intelligence (ICRAI 2012)

[8] Muhammad H K. Mehak C. Taimoor T. Qurat-ul-Ain F. Umer I. *Fabrication and Modelling of Segway*, International Conference on Mechatronics and Automation

[9] Chantarachit S. *Development and Control Segway by LQR adjustable Gain*, 2019 International Conference on Information and Communications Technology (ICOIACT)

[10] Curiel O G. *Diseño y construcción del sistema de tracción eléctrica basado en motores de CD sin escobillas para un vehículo de dos ruedas,* Universidad Tecnológica de la Mixteca

Diseño Colaborativo en Mecatrónica, Capítulo 9, pp. 113 - 125
ISBN: 978-607-9394-23-3, 2021

Diseño de un sistema inteligente para la recomendación de terapias de rehabilitación de fractura de tobillo

Cotri-Melecez Berenice[1], Quiroz-Compean Griselda[2], Torres-Treviño Luis[2], Rodriguez-Liñan Angel[2].

[1]Escuela de Ingenierías y Tecnologías
Universidad de Monterrey
Nuevo León, México.
[2]Universidad Autónoma de Nuevo León, Facultad de Ingeniería Mecánica y Eléctrica
Avenida Universidad S/N, Cd. Universitaria, San Nicolás de los Garza, N.L., CP 66455, México.
griselda.quirozcm@uanl.edu.mx

Resumen

Este artículo presenta el diseño de un sistema inteligente para la recomendación automática de terapias de rehabilitación de fracturas de tobillo basado en un sistema de lógica difusa (SLD). El diseño del SLD considera la experiencia de especialistas en rehabilitación física y las condiciones clínicas del paciente, mediante la propuesta de reglas difusas y el mecanismo de inferencia de Mamdani aplicado a sistemas difusos. El desempeño del SLD se probó con datos de 11 expedientes clínicos de pacientes que sufrieron fractura de tobillo. Los resultados obtenidos indican que el sistema es capaz de proporcionar recomendaciones de forma automática sobre la terapia de rehabilitación física, los cuales pueden ser de utilidad en esquemas de optimización de recursos, tanto técnicos como humanos, requeridos en programas de rehabilitación de fractura de tobillo.

Palabras clave: Sistemas inteligentes, lógica difusa, rehabilitación física, fractura de tobillo.

1. Introducción

La fractura de tobillo es una de las lesiones de miembro inferior con la mayor tasa de atención en el área de emergencias de los hospitales [1]. Esta fractura representa el 4% de los casos generales de fracturas [2] y más del 20% de fracturas de extremidades inferiores [3]. Se ha reportado que el 9% de la carga de un especialista en traumatología se debe a los tratamientos de fracturas de tobillo [1]. Después de que una persona ha sufrido una fractura de tobillo y ha sido tratada, la indicación es realizar un programa de rehabilitación física (PRF) para su recuperación [4]. Los PRFs convencionales están formados por un conjunto de sesiones de ejecución de terapias de rehabilitación física, que pueden ser de dos tipos: ejercicios de movilidad y medios físicos (ver Figura 1). Las recomendaciones que se aplican al paciente en cada sesión son estandarizadas. En la práctica clínica, los especialistas recomiendan realizar todas las terapias en su conjunto a lo largo de todo el tiempo de duración del PRF; es decir, no hay una selección de terapias en función del avance del paciente en la recuperación de sus capacidades biomecánicas.

Lo anterior, genera la pregunta de si existen algunas terapias que pueden ser omitidas a lo largo del PRF en función de mediciones de las capacidades biomecánicas del paciente y del alcance de los objetivos de la terapia. Por tal motivo, en este trabajo se propone el diseño de un sistema inteligente para la recomendación automatizada de las terapias de rehabilitación, considerando la información clínica del paciente, incluida la medición de variables biomecánicas. Esto implicaría que a lo largo del PRF se fueran recomendando sólo las terapias requeridas para avanzar en los objetivos y no el conjunto completo, tal como se realiza en la práctica clínica actual.

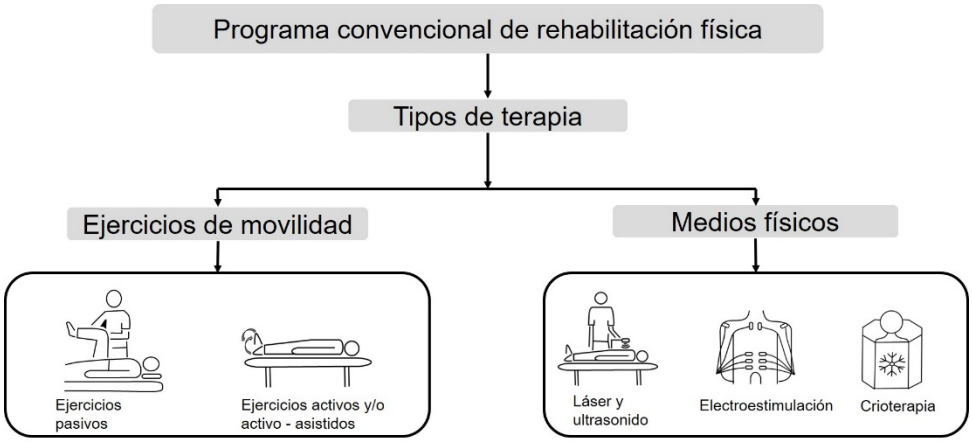


Figura 1. Los programas de rehabilitación física (PRFs) convencionales incluyen dos tipos de terapias: ejercicios de movilidad y medios físicos.

Por la naturaleza de las variables relacionadas en el proceso de recomendación de terapias de rehabilitación, se propone usar la plataforma de la lógica difusa [5,6] para el diseño de un sistema inteligente que tome en cuenta las condiciones del paciente y genere la recomendación de terapias personalizadas en función de ello [7]. En adelante, se hará referencia al sistema propuesto como sistema lógico difuso (SLD). Este sistema busca apoyar la programación y buen uso de los recursos utilizados en los PRFs, mediante la reducción del número de terapias requeridas en las sesiones de rehabilitación física [3], ya que el SLD solo recomendará las terapias para la recuperación de las condiciones clínicas y/o capacidades biomecánicas afectadas del paciente.

En las últimas décadas, los sistemas inteligentes se han utilizado en diferentes áreas, y en el campo de la medicina se han propuesto algunos para el diagnóstico de diferentes patologías. En el caso de patologías del sistema musculoesquelético, Maftouni *et al*. propusieron el diseño de un sistema inteligente basado en un SLD, para el diagnóstico de espondilitis anquilosante [8]. Por su parte, Amiri y Khadivar reportaron un sistema inteligente, basado en el método Delphi difuso con metodología híbrida, para el diagnóstico y tratamiento de trastornos musculoesqueléticos en la articulación de muñeca [9].

Sobre sistemas inteligentes diseñados para el diagnóstico de otras patologías, Polat y Günes reportaron el diseño de un sistema experto, basado en inferencia neuro-difusa adaptativa para el diagnóstico de diabetes [10]. Kelec *et al*. reportaron el diseño de un sistema experto difuso, aplicando un método neuro-difuso para el diagnóstico de cáncer de mama [11]. Muthukaruppan y Er diseñaron un sistema difuso experto, usando el método de optimización de enjambre de partículas para el diagnóstico de enfermedad coronaria [12]. Asl y Zarandi reportaron el diseño de un sistema experto difuso para el diagnóstico de leucemia, usando el mecanismo de inferencia de Mamdani [13]. Sobre esta misma enfermedad, Hosseini *et al*. reportaron un sistema experto difuso y un algoritmo genético, para el diagnóstico de leucemia linfocítica aguda en niños [14]. Biyouki *et al*. propusieron un sistema de clasificación neuro-difusa, con algoritmo *K-means* y algoritmo gradiente para el diagnóstico de hipotiroidismo [15].

Considerando estos antecedentes, el objetivo de este trabajo es presentar el diseño de un sistema inteligente basado en lógica difusa, para la recomendación automatizada del PRF de la fractura de tobillo. Como se ve en los antecedentes de la aplicación de sistemas inteligentes en el campo de la medicina, existen pocos trabajos enfocados al diagnóstico de patologías del sistema musculoesquelético; más aún, hasta el conocimiento de los autores no se ha reportado la automatización de la recomendación de las terapias para la rehabilitación de la fractura de tobillo, por lo que este trabajo busca contribuir en este sentido. El resto del documento está organizado de la siguiente manera: en el punto 2 se presentan los métodos utilizados para el diseño del SLD, en el punto 3 se muestran los experimentos y se reportan los resultados, posteriormente en el punto 4 se presenta

la discusión acerca de los resultados obtenidos, y, por último, en el punto 5 se presentan las conclusiones de este trabajo.

2. Metodología

El diseño del recomendador se divide en tres etapas. La primera consiste en el análisis de datos de expedientes clínicos de personas que sufrieron fractura de tobillo. El objetivo de esta etapa es definir las variables de entrada y salida del sistema inteligente SLD. En la segunda etapa se definen las reglas difusas que relacionan las variables de entrada y variables de salida del sistema, de acuerdo con las recomendaciones proporcionadas por los especialistas en rehabilitación física y las condiciones clínicas de los pacientes. Finalmente, la tercera etapa consiste en la implementación numérica del SLD. Estas tres etapas se describen en las siguientes subsecciones. Esta metodología se ilustra en la Figura 2.

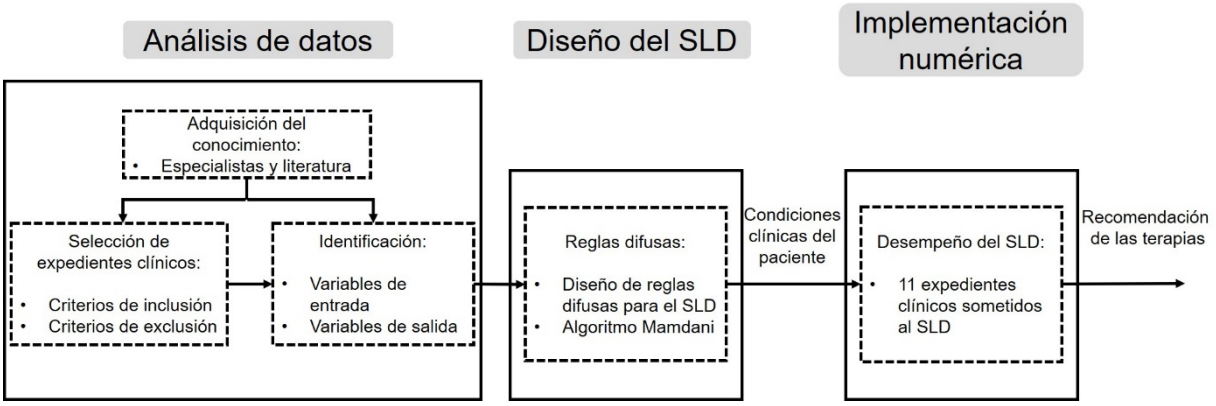


Figura 2. Diagrama de la metodología usada en el diseño e implementación del sistema lógico difuso (SLD) para la recomendación de terapias de rehabilitación de fractura de tobillo.

2.1 Análisis de datos

La base de datos utilizada en este trabajo está formada por los expedientes clínicos de 11 pacientes que sufrieron fractura de tobillo (entre 2017 y 2019), y a quienes se les recomendó llevar un PRF en el Departamento de Medicina del Deporte y Rehabilitación Física del Hospital Universitario, Dr. José Eluterio González de la Universidad Autónoma de Nuevo León, en la ciudad de Monterrey, N.L., México. El análisis de expedientes se realizó en colaboración con médicos especialistas en rehabilitación, iniciando con la selección de expedientes e identificación de variables y sus valores. Primeramente, se definieron los criterios de inclusión y exclusión. Los criterios de inclusión de los expedientes fueron los siguientes: expedientes clínicos de personas adultas que sufrieron fractura de tobillo, sexo indistinto y que hayan cumplido con el PRF indicado; es decir, que hayan sido dadas de alta. Los criterios de exclusión fueron: pacientes menores de edad y pacientes con prótesis.

Los expedientes clínicos analizados tienen dos secciones de registro de datos, el registro preoperatorio y el posoperatorio. El primero contiene los datos de registro de ingreso del paciente y la nota inicial de emergencias; mientras que el segundo contiene los registros de datos del departamento de traumatología y del centro de rehabilitación. En estos últimos departamentos se realiza la exploración física del paciente, se registra la información de las condiciones clínicas y las capacidades biomecánicas del tobillo tratado y se valora al paciente para la recomendación del PRF.

Las condiciones clínicas y capacidades biomecánicas del paciente son estimadas y medidas en los departamentos de traumatología y en el centro de rehabilitación física. En la práctica clínica los especialistas en rehabilitación consideran estas estimaciones y mediciones para indicar la recomendación de las terapias de rehabilitación del tobillo [16]. Las condiciones clínicas son variables cualitativas que se estiman de acuerdo con escalas estandarizadas, las cuales se describen a

continuación. La condición del dolor se estima usando la escala visual análoga (EVA) que toma valores entre 0 y 10, donde 0 indica ausencia de dolor, 1 a 3 indica dolor leve, 4 a 7 indica dolor moderado y 8 a10 indica un dolor fuerte. La condición de edema se registra de acuerdo con la técnica del godete, con una escala de tres valores indicada por una equis (x), dos equis (xx) o tres equis (xxx), donde x representa edema leve, xx edema moderado y xxx edema severo.

La condición de limitación de movimiento se registra indicando si existe o no la condición. Los cambios tróficos en el tobillo del paciente se registran indicando si existe o no la condición. La condición de contractura se registra indicando si existe o no esta condición en el tobillo del paciente. La condición de la herida (cicatriz) se registra indicando si está adherida a los planos profundos o no está adherida. El tipo de marcha se registra considerando dos tipos de patrón de marcha, el patrón de la marcha no deseada llamada marcha disbásica y el patrón de la marcha deseada llamada marcha eubásica. La condición de fuerza se registra en la escala de Lovett y Daniels, la cual toma valores entre 0 y 5, registrando 0 para una fuerza nula, 1 para una fuerza pobre, 2 para una fuerza deficiente, 3 para una fuerza regular, 4 para una buena fuerza y 5 para una fuerza normal.

La condición de sensibilidad se registra indicando si ésta se encuentra alterada o no alterada. La estabilidad se registra indicando si existe o no esta condición. La consolidación ósea se registra indicando si existe o no esta condición. Finalmente, las capacidades biomecánicas se registran midiendo los rangos óptimos de movimiento (ROMs) de la articulación afectada; estos son variables cuantitativas que se miden en grados (°) y para los movimientos del tobillo se registran cuatro ROMs: flexión dorsal de 10° a 20°, flexión plantar de 20° a 45°, inversión de 10° a 35°
y eversión de 10° a 25°. Estas condiciones clínicas y capacidades biomecánicas descritas están resumidas en la Tabla 1.

Después de que las condiciones clínicas y capacidades biomecánicas del paciente han sido estimadas y/o medidas, registradas y valoradas por el especialista, y si éstas no son las deseadas, se le indica al paciente asistir al centro de rehabilitación física, en donde se le dará la recomendación de llevar a cabo un PRF para cumplir con los objetivos de rehabilitación indicados por el especialista. En cada cita del paciente al centro de rehabilitación, se le proporciona un conjunto de recomendaciones, las cuales consisten en terapias estandarizadas que debe ejecutar de acuerdo con el número de series indicado por el especialista (entre 10 y 15 series, en cada sesión realizada cada día).

El conjunto de recomendaciones se puede dividir en dos tipos: medios físicos y ejercicios de movilidad. Los medios físicos incluyen ultrasonido en profundidad superficial (US) de 3 MHz durante 5 minutos. La terapia interferencial se recomienda para tres condiciones: edema (TIE), dolor (TID) y consolidación (TIC), los rangos de aplicación para cada caso son, 0 a 200 Hz, 80 a 150 Hz y 100 Hz respectivamente durante 15 minutos. La terapia de cambios térmicos (HYC) que consiste en la aplicación diaria de frío y calor alternadamente por 15 minutos. La ejecución de ejercicios de propiocepción (P) una sesión diaria de 10 repeticiones. La ejecución de ejercicios de movilidad (EM) en flexión dorsal, flexión plantar, inversión y eversión, una sesión diaria de 10 repeticiones. La recomendación de masaje antiedema (MAE) aplicado por un tiempo de 5 minutos, uno diario. La recomendación de masaje antiadherente (MH) para la cicatriz aplicado por un tiempo de 5 minutos, uno diario. La ejecución de ejercicios isométricos (EIM) de 8 a 10 repeticiones, una diaria. La ejecución de ejercicios isotónicos (EIT) de 8 a 10 repeticiones, una diaria. La aplicación de masaje descontracturante (MD) por 5 minutos, una diaria. El ejercicio de caminata para la reeducación de la marcha (RM) de 10 a 15 vueltas en una pista de 3 metros, uno diario. Finalmente, la aplicación de ejercicios de estiramiento (EE) por 5 minutos, una sesión diaria. El conjunto completo de terapias se resume en la Tabla 2.

2.2 Diseño del SLD

La lógica difusa surge de la necesidad por comprender la forma en la que la mente humana realiza el proceso de la toma de decisiones, establece modelos matemáticos y sistemas, emulando lo que el ser humano hace de forma natural por experiencia o conocimiento [6]. La incertidumbre es un factor que está presente en la mayor parte de las situaciones más complejas, la lógica difusa se utiliza para la solución de problemas con alto grado de incertidumbre, para el manejo de variables lingüísticas y consi-

Tabla 1. Definición de las condiciones clínicas y capacidades biomecánicas que definen el estado de un paciente con fractura de tobillo.

Variables de entrada			
Descripción	**Tipo**	**Escala**	**Valores**
Dolor	Cualitativa	0 1 a 3 4 a 7 8 a 10	Sin dolor Leve Moderado Severo
Edema	Cualitativa	x xx xxx	Leve Moderado Severo
Limitación De movimiento	Cualitativa	Si No	
Cambios tróficos	Cualitativa	Si No	
Contractura	Cualitativa	Si No	
Herida (cicatriz)	Cualitativa	Adherida No adherida	
Marcha	Cualitativa	Disbásica Eubásica	
Fuerza	Cualitativa	0 1 2 3 4 5	Nula Pobre Deficiente Regular Buena Normal
Sensibilidad	Cualitativa	Alterada No alterada	
Estabilidad	Cualitativa	Si No	
Consolidación	Cualitativa	Si No	
ROMs	Cuantitativa	Flexión dorsal Flexión plantar Inversión Eversión	10° a 20° 20° a 45° 10° a 35° 10° a 25°

dera el conocimiento y experiencia de expertos en el tema [5]. Un SLD se compone de un fusificador que convierte las variables de entrada escalares a entradas difusas, de reglas difusas, inferencia difusa, y de un defusificador que convierte el resultado difuso del SLD a un resultado escalar [6]. El diseño del recomendador SLD para rehabilitación de fractura de tobillo, parte de la definición de las variables de entrada y salida identificadas en el análisis de datos. Las variables de entrada son las condiciones clínicas del paciente resumidas en la Tabal 1, que simbólicamente se representan por x_i para $i = 1,2, \dots 12$. La Tabla 3 muestra la definición de las variables de entrada. Las variables de salida son el conjunto de terapias de rehabilitación resumidas en la Tabla 2 y se representan simbólicamente como y_i para $i = 1,2, \dots 14$, tal como se muestra en la Tabla 4.

Después de identificar las variables de entrada y salida, se definen las reglas difusas del tipo SI-ENTONCES relacionando de esta forma, las variables de entrada con las variables de salida. Las reglas difusas se diseñaron para cada salida del recomendador de acuerdo con la experiencia del especialista, los registros en los expedientes clínicos y los valores de las condiciones clínicas del paciente. Se desarrollaron 14 subsistemas que contienen las posibles combinaciones de las variables de entrada formando las reglas difusas para cada variable de salida. Las entradas y las reglas difusas son los elementos necesarios para que el mecanismo de inferencia proporcione el resultado esperado del SLD. El resultado proporcionado por el SLD debe apoyar los objetivos de rehabilitación de acuerdo con las

condiciones clínicas y capacidades biomecánicas que presenta el paciente en la exploración física y así mismo a la recuperación del tobillo fracturado del paciente.

Tabla 2. Conjunto de terapias recomendadas para la rehabilitación de fractura de tobillo.

Terapia	Acrónimo	Ejecución
Ultrasonido en profundidad superficial	US	Una sesión diaria 3 MHz durante 5 minutos
Terapia interferencial edema	TIE	Una aplicación diaria de 15 minutos de 0 a 200 Hz
Terapia interferencial dolor	TID	Una aplicación diaria de 15 minutos de 80 a 150 Hz
Terapia interferencial consolidación	TIC	Una aplicación diaria de 15 minutos de 100 Hz
Terapia de cambios térmicos	HYC	Una sesión diaria de frío y calor alternadamente por 15 minutos
Ejercicios de propiocepción	P	Una serie diaria de 10 repeticiones
Ejercicios de movilidad en flexión dorsal, flexión plantar, inversión y eversión	EM	Una serie diaria de 10 repeticiones
Masaje antiedema	MAE	Un masaje diario de 5 minutos
Masaje antiadherente	MH	Un masaje diario de 5 minutos
Ejercicios isométricos	EIM	Una serie diaria de 8 a 10 repeticiones
Ejercicios isotónicos	EIT	Una serie diaria de 8 a 10 repeticiones
Masaje descontracturante	MD	Un masaje diario de 5 minutos
Ejercicio de caminata para reeducación de la marcha	RM	Una serie diaria de 10 a 15 vueltas en una pista de 3 metros
Ejercicios de estiramiento	EE	Una sesión diaria por 5 minutos

Tabla 3. Variables de entrada del SLD.

Variables de entrada	
Símbolo	**Variable**
x_1	Dolor
x_2	Edema
x_3	Limitación de movimiento
x_4	Cambios tróficos
x_5	Contractura
x_6	Herida
x_7	Marcha
x_8	Fuerza
x_9	Sensibilidad
x_{10}	ROMs
x_{11}	Estabilidad
x_{12}	Consolidación

Después de identificar las variables de entrada y salida, se definen las reglas difusas del tipo SI-ENTONCES relacionando de esta forma, las variables de entrada con las variables de salida. Las reglas difusas se diseñaron para cada salida del recomendador de acuerdo con la experiencia del especialista, los registros en los expedientes clínicos y los valores de las condiciones clínicas del paciente. Se desarrollaron 14 subsistemas que contienen las posibles combinaciones de las variables de entrada formando las reglas difusas para cada variable de salida. Las entradas y las reglas difusas son los elementos necesarios para que el mecanismo de inferencia proporcione el resultado esperado del SLD. El resultado proporcionado por el SLD debe apoyar los objetivos de rehabilitación de acuerdo con las condiciones clínicas y capacidades biomecánicas que presenta el paciente en la exploración física y así mismo a la recuperación del tobillo fracturado del paciente.

Tabla 4. Variables de salida del SLD.

Símbolo	Variable
y_1	US
y_2	TIE
y_3	TID
y_4	HYC
y_5	TIC
y_6	P
y_7	EM
y_8	MAE
y_9	MH
y_{10}	EIM
y_{11}	EIT
y_{12}	MD
y_{13}	RM
y_{14}	EE

Para el diseño de las regalas difusas se programó la relación de las variables de entrada de acuerdo con valor de registro en los expedientes clínicos. Si la variable de entrada es del tipo dicotómica o politómica, en caso de ser una variable dicotómica, ésta solo puede tomar dos valores 1 o 5, donde el 1 indica que la condición clínica no está presente, y el valor 5 indica que la condición clínica está presente. Si la variable de entrada es del tipo politómica, entonces se utiliza la escala del 1 al 5 para considerar los niveles requeridos donde 1 indica que la condición clínica se presenta de forma muy leve, 2 indica que la condición clínica se presenta de manera leve, 3 indica que la condición se presenta de manera moderada, 4 indica que la condición clínica se presenta fuerte y 5 indica que la presencia de la condición clínica es muy fuerte.

2.3 Implementación numérica del algoritmo SLD

Un algoritmo con inferencia Mamdani fue programado en lenguaje Scilab 6.1.0. Se integraron las reglas difusas diseñadas, considerando las posibles combinaciones de las variables de entrada para cada variable de salida del SLD. Al terminar el diseño, se verificó el funcionamiento de cada regla difusa y posteriormente, el desempeño del sistema completo. Los expedientes clínicos fueron probados en el SLD (incluyendo cada cita) de los pacientes que sufrieron fractura de tobillo.

3. Resultados

En el análisis de datos para la selección de expedientes clínicos, se encontró que un total de 11 expedientes clínicos de pacientes que sufrieron fractura de tobillo, cumplieron con los criterios de inclusión y exclusión mencionados en el análisis de datos. Los 11 expedientes clínicos seleccionados fueron analizados para identificar las variables de entrada y salida para el diseño del recomendador SLD y sus reglas.

Las variables x registradas por el médico especialista y que representan las condiciones clínicas del paciente, son las entradas al recomendador SLD y se registran en el sistema como un vector de 12 variables como $X_n = [x_1, x_2, x_3, \dots x_{12}]$. Las variables de salida y son los resultados de la indicación de las terapias para la rehabilitación de la fractura de tobillo y son representadas por un vector de 14 variables como $Y_n = [y_1, y_2, y_3, \dots y_{14}]$, donde cada variable y representa la recomendación para cada terapia.

El diseño de las reglas difusas consiste en el desarrollo de tablas numéricas que contiene las posibles combinaciones de las variables de entrada y salida para el resultado deseado del SLD. Se desarrollaron 14 subsistemas de tablas que representan a cada una de las terapias de recomendación para la rehabilitación de fractura de tobillo. Para ilustrar el método, en este trabajo se desarrolla un

subsistema que representa la recomendación de la terapia de masaje descontracturante (MD) y_{12}. La terapia y_{12} está relacionada con las variables de entrada x_1 y x_5. La variable de entrada x_1 se registra en tres niveles de dolor en una escala del 0 al 10, estos son considerados como se puede observar en la Tabla 5, con los valores del uno al cinco mostrados en la columna de la izquierda: los niveles uno y dos indican un dolor leve; los niveles tres y cuatro indican un dolor moderado; y el nivel cinco indica un nivel severo de dolor; la variable de entrada x_5 solo puede tomar dos valores: si se presenta la condición clínica o no, indicadas con los números cinco y uno respectivamente en la columna central de la Tabla 5. La columna de la derecha indica si la terapia y_{12} será recomendada o no, donde el número uno indica que la terapia y_{12} no será recomendada, mientras que el número cinco indica que la terapia y_{12} será recomendada.

Tabla 5. Reglas difusas para la terapia (MD) y_{12}.

Entrada 1	Entrada 2	Salida
x_1	x_5	y_{12}
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
1	5	5
2	5	1
3	5	1
4	5	1
5	5	1

Las reglas del tipo SI-ENTONCES correspondientes a la Tabla 5 se describen en la siguiente lista:

Si *x_1 es muy leve y x_5 no está presente* ***entonces*** *y_{12} no se recomienda.*
Si *x_1 es leve y x_5 no está presente* ***entonces*** *y_{12} no se recomienda.*
Si *x_1 es moderado y x_5 no está presente* ***entonces*** *y_{12} no se recomienda.*
Si *x_1 es severo y x_5 no está presente* ***entonces*** *y_{12} no se recomienda.*
Si *x_1 es muy severo y x_5 no está presente* ***entonces*** *y_{12} no se recomienda.*
Si *x_1 es muy leve y x_5 está presente* ***entonces*** *y_{12} se recomienda.*
Si *x_1 es leve y x_5 está presente* ***entonces*** *y_{12} no se recomienda.*
Si *x_1 es moderado y x_5 está presente* ***entonces*** *y_{12} no se recomienda.*
Si *x_1 es severo y x_5 está presente* ***entonces*** *y_{12} no se recomienda.*
Si *x_1 es muy severo y x_5 está presente* ***entonces*** *y_{12} no se recomienda.*

Se desarrollaron 14 subsistemas similares al de la Tabla 5 para cada terapia de recomendación. La relación entre las variables de entrada y las variables de salida para cada subsistema se simbolizan con la función $f(x)$. En el caso de la relación que da lugar a la salida y_{12} ilustrada en el ejemplo anterior, sería $y_{12} = f_{12}(x_1, x_5)$. Las relaciones entre cada salida y las variables de entrada de las que depende se describen en las siguientes ecuaciones:

$$y_1 = f_1(x_1) \quad (1)$$
$$y_2 = f_2(x_2) \quad (2)$$
$$y_3 = f_3(x_2) \quad (3)$$
$$y_4 = f_4(x_1, x_2) \quad (4)$$
$$y_5 = f_5(x_{12}) \quad (5)$$
$$y_6 = f_6(x_3,\ x_7, x_9, x_{11}) \quad (6)$$
$$y_7 = f_7(x_{3,}\ x_{10}) \quad (7)$$
$$y_8 = f_8(x_1, x_2) \quad (8)$$
$$y_9 = f_9(x_6) \quad (9)$$

$$y_{10} = f_{10}(x_8) \quad (10)$$
$$y_{11} = f_{11}(x_8) \quad (11)$$
$$y_{12} = f_{12}(x_1, x_5) \quad (12)$$
$$y_{13} = f_{13}(x_7) \quad (13)$$
$$y_{14} = f_{14}(x_3, x_5, x_{10}) \quad (14)$$

Una vez que las variables de entrada son introducidas al SLD, se obtiene el resultado de las variables de salida y. Enseguida se muestra el ejemplo de un caso que se sometió al SLD. La Tabla 6 muestra las condiciones clínicas registradas en el expediente de un paciente que sufrió fractura de tobillo, e indica las variables de entrada x para el SLD y sus valores. En la Tabla 7 se muestran los resultados de terapias recomendadas que proporcionó el SLD; en la columna de la izquierda están las variables de salida y, en la columna central se muestran las recomendaciones indicadas en el centro de rehabilitación, y en la columna de la derecha se muestran las recomendaciones indicadas por el SLD.

Tabla 6. Condiciones clínicas de paciente que sufrió fractura de tobillo.

Variables de entrada	
x_1	5
x_2	xx
x_3	No
x_4	No
x_5	No
x_6	Adherida
x_7	Disbásica
x_8	4
x_9	Alterada
x_{10}	Incompletos
x_{11}	Si
x_{12}	Si

Tabla 7. Resultados de la recomendación en el centro de rehabilitación y el SLD.

Recomendación		
Variables de salida	**Centro de rehabilitación**	**SLD**
y_1	1	1
y_2	1	1
y_3	1	0
y_4	1	1
y_5	1	0
y_6	1	1
y_7	1	1
y_8	1	0
y_9	1	0
y_{10}	1	1
y_{11}	1	0
y_{12}	1	0
y_{13}	1	1
y_{14}	1	1

Para la interpretación de la recomendación, es decir, la variable de salida, un resultado con el número uno indica que la terapia es recomendada, y un resultado con el número cero indica que la

terapia no es recomendada. Para este caso en particular, las terapias recomendadas por el SLD son $y_1, y_2, y_4, y_6, y_7, y_{10}, y_{13}$ y y_{14}, mientras que las terapias que no se recomendaron son $y_{3,} y_5, y_8, y_9, y_{11}$ y y_{12}. En este trabajo 28 casos más se sometieron al SLD, en total se contó con 35 casos. En la Figura 3 se muestra el total de expedientes sometidos al recomendador SLD, el total de las citas llevadas a cabo en el centro de rehabilitación y el resultado de las recomendaciones proporcionadas por el sistema inteligente SLD. En la misma Figura 3 se observa que el expediente 1 cumplió con cuatro citas en el centro de rehabilitación; los expedientes dos, tres, y cuatro cumplieron con tres citas cada uno; los expedientes cinco y seis tuvieron dos citas cada uno; el expediente siete tuvo una cita; el expediente ocho cumplió tres citas; el expediente nueve tuvo seis citas; el expediente 10 tuvo una cita; y el expediente 11 cumplió con siete citas. De los resultados obtenidos, se determina la frecuencia en la que se está recomendando cada terapia, esta información la podemos apreciar en la Figura 4.

E11	C7														
	C6														
	C5														
	C4														
	C3														
	C2														
	C1														
E10	C1														
E9	C6														
	C5														
	C4														
	C3														
	C2														
	C1														
E8	C3														
	C2														
	C1														
E7	C1														
E6	C2														
	C1														
E5	C2														
	C1														
E4	C3														
	C2														
	C1														
E3	C3														
	C2														
	C1														
E2	C3														
	C2														
	C1														
E1	C4														
	C3														
	C2														
	C1														
Expediente	Cita	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}

Figura 3. Resultados del recomendador SLD. E1 a E11 son los expedientes clínicos y C1 a C7 son las citas en el centro de rehabilitación.

4. Conclusiones

De acuerdo con los resultados que se obtuvieron de los 35 casos de recomendaciones por el sistema inteligente SLD, se encontró que las variables de entrada x que son consideradas para la indicación de la terapia de rehabilitación de fractura de tobillo son las 12 variables definidas en este trabajo, así como se definieron 14 recomendaciones de terapia para su tratamiento indicadas con las variables y. Las reglas difusas fueron diseñadas de tal forma que el sistema inteligente es capaz de proporcionar la recomendación de manera automatizada. La relación de las variables de entrada con las variables de salida es incluyente, una variable de entrada puede estar relacionada con una o más variables de salida a la vez, donde la programación de las reglas hace la diferencia para cada respuesta esperada.

En la Figura 3 se muestran los resultados obtenidos de los 35 casos sometidos al SLD, donde se observa información relevante acerca de la recomendación que más frecuentemente se indica para las terapias, así como la terapia que con menos frecuencia es recomendada. La mayoría de las terapias son recomendadas en la primera y segunda cita del paciente al centro de rehabilitación, mientras que, de la segunda a la cuarta cita no es necesaria la indicación de algunas de las terapias, y conforme evoluciona de manera positiva la condición del paciente, son necesarias una menor cantidad de terapias para su recuperación. Esto marca una diferencia entre la recomendación establecida y estandarizada indicada en el centro de rehabilitación y la recomendación de las terapias por parte del SLD, lo que nos brinda la oportunidad de identificar elementos clave del proceso para su optimización.

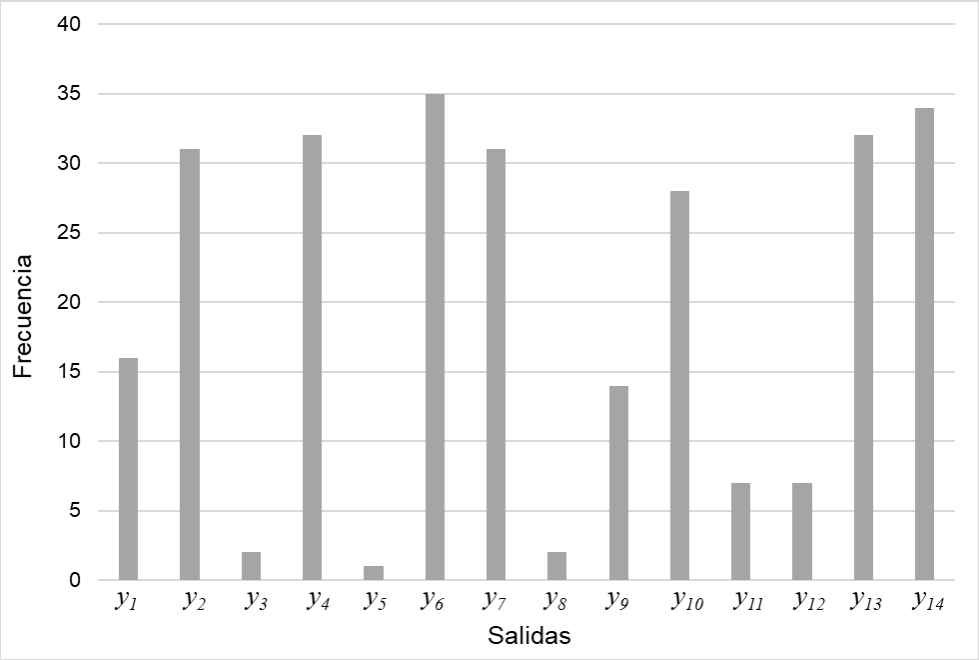


Figura 4. Frecuencia de terapias recomendadas por el SLD.

En la Tabla 8 se muestran los resultados en porcentajes de la frecuencia en que fueron recomendadas las terapias para la rehabilitación de la fractura de tobillo usando el recomendador automatizado SLD. Se puede observar que hay terapias que se recomiendan en todos los casos y terapias que se recomiendan en muy pocos casos. Los porcentajes de recomendación de cierta terapia, permite la planeación de los recursos en el centro de rehabilitación en cuanto a la disponibilidad de equipo, así como a la organización de personal especializado.

Tabla 8. Frecuencia de recomendación por terapia de 35 casos evaluados en el SLD.

Terapia	Frecuencia de recomendación (%)
y_1	46
y_2	88.6
y_3	5.7
y_4	91
y_5	3
y_6	100
y_7	88.6
y_8	5.7
y_9	40
y_{10}	80
y_{11}	20
y_{12}	20
y_{13}	91
y_{14}	97

Los resultados en la Tabla 8 muestran que la terapia y_6 se recomendó el 100% de los casos, después sigue la terapia y_{14} recomendada en el 97% de los casos, luego siguen las terapias y_4 y y_{13} que se recomendaron en un 91% de los casos, enseguida las terapias y_2 y y_7 que se recomendaron el 88.6% de los casos, luego la recomendación y_{10} indicada en el 80% de los casos, después la terapia y_1 que se recomendó en el 46% de los casos, luego la terapia y_9 indicada el 40% de los casos y entre las terapias que se recomiendan en el menor número de casos están las terapias y_{11} y y_{12} indicadas el 20% de los casos, así como las terapias y_8 y y_3 recomendadas en el 5.7% de los casos y por último está la terapia y_5 recomendada solo en el 3% de los casos.

De acuerdo con el diseño del sistema inteligente basado en lógica difusa y sus resultados, se puede concluir que es posible realizar la automatización de la recomendación para la terapia de rehabilitación de fractura de tobillo. La automatización de este proceso aporta más elementos hacia la automatización de procesos de rehabilitación en este campo de la medicina, además del uso de dispositivos automatizados para la ejecución de los ejercicios de movilidad, que actualmente está reportado en la literatura.

Con la aplicación del SLD diseñado en este trabajo para el campo de la rehabilitación, es posible recomendar una terapia personalizada de una manera automatizada, a través de la reducción de terapias, basada en las condiciones del paciente, lo cual podría apoyar a la optimización de recursos en un centro de rehabilitación y reducir el número de sesiones indicadas, mejorando el tiempo de las sesiones, y así, a la recuperación del paciente. También es posible utilizar el SLD como una herramienta de apoyo para el especialista o el médico en formación. Este trabajo reporta el diseño de un sistema inteligente basado en información clínica y supervisión de expertos, la validación experimental del mismo queda como trabajo futuro de esta investigación.

Referencias

[1] Kearney R. *et al.*, *"Ankle injury rehabilitation (AIR): A feasibility randomised controlled trial comparing functional bracing to plaster cast in the treatment of adult ankle fractures," Pilot Feasibility Stud.*, vol. 5, no. 1, pp. 1–8, 2019, doi: 10.1186/s40814-019-0441-6.

[2] Odak S., Ahluwalia R., Unnikrishnan P., Hennessy M., and Platt S., *"Management of Posterior Malleolar Fractures: A Systematic Review," J. Foot Ankle Surg.*, vol. 55, no. 1, pp. 140–145, 2016, doi: 10.1053/j.jfas.2015.04.001.

[3] Matthews P. *et al.*, *"Early motion and directed exercise (EMADE) versus usual care post ankle fracture fixation: Study protocol for a pragmatic randomised controlled trial," Trials*, vol. 19, no. 1,

pp. 1–10, 2018, doi: 10.1186/s13063-018-2691-7.

[4] Bullock S., Allen G., Watson M., and Wilson D., *"Predicting poor outcome from simple ankle injuries: A prospective cohort study," Br. J. Radiol.*, vol. 91, no. 1081, 2018, doi: 10.1259/bjr.20170213.

[5] Zadeh L., *"Fuzzy Sets," Inf. Control*, vol. 8, pp. 338–353, 1965.

[6] Mendel J., *"Fuzzy Logic System for Engineering: A Tutorial," IEEE*, 1995.

[7] Nilsson G., Jonsson K., Ekdahl C., and Eneroth M., *"Effects of a training program after surgically treated ankle fracture: A prospective randomised controlled trial," BMC Musculoskelet. Disord.*, vol. 10, no. 1, pp. 1–11, 2009, doi: 10.1186/1471-2474-10-118.

[8] Maftouni M., Turksen I., Zarandi M., and Roshani F., *"Type-2 fuzzy rule-based expert system for Ankylosing spondylitis diagnosis," Annu. Conf. North Am. Fuzzy Inf. Process. Soc. - NAFIPS*, vol. 2015-Septe, pp. 0–4, 2015, doi: 10.1109/NAFIPS-WConSC.2015.7284195.

[9] Amiri F. and Khadivar A., *"Fuzzy ekspertni sustav za dijagnozu i liječenje mišićnoskeletnih poremećaja u ručnom zglobu," Teh. Vjesn.*, vol. 24, pp. 147–155, 2017, doi: 10.17559/TV-20150531034750.

[10] Polat K. and Güneş S., *"An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease," Digit. Signal Process. A Rev. J.*, vol. 17, no. 4, pp. 702–710, 2007, doi: 10.1016/j.dsp.2006.09.005.

[11] Kele A., Kele A., and Yavuz U., *"Expert system based on neuro-fuzzy rules for diagnosis breast cancer," Expert Syst. Appl.*, vol. 38, no. 5, pp. 5719–5726, 2011, doi: 10.1016/j.eswa.2010.10.061.

[12] Muthukaruppan S. and Er M., *"A hybrid particle swarm optimization based fuzzy expert system for the diagnosis of coronary artery disease," Expert Syst. Appl.*, vol. 39, no. 14, pp. 11657–11665, 2012, doi: 10.1016/j.eswa.2012.04.036.

[13] Asl A. and Zarandi M., *"A type-2 fuzzy expert system for diagnosis of leukemia," Adv. Intell. Syst. Comput.*, vol. 648, no. CII, pp. 52–60, 2018, doi: 10.1007/978-3-319-67137-6_6.

[14] Hosseini R., Latifi F., and Mahdi M., *"A Fuzzy-GA Approach for Parameter Optimization," J. Adv. Comput. Eng. Technol.*, vol. 2, no. 2, pp. 33–42, 2016.

[15] Biyouki S., Turksen I., and Zarandi M., *"Fuzzy rule-based expert system for diagnosis of thyroid disease," 2015 IEEE Conf. Comput. Intell. Bioinforma. Comput. Biol. CIBCB 2015*, 2015, doi: 10.1109/CIBCB.2015.7300333.

[16] Hoppenfeld S. and Murthy V., *Fracturas tratamiento y rehabilitación*. Madrid: Marban, 2001.

Diseño Colaborativo en Mecatrónica, Capítulo 10, pp. 126 - 140
ISBN: 978-607-9394-23-3, 2021

Manejo de LED's WS2812: Un caso de estudio para integrar código ensamblador en Arduino

Santiago Espinosa Felipe, Barahona Ávalos Jorge Luis

Universidad Tecnológica de la Mixteca
Instituto de Electrónica y Mecatrónica
Km. 2.5 Carretera a Acatlima
Huajuapan de León, Oaxaca
fsantiag@mixteco.utm.mx, jbarahona@mixteco.utm.mx

Resumen

Con la serie WS2812 se hace referencia a LED's RGB, con un encapsulado 5050, que integran un controlador inteligente para facilitar su manejo a través de una línea serial de entrada (DI), el encapsulado incluye un bit de salida (DO) para una conexión en cascada con otros LED's. La información que recibe el controlador se codifica por el ancho de los niveles alto y bajo en cada bit, se requieren periodos de tiempo muy precisos para distinguir los ceros de los unos. Si se pretende manejar estos LED's desde una tarjeta Arduino, con las funciones del entorno no se pueden conseguir los periodos de tiempo requeridos, para ello es necesario agregar instrucciones de más bajo nivel. En este documento se muestra cómo integrar código ensamblador en los sketches de Arduino para el manejo de estos LED's, este aspecto se puede aprovechar en otras aplicaciones con requerimientos de temporización muy explícitos. En los resultados se muestra el desempeño del código ensamblador en dos ejemplos de aplicación, en el primero se maneja un círculo con 16 LED's y en el segundo, un panel de 256 LED's, dispuesto como una matriz de 16 x 16.

Palabras clave: Arduino, Ensamblador, LED's WS2812.

1. Introducción

Las tarjetas Arduino se han hecho muy populares por diversos factores, entre los que destaca su precio, portabilidad y flexibilidad del entorno de desarrollo. Las tarjetas y el IDE de Arduino se crearon como herramientas para los estudiantes del Instituto de Diseño Interactivo Ivrea (en Italia), con la intención de proporcionar una forma fácil y económica para que los estudiantes pudieran hacer obras interactivas que incluyeran sensores y actuadores [1]. Por ello, el entorno y organización de las tarjetas son muy intuitivos, basta con conocimientos básicos de programación y electrónica para el desarrollo de proyectos interesantes.

El creciente uso de las tarjetas Arduino y la creación de un extenso número de tarjetas de expansión (*shields*), ha hecho que los usuarios de microcontroladores y desarrolladores de sistemas embebidos también se interesen por su uso, porque facilita una rápida evaluación de prototipos y porque el entorno permite el acceso a los registros de configuración del microcontrolador (MCU) incluido en la tarjeta, aprovechando todas sus características, como el manejo de recursos por medio de interrupciones.

Sin embargo, cuando una aplicación tiene requerimientos de temporización muy estrictos, que no se pueden conseguir con el uso de los temporizadores del MCU, estos requerimientos no se podrán cubrir con el uso de las funciones de Arduino, porque incorporan código adicional y su ejecución requiere de ciclos de reloj de la CPU, de manera que no se tiene la certeza del tiempo invertido en su ejecución. En esos casos es necesario emplear código ensamblador, el cual, por ser un lenguaje de bajo nivel,

requiere de conocimientos sobre la organización del microcontrolador, pero con las ventajas de que se genera un código máquina compacto y, se puede estimar con precisión su tamaño y tiempo de ejecución.

Un caso de estudio que requiere de periodos estrictos de tiempo es el manejo de los LED's RGB WS2812, estos LED's son manufacturados por la empresa World-Semi [2] y su manejo se hace por medio de una línea serial de datos, sin importar cuantos LED's se conectan en cascada. Esta es la motivación principal del trabajo, mostrar cómo integrar código ensamblador en un *sketch* de Arduino, específicamente para el manejo de los LED's WS2812, pero este híbrido de programación se puede aplicar a otras aplicaciones o simplemente como un mecanismo didáctico, para evaluar la ejecución de código ensamblador utilizando todo el ecosistema de recursos que Arduino proporciona.

2. Los microcontroladores empleados por Arduino

Las tarjetas Arduino son manejadas por microcontroladores AVR, esta familia fue comercializada por la compañía Atmel, aunque actualmente pertenece a la compañía Microchip [3]. Las Arduino UNO, arduino Nano y arduino LilyPad incluyen un ATMega328P y las arduino Mega están soportadas por un ATMega2560. En la figura 1 se muestran estos modelos de tarjetas, que son de las más populares.

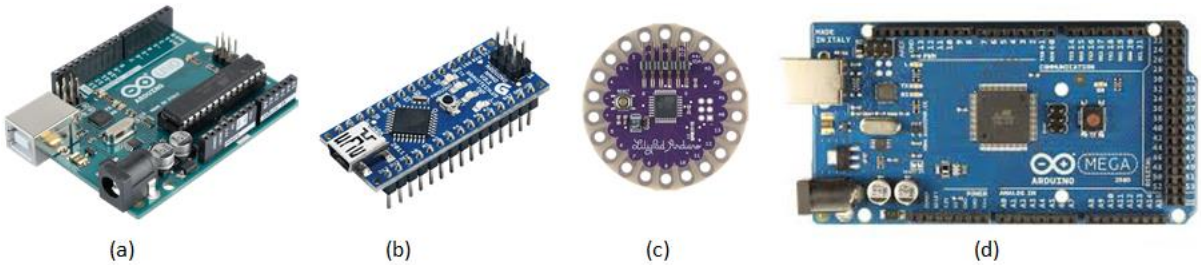


Figura 1. Tarjetas Arduino (a) UNO, (b) Nano, (c) LilyPad y (d) Mega [1].

La familia AVR tiene una arquitectura RISC de 8 bits, literalmente con el acrónimo RISC se hace referencia a computadoras con un repertorio de instrucciones reducido (*Reduced Instruction Set Computers*), sin embargo, la característica más importante de este tipo de arquitectura es la simplicidad en el hardware, con lo que se consigue un alto rendimiento. En los AVR el rendimiento se maximiza porque las instrucciones se ejecutan en una segmentación a dos etapas, en la primera, la instrucción es capturada de la memoria Flash y en la segunda, la instrucción pasa a ejecutarse con el apoyo de la ALU, mientras eso sucede, se realiza la captura de la siguiente instrucción, de esta manera, hay un paralelismo oculto al programador, puesto que en un instante de tiempo se tienen 2 instrucciones dentro del MCU. Con ello, la productividad en los AVR se aproxima a una instrucción por ciclo de reloj [4].

Los microcontroladores AVR incluyen 3 tecnologías de memoria diferentes en el chip, una memoria Flash para el código, un espacio de memoria SRAM para el manejo de los datos y un espacio de EEPROM para datos que se van a conservar en ausencia de energía. Por esta separación en los espacios y buses para el manejo de memoria, la familia AVR se considera con una arquitectura Harvard. Una característica en el éxito en la familia AVR es la organización de la memoria Flash, esta contiene dos secciones, una sección de arranque y una sección de aplicación. A través de fusibles de configuración se determina el límite entre secciones y se establece si, después de un reinicio, el MCU iniciará la ejecución en la sección de aplicación o en la sección de arranque. La sección de arranque fue diseñada para la ubicación de un cargador, es decir, un programa que evalúa si existe una nueva versión de la aplicación y proporciona las facilidades para su escritura en su respectiva sección, por ello, los microcontroladores AVR se pueden auto-programar sin la necesidad de hardware adicional [4].

En la figura 2 se muestra al núcleo AVR, se observa una organización que permite la segmentación a dos etapas, así como los 3 espacios de memoria con diferentes tecnologías. Otro

bloque que resalta es el archivo de registros, con 32 registros de propósito general, con los nombres de R0, R1 hasta R31. La ALU sólo puede realizar operaciones con estos registros, si la información de la memoria va a procesarse, primero se debe transferir a los registros. Por ello, esta arquitectura también es considerada del tipo Registro-Registro. A través de cargas se transfiere de memoria a registros y con almacenamientos la transferencia es de registros a memoria.

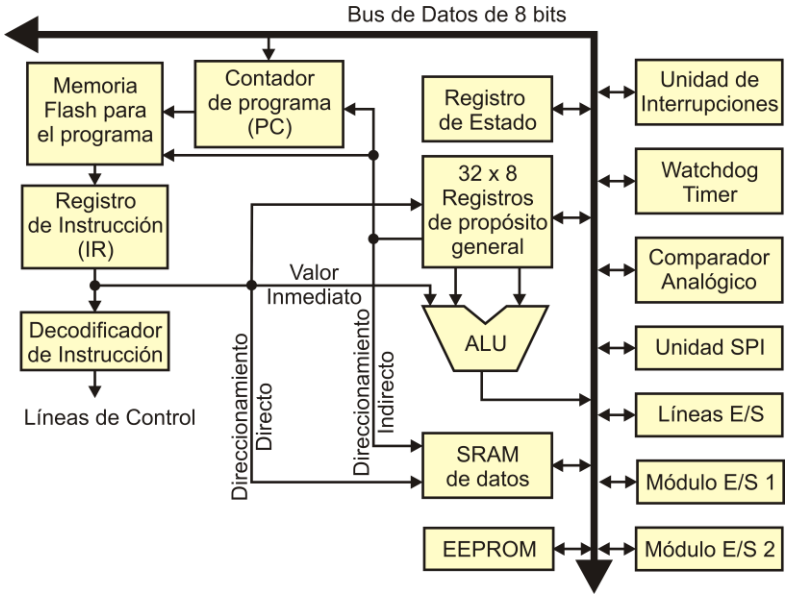


Figura 2. El núcleo AVR [4].

Los bloques a la derecha de la figura 2 corresponden a los recursos del microcontrolador, estos son manejados a través de Registros I/O y difieren entre los miembros de la familia AVR, sólo el núcleo básico es compartido por todos ellos, por lo que son capaces de ejecutar el mismo repertorio de instrucciones.

2.1 El Microcontrolador ATMega328P

Las pruebas realizadas en este trabajo se hicieron con tarjetas Arduino Nano y Arduino UNO, ambas incluyen al Microcontrolador ATMega328P, este es un miembro de la familia AVR con las características y recursos que se listan en la tabla 1.

Tabla 1. Características y recursos del ATMega328P [3].

Flash (Código) :	32 Kbyte	**Interfaz TWI (I^2C) :**	1
EEPROM (Datos) :	1 Kbyte	**SPI (Maestro/Esclavo) :**	1
SRAM (Datos) :	2 Kbyte	**ADC :**	1 de 10 bits
Terminales Ent/Sal :	23	**Comparador Analógico :**	1
Frec. Máxima :	20 MHz	**USART :**	1
Voltaje de alimentación :	1.8 – 5.5 V	**Temporizador de 16 bits :**	1
Watchdog Timer :	1	**Temporizador de 8 bits :**	2
Oscilador Interno :	Si	**Interrupciones :**	26

Las 23 terminales de entrada o salida están distribuidas en 3 puertos, los puertos B y D son de 8 bits y el puerto C es de 7 bits, sus terminales se renombran en Arduino para facilitar su manejo desde

el entorno de desarrollo. En la tabla 2 se muestra la relación, a las terminales del puerto C se les renombra como A0 a A5 porque corresponden con las entradas de un multiplexor analógico que está conectado con el ADC, aunque nada impide que se utilicen como entradas o salidas digitales. La terminal más significativa del puerto C tiene la función alterna del *reset*, por ello no queda disponible para uso general. Otras dos terminales que Arduino utiliza son la 6 y 7 del puerto B, en estas se conecta un cristal de 16 MHz, aunque el microcontrolador tiene un oscilador interno que puede operar hasta 8 MHz, en Arduino se trabaja a una frecuencia más alta empleando un oscilador externo.

Tabla 2. Relación entre las terminales de Arduino y el ATMega328P.

Arduino	ATMega328P	Arduino	ATMega328P	Arduino	ATMega328P
0	PD0	8	PB0	A0	PC0
1	PD1	9	PB1	A1	PC1
2	PD2	10	PB2	A2	PC2
3	PD3	11	PB3	A3	PC3
4	PD4	12	PB4	A4	PC4
5	PD5	13	PB5	A5	PC5
6	PD6	XTAL 1	PB6	Reset	PC6
7	PD7	XTAL 2	PB7		

2.2 Niveles de programación en el ATMega328P

En la figura 3 se muestra una estructura jerárquica con los niveles de programación para el MCU ATMega328P, Arduino puede considerarse como una biblioteca de funciones que se ejecuta sobre código C/C++, por ello, todos los elementos del estándar de C/C++ pueden ser empleados en un *sketch* de Arduino e incluso se puede tener acceso a los Registros I/O para realizar la configuración y control de los recursos internos, así como para conocer su estado.

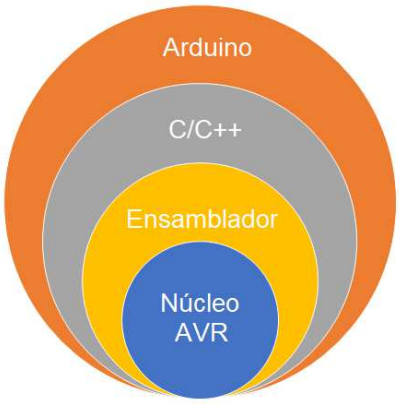


Figura 3. Niveles de Programación para el ATMega328P.

Se observa en la figura 3 que por debajo del lenguaje C/C++ está el ensamblador, de hecho, el compilador AVR-gcc reconoce a la función asm( ), mediante la cual se pueden integrar instrucciones en ensamblador dentro de un programa en C. En un *sketch* de Arduino también se puede utilizar la función asm( ), sin embargo, una mejor opción consiste en integrar funciones completas, las cuales se deben declarar como externas al principio del *sketch* con la sintaxis:

```
extern "C" void  funcion();
```

La definición de las funciones se coloca en un archivo con extensión S, para que sea interpretado como ensamblador. En la figura 4 se muestra la forma de integrar el nuevo archivo, en (a) está el botón para mostrar las opciones con los archivos del proyecto, en (b) se resalta la opción para un nuevo archivo y en (c) se ejemplifica la inclusión de un archivo, el nombre no es importante, sólo la extensión.

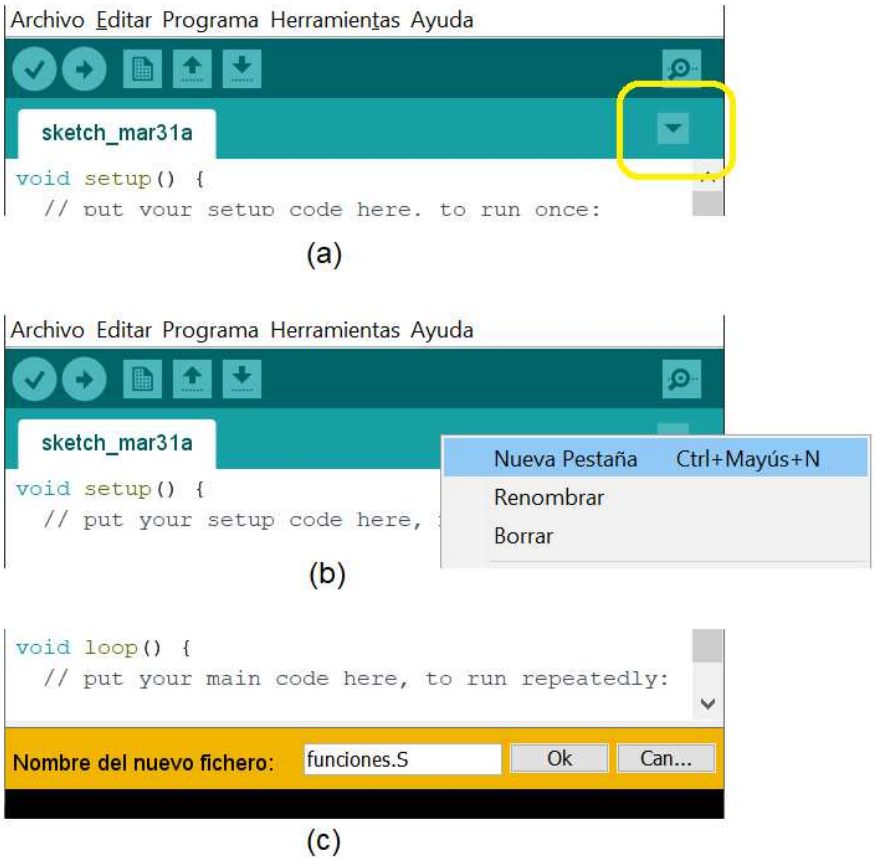


Figura 4. Integración de un nuevo archivo en un *sketch* de Arduino.

Para comprender los elementos requeridos en la inclusión de funciones en lenguaje Ensamblador dentro de un *sketch* de Arduino, en la figura 5 se muestra un ejemplo, se trata del clásico parpadeo de un LED, ubicado en la terminal 13 de una tarjeta Arduino que corresponde con el bit 5 del puerto D. En el *sketch* se utilizan 3 funciones, una para configurar la terminal como salida, otra para colocar un nivel lógico alto y la última para colocar un nivel lógico bajo.

En el archivo Principal.ino solo se declaran las funciones como externas para que se utilicen dentro del *sketch*. En el archivo Funciones.S se incluye una biblioteca para que se reconozcan todos los nombres de los Registros I/O, luego, con el apoyo de una directiva cada función se declara como global, y posteriormente se escribe el código ensamblador con el cuerpo de la función. Debe emplearse la función _SFR_IO_ADDR( ) en todos los Registros I/O para que no se interpreten como constantes.

En lo que respecta al código, el Registro I/O DDRB es utilizado para definir la dirección del puerto B, con 0 una terminal es entrada y con un 1 se definen las salidas. En el registro PORTB se va a escribir cuando un puerto es salida, para que el dato se refleje en las terminales del puerto. La instrucción SBI P, S pone en alto al bit S del Registro I/O P, mientras que la instrucción CBI P, S pondrá en bajo al bit S del Registro I/O P. En ambos casos, no se modifican otros bits.

Principal.ino

```
extern "C" void conf_pin();
extern "C" void led_on();
extern "C" void led_off();

void setup() {
   conf_pin();
}

void loop() {
  led_on();
  delay(500);
  led_off();
  delay(500);
}
```

Funciones.S

```
#include "avr/io.h"

.global conf_pin
conf_pin:
   sbi _SFR_IO_ADDR(DDRB), 5
   ret

.global led_on
led_on:
   sbi _SFR_IO_ADDR(PORTB), 5
   ret

.global led_off
led_off:
   cbi _SFR_IO_ADDR(PORTB), 5
   ret
```

Figura 5. Ejemplo de uso de Ensamblador en Arduino.

Es evidente que existe una política para el manejo de los registros de propósito general, una función puede recibir argumentos en los registros de R25 a R8, el retorno se ubica a partir de R25 como el byte menos significativo y dentro de la función se pueden modificar libremente los registros R18 a R25, así como los apuntadores X y Z, si se utilizan otros registros, su valor debe conservarse con el apoyo de la pila [5].

3. Los LED's WS2812

Los LED's WS2812 son manufacturados por la empresa World-Semi [2], cada LED RGB incluye un control inteligente integrado en un encapsulado 5050 (5 mm x 5 mm), en la figura 6 se muestra el aspecto de estos LED's. En cada uno de los tres colores primarios se puede variar el brillo con 256 niveles, por lo que teóricamente se pueden conseguir: 256 x 256 x 256 = 16, 777, 216 colores diferentes, sin embargo, en la práctica se distinguen pocos colores.

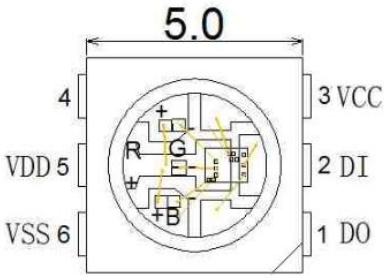


Figura 6. Un LED WS2812 [2].

Cada LED es manejado a través de una línea serial, recibiendo la información del color en la terminal DI, esta consiste en una trama de 3 bytes (24 bits) con el formato mostrado en la figura 7. Lo que significa que primero se recibe el valor con la intensidad del LED verde, luego la intensidad para el LED rojo y finalmente, la intensidad para el LED azul, en todos los casos, se inicia con el bit más significativo.

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0

Figura 7. Formato de la información RGB para un LED.

Para una conexión en cascada, la salida DO de un LED se conecta con la entrada DI del siguiente. Un LED se queda con los 24 bits recibidos, si recibe información adicional, la transfiere hacia su salida DO, de manera que, con N LED's conectados en cascada el microcontrolador debe enviar 3 x N bytes.

En la figura 8 se muestra la conexión de 3 LED's en cascada y la secuencia de información que cada uno recibe, el LED1 recibe 3 paquetes de 24 bits, utiliza el primero y transfiere los otros dos, el LED2 recibe 2 el segundo y tercer paquete, utiliza el segundo y transfiere el tercero, finalmente, el LED3 recibe y utiliza el tercer paquete de 24 bits.

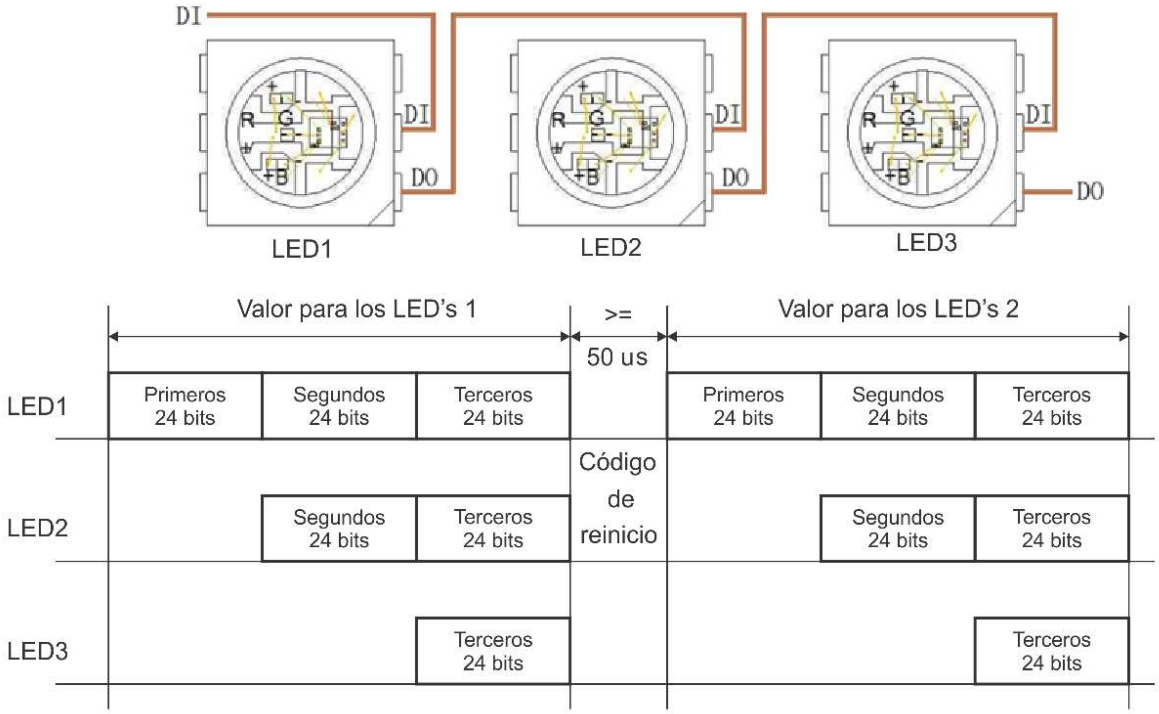


Figura 8. Conexión de 3 LED's con la secuencia de información recibida.

Una vez que se recibe la información para cada uno de los LED's, el microcontrolador debe esperar un tiempo mínimo de 50 us antes de enviar una nueva secuencia de información, a este tiempo se le conoce como un código de reinicio.

La definición de un 1 y un 0 se realiza con el ancho de los niveles alto y bajo, en la figura 9 se muestran los periodos de tiempo involucrados y en la tabla 3 se puede ver la duración de cada periodo, en cada caso se tiene una tolerancia de ± 150 ns. Por lo tanto, el periodo para un 0 lógico es de 1150 ns ± 300 ns y para un 1 lógico es de 1300 ns ± 300 ns [2].

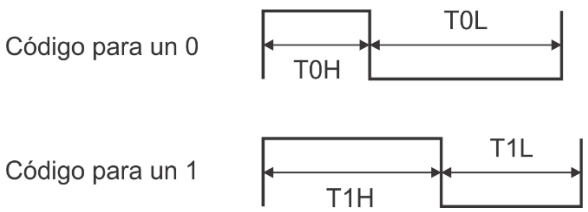


Figura 9. Periodos de tiempo involucrados en cada bit [2].

Las tarjetas Arduino incluyen un cristal de 16 MHz, por lo tanto, el periodo del ciclo de reloj es de 62.5 ns, para relacionar la duración de los periodos de tiempo con los ciclos de reloj en Arduino, en la tabla 3 también se incluye una columna con esa información.

Tabla 3. Duración de cada uno de los periodos de tiempo.

Periodo	Duración	Ciclos de reloj en Arduino
T0H	350 ns ± 150 ns	5.6 ± 2.4 ns
T0L	800 ns ± 150 ns	12.8 ± 2.4 ns
T1H	700 ns ± 150 ns	11.2 ± 2.4 ns
T1L	600 ns ± 150 ns	9.6 ± 2.4 ns

Estos requerimientos en ciclos de reloj hacen necesario el uso de lenguaje ensamblador, en alto nivel no se tiene la certeza de cuantos ciclos de reloj se invertirán en cada instrucción.

3.1 Generación de señales con Ensamblador

Una secuencia de instrucciones que convierte un byte a su equivalente serial se muestra en la figura 10, de acuerdo con la figura 7, primero se debe convertir al bit más significativo y por último al menos significativo. La secuencia toma la información del registro R22 y la salida serial es colocada en la terminal 3 del puerto B.

```
  ldi  r21, 8                  // contador de bits
loop_bit:
  sbi  _SFR_IO_ADDR(PORTB), 3  // Inicia en alto
  nop
  nop
  nop
  sbrs r22, 7                  // Si es un 1 salta
  cbi  _SFR_IO_ADDR(PORTB), 3  // Salida en bajo
  nop
  nop
  lsl  r22                     // Desplaza a la izq.
  cbi  _SFR_IO_ADDR(PORTB), 3  // Salida en bajo
  nop
  nop                          // nop's para ajustar
  nop                          // el tiempo
  nop
  dec  r21                     // Decrementa contador
  brne loop_bit                // Repite si no es cero
```

Figura 10. Secuencia en ensamblador que convierte de paralelo a serie.

Para cada bit, la salida en PB3 comienza en alto, sin importar si se trata de un 0 o un 1 lógico, luego se evalúa al bit más significativo, si es un 1 lógico se salta la instrucción que coloca la salida en bajo, porque el nivel alto es más extenso en un 1 lógico. Las instrucciones NOP (no operación) que se agregan, sirven para completar la cantidad de ciclos requerida. Un desplazamiento a la izquierda (instrucción LSL) hace que en la siguiente iteración nuevamente se evalúe al bit más significativo.

Para analizar la duración de cada periodo, tanto para un 0 lógico y para un 1 lógico, en la figura 11 se compara la ejecución de las instrucciones, indicando cuantos ciclos de reloj se requieren en cada

caso, para luego sumar y obtener los totales, que se observa caen en el intervalo que se mostró en la tabla 3. Esta precisión en la estimación de los periodos de tiempo es el valor agregado que se obtiene al incorporar código ensamblador en un *sketch* de Arduino.

UN CERO LÓGICO

```
  ldi  r21, 8
loop_bit:                          // Ciclos de reloj
  sbi  _SFR_IO_ADDR(PORTB), 3      //   2  (Alto)
  nop                              //   1
  nop                              //   1
  nop                              //   1
  sbrs r22, 7                      //   1
  cbi  _SFR_IO_ADDR(PORTB), 3      //   2 (Bajo)
  nop                              //   1
  nop                              //   1
  lsl  r22                         //   1
  cbi  _SFR_IO_ADDR(PORTB), 3      //   2
  nop                              //   1
  nop                              //   1
  nop                              //   1
  nop                              //   1
  dec  r21                         //   1
  brne loop_bit                    //   2
```

6 ciclos

14 ciclos

UN UNO LÓGICO

```
  ldi  r21, 8
loop_bit:                          // Ciclos de reloj
  sbi  _SFR_IO_ADDR(PORTB), 3      //   2  (Alto)
  nop                              //   1
  nop                              //   1
  nop                              //   1
  sbrs r22, 7                      //   2
  cbi  _SFR_IO_ADDR(PORTB), 3      //   0
  nop                              //   1
  nop                              //   1
  lsl  r22                         //   1
  cbi  _SFR_IO_ADDR(PORTB), 3      //   2 (Bajo)
  nop                              //   1
  nop                              //   1
  nop                              //   1
  nop                              //   1
  dec  r21                         //   1
  brne loop_bit                    //   2
```

10 ciclos

9 ciclos

Figura 11. Conteo de ciclos de reloj para cada dígito binario.

La secuencia que se muestra en la figura 10 es para la conversión de paralelo a serie de 1 byte, debe emplearse 3 veces para cada uno de los LED's, en el orden Verde, Rojo y Azul, de acuerdo con la figura 7. En un arreglo con N LED's la secuencia debe emplearse 3xN veces, como se muestra en los ejemplos que se describen a continuación.

3.2 Ejemplo 1: Círculo con 16 LED's

En la figura 12 se muestran 16 LED's conectados en forma circular, exhibiendo la imagen real y el esquemático, en donde se aprecia la conexión en cascada. Para su manejo, se desarrolló la función **void Send_Array( )**, esta se realizó en lenguaje ensamblador y se encarga del envío de los 48 bytes, 3 para cada uno de los LED's.

En la figura 13 se muestra el código de la función **void Send_Array( )**, se observa la inclusión del código de la figura 10, con el que se realiza la conversión de paralelo a serie de un byte. Un aspecto importante es que la información a exhibir debe ser colocada desde un *sketch* en un arreglo denominado **leds,** para que en la función se realice su acceso. La función **Send_Array** coloca al apuntador Z al comienzo del arreglo y realiza la generación de las secuencias seriales. En el código se aprecian dos ciclos anidados, el ciclo externo para los 48 bytes y el ciclo interno para los 8 bits de cada uno de los bytes.

En cuanto a los registros empleados, el apuntador Z es de 16 bits y se forma con los registros R30 para la parte baja y R31 para la parte alta, R18 es el contador de datos que controla el ciclo externo, cada dato es obtenido de la memoria y colocado en R21, finalmente, R22 es el contador de bits y controla el ciclo interno.

ISBN: 978-607-9394-23-3, 2021

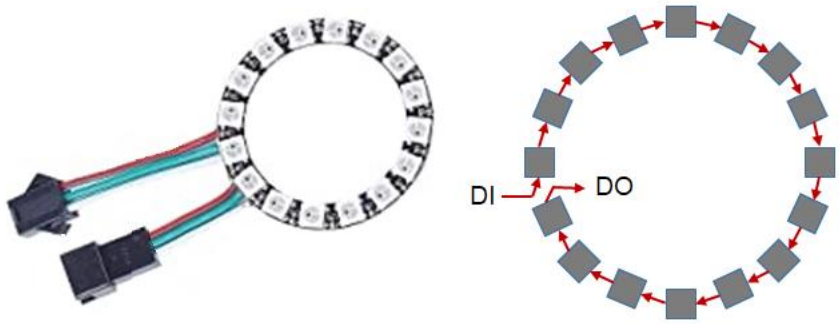


Figura 12. Arreglo de 16 LED's WS2812 dispuestos en forma circular.

```
Send_Array:
  ldi  r30, lo8(leds)          // Z apunta al comienzo del arreglo
  ldi  r31, hi8(leds)
  ldi  r18, 0                  // Contador de datos
loop_byte:
  ld   r22, z+
  ldi  r21, 8                  // Contador de bits
loop_bit:
  sbi  _SFR_IO_ADDR(PORTB), 3  // Inicia en alto
  nop
  nop
  nop
  sbrs r22,7                   // Si en un 1 salta
  cbi  _SFR_IO_ADDR(PORTB), 3  // Salida en bajo
  nop
  nop
  lsl  r22                     // Desplaza a la izquierda
  cbi  _SFR_IO_ADDR(PORTB), 3  // Salida en bajo
  nop
  nop                          // NOP's para ajustar
  nop                          // el tiempo
  nop
  dec  r21                     // Decrementa el contador
  brne loop_bit                // Repite si no es cero
  inc  r18                     // Incrementa el contador de datos
  cpi  r18, 48                 // Compara con 48
  brlo loop_byte               // Repite si faltan datos
  ret
```

Figura 13. Función que envía información a 16 LED's WS2812

La función **Send_Array** se puede manipular dentro de un *sketch* de Arduino, en la figura 14 se muestran los elementos necesarios para su manejo, se requiere de: un espacio para el almacenamiento de la información, el llenado del espacio de almacenamiento y la llamada a la función **Send_Array**. La información para llenar el espacio de almacenamiento y la frecuencia en que esta información se debe actualizar depende de la aplicación.

En particular, con este arreglo circular de LED's se construyó una lámpara manejada por medio de una tarjeta Arduino Nano, los LED's cambian de color o intensidad al recibir un comando serial desde un módulo Bluetooth HC-05, por lo que también se personalizó una interfaz para un móvil con la App Bluetooth Electronics, disponible para SO Android [6]. En la sección de resultados se muestran imágenes de la lámpara RGB en operación.

```
extern "C" void Send_Array(void);

struct cRGB  {uint8_t g; uint8_t r; uint8_t b; };
struct cRGB  leds[16];            // Espacio de almacenamiento

for(uint8_t  i = 0; i < 16; i++) {
    leds[i].r = rojo;             // Llenado del espacio de
    leds[i].g = verde;            // almacenamiento
    leds[i].b = azul;
}
Send_Array();                     // Manejo de los LED's
```

Figura 14. Elementos para el manejo de LED's WS2812 desde un sketch de Arduino.

3.3 Ejemplo 2: Panel de 16 x 16 LED's

En la figura 15 se muestra un panel de 16 x 16 LED's dispuestos en forma matricial, exhibiendo la imagen real y el esquemático, en donde se aprecia la forma en que se realiza la conexión en cascada, el orden es importante para el envío de la información. La función **Send_Array** se debe acondicionar porque ahora se tienen 256 LED's y cada uno requiere de 3 bytes, por lo que debe enviar un total de 768 bytes de información.

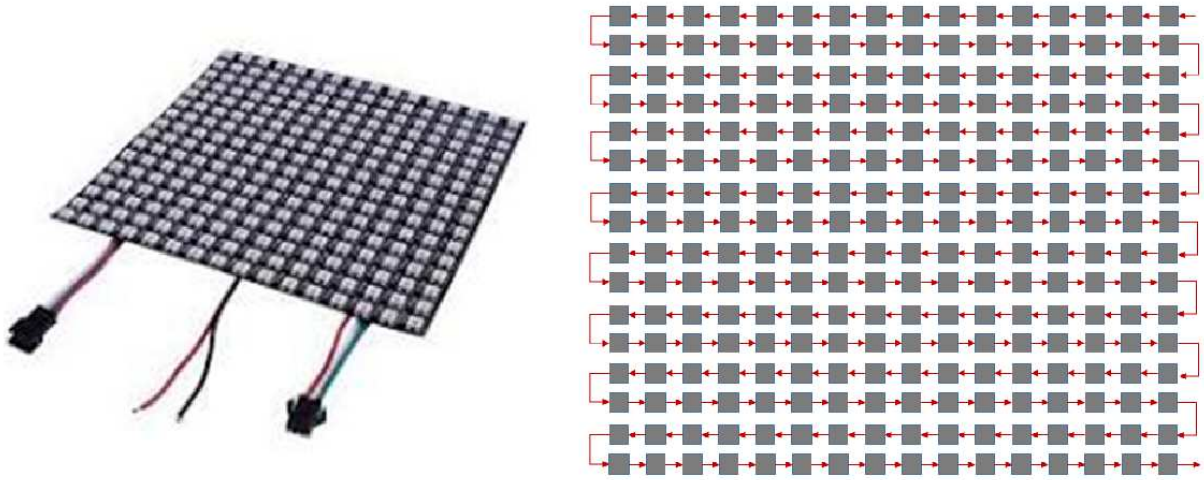


Figura 15. Panel de 16 x 16 LED's WS2812 dispuestos en forma matricial.

El envío de 768 bytes requiere de 3 ciclos anidados en la función **Send_Array**, el conteo de 768 se realiza con un ciclo externo con 3 iteraciones, un ciclo intermedio de 256 iteraciones y un ciclo interno de 8 iteraciones, una por cada bit. El conteo de 256 iteraciones con un registro de 8 bits se consigue inicializando el registro con 0 y haciendo un decremento en cada iteración hasta que vuelva a tener 0. Si un registro tiene 0, después un decremento tendrá 255 (0xFF).

Para enviar información a los LED's se requiere de los elementos mostrados en la figura 14, un espacio para el almacenamiento de la información, el llenado del espacio de almacenamiento y la llamada a la función **Send_Array**. En este caso, el arreglo **Leds[ ]** debe ser de 256 datos del tipo **cRGB**

y para su llenado se debe considerar la forma en que se realiza la conexión en cascada de los LED's, que se muestra en la figura 15.

Para probar este panel de LED's se enviaron barras de colores que se desplazan dinámicamente, así como 3 imágenes de 16 x 16 pixeles. El cambio en la información a mostrar se realiza con un botón, las pruebas se hicieron con una Arduino Uno y en la sección de resultados se muestran imágenes con el panel de LED's en operación.

Un aspecto importante a considerar es el tamaño de la memoria SRAM del microcontrolador, porque solo tiene 2048 bytes de propósito general. Cada imagen ocupa 768 bytes, por las tres imágenes se requiere de 2304 bytes, de manera que las imágenes no se pueden ubicar en la memoria SRAM y la alternativa es colocarlas en la memoria Flash. Para no crecer el *sketch* con la información de las imágenes, esta se puede colocar en un archivo externo, en la figura 16 se muestra la forma en que se declara una imagen ubicada en memoria Flash y su acceso para copiar la información a SRAM, la función **pgm_read_byte( )** es parte del compilador AVR-GCC y se puede emplear en Arduino.

Principal.ino

```
#include "Imagen.h"                // Archivo con la Imagen

extern "C" void Send_Array(void);  // Para el manejo de los LED's

// Se indica que el arreglo es externo
extern const uint8_t  Imagen[] PROGMEM;

struct cRGB  { uint8_t g; uint8_t r; uint8_t b; };
struct cRGB  leds[256];            // Arreglo para los LED's

uint16_t  k, l;

// El arreglo de LED's se llena con
// los datos de la imagen
for(k = 0, l = 0; k < 256; k++, l +=3) {
   leds[k].r = pgm_read_byte(&Imagen[l]);
   leds[k].g = pgm_read_byte(&Imagen[l + 1]);
   leds[k].b = pgm_read_byte(&Imagen[l + 2]);
}

Send_Array();                      // Manejo de los LED's
```

Imagen.h

```
// Arreglo constante en Memoria Flash
const uint8_t  Imagen[] PROGMEM = {
  100,
  100,
  100,
  100,
  100,
  100,
  50,
  50,
  50,
  0,
  0,
  0,

  . . .   // 768 bytes
  0,
  0,
  0 };
```

Figura 16. Elementos para el llenado de un arreglo con información ubicada en memoria Flash.

La información de la imagen se puede generar de forma manual o con el apoyo de alguna herramienta de procesamiento de imágenes. En particular, las imágenes de prueba se descargaron de internet como íconos de 16 x 16, luego, con el apoyo del software GIMP se exportaron al formato PNM con la modalidad de archivo ASCII, en este formato se pueden visualizar los valores RGB de cada uno de los pixeles, esta información se acondicionó para crear el arreglo de cada imagen y colocarla en su respectivo archivo.

4. Resultados

En la sección anterior se describió la forma en que se pueden manejar los LED's WS2812, al integrar código ensamblador en los *sketches* de Arduino. También se mostró el código para el manejo de 2 arreglos de LED's con distribuciones diferentes, en esta sección se muestran los resultados de la puesta en operación del código desarrollado, en la figura 16 se muestra la lámpara implementada con el círculo de 16 LED's, esta lámpara es controlada por una tarjeta Arduino Nano a la cual se le ha conectado un módulo HC-05, para que pueda recibir comandos vía Bluetooth.

Figura 17. Lámpara realizada con un círculo de 16 LED's.

En la figura 17 también se puede ver la interfaz desarrollada con la App Bluetooth Electronics [6], esta es una App muy intuitiva que facilita el diseño de interfaces para el envío de información por medio de Bluetooth, en la interfaz desarrollada se incluyeron 3 *sliders* (barras de desplazamiento) y 7 botones. Desde un móvil se envía información según el evento, con cada cambio en un *slider* se envía un carácter ASCII que identifica al objeto, el valor del *slider* (entre 0 y 255) como una secuencia ASCII y un carácter de fin de trama. Para los botones se envía un carácter ASCII que identifica el botón seguido del carácter de fin de trama.

Con los *sliders* se puede cambiar de manera independiente las intensidades de los colores R, G y B en todos los LED's. Los botones están en dos grupos, en un grupo están los botones con los que se cambia la intensidad de manera automática de cada uno de los colores, dando un efecto de aumento y decremento de la intensidad luminosa, y los botones del otro grupo solicitan el encendido de secuencias intercaladas de colores, para demostrar que los LED's pueden tener colores independientes entre sí. En la figura 18 se muestran 3 ejemplos con diferentes tonalidades para la lámpara RGB, conseguidas con los *sliders*.

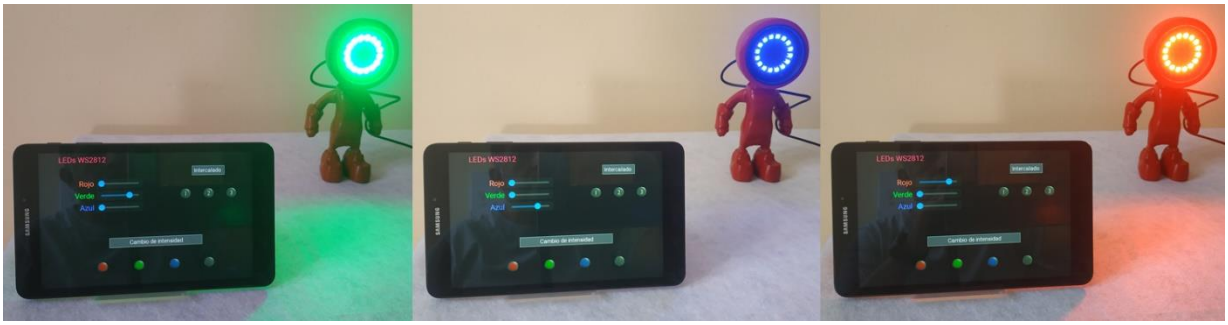


Figura 18. Lámpara RGB en operación.

El otro arreglo de LED's con el que se hicieron las pruebas fue un panel de 16 x 16 LED's, este se manejó con una Arduino Uno, a la cual, además del panel se le conectó un botón. Cuando el sistema se enciende muestra barras de colores, las cuales cambian de manera automática cada segundo, haciendo un desplazamiento para conmutar entre 8 colores diferentes. Por medio del botón se pueden seleccionar pequeñas imágenes precargadas en la memoria Flash del microcontrolador.

En la figura 19 se muestran las barras de colores y las diferentes imágenes que se seleccionaron para su exhibición. Un aspecto importante es que no se distinguen las diferentes intensidades en los colores, por ejemplo, si un LED tiene 0's en sus diferentes valores: R, G y B, el LED está apagado y se considera como negro, pero basta un valor de 50 en R, G y B para que se vea en color blanco, es decir, no se distingue un tono de gris, a pesar de solo tener el 20 % de la intensidad máxima.

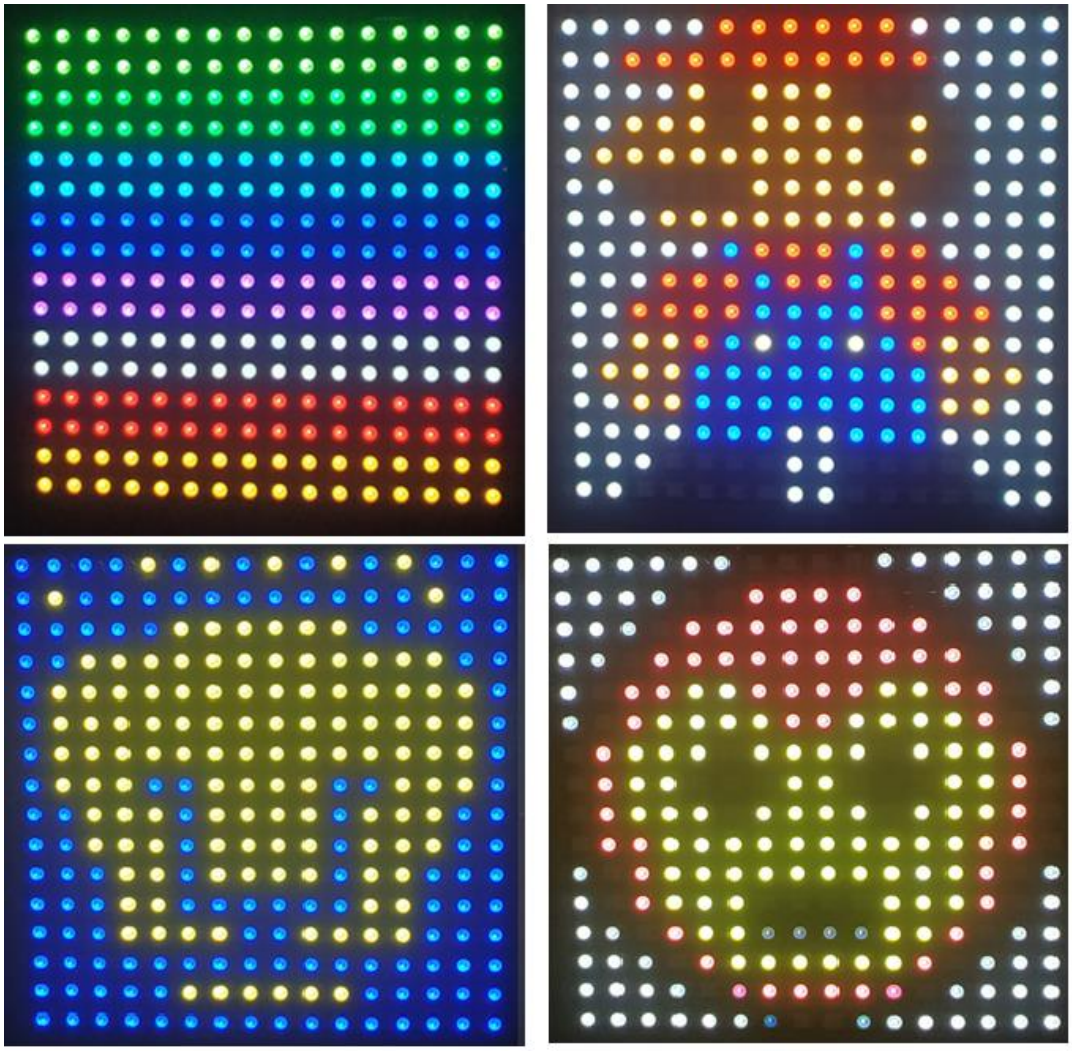


Figura 19. Pruebas en el panel de 16 x 16 LED's.

4.1 Consumo de corriente

Se debe considerar el consumo de corriente en los LED's para no dañar a la tarjeta Arduino, a la fuente de alimentación o al puerto USB de una computadora, dependiendo de donde se tome el suministro de energía. El círculo de 16 LED's RGB con un color blanco consume 320 mA, sin problema se puede alimentar desde la Arduino, conectada a uno de los puertos USB de una computadora, dado que tienen la capacidad de suministrar hasta 500 mA.

El panel de 16 x 16 LED's con la imagen de colores tuvo un consumo de 1.174 A, este consumo es significativo, se probó alimentar a la tarjeta Arduino con un adaptador de 12 Vcc y 1.5 A, pero el regulador de la tarjeta se sobrecalentaba, por lo que fue necesario emplear un regulador de 5 V y alimentar al microcontrolador directamente en las terminales marcadas con 5V y GND de la tarjeta Arduino.

5. Conclusiones

Arduino facilita el desarrollo de proyectos, sin embargo, al trabajar solo en ese nivel de programación, muchos detalles del microcontrolador quedan ocultos al desarrollador. Es importante comprender su organización para un mejor aprovechamiento de sus recursos.

La integración de rutinas con código ensamblador de una forma casi transparente, hace que las tarjetas Arduino se puedan utilizar en proyectos con requerimientos explícitos de temporización o simplemente, para conocer y evaluar las instrucciones AVR. A nivel docencia este aspecto resulta fundamental porque hace posible la enseñanza a distancia, las tarjetas Arduino son muy fáciles de conseguir y las prácticas se pueden realizar sin requerimientos adicionales de hardware o software.

La forma en que los LED's WS2812 se estructuran facilita mucho el crecimiento de paneles o arreglos de LED's con diferentes organizaciones. Las restricciones en su manejo son: el consumo de corriente, la memoria y la velocidad de refresco, se requiere aproximadamente de 8 ms para enviar la información a un panel con 256 LED's, pero, pensando en una posible animación, para determinar el número de cuadros por segundo se debe considerar el tiempo requerido durante el llenado de la memoria SRAM.

El manejo de estos LED's resultó un reto atractivo e interesante, es cierto que en internet se pueden encontrar bibliotecas con funciones para su manejo, sin embargo, este trabajo hace posible el desarrollo de aplicaciones sin el uso de esas bibliotecas o facilita su comprensión, para una posible adecuación u optimización.

Referencias

[1] Arduino – Home, https://www.arduino.cc/, 2014, última visita en abril de 2021.
[2] Digital RGB LED, http://www.world-semi.com/solution/list-4-1.html, 2016, última visita en abril de 2021.
[3] ATmega328P 8-bit AVR Microcontrollers, https://www.microchip.com/wwwproducts/en/ATmega328P, 2016, última visita en abril de 2021.
[4] Santiago F. *"Los Microcontroladores AVR de ATMEL",* Universidad Tecnológica de la Mixteca, México, 1a. Edición, 2014.
[5] Danescu R., "Working with the AVR Assembly language", Technical University of Cluj-Napoca, Disponible en: http://users.utcluj.ro/~rdanescu/dmp-lab06_new.pdf, última visita en abril de 2021.
[6] Bluetooth Electronics, https://www.keuwl.com/apps/bluetoothelectronics/, 2015, última visita en abril de 2021.

Diseño Colaborativo en Mecatrónica, Capítulo 11, pp. 141 - 153
ISBN: 978-607-9394-23-3, 2021

Detección de movimiento mediante procesamiento de imágenes aplicada a un sistema de seguridad

Aldo-Francisco Muñoz-Vargas[1], Juan-Manuel Ramos-Arreguín✉, Saúl-Tovar Arriaga, Marco-Antonio Aceves-Fernández, Jesús-Carlos Pedraza-Ortega, Edgar Alejandro Rivas Araiza

Universidad Autónoma de Querétaro, Facultad de Ingeniería
✉jsistdig@yahoo.com.mx, [1]aldomuoz98@gmail.com

Resumen

Este desarrollo presenta la solución a un problema de un sistema de seguridad en etapa de prototipo, aplicando visión por computadora con OpenCV. El prototipo es capaz de tomar fotografías cada vez que el sensor infrarrojo detecta el movimiento de alguna persona u objeto en su rango de operación. El sistema de vigilancia está formado por una tarjeta de desarrollo Raspberry Pi 3B+, y un sensor de movimiento PIR. Se toma fotografías siempre que se detecta movimiento, y en su primera versión, se tienen falsos positivos, en especial después de que el sensor envió una señal de cambio, tomando fotografías sin que exista movimiento, por lo que se almacenan fotos sin el requisito adecuado de movimiento. Debido a esto, se mejora el funcionamiento aplicando visión por computadora, aplicando un método de substracción de fondo y algunas adecuaciones mediante programación para obtener una solución funcional reduciendo el número de falsos positivos. Además, se analizan y comparan algunos otros métodos de detección de movimiento para determinar cuál es más adecuado, tales como MOG2 y KNN.

Palabras clave: Sistema de seguridad, visión por computadora, detección de movimiento, substracción de fondo, OpenCV, Raspberry Pi.

1. Introducción

1.1 Antecedentes

Algunos antecedentes de sistemas similares, tanto comerciales como prototipos desarrollados en sistemas embebidos como es el caso del sistema con el que se trabaja en este proyecto son presentados a continuación.

En 2016 Hossen et al. desarrollan un sistema de vigilancia basado en detección y estimación de movimiento usando flujo óptico [1], basado en el algoritmo Horn-Schunck, resultando en un sistema computacionalmente rápido sin requerir de hardware especial para el procesamiento de las imágenes. También se presenta un sistema de vigilancia por parte de Jayakumar y Muthulakshmi [2], un sistema embebido implementado en una tarjeta Raspberry Pi y usando IoT para almacenamiento de fotografías y envío de alertas; se usa detección de rostros para saber si la persona está autorizada o no. Yang et. al. presentan un sistema de vigilancia para interiores, con la capacidad de detección múltiple de personas, seguimiento y análisis de comportamiento [3].

En 2018 Pawlenka et al. desarrollan un sistema de seguridad considerando el incremento de la automatización del día a día, al estar este basado en un microcontrolador de un solo chip y sensores de

para detectar el acceso no autorizado a un área basados en contacto magnético, medición de monóxido de carbono, detección de movimiento y medición de temperatura y humedad; todo este conjunto de sensores justificados debido a que las casas inteligentes requieren de una manera de obtener información de su entorno. En este desarrollo un sensor PIR (detector de infrarrojos) fue utilizado como detector de movimiento. El procesamiento de las señales fue realizado mediante una red de microcontroladores y comunicación con protocolos como RS232 y RS485 [4].

En 2019 se presentó un sistema de seguridad físico que utiliza internet de las cosas mediante una plataforma llamada *Zolertia Remote*, tratando de dar una alternativa al de vigilancia. Consta de tres secciones, una red de sensores inalámbrica, un servidor privado *MQTT* (*Message Queue Telemetry Transport*) y una aplicación *Android* [5].

En 2020 se llevó a cabo una aplicación similar a las anteriores en la que mediante una *Raspberry Pi* como servidor se controló un conjunto de cámaras a través de una aplicación en *Android*, este sistema fue desarrollado de modo que en caso de detección de cambio del entorno se tomarían imágenes o videos guardándose en la memoria de la *Raspberry* y enviando un correo de alerta o notificaciones en la aplicación móvil, siendo estos datos accesibles a través de la aplicación que consultaría los datos del servidor [6].

De igual forma, en 2020 como trabajo de Zong Chen, un sistema de seguridad basado en sensores de movimiento e identificación de rostros fue desarrollado para detectar actividad sospechosa y reportar al posible infractor, este desarrollo utiliza Python, una Raspberry Pi, una base de datos de rostros y un sensor piroeléctrico, siendo probado en diferentes situaciones grabando video de las situaciones de prueba a las que se sometió el dispositivo [7].

Por otro lado, existen sistemas comerciales como es el caso de los dispositivos de la marca SEEDDARY (entre otras), los cuales suelen constar de una cámara con capacidad para visión diurna y nocturna, detección de movimiento, ajuste de ángulo de visión de la cámara dependiendo del modelo, grabación de video y vigilancia en tiempo real, así como opciones para consultar tales como son aplicaciones móviles y en el navegador [8]. Algo importante a destacar es que, aunque este tipo de sistemas es completamente funcional, estos son cerrados y no permiten la integración o experimentación con capacidades nuevas tales como podría hacerse mediante el uso de una herramienta como OpenCV o algún otro software, lo cual se busca en este desarrollo. Además, un sistema abierto como es en el que se desarrolló este prototipo, puede ser útil como sistema central, de monitoreo y posible control de factores ambientales dentro de un área específica.

1.2 Funcionamiento previo del dispositivo

En la figura 1, se puede observar el funcionamiento general del sistema de seguridad. La pieza central es la tarjeta Raspberry Pi 3B+, que tiene una comunicación continua con el dispositivo móvil Android. Este dispositivo se usa para configurar el funcionamiento del sistema de seguridad desde la aplicación móvil, la cual permite activar o desactivar el sistema y tomar capturas a voluntad, siempre y cuando se encuentre en la misma red local. La nube MEGA se usa para almacenar las imágenes capturadas y permite la consulta de las mismas desde cualquier lugar con una conexión a internet. Al detectar el sensor infrarrojo una presencia, se envía una señal para la captura de imágenes. Cuando lo que provocó el cambio desaparece del rango del sensor, se deja de enviar la señal. Como se observa en el diagrama, cada imagen registrada es nombrada con la fecha de captura para facilitar al usuario información sobre la misma.

El prototipo real se muestra en la figura 2, el diseño de la carcasa fue propuesto y diseñado de acuerdo a las necesidades, que en este caso eran poder ajustar el ángulo de visión de la cámara y el sensor, así como un diseño compacto [9]. En la tabla 1 se muestran los materiales que componen al sistema completo.

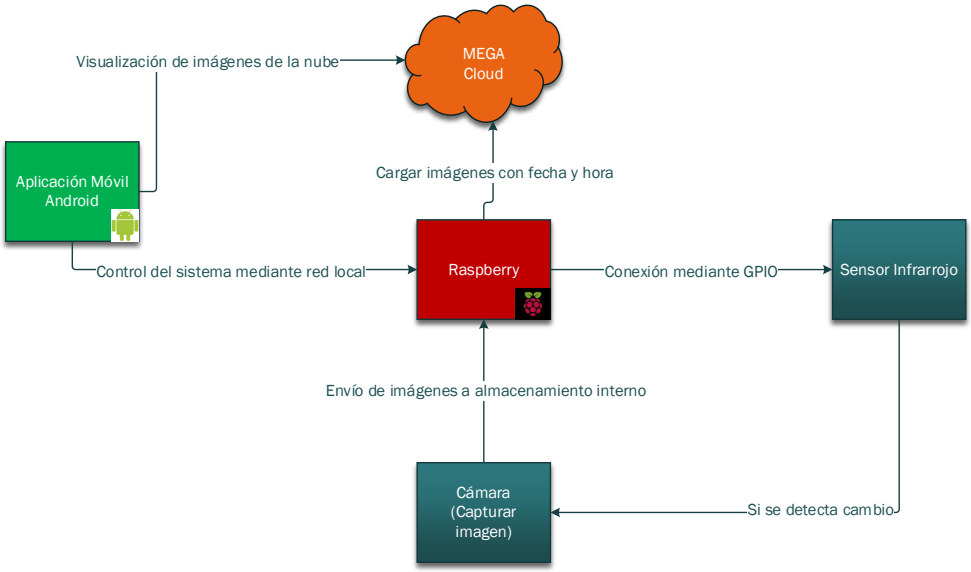


Figura 1. Funcionamiento general del sistema de seguridad.

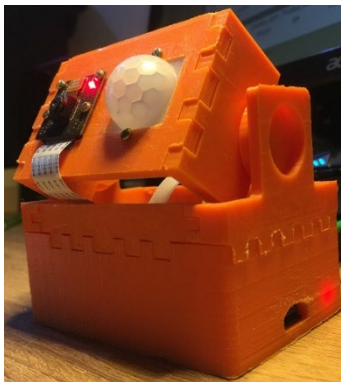


Figura 2. Carcasa impresa con los elementos de hardware.

Tabla 1. Tabla de materiales.

Descripción	Cantidad
Raspberry Pi 3B+	1
NoIR Camera	1
PI Camera	1
5 V DC Power Supply	1
32 GB SD card	1
PIR sensor	1
Case	1

Es importante mencionar que la cámara que se usa tiene capacidades para visión nocturna. Sin embargo, las imágenes capturadas en condiciones de luz de día no son fieles en cuanto los colores. La cámara utilizada es propia de Raspberry siendo esta el modelo NoIR. Además, una lámpara de luz infrarroja es utilizada de manera independiente, y solo se activa en condiciones de baja iluminación, proveyendo de la radiación infrarroja necesaria para que la cámara pueda capturar información. En la

figura 3 y 4 se muestran ejemplos de captura del sistema en las dos condiciones anteriormente mencionadas, con luz visible e infrarroja, en esta última sin luz visible o en condiciones de obscuridad.

Figura 3. Imagen en condiciones de luz visible.

Figura 4. Imagen en condiciones de luz infrarroja.

Finalmente, se muestra la interfaz diseñada para la aplicación en Android en la figura 5, presentando imágenes capturadas. En el título de cada imagen se muestra la fecha y hora. En la figura 6 se muestran dos capturas de la aplicación en Android, donde una persona fue captada caminando frente al sistema mientras estaba funcionando.

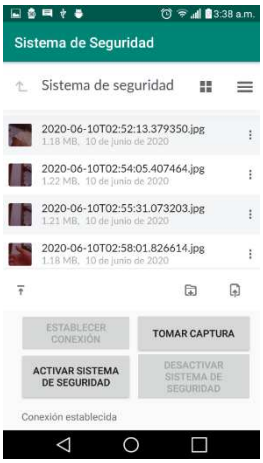


Figura 5. Aplicación móvil en funcionamiento mostrando imágenes como archivos en la nube.

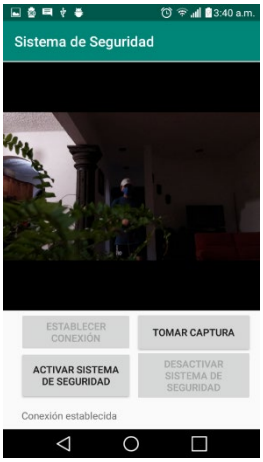


Figura 6. Aplicación móvil en funcionamiento mostrando una imagen desde la nube.

1.3 Problemática

El sensor infrarrojo PIR envía una señal cuando un objeto provoca un cambio en su elemento primario. Sin embargo, también enviaba la señal de detección de movimiento cuando no existía algún tipo de perturbación en el rango de operación, especialmente en momentos cercanos a una detección correcta, resultando en toma de imágenes sin información útil para el usuario, utilizando espacio de memoria tanto interna de la Raspberry como de la nube.

En la figura 7, se observa una captura de la muestra de imágenes tomadas con información que no correspondía con algún evento de detección de movimiento y algunas imágenes correctas, es posible notar algunas personas pasar, pero también imágenes sin ningún tipo de sujeto; esta captura es tomada del sistema de archivos generado por el sistema.

Figura 7. Muestra de imágenes del sistema de archivos del sistema de seguridad.

El sistema operó de una mejor forma en otras pruebas, teniendo una correspondencia clara entre las imágenes capturadas con algún objeto o sujeto en la imagen y el total de las imágenes capturadas, pero el caso mostrado en la figura 7 muestra uno de los peores escenarios que se presentaron, independientemente de ello, se puede observar y a través de más pruebas que se realizaron que era

totalmente seguro que si algún objeto activaba el sensor infrarrojo, se tomaría una imagen. Esta es la base para el desarrollo utilizando detección de movimiento con procesamiento de imágenes para mejorar la confiabilidad del sistema.

2. Metodología

De acuerdo al funcionamiento explicado en la sección anterior, se partió del hecho de que era seguro que el sensor detectase si algo se movía en su rango de operación, por lo que, a partir de ello, sería posible desarrollar un algoritmo mediante el cual, se analizaran una serie de imágenes consecutivas y así determinar si efectivamente algo se movía, tomando la decisión de capturar las imágenes.

2.1 Etapas de desarrollo

Investigación de algoritmos de detección de movimiento. En esta etapa se investigaron varias formas de detectar movimiento en una serie de imágenes consecutivas.

Captura de imágenes. Es necesario capturar imágenes a una tasa de fotogramas constante que permita el análisis utilizando Python y OpenCV, por lo que se investigó y experimentó una forma de lograr este objetivo a través de funciones disponibles para las herramientas mencionadas.

Prueba de los algoritmos investigados. Se probaron algunos de los métodos de substracción de fondo y se concluye que la mejor opción es substracción de fondo, considerando los requisitos del prototipo y su bajo costo computacional.

Implementación del algoritmo de detección de movimiento en la Raspberry Pi. Una vez conocido el funcionamiento y comportamiento del algoritmo de detección de movimiento, se modificó para que operara como una función de entrada y salida. Para esto, se reciben dos parámetros en la entrada, el primero es la imagen de fondo y el segundo es la imagen contra la cual comparar. La salida, que devuelve una respuesta booleana en caso de detectar o no movimiento. Debido a las características operativas del código (secuencial), se decide que la imagen de fondo debe ser tomada en el instante en que el sensor de movimiento detecte un cambio, debido a que el rango de detección del sensor es mayor que el alcance de visión de la cámara. Por lo tanto, la imagen obtenida como base queda sin ningún objeto o sujeto en la imagen. Cada imagen siguiente es analizada por el algoritmo de detección de movimiento comparándola con la imagen de fondo.

Ajustes de la integración del algoritmo de detección de movimiento: Una de las grandes limitaciones del algoritmo utilizado es su susceptibilidad a fallas en caso de cambios de iluminación. Como primera consideración para resolver este problema, se implementa en la Raspberry Pi la detección de movimiento, tomando la imagen sin cambios en el instante en que el sensor detecta movimiento. Sin embargo, debido a cómo se integra el algoritmo en Python, si ocurre un cambio drástico en la iluminación, se interpreta como un movimiento, capturando imágenes consecutivas sin detenerse hasta saturar la memoria. Para evitar esta saturación, se programa un límite haciendo que este proceso se detenga luego de cierta cantidad de imágenes. Esto es conveniente si lo que dispara el sistema, se mantiene mucho tiempo frente a él (mayor que el número de cuadros permitidos). Tan pronto como este objeto o sujeto se mueve, las imágenes son capturadas nuevamente, y así hasta que el fondo se mantenga constante.

A continuación, se presentan algunos métodos de detección de movimiento que fueron considerados en la implementación de esta solución.

2.2 Métodos de detección de movimiento mediante visión por computadora.

Substracción de fondo: La substracción de fondo es un paso de preprocesamiento importante en muchas aplicaciones basadas en la visión. Por ejemplo, un contador de visitantes donde una cámara

estática toma el número de personas que entran o salen de una zona, o una cámara de tráfico que extrae información sobre los vehículos. En todos estos casos, primero debe extraer a la persona o vehículos solos.

Si se tiene una imagen del fondo solo, como una imagen de la zona sin visitantes o una imagen de la carretera sin vehículos simplemente se resta la nueva imagen del fondo obteniendo imágenes solo de los objetos sobre el fondo. Sin embargo, es posible que no se tenga una imagen de este tipo, por lo que se deben utilizar otros métodos. Incluso objetos como sombras hacen más complejo el análisis dado que la sombra también se está moviendo, una simple resta marcará eso también como primer plano [10].

Substracción de fondo MOG y MOG2: MOG es un algoritmo de segmentación de fondo o primer plano basado en una mezcla gaussiana. Utiliza un método para modelar cada píxel de fondo mediante una mezcla de distribuciones K gaussianas (K = 3 a 5). Los pesos de la mezcla representan las proporciones de tiempo que esos colores permanecen en la escena. Los probables colores de fondo son los que permanecen más tiempo y más estáticos [10].

MOG2 también es un algoritmo de segmentación de fondo o primer plano basado en mezclas gaussianas. Una característica importante de este algoritmo es que selecciona el número apropiado de distribución gaussiana para cada píxel [10]. Este último es uno de los algoritmos probados en el sistema como se mostrará, a diferencia de MOG con el que se tuvo dificultades de implementar debido a incompatibilidad con versiones de OpenCV.

Substracción de fondo KNN: Se basa en el algoritmo de los k-vecinos más cercanos (k-NN). La entrada consta de los k ejemplos de entrenamiento más cercanos en el espacio de características. El resultado depende de si se utiliza k-NN para clasificación o regresión [11]:

- En la clasificación k-NN, la salida es una pertenencia a una clase. Un objeto se clasifica mediante un voto de pluralidad de sus vecinos, y el objeto se asigna a la clase más común entre sus k vecinos más cercanos (k es un número entero positivo, típicamente pequeño). Si k = 1, entonces el objeto simplemente se asigna a la clase de ese único vecino más cercano.

- En la regresión k-NN, la salida es el valor de propiedad del objeto. Este valor es el promedio de los valores de k vecinos más cercanos.

3. Resultados

3.1 Comparación entre métodos de detección de movimiento

Tres métodos fueron probados, aunque otros dos se mencionaron anteriormente, esto debido a errores con OpenCV en la Raspberry Pi. Todos los resultados presentados a continuación se probaron con condiciones de luz visible e infrarroja.

Método de sustracción de fondo: Para este método los pasos seguidos fueron los siguientes:

- Conversión de escala de grises y eliminación de ruido.
- Operación de resta entre el fondo y el primer plano.
- Aplicar un umbral a la imagen resultante de la resta.
- Detección de contornos o manchas.

La ecuación general que representa la substracción de fondo se presenta a continuación:

$$O_{img} = A_{img} - S_{img} \tag{1}$$

Donde O_{img} representa la imagen resultada debido a la substracción, A_{img} es la imagen con alteraciones respecto a S_{img} y S_{img} es la imagen del fondo estático.

Cabe mencionar que este es el método que se decidió utilizar para la operación final, esto es debido a que la primera imagen tomada o fondo funciona excelente para la aplicación, cada vez que el sensor detectaba algún movimiento el algoritmo comenzaba a funcionar, finalmente, la decisión de si se debe tomar una imagen o no depende de la imagen umbral, además de eso, los otros métodos están destinados a casos en los que el fondo no es constante o no es posible obtener una imagen del fondo.

Una imagen captada por el prototipo en una zona sin ningún tipo de cambio es la que se observa en la figura 8, siendo esta una puerta y una pared.

Figura 8. Imagen de fondo.

Una vez que se activa el sistema de seguridad y se ha detectado un cambio, se realiza el procesamiento para determinar si hay algo en la imagen, como se verá a continuación. La figura 9 muestra una mano, que desencadena el proceso de análisis, como imagen de referencia se utiliza la imagen de la figura 8.

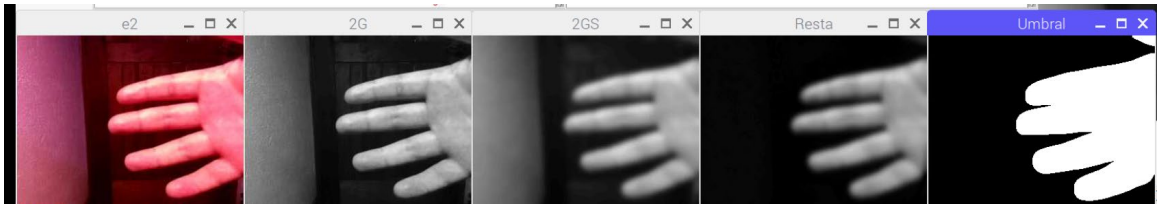


Figura 9. Proceso de detección de movimiento utilizando substracción de fondo en condiciones de luz visible.

En la figura 9 se observa el proceso que ocurre en el algoritmo, en la primera imagen de izquierda a derecha se observa la imagen RGB, luego se observa la imagen en escala de grises, el siguiente paso es aplicar un filtro gaussiano para reducir detalles innecesarios , la cuarta fase es la resta entre la imagen de fondo original (que también pasó por los pasos antes mencionados) y la imagen con el cambio resultante en la cuarta imagen, finalmente, después de una operación de umbralización, una imagen con solo dos tonos, blanco o negro es obtenida, así al encontrar un píxel en blanco es suficiente para inferir detección de movimiento y por lo tanto guardar la imagen RGB original en la memoria. Para eso, un bucle que analiza varios píxeles en toda la imagen decide con una condición; se podría hacer analizando cada píxel, pero para hacer el proceso menos intensivo computacionalmente, se analizaron solo partes de la matriz de píxeles. El mismo proceso se observa en la figura 10, pero con luz infrarroja.

Figura 10. Proceso de detección de movimiento utilizando substracción de fondo en condiciones de luz infrarroja.

Método de substracción de fondo MOG2: Para este método, se utilizó la función que estaba incluida en OpenCV. La integración con el sistema de seguridad fue sencilla, aunque no hubo control de los parámetros como en el método de sustracción de fondo. También se encontraron los contornos del objeto para que se pudiera dibujar un rectángulo alrededor de la imagen para indicar dónde estaba la diferencia, este proceso también se implementó para los resultados finales. La imagen de la izquierda en la figura 11 llamada "Umbral" es el resultado de este método.

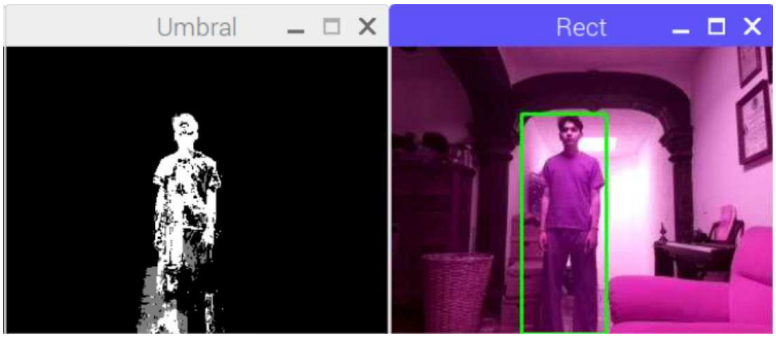


Figura 11. Detección de movimiento utilizando substracción de fondo MOG2 en condiciones de luz visible.

En la figura 12, se presenta el mismo resultado con luz infrarroja, nótese que este método también es susceptible a la detección de sombras como objetos adicionales.

Figura 12. Detección de movimiento utilizando substracción de fondo MOG2 en condiciones de luz infrarroja.

Método de substracción de fondo KNN: La forma en que se implementó es la misma que la anterior. En las figuras 13 y 14 se pueden ver los resultados en condiciones de luz visible e infrarroja.

Figura 13. Detección de movimiento utilizando substracción de fondo KNN en condiciones de luz visible.

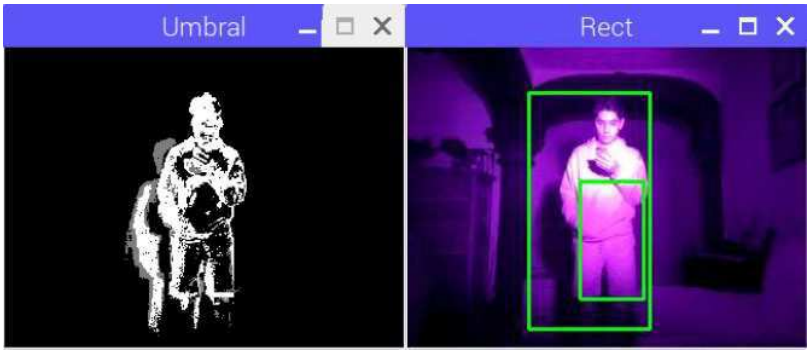


Figura 14. Detección de movimiento utilizando substracción de fondo KNN en condiciones de luz infrarroja.

Se puede notar que los dos últimos métodos son bastante similares en resultados, aunque no se puede mostrar en este texto, cuando el sujeto en las imágenes se mantuvo quieto, ambos algoritmos comenzaron a hacer como si este desapareciera, mientras que el método de substracción de fondo puro no, esa es otra razón de que sea el método a implementar en el prototipo final, sin embargo, cualquier método podría ser útil para esta aplicación ya que el seguimiento no es relevante para este proyecto.

3.2 Resultados finales

En las siguientes figuras, el algoritmo de substracción de fondo se presenta nuevamente, en la ubicación donde se probaron los otros dos métodos MOG2 y KNN, también, se agregó el rectángulo de seguimiento.

En la figura 15 se muestra la implementación mediante Python de tres barras utilizadas para determinar los parámetros que funcionan bien para este proceso. La primera cambia el primer parámetro del filtro gaussiano, el parámetro "ksise", la segunda el parámetro “sigma” que es parte del filtro de desenfoque gaussiano y la tercera barra “Thresh” cambia el umbral de lo que se considera parte del objeto en movimiento y lo que no.

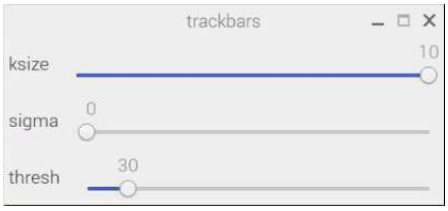


Figura 15. Barras para configurar parámetros del algoritmo de detección de movimiento.

La conclusión de la calibración de estos parámetros fue que "ksise" debe ser alto como se presenta en la última imagen, en realidad tiene un valor de 21 mientras sigma es cero, esto hace que la imagen con filtrado gaussiano sea extremadamente borrosa, que es lo mejor para este método, de modo que cuando alguien entra en el rango de visión, puede ser tratado como un solo objeto.

Para el valor de umbral, se determinó que alrededor de 30 de 255 era lo suficientemente bajo para esta aplicación.

Los resultados con condiciones de luz visible se presentan en la figura 16.

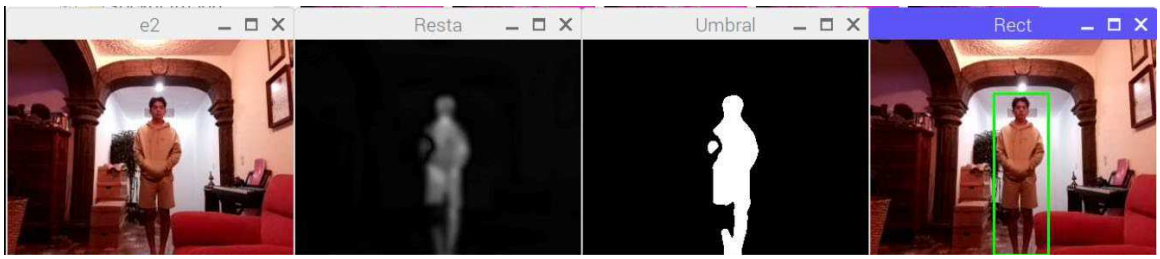


Figura 16. Resultados en condiciones de luz visible.

Por otro lado, con condiciones de luz infrarroja, el resultado es mostrado en la figura 17.

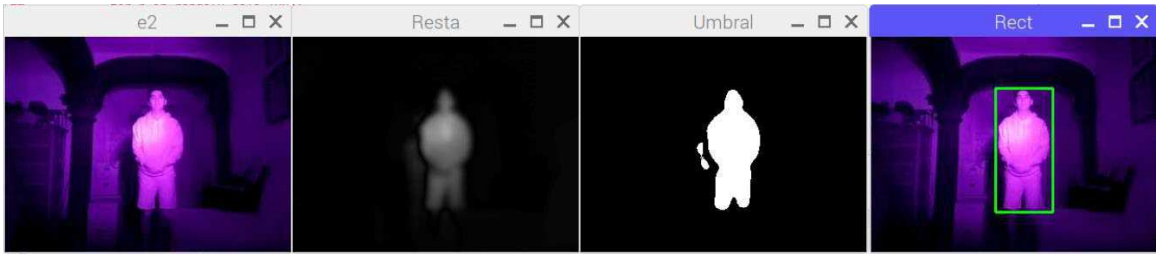


Figura 17. Resultados en condiciones de luz infrarroja.

Cada vez que se detecta un píxel blanco en la imagen “Umbral” (figura 16 y 17), el código almacena el último cuadro correspondiente a movimiento, también debe tenerse en cuenta que ninguna de las imágenes anteriores está planeada para ser observada por el usuario, estas se muestran solo con fines ilustrativos, el usuario puede ver la imagen contenida en la ventana “rect” únicamente.

Cada imagen se almacena en una carpeta con la fecha, y cada imagen también tiene la fecha en que se tomó como nombre, como se observa en la figura 5.

Comparación con el prototipo sin detección de movimiento: En la figura 18, se tiene una gráfica del funcionamiento del prototipo sin detección de movimiento por aproximadamente 24 horas, en el eje horizontal se tiene el tiempo de funcionamiento, mientras en el vertical, eventos de captura de imágenes, cada punto verde representa un dato donde la imagen muestra información que corresponde con detección de movimiento por parte del sensor, por otro lado, cada punto rojo corresponde con imágenes capturadas en total, cada punto representa el número de imágenes capturadas en un intervalo de una hora, por lo tanto el número de imágenes erróneas es la diferencia entre el número total de imágenes capturadas menos el número de imágenes correctas.

Por lo tanto, se puede observar en la figura 18, que el sistema, aunque cumple con su objetivo de registrar imágenes de algún objeto o sujeto que cruzó en su campo de operación, también captura imágenes en momentos donde no existió ningún tipo de evento, debe señalarse que esta gráfica

corresponde con el funcionamiento en una de las peores condiciones, sin embargo, este solía entregar imágenes que sí correspondían con movimiento. Se sospecha que este hecho podría estar relacionado con algún tipo de interferencia con señales electromagnéticas debido al tipo de cableado utilizado en el prototipo al no haber utilizado circuitos impresos como medio de conexión entre el sensor y la tarjeta embebida.

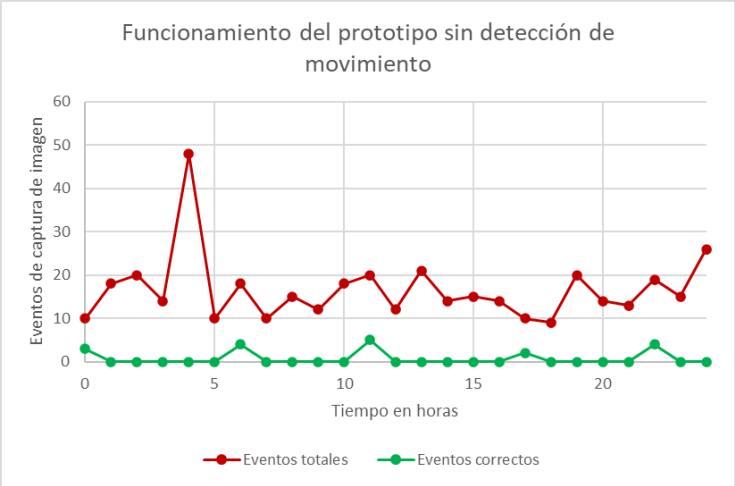


Figura 18. Funcionamiento del prototipo anterior.

En contraste, una prueba de funcionamiento de alrededor de 24 horas fue realizada empleando la substracción de fondo para la detección de movimiento, esta vez con resultados satisfactorios al capturar imágenes debidas a movimiento en el rango de operación, lo anterior se describe en la figura 19, donde se aprecian eventos de captura de imágenes correctos. En el evento cercano a 40 capturas de imágenes se puede observar que existen algunas imágenes tomadas extra con respecto al número de eventos correctos, esto fue debido al cambio de posición de un objeto del fondo, por lo que al ser comparada la imagen con el objeto en una posición diferente contra la imagen con el fondo estático se obtuvo una diferencia.

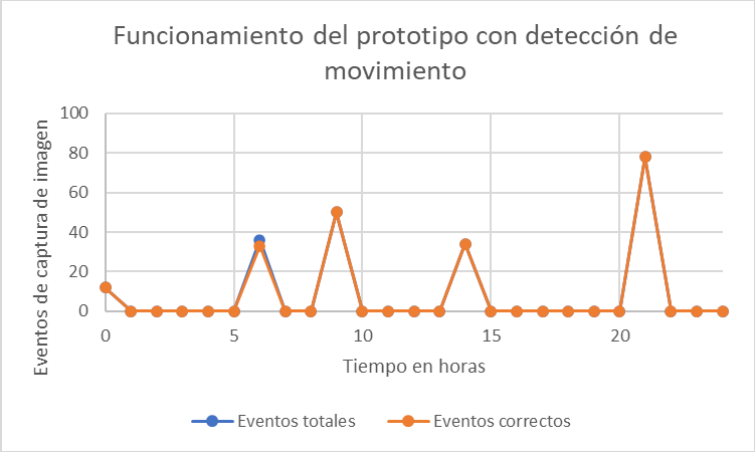


Figura 19. Funcionamiento del prototipo después de la implementación del algoritmo de detección de movimiento.

4. Conclusiones

Si bien el método elegido no es perfecto por las condiciones de luz, al tener problemas con las sombras, la forma no es tan importante para el sistema de vigilancia, con solo poder detectar un píxel blanco como se explicó anteriormente, es suficiente para tomar imágenes, esto hace el prototipo más confiable que la versión anterior. Además, al ser un dispositivo desarrollado en software libre (OpenCV y Python) y una plataforma como es Linux, es totalmente viable, realizar mejoras o emplear el prototipo en alguna otra aplicación donde se requiera de detección de movimiento. Finalmente, disponiendo aun de entradas y salidas digitales, otro tipo de sensores podrían instalarse al dispositivo ofreciendo mayor información acerca de un área.

Referencias

[1] M. K. Hossen y S. H. Tuli, “A surveillance system based on motion detection and motion estimation using optical flow”, en *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, may 2016, pp. 646–651. doi: 10.1109/ICIEV.2016.7760081.

[2] A. J. K. Jayakumar y S. Muthulakshmi, “Raspberry Pi-Based Surveillance System with IoT”, en *Intelligent Embedded Systems*, Singapore, 2018, pp. 173–185. doi: 10.1007/978-981-10-8575-8_19.

[3] C. Yang, T. Chou, F. Chang, C. Ssu-Yuan, y J. Guo, “A smart surveillance system with multiple people detection, tracking, and behavior analysis”, en *2016 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, abr. 2016, pp. 1–4. doi: 10.1109/VLSI-DAT.2016.7482569.

[4] T. Pawlenka y J. Škuta, “Security system based on microcontrollers”, en *2018 19th International Carpathian Control Conference (ICCC)*, may 2018, pp. 344–347. doi: 10.1109/CarpathianCC.2018.8399653.

[5] A. Cunalata y D. Alberto, “Desarrollo de un prototipo de sistema de seguridad contra intrusos utilizando protocolos de IoT sobre la plataforma Zolertia Remote”, jul. 2019, Consultado: oct. 14, 2020. [En línea]. Disponible en: http://bibdigital.epn.edu.ec/handle/15000/20318

[6] J. Marín Rodríguez, “Desarrollo de un sistema de video vigilancia con servidor VPN Raspberry Pi y APP para móvil.”, may 2020, Consultado: oct. 14, 2020. [En línea]. Disponible en: https://riunet.upv.es/handle/10251/143050

[7] Dr. J. I. Zong Chen, “Smart Security System for Suspicious Activity Detection in Volatile Areas”, *JITDW*, vol. 02, núm. 01, pp. 64–72, mar. 2020, doi: 10.36548/jitdw.2020.1.006.

[8] “Seedary | Tienda Online”. https://seedaryoficial.mercadoshops.com.mx/ (consultado abr. 29, 2021).

[9] Muñoz A., Ramos J., et al. "*Sistema de seguridad basado en un sistema embebido con cámara para día y noche con enlace IoT*”, La Mecatrónica en México, México, Vol. 10 No. 1 Consultado: abril 25, 2021. [En línea]. Disponible en: http://www.mecamex.net/revistas/LMEM/revistas/LMEM-V10-N01-03.pdf

[10] “Background Subtraction — OpenCV 3.0.0-dev documentation”. https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html (consultado dic. 27, 2020).

[11] “Detección de movimiento con OpenCV y Python”, *Programar fácil con Arduino*. https://programarfacil.com/blog/vision-artificial/deteccion-de-movimiento-con-opencv-python/ (consultado dic. 27, 2020).

Diseño Colaborativo en Mecatrónica, Capítulo 12, pp. 154 - 166
ISBN: 978-607-9394-23-3, 2021

Análisis de la marcha humana implementando la cinemática de las articulaciones de la cadera y la rodilla

Lezama-Quintero Luis Arturo*, Gómez-Pavón Luz del Carmen, Arriaga-Arriaga Cesar Augusto

Maestría en Ingeniería Electrónica, Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y Av. 18 Sur, Edif. FCE6, Ciudad Universitaria, Col. San Manuel, Puebla, Pue., México.
*luis.lezamaquin@alumno.buap.mx

Resumen

En este artículo se presenta el análisis del ciclo de la marcha bípeda humana, en las articulaciones de la cadera y la rodilla. Se presenta el modelo matemático de la cinemática directa e inversa de las extremidades inferiores solo tomando en cuenta la articulación de la cadera y de la rodilla, con el objetivo de determinar de manera gráfica el comportamiento que tienen ambas articulaciones en el ciclo de la marcha humana. El modelo de la cinemática directa se desarrolla utilizando el método de Denavit-Hartenberg para obtener las matrices de transformación homogéneas, por otro lado, el modelo de la cinemática inversa se desarrolla utilizando el método gráfico para obtener las ecuaciones que representen los ángulos de la articulación de la cadera y la rodilla.

Palabras clave: Extremidades inferiores, articulaciones, ciclo de marcha, cinemática, flexión, extensión.

1. Introducción

El ser humano es capaz de efectuar diversas actividades en las que involucra diversas partes del cuerpo, un ejemplo de ellas es la marcha humana, donde se implica principalmente las extremidades inferiores, las cuales se extienden desde la región glútea hasta los pies y están conectadas a la parte inferior del tronco. Las extremidades inferiores son complejas desde un punto de vista anatómico y biomecánico debido a los distintos músculos, nervios y ligamentos que los componen, los cuales permiten realizar distintos tipos de movimientos, pero están compuestas principalmente por 3 articulaciones, la cadera, la rodilla y el tobillo [1].

La articulación de cadera permite a las extremidades inferiores, moverse a través de los tres planos de referencia del cuerpo humano. Es una de las articulaciones más grandes y estables del cuerpo, posee una estabilidad intrínseca proporcionada por su configuración esférica relativamente rígida. Tiene una gran movilidad, que posibilita la locomoción normal en el desarrollo de las actividades diarias, por ello es conveniente definir las capacidades de movimiento con las que cuenta [2]. Debido a la configuración esférica de la cadera, esta puede tener movimientos en los 3 planos (plano sagital, plano frontal y plano transversal), sin embargo, en este trabajo se enfocará únicamente en el plano sagital, siendo movimientos de flexión y extensión. La cadera puede realizar movimientos de flexión en un rango de entre 0° hasta 120° o 140° y en extensión de 0° hasta 10° o 30° [3].

La articulación de rodilla soporta fuerzas y momentos elevados y se sitúa entre los brazos de palanca más largos del cuerpo (el fémur y la tibia), haciéndola particularmente susceptible a la lesión. El movimiento de la rodilla ocurre simultáneamente en los tres planos, pero prácticamente se desarrolla más en un sólo plano, representando todo el movimiento de la articulación. Por ello, los análisis biomecánicos básicos pueden limitarse al movimiento en el plano sagital [2]. Para la articulación de la rodilla se tiene un movimiento de flexión que abarca un rango de 0° hasta 135° o 150° y para la extensión se tienen valores de 0° a 10° [3].

La articulación del tobillo está formada por tres huesos: el peroné, la tibia y el astrágalo; esta articulación permite los movimientos de flexión-extensión del pie. Aunque frecuente el tobillo se modela en el análisis de la marcha como una estructura rígida que permite una estabilidad adecuada para mantener el peso del cuerpo, es importante considerar que las funciones del pie varían en función del terreno donde se actúa, debido a que puede actuar como resorte durante la transferencia de peso y como palanca durante el despegue [2]. En el caso del tobillo, se tiene una flexión de 0° a 50° y de 0° a 20° o 30° en extensión [3].

El análisis de la marcha bípeda ha sido de interés desde hace ya algún tiempo. Las investigaciones realizadas han permitido valorar los distintos parámetros de la marcha de forma objetiva y eficaz, apreciar los factores que pueden modificarla, diagnosticar alteraciones del patrón de marcha en diversas patologías y lesiones traumáticas, realizar un control y seguimiento de pacientes para observar su evolución y permitir desarrollar nuevos dispositivos que permitan apoyar a solucionar estas lesiones [4].

En este artículo se revisará el análisis de la marcha para las articulaciones de la cadera y la rodilla, se obtendrá el modelado cinemático directo y el modelado cinemático inverso para éstas, la razón de enfocarnos solo en el modelado cinemático es porque permite estudiar el movimiento del sistema, en este caso el comportamiento de las articulaciones de estudio, despreciando las fuerzas que originan dicho movimiento, por lo tanto no se involucran ecuaciones diferenciales como lo es en el caso de la dinámica, por último se mostrará de manera gráfica el comportamiento que toman ambas articulaciones al momento de caminar.

2. Análisis de la marcha

La marcha humana es un proceso de locomoción en el que nuestro cuerpo, estando de pie, se desplaza de un lugar a otro siendo su peso soportado de forma alternante por ambos miembros inferiores. Mientras el cuerpo se desplaza sobre la pierna de soporte, la otra pierna se balancea hacia delante como preparación para el siguiente apoyo. Uno de los pies se encuentra siempre en el suelo y en el período de transferencia de peso del cuerpo de la pierna retrasada a la adelantada, existe un breve intervalo de tiempo durante el cual ambos pies descansan sobre el suelo [5]. Un periodo de la marcha humana se divide en dos fases: la fase de apoyo y la fase de oscilación o balanceo.

La fase de apoyo es un intervalo de contacto del pie con el suelo, es posible subdividir esta fase en 5 momentos que son: contacto del talón, apoyo plantar, apoyo medio, elevación del talón y despegue del pie, en la figura 1 se pueden apreciar estos 5 momentos de la fase de apoyo [6, 7, 8, 9].

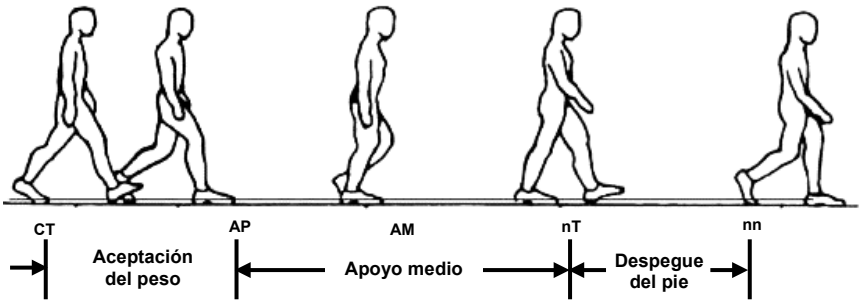


Figura 1. Subdivisiones de la fase de apoyo [6].

La fase de oscilación es un intervalo de ausencia de contacto del pie con el suelo o balanceo, esta fase se puede dividir en 3 intervalos que son: aceleración, balanceo medio y deceleración. Cada una de estas subdivisiones constituye aproximadamente un tercio de la fase de balanceo [10]. El primer

tercio se caracteriza por la rápida aceleración del extremo de la pierna inmediatamente después de despegar los dedos del suelo. Durante el segundo tercio, la pierna balanceada pasa a la otra pierna, moviéndose delante de la misma, ya que está en fase de apoyo y el último tercio se caracteriza por la deceleración de la pierna que se mueve rápidamente cuando se acerca al final del intervalo [6,11], lo expuesto anteriormente se puede apreciar en la figura 2.

Figura 2. Subdivisiones de la fase de balanceo [6].

Un periodo normal representa el 100% de marcha, cada fase tiene un porcentaje específico, de esta manera el 60% corresponde a la fase de apoyo y el restante 40% corresponde a la fase de oscilación [12]. Se debe tener presente que sumando los ciclos que se están produciendo de manera simultánea en ambos miembros inferiores, en algún momento, los pies se encuentran en contacto con el suelo, este nuevo periodo se lo denomina fase de doble apoyo, el cual en dos ocasiones se encuentra en la fase de doble apoyo, siendo la primera desde el 0% del ciclo hasta un 10% o 15 %, la segunda va de un 40% o 50% hasta el 65 %. Estas fases se pueden visualizar en la figura 3 en el cual se presenta el ciclo de la marcha [13].

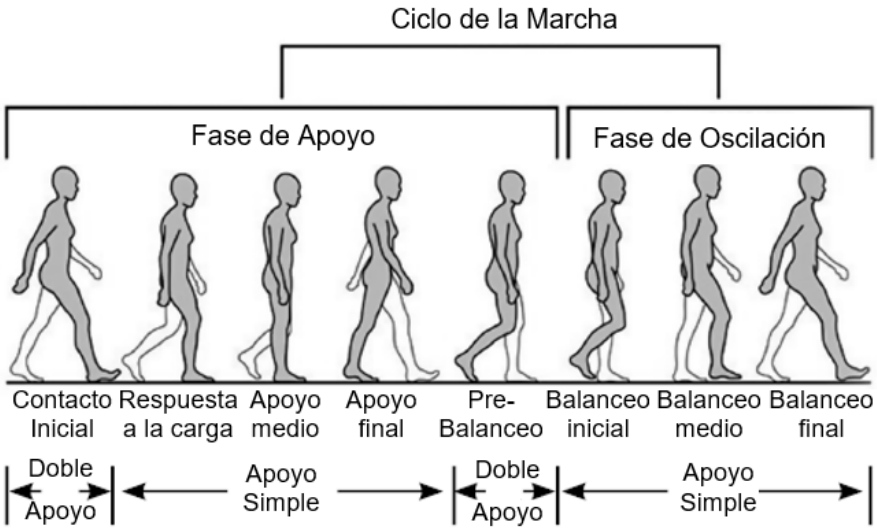


Figura 3. Ciclo de la marcha humana [12].

2.1 Articulación de la cadera

En la marcha humana, el comportamiento de la cadera desde el plano sagital empieza con el contacto inicial del talón con el suelo, en donde la cadera presenta un ángulo de 30° de flexión, posteriormente, la articulación comienza a extenderse hasta que la planta del pie este en contacto total con la superficie, disminuyendo la flexión a 10° y enseguida, se alinea de manera vertical con el torso del cuerpo hasta alcanzar su posición neutral de 0°. En el instante que el talón se separa del suelo, la cadera alcanza un ángulo máximo de hiperextensión de 20° y mantiene dicho ángulo hasta el despegue total de los dedos. Finalmente, durante la fase de oscilación la cadera parte de una posición neutra de 0° y se prepara para empezar el nuevo paso alcanzando su flexión inicial de 30° Ciclo de la marcha humana [14]. Lo antes descrito se puede apreciar en la figura 4.

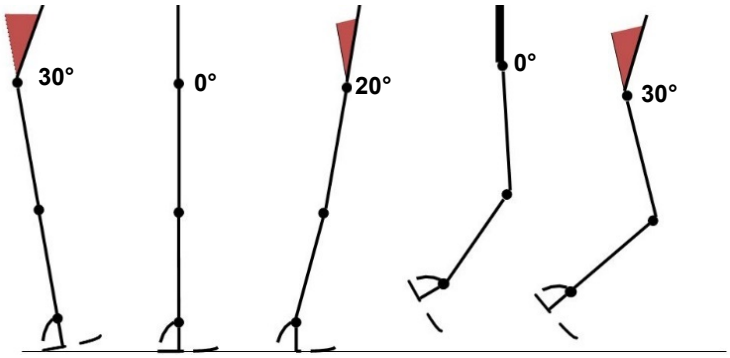


Figura 4. Ángulos de la articulación de la cadera en la marcha [3].

2.2 Articulación de la rodilla

El comportamiento de la rodilla comienza antes de entrar en contacto con el suelo se encuentra completamente extendida, posteriormente, el talón al entrar en contacto con la superficie comienza a flexionarse hasta que la planta del pie este totalmente plana alcanzando un ángulo de 20° de flexión y luego se extiende hasta posicionarse en 10° de flexión. Antes de que el talón pierda contacto con el suelo, la articulación se extiende completamente a 4° y al momento de elevar el talón y los dedos de la superficie, la rodilla se mueve rápidamente a 40° de flexión plantar. Al empezar la fase de oscilación, esta articulación llega a su ángulo de flexión máxima de 65° y al prepararse para el nuevo paso, la rodilla se va extendiendo rápidamente hasta llegar a su posición neutra de 0° [14]. Lo antes descrito se puede apreciar en la figura 5.

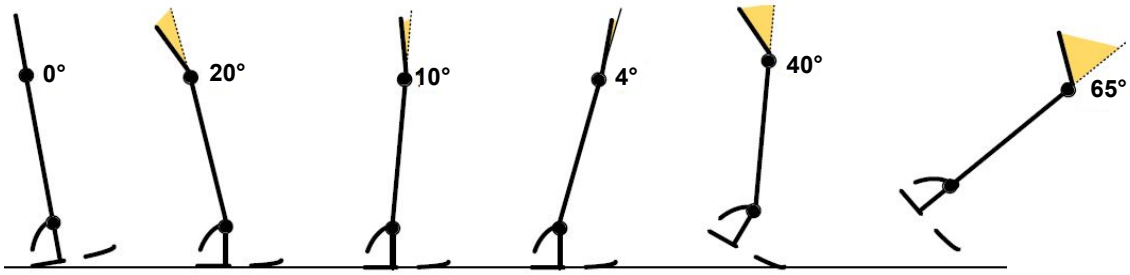


Figura 5. Ángulos de la articulación de la rodilla en la marcha [3].

3. Cinemática de los miembros inferiores

En términos generales la cinemática se encarga de la descripción del movimiento de los cuerpos, sin tomar en cuenta sus causas ni efectos sobre otros cuerpos. Es decir, no se involucran variables de fuerza o energía, únicamente desplazamientos, velocidades y aceleraciones [15].

Como una aplicación específica de ésta, se encuentra la cinemática del robot, que se encarga de describir el movimiento de un cuerpo controlable con varios grados de libertad y marcos de referencia, estableciendo las relaciones necesarias para migrar entre ellos. Dentro de ella, se subdividen la Cinemática directa y Cinemática inversa [15, 16].

3.1 Cinemática directa de cadera y rodilla

La solución de la cinemática directa consiste en encontrar el valor de la posición final del modelo, esta solución es una función de los valores articulares, es decir, el valor traslacional o rotativo de las articulaciones. Existen diferentes métodos para resolver este problema, para el desarrollo de este trabajo se hizo uso de las matrices de transformación homogénea. Una matriz de transformación homogénea se puede considerar que consiste en cuatro submatrices [17]:

$$T = \begin{bmatrix} R_{3x3} & p_{3x1} \\ f_{1x3} & k_{1x1} \end{bmatrix} = \begin{bmatrix} matriz\ de\ rotación & vector\ de\ posición \\ perspectiva & escalado \end{bmatrix} \quad (1)$$

De la ecuación 1 se puede apreciar que la submatriz superior izquierda (**R_{3x3}**) representa la matriz de rotación, la submatriz superior derecha (**p_{3x1}**) representa el vector de posición del origen del sistema de coordenadas rotado con respecto al sistema de referencia, la submatriz inferior izquierda (**f_{1x3}**) representa la transformación de perspectiva y el cuarto elemento diagonal (**k_{1x1}**) es el factor de escala global. Generalmente en aplicaciones de robótica, este factor de escala será siempre igual a 1 [18].

Para analizar la cinemática directa del sistema se utiliza el método de representación de Denavit-Hartenberg, el cual establece en forma sistemática un sistema de coordenadas para cada elemento de la cadena articulada. Dicha representación resulta en una matriz de transformación homogénea que representa cada uno de los sistemas de coordenadas de los elementos en la articulación con respecto al sistema de coordenadas del elemento previo, denominada matriz de transformación D-H. El método propuesto ofrece la ventaja de conocer tanto la posición final del sistema como la posición de cada una de sus articulaciones [17, 18].

En la figura 6 se presenta el modelo simplificado equivalente a una pierna humana, de la cual se deduce que es de tipo **RRR**, donde **R** se refiere al movimiento rotativo y está formado por elementos llamados articulaciones (**i_1, i_2, i_3**), eslabones (**a_1, a_2, a_3**) y ángulos (**θ_1, θ_2, θ_3**).

Partiendo de la figura 4 se obtiene la representación de Denavit-Hartenberg, mostrado en la tabla 1, la cual se reemplaza en la matriz de transformación homogénea, que consiste en un arreglo de 4x4 compuesta con información de la posición y orientación del sistema de referencia unido al **i** enlace del sistema en relación con el sistema de referencia de la conexión **i-1**.

Tabla 1. Parámetros de Denavit-Hartenberg.

i	**α_i**	**a_i**	**d_i**	**θ_i**
1	0	a_1	0	θ_1
2	0	a_2	0	θ_2

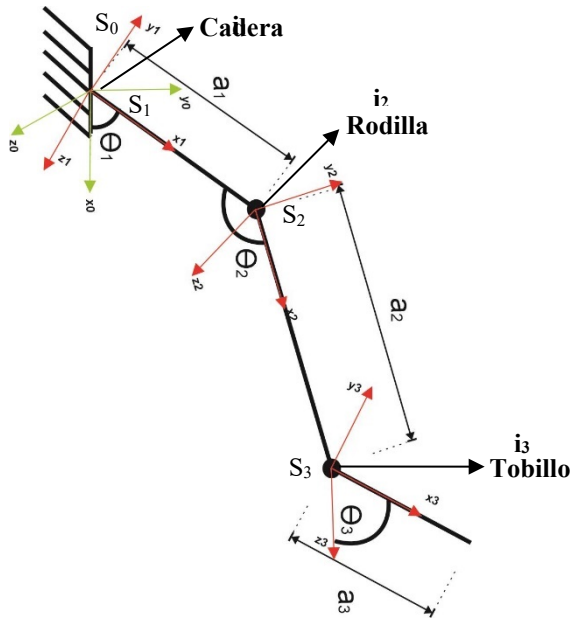


Figura 6. Diagrama de cuerpo libre de una pierna para la cinemática directa.

De esa manera, la primera matriz $\mathbf{A_1^0}$ representa la posición y orientación del sistema de coordenadas $\mathbf{S_1}$ del modelo con respecto al sistema de coordenadas $\mathbf{S_0}$, siendo $\mathbf{A_1^0}$ la articulación de la cadera. La segunda matriz $\mathbf{A_2^1}$ representa la posición y orientación del sistema de coordenadas $\mathbf{S_2}$ del modelo con respecto al sistema de coordenadas $\mathbf{S_1}$, siendo $\mathbf{A_2^1}$ la articulación que representa la rodilla. Con lo antes expuesto se obtiene tienen las siguientes matrices de transformación homogénea:

$$A_1^0 = \begin{bmatrix} \cos\theta_1 & -\operatorname{sen}\theta_1 & 0 & a_1\cos\theta_1 \\ \operatorname{sen}\theta_1 & \cos\theta_1 & 0 & a_1\operatorname{sen}\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$A_2^1 = \begin{bmatrix} \cos\theta_2 & -\operatorname{sen}\theta_2 & 0 & a_2\cos\theta_2 \\ \operatorname{sen}\theta_2 & \cos\theta_2 & 0 & a_2\operatorname{sen}\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Dónde:

$\boldsymbol{\theta_1}$: Angulo formado en la articulación de la cadera.
$\boldsymbol{\theta_2}$: Angulo formado en la articulación de la rodilla.

Una vez obtenidas las matrices de transformación del eslabón de la cadera, ecuación 2, y la rodilla, ecuación 3, se multiplican para encontrar la matriz de transformación homogénea $\mathbf{A_2^0}$, que va del sistema de referencia $\mathbf{S_0}$ (articulación de la cadera) al sistema de coordenadas $\mathbf{S_2}$ (articulación de la rodilla), obteniendo:

$$\mathrm{A}_2^0 = \mathrm{A}_1^0\mathrm{A}_2^1 \tag{4}$$

$$A_2^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\operatorname{sen}(\theta_1 + \theta_2) & 0 & a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2) \\ \operatorname{sen}(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & a_1 \operatorname{sen}\theta_1 + a_2 \operatorname{sen}(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

La ecuación 5 es un arreglo de dimensiones de 4x4 que representa el estado de las articulaciones de la cadera y la rodilla, como la rotación y la traslación, para este análisis cinemático solo nos interesa la traslación del sistema sobre el plano sagital. Cabe mencionar que la ecuación 5 se muestra de forma simplificada, esta simplificación se logró utilizando identidades trigonométricas de suma de senos (ecuación 6) y cosenos (ecuación 7).

$$\operatorname{sen}(a \pm b) = \operatorname{sen} a \cos b \pm \cos a \operatorname{sen} b \quad (6)$$

$$\cos(a \pm b) = \cos a \cos b \pm \operatorname{sen} a \operatorname{sen} b \quad (7)$$

3.2 Cinemática inversa de cadera y rodilla

La cinemática inversa de un manipulador es usada para calcular los ángulos formados en las articulaciones de este con respecto a un punto de referencia dado del efector final. A diferencia del modelo cinemático directo, determinar el modelo cinemático inverso puede resultar complicado, ya que pueden existir múltiples soluciones o en el peor de los casos ninguna solución, esto depende de la forma en el que se aborde el problema. Para este caso, se calcularon las ecuaciones de $\boldsymbol{\theta_1}$ y $\boldsymbol{\theta_2}$ por el método gráfico, el cual hace uso de herramientas de geometría y trigonometría, además de la configuración mostrada en el sistema de la figura 7 [15].

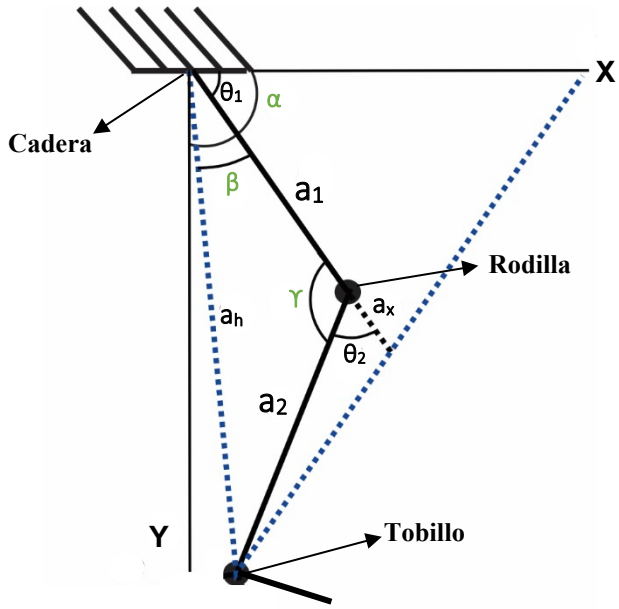


Figura 7. Diagrama de cuerpo libre de una pierna para la cinemática inversa.

La primera coordenada articular que nos interesa obtener es $\boldsymbol{\theta_1}$, la cual es la diferencia entre el ángulo **α** y el ángulo **β**, como se aprecia en la ecuación 8:

$$\theta_1 = \alpha - \beta \quad (8)$$

Para obtener **α** se utiliza la función trigonométrica de la tangente, la cual equivale al cateto opuesto (**C.O.**), en este caso es la distancia de la componente **y**, sobre el cateto adyacente (**C.A**.), distancia de la componente **x**, como se aprecia en la ecuación 9:

$$\tan \alpha = \frac{C.O.}{C.A.} = \frac{y}{x} \therefore \alpha = \tan^{-1}\frac{y}{x} \tag{9}$$

En el caso de **β**, se determina partiendo nuevamente de la función trigonométrica de la tangente, teniendo en cuenta que en este caso el **C.A.** es una sumatoria de distancias, como se observa en la ecuación 10.

$$\tan \beta = \frac{C.O.}{a_1 + a_x} \tag{10}$$

La tangente de un ángulo tiene relación con la función del seno y el coseno, por lo que se realiza un despeje para obtener el **C.O.** de la función trigonométrica del seno (ecuación 11) y el **C.A.** de la función trigonométrica del coseno (ecuación 12), posteriormente se sustituyen en la ecuación 10 para obtener la ecuación 13.

$$\operatorname{sen} \theta_2 = \frac{C.O.}{h} = \frac{C.O.}{a_2} \therefore C.O. = a_2 \operatorname{sen} \theta_2 \tag{11}$$

$$\cos \theta_2 = \frac{C.A.}{h} = \frac{C.A.}{a_2} \therefore C.A. = a_2 \cos \theta_2 \tag{12}$$

$$\tan \beta = \frac{a_2 \operatorname{sen} \theta_2}{a_1 + a_2 \cos \theta_2} \therefore \beta = \tan^{-1}\frac{a_2 \operatorname{sen} \theta_2}{a_1 + a_2 \cos \theta_2} \tag{13}$$

Finalmente, las ecuaciones 9 y 13 se sustituyeron en la ecuación 8 para así obtener la ecuación 14, la cual representa a $\boldsymbol{\theta_1}$.

$$\theta_1 = \tan^{-1}\frac{y}{x} - \tan^{-1}\frac{a_2 \operatorname{sen} \theta_2}{a_1 + a_2 \cos \theta_2} \tag{14}$$

Dónde:

C.O.: Cateto Opuesto.
C.A.: Cateto Adyacente.
h: hipotenusa.

Para obtener la segunda coordenada articular $\boldsymbol{\theta_2}$, se parte de la ley de coseno y obtener una ecuación que relacione al ángulo $\boldsymbol{\gamma}$, como se aprecia en la ecuación 15.

$$a_h^2 = a_1^2 + a_2^2 - 2a_1 a_2 \cos \gamma \tag{15}$$

Se despeja el $\boldsymbol{\cos \gamma}$ de la ecuación 15 y partiendo de la igualdad de $\boldsymbol{\gamma + \theta_2 = 180}$ se despeja $\boldsymbol{\gamma}$ ya que nos interesa encontrar el valor del $\boldsymbol{\cos \theta_2}$, como se aprecia en la ecuación 16 y así obtener la ecuación 18.

$$\cos \gamma = \cos(180 - \theta_2) = -\cos \theta_2 \tag{16}$$

$$-\cos \theta_2 = \frac{a_1^2 + a_2^2 - a_h^2}{2a_1 a_2} \tag{17}$$

$$\cos \theta_2 = \frac{a_h^2 - a_1^2 - a_2^2}{2a_1 a_2} \tag{18}$$

Posteriormente utilizando una identidad del teorema de Pitágoras, mostrado en la ecuación 19, se despeja $\mathbf{sen}^2\,\boldsymbol{\theta_2}$ y se obtiene la ecuación 20, por último, ambas ecuaciones 18 y 20 se sustituyen en la ecuación 21 para obtener la ecuación 22 que representa a $\boldsymbol{\theta_2}$.

$$\operatorname{sen}^2\theta_2 + \cos^2\theta_2 = 1 \tag{19}$$

$$\operatorname{sen}^2\theta_2 = \pm\sqrt{1-\cos^2\theta_2} \tag{20}$$

$$\tan\theta_2 = \frac{\sin\theta_2}{\cos\theta_2} \tag{21}$$

$$\theta_2 = \tan^{-1}\frac{\pm 2a_1a_2\sqrt{1-\cos^2\theta_2}}{a_h^2-a_1^2-a_2^2} \tag{22}$$

Las ecuaciones 14 y 22 son expresiones que permiten calcular los valores angulares de $\boldsymbol{\theta_1}$ y $\boldsymbol{\theta_2}$ para que el extremo final, en este caso el pie, se posicione en las coordenadas deseadas y con la orientación requerida, como se mencionó anteriormente es un análisis con mayor complejidad que la cinemática directa ya que puede haber varias soluciones e inclusive no existir solución analítica, esto dependiendo de cómo se aborde el problema, en este caso la forma en que se tomó el diagrama de cuerpo libre del sistema nos permitió resolver la cinemática inversa.

4. Resultados

Una vez que se ha calculado la cinemática del sistema, se procedió a realizar la simulación del comportamiento de la marcha humana, utilizando la submatriz de traslación obtenida de la ecuación 4 y las ecuaciones que representan los ángulos $\boldsymbol{\theta_1}$ (ecuación 14) y $\boldsymbol{\theta_2}$ (ecuación 22), en la ecuación 23 y 24 se muestra la submatriz de traslación y las ecuaciones de forma matricial respectivamente.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) \\ a_1\operatorname{sen}\theta_1 + a_2\operatorname{sen}(\theta_1+\theta_2) \end{bmatrix} \tag{23}$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \tan^{-1}\frac{y}{x} - \tan^{-1}\frac{a_2\operatorname{sen}\theta_2}{a_1+a_2\cos\theta_2} \\ \tan^{-1}\frac{\pm 2a_1a_2\sqrt{1-\cos^2\theta_2}}{a_h^2-a_1^2-a_2^2} \end{bmatrix} \tag{24}$$

Sin embargo, la cinemática del sistema se obtuvo refiriendo el ángulo $\boldsymbol{\theta_1}$ a partir del semieje positivo **x**. Por tal motivo, para poder modelar el comportamiento de la marcha humana, es necesario realizarlo desde una posición de equilibrio en el semieje negativo **y**, esta es la posición en la que se encuentran realmente las piernas, por lo que debe aplicarse un cambio de variable, sustituyendo $\boldsymbol{\theta_1}$ por $\boldsymbol{\theta_1} - \frac{\boldsymbol{\pi}}{\mathbf{2}}$ [19].

Tomando en cuenta este cambio de variable, junto con las identidades trigonométricas $\mathbf{sen}(\boldsymbol{\theta_1} - \frac{\boldsymbol{\pi}}{\mathbf{2}}) = -\mathbf{cos}\,\boldsymbol{\theta}$ y $\mathbf{cos}(\boldsymbol{\theta_1} - \frac{\boldsymbol{\pi}}{\mathbf{2}}) = \mathbf{sen}\,\boldsymbol{\theta}$, se obtiene las ecuaciones 25 y 26.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1\operatorname{sen}\theta_1 + a_2\operatorname{sen}(\theta_1+\theta_2) \\ -a_1\cos\theta_1 - a_2\cos(\theta_1+\theta_2) \end{bmatrix} \tag{25}$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \tan^{-1}\frac{y}{x} - \tan^{-1}\frac{a_2\operatorname{sen}\theta_2}{a_1+a_2\cos\theta_2} + \frac{\pi}{2} \\ \tan^{-1}\frac{\pm 2a_1a_2\sqrt{1-\cos^2\theta_2}}{a_h^2-a_1^2-a_2^2} \end{bmatrix} \tag{26}$$

Se implementó una función que permite simular el comportamiento de la marcha a partir de la cinemática utilizando las ecuaciones 25 y 26. Esta función permite evaluar los parámetros obtenidos en la sección 2 y obtener de manera gráfica el comportamiento de las articulaciones de la cadera y la rodilla durante el ciclo de la marcha.

En la figura 8 se presenta de manera gráfica el comportamiento de un ciclo de la marcha humana en la articulación de la cadera, se observa que cumple aproximadamente con el porcentaje de 60% de fase de apoyo y 40% de fase de balanceo. Además, se puede apreciar que los valores de los ángulos de flexión, tienden a visualizarse como pendientes positivas, por lo que la flexión en esta articulación se encuentra aproximadamente en el rango de -18° a 30°, por otro lado, la extensión tiende a visualizarse como pendientes negativas, en este caso se observa que la extensión en esta articulación se encuentra en el rango de 30° a -18°, por lo que los resultados obtenidos son similares a los presentados en la sección 2 del análisis de la marcha. Por otro lado, el tiempo obtenido por un ciclo de marcha para esta articulación es de 0.9 segundos, el cual es un valor muy similar al presentado en otros artículos relacionados a la marcha humana [4, 20, 21].

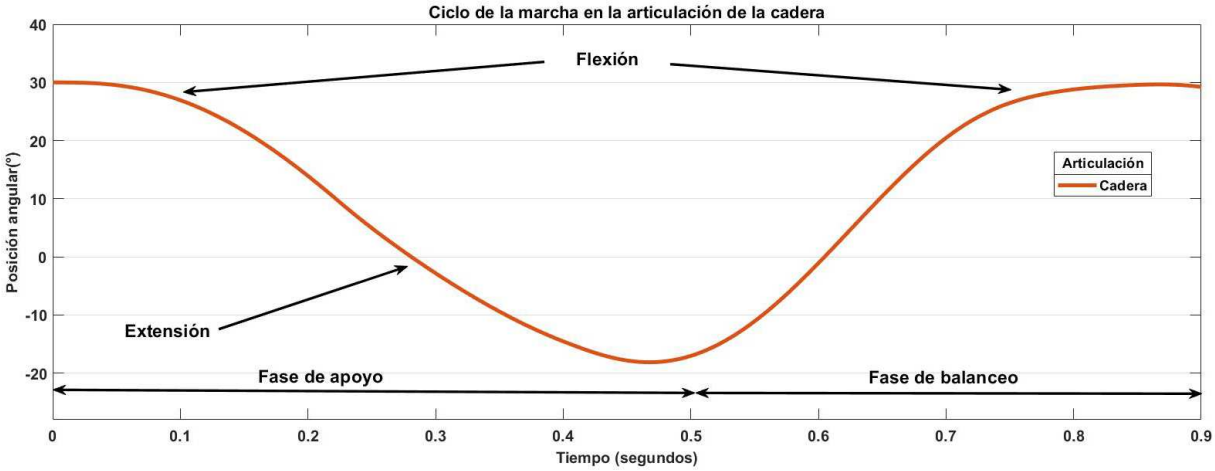


Figura 8. Ciclo de la marcha en articulación de la cadera.

En el caso de la articulación de la rodilla, su comportamiento en un ciclo de la marcha humana se muestra en la figura 9, de la misma manera se cumple aproximadamente con el porcentaje de 60% de fase de apoyo y 40 % de fase de balanceo, los ángulos de flexión tienden a visualizarse como pendientes positivas, por lo que la flexión en esta articulación se encuentran aproximadamente en los rangos de 0° a 18° y de 5° a 55°, en el caso de la extensión tiende a visualizarse como pendientes negativas, en este caso se observa que la extensión en esta articulación se encuentra en los rangos de 18° a 5° y de 55° a 0°, por lo que de madera similar los resultados obtenidos se aproximan a los ángulos presentados en la sección 2. Para esta articulación el tiempo obtenido para un ciclo de marcha es de 1.1 segundos, similar a lo reportado previamente [4, 20, 21].

Por último, se realizó una simulación de la marcha humana por un periodo de 12 segundos, obteniendo los resultados mostrados en la figura 10 y en la figura 11, en ambas graficas se puede apreciar que se obtuvieron 12 ciclos de la marcha aproximadamente, en otras palabras, se realizaron 24 pasos en 12 segundos, teniendo en cuenta que una persona realiza una caminata promedio de 100 pasos por minuto, independientemente de si es hombre o mujer [22].

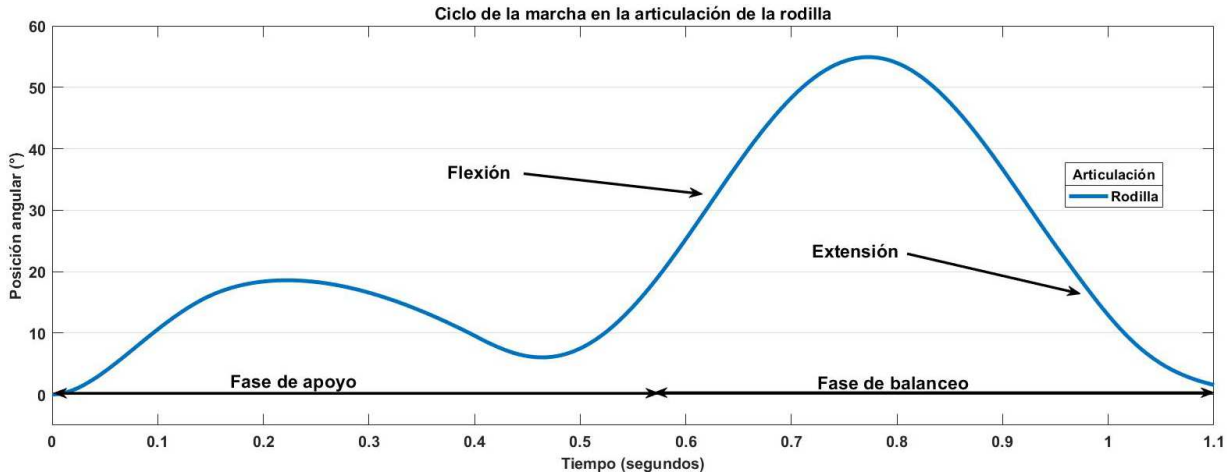


Figura 9. Ciclo de la marcha en articulación de la rodilla.

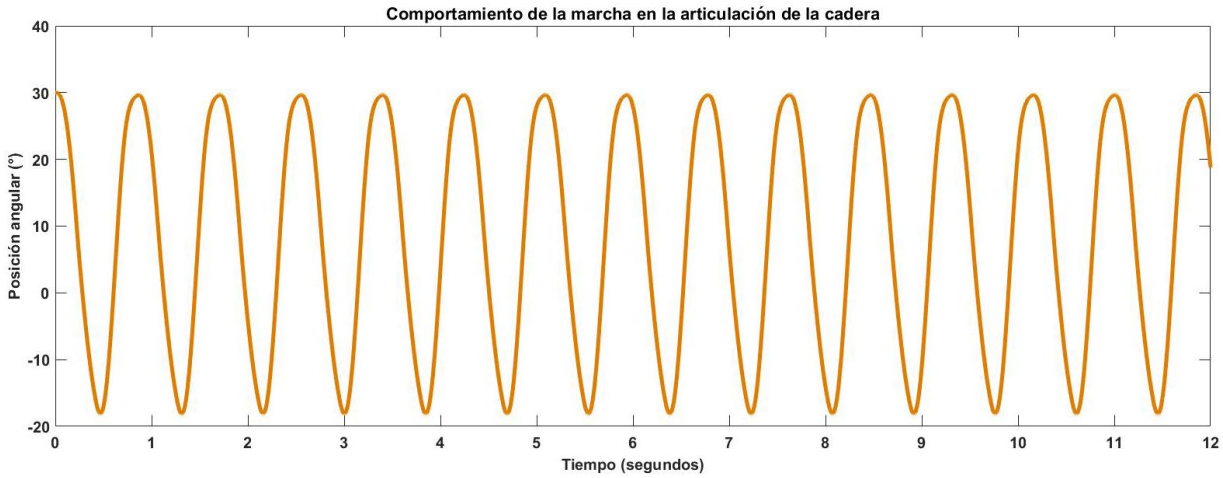


Figura 10. Comportamiento de la marcha en articulación de la rodilla.

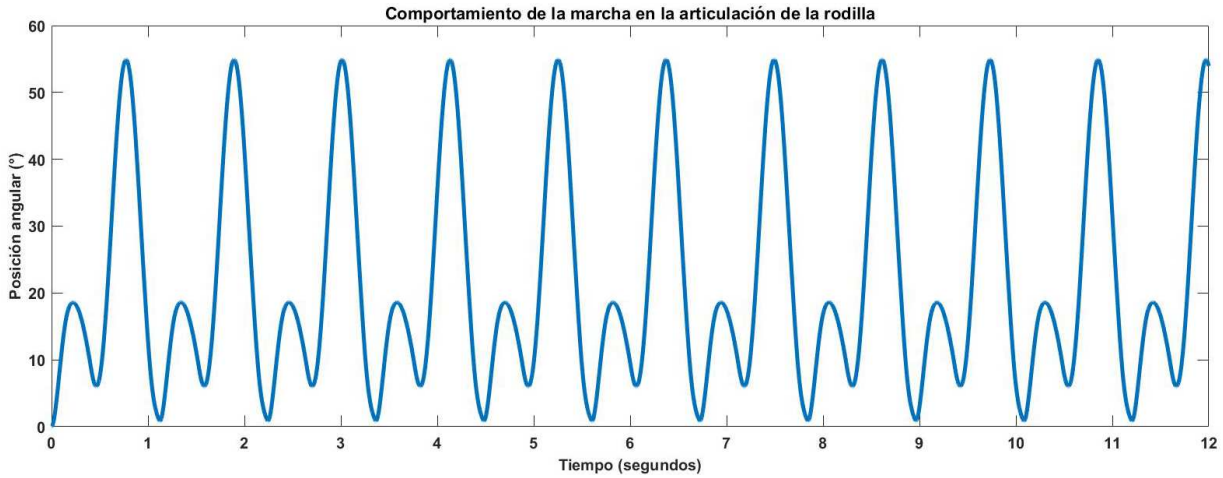


Figura 11. Comportamiento de la marcha en articulación de la rodilla.

5. Conclusiones

Una de las capacidades del ser humano es la habilidad de caminar de forma erguida, simultánea y aparentemente sin ningún esfuerzo, llegando a ser como un evento continúo de transferencia de peso de una pierna a la otra, en otras palabras, como desequilibrios sucesivos, con el objetivo de desplazarse hacia adelante. La marcha humana es un proceso de locomoción único, por lo que un individuo, posee un comportamiento diferente al de cualquier otra persona.

El análisis matemático realizado permitió obtener las ecuaciones que representan el movimiento de las extremidades inferiores en las articulaciones de la cadera y la rodilla, por lo que al evaluar la información obtenida de la literatura sobre el comportamiento de la marcha humana que toman las articulaciones de interés en dichas ecuaciones, se logra obtener un comportamiento similar al de la marcha humana, lo que garantiza que el análisis realizado puede ser implementado para detectar anomalías en la marcha humana o desarrollar un sistema de rehabilitación que apoye a pacientes que presenten alguna anomalía en el ciclo de la marcha, siendo esta la principal aportación de este artículo.

De acuerdo con los resultados obtenidos en la simulación, se observa que existen valores similares en los ángulos de las articulaciones de la cadera y la rodilla con respecto a los valores teóricos presentados del ciclo de la marcha. De esta manera, queda demostrado que se pueden obtener resultados muy próximos a los reales a partir del análisis de la cinemática directa e inversa. Además, de demostrarse que cada persona tiene un proceso de locomoción diferente pero muy similares a los mostrados en la teoría.

Por otro lado, este análisis es un paso necesario para completar la labor de modelar una etapa de control, con el fin de lograr el seguimiento de la posición angular en cada articulación y lograr replicar el ciclo de la marcha sobre el plano sagital en un sistema de rehabilitación de dos grados de libertad en cadera y rodilla, y así poder apoyar en la rehabilitación de pacientes que cuenten con algún problema de discapacidad en los miembros inferiores.

Referencias

[1] Moore K., Dalley A. y Agur A., "*Anatomía con orientación clínica"*, Ed. Ovid Technologies, Barcelona, 7° edición, 2013.
[2] Nordin M. y Frankel V., *"Biomecánica básica del sistema musculoesquelético"*, Ed. McGraw-Hill, España, 3° edición, 2004.
[3] Martínez L., *"Sistema para terapia activa de miembros inferiores en humanos"*, Tesis de Maestría, 2019.
[4] Luengas L., Marín C. y González J., *"Modelo de la marcha bípeda humana usando Modelica"*, Visión electrónica, vol. 7, no 2, p. 110-124, 2013.
[5] Perry J. and Davids J., *"Gait analysis: normal and pathological function"*, Journal of Pediatric Orthopaedics, vol. 12, no. 6, p. 815, 1992.
[6] Osorio J., & Valencia M., "*Bases para el entendimiento del proceso de la marcha humana*", Archivos de Medicina, vol. 13, no. 1, p. 88-96, 2013.
[7] Jahn K, Dieterich M., "*Recent advances in the diagnosis and treatment of balance disorders*", J Neurol, vol. 258, no. 12, p. 2305-2308, 2011.
[8] Enkelaar L, Smulders E, Geurts A., Weerdesteyn V., "*A review of balance and gait capacities in relation to falls in persons with intellectual disability*", Research in developmental disabilities, vol.13, no. 1, p. 291-306, 2012.
[9] Cailliet R., "*Anatomía funcional Biomecánica*", Ed. Philadelphia, 3° edición, 1992.
[10] Benini R, Amor I., Shevell M., "*Clinical clues to differentiating inherited and noninherited etiologies of childhood ataxias*", Journal of Pediatrics, vol. 160, no. 1, p. 152-157, 2012.
[11] Viswanathan A., Sudarsky L., "*Balance and gait problems in the elderly*", Hand Clin Neurol, vol. 103, p. 623-634, 2012.

[12] Chambers H. and Sutherland D., "*A practical guide to gait análisis*", JAAOS-Journal of the American Academy of Orthopaedic Surgeons, vol. 10, no. 3, p. 222-231, 2002.
[13] Pericé A. "*Significado de la postura y de la marcha humana:(teología, antropología, patología)*" Ed. Complutense, España, 1° edición,1996.
[14] Hernández F. "*Diseño y construcción de prototipo neumático de prótesis de pierna humana*", Tesis de Licenciatura, 2008.
[15] F. Reyes, "*Robótica-Control de robots manipuladores*", Ed. Alfaomega, México, 1° edición, 2011.
[16] Kelly R. and Santibáñez V., "*Control de movimiento de robots manipuladores*", Ed. Pearson educación, Madrid, 1° edición, 2003.
[17] Lee H. and Chang S., "*Development of a cad/cae/cam system for a robot manipulator*", Journal of Materials Processing Technology, vol.140, no. 1-3, p. 100-104, 2003.
[18] Giraldo L., Delgado E., and Castellanos G., "*Cinemática inversa de un brazo robot utilizando algoritmos genéticos*", Revista Avances en Sistemas e Informática, vol. 3, no. 1, p. 29-34, 2006.
[19] F. Reyes, "*MATLAB Aplicado a Robótica y Mecatrónica*", Ed. Alfaomega, México, 1° edición, 2012.
[20] Du Z., Zhang Z., Long Y. and Dong W., "*Gait tracking for lower extremity exoskeleton based on sliding mode control with CMAC compensation*", IEEE International Conference on Mechatronics and Automation, pp. 1191-1196, 2016.
[21] Bravo L. y Garzón M., "*Modelamiento de la marcha humana por medio de gráficos de unión*", Tecnura, vol. 8, no. 16, p. 26-42, 2005.
[22] Beets M, Morgan C, Banda J, Bornstein D, Byun W, Mitchell J, Munselle L, Rooney L, Beighle A, Erwin H. "*Convergent validity of pedometer and accelerometer estimates of moderate-to- physical activity of youth*", J Phys Act Health, vol. 8, no. s2, p. S295-305, 2011.

Diseño Colaborativo en Mecatrónica, Capítulo 13, pp. 167 - 184
ISBN: 978-607-9394-23-3, 2021

Una comparación cuantitativa entre el diseño clásico y automático (AutoML) de modelos de aprendizaje profundo en tareas de clasificación

Ángel de Jesús Osorio Guevara[1]✉, Sebastián Salazar Colores[2]✉

[1] Universidad de las Américas Puebla (UDLAP)
[2] Centro de Investigaciones en Óptica (CIO)

✉angel.osorioga@udlap.mx, ✉sebastian.salazar@cio.mx

Resumen

En la última década la investigación y desarrollo de modelos de inteligencia artificial y en concreto de aprendizaje profundo han tenido un gran auge. Lo anterior está fuertemente motivado por los excelentes resultados obtenidos en tareas de clasificación, detección de patrones, regresión y generación de datos de otras áreas científicas, pero sobre todo por sus aplicaciones comerciales. En un inicio el desarrollo de modelos de inteligencia artificial (IA) requería fuertes conocimientos y habilidades de parte del desarrollador restringiendo su utilización a expertos. Esfuerzos realizados por empresas tecnológicas orientados a democratizar y acercar el aprendizaje profundo a más personas han resultado en el desarrollo de librerías como Tensorflow, Pytorch o Keras. Sin embargo, aún tienen una curva de aprendizaje de conocimientos y habilidades que la alejan de expertos de otras áreas. Actualmente, existe un área de investigación incipiente que permite desarrollar modelos de IA de forma automática y con requerimientos de conocimientos mínimos: AutoML (aprendizaje máquina automático o auto machine learning por su nombre en inglés). En este artículo hacemos un estudio comparativo de un framework de AutoML de código libre llamado AutoKeras contra una librería muy popular llamada Keras. Para tener un buen marco de comparación la tarea de clasificación de imágenes fue aplicada sobre tres bases de datos de distintas características: Fashion MNIST, Cifar10 e ISBI 2017. Los resultados obtenidos muestran que los resultados de AutoML son de acuerdo con las métricas de exactitud, precisión y sensibilidad un 7%, 23% y 24 % superiores que los obtenidos mediante un diseño manual usando Keras por un desarrollador principiante con apenas una tercera parte de líneas de código.

Palabras clave: Aprendizaje máquina (*ML*), Aprendizaje máquina automático (*Auto ML*), Clasificación, Red neuronal convolucional (*CNN*).

1. Introducción

La inteligencia artificial (IA) surge alrededor de los años 50 cuando algunos científicos en el área se hicieron la pregunta de si a una máquina se le podía hacer pensar [1], pregunta cuyas ramificaciones se siguen estudiando hoy en día. Como definición se puede decir que la IA involucra un esfuerzo por automatizar tareas intelectuales que normalmente serían realizadas por humanos. Los primeros enfoques de este tipo tenían como base la creación de reglas codificadas fabricadas por expertos, ya que se creía que una computadora podía obtener una inteligencia similar a las de los humanos si se contaba con un gran conjunto de reglas previamente definidas [1], sin embargo, las limitaciones de este tipo de enfoque son bastante amplias, ya que adaptar y escalar estos modelos a tareas más complejas requiere cantidades ingentes e imprácticas de recursos humanos.

Más adelante se comenzaron a tomar en cuenta otros paradigmas, como lo es el aprendizaje automático o aprendizaje máquina (ML, pos sus siglas en inglés). Como se mencionaba anteriormente, para desarrollar un modelo de IA era necesario programar explícitamente una serie de reglas, no obstante, en el caso del aprendizaje máquina, si por ejemplo se necesita realizar una tarea como es la clasificación de imágenes, se requiere "alimentar" al modelo con ejemplos y respuestas, en este caso con imágenes y sus etiquetas de clase [2]. Por tanto, un modelo de aprendizaje máquina tiene como fin de modelar a través de la experiencia una función que le sirva para realizar una tarea con el mínimo error posible [3]. Por lo que el aprendizaje máquina es más un método de entrenamiento que uno programado explícitamente.

Para procesar efectivamente bases de datos que contienen, por ejemplo, imágenes, es necesario buscar representaciones significativas para el modelo con el fin de que pueda ser entrenado eficientemente, de esto se encarga el aprendizaje profundo, una rama del aprendizaje máquina que basa su estructura en capas jerárquicas de distintos niveles de abstracción [4]. En el caso de imágenes, el aprendizaje profundo funciona como un cedazo o filtro que va removiendo características sin importancia para que, con el paso de las capas, solo queden las características más imprescindibles[1]. Cada capa cuenta con un arreglo de neuronas, las cuales son unidades que tienen un valor numérico dependiendo de los datos de entrada, estas a su vez están, normalmente, totalmente conectadas con neuronas pertenecientes a capas adyacentes [2]. Al arreglo de capas con sus respectivas neuronas cuya cardinalidad es de al menos 3 se le llama red neuronal profunda.

En los últimos años el aprendizaje profundo ha mostrado un desempeño superior a otros paradigmas de IA tales como bosques aleatorios (random forest), máquinas de soporte vectorial (SVM's), clasificador Naive Bayes en una infinidad de aplicaciones tales como clasificación de imágenes, detección de patrones en señales médicas, conducción autónoma, desarrollo de nuevas moléculas para medicamentos, etc. Es por lo anterior el creciente interés por este paradigma y la búsqueda de potenciales aplicaciones en prácticamente todas las áreas de tecnología y conocimiento humano.

Al momento de diseñar un sistema basado en una red neuronal profunda existen normalmente dos opciones: en la primera, los desarrolladores codifican la estructura neuronal desde cero, y en la segunda se apoyan de bibliotecas diseñadas por grandes corporaciones tecnológicas y universidades, tales como Pytorch, Tensorflow, Keras, y Caffe. Dada la complejidad y el costo extremadamente alto al desarrollar la primera opción, esta no se utiliza en la industria salvo en casos muy específicos donde las redes neuronales no son muy grandes y además es vital tener implementaciones con un rendimiento excepcional. Por tanto, en la actualidad el utilizar bibliotecas es el estándar de la industria, sin embargo, el conseguir expertos en inteligencia artificial dada su alta demanda en el mercado para que elijan y desarrollen las mejores estructuras y modelos neuronales hace difícil que se produzcan sistemas eficaces.

En la actualidad un nuevo sistema o paradigma de diseño automatizado está emergiendo y teniendo un gran impacto en la industria, este paradigma es conocido como aprendizaje máquina automático o *AutoML*. En este tipo de sistema, un método basado en estadística o inteligencia artificial busca automáticamente las arquitecturas e hiperparámetros necesarios para diseñar de la mejor manera el modelo neuronal frente al problema al que uno se enfrente, usando operaciones de morfismo de red, las cuales consisten en agregar más capas, filtros, etc. [5], por lo que no se requieren conocimientos especializados para operar con este método. Entre las principales propuestas de métodos y soluciones basadas en *AutoML* se pueden mencionar *auto-sklearn*, *Ludwig*, Amazon *Lex*, *AutoFolio*, *H2O.ai* y *AutoKeras*.

Con la finalidad de establecer claramente las ventajas y desventajas de optar por el uso de utilizar bibliotecas y el sistema de *AutoML* en tareas de clasificación, en este trabajo se realiza una comparación cuantitativa entre ellos, teniendo como criterios el número de líneas requeridas, precisión alcanzada y hardware requerido sobre tres bases de datos. En particular, se comparan *Keras* contra *AutoKeras*, dado que son librerías de código abierto muy populares en investigaciones actuales.

El resto del presente trabajo está distribuido como se muestra a continuación. En la Sección 2 se presentan los conceptos teóricos que sustentan nuestra investigación. En la sección 3 y 4 se exponen la metodología y los resultados obtenidos, respectivamente. Por último, en la sección 5 se presentan las conclusiones y las perspectivas de futuras investigaciones.

2. Fundamentos teóricos

2.1 Conceptos detrás de las redes neuronales convolucionales

Antes de discutir los resultados, es conveniente conocer o recordar algunos de los conceptos básicos detrás del funcionamiento de modelos de machine learning, conceptos que no resultan tan complicados y que aplicados terminan haciendo cosas increíbles. Se comenzará hablando de redes neuronales y un poco del trasfondo matemático que rige su comportamiento, para finalizar con definiciones que tienen que ver con el análisis estadístico del desempeño de los modelos.

Como se mencionaba anteriormente, para el aprendizaje automático se necesitan en general tres elementos:

- **Datos de entrada**: Los cuales para los objetivos del presente trabajo serán imágenes.
- **Ejemplos de los resultados esperados**: Si se tratan con imágenes entonces, los ejemplos serán sus respectivas etiquetas o *labels* .
- **Una manera de asegurarnos que el modelo está aprendiendo:** Con esto se controla el comportamiento del modelo hasta que se encuentre el mínimo error entre los valores predichos y reales.

Tal como se muestra en la Figura 1 es evidente la necesaria existencia de una capa neuronal de entrada y otra de salida. Conforme las imágenes van pasando por capas neuronales intermedias, estas se van volviendo más abstractas, obteniendo una representación más significativa para la computadora [1][2]. Los parámetros más importantes para las neuronas a aprender son: la activación de las neuronas, las funciones peso y el sesgo [6]. Al trabajar con imágenes la activación de las neuronas dependerá de la representación numérica del color de cada píxel de la imagen de entrada. En conjunto, con los tres factores se puede realizar una suma ponderada de la forma

$$a = b + \sum_{j=0}^{n} w_j x_j \tag{1}$$

dónde w_j corresponde a la j –ésima componente del vector de funciones peso, x_j a la j –ésima componente del vector de valores de entrada y b al sesgo de la neurona en la que se requiere el valor de salida. Lo anterior muestra que tanto los valores de entrada como las funciones peso se representan en vectores de n dimensiones, y su producto interno más la tendencia nos dará un valor de salida en la siguiente neurona [2].

Sin embargo, para tareas como el reconocimiento de imágenes es más conveniente utilizar redes neuronales convolucionales, las cuales, en lugar de trabajar con vectores trabajan con tensores, los cuales no son más que un arreglo de matrices. Una convolución en términos sencillos es una operación que nos permite aplicar un filtro varias veces, para esto se requiere de un kernel, el cual es una matriz de $m \times n$ dimensiones cuyas componentes son las funciones peso, dicha matriz realiza un barrido a toda la imagen [3] efectuando productos internos con pequeñas secciones de la misma de tamaño $m \times n$, y de igual manera se calcula una suma ponderada que nos dará valores de salida para cada componente o neurona de una nueva capa [2][3], la cual está dada por

$$a = b + \sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} x_{j+k,i+l} \tag{2}$$

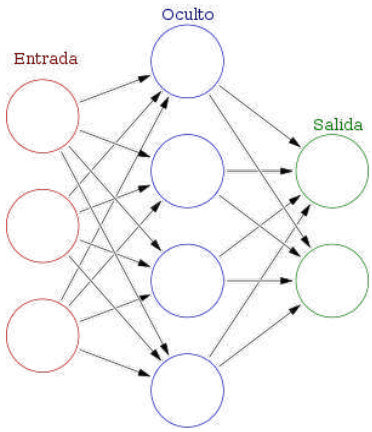


Figura 1. Ejemplo de red neuronal. Al extremo izquierdo y derecho se encuentran las capas de entrada y salida respectivamente, entre ellas están ubicadas las capas ocultas cuyo trabajo es filtrar la información para obtener una representación significativa [7].

La nueva capa o tensor que se crea, como en el ejemplo de la Figura 2, contiene información más condensada en la que se resaltan ciertas características de la imagen. Además, para condensar en mayor medida la información se aplica una operación llamada *Pooling*, la cual generalmente submuestrea la capa y se obtiene generalmente mediante el promedio o valor máximo de cada capa [1][3]. Ambas operaciones se puede realizar un par de veces dependiendo de las necesidades, al final quedará un arreglo tensorial de neuronas con el cual el modelo puede ser capaz de clasificar imágenes, para esto, el tensor se *aplana* para convertirlo en una red neuronal como la de la Figura 1 y obtener un valor de salida.

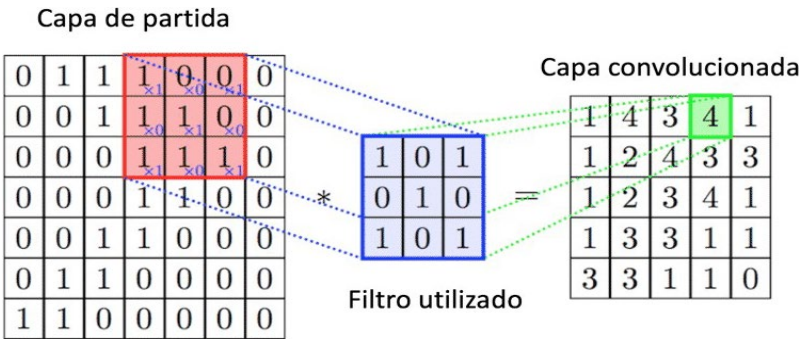


Figura 2. Ejemplo de red neuronal convolucional. A la izquierda se muestra lo que podría ser la imagen de entrada por la que un kernel de 3×3 pasa para obtener la cuarta neurona en la primera fila de la primera capa oculta o convolucionada [8].

Por último, para rastrear el rendimiento de nuestro modelo es indispensable introducir una función costo, la cual indicará que tan lejos están los valores reales de los predichos. Esta función depende de w_j y b, y la tarea a realizar con la función de costo es la minimización para encontrar el mejor conjunto posible de funciones peso y tendencias, para esto se utiliza el gradiente descendente [2][6], el cual, opuesto al gradiente, es un vector cuya dirección apunta hacia donde la función decrece más rápido, para esto se dan pequeños *pasos* hasta llegar al punto crítico adecuado [2][4], por lo que la actualización de w_j y b se verían como

$$w_j' = w_j - \eta \frac{\partial c}{\partial w_j} \tag{3}$$

$$b_i' = b_i - \eta \frac{\partial c}{\partial b_j} \tag{4}$$

dónde C denota la función costo y η la tasa de aprendizaje, el cual es el tamaño de los *pasos en la dirección del gradiente descendente* [1], de igual manera, las derivadas parciales son parte del gradiente mientras que el apóstrofe denota las funciones peso y tendencias actualizadas. Todo el proceso, incluyendo las convoluciones, el *pooling* y la minimización de la función costo se repite varias veces hasta que el modelo sea capaz de predecir con cierta precisión.

Durante todo el proceso existen otros pasos intermedios que se tienen que considerar, uno de estos pasos importantes es la inclusión de las llamadas funciones de activación entre capas, pues es necesario mantener nuestros valores de salida dentro de ciertas cotas, y dos de las más utilizadas son las funciones *relu* y *softmax*. La primera obtiene un valor de 0 para valores negativos y mantiene el valor de los números positivos mediante una función lineal [9], mientras que la segunda es utilizada ampliamente cuando se requiere de clasificar entre más de dos categorías, y se encarga de normalizar un vector n-dimensional con componentes reales arbitrarias, esto es, convierte las componentes del vector en probabilidades [1]. Por otro lado, *dropout* es una técnica que permite optimizar el aprendizaje de nuestro modelo [4], en términos generales, este se refiere al proceso en el que se ignoran algunas neuronas en una capa de manera probabilística, limitando así los parámetros a aprender (funciones peso y tendencias) para obtener representaciones todavía más reducidas; hacer esto en repetidas ocasiones se puede aproximar al proceso de entrenar la red con diferentes arquitecturas para obtener un promedio de los valores de salida [10].

Una vez que el proceso de entrenamiento ha terminado, es necesario establecer una estrategia que nos permita conocer el desempeño del modelo de una manera más meticulosa, para esto existen diversas técnicas y una de las más comunes es el uso de matrices de confusión en las que se puede observar gráficamente los casos en los que el modelo acierta o erra. De esta información se pueden obtener dos métricas muy importantes, la sensibilidad y la precisión. Para empezar, es importante aclarar que existe una diferencia entre la exactitud y la precisión, la primera se puede definir como la proporción entre el número de muestras correctamente clasificadas y el número total de casos, sin embargo, esta presenta desventajas ya que no toma en cuenta la distribución de las clases en una base de datos [11], por otro lado, la precisión tiene que ver con la cercanía que tienen las predicciones entre ellas, independientemente de si son exactas o no [12]. Finalmente, la sensibilidad mide la proporción de clases clasificadas correctamente [11]. Estas tres métricas se presentan en el apartado de Resultados, no obstante, como se verá más adelante, estas cobran más sentido para el caso en el que se tiene que clasificar lesiones de piel.

3. Metodología

En esta sección se muestra detalladamente el proceso realizado al comparar los paradigmas de *ML* y *AutoML*, al igual que los procesos anteriores para realizar dicha tarea.

3.1 Conjuntos de datos de imágenes

Se usaron tres bases de datos para entrenar y probar el modelo, todas con diferentes características. La primera corresponde a la base de datos *Cifar 10*, la cual cuenta con 50000 imágenes para entrenar además de 10000 imágenes que son utilizadas para probar el rendimiento del modelo desarrollado [13]. Esta base de datos tiene etiquetas dentro de 100 tipos de categorías entre las que se encuentran vehículos, animales, insectos, electrodomésticos, etc. La segunda base de datos utilizada fue *MNIST Fashion* la cual incluye 60000 imágenes para entrenar más 10000 para probar; en este caso las imágenes están etiquetadas dentro de 10 categorías las cuales corresponden a prendas de ropa como lo son playeras, vestidos botas, etc. [14].

Ambos conjuntos se obtuvieron del sitio web de Keras. Por último, se trabajó con otro conjunto de imágenes más complejas obtenido del archivo *ISIC*, utilizado para el reto del *ISBI 2017*; en él se encuentran imágenes de lesiones de piel, las cuales están etiquetadas en tres categorías: melanoma, queratosis seborreica y nevo benigno [15]. Esta base de datos cuenta con aproximadamente 2000 imágenes en total.

3.2 Manipulación de las imágenes

Todos los conjuntos de datos fueron acomodados en arreglos. Para la normalización de las imágenes, dichos arreglos se dividieron entre un factor de 255, ya que el píxel con el mayor valor numérico en los tres casos era 255. Para el caso de las imágenes relacionadas con lesiones en la piel, el conjunto se dividió en tres subconjuntos diferentes, uno para entrenar, otro para validar y el último para probar el rendimiento, en una proporción del 60%, 20% y 20% respectivamente, sin embargo, para los otros dos conjuntos se creó un subconjunto de validación, utilizando el 16% de las imágenes de entrenamiento. Las etiquetas fueron codificadas a una representación *one-hot* utilizando el método *to_categorical* perteneciente a la librería de Keras, sin embargo, para la base de datos del *ISBI 2017* no fue necesario hacer esto ya que, por cuestiones de simpleza, las etiquetas ya estaban definidas como 0 para casos de no melanoma y 1 para casos de melanoma.

3.3 Entorno de trabajo

Los experimentos se llevaron a cabo en la plataforma de Google Colab, la cual en su versión gratuita permite el acceso a 12 GB de RAM, un procesador Xeón @ 2.20GHz, al igual que acceso limitado a alguna GPU de alto desempeño. Se nos asignó la tarjeta gráfica de NVIDIA Tesla T4, la cual ofrece hasta 16 GB de VRAM.

3.4 Definición del modelo de red neuronal convolucional

Se definió un modelo secuencial utilizando la librería Keras, la cual permite apilar varias capas de manera ordenada resultado de las operaciones convolucionales y de *pooling* [16] Debido a que el propósito es clasificar imágenes, se utilizaron capas *Conv2D* en conjunto con *MaxPooling*. Para compilar el modelo se utilizó una función de entropía cruzada junto con el optimizador Adam. En el caso de AutoKeras, la herramienta escoge automáticamente las operaciones que más convengan dependiendo de la tarea.

3.5 Eficiencia del modelo

Los modelos creados con *Keras* fueron comparados con aquellos proporcionados por *AutoKeras*. En el caso de los primeros se utilizó la herramienta de análisis *TensorBoard* con el fin de visualizar las métricas de entrenamiento y validación a través de gráficas. Además, en ambos casos se utilizaron matrices de confusión para observar las ocasiones en las que los modelos erraron o acertaron, y en base a esto analizar estadísticas como lo son la sensibilidad y precisión con el fin de obtener un mejor panorama en cuanto al rendimiento de los modelos.

4. Resultados

En este apartado, se presentan los tres casos en los que las redes neuronales fueron probadas, se presentarán primero los resultados obtenidos con *Keras* para ser comparados inmediatamente con los obtenidos utilizando *AutoKeras* utilizando gráficas, matrices y algunas tablas. Solo en el primer caso se mostrará el código utilizado para definir el modelo, esto debido a que se puede inferir de manera trivial la estructura de este para los casos restantes.

4.1 Fashion MNIST

A continuación, se muestran los resultados de las implementaciones de Keras (ML) vs AutoKeras (AutoML) sobre la base de datos Fashion MNIST.

4.1.1 Implementación con Keras (ML)

En la Figura 3 se muestra un arreglo de 10 imágenes correspondientes a cada etiqueta, las cuales fueron obtenidas de la base de datos en cuestión, estas tienen un tamaño de 35 × 35 píxeles y están

en escala de grises. En cuanto al modelo, este se definió como un modelo secuencial con dos capas *Conv2D* y dos capas *MaxPooling* como se observa a continuación en el Código 1 y en la Figura 4.

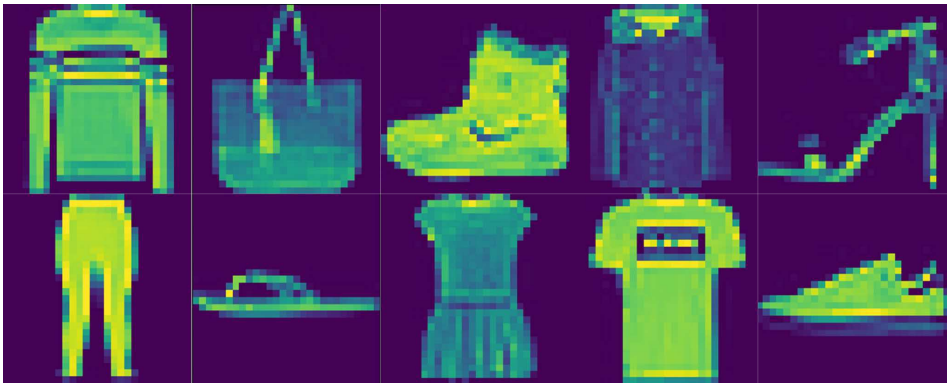


Figura 3. Muestras de las imágenes pertenecientes al conjunto Fashion MNIST [12].

```
model = Sequential([
                    Conv2D(filters = 32,
                           kernel_size = (3, 3),
                           activation = 'relu',
                           input_shape = (28, 28, 1)),
                    MaxPooling2D(pool_size=(2,2)),
                    Conv2D(filters = 64,
                           kernel_size = (3, 3),
                           activation = 'relu'),
                    MaxPooling2D(pool_size=(2,2)),
                    Flatten(),
                    Dense(units = 64, activation = 'relu'),
                    Dropout(0.5),
                    Dense(units = 10, activation = 'softmax')
                   ])
```

Código 1. Código de la definición del modelo implementado en Keras para la clasificación de imágenes del conjunto MNIST Fashion.

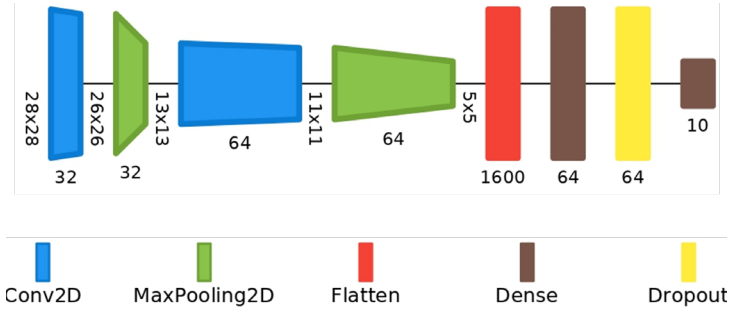


Figura 4. Esquema gráfico del modelo definido con Keras para la clasificación de imágenes del conjunto MNIST Fashion.

Para compilar el modelo, la función de pérdida utilizada fue la función de entropía cruzada en conjunto con un optimizador Adam, el cual, es una extensión del gradiente descendente inspirado en el concepto físico del momento, por lo que considera la pendiente de la posición actual al igual que el valor de los parámetros consecuencia de los valores anteriores [1].

Con esta configuración se logró un valor de pérdida de 0.57 con una exactitud de aproximadamente 91%. Para ver más a detalle el rendimiento de la red, se presentan en la Figura 5 las gráficas de la evolución del valor de pérdida y exactitud para los casos de entrenamiento y validación.

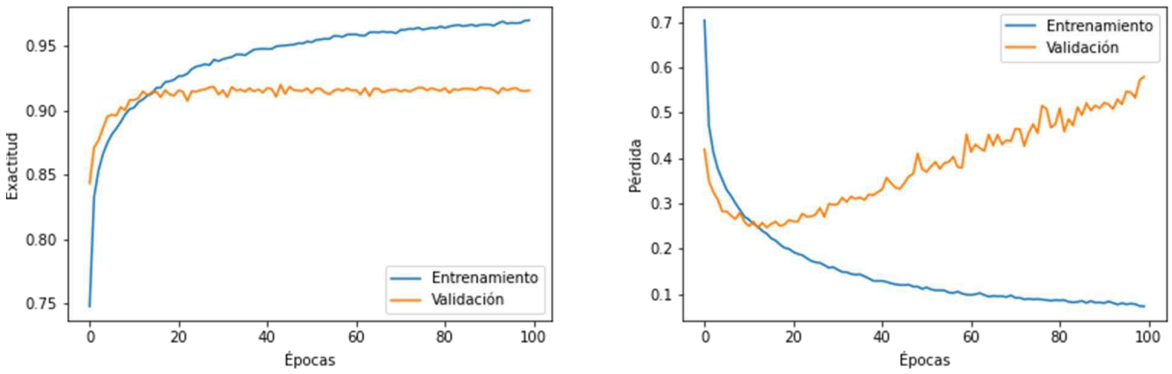


Figura 5. Evolución del valor de exactitud (derecha) y pérdida (izquierda) respecto a las épocas transcurridas utilizando la base de datos *MNIST Fashion*.

En la Figura 6 se muestra una matriz de confusión en la que se observan los aciertos y errores de predicción. En el caso de las gráficas de la Figura 5 se puede decir que entre la época 15 y 20 es cuando el modelo presenta un mejor valor de pérdida y exactitud ya que conforme avanza el proceso de entrenamiento la pérdida aumenta mientras que la precisión se estanca, por lo que parece conveniente establecer como valor recomendado un valor de entre 15 y 20 épocas.

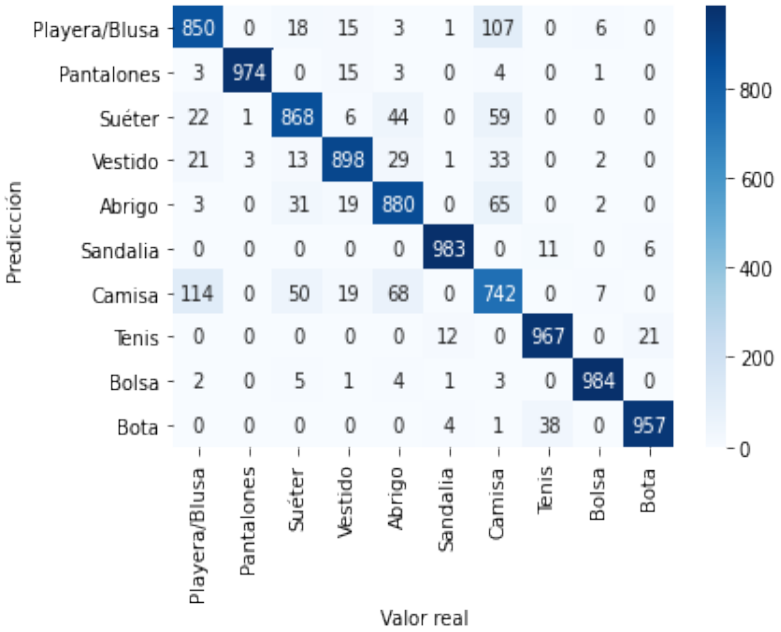


Figura 6. Matriz de confusión para el modelo creado en Keras con la base de datos *MNIST Fashion*.

4.1.2 Implementación con AutoKeras (AutoML)

En este caso lo único que se tienen que proporcionar son los datos para entrenar al igual que el número de intentos para probar con diferentes arquitecturas, para nuestro caso solo se realizó un intento por limitaciones en la memoria de la tarjeta gráfica. A continuación, en la Figura 7, se puede observar el resumen del modelo obtenido por AutoKeras.

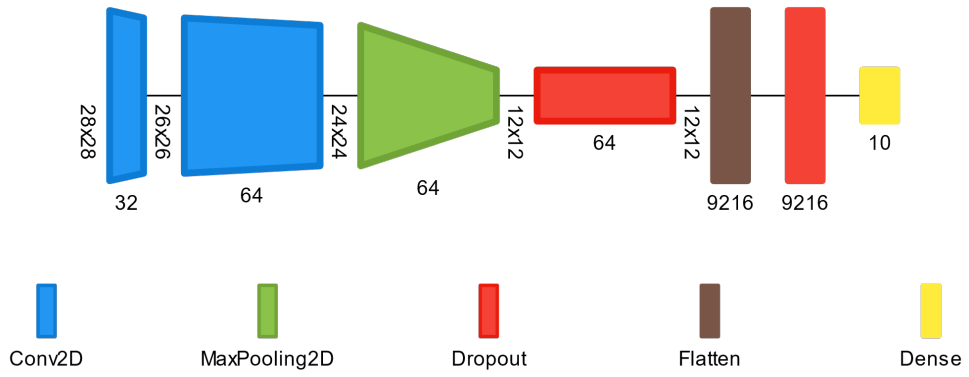


Figura 7. Esquema gráfico del modelo definido por *AutoKeras* para la clasificación de imágenes del conjunto *MNIST Fashion*.

Para esta configuración se obtuvo un valor de pérdida de 0.26 con una exactitud cercana al 92.5%. De igual manera, se presenta la matriz de confusión en la Figura 8 para comparar los aciertos y errores de este modelo con el anterior.

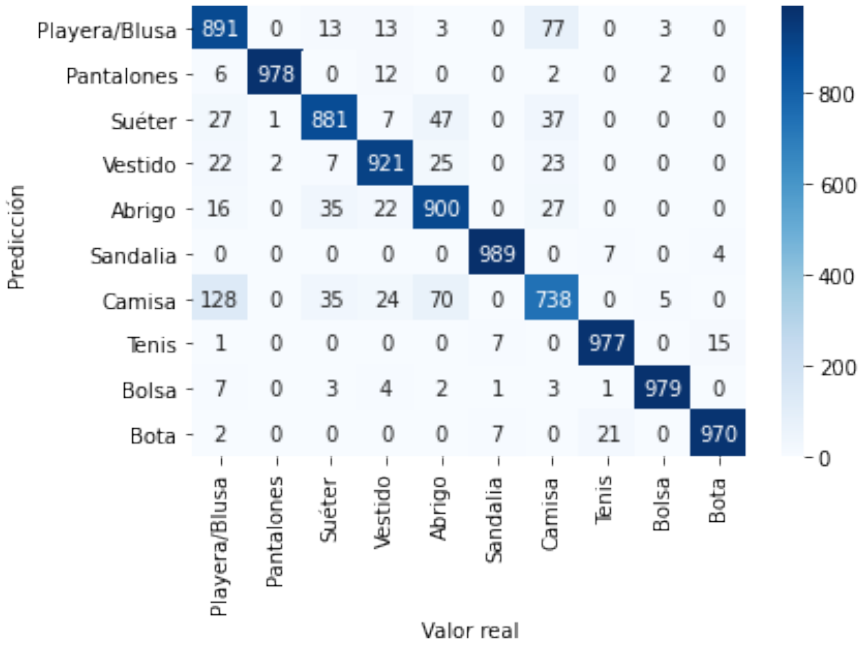


Figura 8. Matriz de confusión para el modelo proporcionado por *AutoKeras* con la base de datos *MNIST Fashion*.

Como se observa, el modelo proporcionado por métodos de *AutoML* es ligeramente más preciso con tan solo un solo intento a pesar de que el modelo proporcionado es casi igual de simple que el

creado en *Keras*. Ambos modelos se pueden comparar de mejor manera si se muestran las métricas de precisión y sensibilidad para cada caso, las cuales se presentan en la Tabla 1.

La base de datos para probar el modelo contiene 1000 imágenes de cada categoría, por lo que se puede decir que está bastante balanceada. Se observa que en ambos casos se tuvieron dificultades para diferenciar entre abrigos, camisas y playeras o blusas, al menos en las matrices se puede observar cómo se predijo que ciertas imágenes eran camisas cuando en realidad eran abrigos o playeras y viceversa. Sin embargo, en general, es evidente que el modelo proporcionado por *AutoKeras* es un poco más preciso.

Tabla 1. Sensibilidad y precisión de las matrices de confusión de la Figura 6 (Keras) y Figura 8 (*AutoKeras*) para el caso de la base de datos *MNIST Fashion.*

	Modelo en Keras		**Modelo en AutoKeras**	
	Sensibilidad (%)	**Precisión (%)**	**Sensibilidad (%)**	**Precisión (%)**
Playera/ Blusa	85	84	89	81
Pantalón	97	100	98	100
Suéter	87	88	88	90
Vestido	90	92	92	92
Abrigo	88	85	90	86
Sandalia	98	98	99	99
Camisa	74	73	74	81
Tenis	97	95	98	97
Bolsa	98	98	98	99
Bota	96	97	97	98

4.2 Cifar 10

A continuación, se muestran los resultados de las implementaciones de Keras (ML) vs AutoKeras (AutoML) sobre la base de datos *Cifar 10*.

4.2.1 Implementación con Keras (ML)

En la Figura 9 se encuentran ejemplos de las imágenes contenidas en el conjunto *Cifar 10*, cada una representa un ejemplo de las categorías presentes. A diferencia de las imágenes anteriores, estas cuentan con tres canales de color RGB y un tamaño de 32 × 32 píxeles.

El modelo se definió de manera similar al anterior con la diferencia de que se agregó otra capa más de convolución con 128 filtros junto con otra de pooling, las mismas funciones de activación fueron utilizadas al igual que la función pérdida y el optimizador. El esquema del modelo se muestra a continuación en la Figura 10.

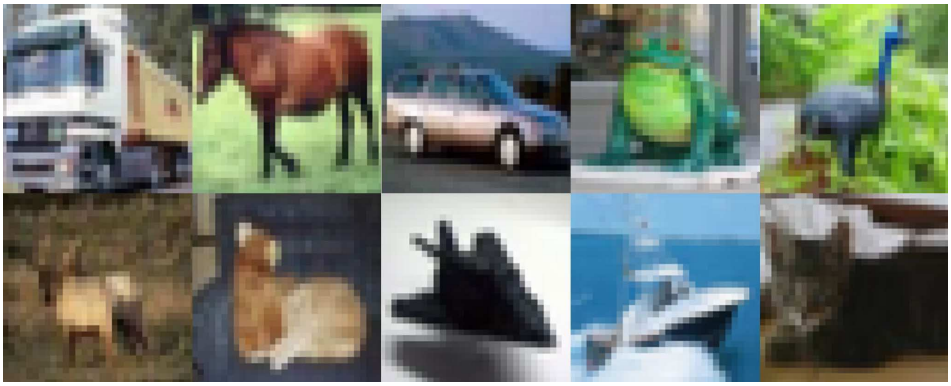


Figura 9. Muestras de las imágenes pertenecientes al conjunto *Cifar 10* [11].

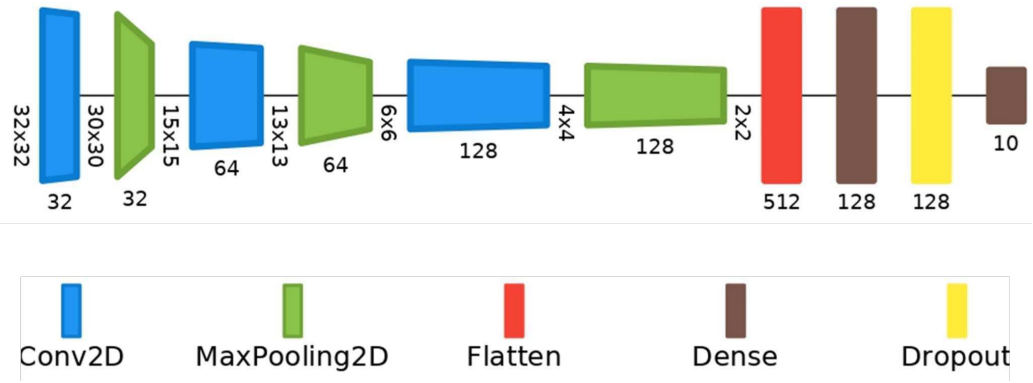


Figura 10. Esquema gráfico del modelo definido con Keras para la clasificación de imágenes del conjunto *Cifar 10*.

Con esta base de datos, el modelo tuvo un valor de pérdida de alrededor de 0.89 con una exactitud del 69%, el cual muestra que el rendimiento del modelo no es realmente adecuado. Similarmente, se muestran en la Figura 11 y 12 las gráficas de la evolución del valor de pérdida y exactitud y la matriz de confusión respectivamente. Para este caso, en la Figura 11 se observa que fueron necesarias 500 épocas para lograr el mejor valor de pérdida y exactitud, de igual manera, existe un ascenso y descenso bastante rápido entre las 100 primeras épocas. En general el comportamiento parece bueno ya que no da señales de sobreajuste, lo que significa que ni la pérdida aumenta ni la precisión se estanca o disminuye durante el proceso de validación.

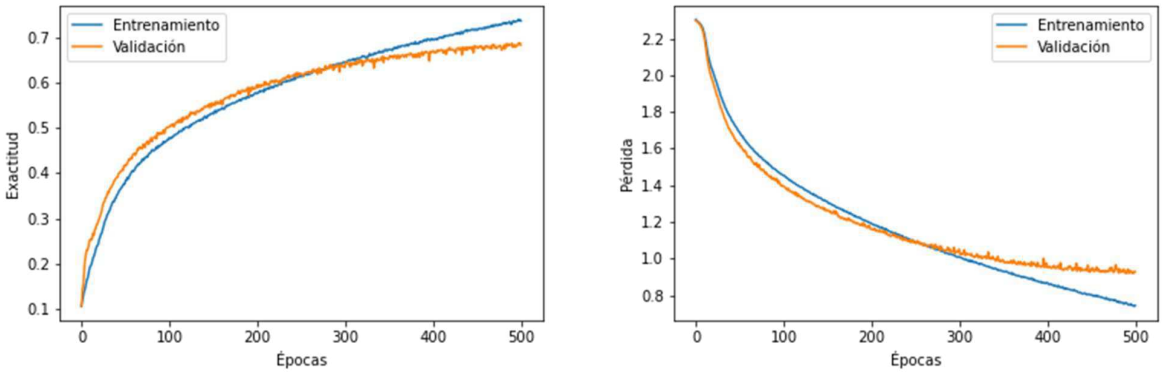


Figura 11. Evolución del valor de exactitud (derecha) y pérdida (izquierda) respecto a las épocas transcurridas utilizando la base de datos *Cifar 10*.

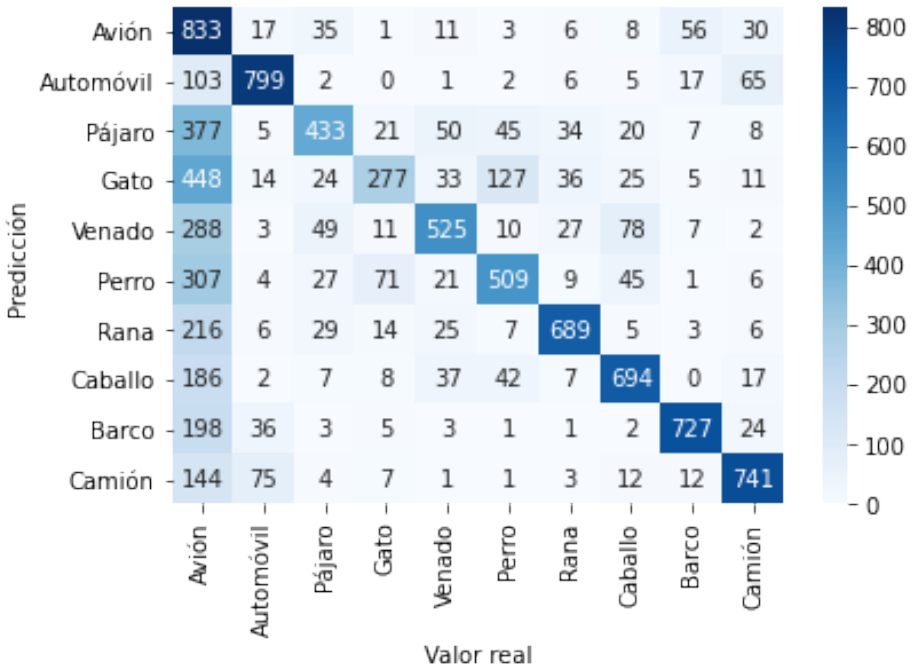


Figura 12. Matriz de confusión para el modelo creado en Keras con la base de datos *Cifar 10*.

4.2.2 Implementación con AutoKeras (AutoML)

En el caso de *AutoKeras* se obtuvo un resultado más destacado, a pesar de que todavía el comportamiento del modelo no es del todo conveniente, es una mejora en cuanto al resultado anterior como se verá más adelante. El resumen del modelo proporcionado se encuentra en la Figura 13.

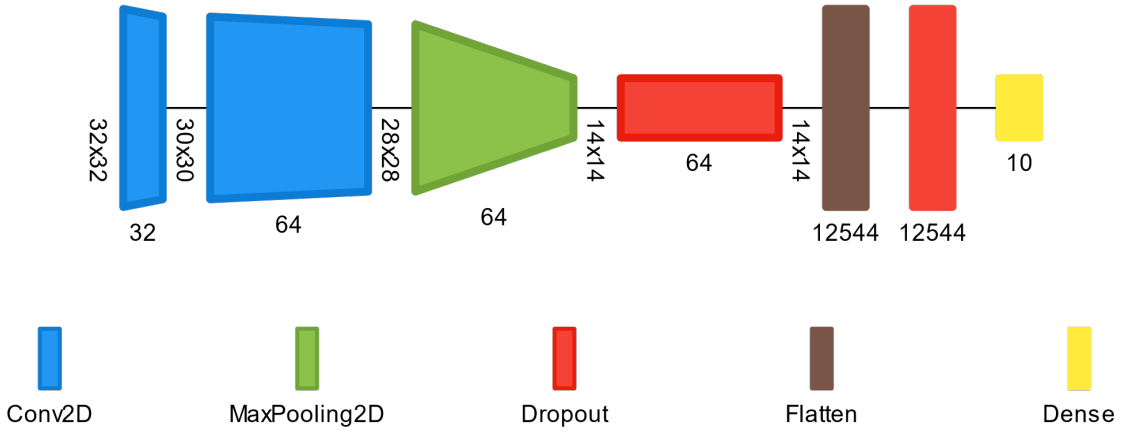


Figura 13. Esquema gráfico del modelo definido por *AutoKeras* para la clasificación de imágenes del conjunto *Cifar 10.*

De esta manera se obtuvo un valor de pérdida de 0.79 con una exactitud del 75%, los cuales, de nuevo, representan una mejora respecto al modelo realizado con *Keras*, sin embargo, es importante destacar que se si hubieran permitido más intentos para buscar diferentes arquitecturas es probable que se hubiera encontrado un modelo que predijera con mayor precisión. La matriz de confusión para este caso se muestra en la Figura 14, de igual manera se muestra la sensibilidad y la precisión alcanzadas para ambos casos en la Tabla 2.

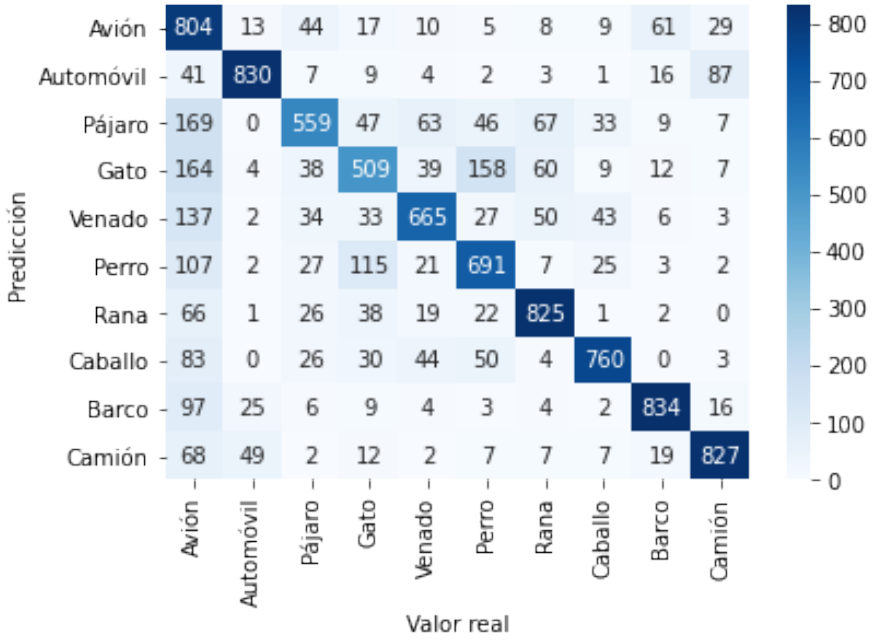


Figura 14. Matriz de confusión para el modelo proporcionado por AutoKeras con la base de datos Cifar 10.

Tabla 2. Sensibilidad y precisión de las matrices de confusión de la Figura 6 (*Keras*) y Figura 8 (*AutoKeras*) para el caso de la base de datos *Cifar 10*.

	Modelo en *Keras*		**Modelo en *AutoKeras***	
	Sensibilidad (%)	**Precisión (%)**	**Sensibilidad (%)**	**Precisión (%)**
Avión	83	27	80	46
Automóvil	80	83	83	90
Pájaro	43	71	56	73
Gato	28	67	51	62
Venado	53	74	67	76
Perro	51	68	69	68
Rana	69	84	82	80
Caballo	69	78	76	85
Barco	73	87	83	87
Camión	74	81	83	84

Al igual que en el caso anterior, el conjunto de imágenes de prueba está balanceado ya que contiene 1000 imágenes por categoría, y también el modelo proporcionado por *AutoKeras* llega a ser más preciso, sin embargo, no es una mejora considerable debido a que en algunos casos llega a ser superado o rinde de la misma manera que el modelo definido manualmente en *Keras*, como se puede ver en los casos del gato, venado y perro. Por otra parte, ambos tienen problemas al clasificar imágenes de aviones, aunque el segundo modelo realiza una mejor tarea, con una precisión del 46% en contraste con el 27% del primero. A grandes rasgos el modelo obtenido por métodos de *AutoML* presenta mejoras, sin embargo, es importante recalcar que solo se probó un tipo de arquitectura, por lo que, si se permitieran más intentos, probablemente el modelo hecho en *Keras* sería superado con creces.

4.3 ISBI 2017

A continuación, se muestran los resultados de las implementaciones de Keras (ML) vs AutoKeras (AutoML) sobre la base de datos *ISBI 2017*.

4.3.1 Implementación con Keras (ML)

Esta base de datos está formada por imágenes más complejas, pues ciertas lesiones de piel llegan a ser bastante parecidas entre ellas. En la Figura 15 se puede apreciar un arreglo de ocho imágenes, las cuatro pertenecientes a la fila de arriba corresponden a melanoma, mientras que las cuatro de la fila de abajo corresponden a imágenes de nevos benignos o de queratosis seborreica. El tamaño de las imágenes tuvo que ser modificado de manera que ahora tienen un tamaño de 224 × 224 píxeles con tres canales de color RGB, cabe destacar que antes del cambio de tamaño las imágenes variaban en cuanto a tamaño y resolución, por lo que algunas se vieron más afectadas que otras durante este proceso.

Se definió un modelo sencillo de dos capas *Conv2D* junto con dos de *pooling*, en los dos casos anteriores se habían usado las funciones de activación *softmax* y *relu*, sin embargo, para este caso se seguirá utilizando la función *relu* junto con una función *sigmoidal* para la predicción del modelo. Esta función arroja un valor de 0 para valores muy negativos, o 1 para valores muy positivos [2], lo que justifica la elección de dicha función ya que el trabajo del modelo es predecir si una imagen corresponde a un caso de melanoma o no.

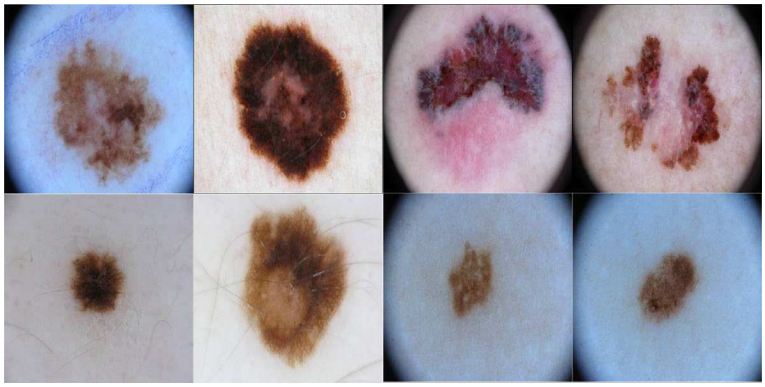


Figura 15. Muestras de las imágenes pertenecientes al conjunto del reto *ISBI 2017* [13].

En la Figura 16, se presenta la definición del modelo representado en un esquema.

De esta manera se obtuvo un valor de pérdida de 1.85 junto con una exactitud de aproximadamente 76%, por lo que ambos resultados no parecen ser adecuados para el correcto rendimiento en este tipo de situaciones, especialmente el del valor de pérdida el cual es mayor a un punto respecto a los valores de las otras dos pruebas. Para un análisis a detalle se consideran también las gráficas de la evolución de ambos valores al igual que la matriz de confusión. En la Figura 17 se muestra que tanto la pérdida como la exactitud no mejoran más que en el primer par de épocas, mientras

que en el proceso de entrenamiento sucede todo lo contrario, por lo que nos encontramos con un sobreajuste del modelo.

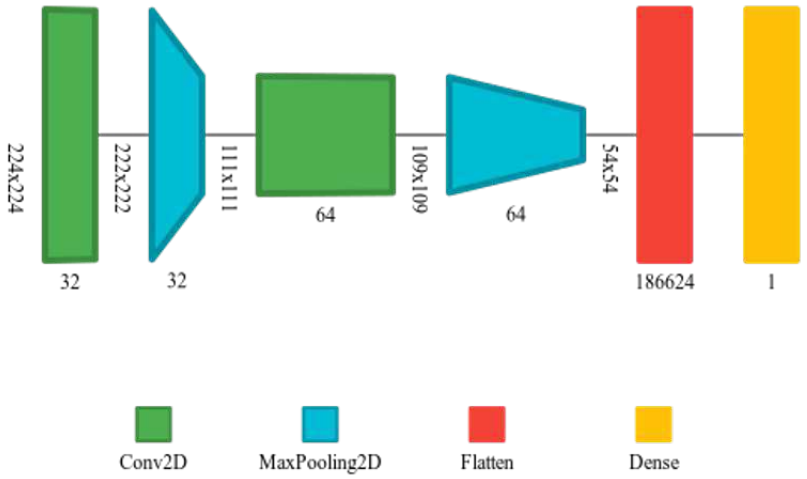


Figura 16. Esquema gráfico del modelo definido con Keras para la clasificación de imágenes del conjunto del reto *ISBI 2017*.

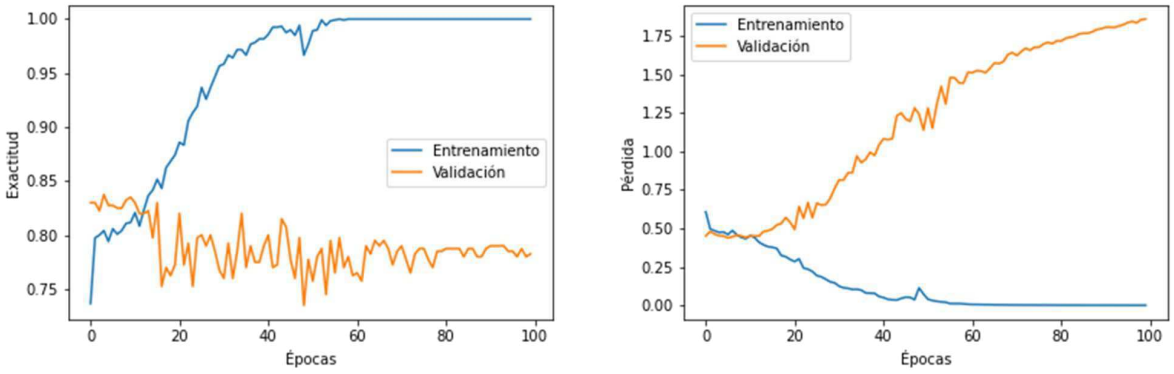


Figura 17. Evolución del valor de exactitud (derecha) y pérdida (izquierda) respecto a las épocas transcurridas utilizando la base de datos del reto *ISBI 2017*.

Se puede observar que el modelo es capaz de reconocer algunos casos de melanoma, lo que es una buena señal, ya que, utilizando la misma arquitectura utilizada en el caso del *Cifar 10*, se obtuvo una exactitud de alrededor del 85% y un valor de pérdida de 0.43. Sin embargo, al momento de analizar la matriz de confusión mostrada en la Figura 18 se observó que existía cierta incapacidad de detectar melanoma, lo cual tiene sentido, pues el conjunto de imágenes tan solo cuenta con 252 imágenes de melanoma.

4.3.2 Implementación con AutoKeras (AutoML)

Ahora, con *AutoKeras* fue posible intentar con cuatro tipos de arquitecturas diferentes, y para esta prueba se obtuvieron resultados más complejos. El esquema del modelo que mejores resultados ofreció se presenta a continuación

Se puede observar en la Figura 19 que ya no se informa acerca del número de capas, tipo de capas, etc. Más bien, se muestran el preprocesamiento realizado en las imágenes al igual que la introducción de una arquitectura *EfficientNet-B7*, la cual consiste en 813 capas distribuidas en 7 bloques, los cuales a su vez contienen módulos con nada más que capas convolucionales y de *pooling* [17].

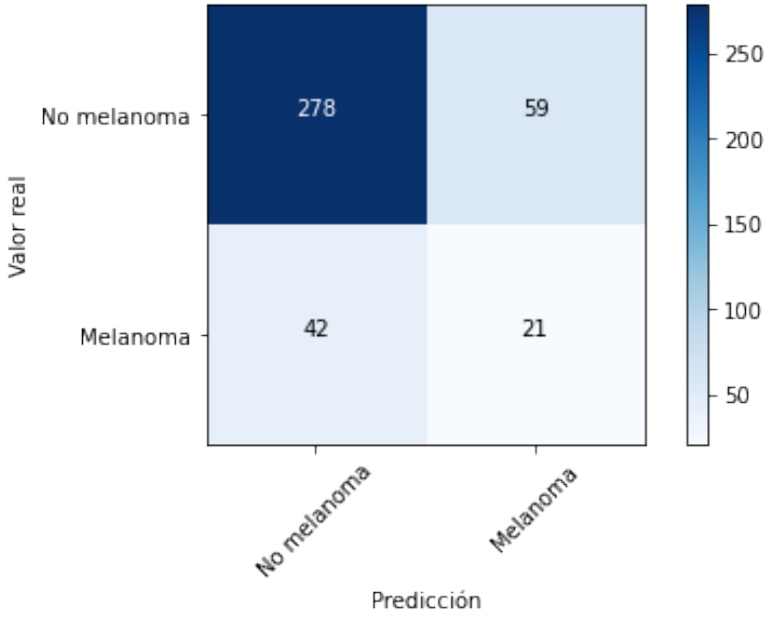


Figura 18. Matriz de confusión para el modelo creado en Keras con la base de datos del reto *ISBI 2017*.

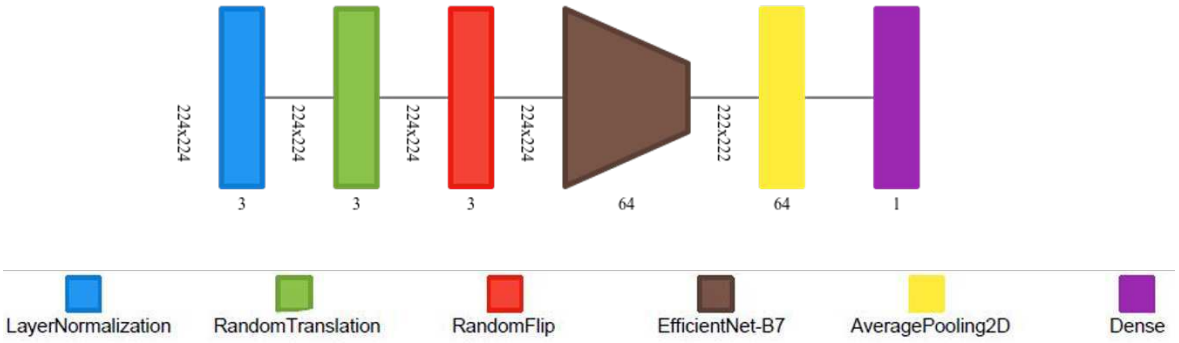


Figura 19. Esquema gráfico del modelo definido por *AutoKeras* para la clasificación de imágenes del conjunto del reto *ISBI 2017*.

Para analizar el comportamiento del modelo proporcionado por *AutoKeras* se muestra también una matriz de confusión en la Figura 20.

Para este caso las matrices de confusión gozan de más importancia debido a que aproximadamente el 12% de las imágenes corresponden a melanoma, lo cual es un factor importante para tomar en cuenta para ver realmente que tan bien los modelos desempeñan sus tareas. Considere el caso más importante, el cual es la detección de melanoma, el modelo definido en *Keras* tiene una sensibilidad de tan sólo el 22%, mientras que el otro tiene un 52%, lo que significa que el segundo modelo es más hábil para clasificar correctamente melanoma. Sin embargo, si se habla de precisión la diferencia es mucho más marcada puesto que con *Keras* se obtiene una precisión del 24%, mientras que con *AutoKeras* se obtiene casi un 65%. Es aquí donde es importante mencionar que la exactitud que se obtiene después de entrenar y probar los modelos puede ser engañosa, ya que no toma en cuenta la distribución, en este caso, de imágenes de melanoma en el conjunto [9], sin embargo, con la precisión se obtiene una perspectiva diferente. Debido a lo anterior se puede decir que el modelo proporcionado por *AutoKeras* rinde mucho mejor, sin embargo, para fines prácticos todavía debe mejorar.

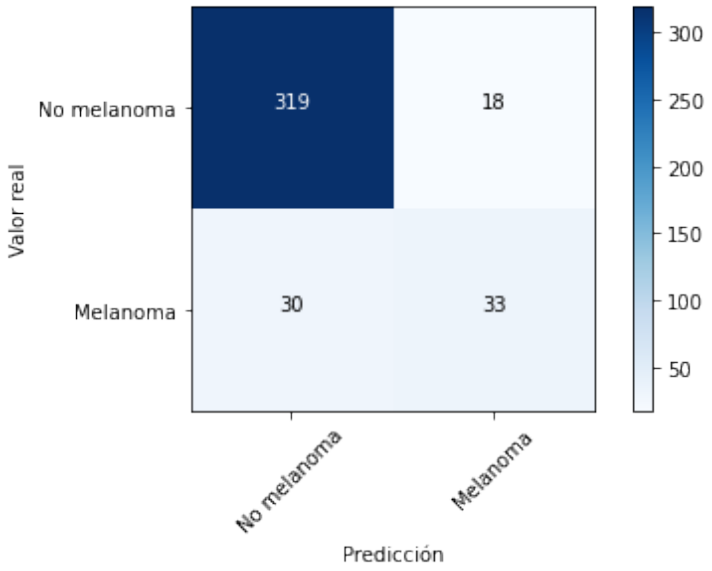


Figura 20. Matriz de confusión para el modelo proporcionado por *AutoKeras* con la base de datos del reto *ISBI 2017*.

A continuación, en la Tabla 3 se muestran de manera condensada los resultados obtenidos para cada caso, resaltando las diferencias entre el modelo creado manualmente y el modelo proporcionados vía *AutoML*. Es evidente que *AutoKeras* fue capaz de superar nuestros modelos en los tres casos. Más aún, si se le permitiera probar con más tipos de arquitecturas, es probable que el desempeño de estos modelos sea mucho mejor. En la base de datos *MNIST Fashion*, ambos modelos rindieron de una manera satisfactoria. Además, en la tabla se incluyen las líneas ejecutadas, sólo contando las utilizadas para la definición, compilación y evaluación del modelo. Se puede observar que para estos casos se ejecutan entre 3 y 4 veces más líneas de código utilizando Keras. Algo muy importante por mencionar es que los casos donde los patrones sean más complejos seguramente se requerirán modelos neuronales más grandes lo cual se incidirá en el número de líneas a usar de la librería de Keras, pero no así en AutoKeras u otro sistema de *AutoML* cuya definición no requiere más codificación.

Tabla 3. Comparación de resultados obtenidos para cada base de datos utilizando *Keras* y *AutoKeras*.

	Exactitud (%)			Precisión promedio (%)			Sensibilidad promedio (%)			Líneas de código ejecutadas		
	MNIST Fashion	*Cifar 10*	*ISBI 2017*	*MNIST Fashion*	*Cifar 10*	*ISBI 2017*	*MNIST Fashion*	*Cifar 10*	*ISBI 2017*	*MNIST Fashion*	*Cifar 10*	*ISBI 2017*
Keras	91	69	76	91	72	24	91	62	22	~13	~15	~11
Auto Keras	**~93**	**75**	**85**	**92**	**75**	**65**	**92**	**73**	**52**	**~4**	**~4**	**~4**

5. Conclusiones

En este trabajo se mostró un análisis cuantitativo sobre el rendimiento de dos paradigmas importantes de la actualidad en la inteligencia artificial: el aprendizaje máquina (ML) y el auto aprendizaje máquina (AutoML) al resolver el mismo problema. Los resultados obtenidos sobre tres bases de datos

distintas mostraron una clara superioridad del enfoque AutoML sobre ML en términos de exactitud, precisión, sensibilidad y líneas codificadas cuando el desarrollador no tiene gran experiencia. Lo anterior, es algo importante; ya que ilustra la posibilidad de realizar modelos neuronales eficaces y confiables a expertos de áreas ajenas a la IA, que ahora requieren conocimientos mínimos del área, permitiendo enfocar sus esfuerzos al problema en sí y no a su implementación. Como trabajo futuro pretendemos desarrollar una metodología híbrida que combine ambos paradigmas, con la finalidad de alcanzar mejores resultados que los enfoques por sí solos.

Referencias

[1] Chollet, F. (2018). *Deep learning with Python*. Manning Publications Co.
[2] Nielsen, M. A. (2015). *Neural networks and deep learning*. Neural Networks And Deep Learning. http://neuralnetworksanddeeplearning.com/
[3] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
[4] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.
[5] Jin, H., Song, Q., & Hu, X. (2019). Auto-Keras: An Efficient Neural Architecture Search System. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1946–1956. https://doi.org/10.1145/3292500.3330648
[6] Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, *29*(3), 31-44.
[7] Colaboradores de Wikipedia. (2021, May 21). *Red neuronal artificial*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Red_neuronal_artificial
[8] Calvo, D. (2018, December 8). *Red Neuronal Convolucional CNN*. Diego Calvo. https://www.diegocalvo.es/red-neuronal-convolucional/
[9] Brownlee, J. (2020, August 20). A Gentle Introduction to the Rectified Linear Unit (ReLU). Machine Learning Mastery. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
[10] Brownlee, J. (2019, August 6). *A Gentle Introduction to Dropout for Regularizing Deep Neural Networks*. Machine Learning Mastery. https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/
[11] Novaković, J. D., Veljović, A., Ilić, S. S., Papić, Ž., & Milica, T. (2017). Evaluation of classification models in machine learning. Theory and Applications of Mathematics & Computer Science, 7(1), 39-46.
[12] Cambridge Dictionary. (2021). *Significado de precisión en inglés*. https://dictionary.cambridge.org/es/diccionario/ingles/precision
[13] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
[14] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.
[15] Codella, N. C. F., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., et al. (2018). Skin lesion analysis toward melanoma detection: A challenge at the 2017 International symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC). 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), 168-172. https://doi.org/10.1109/ISBI.2018.8363547.
[16] Keras Team. (n.d.). Keras documentation: The Sequential model. Keras. https://keras.io/guides/sequential_model/
[17] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (pp. 6105-6114). PMLR

Diseño Colaborativo en Mecatrónica, Capítulo 14, pp. 185 - 202
ISBN: 978-607-9394-23-3, 2021

Protección contra Vibraciones Fuera de Rango en Estructuras de 1 gdl

Candia García Filiberto[1], Ambrosio Lázaro Roberto A.[2], Vivaldo de la Cruz Israel[2], Flores Méndez Javier[2], Benítez Cruz Gerardo A.[1]

Facultad de Ingeniería, Benemérita Universidad Autónoma de Puebla, México.
Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla, México
filiberto.candia@correo.buap.mx

Resumen

El tema de las vibraciones mecánicas toma día con día mayor interés en el diseño de los nuevos productos, sistemas o dispositivos mecatrónicos. Por ello el desarrollar, fabricar y validar dispositivos de medición y protección a bajo costo, es una propuesta de alta factibilidad, y de gran apoyo en el sector académico. Contar con dispositivos didácticos en las instituciones educativas, acerca a los estudiantes a la detección de fallas y cuidado de las estructuras que alojan a las maquinarias o dispositivos rotatorios. Contar con una experiencia previa en el manejo de prototipos que proporcionen datos, interfaces que analicen la información y personal que tome decisiones a favor de la protección de las estructuras de 1 gdl.

Palabras clave: Vibraciones mecánicas, Estructuras 1 gdl, Dispositivo Mecatrónico, Arduino UNO, PLC.

1. Introducción

El presente desarrollo tecnológico, consiste en el diseño de un dispositivo mecatrónico que monitorea en tiempo real el estado de las vibraciones de una estructura destinada a soportar máquinas generadoras de movimientos armónicos (aerogeneradores, bombas, compresores, ventiladores, turbinas, entre otras) que afectan la rigidez de una estructura que ha sido ensamblada por medio de remaches o tornillos. Esta tipología estructural se ha determinado por ser una recurrente solución para la instalación de máquinas/equipos dentro de las instalaciones/procesos industriales de producción o fabricación, debido a que es una incorporación al proyecto original (ya sea para incrementar la eficiencia o eficacia o para compensar errores en el proyecto original). Asimismo, es una práctica usual en las empresas ensambladoras (industria automotriz, aeronáutica, etc.) o de logística (FEDEX, UPS, DHL) cuando se instalan transportadores (bandas, rodillos, etc.) para desplazar los materiales o productos aun no terminados. Por ello es indispensable que las estructuras (figura 1) destinadas a mantener las máquinas/equipos que producen movimientos armónicos, sean lo más rígidas posible, ya que de lo contrario se pueden provocar valores peligrosos de variación de rigidez (pérdida) y generar inclusive condiciones de resonancia [1], [2], [3].

Figura 1. Estructuras de soporte para máquina/equipo rotatorio.

El objetivo por lo tanto es desarrollar un sistema de monitoreo de bajo costo basado en el PLC S7-1200 y Arduino UNO. Con la finalidad de incorporar un sistema de control que permita la desconexión de los equipos de manera segura dentro del proceso y emita las correspondientes alarmas de seguridad, que de acuerdo al grado de riesgo indique la posible falla y su procedimiento de corrección. Los resultados a obtener serán un dispositivo capaz de monitorear condiciones de falla en máquinas/equipos, para determinar la frecuencia de vibración, periodo del movimiento y sus valores máximos periódicos, a través de gráficas en función del tiempo y frecuencia.

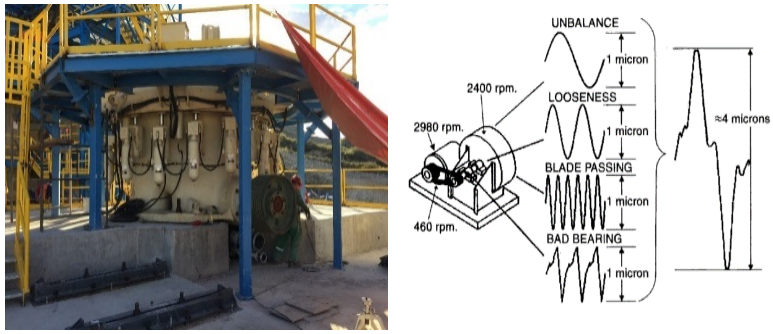


Figura 2. Dispositivo generador de movimientos armónicos de baja frecuencia.

Se ha concluido que el monitorear el comportamiento de la rigidez de una estructura ante el efecto y consecuencias de las cargas dinámicas, es una solución confiable y económica que protege los bienes materiales de mayor relevancia en una empresa.

1.1 Descripción de la situación

Cuando una máquina/equipo rotativo es soportado por una estructura rectangular de acero que ha sido fabricada con remaches o tornillos como medio de sujeción en lugar de soldadura o maquinado en una solo pieza. La estructura debido a la carga armónica permanente tiende a perder rigidez (figura 3) por el aflojamiento de los medios de sujeción, los cuales al alcanzar niveles de resonancia tienden a generar fisuras y falla estructural. Sobre todo, en dispositivos como las turbinas o aerogeneradores que estas destinados a la producción de energía eléctrica.

Figura 3. Ejemplos de perdida de rigidez.

2. Método

La investigación es experimental y el método de solución es científico-deductivo-propositivo. El estudio de referencia involucra la definición de conceptos como vibración y resonancia [1], [2]. La organización del trabajo a desarrollar se estructura en el diagrama de bloques mostrado en la figura 4.

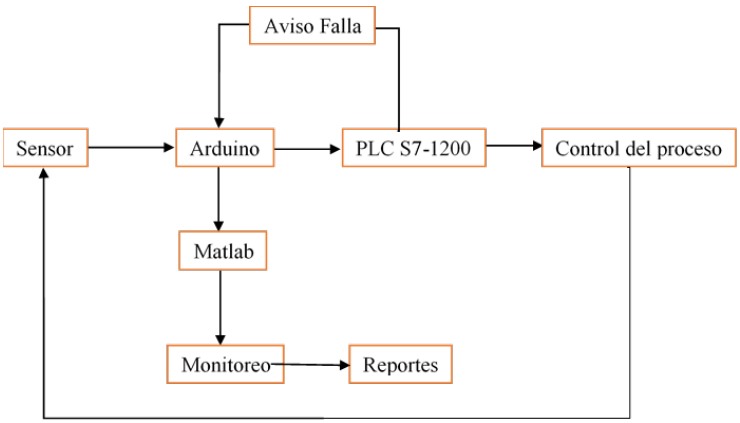


Figura 4. Diagrama de bloques del dispositivo a desarrollar.

2.1 *Estudio técnico-económico*

La relación costo beneficio es favorable, debido a que el dispositivo es relativamente económico con respecto al costo de un equipo rotatorio industrial, que puede verse afectado y destruido por la pérdida de rigidez de la estructura que lo soporta o por entrar en resonancia. La falta de monitoreo conlleva a situaciones donde el daño se hace evidente una vez que la falla es crítica y/o demasiado costosa de reparar, implementar sistemas de monitoreo de salud estructural representa una inversión sumamente rentable, permitiendo reducir costos altamente significativos de mantenimiento, pérdidas de producción por reparaciones, y accidentes que puedan afectar la integridad física de personas que se encuentren cerca de las estructuras al detectar problemas en su etapa más temprana, así como asegurar un tiempo de vida útil de los bienes materiales [4], [5].

3. Desarrollo

El desarrollo es experimental-pragmática, apoyada para su desarrollo en el software de simulación analítica (Matlab/Simulink), en el cual se realiza la simulación de comportamientos límite y posibles consecuencias de fluencia. Para posteriormente integrar tecnologías de hardware y software, que contribuyan al control de la vibración. El desarrollo se divide en las siguientes etapas:

1. Concepción de la idea
2. Determinación de las condiciones de riesgo
3. Selección de equipos materiales y dispositivos
4. Desarrollo de planos mecánicos
5. Desarrollo de planos eléctricos
6. Programación del Arduino UNO.
7. Programación del PLC
8. Integración de mediante de los dispositivos mediante una ampliación como Matlab
9. Monitoreo de prueba y asociación de fallas con eventualidades
10. Instalación y puesta en marcha

3.1 Diagramas de conexión

- La figura 5, muestra el esquema conceptual entre la conexión de entre tecnologías.
- La figura 6, es el diagrama esquemático de conexión del Arduino UNO.
- La figura 7, es el diagrama de alimentación y conexión del PLC
- La figura 8, muestra la conexión de alambrado físico.
- La figura 9, es el diagrama esquemático de alambrado.

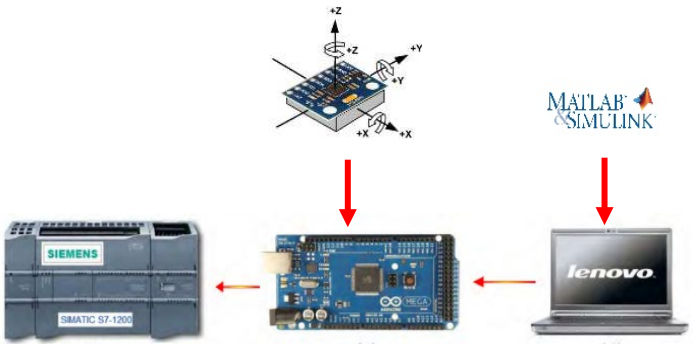


Figura 5. Esquema conceptual de la conexión e integración.

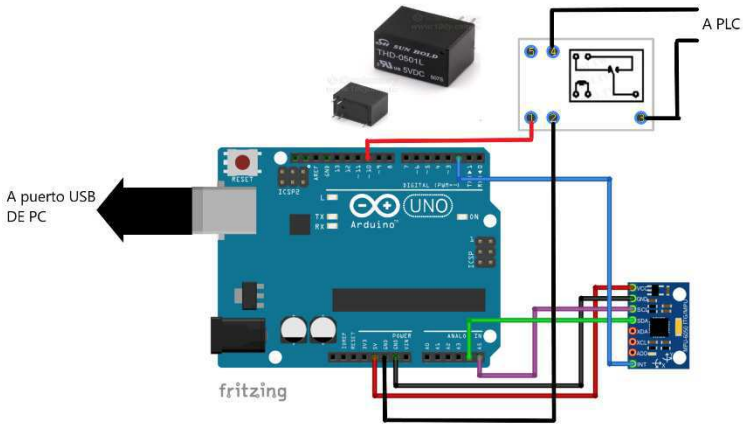


Figura 6. Aproximación conceptual para el desarrollo del esquema eléctrico.

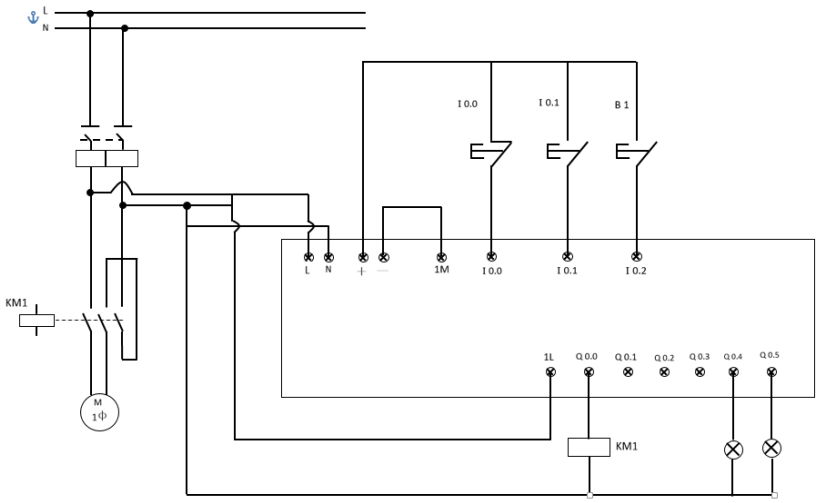


Figura 7. Diagrama de conexión de PLC S7-1200.

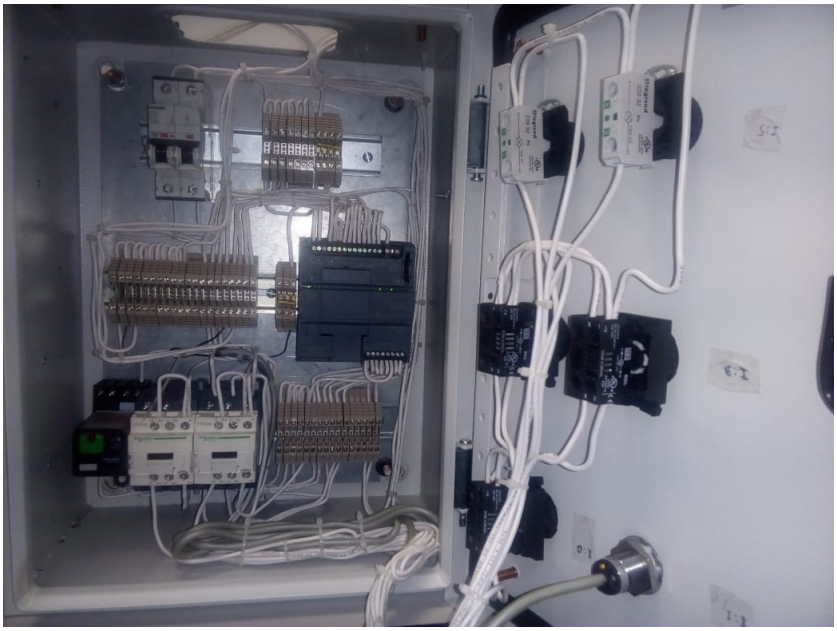


Figura 8. Conexiones de PLC en gabinete.

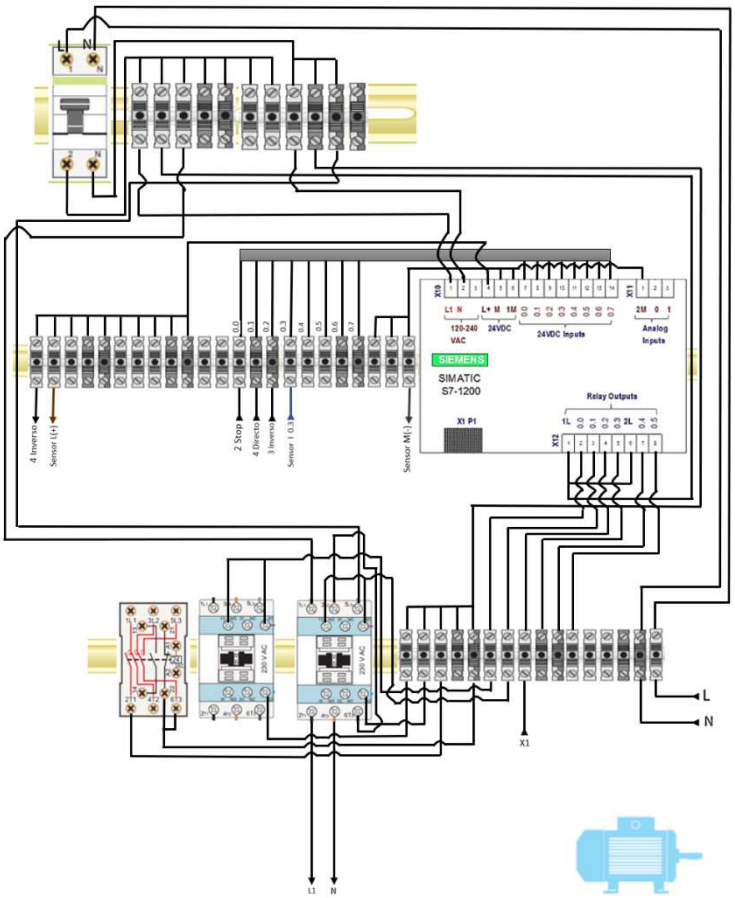


Figura 9. Representación de conexión de PLC y carga de la estructura.

3.2 Programación del PLC

El PLC de SIMATIC S7-1200 cuenta con memoria de trabajo 50 KB; Fuente de alimentación de 120 / 240VCA con DI8 x 24VDC SINK / SOURCE, relé DQ6 x y AI2 a bordo; 4 contadores de alta velocidad (ampliables con placa de señal digital) y 4 salidas de pulso a bordo; la placa de señal expande las E / S a bordo; hasta 3 módulos de comunicación para comunicación en serie; hasta 2 módulos de señal para expansión de E / S; 0.04 ms / 1000 instrucciones; Interfaz PROFINET para programación, HMI y comunicación PLC a PLC. Con sobrada capacidad y robustez del PLC, la figura 10 muestra la programación básica realizada en lenguaje escalera.

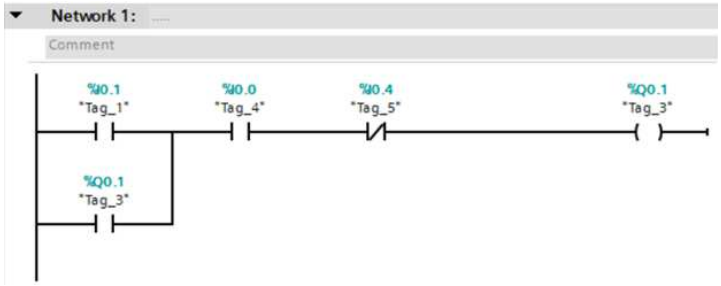


Figura10. Programación de PLC SIMATIC S7-1200.

3.3 Programación de placa Arduino UNO en MATLAB y Simulink

Para tener control en caso de detectarse una condición insegura de trabajo, así como para recolectar muestras de aceleración para monitorear el estado de la estructura, se realizó un script que automatizara las tareas de inicio de la simulación, recolección de muestras del acelerómetro MPU-6050, análisis de las muestras por la transformada rápida de Fourier (FFT), determinar los picos de amplitud máxima en G y mm/s^2 y la frecuencia en Hz donde se presentan dichas amplitudes.

Se utilizaron 2 scripts, el primero tiene la función de realizar las gráficas en función de los valores resultantes de la transformada rápida de Fourier (FFT) e indicar los valores máximos y promedio de amplitud en G y mm/s^2, el segundo se encarga del inicio de la simulación, finalización de la misma, realizar el procesamiento de las señales mediante FFT, obtener los valores máximos de desplazamiento en mm detectados en cada eje (x, y, z) y determinar las variables a graficar por el primer script.

Script 1

```
function createfigure(X1, Y1)
%CREATEFIGURE(X1, Y1)
%  X1:  vector of x data
%  Y1:  vector of y data

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
hold(axes1,'on');

% Create plot
plot1 = plot(X1,Y1,'DisplayName','data1');

% Get xdata from plot for data statistics
xdata1 = get(plot1, 'xdata');
% Get ydata from plot for data statistics
ydata1 = get(plot1, 'ydata');
% Make sure data are column vectors
xdata1 = xdata1(:);
ydata1 = ydata1(:);
 % Get axes ylim
axXLim1 = get(axes1,'xlim');
% Find the max
ymax1 = max(ydata1);
% Get coordinates for the max line
maxValue1 = [ymax1 ymax1];
% Plot the max
statLine1 =
plot(axXLim1,maxValue1,'DisplayName','   y
max','Tag','max y',...
    'Parent',axes1,...
    'LineStyle','-.',...
    'Color',[0 0 1]);

% Set new line in proper position
setLineOrder(axes1,statLine1,plot1);

% Find the mean
ymean1 = mean(ydata1);
% Get coordinates for the mean line
meanValue1 = [ymean1 ymean1];
% Plot the mean
statLine2 =
plot(axXLim1,meanValue1,'DisplayName','   y
mean',...
    'Tag','mean y',...
    'Parent',axes1,...
```

```
    'LineStyle','-.',...
    'Color',[0 0.5 0]);

% Set new line in proper position
setLineOrder(axes1,statLine2,plot1);

% Create ylabel
ylabel('Accel (g)');

% Create xlabel
xlabel('Frequency (Hz)');

% Create title
title('eje x');

% Uncomment the following line to preserve
the X-limits of the axes
% xlim(axes1,[0 60]);
% Uncomment the following line to preserve
the Y-limits of the axes
% ylim(axes1,[0 0.35]);
box(axes1,'on');
grid(axes1,'on');
% Create legend
legend(axes1,'show');

%-------------------------------------%
function setLineOrder(axesh1, newLine1,
associatedLine1)
%SETLINEORDER(AXESH1,NEWLINE1,ASSOCIATEDLINE1
)
%  Set line order
%  AXESH1:  axes
%  NEWLINE1:  new line
%  ASSOCIATEDLINE1:  associated line

% Get the axes children
hChildren = get(axesh1,'Children');
% Remove the new line
hChildren(hChildren==newLine1) = [];
% Get the index to the associatedLine
lineIndex = find(hChildren==associatedLine1);
% Reorder lines so the new line appears with
associated data
hNewChildren = [hChildren(1:lineIndex-
1);newLine1;hChildren(lineIndex:end)];
% Set the children:
set(axesh1,'Children',hNewChildren);
```

Script 2

```
%Inicio de simulación en simulink
%Especifique tiempo de simulación en segundos
set_param('sensor_yz','Simulationcommand','st
art')
set_param('sensor_yz','Simulationcommand','st
op','StopTime','30')

%Establecer un tiempo de pausa del script
mayor al del tiempo de la
%simulación, por ejemplo si se establecen 20
segundos de simulación,
%ponemos 35 segundos de pausa.
pause(35)


%Gráfica de recolección de señales del sensor
sin implementación de fft
figure()
plot(acxyz.signals.values)
title('Lecturas de sensor MPU-6050 sin fft')
xlabel('tiempo (ms)')
ylabel('Aceleración (G)')
grid on

x=acxyz.signals.values
t=acxyz.time(:,1) %Establecer el tiempo de la
muestra para fft
Fs=1/(t(2)-t(1)) %obtener la frecuencia de
muestreo para f  1   ft
[N,m]=size(acxyz.signals.values)
Fs=1/(t(2)-t(1)) %obtener la frecuencia de
muestreo para fft
fprintf('%12.0f data points\n',N)
freq = 0:Fs/length(x):Fs/2; %frequency array
for FFT
xdft=fft(x)
xdft = 1/length(x).*xdft; %Normalize
xdft(2:end-1) = 2*xdft(2:end-1);


%Gráfica de fft para eje x
ejx=acxyz.signals.values(:,1)
tx=acxyz.time(:,1) %Establecer el tiempo de
la muestra para fft
Fsx=1/(tx(2)-tx(1)) %obtener la frecuencia de
muestreo para fft
[N,m]=size(acxyz.signals.values(:,1))
fprintf('%12.0f data points\n',N)
freqx = 0:Fsx/length(ejx):Fsx/2; %frequency
array for FFT
ejxdft=fft(ejx)
ejxdft = 1/length(ejx).*ejxdft; %Normalize
ejxdft(2:end-1) = 2*ejxdft(2:end-1);

figure()
grafft(freqx,abs(ejxdft(1:floor(N/2)+1)))
xlabel('Frequency (Hz)');
ylabel('Accel (g)');
title('eje x');

grid on

%Gráfica de fft para eje Y
ejy=acxyz.signals.values(:,2)
ty=acxyz.time(:,1) %Establecer el tiempo de
la muestra para fft
Fsy=1/(ty(2)-ty(1)) %obtener la frecuencia de
muestreo para fft
[N,m]=size(acxyz.signals.values(:,1))
fprintf('%12.0f data points\n',N)
freqy = 0:Fsy/length(ejy):Fsy/2; %frequency
array for FFT
ejydft=fft(ejy)
ejydft = 1/length(ejy).*ejydft; %Normalize
ejydft(2:end-1) = 2*ejydft(2:end-1);
figure()
grafft(freqy,abs(ejydft(1:floor(N/2)+1)))
xlabel('Frequency (Hz)');
ylabel('Accel (g)');
title('eje y')
grid on

%Gráfica de fft para eje z
ejz=acxyz.signals.values(:,3)
tz=acxyz.time(:,1) %Establecer el tiempo de
la muestra para fft
Fsz=1/(tz(2)-tz(1)) %obtener la frecuencia de
muestreo para fft
[N,m]=size(acxyz.signals.values(:,1))
fprintf('%12.0f data points\n',N)
freqz = 0:Fsz/length(ejz):Fsz/2; %frequency
array for FFT
```

```
ejzdft=fft(ejz);
ejzdft = 1/length(ejz).*ejzdft; %Normalize
ejzdft(2:end-1) = 2*ejzdft(2:end-1);

figure()
grafft(freqz,abs(ejzdft(1:floor(N/2)+1)))
xlabel('Frequency (Hz)');
ylabel('Accel (g)');
title('eje z')
grid on

%-----------------------------%
%conversión de valores de G a mm
convgmm=(acxyz.signals.values)*9810
%Gráfica de recolección de señales del sensor
sin implementación de fft (en
%mm/s^2)

figure()
plot(convgmm)
title('Lecturas de sensor MPU-6050 sin fft')
xlabel('tiempo (ms)')
ylabel('Aceleración (mm/s^2)')
grid on

xmm=convgmm
t=acxyz.time(:,1) %Establecer el tiempo de la
muestra para fft
Fs=1/(t(2)-t(1)) %obtener la frecuencia de
muestreo para f  1   ft
[N,m]=size(acxyz.signals.values)
Fs=1/(t(2)-t(1)) %obtener la frecuencia de
muestreo para fft
fprintf('%12.0f data points\n',N)
freq = 0:Fs/length(xmm):Fs/2; %frequency
array for FFT
xmmdft=fft(xmm)
xmmdft = 1/length(xmm).*xmmdft; %Normalize
xmmdft(2:end-1) = 2*xmmdft(2:end-1);


%Gráfica de fft para eje x (en mm/s^2)
ejxmm=convgmm(:,1)
txmm=acxyz.time(:,1) %Establecer el tiempo de
la muestra para fft
Fsxmm=1/(txmm(2)-txmm(1)) %obtener la
frecuencia de muestreo para fft
[N,m]=size(acxyz.signals.values(:,1))
fprintf('%12.0f data points\n',N)
freqxmm = 0:Fsxmm/length(ejxmm):Fsxmm/2;
%frequency array for FFT
ejxdftmm=fft(ejxmm)
ejxdftmm = 1/length(ejxmm).*ejxdftmm;
%Normalize
ejxdftmm(2:end-1) = 2*ejxdftmm(2:end-1);

figure()
grafft(freqxmm,abs(ejxdftmm(1:floor(N/2)+1)))
xlabel('Frequency (Hz)');
ylabel('Accel (mm/s^2)');
title('eje x');

grid on

%Gráfica de fft para eje Y (en mm/s^2)
ejymm=convgmm(:,2)
tymm=acxyz.time(:,1) %Establecer el tiempo de
la muestra para fft
Fsymm=1/(tymm(2)-tymm(1)) %obtener la
frecuencia de muestreo para fft
[N,m]=size(acxyz.signals.values(:,1))
fprintf('%12.0f data points\n',N)
freqymm = 0:Fsymm/length(ejymm):Fsymm/2;
%frequency array for FFT
ejydftmm=fft(ejymm)
ejydftmm = 1/length(ejymm).*ejydftmm;
%Normalize
ejydftmm(2:end-1) = 2*ejydftmm(2:end-1);

figure()
grafft(freqymm,abs(ejydftmm(1:floor(N/2)+1)))
xlabel('Frequency (Hz)');
ylabel('Accel (mm/s^2)');
title('eje y')
grid on

%Gráfica de fft para eje z (en %mm/s^2)
ejzmm=convgmm(:,3)
tzmm=acxyz.time(:,1) %Establecer el tiempo de
la muestra para fft
Fszmm=1/(tzmm(2)-tzmm(1)) %obtener la
frecuencia de muestreo para fft
[N,m]=size(acxyz.signals.values(:,1))
fprintf('%12.0f data points\n',N)
freqzmm = 0:Fszmm/length(ejzmm):Fszmm/2;
%frequency array for FFT
ejzdftmm=fft(ejzmm);
ejzdftmm = 1/length(ejzmm).*ejzdftmm;
%Normalize
ejzdftmm(2:end-1) = 2*ejzdftmm(2:end-1);

figure()
grafft(freqzmm,abs(ejzdftmm(1:floor(N/2)+1)))
xlabel('Frequency (Hz)');
ylabel('Accel (mm/s^2)');
title('eje z')
grid on

%integrales para conocer el desplazamiento en
mm
% los valores de desplazamiento se guardan en
las variables del workspace
% de MATLAB, de x, y y z en las variables
smmx, smmy y smmz
% respectivamente.

%obtener máximos de aceleración
mxmm=max(convgmm)
%declarar función para realizar la integral y
conocer el desplazamiento en
%x en mm
fun1=@(tm) (mxmm(1,1)*tm)
%Declarar la variable con la que se guardará
el desplazamiento en x en mm
smmx=integral(fun1,0,0.01)
%declarar función para realizar la integral y
conocer el desplazamiento en
%y en mm
fun2=@(tm) (mxmm(1,2)*tm)
%Declarar la variable con la que se guardará
el desplazamiento en y en mm
smmy=integral(fun2,0,0.01)
%declarar función para realizar la integral y
conocer el desplazamiento en
%z en mm
fun3=@(tm) (mxmm(1,3)*tm)
%Declarar la variable con la que se guardará
el desplazamiento en z en mm
smmz=integral(fun3,0,0.01)
```

Con los scripts listos para realizar el procesamiento de las señales, se desarrolla el modelo en Simulink para establecer la comunicación entre el sensor MPU6050, la placa Arduino UNO, y la interfaz para el procesamiento de las señales, el cual consta de las siguientes operaciones para su ejecución (figura 11).

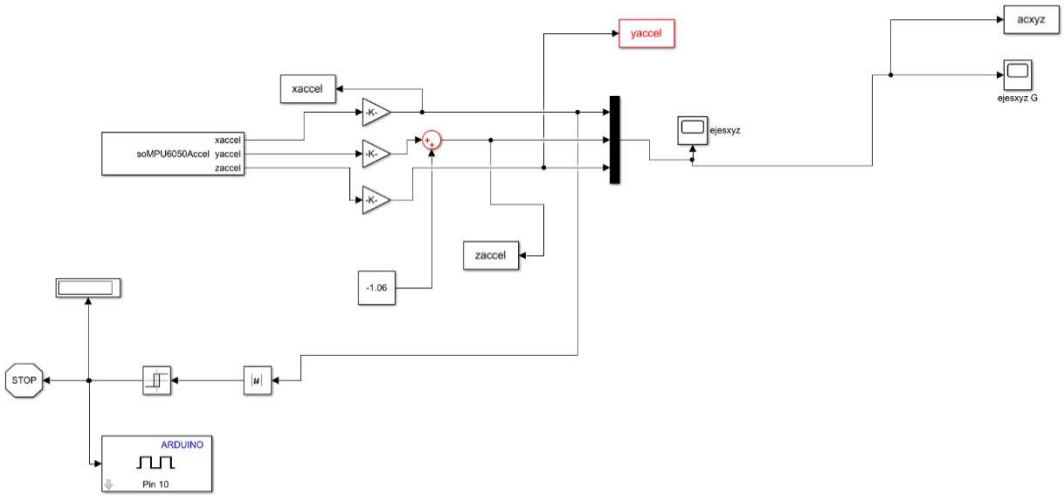


Figura 11. Interfaz en Simulink para recolección de muestras de aceleración por el sensor MPU6050.

Una vez terminada la interfaz, se puede inicializar la simulación directamente desde Simulink para comprobar su funcionamiento o desde el script que contiene las instrucciones para realizar todas las operaciones deseadas, y comprobar la ejecución correcta de la programación.

La estructura metálica (figura 12) fue ensamblada a partir de elementos estructurales fabricados de ángulo y solera comerciales. Material en el cual se ha considerado un E= 200 GPa y un módulo de Poisson de 0.26.

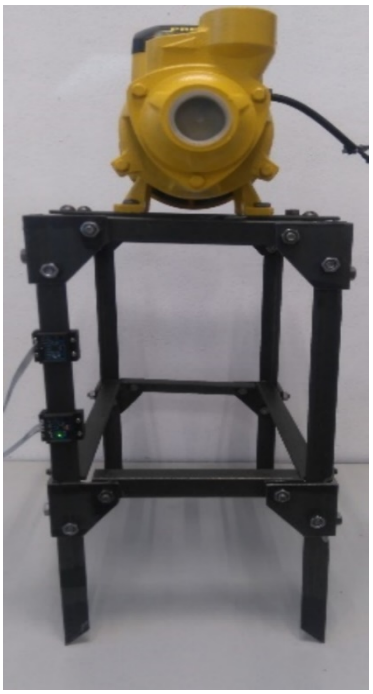


Figura 12. Prototipo construido para monitorear vibraciones.

4. Resultados

En la siguiente lista se muestran los parámetros para las ejecuciones de simulación en Simulink para el prototipo de estructura de 1 gdl. Los resultados se reportan en las figuras 13, 14 y 15 por eje de medición. Así como en la figura 16 se muestran la adquisición de las señales de los tres ejes en conjunto.

- Tiempo de simulación: 30 segundos.
- Inicio de simulación: Con el motor en operación**.
- Características de la carga
- RPM de la carga: 3450
- 120 V
- 60 Hz
- 5 A
- 3.7 Kg
- 0.5 HP
- La localización de la carga varía por cada prueba realizada.
- El eje z se normalizó a cero para las simulaciones, si se desea conocer las magnitudes sin normalizar, se debe desconectar el bloque “Constant” para realizar las ejecuciones de la simulación sin normalizar z a cero, o en su defecto, sumar a los valores en los registros enviados al Workspace de MATLAB el valor establecido en la constante (1.06).

Para propósitos del reporte de pruebas, no se mostrarán todas las gráficas en el reporte por cuestión de espacio y resolución de imágenes, sin embargo, se especifican los resultados de las simulaciones realizadas en la tabla.

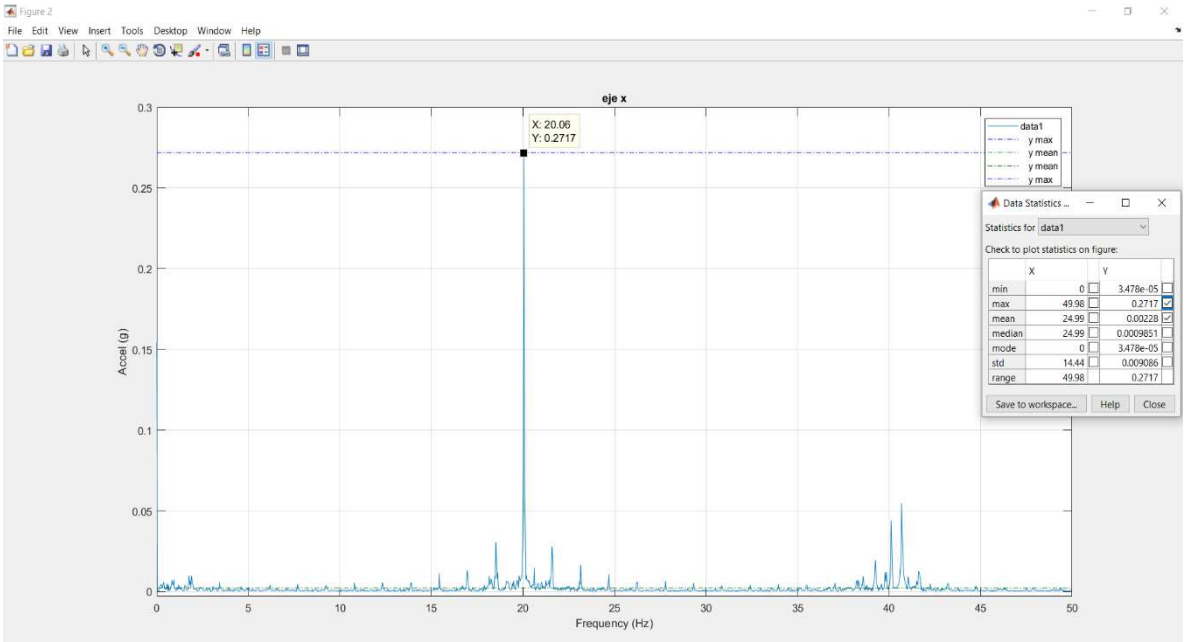


Figura 13. Espectro obtenido por la transformada rápida de Fourier (FFT) de acuerdo a las mediciones realizadas por el sensor MPU-6050 en el eje x.

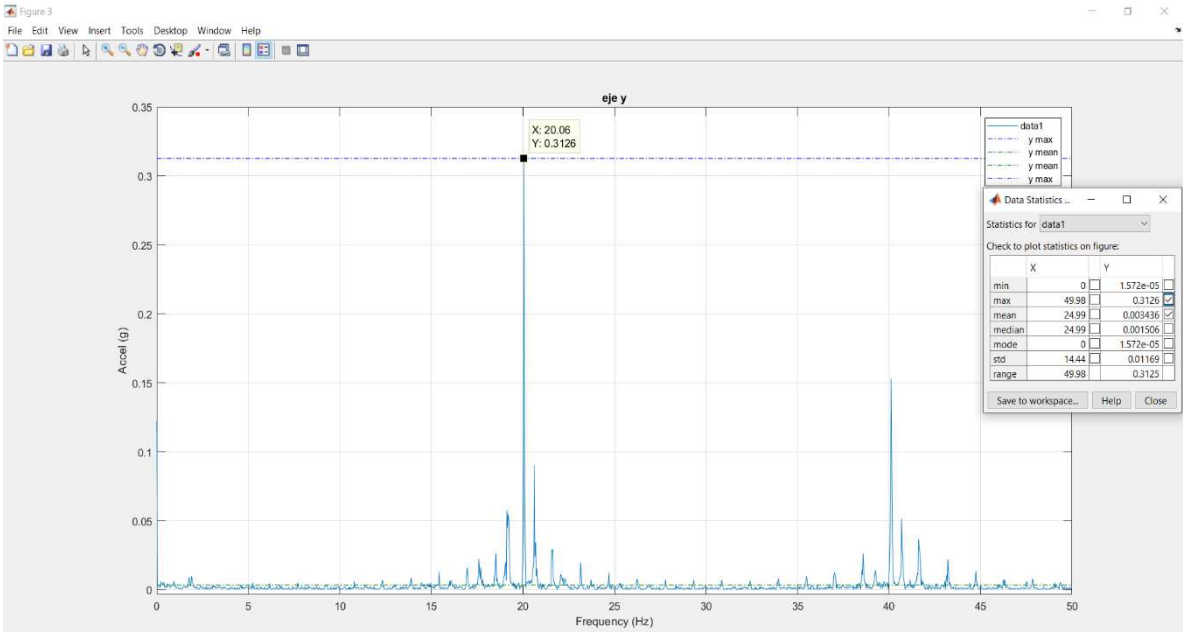


Figura 14. Espectro obtenido por la transformada rápida de Fourier (FFT) de acuerdo a las mediciones realizadas por el sensor MPU-6050 en el eje y.

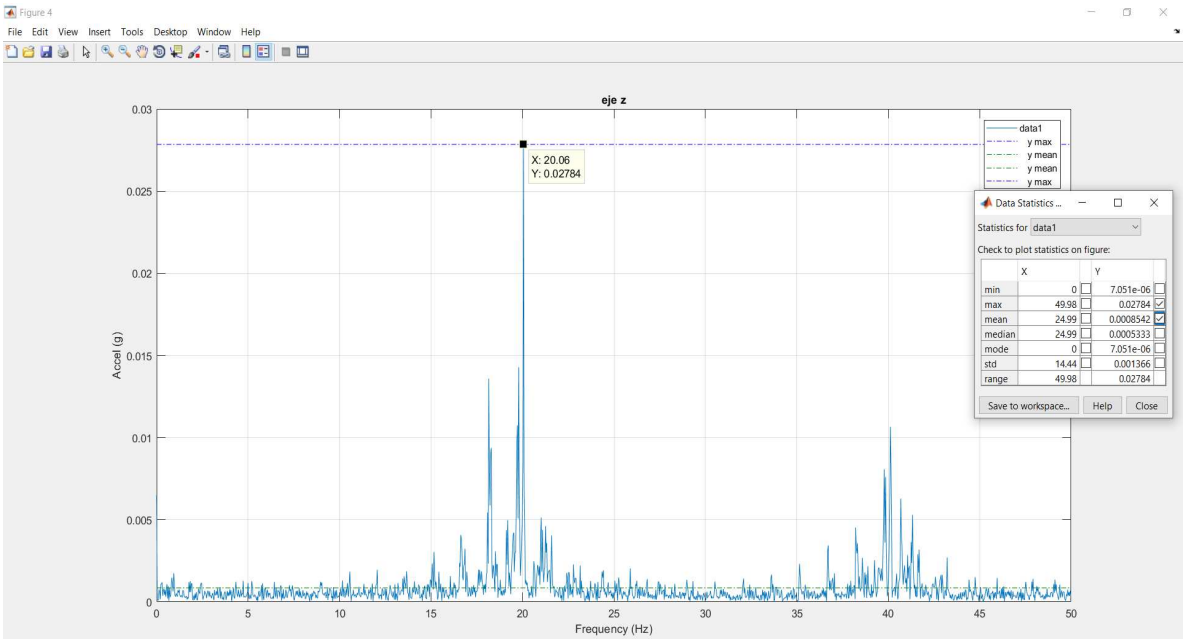


Figura 15. Espectro obtenido por la transformada rápida de Fourier (FFT) de acuerdo a las mediciones realizadas por el sensor MPU-6050 en el eje z.

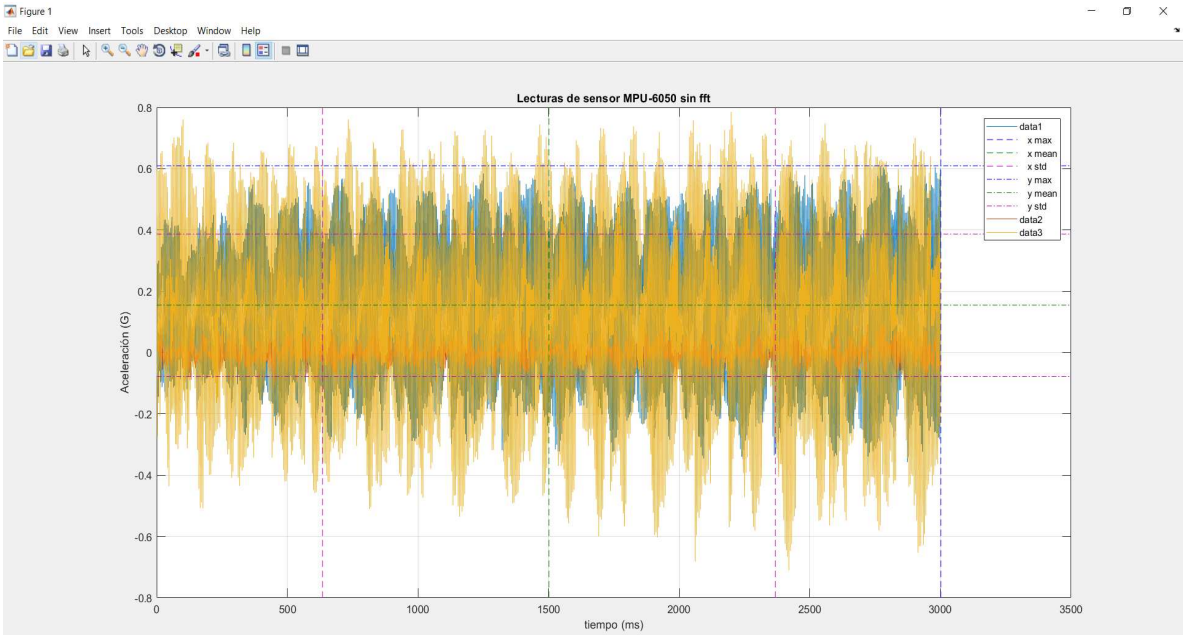


Figura 16. Acelerograma con las mediciones realizadas por el sensor MPU-6050 en los ejes x, y y z, sin implementación de transformada rápida de Fourier (FFT).

Tabla 1. Síntesis de los datos obtenidos por medición directa sin apriete en las conexiones (baja rigidez).

Localización de la carga estructura	Torque conexiones estructura	Valor promedio amplitud eje x (G)	Valor promedio amplitud eje y (G)	Valor promedio amplitud eje z (G)	Valor máximo amplitud eje x		Valor máximo amplitud eje y		Valor máximo amplitud eje z	
					G	Hz	G	Hz	G	Hz
Frontal-izquierda	0	0.002765	0.001571	0.000594	0.5442	19.99	0.07055	40.72	0.04975	19.99
Frontal-derecha	0	0.002837	0.001274	0.0006024	0.4759	19.99	0.04448	40.75	0.04229	19.99
Frontal-centro	0	0.004086	0.001105	.0007738	0.04115	20.07	0.0333	40.75	0.05125	20.07
Centro	0	0.002433	0.001753	0.001349	0.363	19.96	0.06861	40.75	0.0635	40.09
Posterior-izquierda	0	0.00228	0.003436	0.0008542	0.2717	20.06	0.3126	20.06	0.02784	20.06
Posterior-derecha	0	0.001262	0.00117	0.0004689	0.07643	20.03	0.037	20.03	0.006053	38.89
Posterior-centro	0	0.001572	0.001315	0.001086	0.1533	20.03	0.1042	20.03	0.02834	0.6664

Los valores de la tabla 1 se muestran en las figuras 17, 18, 19 y 20, para observar de manera precisa la variación de las amplitudes en función de la posición de la estructura, tanto en conjunto como individual, por eje. Asimismo, las figuras 21, 22, 23, y 24 muestran los resultados de la tabla 2.

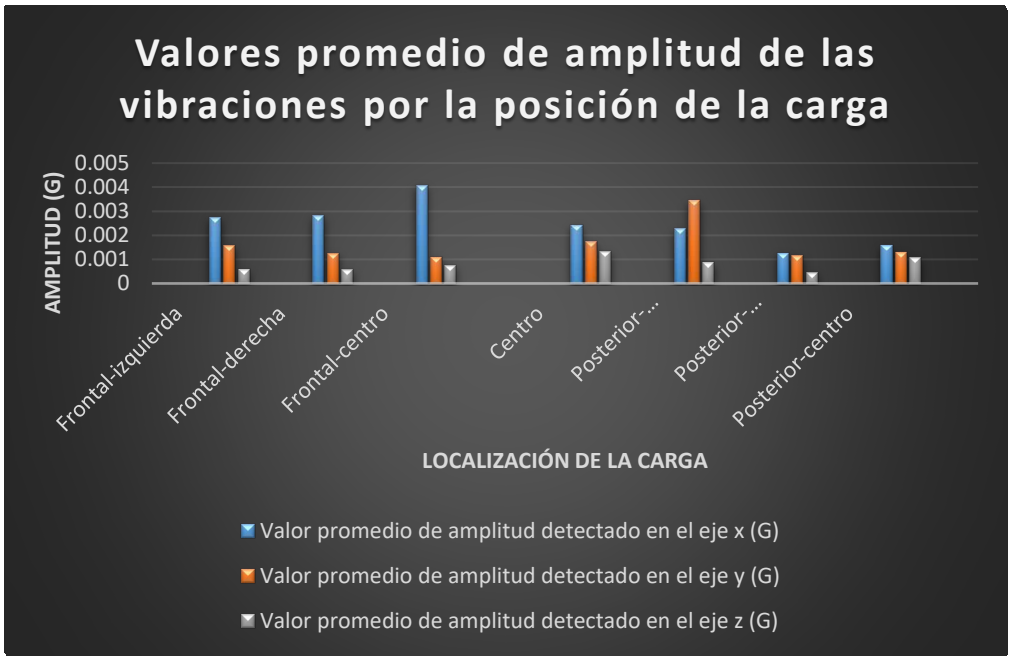


Figura 17. Valores promedio de amplitud de las vibraciones por la posición de la carga, sin precarga.

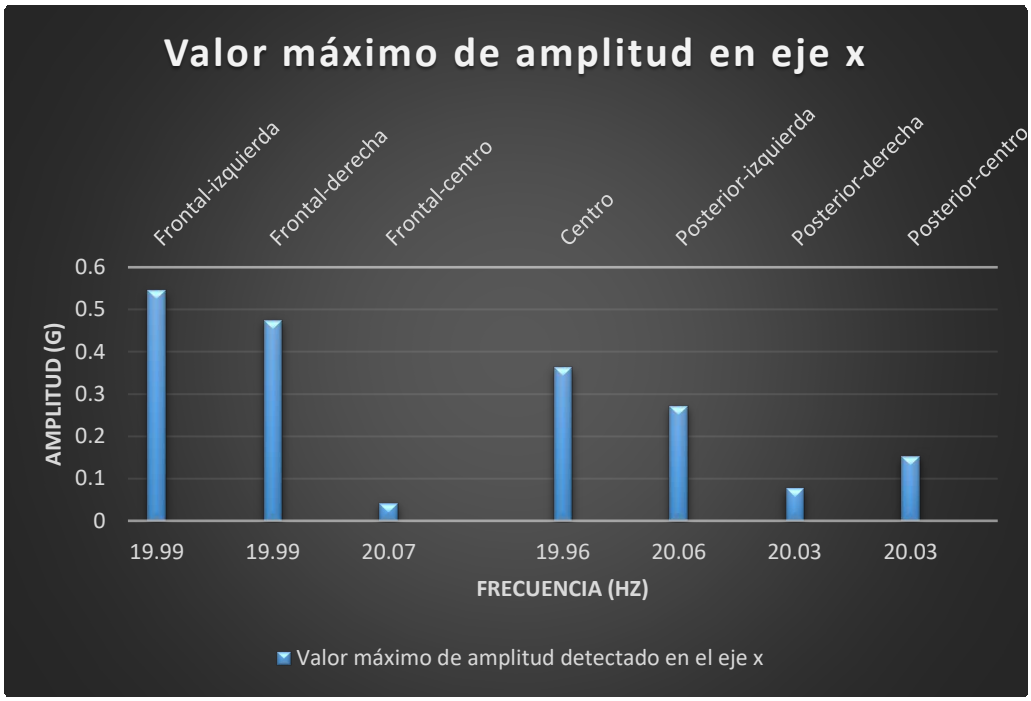


Figura 18. Máxima amplitud en el eje X, sin precarga.

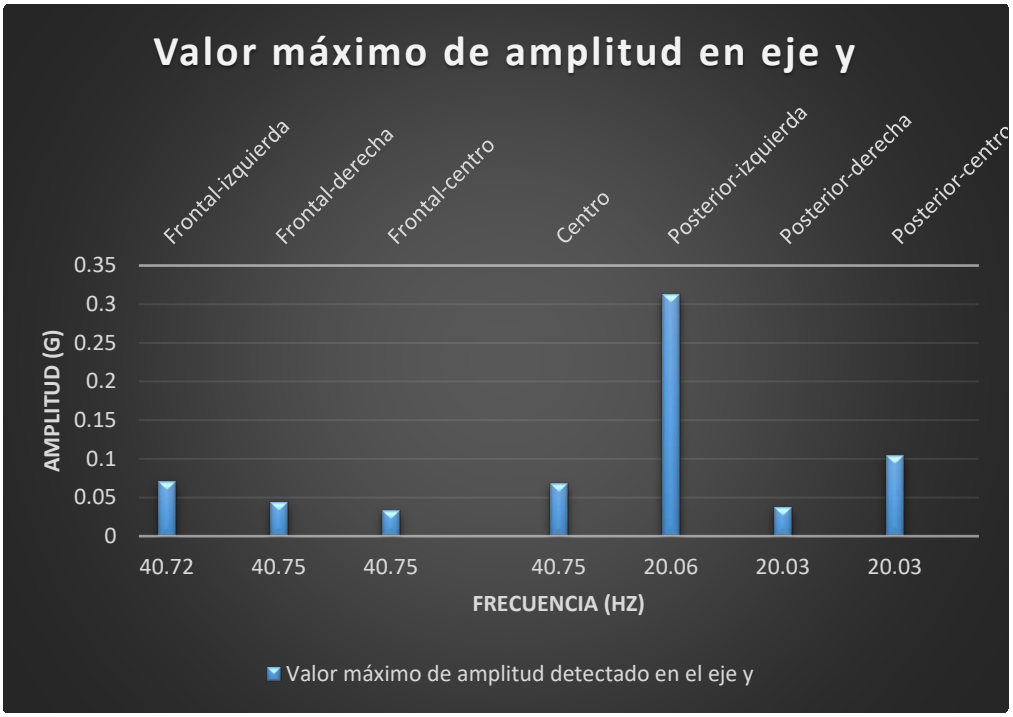


Figura 19. Máxima amplitud en el eje Y, sin precarga.

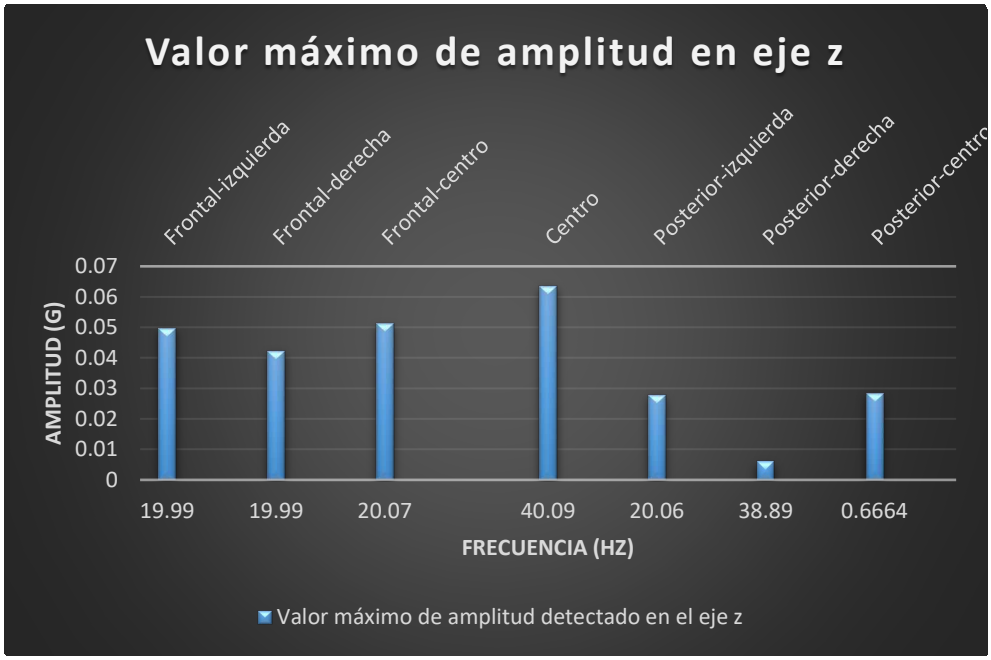


Figura 20. Máxima amplitud en el eje Z, sin precarga.

Se ha identificado que los valores de aceleración detectados por el sensor MPU-6050 varían de acuerdo con la posición que tenga la carga en la estructura, así como del torque que tengan las conexiones.

Tabla 2. Síntesis de los datos obtenidos por medición directa con apriete en las conexiones (alta rigidez).

Localización de la carga en la estructura	Torque conexiones de la estructura kg-cm	Valor promedio amplitud eje x (G)	Valor promedio amplitud eje y (G)	Valor promedio amplitud eje z (G)	Valor máximo de amplitud eje x		Valor máximo de amplitud eje y		Valor máximo de amplitud eje z	
					G	Hz	G	Hz	G	Hz
Frontal-Izquierda	23.03	0.005578	0.003749	0.001974	0.291	40.7	0.1446	20.07	0.04496	20.07
Frontal-derecha	23.03	0.002769	0.001676	0.001154	0.2258	20.03	0.1707	40.69	0.02582	40.09
Frontal-centro	23.03	0.004675	0.002549	0.00235	0.4907	20.03	0.09626	20.03	0.05102	20.03
Centro	23.03	0.00412	0.00335	0.001497	0.3116	19.99	0.1647	20.73	0.06316	19.99
Posterior-derecha	23.03	0.004933	0.005282	0.006117	0.2754	19.93	0.3372	19.93	0.6496	19.96
Posterior-izquierda	23.03	0.002822	0.002487	0.002377	0.1255	40.72	0.1248	40.72	0.02552	20.66
Posterior-centro	23.03	0.004508	0.006862	0.008587	0.256	40.69	0.1546	19.89	0.08914	2.233

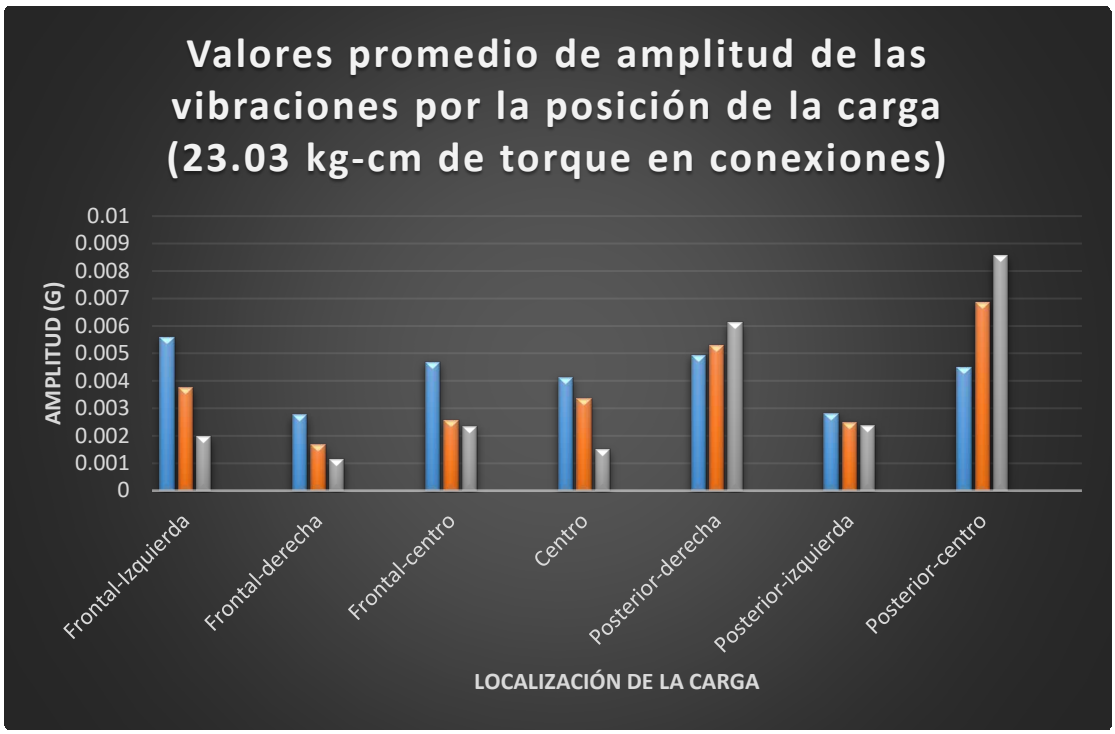


Figura 21. Valores promedio de amplitud de las vibraciones por la posición de la carga, con precarga.

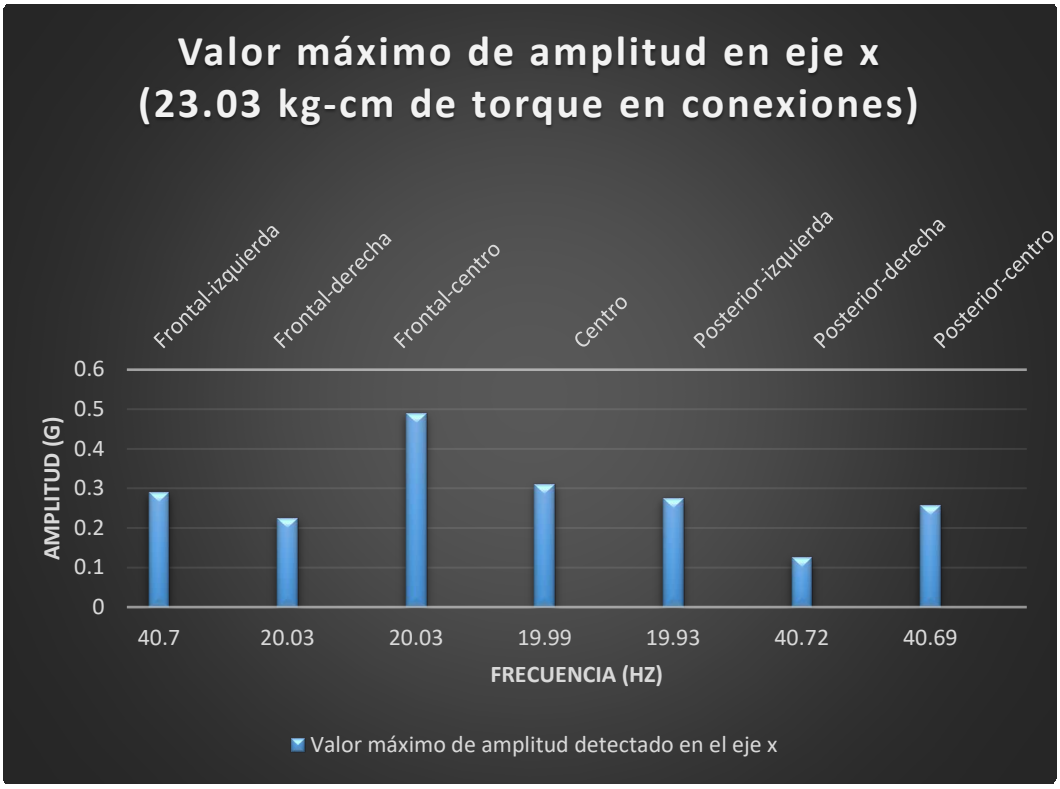


Figura 22. Máxima amplitud en el eje X, con precarga.

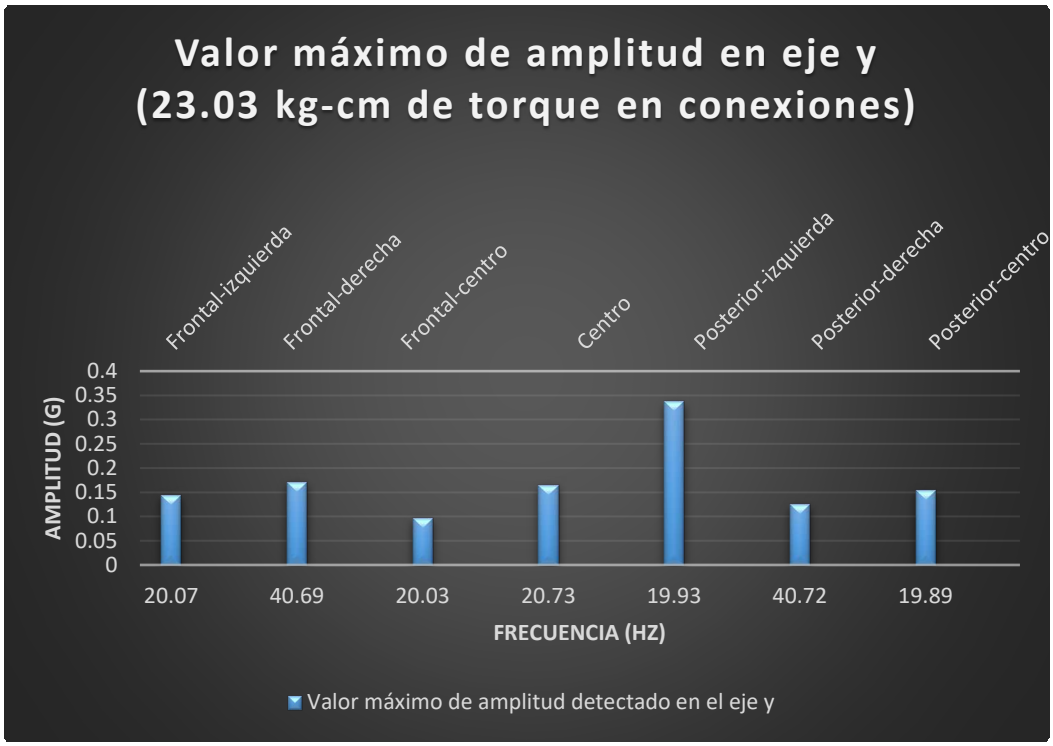


Figura 23. Máxima amplitud en el eje y, con precarga.

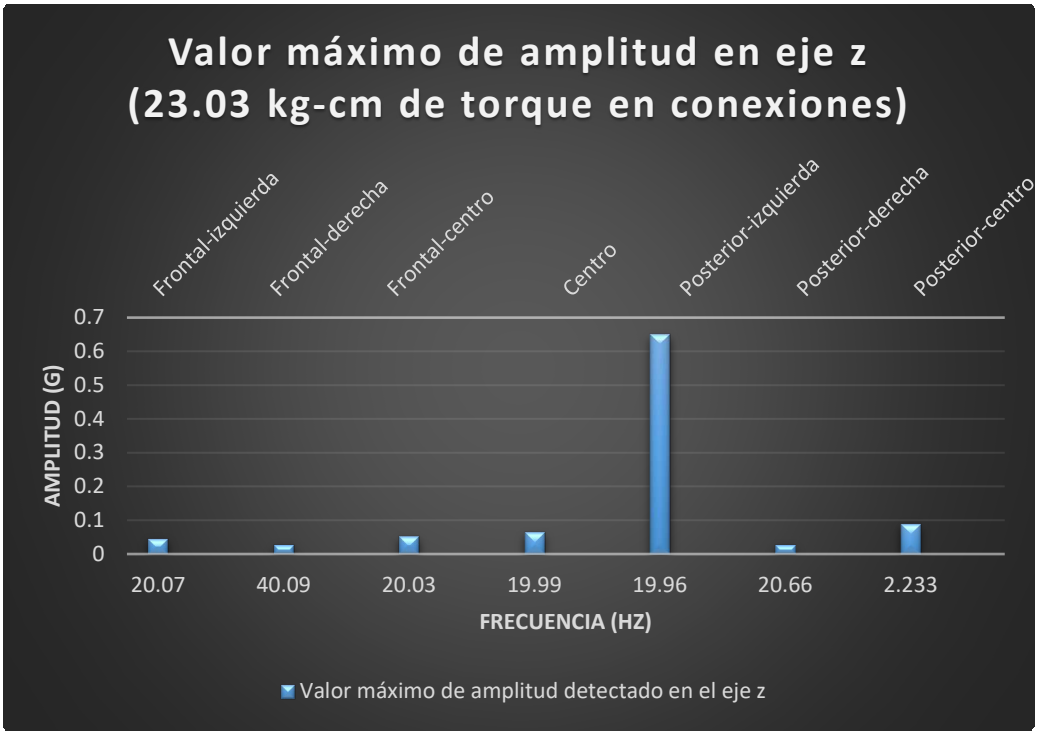


Figura 24. Máxima amplitud en el eje z, con precarga.

5. Conclusiones

Se concluye de manera anticipada que este dispositivo (figura 8 y 12), aporta un nivel de seguridad apropiado, el cual permite predecir daños significativos a los equipos y mantener la continuidad de los procesos de producción, mediante desarrollos tecnológicos a bajo costo [6]. Asimismo, es de suma relevancia el generar avances científico-tecnológicos referidos al campo del monitoreo de salud estructural, debido a su impacto en el sector de seguridad industrial y protección civil. Conocer el estado en que se encuentra la estructura y detectar condiciones de operación inseguras representa una ventaja para poder anticipar y/o minimizar las consecuencias de las fallas, reduciendo probabilidades de accidentes donde se vea involucrada la seguridad de las personas que se encuentran cerca del área donde se encuentra instalada la estructura y también los costos de mantenimiento o reparación a causa de la pérdida de rigidez al detectar problemas en sus primeras etapas, evitando que se magnifiquen.

La construcción de prototipos de bajo costo que permitan cumplir de manera satisfactoria con el monitoreo de salud estructural otorga la posibilidad de ser implementados de forma rápida en donde se requiera asegurar el estado de la rigidez en una estructura y en consecuencia, incrementar la confiabilidad en la operación normal y maximizar el tiempo de vida útil del sistema donde sea aplicado. Debido a la naturaleza de la operación de las estructuras, las vibraciones pueden cambiar de acuerdo con el propósito que deban cumplir dichas estructuras (turbinas eólicas, torres de distribución eléctrica, puentes, represas, edificios, túneles, almacenes de combustibles, ductos, integridad perimetral, etc.), las condiciones para el sistema de monitoreo pueden cambiar en las condiciones de amplitudes de aceleración y frecuencia permisibles. El sistema propuesto permite adecuar los parámetros necesarios para ajustarse a los requerimientos del usuario, estableciendo un rango de operación específico donde evitar falsos positivos de condiciones inseguras de operación.

Referencias

[1] B. R. Martínez, T. J. Pino, H. H. Terán y L. O. Arteaga, Vibraciones Mecánicas II, Sangolquí, Ecuador: Universidad de las Fuerzas Armadas ESPE, 2017.
[2] B. R. Martínez, T. J. Pino, H. H. Terán y L. O. Arteaga, Vibraciones Mecánicas I, Sangolquí, Ecuador: Universidad de las Fuerzas Armadas ESPE, 2017.
[3] E. J. Pope, Soluciones prácticas para el ingeniero mecánico, México D.F.: Mc. GrawHill, 2000.
[4] J. O. Ocampo, «SISTEMA DE BAJO COSTO PARA MONITOREO DE DISTURBIOS DE VIBRACIÓN EN MÁQUINAS ROTATORIAS,» Innovare, pp. 14-30, 2013.
[5] P. E. Mandado, A. J. Marcos, S. C. Fernández y Q. J. Armesto, Autómatas programables y sistemas de automatización, México D.F. Alfaomega, 2009.
[6] C. F. Bernal y N. D. Cortés, Simulación de un banco de pruebas para análisis de vibraciones, Bogotá: Universidad Distrital Francisco José de Caldas, 2016.

Diseño Colaborativo en Mecatrónica, Capítulo 15, pp. 203 - 215
ISBN: 978-607-9394-23-3, 2021

Sistema para la medición de nivel sin contacto basados en ultrasonido

Guillen Soto José María, Jiménez Oronia Francisco, Olalde Mendoza Omero Nicolas, Rodríguez Olivares Noe Amir

Centro de Ingeniería y Desarrollo Industrial
Gerencia de Robótica e Instrumentación
Av. Playa Pie de la Cuesta No. 702. Desarrollo San Pablo. C.P. 76125 Querétaro, Qro. México
jguillen@posgrado.cidesi.edu.mx
correspondencia: noe.rodriguez@cidesi.edu.mx

Resumen

En la industria del petróleo, la industria aeroespacial y en la industria química existen líquidos que por su composición (química) son almacenados en recipientes sellados, entonces es ahí que la medición de nivel se vuelve un factor importante, es por eso que en este artículo se presenta un método de medición de nivel de líquido por medio de ultrasonido, este método está basado en la impedancia que ejercen los materiales al paso de las ondas de sonido, el método de impedancia posee la característica de que es menos propenso a verse afectado por las impurezas que pudiese tener el contenedor. El método de impedancia es comparado con el método de velocidad del sonido, el cual basa su funcionamiento en el tiempo que tarda un eco en ser reflejado en la pared del contenedor, al existir un segundo medio (gas o líquido) este afectara en el vuelo de la onda de ultrasonido.

Palabras clave: amplitud, contenedor, líquido, medición, nivel, ultrasonido, sin contacto, velocidad del sonido.

1. Introducción

En industrias como la aeroespacial, petróleo y química existen en una gran mayoría líquidos que son almacenados dentro de recipientes sellados, estos líquidos son en gran medida mezclas volátiles, inflamables, explosivas o corrosivas [1]. Por tal motivo, actualmente se utilizan métodos que emplean el ultrasonido para determinar un nivel de líquido, esto es en gran medida gracias a que el ultrasonido puede emplearse de manera que no sea invasiva, esto quiere decir, que no se necesita realizar ninguna modificación al contenedor, ya que cualquier modificación a este puede repercutir en accidentes, fallas, costos extra, e incluso en el paro de un proceso de suma importancia para la industria. Entre los distintos métodos recientemente se han venido utilizando cuatro métodos: velocidad del sonido, impedancia acústica, penetración, ondas Lamb [2].

Velocidad del sonido: Se mide la diferencia del sonido entre diferentes materiales, en este caso las mediciones se ven afectadas por el cambio de la señal que se recibe y la velocidad del sonido en un material, es por eso que se puede realizar una medición de nivel [3].

Impedancia ultrasónica: En este tipo de métodos se mide la atenuación de entre un medio líquido y uno gaseoso, por ende, el nivel de líquido se mide comparando el tiempo de atenuación de los ecos [4].

Aunque existen otros métodos para la medición por medio de US (ultrasonido) como lo son métodos de penetración estos usualmente son de arquitectura cerrada, esto quiere decir, que no se

pueden realizar cambios al software y hardware, un ejemplo de ellos son los fabricados por la marca *Class Instruments*™, estos se basan generalmente en la penetración de las ondas de ultrasonido en el medio líquido del recipiente, sin embargo, hay que tomar en cuenta que las impurezas del contenedor pueden afectar en la medición de nivel (Figura 1). La precisión de este tipo de medición puede variar alrededor de un 10% con respecto al nivel real [5].

En los métodos que utilizan la propagación de ondas Lamb basan su medición en el cambio de la onda cuando esta supera la pared del contenedor, sin embargo, al igual que el método de penetración, se deben evitar imperfecciones en el contenedor, ya que esto afecta en la medición [6].

Entonces al tomar en cuenta los cuatro métodos presentados anteriormente y con base al contenedor de prueba se puede determinar que los dos métodos que se utilizaran en este articulo para obtener una medición de ultrasonido con mayor confiabilidad son los basados en velocidad del sonido e impedancia acústica.

Los métodos de penetración y ondas Lamb no son puestos a prueba en este artículo ya que con el primero no se pueden realizar ajustes más allá de los que proporciona el fabricante y en el caso de análisis con ondas Lamb debido a la naturaleza del recipiente (imperfecciones) este método se ve sumamente afectado.

Figura 1. Imperfecciones en un contenedor.

2. Marco teórico

Para poder entender un poco más los métodos a analizar es necesario que se aborden conceptos que hacen referencia a los mismos, como son la velocidad del sonido, impedancia acústica y círculo de energía.

2.1 Velocidad del sonido

Cuando se habla de ultrasonido se refiere a frecuencias que están por encima del alcance del oído humano, es decir por encima de los 18 kHz. Las ondas mecánicas de vibración viajan en la forma de una onda, similar a la que viaja la luz, sin embargo, a diferencia de las ondas de luz, estas requieren de un medio elástico tal como un líquido o un sólido [7]. Entonces, la relación entre la velocidad del sonido en el material (*c*), la frecuencia (*f*), la longitud de la onda (λ) y el período (T), entonces:

$$\lambda = c / f \tag{1}$$

$$\lambda = cT \tag{2}$$

Donde:

λ = longitud de la onda (m).
c = velocidad del sonido en el material. (m/s).
f = frecuencia (Hz).
T = período (s).

2.2 Distancia entre ecos.

Para hacer una medición basada en la velocidad del sonido es necesario determinar la distancia entre ecos, para ello se mide la distancia entre el primer eco recibido y el segundo eco (Figura 2) [8].

Entonces, el principal interés es la distancia que hay entre cada eco, esta viene determinada por la velocidad del sonido en el material y el tiempo de vuelo entre los ecos:

$$E = \frac{ct}{2} \tag{3}$$

Donde:

E = Distancia entre ecos (m).
c = velocidad del sonido en el material. (m/s).
t = tiempo de vuelo (s).

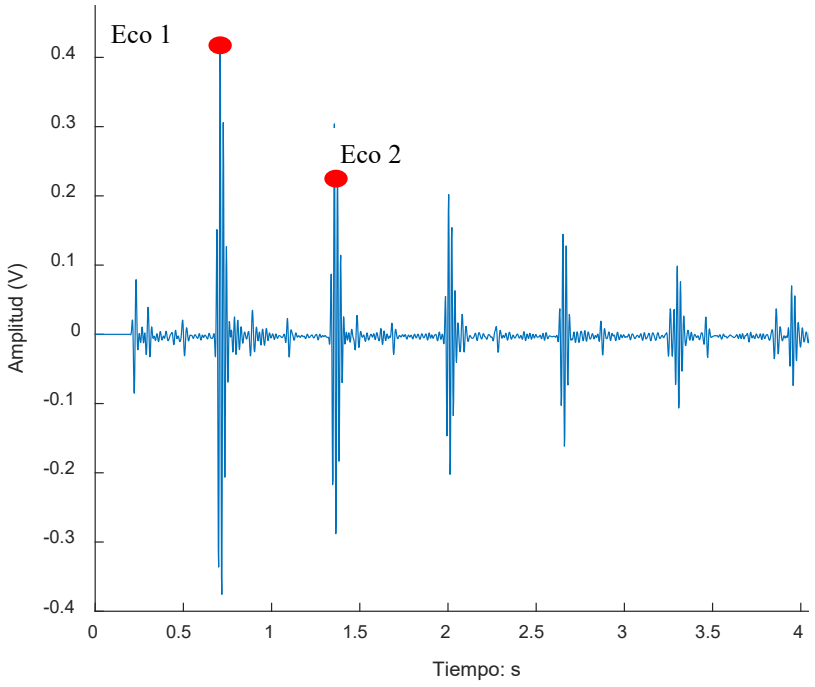


Figura 2. Distancia entre ecos.

2.3 Impedancia acústica.

La impedancia acústica es una característica del medio que atraviesa el ultrasonido, esta relaciona la velocidad que la partícula adquiere en el momento de su vibración y la presión a la que está sometida. La impedancia da idea de la facilidad que un determinado medio ofrece al paso de ultrasonidos a través de este. Se conoce habitualmente con la letra *Z* y es igual al producto de la densidad del medio por la velocidad de transmisión del ultrasonido en ese medio. La reflexión se produce al intentar pasar el ultrasonido de un medio a otro con distinto *Z* [9]. Existen valores para *Z* dependiendo el material, en este artículo se toman en cuenta los mostrados en la Tabla 1. Generalmente cuando hay un cambio de líquido a gas del has de sonido la impedancia de este se modifica siendo en el medio líquido mucho mayor a la impedancia en el medio gaseoso.

Tabla 1. Características del sonido en distintos materiales.

Material	Velocidad del sonido (m/s)	Impedancia Kg/m^2
Acero	5900	46.4000
Aluminio	6420	17.3300
Aire	330	0.0004
Agua	1480	1.4800

2.4 Potencia acústica

El haz de ultrasonido transporta una determinada cantidad de energía producida por el transductor por unidad de tiempo, es lo que se conoce como potencia. La unidad de potencia es el Watt (W). Dividiendo la potencia por la superficie del haz de contacto, obtenemos la intensidad (W/cm^2), que es uno de los parámetros más importantes que hay que tener en cuenta en los tratamientos con ultrasonidos [9].

2.5 Círculo de energía.

Según el modelo de Schmerr [9], el campo sonoro de un transductor redondo en un medio sólido tiene dos partes diferentes (ver Figura 3), a estas partes se les llama campo cercano (zona de Fresnel) y campo lejano (zona de Fraunhofer), en el campo cercano, el haz de energía conserva una forma cilíndrica, mientras que en el campo lejano el haz se propaga con un cierto ángulo de divergencia [9].

Donde en la Figura 4, ***r*** es el radio del transductor, ***d*** es el diámetro del círculo de energía y **ɵ** es el ángulo de divergencia. El círculo que se forma ayuda a saber cuándo existe un cambio entre un medio líquido y uno gaseoso, es por eso que, al tener una característica como la velocidad del sonido o la impedancia del mismo, es posible detectar un cambio y así determinar el nivel de líquido del contenedor [10].

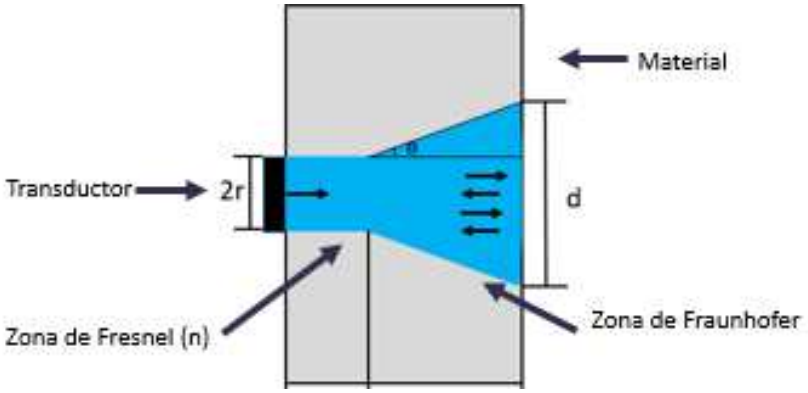


Figura 3. Características del círculo de energía campo cercano (zona de Fresnel) y campo lejano (zona de Fraunhofer).

3. Desarrollo

Para poder implementar los dos métodos para la medición de nivel por ultrasonido, primero se tiene que adquirir una señal de ultrasonido, después esta es procesada para posteriormente trabajar con sus características.

3.1 Adquisición y procesamiento

Se tiene un transductor de la marca GE (General Electric) de 5 MHz de haz recto, el cual tiene un radio de 5 mm, la señal es adquirida por medio de un sistema de adquisición de ultrasonido que tiene como procesadores de la señal un FPGA y un microcontrolador. El microcontrolador se encarga de generar un voltaje analógico de 0v-3.3v, este voltaje es convertido de un rango de 0v-3.3v a un rango de 0v-200v de corriente continua, este proceso es controlado por el microcontrolador ya que el FPGA no cuenta con un módulo de salidas analógicas. En cuanto al FPGA este es seleccionado ya que para obtener una mejor respuesta se debe muestrear a 50 MHz, con esto se asegura que la señal obtenida tiene una mejor resolución, además, con el uso del FPGA se pueden realizar procedimientos de manera simultánea, es decir, en forma paralela.

La tarjeta de adquisición envía la señal sin procesar a la computadora, esta tarjeta de adquisición es tomada de un trabajo anterior, el cual por motivos de confidencialidad no se puede mostrar. La señal es enviada por medio del protocolo RS232 a la computadora, esta indica los parámetros que se requieren, en tanto a voltaje de la excitación del transductor, así como la amplificación de la señal recibida por el transductor. Una vez en la computadora la señal de ultrasonido, por medio de la transformada de Fourier se analizan sus componentes en frecuencia para posteriormente aplicar un filtro FIR de 24 coeficientes y eliminar las señales de ruido que fueron capturadas por el transductor, los 24 coeficientes son seleccionados ya que al crecer el orden del filtro este se vuelve más complejo computacionalmente y esto repercute en la memoria del FPGA si se llegase a implementar en el mismo (ver Figura 4).

El tiempo de muestreo es de 50 MHz y la resolución del ADC es de 10 bits, pero para evitar que el tiempo de recepción se duplique solo se utilizan los 8 bits más significativos, eliminando así los menos significativos que por lo regular son los que contienen más ruido. Teniendo una señal filtrada se puede proceder a aplicar los métodos que se mencionan en la introducción (impedancia acústica y velocidad del sonido).

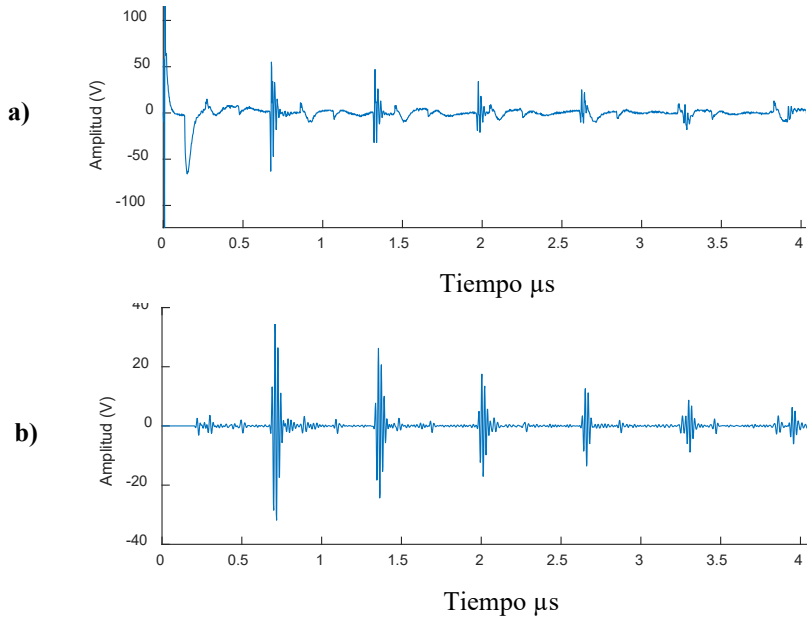


Figura 4. a) señal obtenida por tarjeta de ultrasonido sin filtrar. b) señal filtrada por medio de filtro FIR de orden 30.

3.2 Método de la velocidad del sonido

Para calcular la distancia entre ecos por medio de la velocidad del sonido se debe recordar la ecuación 3, en esta ecuación sustituimos los valores por los que se tienen en recipiente, este recipiente que tiene como espesor de pared L = 6 mm y cuyo material es acero con una velocidad del sonido de 5900 m/s.

Para aplicar este método se determinan 3 estados críticos, cuando se mide arriba del nivel de líquido, cuando se mide debajo del nivel de líquido y cuando se mide entre los dos estados anteriores [11]. Entonces es necesario calcular la distancia entre ecos en esos estados críticos (Figura 6).

En la Figura 6, se muestra un esquema de los estados críticos y lo que se requiere para llegar a este resultado, se muestra que al haber presencia de un líquido el tiempo que tarda la onda en converger es menor que al haber ausencia de un líquido. Este fenómeno se debe a que la velocidad del sonido es diferente entre cada medio, sabiendo eso es posible que se pueda obtener una medición de nivel.

Con base en la Figura 5 se puede calcular la distancia entre ecos, este mismo procedimiento se hace en los 3 estados críticos, obteniendo los resultados presentados en la Tabla 2.

Se puede deducir, que al haber presencia de un líquido se ve afectada la distancia entre el primer eco y el segundo eco, en comparación a cuando no hay presencia de líquido, entonces para llegar a un resultado de medición correcto, es necesario pasar de un estado h_1 a un estado h_2 (Figura 6), al realizar este proceso se puede calcular el nivel de líquido por medio de:

$$\frac{h_1 + h_2}{2} = h \tag{4}$$

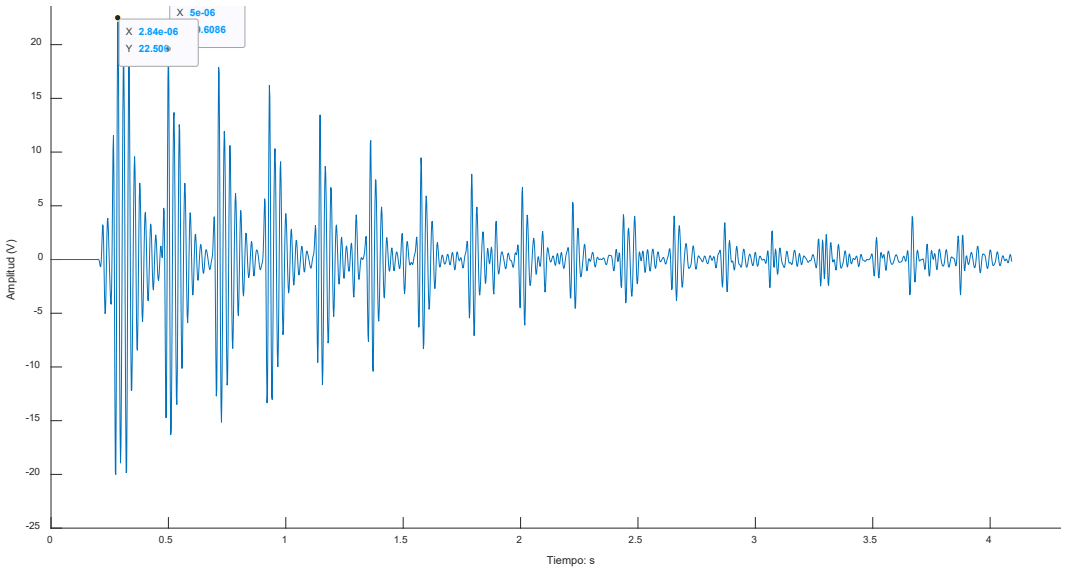


Figura 5. Señal obtenida de tarjeta de ultrasonido.

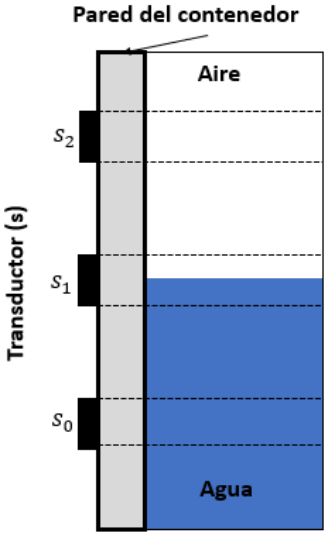


Figura 6. Estados críticos. S_0 medición debajo del nivel de líquido, S_1 medición entre el nivel de líquido y S_2 medición arriba de nivel de líquido.

El tercer estado crítico en se puede calcular por medio de la ecuación 4, pero además, se realiza una corrida en la sabiendo el punto donde se encuentra el nivel de líquido determinaremos cual es el tiempo que tarda el eco en ser reflejado (Tabla 2).

Tabla 2. Distancia calculada entre los dos primeros ecos (mm).

Estado	Distancia entre ecos (mm)
Arriba del agua	6.36
Entre el agua	6.36
Abajo del agua	6.11

3.3 Método de la impedancia acústica

De igual manera que en el método de velocidad del sonido lo que se buscan son 3 estados críticos (Figura 6), pero la diferencia es que en este método se mide la impedancia acústica de una señal de ultrasonido, los parámetros son los mismos para poder hacer una correcta comparación de los métodos. En este método lo que se pretende comprobar es que puede existir un tercer estado crítico si se miden los estados por encima y por debajo del agua.

Lo que se pretende demostrar con este método es que la potencia es diferente en cada estado crítico, es decir que no se repite, esto se debe al coeficiente de atenuación, eso ayudará a obtener un valor estimado en la medición de nivel de líquido. Entonces, partiendo de las impedancias mostradas en la Tabla 1, se puede decir que Z_l = 1.48 kg/m^2, Z_g=0.0004 kg/m^2 y Z_m=46.4 kg/m^2 donde Z es el valor de impedancia del medio líquido (agua), gas (aire) y el medio (acero) respectivamente.

Entonces, para calcular el coeficiente de impedancia se debe de tomar en cuenta la ecuación (5) donde k_t es el coeficiente de impedancia, Z_1 y Z_2 es la impedancia del material, donde el primer coeficiente hace referencia al acero y el segundo coeficiente hace referencia al medio líquido o gaseoso [12].

$$k_t = \frac{Z_1 Z_2}{(Z_1 + Z_2)^2} \tag{5}$$

Ahora bien, con la muestra obtenida con los mismos parámetros del método de velocidad del sonido se puede corroborar que efectivamente los coeficientes de atenuación son diferentes, esto quiere decir que al momento de realizar una medición es poco probable que se empalmen los resultados. Además, se comprueba que generalmente la impedancia en el medio líquido es mucho mayor la impedancia en el medio gaseoso.

Tabla 3. Características del sonido en distintos materiales.

Estado	Coeficiente de atenuación
Arriba del agua	3.4482e-05
Abajo del agua	0.1198

Como existe diferencia entre estos dos estados se puede calcular la potencia del eco y con base a ella se puede estimar donde se encuentra el nivel de líquido, la potencia se calcula por medio de la ecuación (6) [13].

$$P_t = \sum_{i=1}^{n} I_{wi} \pi r^2 = \frac{(P_0 R_{ml}^{i} R_{ma}^{i-1} e^{-2i\alpha L})^2 \pi r^2}{2 Z_m} \tag{6}$$

Donde:
P_t = Potencia total.
P_0 =Potencia inicial.
$R_{ml,ma}$ =Coeficiente de reflexión.
r = Radio del transductor.
Z_m = Impedancia del medio.
L = Espesor del Contenedor.

4. Resultados

Para comprobar cuál de los dos métodos ofrece un mejor resultado, se toma en cuenta un contenedor de acero con L= 6mm, un transductor de r = 5 mm, los valores de impedancia y velocidad del sonido son tomados de la Tabla 1. El banco de pruebas se muestra en la Figura 7 donde se observa que fue marcada con una regleta, esto con el afán de corroborar la medición, el nivel de agua está marcado con una línea punteada. Si bien el contenedor donde se realizan las pruebas es cuadrado esto no afecta a la medición ya que se ha comprobado que con un transductor de haz recto se puede tener una desviación de 8° [14], en un contenedor es más que suficiente esto por las dimensiones que son usados habitualmente en la industria.

Se colocó el transductor en menos 5 cm de la línea punteada, y se fue recorriendo de forma ascendente hasta llegar a 5 cm arriba de donde está la línea punteada, el procedimiento se ilustra en la Figura 10. Se tomó una muestra cada 1 cm, estas muestras fueron almacenadas en la computadora, por medio de *Matlab* se filtran, para poder realizar el procedimiento de cada método.

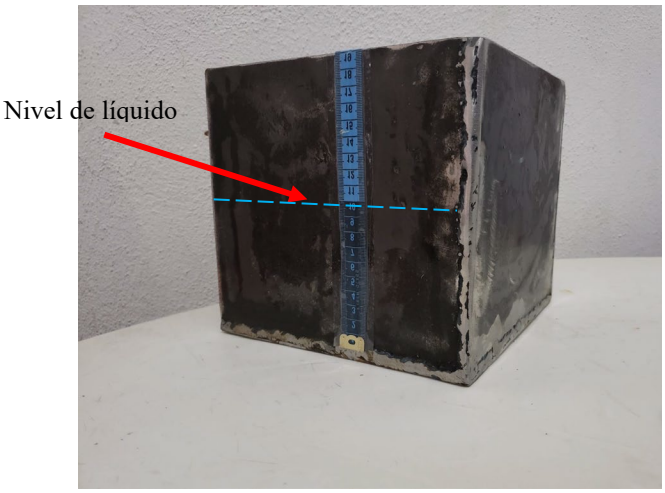


Figura 7. Diseño experimental. La regleta indica el nivel de líquido de manera visual.

4.1 Impedancia acústica

Para poder comprobar que se puede obtener un nivel de líquido se toman dos puntos (2 cm arriba del nivel real y 2 cm por debajo del nivel real), después, estas dos muestras son comparadas para determinar si es posible obtener un determinado nivel de líquido.

En la Figura 8 se puede observar que la potencia por debajo del nivel de líquido es mucho mayor que la potencia por encima del nivel de líquido, esto se debe a la fuerza que ejerce el aire al paso de las ondas de sonido, entonces, se puede decir que es posible medir el nivel de líquido basándose en la potencia del eco.

Una vez sabiendo que se puede realizar una medición de nivel de líquido como se muetra en la Figura 8, entonces, el siguiente paso es realizar una medición cada 1cm de forma ascendente desde el fondo del contenedo. Se determino un umbral de 0.01w esto para determinar donde se encontrban los estados críticos (por debajo del nivel, por encima del nivel y entre el nivel), se encontro que la variación en cuanto a potencia no era significativa hasta llegar a 8 cm, a partir de este punto la potencia comenzaba a variar hasta llegar 12 cm donde este dejaba de variar significativamente, es ahí donde se determina que h_1 =8 cm y h_2 =12 cm (ver Figura 9).

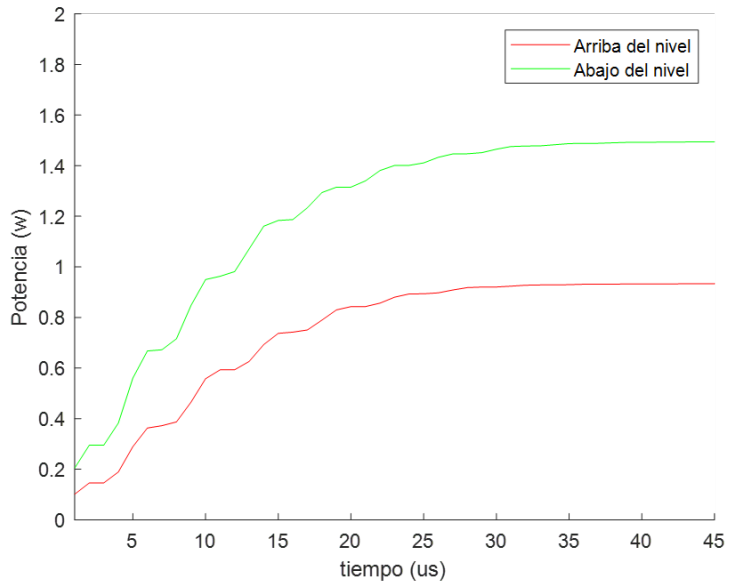


Figura 8. Potencia generada por el eco, en color verde la potencia medida por debajo del nivel de líquido y en color rojo la potencia medida por encima del nivel de líquido.

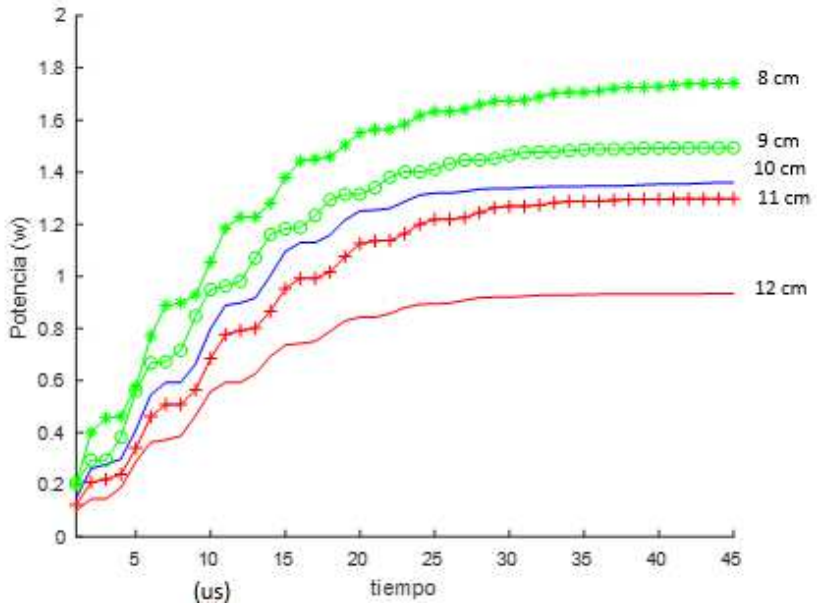


Figura 9. Comparación de muestras. Las líneas de color verde son pruebas realizadas con por debajo del nivel de líquido, las líneas rojas representan pruebas realizadas por debajo del nivel de líquido.

4.2 Velocidad del sonido

De igual manera que con el método de impedancia acústica, es necesario saber si se puede realizar una medición de nivel primero comprobando si hay diferencias entre la información proporcionada por el eco en los dos estados críticos, en la Figura 10 se observa que existe diferencia ente debajo del nivel de líquido (línea azul) y encima del nivel de líquido (línea roja).

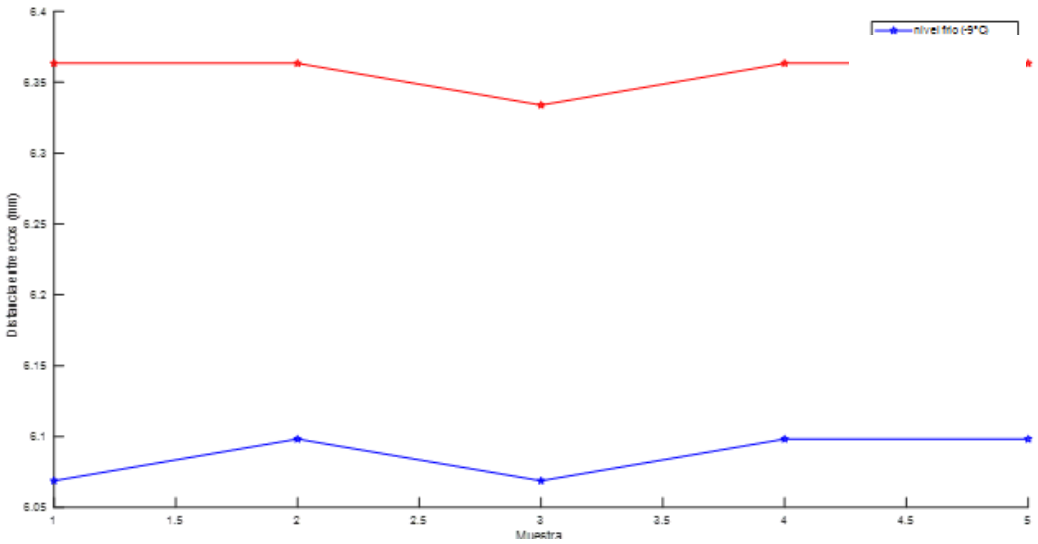


Figura 10. Diferencia medida por encima del nivel de líquido (línea roja) y por debajo del nivel de líquido (línea azul).

Para ser consistentes en el desarrollo de las pruebas ahora se realizan las mediciones de manera ascendente a lo largo del contenedor, estas mediciones se realizan cada 1 cm como en el método de impedancia acústica, en este método (Figura 11) se puede observar que h_1 = 8 y h_2=10.

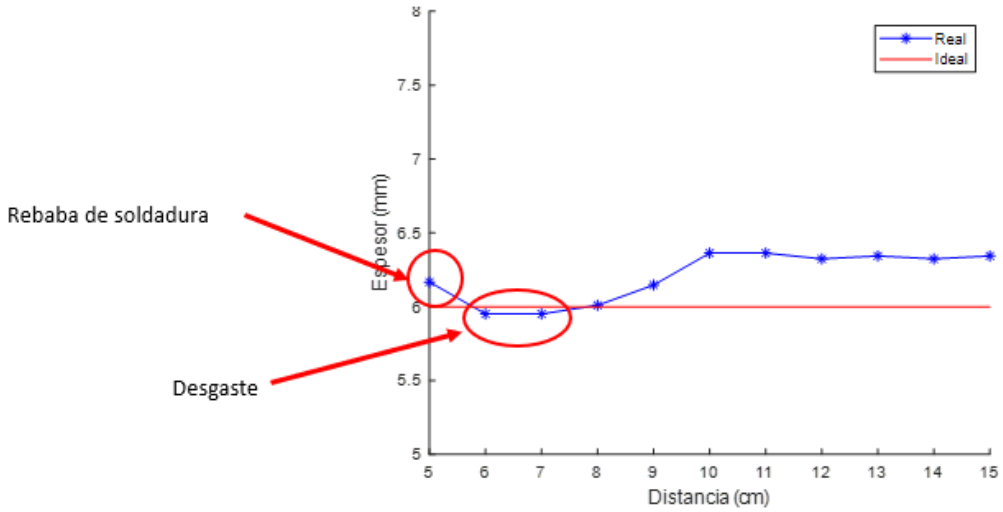


Figura 11. Posibles afectaciones en el contenedor que pudiesen afectar en la medición de nivel por medio del método de velocidad del sonido.

Por otra parte, se puede observar que este método es susceptible a desgaste que pudiese tener el contenedor, esto se debe a que este tipo de métodos también es utilizado para medir espesores de tuberías, al pasar la onda de sonido de un medio (acero) a otro medio (agua) la velocidad del sonido cambia drásticamente y por consecuencia la onda de ultrasonido converge más rápido a cero.

El correcto funcionamiento del contenedor se convierte en un asunto crítico al utilizar este método, ya que al tener una soldadura con el método de velocidad del sonido pudiese marcarse una falsa medición, entonces el error crecería sustancialmente, es por eso que al utilizarse este método es

recomendable realizar varias corridas (varias mediciones) bajo parámetros controlados, entonces, este es otro factor en contra de este método.

4.3 Comparación de los métodos

Si se toma en cuenta h_1 y h_2 en los métodos de impedancia acústica y velocidad del sonido se puede realizar una estimación de un nivel de líquido basándose en la ecuación 4, por lo tanto, se puede decir que en el método de impedancia acústica el nivel de líquido se encuentra en 10 cm y en el método de velocidad del sonido se encuentra en 9 cm.

Otro de los aspectos importantes es considerar las variables que no se pueden controlar como es el desgaste o las rebabas de soldadura que pudiese contener el contenedor ya que estas variables no se pueden controlar, entonces se vuelven un impedimento en el método de velocidad del sonido.

En la Tabla 4 se puede observar los datos obtenidos de la experimentación y un nivel de líquido estimado, esto basándose en el cambio de la onda de sonido y con ayuda de una cinta métrica, además, se considera que la temperatura afecta a ambos métodos, es por eso que se realizan las pruebas a 23°C.

Tabla 4. Resultados de la comparación de los dos métodos (impedancia acústica y velocidad del sonido).

Método	Impedancia acústica	Velocidad del sonido
h_1	8 cm	8 cm
h_2	12 cm	10 cm
h (Calculado)	10 cm	9 cm
h (Ideal)	10 cm	10 cm
Error	0 cm	1 cm
Afectaciones	-	Soldaduras, desgaste
¿Afecta la temperatura?	si	si

5. Conclusiones

En cuanto a la aplicación de un filtrado a la señal de ultrasonido, aunque por el filtro FIR se haya demostrado con anterioridad que puede modificar la señal de ultrasonido, en la aplicación no es relevante ya que el espectro central de la onda no es modificado, obteniendo así una señal óptima para el análisis con los métodos seleccionados.

Con base a los dos métodos analizados en este artículo, se puede concluir que si bien los dos métodos (impedancia acústica y velocidad del sonido) pueden utilizarse para realizar una medición de nivel de líquido en un contenedor, cada una de ellas tiene un punto débil, es decir, a cada uno de estos métodos le afecta una o más variables.

Al método de impedancia acústica la variable que le afecta es la temperatura, ya que esta infiere en la medición, sin embargo la temperatura se puede linealizar para que este factor pueda ser minimizado, en cambio, al método de velocidad del sonido aparte de afectarle la temperatura, le afectan aspectos como las imperfecciones del contenedor, en este caso estas variables no pueden ser

controladas ya que depende del contenido del contenedor y el mantenimiento que se le hay dado que tenga o no imperfecciones el contenedor.

Si se analizan esas variables es posible tomar una decisión en referencia a cuál método es mejor para la aplicación (medición de nivel), el método de impedancia acústica ofrece menos impacto de variables que no se pueden controlar, esto lo hace un método más eficiente para la aplicación.

Referencias

[1] McCulloch, R. W. "*Measurement of liquid level in demanding environments using thermal differential technology.*" *Proc. Adv. Instrum* 45 (1990): 1859-1867.
[2] Zhang, J. L., and Y. L. Shan. "*Research on Ultrasonic Liquid Level Non-contact Measurement in Sealed Vessel*." *Piezoelectric Acoustooptics* 133 (2009): 366-370.
[3] Zhang, Minjuan, and Shiwei Li. "A method of the untouched ultrasonic liquid level measurement with high precision." *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. Vol. 7. IEEE, 2010.
[4] HE, Xi-ping, et al. "The calculation of the echo energy transmission coefficient in ultrasonic liquid level testing [J]." *Journal of Shaanxi Normal University (Natural Science Edition)* 2 (2008).
[5] Hao-hao, Hao, and Xiong Jun-qiao. "A method of liquid level measurement based on ultrasonic echo characteristics." *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. Vol. 11. IEEE, 2010.
[6] Sakharov, V. E., et al. "Liquid level sensor using ultrasonic Lamb waves." *Ultrasonics* 41.4 (2003): 319-322.
[7] Olympus Corporation, Ultrasonic Transducers Technical Notes, Houston, TX, USA:OLYMPUS NDT, 2011.
[8] Gómez A, et. al., “Desarrollo de un sistema electrónico para la generación y adquisición de señales de ultrasonido para el Diablo Instrumentado”, CIDESI, Querétaro, Qro, 2014.
[9] Kuang, Zhen-Bang, and Xiao-Guang Yuan. "Reflection and transmission of waves in pyroelectric and piezoelectric materials." *Journal of Sound and Vibration* 330.6 (2011): 1111-1120.
[10] Zhang, Bin, et al. "A liquid level measurement technique outside a sealed metal container based on ultrasonic impedance and echo energy." *Sensors* 17.1 (2017): 185.
[11] Alleyne, David N., and Peter Cawley. "Optimization of Lamb wave inspection techniques." Ndt & E International 25.1 (1992): 11-22.
[12] Henning B, Rautenberg J. Process monitoring using ultrasonic sensor systems. Ultrasonics. 2006 Dec 22;44 Suppl 1:e1395-9. doi: 10.1016/j.ultras.2006.05.048. Epub 2006 Jun 5.
[13] Zhang, Bin, et al. "A liquid level measurement technique outside a sealed metal container based on ultrasonic impedance and echo energy." *Sensors* 17.1 (2017): 176.
[14] Calvache, Bayron Andrés, and Asfur Barandica. "Prototipo Industrial de un Medidor Ultrasónico de Nivel." *Universidad del Valle. (Colombia* (2004).

Diseño Colaborativo en Mecatrónica, Capítulo 16, pp. 216 - 226
ISBN: 978-607-9394-23-3, 2021

Propuesta de prototipo de ventilador emergente a partir de una bolsa de reanimación.

Rivera Leyva Jossie E.[1*], Guzmán-Quezada Erick E.[1], Napoles Medran José M.[1], Schiavone Valdez Sergio D.[2], Curiel Anaya Jorge R.[2]

[1] Departamento de Electromecánica, Universidad Autónoma de Guadalajara
[2] Departamento de Biotecnología, Universidad Autónoma de Guadalajara
Guadalajara, Jalisco, México
[*]jleyva@edu.uag.mx

Resumen

La intención de este trabajo es describir el proceso de desarrollo de un ventilador mecánico basado en una bolsa de reanimación a la cual se hicieron las adecuaciones necesarias para poder automatizar la ventilación manual tradicional. La idea surge a través de la necesidad de construir dispositivos médicos más accesibles y de fácil uso para las situaciones medicas emergentes como la que se vive con la actual pandemia generada por el COVID-19 y que puedan presentarse en localidades rurales donde no se cuente con equipamiento de alta tecnología. Es por esto por lo que se ha decidido utilizar técnicas de desarrollo económicas como es la impresión 3D y elementos de hardware para prototipado rápido teniendo la finalidad de dejar los archivos e información OpenSource para quien desee continuar su diseño, mejorarlo o reproducirlo. Actualmente se cuenta con un prototipo funcional que permite la medición, adquisición y registro de la presión generada por la bolsa de reanimación a través de un hardware y software diseñados para cumplir las respectivas especificaciones y de una estructura mecánica que en conjunto podrá suministrar oxígeno a las pacientes con problemas respiratorios.

Palabras clave: Ventilador mecánico, asequible, COVID-19, equipo médico emergente, impresión 3D

1. Introducción

El uso de la ventilación mecánica ha demostrado que puede salvar vidas desde la década de 1950 con la epidemia de la poliomielitis y desde entonces ha tenido un incremento importante en el uso de soporte ventilatorio, sin mencionar que se relaciona estrechamente con el cuidado critico de la salud [1]. Con el tiempo, los pacientes con problemas pulmonares pueden llegar a desarrollar fallas respiratorias, llevándolos a necesitar soporte de ventilación mecánica. Este tipo de dispositivos asiste a los pacientes a inspirar y espirar, permitiendo de esta manera el intercambio de oxígeno y dióxido de carbono [2]. Aunque muchos de los hospitales en México tienen la capacidad de contar con este tipo de equipos, no todos pueden hacerse de uno debido a sus altos costos de adquisición que pueden llegar tener, estando cerca de los $30,000 USD.

Como es bien sabido, la epidemia generada a inicios de 2020 a causa de la COVID-19 se expandió rápidamente por el mundo, teniendo como punto de partida la ciudad de Wuhan, China, a finales del año 2019. La propagación de esta enfermedad tuvo un incremento exponencial por todo el mundo, el primer caso confirmado en México se confirmo el día 27 de febrero de 2020 y 64 días después, 30 de abril, ya se tenia registro de 19,224 casos y 1,859 fallecidos [3, 4]. Para este momento, todos los sistemas nacionales, tanto internacionales como nacionales, previeron una falta de equipos de ventilación mecánica en las instituciones de salud que pudieran cubrir todos los casos de enfermos para cuándo llegará el pico de la pandemia. Desafortunadamente, los vendedores y fabricantes de estos equipos tuvieron un incremento notable en la demanda de ventiladores, pero era imposible cumplir todos

los requisitos que se presentaban en todo el mundo. Esto llevo a que ventiladores mecánicos de emergencia fueran desarrollados en diferentes instituciones, tanto públicas como privadas [5]. El uso de tecnologías de manufactura de pequeña escala, como las impresoras 3D y los microcontroladores *OpenSource* ayudó a que prototipos de todo el mundo se empezarán a dar a conocer rápidamente, teniendo como objetivo común el compartir la información libremente para que más personas puedan tomar los archivos desde internet y reproducirlos.

A continuación, se presenta una propuesta del desarrollo de un ventilador mecánico que se suma a esta colaboración internacional por desarrollar equipos asequibles, teniendo en cuenta la manufactura rápida y simple. Este prototipo se divide en tres etapas de ingeniería, la primera de ellas es el desarrollo de un hardware diseñado para cumplir las necesidades del proyecto, este consta de un circuito impreso capaz de comunicarse con un microcontrolador de desarrollo libre permitiendo la adquisición, procesamiento y despliegue de la información. Seguido a esto, se desarrollo un software, encargado de tomar las decisiones a través de la información adquirida por el hardware. Por último, se construyó una estructura mecánica que tiene la tarea de almacenar los elementos anteriores y de interactuar con los usuarios a la cual se le conectarán todos los elementos movibles que este equipo contempla.

2. Métodos

Todo dispositivo electrónico que utilizamos en la actualidad necesita de tres grandes equipos de ingeniería que sean capaces de llevar el concepto de una idea hasta una etapa de manufactura. Estos equipos, aunque la industria no obligatoriamente los clasifica así, podemos decir que siempre necesitan de un área de electrónica que pueda desarrollar la parte de hardware, un área de software que desarrolle algoritmos específicos para cada función y, por último, la parte mecánica, la cual debe ser capaz de proteger los dos campos anteriores e interactuar con el medio ambiente y con los usuarios que se contempla que el proyecto será expuesto.

Este proyecto no es la excepción a la continuación de este tipo de diseños de nivel industrial, a continuación, se presenta la descripción de las tres etapas de diseño que fueron necesarias para poder desarrollar este proyecto.

2.1 Etapa de hardware

Se desarrollo una tarjeta electrónica capaz de medir la presión de aire generada por una bolsa de reanimación a través de un sensor de presión piezoresistivo de la marca NXP, este sensor tiene una alta precisión y convierte la información de entrada en un voltaje lineal de corriente directa a la salida (etapa 1). Desafortunadamente el rango de operación va de los 20 mV a los 50 mV, haciendo difícil la detección de los cambios pequeños en el voltaje. Es por esto por lo que se decidió agregar un amplificador de instrumentación modelo AD620 de 8 pines, perteneciente a la marca Analog Devices con una ganancia de 20, llegando así a un rango de 400 mV a 1V (etapa 2), teniendo como punto central 0 v, como un máximo negativo de -500 mV y un máximo positivo de +500 mV. Para hacer esto, la hoja de especificaciones del fabricante menciona que se debe utilizar la siguiente ecuación:

$$G = 1 + (50k\Omega / Rg) \tag{1}$$

Donde la *G* es el valor de la ganancia de la señal de entrada con respecto a la señal de salida y *Rg* es la resistencia de ganancia que permite varias los valores de amplificación del circuito, para este caso se propuso usar una resistencia aproximada de 2.631kΩ.

Seguido a esto se implemento un circuito de offset basado en un amplificador operacional de propósito general conectado a una resistencia variable que le permite desplazar la señar de 0 a 5 v de forma positiva. El objetivo de usar esta etapa fue desplazar la parte de voltaje negativa generada por el sensor de presión hasta llegar a obtener una señal totalmente positiva, dejando así la señal en un punto de operación que va desde 1 v a 2 v, teniendo como punto central 1.5 v (etapa 3). La salida de este

punto del circuito se conecta a uno de los seis convertidores analógico/digital que se encuentra en la tarjeta de desarrollo seleccionada para este proyecto y que cuenta como componente principal con el microcontrolador Atmega328p.

Por último, es importante mencionar las etapas extras con las que cuenta la tarjeta, que si bien no tienen un impacto directo en la lectura de datos del sensor de presión son importantes para el correcto funcionamiento del proyecto en general. La primera de ellas es un regulador de voltaje que puede convertir un voltaje de entrada de 7 volts hasta 35 v en una salida de 5 v, con este voltaje se alimenta el resto de las etapas de todo el circuito. La siguiente consta de un circuito integrado conocido como “bomba de carga”, el objetivo de este circuito es invertir un voltaje de entrada a su forma negativa, lo que significa que toma los 5 v de entrada para poder convertirlos en -5v, este componente se integro en el circuito debido a que los amplificadores requieren tanto de un voltaje positivo como de uno negativo para su propia alimentación. La tercera etapa consta de dos LEDs que sirven como indicadores de que el circuito este encendido y que la alimentación 5 y -5 v están funcionando correctamente. Seguido a esto se colocaron cinco resistencias variables que permitirán el ajuste de los diferentes parámetros del ventilador mecánico. Como quinta etapa se agrego un modulo de motor a pasos A4988, el cual estará encargado de recibir las instrucciones provenientes del microcontrolador para mover una pieza mecánica que servirá para presiona la bolsa de reanimación. Por último, se incluyo una pantalla de cristal líquidos donde se muestran la información necesaria para los usuarios de este ventilador mecánico. Todas estas etapas pueden ser observadas en el diagrama de flujo que se muestra en la figura 1.

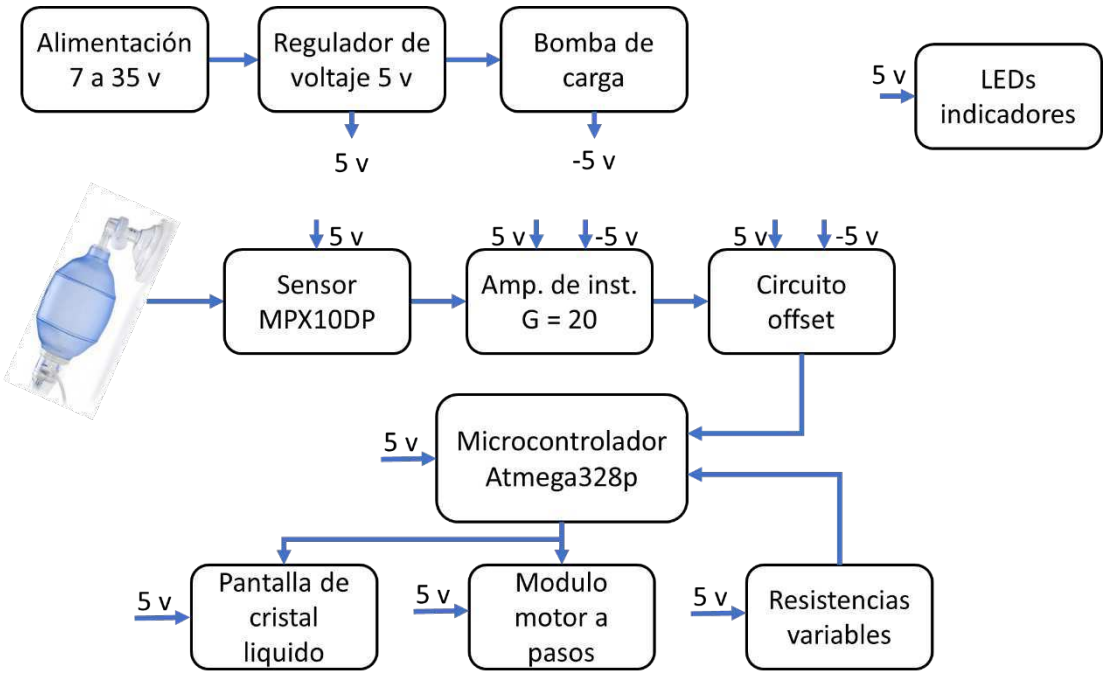


Figura 1. Diagrama a bloques de circuito electrónico.

2.2 Procesamiento de las señales

Consiste en un circuito de ventilación convencional para paciente adulto y una cánula de ventilación invasiva, conectados a una bolsa de reanimación tamaño adulto con la que se ventilará de manera mecánica al paciente montado en una base y a través del movimiento de un motor. La base se tomó de un recurso en línea open source para sus pruebas iniciales.

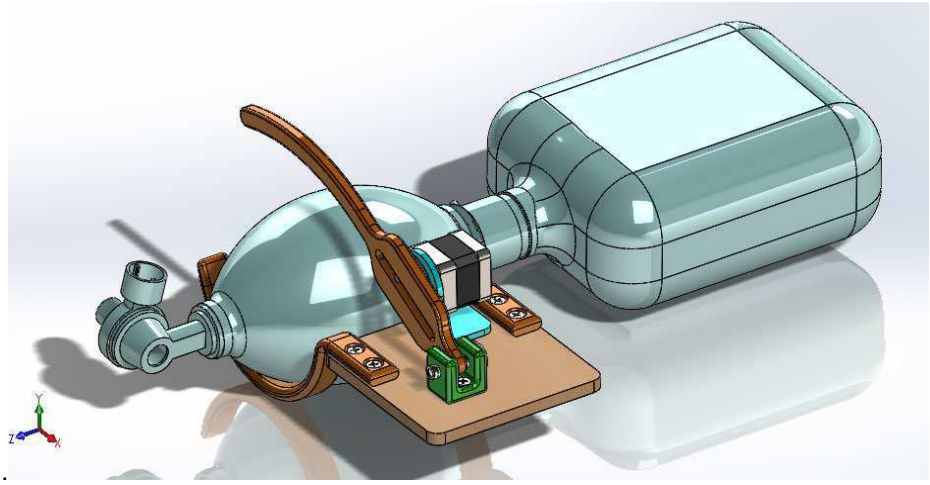


Figura 3. Ejemplo de simulación de la base.

2.2.1 Software

El software permite controlar todos los elementos electrónicos y la seguridad del paciente y el equipo. Lo hace de manera modular y controlando el procesamiento de cada tarea dependiendo de las entradas del usuario.

Montado sobre una base de Arduino uno, el software se realizó de manera modular controlando cada canal de conversiones de Analógico a Digital de la tarjeta con la finalidad de realizar conversiones del motor a pasos y pasarlo a el control binario en ambos sentidos de giro del motor con la finalidad de regresar el mecanismo a su posición original a determinarse por el giro del potenciómetro de rango de volumen, esto permitirá controlar el brazo para la presión sobre la bolsa de reanimación. Los canales de conversión se declararon para cada parámetro a controlar en el ventilador mostrados en la tabla.

Tabla 1. Detalle de parámetros ventilatorios propuestos en el prototipo a determinar por el operador

Parámetro		Rangos	Alarma
Volumen	Input	2 a 2000 ml	+- 10%
Respiraciones por minuto (BPM)	Input	0 a 40	+- 10%
Modo ventilatorio	Input	AC / Control volumen	
I:E	Input	1:2, 1:3, 1:4	
Presión en vía aérea	Sensor	20 a 40 cmH2O	+- 10%
FiO2	Sensor	21 - 100 %	+- 10%
Duración del ciclo respiratorio en segundos	Local	π=60/BPM	N/A
Tiempo de exhalación	Local	Texh=E*π/(1+E)	
PEEP	Input	0 a 20cmH2O	+- 10%
Esfuerzo del paciente	Sensor		

La lectura de dichos canales se realiza para el control del equipo y de manera modular se establecen los parámetros propuestos por el usuario, el equipo también cuenta con alarmas de seguridad ya que se trata de un equipo que mide constantemente la respuesta del paciente y de la presión proximal con la finalidad de salvaguardar la seguridad del paciente y su vía aérea. Entre las alarmas de seguridad se encuentran:

Presión alta: Esto puede venir de varias fuentes relativas a la vía aérea del paciente indicando obstrucción en el circuito.
Presión baja: Puede indicar fuga en el circuito ventilatorio ocasionado por desconexión.
Esfuerzo significativo del paciente
Parámetro fuera de rango o incompatible

Estas alarmas se alimentan de las mediciones del sensor de presión que permite medir esfuerzos del paciente por ser de característica diferencial y la presión en vía aérea por seguridad.

El control del motor manda información de salida para un circuito que permite controlar el sentido del motor y sus pasos, a través de la lectura de canales de analógico a digital se realiza una conversión que permite hacer equivalente dicha medición a los pasos del motor y su sentido.

2.3 Diseño mecánico

La base está diseñada para impresión 3D fácilmente escalable y con material con posibilidad de desinfección. Sus piezas son de una plataforma *Open Source* por lo que no es necesario modificar dado su acceso abierto.

El motor utilizado en combinación con un *driver* A4988 permiten movilizar las piezas del brazo que oprime la bolsa de reanimación es un motor de 200 pasos. El circuito A4988 requiere de dos alimentaciones, ambas provenientes de la tarjeta que con anterioridad se describe como el diseño de hardware. El control de Vref en el *encoder* permite controlar la Intensidad a la que trabajará el motor con la finalidad de alimentarlo hasta 1ampere.

2.4 Blender

Como parte de su diseño inherente a su uso, las bolsas de reanimación cuentan con conexión a través de una manguera para proporcionar Oxígeno al 100% de concentración a través de un tanque portátil o toma de pared de Oxígeno medicinal. Sin embargo si es necesario regular la fracción inspirada de Oxígeno (FiO_2), hay dos formas de hacer *blending* o mezclas de gases: uno usa un flujómetro de Oxígeno y aprovecha la válvula de admisión de la bolsa de reanimación; el otro, usa un flujómetro de Oxígeno y un flujómetro de aire médico.

Los flujómetros propuestos para la mezcla son los modelos ya utilizados en hospitales de manera estandarizada. Estos dispositivos también se conocen como rotámetros compensados o como medidores de flujo de tubo Thorpe.

Blender 1: mezclado por compensación (usando válvula de admisión de la bolsa de reanmiación.
Materiales: Flujómetro de oxígeno

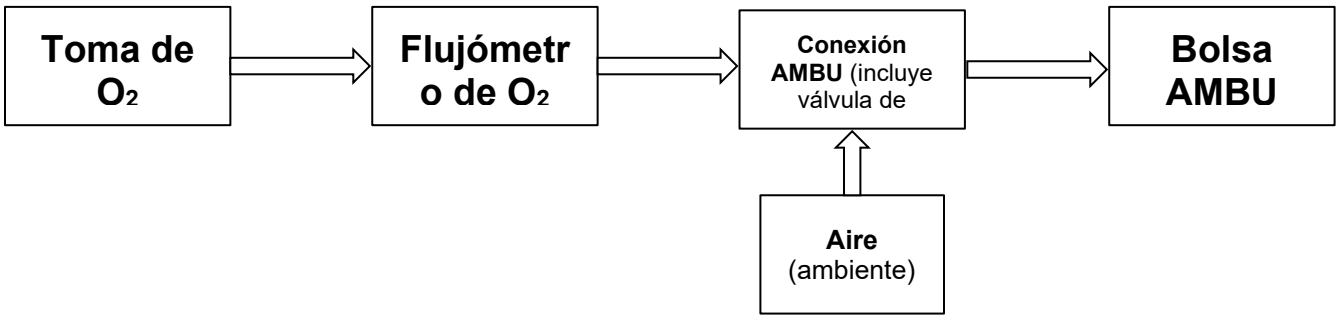


Figura 5.- Diagrama de bloques del Blender Tipo 1

Cuando se administra oxígeno con un FiO_2 de 100% usando una bolsa de reanimación, la bolsa reservorio de O2 nunca se desinfla; en cambio, si el flujo de Oxígeno es menor al respiratorio (la bolsa reservorio estará colapsada), se abre una válvula de seguridad (válvula de admisión de aire) que compensa con aire del ambiente la caída de presión provocada por la expansión de la bolsa.

Aprovechando este mecanismo se puede mezclar oxígeno con aire para obtener un FiO_2 con sólo cambiar el flujo de oxígeno (usando el flujómetro indicado) pues el resto es compensado con aire. La expresión con la que se puede calcular el flujo de oxígeno necesario, dado un flujo respiratorio y un FiO_2 es la siguiente:

$$QO2 = QT(FiO2-21)/79 \quad (1)$$

donde QO2 es el flujo de O2 (L/min) a administrar con el flujómetro, QT es el flujo respiratorio (L/min) y FiO2 es la fracción inspirada de oxígeno, puesta en forma porcentual [2].

Nota: La ecuación (1) está reportada en la literatura médica, pero a veces usan el término O2% en vez de FiO2

Alternativamente, QT de la ecuación (1) puede ser expresado como la multiplicación de la frecuencia respiratoria, f (respiraciones/min), por el volumen tidal, VT (litros/respiración). En términos matemáticos: QT=f *VT (Tobin, M., 2013. pg.74). Sustituyendo QT de la ecuación (1) se obtiene:

$$QO2 = f * VT(FiO2-21)/79 \quad (2)$$

Si se cuenta con un sensor de oxígeno, las ecuaciones no son necesarias, pero sirven como referencia.

Blender 2: mezclado usando dos flujómetros
Costo: 3000 + costo de tubo extra y
Materiales: flujómetro de oxígeno, flujómetro de aire, conector de tres vías, tubo adicional.

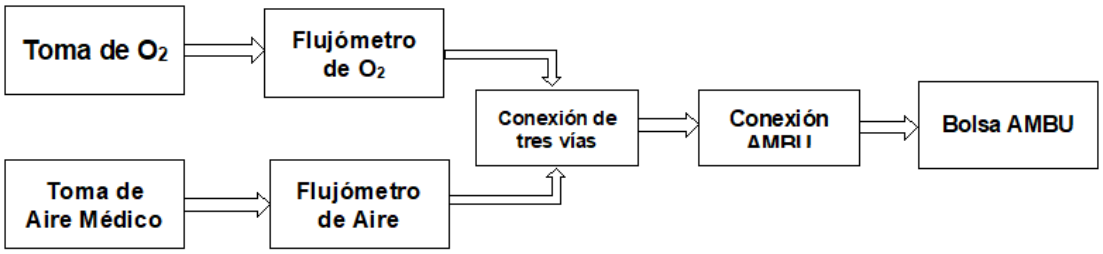


Figura 6- Diagrama de bloques del Blender 2

El aire médico tiene una composición prácticamente idéntica a la del aire del ambiente pero proviene de una toma de gas. Un flujo de aire médico puede ser mezclado con otro de O2 usando un conector de tres vías, el cual va conectado a la entrada del AMBU originalmente destinada para O2. A partir de la ecuación (1) se calcula el QO2 requerido y se sabe que Qaire=QO2-QT.

Esta forma de mezclar gases requiere que la bolsa reservorio se mantenga inflada. La válvula de admisión tiende a abrirse cuando la bolsa de reanimación se expande aunque la bolsa reservorio esté inflada (dependiendo de la marca y modelo), lo cual lleva a que el FiO_2 real se un poco menor al esperado. Si se cuenta con un sensor de oxígeno, nuevamente quedan las ecuaciones de referencia.

2.5 Equipamiento y herramientas

El equipamiento utilizado para pruebas y medición fue el siguiente:

1. Osciloscopio, multímetro y canales de ADC del Arduino para medición de parámetros.
2. Equipo Biopac, mediciones de flujo para comparación de pasos de motor para determinar el volumen correspondiente al movimiento del brazo. Se trata de un equipo para pruebas de fisiología con el cual el sensor de flujo se mide a la salida de la bolsa de reanimación con la finalidad de conocer el flujo y volumen. Ésta prueba corresponde a la medición de capacidad pulmonar en una prueba de fisiología pulmonar utilizando el equipo de medición de parámetros fisiológicos.
3. Software para programación y administración de proyectos de programación.
4. Bolsa de reanimación y circuito de paciente grado médico (marca AMBU).
5. Impresora 3D para base y brazo motor.

3. Resultados del diseño propuesto

El equipo está diseñado y pensado en 2 modos venitlatorios que permiten las siguientes funcionalidades que la mayoría de los prototipos de fabricación rápida no tienen:

- Control de oxígeno y su concentración y medición de dicha mezcla
- Mide la reacción del paciente a través de un sensor de presión diferencial por lo que se puede detectar el esfuerzo del paciente si es que tuviera la necesidad de un modo ventilatorio no controlado.

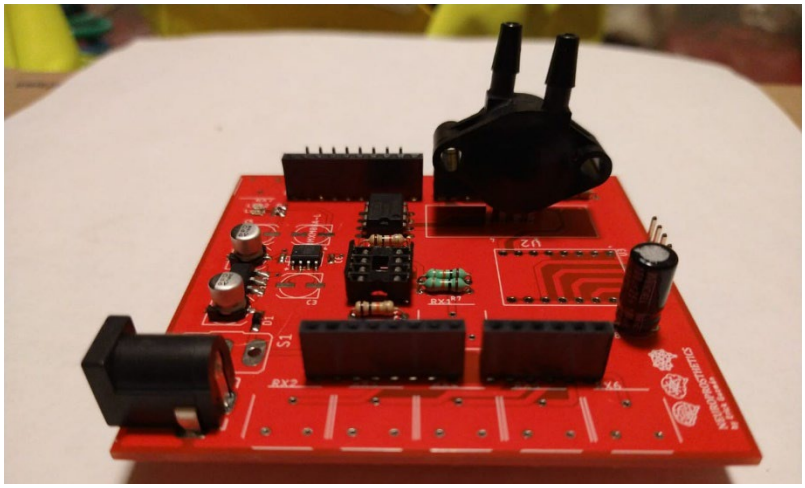


Figura 7. Tarjeta previa al ensamblaje de componentes.

- Los controles permiten modificar los siguientes parámetros:
    - Modos ventilatorios (Asisto/Control y Controlada por Volumen)
    - PEEP
    - Volumen
    - Frecuencia respiratoria
    - Fracción inspirada de Oxígeno
    - Límite de esfuerzo del paciente
    - Frecuencia respiratoria
- Bajo costo de fabricación y viabilidad

Figura 7. Resultados de medición del sensor de presión diferencial en equipo de laboratorio (Osciloscopio).

El equipo es de muy bajo costo y sus componentes permiten el ensamblaje rápido. Las piezas se podrán producir a gran escala o en un ensamblaje de manufactura rápida con ayuda de una impresora 3D, cortadora láser y elementos electromecánicos que son estándar y de fácil acceso.

Existen una gran cantidad de proyectos que actualmente se están realizando de manera académica en México e incluso como en el caso de Estados Unidos se han logrado producir de manera masiva con aprobación de la FDA. Sin embargo, en algunos casos carecen de viabilidad a pesar de tener varios elementos similares. El caso del Instituto Tecnológico de Massachussets tiene gran viabilidad y cuenta con pruebas con miras a su comercialización. Una gran desventaja en varios diseños, es la falta del conocimiento de los procesos ventilatorios y sus implicaciones en el caso de otros proyectos que si bien cuentan con un diseño que ventila y permite generar un volumen, carecen de lo especializado de un control de los modos ventilatorios y los sensores necesarios para poder ventilar adecuadamente a un paciente. Por mencionar un ejemplo los equipos ventilatorios manejan un porcentaje de rango de entrega de volumen contra la medición exacta en ml. Es decir, mencionan que pueden entregar 40% o 60% sin saber exactamente la cantidad de mililitros o volumen corriente entregados al paciente que debe oscilar de acuerdo a las guías de práctica clínica de España para pacientes con COVID-19 en rangos de 4 a 6 ml/kg de peso. El equipo es de muy bajo costo para producir y la mano de obra ha sido de un grupo de trabajo que busca aportar un bien a la sociedad a través del diseño de un modelo asequible.

Tabla 2. Costos del prototipo sin suministro de Oxígeno.

Concepto	**Costo estimado en USD**
Tarjeta ARDUINO	$5.00
Tarjeta de periféricos y control con componentes	$35.00
Base (impresión 3D) con motor	$17.00
Bolsa de reanimación	$40.00
Circuito Paciente	$12.00
Eliminador	$12.00
Total estimado en USD	$121.00

A diferencia de otros prototipos, ésta propuesta es de muy bajo costo (rondará los $121 USD) y todos los materiales son de fácil y rápido acceso/obtención. Además, cuenta con elementos de impresión 3D que debido a su desarrollo en los últimos años se puede encontrar fácilmente y además es económico (se incluirá el archivo para su descarga e implementación). Se requerirá un tanque o toma mural de oxígeno que es de uso esencial en los hospitales por lo que no habría tanto problema para poder conseguirlo. Medirá los siguientes parámetros: Volumen, PEEP (presión positiva al final de la exhalación), frecuencia respiratoria, relación inspiración/exhalación, límite del esfuerzo del paciente y modo ventilatorio.

Además de las pruebas de funcionamiento donde se determina el funcionamiento del equipo, se buscará una prueba con un equipo de simulación para medicina con la finalidad de buscar una validación con parámetros fisiológicos que puedan determinar que tan confiable es el equipo y simular su uso en pacientes detallando resultados a partir de dicha prueba. Otro de los resultados en las pruebas realizadas en el prototipo fue la base, ya que se buscó perfeccionarla a una base similar a la figura por la falta de fricción y uso en el brazo para mecanizar el movimiento.

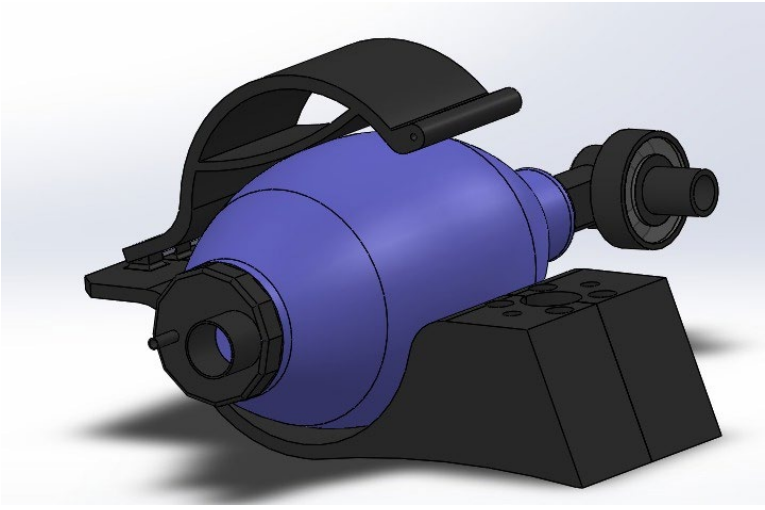


Figura 7. Nueva base a considerar

Otro punto importante entre los hallazgos fue que los materiales deben ser compatibles con el ambiente hospitalario ya que si bien se trata de un prototipo que se usa por el momento para realizar pruebas y dentro de un ambiente de laboratorio se deberán realizar las siguientes consideraciones al momento de rediseñar algunos aspectos del prototipo en su segunda versión:

1. Material de la base y mecanismo de uso.
2. Caja donde se montará todo el circuito y sus componentes debe estar aislada correctamente ya que se trata de un ambiente saturado de oxígeno y puede ocurrir un accidente.
3. No se deben usar materiales como madera ya que pueden ser propensos a acumular bacterias o infección y difíciles de desinfectar.
4. El equipo puede funcionar sin una conexión de oxígeno ya que puede tomarse concentración al 21% o al 100% si se conecta directamente a la bolsa de reanimación.
5. Filtros de los orificios de entrada - El ventilador utiliza todo estandarizado y fácilmente adaptable de material dentro del ambiente hospitalario.

4. Conclusiones y pasos a seguir

El prototipo dista de ser uno que se pueda utilizar en pacientes directamente, sin embargo está encaminado a ser una solución para situaciones de emergencia ya que durante 2020 si bien los países

estaban dispuestos a realizar la adquisición de los equipos de manera emergente para el manejo de pacientes con SARS-COV2, no había recursos disponibles para atender a dicha población. Algunas de las medidas extremas fueron habilitar las fábricas de estándares automotrices para producir de manera acelerada ventiladores que pudieran atender a la población en la situación de emergencia. Algunos prototipos fabricados en el país y en el extranjero fueron derivados de los esfuerzos previos de las universidades con la finalidad de acelerar el proceso de producción de diseños nuevos, fáciles de producir y de muy bajo costo comparados con otros existente para aliviar la presión en los sistemas de salud.

Los ventiladores mecánicos por tratarse de equipos que son soporte de vida requieren de pruebas exhaustivas en su desarrollo lo cual debe realizarse con tiempo y de manera segura ya que se trata de equipo que proporciona al paciente soporte ventilatorio. En el caso de este equipo se busca sea de clasificación de COFEPRIS “CLASE II”. En cumplimiento con la clasificación de COFEPRIS el equipo permanece menos de 30 días en el cuerpo de manera invasiva por lo que se clasifica como Clase II por lo que se debe buscar que sus características estén dentro de los parámetros que se marcan para su registro sanitario.

La importancia de impulsar estos proyectos a nivel de prototipado es de suma trascendencia ya que se trata de comprender el funcionamiento a profundidad de un equipamiento médico que potencialmente puede continuar salvando vidas. El prototipado de estos equipos permite avanzar en el conocimiento y desarrollo tecnológico de pruebas, de equipos y la disponibilidad de la tecnología. Para este prototipo en particular se busca llegar a su funcionamiento y sus pruebas con simuladores de paciente con funcionamiento fisiológico para probar dicho software y hardware. También se busca en un momento dado impulsar este tipo de prototipos continuando con la mejora de la disponibilidad de dichos equipos, en los últimos años todas las epidemias han tenido que ver con enfermedades respiratorios como lo son la polio y la influenza por lo que el tipo de equipamiento relativo a las enfermedades respiratorios requiere de un impulso continua a la mejora y asequibilidad de dichos dispositivos médicos con miras a su producción incluso y diseño a nivel nacional, de ahí su trascendencia.

En estos tiempos lo que normalmente puede tomar muchos años de desarrollo al apoyarse de tecnologías como la impresión 3D y la manufactura acelerada, así como los recursos compartidos en la red y comunidades dedicadas al desarrollo y avance de estos dispositivos han permitido acelerar el proceso para que dichos equipos estén disponibles a través de prototipos y avances realizados en dichos proyectos. De manera posterior se buscará la recuperación de pacientes COVID con algunas ayudas como lo son la rehabilitación pulmonar en casa o ejercicios para fortalecer la capacidad vital pulmonar, sin embargo dispositivos médicos como ventiladores mecánicos y monitores continuarán siendo parte de la historia relacionados a la pandemia y enfermedades que han aquejado a la humanidad por lo que el crecimiento en el desarrollo tecnológico en dicha área es de vital importancia.

Referencias

[1] Tobin, M. J. (1994). Mechanical ventilation. New England Journal of Medicine, 330(15), 1056-1061.
[2] Al Husseini, A. M., Lee, H. J., Negrete, J., Powelson, S., Servi, A. T., Slocum, A. H., & Saukkonen, J. (2010). Design and prototyping of a low-cost portable mechanical ventilator. Transactions of the ASME-W-Journal of Medical Devices, 4(2), 027514.
[3] Farré, R., Puig-Domingo, M., Ricart, P., & Nicolás, J. M. (2020). Ventiladores mecánicos de emergencia para la COVID-19. Archivos De Bronconeumologia, 56, 7.
[4] Suárez, V., Quezada, M. S., Ruiz, S. O., & De Jesús, E. R. (2020). Epidemiología de COVID-19 en México: del 27 de febrero al 30 de abril de 2020. Revista clinica espanola, 220(8), 463-471.
[5] Pearce, J. M. (2020). A review of open source ventilators for COVID-19 and future pandemics. F1000Research, 9.

[1] MÉXICO AUMENTÓ CUATRO VECES SU PRODUCCIÓN DE VENTILADORES DE RESPIRACIÓN ASISTIDA. Junio, 2020. México.
https://www.elhospital.com/temas/Mexico-aumento-cuatro-veces-su-produccion-de-ventiladores-de-respiracion-asistida+134664

[2] Guillaume, D. W., & DeVries, D. (1992). *Ambient pressure air/oxygen blender. Journal of Biomedical Engineering*, 14(2), 154. doi:10.1016/0141-5425(92)90021-c] Guillaume, D. W., & DeVries, D. 1992. pg 154.

[3].- Lohr, Steve, New York Needed Ventilators. "*So They Developed One in a Month.*" NY Times, Abril 2020. Estados Unidos.
https://www.nytimes.com/2020/04/20/technology/new-york-ventilators-coronavirus.html
[4].- *Suman a 146 hospitales privados para enfrentar la Crisis Sanitaria*. El Informador. Abril, 2020. México.
https://www.informador.mx/Suman-a-146-hospitales-privados-para-enfrentar-la-crisis-sanitaria-l202004140001.html
[5].-Wells, Chad R. et al. *Projecting the demand for ventilators at the peak of the COVID-19 outbreak in the USA*. The Lancet - Infectious Diseases. Abril 2020. Estados Unidos.
https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30315-7/fulltext
[6].- Tobin, M., 2013. *Principles And Practice Of Mechanical Ventilation.* 3rd ed. New York: McGraw-Hill Medical, p.74.

Diseño Colaborativo en Mecatrónica, Capítulo 17, pp. 227 - 245
ISBN: 978-607-9394-23-3, 2021

Diseño de un péndulo invertido sobre dos ruedas como prototipo de prácticas para Control Automático.

Araguz-Lara Paulina, Flores-Gómez Amalia, Garza-Martínez Mateo, Orrante-Sakanassi Jorge Alberto

Tecnológico Nacional de México/Instituto Tecnológico de Matamoros

Resumen

Los laboratorios del Instituto Tecnológico de Matamoros (ITM) carecen de plantas funcionales que permitan a los estudiantes materializar o experimentar con lo que se aprende en el salón de clases en los cursos de Control Automático. Ante esta premisa, los resultados obtenidos en este proyecto servirán para su uso posterior en las prácticas de laboratorio de las asignaturas afines a Control Automático del ITM, permitiendo al estudiante aterrizar la teoría aprendida en un sistema no lineal.

Se diseñó y construyó un prototipo funcional de un sistema péndulo invertido sobre dos ruedas de bajo costo. Se presenta también una comparación y reducción de costos al elaborar el diseño propuesto. Además, se diseñan, analizan e implementan esquemas de control PID al sistema por medio de pruebas experimentales implementadas en tiempo real utilizando la tarjeta Arduino UNO como medio de adquisición de datos y control embebido. Finalmente, se presenta el modelo matemático no lineal de la planta el cual es obtenido mediante la metodología Euler-Lagrange.

Palabras clave: Euler-Lagrange, Péndulo invertido, PWM, Sistema no lineal, Control PID.

1. Introducción

El diseño de control ha tomado gran importancia hoy en día, no solo en el ámbito ingenieril, si no en el día a día de los seres humanos. Aun así, es preciso mencionar que, para un estudiante, el acceso a una planta funcional suele ser algo exorbitantemente costoso, ocasionando una gran limitación al momento de intentar adquirir una, y por lo tanto, limitaciones a la hora de poner en práctica la teoría aprendida en el aula. Ante esta premisa, a falta de material de laboratorio para las materias afines a Control Automático en el Instituto Tecnológico de Matamoros (ITM), se pretende realizar un prototipo de bajo costo que permita cubrir las necesidades de ensayos experimentales requeridos en esta asignatura y afines.

Los resultados obtenidos en este trabajo servirán para su uso posterior en las prácticas de laboratorio de las asignaturas afines a Control Automático del ITM, permitiendo al estudiante aterrizar la teoría aprendida en un sistema no lineal. Y es que es bien sabido que la teoría siempre debe de ir de la mano de la práctica para traer a la realidad ese conocimiento intangible que muchas veces no queda totalmente claro si no se concreta en nuestra realidad. Además, generalmente en los cursos de control se proponen ejercicios abstractos que no permiten tener un panorama completo sobre el comportamiento de la planta ni permiten desarrollar técnicas de diseño enfocadas a un objetivo específico. Con esta planta, el estudiante podrá utilizar su capacidad de análisis y razonamiento, además de realizar un sinfín de prácticas beneficiosas para su formación académica.

Aunado a todo esto, los resultados obtenidos en este proyecto servirán para su uso posterior en los cursos de licenciatura y posgrado en el área de control, además, también reforzarán las técnicas de

diseño estudiadas en los cursos introductorios de control. El control proporcional-integral-derivativo se puede implementar en una gran cantidad de sistemas y aporta grandes ventajas al control de la planta, ya que con la simple sintonización de 3 ganancias, es posible obtener una respuesta de control aceptable, con rechazo a perturbaciones y sin necesidad de conocer los parámetros de la planta.

2. Objetivos

Diseñar, analizar, e implementar esquemas de control a un sistema un péndulo invertido sobre dos ruedas como prototipo de apoyo para las prácticas de laboratorio de las asignaturas afines a Control Automático. Se pretende armar un prototipo funcional de bajo costo a comparación de aquellos prototipos que se venden en el mercado.

Con el prototipo puesto en marcha será posible el modelado, análisis y diseño de sistemas de control, y la realización de ensayos experimentales de propuestas de investigación de la institución.

OBJETIVOS ESPECÍFICOS:

- Obtener los materiales del prototipo a bajo costo
- Construir el prototipo funcional del sistema péndulo invertido sobre dos ruedas
- Obtener el modelo matemático no lineal de la planta.
- Diseñar un programa y realizar pruebas experimentales para obtener un controlador eficiente.
- Obtener un controlador proporcional-integral-derivativo para la planta.
- Equilibrar la planta con el controlador diseñado e implementado.

.

3. Limitaciones

Debido a la pandemia y contingencia sanitaria provocada por el virus SARS-CoV-2 (COVID-19) no fue posible las reuniones presenciales del equipo, por lo tanto, la comunicación se realizó a través de videollamadas y recursos digitales. Esto se llevó a cabo para acatar las medidas de seguridad sanitaria impuestas por la Secretaría de Salud, entre ellas, suspensión de actividades no esenciales y cumplimiento de cuarentena hasta nuevo aviso.

Aunado a lo anteriormente mencionado y al cierre de negocios, la obtención de materiales se realizó en tiendas en línea casi en su totalidad, esperando un retraso en el desarrollo del proyecto al estar a expensas de la entrega de productos por medio de paqueterías bastante congestionadas gracias al gran aumento de demanda del servicio en tiempos de pandemia.

4. Antecedentes

Aunque se sabe que las primeras aplicaciones del Control Automático aparecieron hace más de 2000 años, fue la Revolución Industrial la que detonó su desarrollo como un conjunto de conocimientos científicos destinados a resolver problemas tecnológicos. Desde entonces, el uso del Control Automático ha sido fundamental para que las actividades productivas del ser humano se hagan cada vez más eficaces incrementando la calidad y la repetibilidad de los productos [1].

Si el sistema opera alrededor de un punto de equilibrio y si las señales involucradas son pequeñas, es posible aproximar el sistema no lineal mediante un sistema lineal. Este sistema lineal es equivalente al sistema no lineal, considerado dentro de un rango de operación limitado. Tal modelo linealizado (lineal e invariante con el tiempo) es muy importante en la ingeniería de control [2].

El péndulo invertido (Figura 5.1) es conocido por ser uno de los problemas más importantes y clásicos de la teoría de control. El sistema se compone de un carro sobre el cual se monta un péndulo

que puede girar libremente. El carro deberá moverse para compensar el desplazamiento del péndulo y mantenerlo, así, en equilibrio, esto es, en su parte vertical superior [3].

Los péndulos invertidos son una familia de dispositivos que constituyen un banco de pruebas muy completo e interesante para la ingeniería de control no lineal. El más estudiado de los miembros de esta familia es el denominado control invertido sobre un vehículo. Consiste en un péndulo o varilla que gira libremente por uno de sus extremos mediante una articulación situada sobre un carro que se mueve sobre una guía rectilínea horizontal bajo la acción de una fuerza F, que es la acción de control con la que se pretende actuar sobre la posición de la varilla [3].

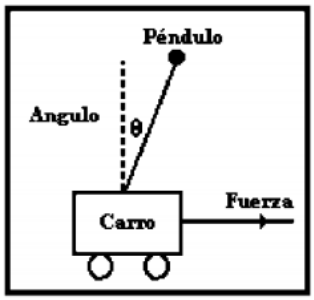


Figura 5.1. Diagrama péndulo invertido.

5. Marco teórico

5.1 Sistemas no lineales

Un sistema es no lineal si no se cumple el principio de superposición. Por tanto, para un sistema no lineal la respuesta a dos entradas no puede calcularse tratando cada entrada a la vez y sumando los resultados [2]. La dinámica de un sistema no lineal puede ser descrita mediante una representación en espacio de estados cuya forma es la siguiente:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned} \tag{5.1}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}; u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}; f(x,u) = \begin{bmatrix} f_1(x,u) \\ f_2(x,u) \\ \vdots \\ f_n(x,u) \end{bmatrix} \tag{5.2}$$

donde $x \in \square^n$ es la variable de estado, $u \in \square^m$ es la señal de entrada, $y \in \square^q$ es la señal de salida. Además, $f(x,u)$ y $g(x,u)$ son funciones no lineales [2].

5.2 Dinámica Euler-Lagrange

La función de Lagrange o Lagrangiano $\mathcal{L}(q,\dot{q})$ resume la dinámica de un sistema por medio de las energías que actúan sobre él. Se define como la energía cinética $\mathcal{K}(q,\dot{q})$ menos la energía potencial $\mathcal{P}(q)$, es decir el Lagrangiano de un robot manipulador de n grados de libertad se define como:

$$\mathcal{L}(q,\dot{q}) = \mathcal{K}(q,\dot{q}) - \mathcal{L}(q) \tag{5.3}$$

donde $q = [q_1, \cdots, q_n]^T$ es la coordenada generalizada, mientras que $\dot{q}$ es la velocidad de la coordenada generalizada.

Conociendo el Lagrangiano, se pueden obtener las ecuaciones de movimiento mediante la sustitución directa de la expresión para la función de Lagrange en la ecuación de Euler-Lagrange, que se expresa como:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q} = F \tag{5.4}$$

donde F es el vector de fuerzas generalizadas del sistema.

Esta ecuación se utiliza para describir cualquier sistema mecánico por medio de coordenadas generalizadas de posición y su velocidad, mientras que el Lagrangiano se utiliza para los sistemas con fuerzas conservativas incluidas en la energía potencial como la fuerza de gravedad. El Lagrangiano puede ser representado por un modelo matricial descrito como:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = F \tag{5.5}$$

donde $M(q)$ es la matriz de inercia, $C(q,\dot{q})$ son las fuerzas centrifugas y de Coriolis y $G(q)$ el vector de gravedad [4]. Para la obtención de las ecuaciones dinámicas del sistema, se recomienda seguir los siguientes pasos:

1. Definir el vector de coordenadas generalizadas q.
2. Calcular la energía cinética $\mathcal{K} = (q,\dot{q})$.
3. Calcular la energía potencial $\mathcal{P}(q)$.
4. Calcular el Lagrangiano: $\mathcal{L}(q,\dot{q}) = \mathcal{K}(q,\dot{q}) - \mathcal{L}(q)$
5. Calcular las ecuaciones de movimiento Euler-Lagrange (5.4).
6. Representar el modelo en la forma matricial (5.5).

5.3 Control proporcional-integral-derivativo

La combinación de la acción de control proporcional, la acción de control integral y la acción de control derivativa se denomina acción de control proporcional-integral-derivativa. La ecuación de un controlador con esta acción combinada está dada por:

$$u(t) = K_p e(t) + \frac{K_p}{T_i}\int_0^t e(t)dt + K_p T_d \frac{de(t)}{dt} \tag{5.6}$$

donde K_p es la ganancia proporcional, T_i es el tiempo integral y T_d es el tiempo derivativo. La utilidad de los controles PID estriba en que se aplican en forma casi general a la mayoría de los sistemas de control.

Un PID, acrónimo de proporcional, integral y derivativo, en un mecanismo de control que se encarga de aplicar una acción correctora para disminuir la diferencia entre la salida de un sistema y la referencia que ésta debería alcanzar, es decir, su error [5]. Un PID, como indica su nombre, actúa en tres niveles diferentes [3]:

- PROPORCIONAL: Determina la reacción del error actual.
- INTEGRAL: Corrige la integral del error para reducirlo a cero (error estacionario).
- DERIVATIVO: Determina la reacción del tiempo (derivada) en el que el error se produce.

A continuación, se estudia el problema de seleccionar las ganancias del controlador PID de manera que la respuesta en lazo cerrado tenga las características deseadas de respuesta transitoria, para eso se presenta el método heurístico [6].

5.4 Sintonización de control PID por medio del método heurístico

A principios de la década de los 90's, los algoritmos heurísticos se implementaron por primera vez como un medio alternativo de ajuste de controladores, empleándolos para determinar estimaciones iniciales de los parámetros que constituyen los controladores tipo PID. El control proporcional, integral y derivativo consiste en la suma de tres acciones de control en uno solo: a) Control Proporcional, b) Control Integral y c) Control Derivativo. En el Control Proporcional la salida del controlador utiliza una "proporción" del error del sistema para su acción. Su desventaja radica en introducir un error de desviación en el sistema debido al tiempo que tarda en calcular y aplicar la señal de control. Con respecto al Control Integral, la salida del controlador es proporcional a la cantidad de tiempo que existe en el error presente en el sistema.

La acción integral elimina el desfase introducido por el control proporcional, pero introduce un retardo de fase en el sistema debido a tiempo que tarda en establecerse el sistema. Mediante el Control Derivativo la salida del controlador es proporcional a la velocidad de cambio del error (de/dt), en otras palabras, la derivada se utiliza para elimina el desfase introducido por la acción integral (se compensan uno con respecto al otro); así, la suma de las tres acciones de control sumados en uno solo conforman un controlador robusto de tal manera que es eficiente para conseguir el efecto deseado de un sistema; dicho de otra forma, logra reducir e incluso eliminar el error [6].

5.5 Planta Péndulo invertido sobre dos ruedas

A pesar de su estructura simple, el péndulo invertido se considera el punto de referencia más fundamental en cuanto a los sistemas de ingeniería. Para este sistema, existen diferentes versiones, ofreciendo una variedad de interesantes desafíos de control. Los tipos más familiares del péndulo invertido son el péndulo rotacional de un solo brazo, el péndulo invertido sobre dos ruedas y el péndulo invertido doble. Las versiones menos comunes son el péndulo rotacional de dos enlaces, el péndulo invertido dual de tipo paralelo, el péndulo triple invertido, el péndulo invertido cuádruple y el péndulo 3D o esférico [7].

El sistema péndulo invertido sobre dos ruedas es un sistema dinámico inherentemente inestable con un comportamiento altamente no lineal y por su semejanza a otros problemas de la ingeniería de control como son los controladores de brazos robóticos, lanzadores de cohetes espaciales,

controladores de vehículos aéreos para despegue y aterrizaje vertical por mencionar algunos, es considerado como uno de los problemas clásicos más importantes en la teoría de control, siendo de gran interés en su investigación al ser utilizado como un banco de pruebas para la evaluación de una amplia gama de métodos y estrategias de control lineal y no lineal.

El péndulo invertido sobre dos ruedas es un sistema electromecánico inestable y no lineal que consta de dos ruedas laterales que tienen contacto con la superficie del suelo. Las ruedas se mueven de forma independiente para equilibrarse en el centro de gravedad por encima del eje de rotación de las ruedas.

Las ruedas son accionadas por dos motores acoplados a cada uno de ellos. Los motores pueden ser de tipo DC y son controlados por señales eléctricas mediante un sistema de control basado en la lectura de inclinación y la velocidad de su centro de gravedad.

El objetivo del control es estabilizar el cuerpo del sistema manteniéndolo en posición de equilibrio vertical, como se muestra en la Figura 5.2. La compleja dinámica inherente a esta plataforma encuentra su aplicación en el diseño y desarrollo de sistemas de control para automóviles, naves espaciales, transporte nacional, transporte militar, entre otros [8].

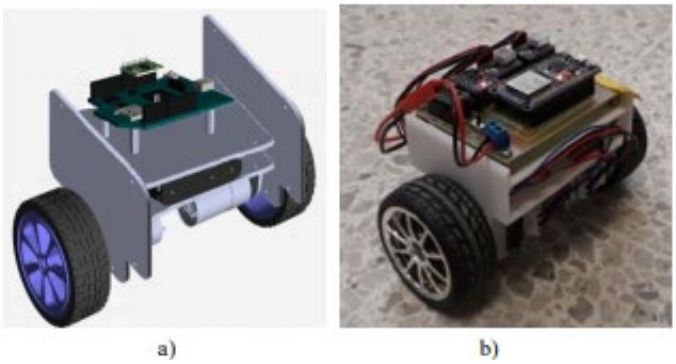


Figura 5.2- Péndulo invertido sobre dos ruedas. a) Diseño, b) Implementación.

La Figura 5.3 muestra las vista lateral y frontal de la idea básica de un péndulo invertido de sobre dos ruedas:

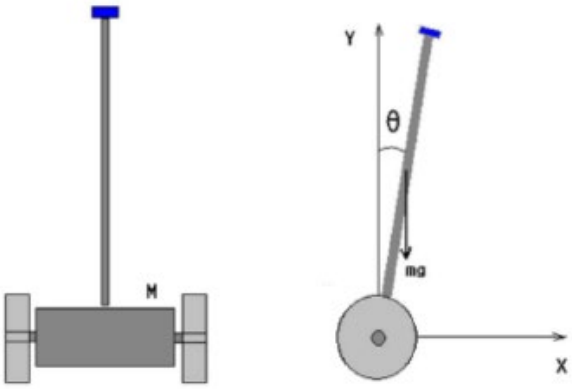


Figura 5.3- Vista lateral y frontal del sistema ideal de péndulo invertido de dos ruedas.

Para equilibrar el péndulo se requiere un ángulo de inclinación cero, es decir, $\theta = 0$. Una vez que el sistema está equilibrado, se le puede ordenar que se mueva hacia adelante, hacia atrás, gire a la derecha o a la izquierda. El sistema consta de tres partes principales [9]:

1. LAS RUEDAS: mueve el sistema hacia atrás o hacia adelante para equilibrar el cuerpo del sistema.
2. EL CHASIS: sostiene los motores, circuitos y cualquier pieza requerida para el sistema.
3. EL PÉNDULO: las partes del sistema que causan la inestabilidad y necesita ser estabilizado. Esta parte puede estar oculta dentro del chasis.

6. Métodos y materiales

6.1 Sujetos

Este proyecto se realiza utilizando la planta péndulo invertido sobre dos ruedas (de bajo presupuesto) realizada por los integrantes del equipo y asesor del proyecto, pertenecientes al Instituto Tecnológico de Matamoros. El prototipo resultante será utilizado para futuras prácticas relacionadas con cursos afines al Control Automático y está dirigido a:

- Estudiantes y profesores del Instituto Tecnológico de Matamoros.
- La comunidad científica en el área de control e ingenierías afines.
- Estudiantes de licenciatura en las áreas de control y mecatrónica.

6.2 Materiales y presupuesto

Para la realización del prototipo funcional, se utilizaron materiales de bajo costo y fácil acceso. Los materiales utilizados y su precio aproximado en el mercado se muestran en la **Tabla 6.1.** La mayoría de los componentes electrónicos se obtuvieron en la página de compra en línea Mercado Libre. Cabe aclarar que los precios van de la mano con distintos factores como: zona geográfica, inflación, escasez de producto, etc. Por lo tanto, es muy probable que existan variaciones de precios en el mercado.

El presupuesto necesario para realizar un prototipo funcional (con los materiales mencionados anteriormente) es de aproximadamente: **2,722.28 PESOS MEXICANOS** (ver Tabla 6.1).

Si se hace una comparación del precio aproximado de una planta funcional ofertada en el mercado contra la realizada en el presente proyecto, es notable la gran diferencia de precio y accesibilidad para un estudiante o investigador promedio en México.

Para hacer una comparación real se realizó una cotización de un kit completo de una planta péndulo lineal con empresas dedicadas a la distribución de equipos de laboratorio de ingeniería. Como en todo, se ofertan plantas sencillas desde precios aproximados de **$5,000 DOLARES CANADIENSES**, es decir, aproximadamente **$78,838.54 PESOS MEXICANOS.**

Si se busca un kit completo de planta péndulo invertido, se oferta en aproximadamente **$13,600 DÓLARES CANADIENSES**, es decir, cerca de **$214,440.83 PESOS MEXICANOS.**

Al comparar el costo de planta construida en el presente proyecto (2,722.28 PESOS MEXICANOS) vs las plantas de péndulo invertido ofertadas en el mercado, es preciso notar que se logró una reducción de costos de aproximadamente un 95 %.

Cabe aclarar que estos precios pueden variar según: la empresa donde la planta se adquiera, región, divisa o año de consulta. Aunque al adquirir una planta ofertada en el mercado se obtienen más beneficios (garantías, materiales de mejor calidad, softwares y manuales de usuario especializados, prácticas previamente preparadas, etc.), no se puede ignorar que para muchos estudiantes es imposible

adquirir alguna por sus elevados precios; por lo tanto, esta puede ser una excelente opción para aplicar la teoría aprendida en los cursos afines a Control Automático o ejercer investigación sobre ella.

Tabla 6.1. Presupuesto para compra de materiales del prototipo.

MATERIAL	UNIDAD	PRESENTACIÓN	PRECIO	CANTIDAD	TOTAL
Tablón de madera (¼'' grosor, 5 ¼ '' longitud, 2 ¾'' anchura)	1	Objeto	50.00	1	50.00
Varilla atornillada (ø¼'')	1	Objeto	8.34	2	16.68
Tuerca (hexagonal de ¼'')	1	Paquete	0.45	32	14.40
Arduino UNO	1	Objeto	124.00	1	124.00
Motorreductor Pololu	1	Objeto	620.00	2	1,240.00
Llantas	2	Paquete	70.00	1	70.00
Brackets para montar motorreductores	2	Paquete	230.00	1	230.00
Modulo L298N Puente H	1	Objeto	50.00	1	50.00
Protoboard	1	Objeto	50.00	1	50.00
Cables jumper para Arduino	65	Paquete	39.00	1	39.00
Acelerómetro MPU 6050	1	Objeto	35.20	1	35.20
Batería de 9 V	1	Objeto	60.00	1	60.00
Conector de batería 9V con plug para Arduino	1	Objeto	28.00	1	28.00
Batería lipo	1	Objeto	400.00	1	400.00
Balanceador de carga para batería lipo	1	Objeto	295.00	1	295.00
Cinchos	20	Paquete	20.00	1	20.00
TOTAL				**2,722.28 MXN**	

Además de los materiales utilizados para la elaboración de la planta, se utilizaron las siguientes herramientas para su construcción; y la simulación e implementación del controlador:

- Equipo de cómputo con la versión de Windows 10
- Software IDE Arduino 1.6.12
- Artículos científicos IEEE.
- Pistola para barras de silicón
- Taladro
- Cautín y soldadura
- Kit de desarmadores
- Segueta

6.3 Metodología

6.3.1 PRUEBA PRELIMINAR DE LOS ACTUADORES Y SENSOR DEL SISTEMA

Como primer paso se procede a probar los actuadores del sistema. Para esto, los motorreductores se alimentan con la batería de 12 V y se conectan al puente H cómo se muestra en el diagrama mostrado en la Figura 6.1. Una vez conectados, se prueba su funcionalidad.

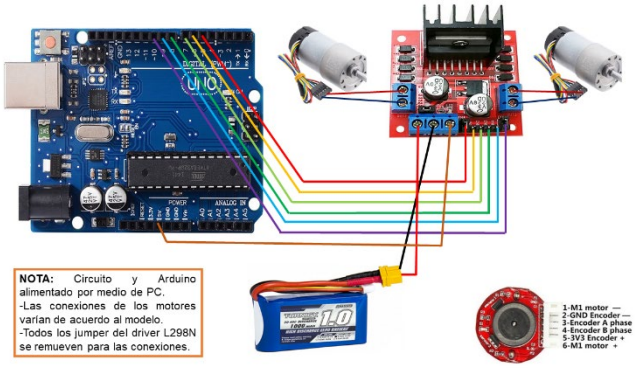


Figura 6.1. Armado de Circuito para pruebas de motorreductores.

Una vez probados los motorreductores, se procede a probar la funcionalidad del sensor del sistema, el acelerómetro MPU 6050. Para esto, se agregará al diagrama mostrado en la Figura 3.1; realizando las conexiones que se muestran en la Figura 6.2.

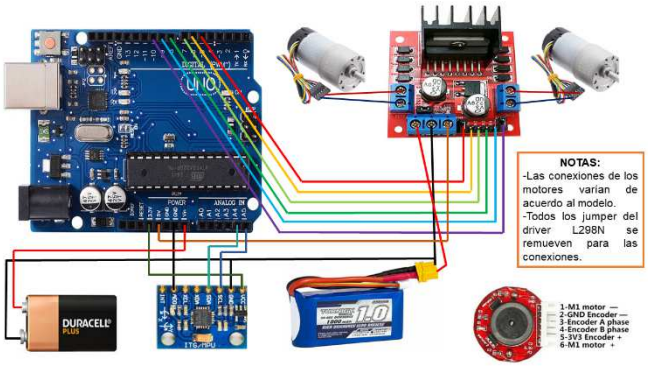


Figura 6.2. Armado de Circuito para prueba de acelerómetro MPU 6050.

6.3.2 CONSTRUCCIÓN DEL PROTOTIPO PÉNDULO INVERTIDO SOBRE DOS RUEDAS

Una vez probados los actuadores y sensor del sistema, se procede a construir el chasis de la planta. Para la realización del presente proyecto se utiliza el diseño que se muestra en la Figura 6.3. Cabe mencionar que el diseño puede ser alterado de acuerdo con las necesidades de cada proyecto.

Figura 6.3. Chasis de la planta péndulo invertido sobre dos ruedas.

Una vez armado el chasis, se procede a realizar el acomodo del circuito mostrado en la Figura 6.2 distribuyéndolo a lo largo de los pisos de la planta.

6.3.3 MODELADO NO LINEAL DEL SISTEMA

Se procede con el modelado de la planta péndulo invertido sobre dos ruedas mediante la metodología de Euler-Lagrange; en la cual se identifican las fuerzas generalizadas que actúan sobre la planta y a partir de sus ecuaciones de energía cinética y potencial se desprende su modelo dinámico. También se integra el modelo dinámico del servomotor para conseguir un sistema más completo.

6.3.4 DISEÑO E IMPLEMENTACIÓN DE CONTROLADOR PID

En esta etapa, se propone utilizar un controlador PID para controlar al sistema. Este controlador es de fácil implementación ya que sus ganancias pueden ser sintonizadas a través de método heurístico. Cabe destacar que la aplicación del controlador PID no requiere el conocimiento completo del modelo de la planta, lo que lo hace un control muy versátil y de fácil implementación. Sin embargo, la aplicación de otros esquemas de control más complejos y de mejores características de desempeño, requiere el conocimiento de ciertos parámetros del sistema

6.3.5 EXPERIMENTACIÓN CON EL CONTROLADOR PID DISEÑADO ANTERIORMENTE

Una vez obtenido un PID satisfactorio, se realizan pruebas experimentales que validen el buen desempeño y efectividad del controlador. Si se obtienen resultados adecuados o inmejorables, se procede a la etapa de documentación, en caso contrario, se manipulan las ganancias del controlador y reinicia la prueba de funcionalidad.

Diseño Colaborativo en Mecatrónica, Capítulo 17, pp. 227 - 245
ISBN: 978-607-9394-23-3, 2021

7. Desarrollo

Una vez armado la planta Péndulo Invertido sobre dos ruedas se procede a determinar el modelo no lineal del mismo utilizando la metodología Euler-Lagrange. La planta construida se muestra en la Figura 7.1.

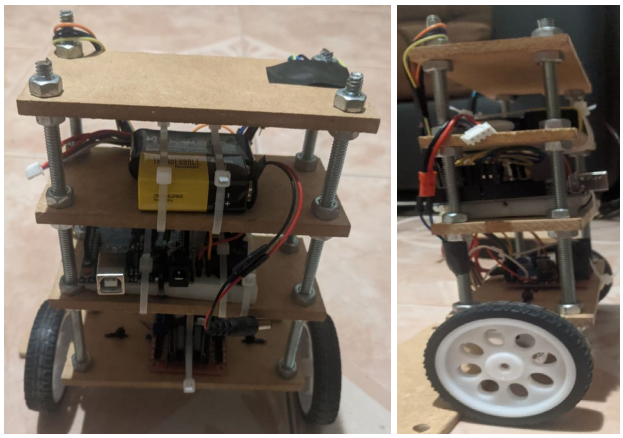


Figura 7.1. Diseño final del prototipo de la planta péndulo invertido sobre dos ruedas.

7.1 Modelado de la planta Péndulo Invertido sobre dos ruedas

El subsistema mecánico consiste en el cuerpo del péndulo invertido y dos ruedas. El cuerpo puede ser modelado como un PÉNDULO INVERTIDO con la masa concentrada en el centro de gravedad y eje de rotación sobre el eje de las ruedas (Figura 7.2).
Para la derivación de ecuaciones de movimiento se usa un modelo plano donde el robot se mueve a lo largo de un eje horizontal. Se supone que no hay deslizamiento entre las ruedas y el suelo [10]. Las ecuaciones de Lagrange de segundo orden se utilizan para modelar un subsistema mecánico [4].

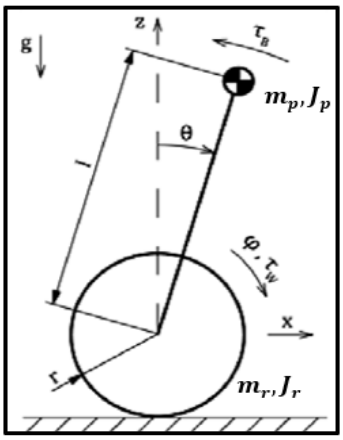


Figura 7.2. Diagrama de péndulo invertido sobre dos ruedas.

Donde:
Θ= Ángulo del péndulo sobre la vertical
φ = Ángulo de la rueda
x= Posición de la rueda
l_{cm}= Distancia del centro de gravedad del péndulo al eje de giro
J_p= Momento de inercia del péndulo
J_r= Momento de inercia de la rueda
m_p= Masa del péndulo
m_r = Masa de la rueda

Se utilizarán dos coordenadas generalizadas $\{\varphi,\theta\}$. A su vez, la distancia x se puede obtener a partir de φ con la siguiente relación:

$$\varphi = \frac{\dot{x}}{r} \tag{7.1}$$

Como primer paso es necesario determinar el Lagrangiano de la planta. Por lo tanto:

$$\mathcal{L} = \mathcal{K} - \mathcal{P} \tag{7.2}$$

Donde:
$\mathcal{L}$ = Lagrangiano del sistema"
$\mathcal{K}$ = Energía cinética de la planta
$\mathcal{P}$ =Energía potencial de la planta

La energía cinética de un cuerpo se representa por:

$$K = \sum_{i}^{n} \frac{1}{2} m_i V_i^2 + \frac{1}{2} J_i \omega_i^2 \tag{7.3}$$

Donde:
K= Energía cinética de un cuerpo
n= Cantidad de coordenadas generalizadas
m_i= Masa del i-ésimo cuerpo en movimiento
V_i= i-ésima velocidad lineal
J_i= Momento de inercia del i-ésimo cuerpo en movimiento
ω_i= i-ésima velocidad angular

Para el cálculo de la energía cinética total de la planta se consideran las energías cinéticas provocadas por péndulo (cuerpo del péndulo invertido) y sus dos ruedas, obteniendo lo siguiente:

$$K = \frac{1}{2} m_p \left(\dot{x}^2 + 2\dot{x}\dot{\theta} l_{cm} \cos\theta + {l_{cm}}^2 \dot{\theta}^2 \right) + \frac{1}{2} J_p \dot{\theta}^2 + m_r \dot{x}^2 + \frac{J_r}{r^2} \dot{x}^2 \tag{7.4}$$

Donde:

K_p= Energía cinética del péndulo
m_p= Masa del péndulo
x= Posición del péndulo sobre la horizontal
Θ= Inclinación del péndulo sobre la vertical
l_{cm}= Distancia del centro de gravedad del péndulo al eje de giro
J_p= Inercia del péndulo
K_r= Energía cinética de la rueda

m_r= Masa de la rueda
r= Radio de la rueda
J_r= Inercia de la rueda

Una vez obtenida la energía cinética de la planta, se procede a determinar su energía potencial, la cual se obtiene mediante la ecuación:

$$P = mgh \tag{7.5}$$

Donde:
P=Energía potencial
m=Masa del cuerpo
g=Constante de gravedad
h=La distancia respecto a la referencia

Por lo tanto, la energía potencial de la planta es la siguiente:

$$P = m_p g l_{cm} \cos\theta \tag{7.6}$$

Una vez obtenidas las energías cinética y potencial del sistema, y la energía disipada de las ruedas, se puede obtener su Lagrangiano:

$$\mathcal{L} = \frac{1}{2} m_p \left(\dot{x}^2 + 2\dot{x}\dot{\theta} l_{cm} \cos\theta + l_{cm}{}^2 \dot{\theta}^2 \right) + \frac{1}{2} J_p \dot{\theta}^2 + \frac{1}{2} m_r \dot{x}^2 + \frac{1}{2} \frac{J_r}{r^2} \dot{x}^2 - m_p g l_{cm} \cos\theta \tag{7.7}$$

Las ecuaciones de segundo orden de Lagrange se definen como:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = \sum_{\alpha} \vec{F}_{\alpha} \frac{\partial \vec{x}_{\alpha}}{\partial q_i} + \sum_{\beta} \tau_{\beta} \frac{\partial \theta_{\beta}}{\partial q_i} \tag{7.8}$$

Donde:
q_i = Posición angular o lineal
$\dot{q}_i$ = Velocidad angular o lineal
F= Fuerzas del sistema
τ = Torques del sistema
x= Posición del péndulo sobre la horizontal
Θ= Inclinación del péndulo sobre la horizontal

Una vez obtenido el Langrangiano, también se analizan las fuerzas generalizadas involucradas en el sistema que se muestran en la Figura 7.3. Esto se hace para tener todas las ecuaciones de Lagrange y obtener un modelo más completo.

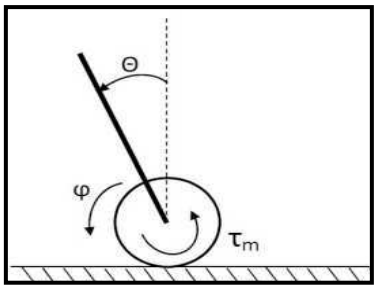


Figura 7.3. Diagrama de péndulo invertido sobre dos ruedas

Obteniendo lo siguiente:

$$\left(m_p + m_r + \frac{J_r}{r^2}\right)\ddot{x} + m_p l_{cm}\left(\ddot{\theta}\cos\theta - \dot{\theta}^2 sen\theta\right) = \frac{\tau_m}{r} - \dot{x}\left(\frac{\mu_e}{r^2} + \mu_s\right) + \frac{\mu_e}{r}\dot{\theta} \tag{7.9}$$

$$m_p \ddot{x} l_{cm}\cos\theta + \left(m_p l_{cm}{}^2 + J_p\right)\ddot{\theta} - m_p g l_{cm} sen\theta = -\tau_m + \mu_e\left(\frac{\dot{x}}{r} - \dot{\theta}\right) \tag{7.10}$$

Asignando variables a las constantes para obtener nuevos parámetros, se obtiene:

$$\begin{aligned} A &= m_p + m_r + \frac{J_r}{r^2} \\ M &= m_p l_{cm}{}^2 + J_p \end{aligned} \tag{7.11}$$

Por lo tanto, las ecuaciones no lineales que representan la planta son las siguientes:

$$\ddot{x} = -\frac{m_p l_{cm}}{A}\left(\ddot{\theta}\cos\theta - \dot{\theta}^2 sen\theta\right) + \frac{\tau_m}{Ar} - \dot{x}\frac{\left(\frac{\mu_e}{r^2} + \mu_s\right)}{A} + \frac{\mu_e}{Ar}\dot{\theta} \tag{7.12}$$

$$\ddot{\theta} = -\frac{m_p l_{cm}}{M}\ddot{x}\cos\theta + \frac{m_p g l_{cm}}{M} sen\theta - \frac{\tau_m}{M} + \frac{\mu_e}{M}\left(\frac{\dot{x}}{r} - \dot{\theta}\right) \tag{7.13}$$

Donde:
τ_m =Torque del motor
$\dot{\theta}$ = Velocidad angular del péndulo
$\ddot{\theta}$ =Aceleración angular del péndulo
φ = Ángulo de la rueda del péndulo
$\dot{x}$ = Velocidad de la rueda
$\ddot{x}$ = Aceleración de la rueda
μ_e=Coeficiente de fuerza viscosa en el eje
μ_s=Coeficiente de fricción en el suelo

El sistema también consta de motorreductores CD (Figura 7.4) que se utilizan como actuadores del robot. El subsistema proporciona directamente un movimiento giratorio y, junto con las ruedas, permite que el robot realice movimientos. El motor tiene una resistencia R_m e inductancia L_m. La corriente que circula por el circuito es I_m, la tensión aplicada en bornes del motor es V_{in} y su velocidad de giro es de $\dot{\theta}_m$. La fuerza contraelectromotriz ejercida por el motor es e [10].

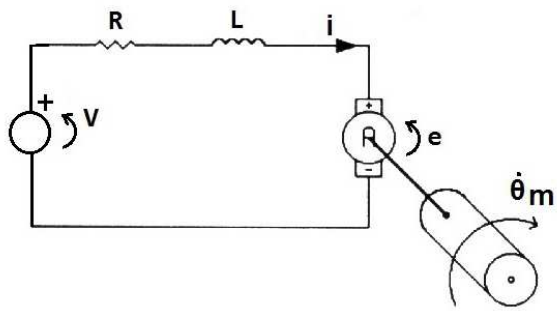


Figura 7.4. Esquema del modelo eléctrico del motor

Por lo tanto, al agregar el modelo resultante del motorreductor a las ecuaciones no lineales de la planta obtenidas anteriormente ((7.12) y (7.13)), el resultado es el siguiente:

$$\ddot{x} = -\frac{m_p l_{cm}}{A}\left(\ddot{\theta}\cos\theta - \dot{\theta}^2 sen\theta\right) + \frac{1}{Ar}\left(\frac{K_m K_s}{R_m}u - \frac{K_m K_b}{R_m}\left(r\dot{x} - \dot{\theta}\right)\right) - \dot{x}\frac{\left(\frac{\mu_e}{r^2} + \mu_s\right)}{A} + \frac{\mu_e}{Ar}\dot{\theta} \tag{7.14}$$

$$\ddot{\theta} = -\frac{m_p l_{cm}}{M}\ddot{x}\cos\theta + \frac{m_p g l_{cm}}{M}sen\theta - \frac{1}{M}\left(\frac{K_m K_s}{R_m}u - \frac{K_m K_b}{R_m}\left(r\dot{x} - \dot{\theta}\right)\right) + \frac{\mu_e}{M}\left(\frac{\dot{x}}{r} - \dot{\theta}\right) \tag{7.15}$$

Donde:
V_{in}=Tensión aplicada en los bornes del motor
R_m=Resistencia de armadura
K_b=Constante de fuerza electromotriz
K_m=Constante de torque/par del motor
K_g= Relación de transmisión
B_m=Coeficiente de fricción viscosa
K_s=Constante de relación entre acción de control y tensión aplicada

Si simplificamos y agrupamos términos, **el modelo completo de la planta péndulo invertido sobre dos ruedas** queda expresado de la siguiente forma:

$$\ddot{x} = -\frac{m_p l_{cm}}{A}\left(\ddot{\theta}\cos\theta - \dot{\theta}^2 sen\theta\right) + K_1 u - K_2\dot{x} + \left(\frac{\mu_e}{Ar} + \frac{K_m K_b}{R_m Ar}\right)\dot{\theta} \tag{7.16}$$

$$\ddot{\theta} = -\frac{m_p l_{cm}}{M}\ddot{x}\cos\theta + K_3 sen\theta - K_4 u - K_5\dot{\theta} + \left(\frac{K_m K_b r}{R_m M} + \frac{\mu_e}{Mr}\right)\dot{x} \tag{7.17}$$

$$K_1 = \frac{K_m K_s}{R_m A r}$$
$$K_2 = \frac{\frac{\mu_e}{r^2} + \mu_s + \frac{K_m K_b}{R_m}}{A}$$
$$K_3 = \frac{m_p g l_{cm}}{M} \tag{7.18}$$
$$K_4 = \frac{K_m K_s}{R_m M}$$
$$K_5 = \frac{K_m K_b}{R_m M} + \frac{\mu_e}{M}$$

Donde:
m_p= Masa del péndulo
x= Posición del péndulo sobre la horizontal
Θ= Inclinación del péndulo sobre la vertical
l_{cm}= Distancia del centro de gravedad del péndulo al eje de giro
J_p= Inercia del péndulo
m_r= Masa de la rueda
r= Radio de la rueda
J_r= Inercia de la rueda
$\dot{\theta}$ = Velocidad angular del péndulo
$\ddot{\theta}$ =Aceleración angular del péndulo
$\dot{x}$ = Velocidad de la rueda
$\ddot{x}$ = Aceleración de la rueda
μ_e=Coeficiente de fuerza viscosa en el eje
μ_s=Coeficiente de fricción en el suelo
V_{in}=Tensión aplicada en los bornes del motor
R_m=Resistencia de armadura
K_b=Constante de fuerza electromotriz
K_m=Constante de torque/par del motor
K_s=Constante de relación entre acción de control y tensión aplicada

Los parámetros utilizados en las ecuaciones (7.16) y (7.17) se presentan en el apartado 8.4 del presente artículo.

8. Resultados

8.1 Prototipo funcional de una planta Péndulo Invertido sobre dos ruedas.

Como resultado principal del presente proyecto, se obtuvo un prototipo funcional de un péndulo invertido sobre dos ruedas siendo capaz de equilibrarse aun cuando se ve expuesto a perturbaciones externas. Esto lo logra teniendo pequeños movimientos hacia enfrente y atrás para mantenerse en equilibrio y no caer aún se lleguen a presentar algunas perturbaciones externas. Debido a los movimientos realizados por la planta, esta debe de tener un área de trabajo no tan amplia, pero si libre para poder realizar los movimientos que sean adecuados para mantener el equilibrio y no caer.

Este prototipo también es capaz de transportar un vaso con agua (como perturbación al modelo) sin derramar el líquido, teniendo solo unas pequeñas variaciones entre el funcionamiento con vaso y sin vaso, ya que, al agregar el peso extra de este objeto, el carro debe de realizar más esfuerzo y utilizar un área más amplia para realizar sus movimientos (ver Figura 8.1).

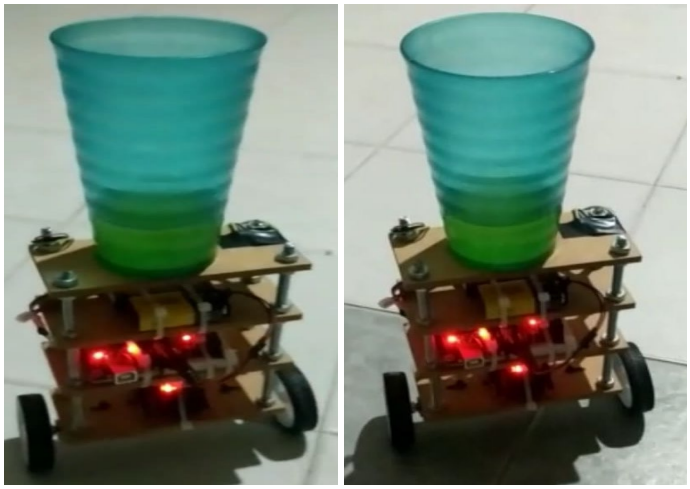


Figura 8.1. Planta transportando un vaso con agua.

Por lo tanto, el resultado final de la construcción del prototipo funcional se muestra en las Figura 8.2.

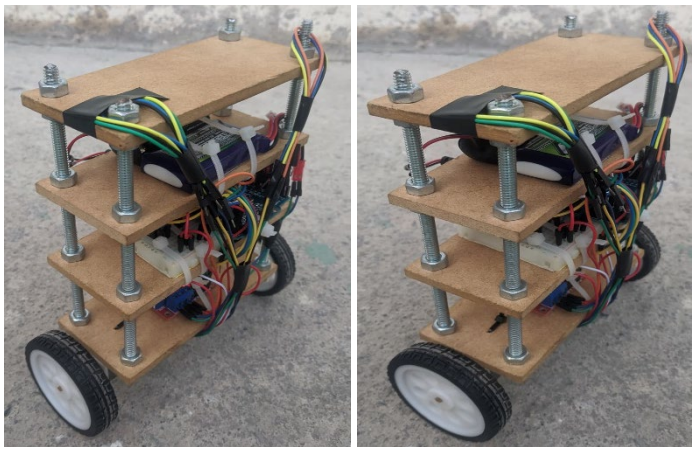


Figura 8.2. Planta péndulo invertido sobre dos ruedas equilibrado

8.2 Modelado de la planta Péndulo Invertido sobre dos ruedas

Como resultado final se obtuvo que el modelado matemático completo de la planta está descrito por las ecuaciones no lineales (7.16) y (7.17) (ver apartado 7.1).

8.3 Ganancias del controlador PID

El programa realizado permite implementar de la manera más sencilla y completa los componentes adquiridos (ver apartado 6.2) utilizando librerías (por ejemplo, la librería PID_v1.h) para obtener un óptimo funcionamiento; logrando así la implementación del controlador PID a la planta para cumplir con el objetivo principal de este proyecto: controlar la planta y lograr que se equilibre declarando las ganancias K_p, K_i, K_d en el programa realizado. Como se mencionó anteriormente, se realizan pruebas de manera experimental para encontrar los valores más adecuados que den como resultado un controlador eficiente. Por lo tanto, los valores idóneos resultaron ser los siguientes:

8.1 Tabla de valores para el controlador PID.

GANANCIA	VALOR
K_p	68
K_i	340
K_d	3.4

8.4 Parámetros de la planta Péndulo Invertido sobre dos ruedas

A continuación, la Tabla 8.2 presenta los valores de los parámetros utilizados en las ecuaciones no lineales ((7.16) y (7.17)) resultantes del modelado matemático (ver sección 7.1) del Péndulo invertido sobre dos ruedas.

8.2 Parámetros de la planta Péndulo Invertido sobre dos ruedas.

PARÁMETRO	DESCRIPCIÓN	VALOR	UNIDADES
m_p	Masa del péndulo	0.740	kg
m_r	Masa de la rueda	0.030	kg
l_{cm}	Distancia del centro de gravedad del péndulo al eje de giro	0.061	m
r	Radio de la rueda	0.036	m

Para cubrir en su totalidad los parámetros del modelo y poder realizar simulaciones, aun es necesario obtener los parámetros del motor con alguna metodología existente en la literatura. Por otro lado, el momento de inercia de las ruedas y del péndulo, por su complejidad para ser calculado, sería necesaria la utilización de algún software de modelado mecánico en 3D como Solidworks o AutoCAD.

9. Conclusiones

Este proyecto se realizó con la finalidad de obtener un sistema péndulo invertido sobre dos ruedas, que consiste en un carro con dos llantas que se mantiene en pie sin caer, balanceándose hacia adelante y atrás. Esto lo logra gracias a la ayuda de los componentes utilizados: una tarjeta Arduino UNO, un par de motores, un driver puente H para controlar los motores, y un dispositivo que mide la velocidad angular y aceleración (giroscopio y acelerómetro). Estos dispositivos trabajan en conjunto para poder realizar el objetivo de mantener la planta balanceada.

Al concluir todas las etapas del presente proyecto, sí se obtuvo una planta estable y con las características necesarias para realizar experimentación con ella. Para comprobar que sí se balanceara de manera óptima, se colocó un vaso con agua encima del chasis. Como resultado se obtuvo que el péndulo fue capaz de mantenerse estable sin derramar el agua, por lo tanto, los resultados finales obtenidos fueron satisfactorios, reproducibles y repetibles, así que consideramos que sí se cumplió el objetivo principal. Dicho lo anterior, cabe mencionar que se pueden realizar algunas mejoras que se proponen a continuación.

Se recomienda la implementación de los enconder que tienen los motores, ya que debido a los inconvenientes de tiempo fue difícil implementarlos al sistema; pero, si se desean implementar se podría lograr el balance total de la planta en un estado estacionario sin presentar movimientos o un resultado más favorable. Lo anterior sería recomendable utilizando otro microcontrolador diferente al Arduino, con características de tiempo de muestreo más pequeño y lectura de encoder independiente.

También hay que tomar en cuenta que NO debe realizarse con ningún material frágil; por ejemplo, al inicio se utilizó acrílico para el chasis de la planta, pero al final se elaboró en madera, ya que al tratar de atornillar los brackets al acrílico, este se fisuró; además que se corría gran riesgo de fractura al momento de realizar las pruebas experimentales con el controlador por las constantes caídas que sufre la planta.

Para obtener un mejor resultado también podrían realizarse simulaciones o diseños analíticos del controlador PID utilizando la técnica que mejor se acople al proyecto. En este caso, podría utilizarse el modelado obtenido con la metodología Euler-Lagrange, linealizarlo y utilizar Simulink/MATLAB para simular antes de implementar el controlador.

Referencias

[1] Hernández-Guzmán, V.M., Silva-Ortigoza, R. and Carrillo-Serrano, R.V. *Control Automático: Teoría de diseño, construcción de prototipos, modelado, identificación y pruebas experimentales*. COLECCIÓN CIDETEC, 2013.
[2] Ogata, K. *Ingeniería de control moderna*. Pearson Educación. 2003.
[3] Hernández Largacha, A., Legaspi Martínez, M. and Peláez Martín, J. *Control inteligente del péndulo invertido*. 2012.
[4] Kelly, R. and Santibáñez, V. *Control de movimiento de robots manipuladores*. Pearson educación. 2003.
[5] Kuo, B.C., 1987. *Automatic control systems*. Prentice Hall PTR. 1992
[6] Ambriz Polo, Juan Carlos. *Diseño e Implementación de un Controlador PID Autoajustable basado en un algoritmo híbrido* (Tesis de posgrado). Centro Universitario UAEM Atlacomulco, Atlacomulco, Estado de México, México. 2015.
[7] Boubaker, O. *The Inverted Pendulum Benchmark in Nonlinear Control Theory: A Survey. International Journal of Advanced Robotic Systems*. 2013. https://doi.org/10.5772/55058
[8] Jiménez Fabián, Ruge Ilber, Jiménez Andrés. *Modeling and Control of a Two Wheeled Self-Balancing Robot: a didactic platform for control engineering education.* En 18th LACCEI International Multi-Conference for Engineering, Education and Technology. Latin American and Caribbean Consortium of Engineering Institutions (LACCEI), Virtual. 2020.
[9] Khaled, Mahmoud & Mohammed, Ali & Ibraheem, M.Shaaban & Ali, Reda. *Balancing a Two Wheeled Robot.* 10.13140/RG.2.2.25634.63683. 2009.
[10] Pincin Santiago Javier. *Diseño, construcción y control de un robot balancín* (Tesis de pregrado). Universidad Nacional de Cuyo, Río Negro, Argentina. 2016.

Diseño Colaborativo en Mecatrónica, Capítulo 18, pp. 246 - 264
ISBN: 978-607-9394-23-3, 2021

GIGA5 Vaccine: Un software para la vacunación masiva en todas las etapas de la vida.

Navarro Álvarez, Thiago Andrés.
Navarro Parra, Sandra Liliana.

Fundación Universitaria Navarra - UNINAVARRA
thiago.navarro@uninavarra.edu.co

Resumen

El presente artículo busca elaborar una discusión reflexiva en torno al registro y gestión de información en el proceso de vacunación de usuarios durante todo el periodo de la vida, situación apremiante en el marco de la pandemia mundial de la COVID-19 (Virus SARS-CoV-2). De igual forma, el presente documento, sitúa la importancia de construir herramientas que permitan la recopilación y el manejo de datos, como un mecanismo para fortalecer los procesos de gestión y planeación del sistema de salud. Es por esto, que se propone la creación del ***Sistema de vacunación inteligente GIGA5 Vaccine Center,*** *compuesto de dos aplicativos para Usuarios y Profesionales de la salud, con los que se busca fortalecer el sistema de carnetización y seguimiento a los pacientes, toda vez que presenta un modelo de gestión de datos, que pretende optimizar el procedimiento y acelerar la trazabilidad de la información asociada a la vacunación, en todas las etapas de la vida, por medios de software especializados.*

Palabras clave: Programa de vacunación, software, App, gestión de datos, carné vacunación.

1. Introducción

La salud es un tema que desde los inicios del hombre ha estado relacionada con la investigación. Siendo esta la fuente de exploración donde se toma como eje principal el cuerpo humano. Se ha estudiado todo su funcionamiento, se ha analizado cada hueso, célula y órgano que lo compone. Además, ha dado paso para estudiar cada patología o virus. Es por ello, que el proceso de vacunación desempeña un papel fundamental dentro del plan de salud de cada persona. Se comprende que la vacunación es una suspensión de microorganismos vivos, inactivados o muertos, fracciones de estos o partículas proteicas que al ser administrados inducen una respuesta inmune que previene la enfermedad contra la que está dirigida. De los campos de mayor importancia en la promoción de la salud es la prevención de las enfermedades infecciosas mediante las vacunas.

El proceso de vacunación se encuentra acompañado de un registro, que en la mayoría de los casos se le denomina carné, el cual cumple la función de evidenciar el registro de vacunas realizadas a una persona desde su nacimiento hasta la actualidad. La información constatada en el documento permite a los trabajadores de la salud realizar los diferentes análisis e indicarle al paciente la actualización de una vacuna, al mismo tiempo indicar cuáles tiene y cuáles necesita. A ello, se le suma que el carné debe contener información general de identificación de cada paciente.

En estos momentos en Colombia el carné de vacunación es entregado inicialmente a la madre de manera física, ya que son los niños los que deben cumplir con un número de vacunas desde su nacimiento. Incluyen todas las actividades destinadas a la prevención, detección y tratamiento oportuno de enfermedades, en la cual se recibe información sobre el desarrollo normal, nutrición, sueño, seguridad, enfermedades infecciosas y otros temas importantes como lo que se puede esperar a medida que el niño crezca.

El carné se entrega en material de cartón plastificado. Debido a su manipulación con el tiempo se deteriora y la información registrada se puede perder. Sin dejar de un lado aquellos que por accidentes caseros dejan de existir y el proceso de vacunación queda nulo referente a la información registrada. Por tal razón, *GIGA5 innova* de la mano con la tecnología y propone la creación de una App desarrollada para plataformas Android. Su objetivo será funcionar como un "carné inteligente de vacunación" que permita a los usuarios tener información precisa sobre las vacunas que se ha aplicado, los refuerzos y con alcance a otros miembros de su círculo familiar para tener un control de estas. También contará con información de promoción y prevención enfocado al tema de la salud. Los usuarios que descarguen y completen el proceso de registro en la App quedarán registrados en una base de datos que se consultará por parte de los médicos y/o personal relevante para gestionar su servicio de vacunación.

El proyecto **GIGA5 Vaccine Center** llega para romper esquemas tradicionales y relacionar una vez más la salud y la tecnología. De este modo, se permite estar a la vanguardia y en la revolución de las aplicaciones móviles. Las cosas han cambiado drásticamente, hoy en día, se experimenta una gran transformación. Una de ellas, es que las apps tienen la característica de usar y tener bases de usuarios de tamaño limitado. Gracias a los dispositivos móviles y las apps cualquiera puede estar al tanto de todo lo que ocurra y estar conectado a él desde cualquier lugar en cualquier momento, aumentado de esta manera la eficiencia.

De igual forma, la propuesta **GIGA5 Vaccine Center**, fruto de la situación global de Pandemia que vivimos actualmente por la COVID-19 (Virus SARS-CoV-2), se sitúa como una opción pertinente para la gestión, conocimiento e información sobre los procesos de vacunación que se adelantan en el país, y que involucran directamente a los usuarios y personal médico, como actores partícipes del proceso de registro y seguimiento a sus procesos de vacunación. Hoy más que antes, la vacunación retoma un papel protagónico en el cuidado y proceso de salud de las personas, y alternativas como GIGA5 permite una optimización del proceso de comunicación entre sistema de salud y pacientes.

Bajo este panorama, el **sistema de vacunación inteligente GIGA5**, permite el acercamiento directo del paciente a su programa de vacunación durante toda la vida, al igual que en el marco de la pandemia y la vacunación contra la COVID-19, en tanto ofrece información detallada sobre las dosis aplicadas, tipos de vacunas, fechas de vacunación, momentos y espacios y oportunidades para futuras vacunas, al tiempo que otorga datos relevantes a las IPS, para la gestión oportuna de proceso de vacunación y acompañamiento a los pacientes en su proceso integral de salud.

También, el software que propone GIGA5 Vaccine, acompañado de una App para plataformas iOS y Android, funciona como un sistema de vacunación inteligente ante las industrias y entidades prestadoras de servicios de salud. La aplicación contará con un sistema de notificaciones que permitirá recordar las próximas vacunaciones, y se avisará por notificación tipo push que se activan, aunque la app esté cerrada, a todos los integrantes del grupo.

Contará con otro beneficio, que es el listado con mapas y GPS de los lugares donde el paciente se puede dirigir para su respectiva vacuna, al seleccionar el lugar se tramita la cita, fecha y hora; y se convalida la disponibilidad en el calendario de la institución de salud. Luego de este proceso, se presentan dos opciones dependiendo de la institución. Si la institución elige automáticamente, se concreta la cita y queda como notificación, si la institución tiene manual le llega por medio de un mail y la institución debe aprobar la cita. Estos deben estar afiliados y cargar la disponibilidad y gestionar las citas desde Backend de instituciones. De la misma manera, el personal médico tendrá el acceso a la app para conocer el historial de vacunación de cada paciente.

Por otro lado, Informa al paciente sobre la promoción y prevención de enfermedades. Además, preserva el cuidado del medio ambiente, mediante la no utilización del papel pasando a la digitalización; de esta manera se reduce las necesidades de espacio y almacenamiento, favoreciendo la conservación y distribución de la información, por último, se pretende lograr la cobertura tecnológica en telemedicina en inmunoprevención, prevención en pacientes gestantes, pediátricos y adultos.

No obstante, para reconocer la importancia que tiene el uso de la herramienta en el registro y gestión de los datos sobre el proceso de vacunación, a la vez que se reflexiona sobre la importancia de los procesos de vacunación en la actualidad, es necesario, examinar el campo de trabajo analítico que se ha presentado sobre este campo de estudio, particularmente sobre la relación entre los procesos de vacunación y la gestión de información por medio de las herramientas como los carnés de vacunación, y en el caso de este artículo, con los aplicativos web.

Para el abordaje de los trabajos analíticos sobre el proceso de vacunación y la gestión de información a partir del uso de herramientas, se han organizado dos campos de reflexión sobre los documentos presentados. Por un lado, un escenario en el que se presentan trabajos que centran su interés en revisar cómo y cuáles son las dinámicas del proceso de vacunación y su incidencia en las prácticas que hacen seguimiento a este ejercicio. Y un segundo camino, en el que se ubican los documentos en donde se generan preguntas analíticas sobre el comportamiento de las herramientas que hacen registro, seguimiento y control al proceso de vacunación.

Sobre la vacunación.

Es importante tener en cuenta los estudios relacionados con la vacunación, su implicación con la inmunización del paciente y la preocupación del personal de salud, entorno a lograr un esquema de vacunación óptimo, es así como en el estudio denominado "*Oportunidades perdidas de vacunación. Participación del personal de salud*" de Federicci, M., Gatica, C., Nalda y colaboradores (1999), recopila las opiniones del personal de salud relacionado directamente con la verificación de la información del estado de vacunación del paciente, constatando en el presente estudio, que el 87 % de los médicos encuestados del Hospital, refirieron pedir el carné o libreta de salud siempre o a menudo, para confirmar el estado de vacunación del paciente. Las oportunidades perdidas de vacunación, son definidas en el presente estudio, como aquellas detectadas por el perrsonal de salud, durante las visitas de los pacientes al servicio de salud, con el esquema de vacunación incompleto, sin tener contraindicaciones y sin utilizar estas visitas para recibir las dosis necesarias.

El estudio describe, la importancia de la solicitud del carné por parte del personal de salud, en el momento del seguimiento al cumplimiento al sistema de vacunación , como la forma de credibilidad o confianza que tiene el personal de salud, para confirmar el estado de inmunización de un paciente.

En el artículo, titulado *"Oportunidades perdidas de vacunación"* de los autores Brieschke, Manterola, Luppino y otros (2001), los autores buscan analizar y revisar en el hospital, el escenario de 'oportunidades pérdidas' en pacientes con proceso de vacunación, con el propósito de determinar cuáles son las implicaciones y causas que producen el comportamiento en el sistema y proceso de vacunación.

El proceso investigativo se obtienen algunos datos, asociados a lo que los autores determinan como 'oportunidades perdidas', que relacionan los escenarios de salud con la falta de control médico que se realiza sobre el ejercicio de la vacunación.

En el segundo ejercicio investigativo, titulado *"Esquemas atrasados de vacunación y oportunidades perdidas de vacunación en niños de hasta 24 meses: estudio multicéntrico."* De Gentile, Bakir, Firpo y otros autores (2011), se busca caracterizar el grado de cumplimiento del esquema de inmunización del PNI, analizando la proporción de EAV (esquemas atrasados de vacunación) y OPV (oportunidades pérdidas de vacunación) en la consulta actual, y determinar los factores asociados a su ocurrencia.

Con este ejercicio analítico, se busca reconocer las variables que operan en los procesos de vacunación, especialmente en los procesos médicos de los niños, y en las dinámicas de control y eficiencia sobre el ejercicio de revisión y seguimiento.

Otro ejercicio que se articula con lo expuesto es el artículo titulado *"Las vacunaciones caen durante la pandemia"*, de Fernando Moraga (2020), presenta y reflexiona sobre el proceso de vacunación fruto de la pandemia de COVID-19, especialmente en el impacto e importancia de las

vacunas, así como su influencia en los sistemas de salud y la gestión del sistema de vacunación en las diferentes instituciones de carácter global y nacional, en el orden de los países.

Con el trabajo titulado "*Factores sociodemográficos y culturales maternas y el cumplimiento del calendario de vacunación en niños menores de 5 años en el centro de salud Las Palmas – Tingo María 2017"* de Sayle Flores (2017), se examinan los factores sociales, demográficos y culturales de la madre, a la hora de generar y gestionar el proceso de vacunación de los niños. El estudio se realiza en una muestra de 89 madres, y por medio de la prueba Chi Cuadrada, se establecen mediciones sobre el cumplimiento y efectividad del sistema de vacunación desde las madres.

En el artículo *"Factores que influyen en el incumplimiento del esquema de vacunación en niños menores de 5 años"*, de Cristina Miche (2017), se realiza un ejercicio de estado arte, en donde se revisan sistemáticamente artículos de tipo científico sobre el proceso de implementación del sistema de vacunación.

Cada artículo fue observado y revisado de acuerdo con el proceso de evaluación GRADE, en el que se hallaron factores asociados al incumplimiento del sistema de vacunación como contraindicaciones del personal de vacunación, instrucción y conocimiento de las madres, nivel socioeconómico de los padres, y susceptibilidad a enfermedades prevenibles por vacuna.

De igual forma, en la tesis que lleva de título *"Vacunación para los estudiantes, personal y Docentes de la carrera de medicina humana"* de Gallardo, Castillo y Sandoval (2020), los autores pretenden instituir los lineamientos para la implementación del Esquema nacional de vacunación vigente, en Perú, a partir de los procesos de formación y sensibilización sobre el proceso de vacunación en toda la población estudiantil. Este trabajo, no solo pretende consolidar un ejercicio informativo, sino reflexivo sobre las dinámicas de vacunación y su impacto en los procesos educativos. Hay una pregunta detrás por la vacunación y su relación como proceso.

En concordancia con el campo de análisis de estos trabajos analítico, en la tesis *"Conocimiento de las madres sobre vacunas y su influencia en el cumplimiento del esquema de vacunación",* de las autoras Bustos, Díaz y Soto (2017), se genera una reflexión analítica sobre el papel y cercanía de las madres en la efectividad y cumplimiento de los planes de vacunación. Para esto, las autoras enfatizan en la necesidad de retomar elementos como los factores socioeconómicos y nivel académico de las madres, así como las variables de cumplimiento del sistema de vacunación.

Es importante retomar que esta tesis, al igual que los documentos presentados anteriormente, parten de la necesidad de abrir preguntas sobre el papel de los actores (como la madre) en el cumplimiento del sistema de vacunación, así como las relaciones sociales que se tejen en él.

Siguiendo esta línea reflexiva sobre el marco social en que se producen los sistemas de vacunación, encontramos el artículo denominado "*Factores Socioculturales que influyen en el Incumplimiento del calendario de vacunación en los lactantes atendidos en el Centro de Salud Pedro Pablo Atusparia, Chiclayo 2016",* de Ruth Cabrejos (2017), se retoma la discusión por los factores asociados que influyen en el incumplimiento del sistema de vacunación, particularmente, por la relación de la madre en la gestión y seguimiento del proceso de niños y niñas. Se realiza el ejercicio analítico a modo de estudio de caso, en un centro de salud, y con la mirada reflexiva sobre el papel del actor de la madre. En este caso resalta el procedimiento metodológico en el que la muestra de madres aborda a 100 personas.

Finalizando en este orden de conceptos, que retroalimentan y justifican más aún, el porqué GIGA5, se preocupa por el control y seguimiento de un eficiente plan de vacunación, se contempla el artículo *"Causas del incumplimiento del esquema de vacunación en niños menores de 5 años en el Centro de Salud Indio Guayas del año 2018*" de Alban F, & Minga M. (2019), el cual determina las causas del incumplimiento del esquema de vacunación en estos primeros años de vida, describiendo así, las principales vacunas que no se cumplen en el esquema de vacunación y la influencia de diferentes factores, que pueden estar relacionados con el incumplimiento, tales como, familiares, pediátricos y los inherentes al sistema de salud.

Registro, gestión y sistema de vacunación.

En el trabajo investigativo, titulado "*Utilidad del carnet único de vacunación como instrumento de apoyo para las madres de los niños menores de 1 año 6 meses que acuden al Área de Vacunacion del Centro de Salud Chimbo Distrito 2 Unidad Operativa 3 provincia de Bolívar periodo enero septiembre 2014"*, el autor Jonathan Goyes (2014), busca analizar cuál ha sido el impacto del proceso de implementación del carné de vacunación en el Ecuador, principalmente en la gestión de información y control del proceso por parte de los actores.

En el presente ejercicio, el autor buscar analizar el impacto del carné de vacunación, en la gestión de información para las madres, particularmente en el proceso cotidiano que viven los pacientes y el espacio médico.

Así mismo, se encontró el documento que lleva como nombre *"Sistema web para la gestión y monitoreo del plan de vacunación del municipio de Medellín – SISMOVAC"* de Gómez – Echeverry y otros autores (2018), se busca analizar, diseñar e implementar la herramienta SISMOVAC en la ciudad de Medellín, con la que se pretende gestionar la información del programa de vacunación de niños.

Con el trabajo los autores resaltan el trabajo en dos líneas analíticas: la reflexión sobre el uso de las herramientas tecnológicas en los sistemas de gestión de información sobre las vacunas, así mismo, se examina el impacto de las dinámicas de vacunación en los sistemas de control médico.

Por la misma línea analítica, en el documento *"Los orígenes de la Cartilla Nacional de Vacunación",* de Vazquez Santaella (2014), se presenta cuál ha sido el marco histórico de implementación de la cartilla nacional de vacunación en México, mediante decreto presidencial en 1978, con el propósito de fortalecer el Sistema nacional de salud en el país.

En el trabajo se resalta la importancia que tuvo su implementación para profundizar y fortalecer el sistema de salud, y especialmente, el seguimiento al proceso de adaptación de la población respecto al ejercicio de las vacunas.

Así mismo, y en línea de este campo analítico presentado, en el artículo denominado *"Software para el control de calendarización de vacunas de niños y niñas de 0 a 4 años para la sub-área de sistema de asignación integral de la salud del hospital"*, de Angulo y Contreras (2013), se hace una aproximación reflexiva a la importancia de los software, en la era actual (globalizada), para el desarrollo, gestión y registro de los datos sobre el proceso de vacunación, en el que se puedan organizar y sistematizar la información que el sistema de salud requiere en este proceso.

Se hace una mirada comprensiva al uso de herramientas virtuales para el conocimiento de las variables que operan a la hora de desarrollar el sistema de vacunación en una sociedad determinada, y, sobre todo, en la gestión de los pacientes, especialmente en niños y niñas. En este mismo sentido analítico, se presenta la investigación "Digital mobile technology facilitates HIPAAsensitive perioperative messaging, improves physician-patient communication, and streamlines patient care", de Gordon y Li (2015), en donde se resalta el uso y gestión de tecnologías en los procesos de actuación de procesos del sistema de salud.

En el trabajo *"Propuesta para la implementación de una aplicación móvil con tarjeta de vacunación digital para llevar el registro de vacunas de niños y niñas de 0 a 5 años de la posta SASBI de Villa El Salvador"* de Miriam Chaupi (2020), se vuelve sobre la importancia de la creación de una App, que permita el registro, desarrollo y gestión de datos sobre el sistema de vacunación para la prevención de muertes y seguimiento, especialmente en la población entre los 0 y 5 años de edad.

Sigue el mismo ejercicio de trabajo, el documento "Dispositivos digitales móviles en educación: el aprendizaje ubicuo" de Esteban Vazquez (2015) en el que se insiste en el campo analítico de las herramientas virtuales en los procesos de recolección y transformación de datos.

Por otro lado, pero siguiendo la misma línea analítica, el trabajo *"Minería de Datos con Redes Neuronales Artificiales: Aplicación en Vacunas – Tuberculosis.",* de Guzman, Carrillo, Villamayor y otros autores (2004), explora un campo sobre el proceso de gestión y manejo de datos que se presenta como 'Minería de datos', desde el cual se centra la mirada en los procesos de gestión de grandes redes de información en las que se trazan datos, establecen variables y modelaciones matemáticas para el diseño y gestión de procesos.

Este trabajo reconoce por los menos tres lugares analíticos por los cuales pensar la relación entre herramientas tecnológicas y gestión de datos asociados a los procesos de vacunación: 1) el desarrollo y potencia del conocimiento sobre los procesos de vacunación, 2) la gestión y movimiento de los datos, y 3) el cruce y establecimiento de variables de la información recolectada, en la que operan de manera pertinente los softwares desarrollados.

Y cerrando el marco de los trabajos presentados en este campo, la tesis titulada *"Propuesta para el diseño de un sistema de Información para el registro de vacunación en el hospital Alberto Sabogal Sologuren"* de Yovan Salic (2015), pretende como propósito general proponer el diseño de un sistema de información para el registro de vacunación del Hospital Alberto Sabogal Sologuren, a partir de tres elementos abordados en el ejercicio: una hoja de definición sobre los requerimientos y necesidades del sistema de vacunación, también una lista cotejo sobre el proceso de vacunación efectuado, así como una hoja de registro sobre el proceso de vacunas por año.

Desde el ejercicio investigativo, se busca ahondar en el análisis y desarrollo de estrategias para el conocimiento del proceso de vacunación, así como su dinámica en los procesos de efectividad en el hospital. Se reflexiona sobre el papel de la información y su gestión.

2. Diseño de la APP

Un sistema operativo móvil se entiende como un conjunto de programas utilizados en dispositivos móviles y que permiten el uso del hardware de este tipo de dispositivos mediante el suministro de servicios a las aplicaciones que se ejecutan en el sistema, debido a que durante los últimos años es evidente el crecimiento del uso de dispositivos móviles tales como teléfonos inteligentes y tabletas, el desarrollo de software mediante aplicaciones de este tipo se hace cada vez más necesario a la hora de pensar en la creación de herramientas de software orientadas al usuario final.

Actualmente existe una gran variedad de sistemas operativos móviles, sin embargo, los dos de mayor cobertura, uso y por lo tanto domino del mercado son Android, con una participación del 71,24% del mercado y por otra parte IOS con el 28.26% tal como se observa en la figura N°1 de acuerdo con reporte del NetMarketShare para abril de 2021.

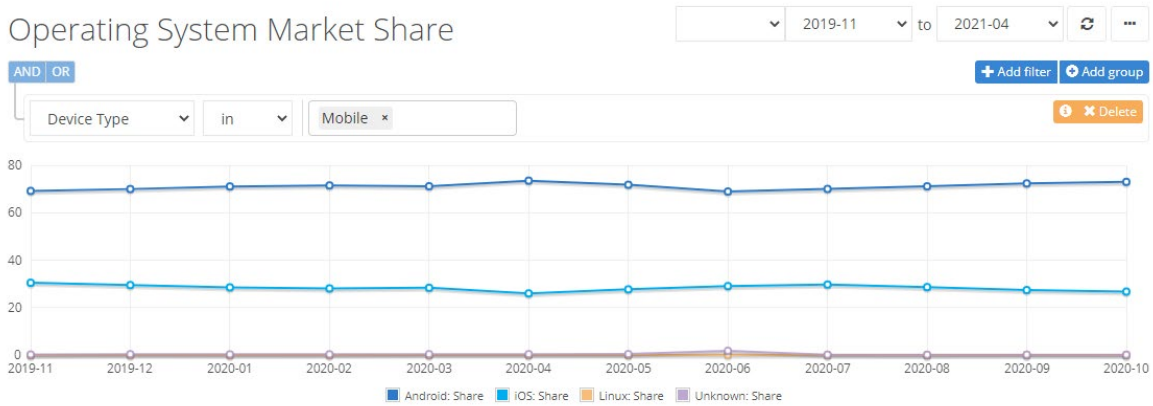


Figura 1. Participación de Mercado Sistemas Operativos Móviles.
Fuente: https://netmarketshare.com/

Esta situación soporta entonces el hecho de que la App de Vacunación Inteligente haya sido desarrollada para estos dos sistemas operativos y bajo el framework Ionic, que permite gestionar un código fuente base para ambos sistemas con tecnologías html, css y javascript e integraciones nativas a través de capacitor. Se trata entonces, como ya se ha mencionado, de una aplicación que busca generar un sistema de vacunación inteligente en el que empresas farmacéuticas, entidades prestadoras de servicios de salud, profesionales de la salud y personas naturales, pueden gestionar y hacer seguimiento al proceso de vacunación.

Para el desarrollo de la aplicación, se estructuró un esquema de trabajo abordado durante cinco grandes fases desde el levantamiento de los requerimientos como insumo para el proceso de desarrollo y programación, hasta el testeo y publicación del software, a continuación, se hace una breve presentación del proceso y sus fases:

1. Levantamiento de requerimientos: esta es la primera y más importante etapa en el ciclo de desarrollo del software, en este caso fue adelantada a partir de la observación del proceso tradicional de vacunación y de la entrevista a algunos de los grupos de interés vinculados al mismo.
Requerimientos mínimos:
- Android: API 21+ (Android 5 o superior)
- IOS: iOS 12+ is supported.

2. Diseño y Prototipaje: una vez identificadas las necesidades de los diferentes grupos de interés y perfiles asociados al sistema de vacunación se trabajó en la construcción de un bosquejo o mockup interactivo el cual fue desarrollado en la plataforma Adobe XD para su posterior validación.
- Prototipado: **Adobe XD 31.2**.

3. Modelación de Datos: con el mockup validado se procedió al diseño y estructuración de las bases de datos sobre las cuales corre el software de la aplicación, en este caso se trata de una base de datos NoSQL Couchbase, la cuales dentro de sus principales características cuenta con estar orientada a documentos, distribuida, de alto rendimiento, fácilmente escalable y de propósito general. De igual forma la base de datos generada integra una solución de movilidad (Couchbase Mobile), que incluye una base de datos simplificada en el dispositivo móvil (Couchbase Lite) que se sincroniza con una base de datos centralizada, y permite su funcionamiento en modo offline, lo que se convierte en una de las principales características de este desarrollo.
– Base de datos: **Couchbase Community Edition 6.6.0**.

4. Programación: finalizado el diseño y desarrollo de la base de datos se procedió a la programación para la implementación de las funcionalidades de la APP teniendo en cuenta los niveles de usuarios que se presentan en el diagrama de caso de uso.
- Framework para apps Híbridas**: Ionic 5, Angular 11 y CapacitorJS 3.**
- **Lenguajes: JavaScript, CSS, HTML**.

5. Backend: el proceso de desarrollo finaliza con la implementación del administrador (Backend) de la plataforma y los ajustes del desarrollo web, acciones tales como la conexión con la base de datos o la comunicación con el servidor de hosting, esta fase fue desarrollada en Java Script y NodeJS como webserver. El proceso de backend finalizó con actividades de testeo y ajustes de programación para su posterior publicación.
- Sistema Operativo: **Centos 8.**
- Web Server: **NodeJS 10.24**.

Como se puede observar en el diagrama de Infraestructura (figura N° 2), GIGA5 Vaccine es una app enfocada a un paradigma offline first, esto significa que cuenta con una base de datos cifrada en el teléfono que sincroniza cada vez que hay conexión a internet con el servidor, esto como ya se ha mencionado genera una de las principales características y diferenciación de la app, ya que permite de la mayoría de las funcionalidades estén disponibles sin conexión a internet; para la sincronización se utiliza la base de datos couchbase y las apis de consulta que trabajan bajo nodeJS.

2.1 Ventajas comparativas de la App GIGA5

Los principales diferenciales y esquemas de generación de valor de la app se encuentran entonces no solo en la posibilidad de generación de funcionalidades sin necesidad de conexión a internet, sino de otras importantes funcionalidades a destacar tales como:

- La posibilidad de escaneo de un código QR para el rápido acceso a los datos del paciente, permitiendo tener una historia clínica que puede ser accedida rápidamente por el personal médico.

- La alimentación de información con un esquema de fotografías de soporte al proceso de vacunación, lo cual apoya el seguimiento y control al proceso de vacunación, reduciendo de igual forma el tiempo interacción del personal médico con la herramienta y generando soportes para futuros procesos de auditoria.

- La digitalización de algo tan importante como el carné de vacunación infantil, para madres gestantes y tercera edad, genera un salto en el paradigma actual donde aún no había llegado la adopción digital, brindando así ventajas de portabilidad y rápido acceso incluso funciones de alertas con notificaciones para apoyar y evitar que se salten o retrasen vacunas claves del calendario.

- La customización es otra funcionalidad a destacar, permite que el contenido de promoción y prevención tan importante para la salud de la población pueda ser dirigido según las características del usuario, logrando así llegar con la información oportuna y adecuada a cada tipo de paciente.

- Otro punto importante es la posibilidad de interconexión con otros sistemas pudiendo nutrirse de información y brindar información a otros sistemas logrando que fluya el conocimiento médico a través de la Big Data y reduciendo el retipeo de información del paciente y perdida de historias médicas, gran problema actual del área de la salud, como se muestra en la figura N°2 sobre el diagrama de infraestructura.

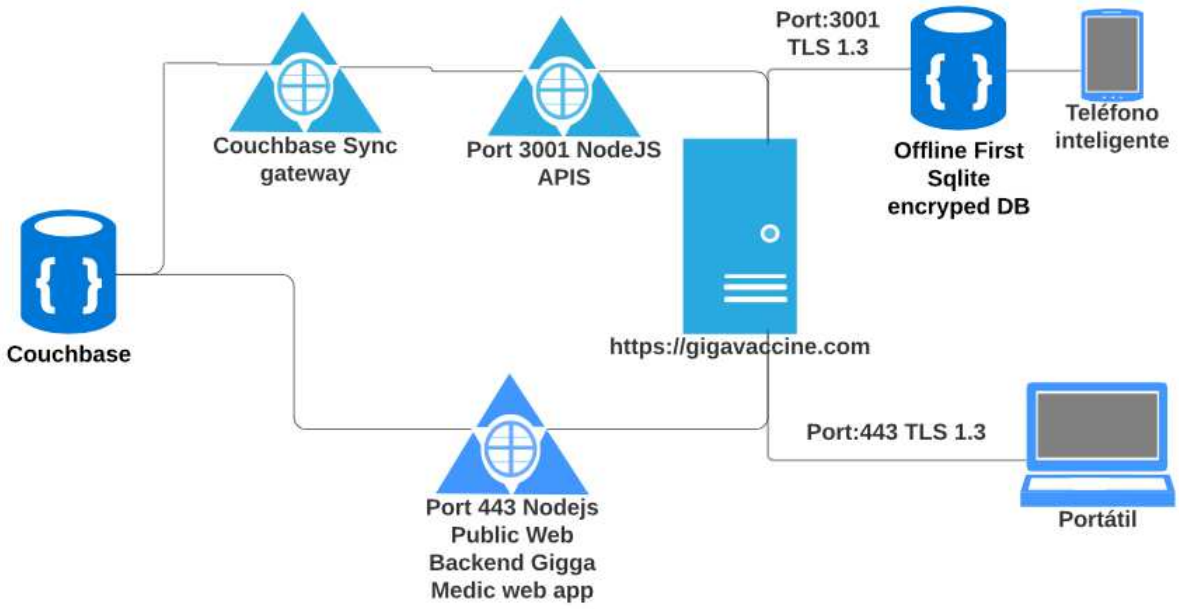


Figura 2. Diagrama de Infraestructura. Fuente: Elaboración Propia

2.2 Funcionamiento y operatividad de GIGA5

Ya en su funcionamiento GIGA5 Vaccine opera como un ecosistema en que interactúan diferentes roles o perfiles, cada uno de ellos con características, funcionalidades y perfiles diferentes dentro de la aplicación, los cuales interactúan tal y como se puede observar en la figura N° 3 que demuestra el diagrama del caso de uso, nos encontramos entonces frente a los siguientes roles:

a. Usuario: se refiere a las personas naturales, personas que desean vacunarse, así como sus respectivos dependientes.

b. Personal Salud: se encuentran en este rol, el personal que aplica las vacunas, así como los médicos que hacen seguimiento al proceso.

c. Entidad Medica: es el administrador del prestador de salud que ingresa datos administrativos.

d. Administrador GIGA5, Personal de GIGA5 encargado de ingresar Entidades médicas y Gestionar la plataforma.

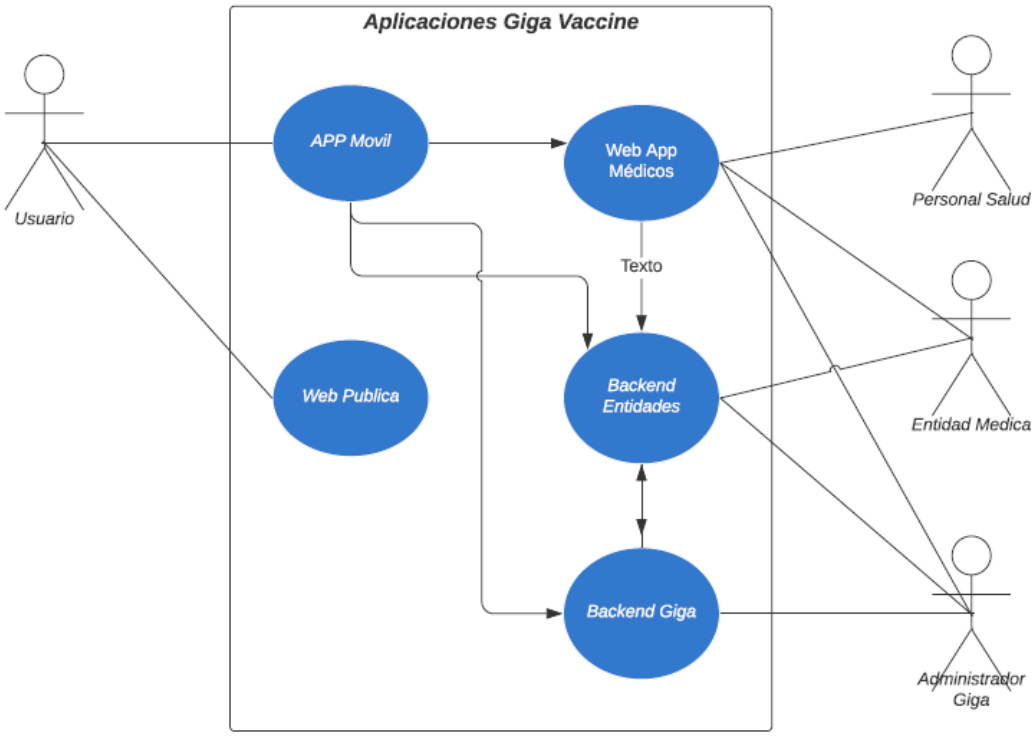


Figura 3. Diagrama de Caso de Uso. Fuente: Elaboración Propia

Cada uno de los roles asociados en el caso de uso, presentan entonces diferentes niveles de funcionalidad siendo los siguientes algunos de ellos, en cuanto a instituciones (tabla Nº 1), profesionales de la salud (tabla Nº 2) y administrador de la App (tabla Nº 3):

Tabla 1. Instituciones.

Instituciones (IPS)	
Login	Se podrán acceder con las credenciales otorgadas por GIGA5.
Usuarios	Podrán crear, editar y eliminar usuarios médicos.
Calendario	Podrán configurar el cupo, horarios disponibles a futuro para que los usuarios puedan generar citas automáticas, editarlas o cancelar, o bien generar cita manual vía mail y su respectiva aprobación.
Historial de vacunaciones	Pueden ver las actividades de los médicos que realizaron las vacunaciones.
Estadísticas	Pueden ver las citas online concretadas, canceladas o ausentes.

Tabla 2. Profesionales de la salud.

Profesionales de la Salud	
Login	Aquí el medico ingresará con las credenciales que se le crearon.
Escanear QR (dashboard si es app de celular)	Se escanea el QR, con esto identifica que tipo de paciente es, y por qué vacuna y/o servicio asiste.
Buscar paciente	Esta opción permite al médico buscar cualquier paciente que esté registrado en la app.
Aplicar vacuna	Después de haber escaneado o buscado al paciente, si es infantil ya le aparecerá la vacuna. Tendrá en las opciones un botón para activar la cámara y tomar la foto del envase de la vacuna. Luego de la foto, la vacuna figurará como aplicada.
Registrar historial	En caso de paciente gestante podrá registrar la historia clínica.
Historial del paciente	Podrá ver el historial de vacunación del paciente.

Tabla 3. Administrador de la App.

Administrador de la App	
Vacunas infantiles	
Promoción y prevención	Permite cargar, editar y eliminar tipos de vacunas, repetición de estas, fechas e indicar las próximas a aplicar.
Centros Vacunación	Permite cargar, editar y eliminar artículos informativos, educativos y leyes, asignados a diferentes categorías.
Zona de marketing	Permite cargar, editar y eliminar los centros de vacunación: nombre, dirección, ubicación GPS, usuarios y contraseñas para médicos e instituciones.
Estadísticas del App	Permite cargar, editar y eliminar banners de publicidad con enlaces.

2.4. Modelo de Negocio

GIGA5 ha sido planteada como un desarrollo tecnológico soportado sobre una necesidad real del mercado y con una clara orientación a materializarse como un proceso de transferencia de resultados de investigación que se concreta en un desarrollo empresarial enfocado a su uso, administración y gestión, de esta manera paralelo al proceso de desarrollo del software de la app se trabajó en la construcción de un modelo de negocio que soporte la generación de una Startup enfocada a la tendencia del healthtech como lo evidencia el marco de la figura Nº 4.

Para la construcción del modelo de negocio se trabajó bajo la metodología del Canvas Business Model (Osterwalder y Pigneur 2010), la cual se centra en el proceso de creación y distribución de valor por parte de una organización o empresa entre los diferentes grupos de interés, para ello la propuesta se centra en definir 9 variables (bloques) que permiten a las organizaciones desarrollar este proceso, a continuación, se presentan cada una de estas variables: segmento de mercado, propuesta de valor, canales de comunicación y distribución, relación con los clientes, fuentes de ingresos, recursos clave, actividades clave, red de aliados, generadores de costo.

1. Segmentos de mercado: en el proceso de vacunación se articulan varios actores y cada uno de ellos se convierte en un segmento de mercado de vital importancia tanto para el desarrollo de la app, como para la gestión del modelo de negocio que busca hacerla sostenible como empresa, en ese orden de ideas GIGA5 busca tener 5 segmentos de mercado que se irán desarrollando y consolidando paulatinamente, siendo estos los usuarios personas naturales más conocidos en el sector salud como pacientes, quienes a su vez se dividirán en dos categorías, el paciente usuario de la versión gratuita de la aplicación y el usuarios de la versión premium; las Entidades Promotoras de Salud (EPS) o Empresas de Medicina Prepagada; las Instituciones Prestadoras de Servicios de Salud (IPS) y las empresas farmacéuticas o proveedoras de vacunas.

2. Propuesta de valor: se busca la generación de beneficios para cada uno de los segmentos de clientes mencionados en el anterior aparte, es por ello que se ha buscado definir igual número de propuestas de valor, las cuales se pueden concentrar para los pacientes o usuarios de la versión gratuita en el contar con un carné de vacunación on line, así como el poder acceder a funcionalidades tales como la solicitud y recordatorio de citas de vacunación; por su parte los pacientes o usuarios de la versión premium tendrán adicional a los anteriormente mencionados servicios de vacunación extramural más la posibilidad de acceso a citas médicas de primer nivel. Por su parte las EPS tendrán acceso a información y seguimiento sobre los procesos de vacunación de sus pacientes, en tiempo real, así como la posibilidad de realizar procesos de actualización de la información base de sus pacientes mediante campañas asociadas a los procesos de vacunación; las IPS tendrán acceso a una interfaz webapp para la gestión de sus servicios de vacunación digital, el carné de vacunación digital de sus usuarios, posibilidad de gestión de citas on line, seguimiento a la gestión de vacunación, recordatorios de dosis, chat médico, reportes on line a Secretarías y el Ministerio de Salud. Finalmente, a las empresas farmacéuticas se les entregarán reportes de la gestión de los datos asociados a los procesos de vacunación.

3. Canales de comunicación y distribución: las estrategias de comunicación y distribución se plantean en dos grandes líneas de acción, por una parte está la gestión de una venta consultiva orientada a la relación con segmentos tales como los de las empresas farmacéuticas, las EPS y las IPS, y por otra acciones de tracción de usuarios mediante campañas de marketing digital y gestión de redes sociales, así como gestión comercial a través de puntos de contacto y control (espacios de amplia circulación de potenciales usuarios).

4. Relación con los clientes: GIGA5 Vaccine como app, se convierte en si misma como la principal herramienta de relación con los clientes, toda vez que, a través de sus funcionalidades permite desarrollar diferentes tipos de reportes que facilitan el conocimiento y entendimiento de los diferentes niveles de usuarios.

5. Fuentes de Ingresos: la monetización del modelo de negocio se plantea mediante la relación comercial con tres de los cinco segmentos de mercado ya relacionados, por una parte, están los

contratos con EPS para recobro de la aplicación de vacunas, las membresías premium para usuarios pacientes y la venta de reportes semestrales y anuales de análisis de datos para las empresas farmacéuticas.

6. Recursos clave: el desarrollo del modelo de negocio y específicamente la posibilidad de concretar las propuestas de valor ya mencionadas se da gracias al poder contar con recursos tecnológicos, físicos y de conocimiento (know how) tales como la misma app y su versión web, los datos generados y administrados a través de la misma, el modelo de vacunación extramural y una red centros de vacunación físicos asociados a la plataforma digital.

7. Actividades Clave: las principales actividades que soportan el proceso de la generación y entrega de las propuestas de valor, así como del funcionamiento del modelo de negocios, son el mantenimiento, actualización y soporte de la app, los procesos de calidad en el servicio y las vacunaciones, la organización y ejecución de las jornadas de vacunación así como las actividades de promoción y prevención de salud, y como soporte a todo lo anterior las acciones de marketing y comunicaciones hacia los diferentes grupos de interés.

8. Red de Aliados: la sostenibilidad y desarrollo de cualquier modelo de negocio se da en gran medida a la capacidad de la organización en relacionarse con diferentes actores del entorno, en este caso GIGA5 plantea un esquema amplio y estrecho de relacionamiento con sus propios segmentos de clientes corporativos (EPS, IPS y farmacéuticas) así como otras organizaciones de orden público asociadas a la formulación y operación de la política pública en materia de salud (Ministerio y Secretarías de Salud).

9. Generadores de costo: la posibilidad del desarrollo de las actividades asociadas al modelo de negocio y operación de GIGA5 se dan gracias a inversión de recursos financieros de la Startup dedicada a su operación y seguimiento, tales inversiones se concentran entonces entre otros rubros tales como el mantenimiento y actualización de la plataforma, compra de vacunas e insumos necesarios para su aplicación, costos de transporte, nómina técnica de administración y servicios de salud.

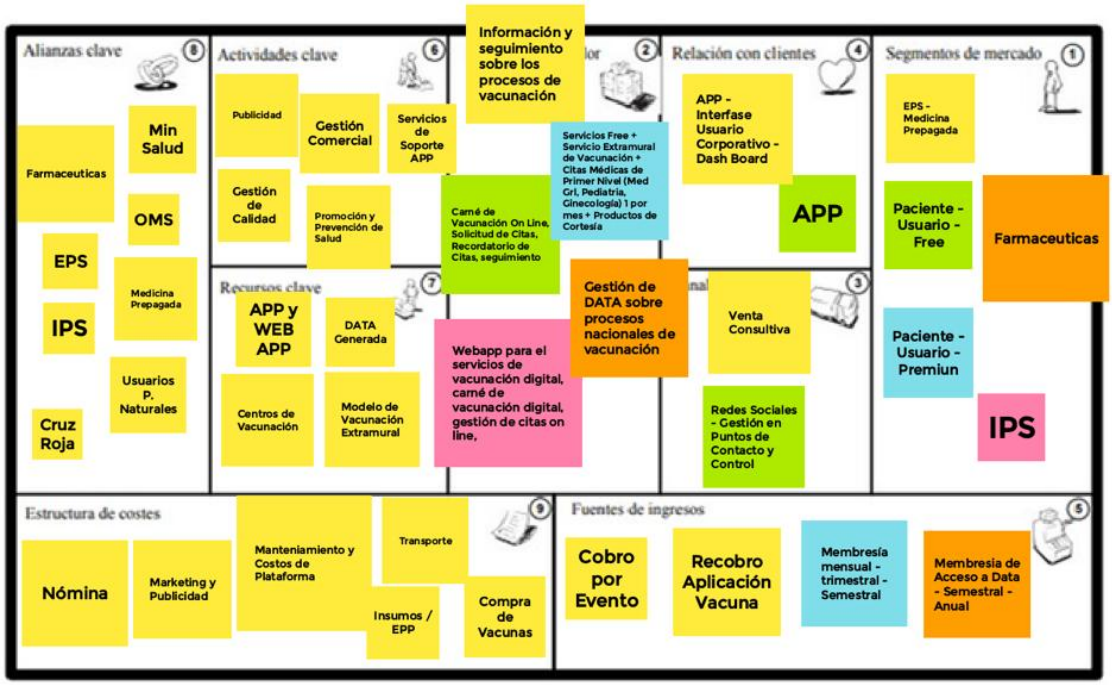


Figura 4. Lienzo Modelo de Negocio GIGA5 Vaccine. Fuente: Elaboración Propia.

3. Resultados

Una vez abordadas las etapas de diseño y desarrollo, así como la definición del modelo de negocio, anteriormente expuesta, se cuenta entonces con dos aplicativos funcionales que hacen parte del Sistema de Vacunación inteligente GIGA5 Vaccine, el primero una app disponible tanto en el sistema operativo Android como IOS y cuya orientación es primordialmente hacia los usuarios personas naturales y otro bajo un esquema de WebApp diseñado y desarrollado para el uso por parte de los profesionales de la salud que alimentan y gestionan el sistema, a continuación se presentan las principales interfases de los dos desarrollos.

3.1 App – Usuario

Este aplicativo cuenta entonces con cuatro grandes interfases en las que los usuarios personas naturales hacen parte y usufructúan el sistema de vacunación, para ellos y sus dependientes GIGA5 Vaccine cuenta entonces entre otros de los siguientes componentes:

- Registro: Esta es la primera interacción del usuario con el sistema y se concentra en la configuración de su perfil, en este momento esta acción puede generarse a través de la creación directa del perfil mediante el suministro de datos tales como nombre, correo electrónico, EPS o empresa de medicina prepagada a la que se encuentra vinculado y teléfono móvil, este último es validado por el aplicativo a través de la generación de un mensaje de texto vía SMS (figura Nº 5).

De igual forma en este proceso los usuarios podrán generar sus dependientes, es decir miembros de su núcleo familiar que serán beneficiarios del sistema de vacunación, los dependientes pueden ser adultos mayores, hijos, o cualquier otra persona que se desee agregar para administrar sus citas y vacunaciones. Este proceso lo podemos encontrar en la figura 5.

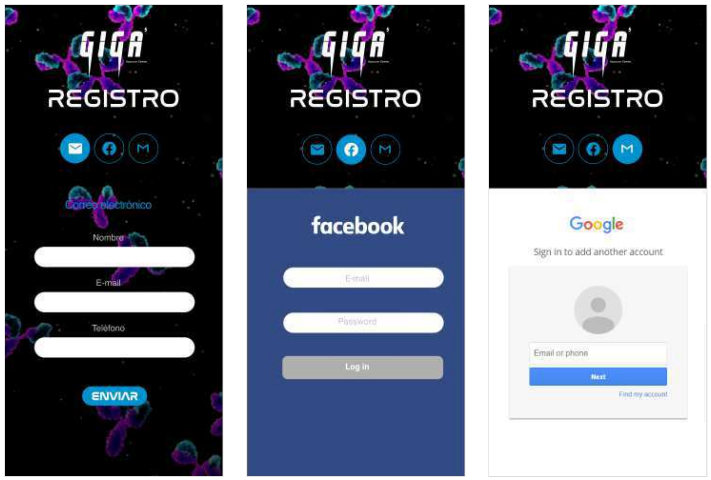


Figura 5. Interfaz de Registro. Fuente: GIGA5 Vaccine.

- Cuenta – Carné Digital: en esta interfaz el usuario cuenta con un código QR, el cual puede ser escaneado por la WebApp médica o de profesionales de la salud y de esta manera acceder al historial de vacunación del usuario y/o sus dependientes, de igual forma el carné digital permite visualizar en forma de lista o calendario las citas pendientes (Se explica el proceso en la figura Nº 6).

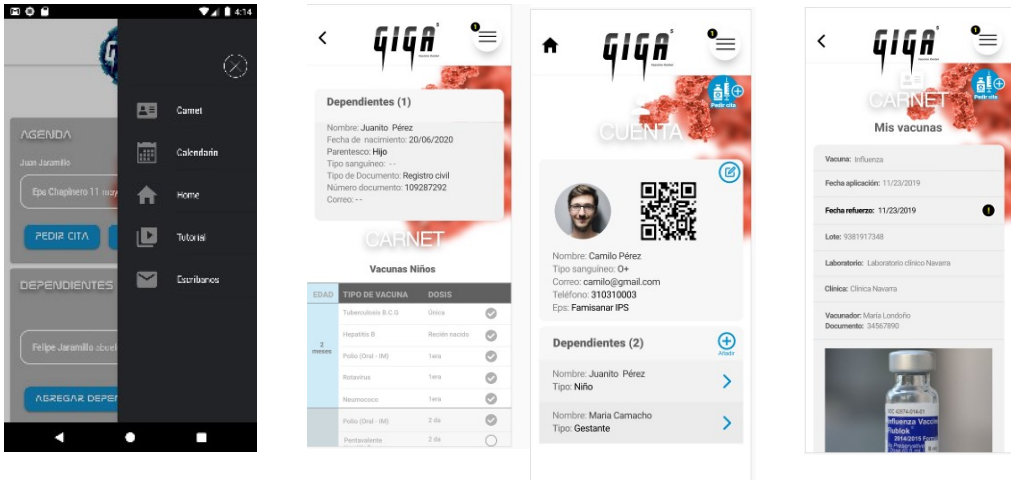


Figura 6. Interfaz Cuenta – Carné Digital . Fuente: GIGA5 Vaccine.

- Agendamiento de Citas – Calendario: Esta es la interfaz con las funcionalidades de mayor relevancia para el usuario paciente, en ella el titular puede asignar para sí mismo o sus dependientes, las citas necesarias para realizar el proceso de vacunación, eligiendo variables tales como ciudad, lugar, tipo de vacuna, fecha y hora para la realización del procedimiento.

Igualmente, permite consultar el proceso de vacunación, bien sea a través del calendario o una lista de citas programadas y pendientes, allí se encuentran entonces disponibles los detalles de la cita y un esquema de georreferenciación del lugar donde se debe acudir a la cita de vacunación. Los elementos se muestran en la figura Nº 7.

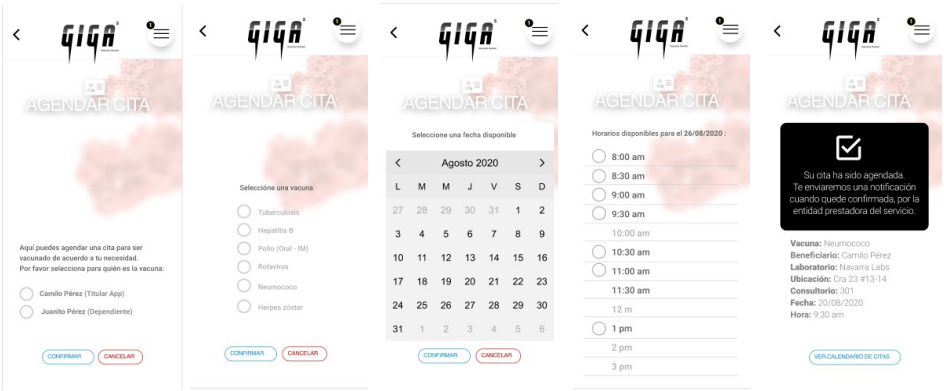


Figura 7. Interfaz Agendamiento – Calendario. Fuente: GIGA5 Vaccine.

- Noticias: finalmente se cuenta con un espacio de noticias en el cual se comparten contenidos relevantes para los usuarios, encaminados no solo a la generación de información relevante asociada a los procesos de vacunación, sino que se convierte en una de las

principales herramientas para que el sistema puede generar programas de promoción y prevención en salud, como se expresa en la figura Nº 8.

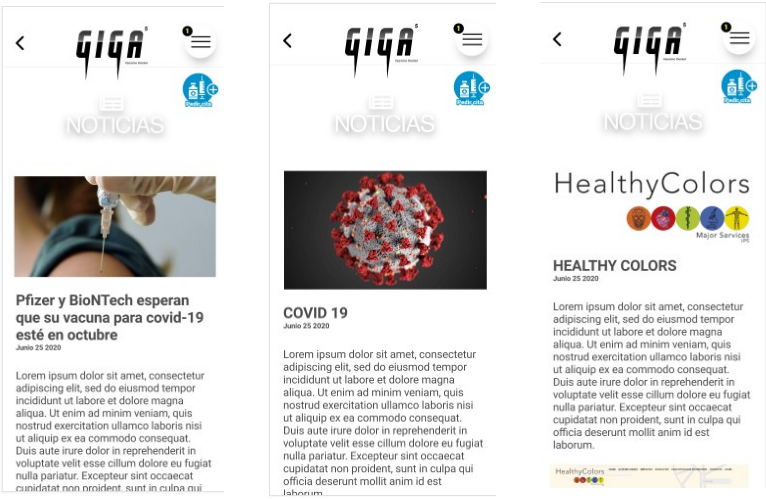


Figura 8. Interfaz Noticias. Fuente: GIGA5 Vaccine.

3.2 WebApp – Profesionales de la Salud

El otro componente fundamental para la consolidación del Sistema, es el proceso de alimentación de información por parte de los profesionales de la salud, a la hora de generar la aplicación de las vacunas, para ello y teniendo en cuenta sus necesidades, se ha desarrollado un aplicativo Webapp para ser consultado y manejado por este perfil de usuarios, para este aplicativo se cuenta entonces con cuatro interfases, contando entonces con los siguientes componentes:

- Registro e ingreso: es la interfaz de acceso para los profesionales de la salud (figura Nº 9) y básicamente permite la generación de este tipo de usuarios, su caracterización al interior de la base de datos y la customización de su perfil, gracias a la información suministrada y clasificada en esta primera interacción, serán posibles seguimientos a las acciones desarrolladas por los diferentes profesionales de la salud, así como reportes por instituciones (IPS, EPS, empresas de medicina prepagada).

Figura 9. Interfaz Registro – Ingreso. Fuente: GIGA5 Vaccine.

- Agenda: la funcionalidad base del desarrollo de este aplicativo, está en la posibilidad de los profesionales de poder consultar y hacer seguimiento a su trabajo asociado a los procesos de vacunación. En esta interfase no solo podrán consultar de manera rápida y sencilla su agenda o plan de trabajo diario, sino igualmente las características básicas de las citas asignadas para el día. El proceso se determina en la figura Nº 10.

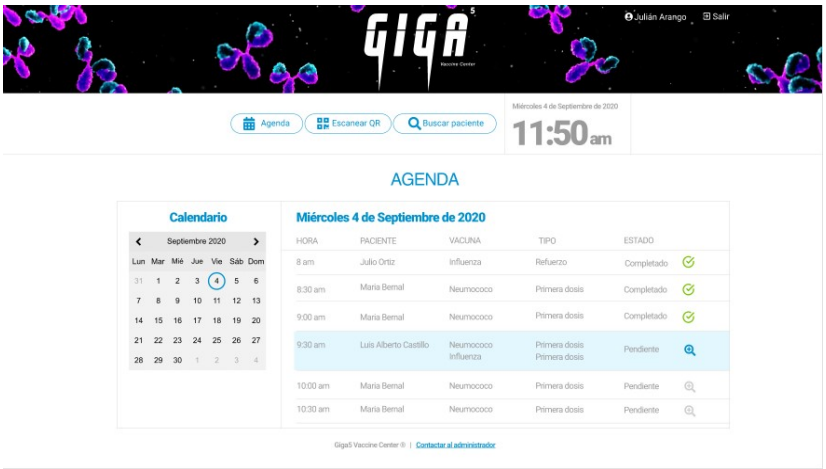


Figura 10. Interfaz Agenda. Fuente: GIGA5 Vaccine.

- Consultas de pacientes: el proceso de consulta de pacientes, está diseñado para tener un rápido acceso a la información de estos a través de escanear los códigos QR, que como ya se mencionó, se han generado para cada uno de los usuarios, persona natural en el marco de la app GIGA5 Vaccine,

gracias a este proceso se da un enlace al carné de vacunación digital y la información en el registrada (Ver proceso en figura Nº 11).

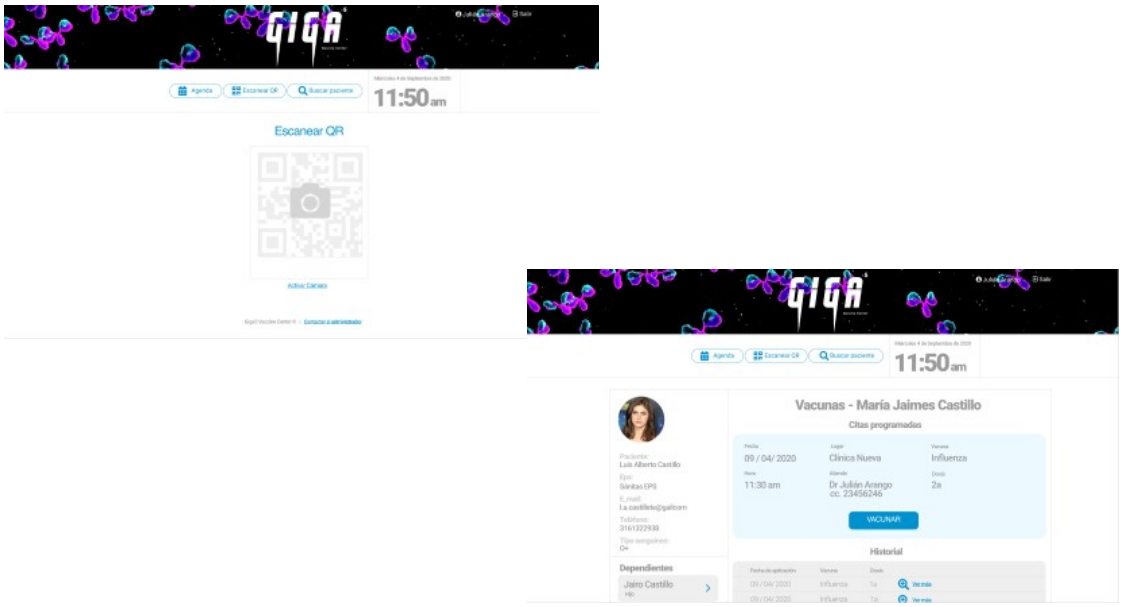


Figura 11. Interfaz Consulta de Pacientes. Fuente: GIGA5 Vaccine

- Gestión de Usuarios: finalmente se encuentra la interfaz de alimentación al sistema de vacunación, mediante la cual los profesionales de la salud registran el proceso de vacunación como tal, generando los datos que alimentan la base de datos central y de esta manera la posibilidad de generar reportes para los diferentes niveles de acceso, en esta parte del proceso se origina también el registro fotográfico, que permite generar el soporte del proceso de vacunación, como es observado en la figura Nº 12.

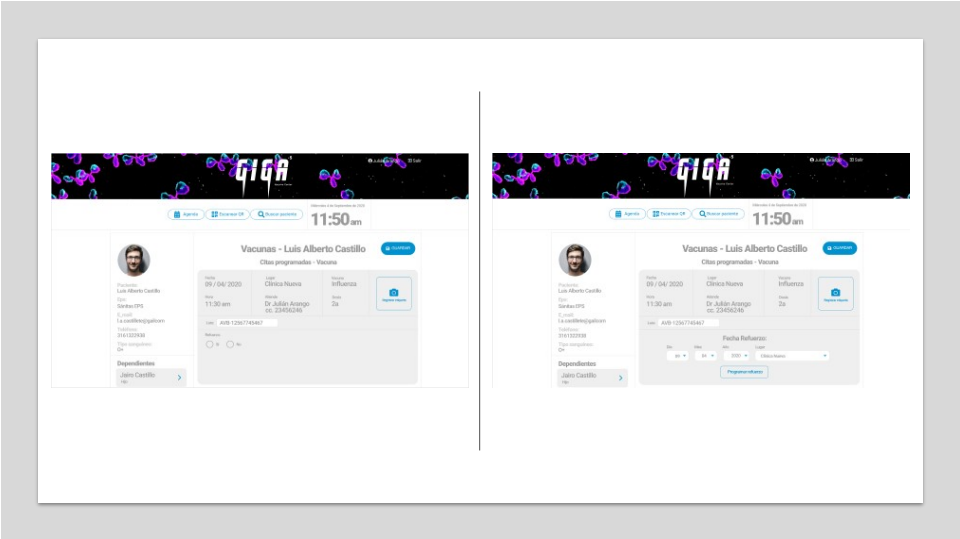


Figura 12. Interfaz Gestión de Usuarios. Fuente: GIGA5 Vaccine.

4. Conclusiones

La implementación, desarrollo y proceso del modelo de negocio del Sistema de vacunación inteligente GIGA5 Vaccine, presenta algunos elementos que permiten ampliar el escenario investigativo sobre los procesos de vacunación, particularmente, en el proceso de gestión de datos y recopilación de la información de pacientes y médicos. Es por esto, que con este artículo se permitió el análisis en varias líneas presentadas anteriormente y que se pueden determinar en los siguientes puntos:

- Los procesos de vacunación, y la gestión de las vacunas, como evidencia en el marco investigativo presentado en el estado del arte, y en el objetivo del presente artículo, presenta fallas y problemáticas agudas, en cuanto al manejo de datos y desconocimiento de los usuarios sobre su incorporación en los programas de vacunación. Se hace fundamental generar preguntas analíticas sobre cómo se gestionan los datos, y su efectividad en el control de la información.

- En el escenario actual de pandemia por la COVID-19 (SARS-CoV-2), el desarrollo de apps como GIGA5, permiten hacer un seguimiento efectivo del proceso de vacunación de los usuarios, no solo por el control de datos y registro de la implementación del número de dosis y la naturaleza de la vacuna, sino que permite el intercambio de información entre las IPS y el usuario, para el acompañamiento de sus pacientes en la solicitud de citas, aplicación de las dosis con todos los detalles necesarios (tipo de vacuna, lugar, recomendaciones, aplicación y seguimiento), y proceso posterior que se articula en el ejercicio de vacunación en Colombia.

- Las herramientas generadas para registrar, gestionar y desarrollar procesos de seguimientos en los usuarios sobre su papel en los programas de vacunación no han sido lo suficientemente efectivos para recopilar y conocer el registro de los usuarios. Se hace necesario construir herramientas que permitan un sistema de gestión desde el usuario y profesionales de la salud.

- El modelo de negocio y propuesta del Sistema de vacunación GIGA5 Vaccine, es un sistema de vacunación inteligente, que permite incorporar y fortalecer la participación de los usuarios y profesionales de la salud, a partir de la gestión de datos, el conocimiento del sistema de vacunación y el cruce de la información.

- La implementación y efectividad del programa, combina su eficiencia en tres puntos fundamentales en el sistema de gestión de datos sobre los procesos de vacunación: el rol de los usuarios y profesionales de la salud, la recopilación de información y cruce de datos, y el ejercicio de análisis de datos para futuros espacios de reconocimiento y acción desde el sistema de salud, en el que se determinan errores, ausencias y oportunidades de acción.

Referencias

Federicci, M., Gatica, C., Nalda, G., Pannocchia, C., Racioppi L., & Senosiain M. *"Oportunidades perdidas de vacunación. Participación del personal de salud"*. Arch Argent Pediatr, 97(1), 3-7.1999.

Brieschke R., Manterola A., Luppino V., Firpo V., & Debbag R. *"Oportunidades perdidas de vacunación"* Med Infant, 8(1), 23-5. 2011.

Gentile Á., Bakir J., Firpo V., Caruso M., Lución M., Abate, H., & Debbag, R. *"Esquemas atrasados de vacunación y oportunidades perdidas de vacunación en niños de hasta 24 meses: estudio multicéntrico.*" Archivos argentinos de pediatría, 109(3), 219-225. 2011.

Llop F. *"Las vacunaciones caen durante la pandemia".* Sema, 2. 2020.

Saldaña S. *"Factores sociodemográficos y culturales maternas y el cumplimiento del calendario de vacunación en niños menores de 5 años en el centro de salud las palmas-tingo maría 2017*". Perú. 2018.

Miche C. *"Factores que influyen en el incumplimiento del esquema de vacunación en niños menores de 5 años."* 2015.

Gallardo, M, Castillo, H., & Sandoval M. *"Vacunación para los estudiantes, personal y docentes de la carrera de medicina humana".* 2020.

Bustos M, Díaz M, & Soto L. *"Conocimiento de las madres sobre vacunas y su influencia en el cumplimiento del esquema de vacunación"* (Doctoral dissertation, Universidad Nacional de Cuyo. Facultad de Ciencias Médicas. Escuela de Enfermería.) 2017.

Cabrejos R. *"Factores socioculturales que influyen en el incumplimiento del calendario de vacunación en los lactantes atendidos en el centro de salud Pedro Pablo Atusparia, Chiclayo 2016*". 2017.

Alban F, & Minga M. *"Causas del incumplimiento del esquema de vacunación en niños menores de 5 años en el Centro de Salud Indio Guayas del año 2018*" (Doctoral dissertation, Universidad de Guayaquil-Ciencias Médicas-Carrera de Enfermería). 2019.

Goyez J., & Toapanta L. *"Utilidad del carnet único de vacunación como instrumento de apoyo para las madres de los niños menores de 1 año 6 meses que acuden al Area de Vacunacion del Centro de Salud Chimbo Distrito 2 Unidad Operativa 3 provincia de Bolívar periodo enero septiembre 2014*" (Bachelor's thesis, Universidad Estatal de Bolivar. Facultad de Ciencias de la Salud y del Ser Humano. Carrera de Enfermeria). 2014.

Gómez L., Muñoz D., Álvarez M., & Giraldo M. *"Sistema web para la gestión y monitoreo del plan de vacunación del municipio de Medellín-SISMOVAC."* Lámpsakos, 1(19), 13-21. 2018.

Vázquez J. *"Los orígenes de la Cartilla Nacional de Vacunación*". Acta pediátrica de México, 35(4), 257-258. 2014.

Angulo L, & Contreras A. *"Software para el control de calendarización de vacunas de niños y niñas de 0 a 4 años para la sub-área de sistema de asignación integral de la salud del hospital San José de la Provincia de Chincha."* 2013.

Gordon, C., Rezzadeh, K., & Li, A. *"Digital mobile technology facilitates HIPAAsensitive perioperative messaging, improves physician-patient communication, and streamlines patient care.*" BioMed Central, N/A. 2015.

Chaupi M. *"Propuesta para la implementación de una aplicación móvil con tarjeta de vacunación digital para llevar el registro de vacunas de niños y niñas de 0 a 5 años de la posta SASBI de Villa El Salvador."* 2020.

Vázquez M. *"Dispositivos digitales móviles en educación: el aprendizaje ubicuo*". Narcea Ediciones. España. 2015.

Guzmán M., Carrillo H., Villaseñor E., Valencia E., Calero R., Morán L., & Acosta A. *"Minería de Datos con Redes Neuronales Artificiales: Aplicación en Vacunas–Tuberculosis.*" Instituto Finlay. Centro de Investigación-Desarrollo y Producción de Vacunas y Sueros. Ave. 2004.

Salic Y. *"Propuesta para el Diseño de un Sistema de Información para el Registro de Vacunación en el Hospital Alberto Sabogal Sologuren*". 2015.