



Universidad Autónoma de Querétaro
Facultad de Informática
Doctorado en Ciencias de la Computación

Control Inteligente Optimizado Orientado a Motores Eléctricos en Sistemas
con Trayectorias Robotizadas
Tesis

Que como parte de los requisitos para obtener el Grado de
Doctorado en Ciencias de la Computación

Presenta:

Hugo Torres Salinas

Dirigido por:

Dr. Juvenal Rodríguez Reséndiz

Dr. Juvenal Rodríguez Reséndiz

Presidente

Dra. Diana Margarita Córdova Esparza

Secretario

Dra. Sandra Luz Canchola Magdaleno

Vocal

Dr. Edson Eduardo Cruz Miguel

Suplente

Dr. José Manuel Álvarez Alvarado

Suplente

Centro Universitario
Querétaro, Qro.
Julio 2023



Dirección General de Bibliotecas y Servicios Digitales
de Información



Control Inteligente Optimizado Orientado a Motores
Eléctricos en Sistemas con Trayectorias Robotizadas

por

Hugo Torres Salinas

se distribuye bajo una [Licencia Creative Commons
Atribución-NoComercial-SinDerivadas 4.0
Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Clave RI: IFDCC-136114

Agradecimientos

Es un gran honor para mí presentar mi tesis y concluir este capítulo de mi vida académica. Me gustaría expresar mi más sincero agradecimiento a todos aquellos que me apoyaron y me acompañaron en este viaje.

A mis amigos y familiares, gracias por estar a mi lado en todo momento. Gracias por su apoyo emocional y su aliento constante. Su amor y paciencia han sido una fuente de fortaleza para mí en los momentos más difíciles.

Quiero agradecer a mi director de tesis el doctor Juvenal por su orientación y mentoría a lo largo de todo este proceso. Su sabiduría, experiencia y en especial el apoyo otorgado fueron fundamentales para llevar a cabo este trabajo.

También quiero agradecer a los miembros del comité tutorial, por su tiempo, sus comentarios y sus sugerencias valiosas. Agradezco su contribución a mi formación y su ayuda en la mejora de este trabajo.

Agradezco particularmente la Facultad de Informática y en general a la Universidad Autónoma de Querétaro por permitirme llevar a cabo mis estudios de doctorado así como al CONACYT por la financiación otorgado durante todo el proceso.

En resumen, estoy agradecido con todos aquellos que han contribuido de alguna manera a mi tesis. Este logro no habría sido posible sin su ayuda. Espero poder retribuir su generosidad de alguna manera en el futuro.

Índice

Contenido

Agradecimientos.....	ii
Índice.....	iii
Índice de Tablas.....	v
Índice de Figuras.....	vi
Abreviaturas y siglas.....	viii
Resumen.....	x
Abstract.....	xi
1 INTRODUCCIÓN.....	1
1.1 JUSTIFICACIÓN Y PLANTAMIENTO DEL PROBLEMA.....	4
2 ANTECEDENTES.....	4
2.1 REVISIÓN LITERARIA.....	4
2.2 ANTECEDENTES TEÓRICOS.....	11
2.2.1 SISTEMAS DE CONTROL.....	11
2.2.2 INTELIGENCIA ARTIFICIAL.....	12
3 HIPOTESIS.....	23
4 OBJETIVOS.....	23
4.1 OBJETIVO GENERAL.....	23
4.2 OBJETIVOS PARTICULARES.....	24
5 METODOLOGÍA.....	24
5.1 TARJETA DE ADQUISICIÓN (DAQ).....	25
5.1.1 DISEÑO ESQUEMÁTICO DEL DAQ.....	30
5.1.2 PROGRAMACIÓN DEL DSPIC.....	32
5.1.3 DISEÑO DE LA TARJETA IMPRESA.....	33
5.1.4 MODELADO DE LA TARJETA DE ADQUISICIÓN.....	34
5.2 MOTOR.....	36
5.3 ENCODER.....	37
5.4 ETAPA DE POTENCIA.....	39
5.5 IMPRESORA 3D.....	42

5.6	CONTROLADOR DIFUSO	43
5.6.1	FUSIFICACIÓN	43
5.6.2	REGLAS DIFUSAS	50
5.6.3	DEFUSIFICACIÓN	51
5.7	OPTIMIZACIÓN POR ALGORITMOS GENÉTICOS.....	52
5.8	INTEGRACIÓN E IMPLEMENTACIÓN.....	55
5.8.1	CONEXIONES Y CABLEADO.....	56
5.8.2	DIAGRAMA DE FLUJO GENERAL	61
6	RESULTADOS.....	63
6.1	PRODUCTOS OBTENIDOS.....	63
6.2	RESULTADOS OBTENIDOS POR EL CONTROLADOR.....	65
7	DISCUSIÓN	69
8	CONCLUSIONES	71
9	BIBLIOGRAFÍA	72

Índice de Tablas

Tabla 1. Trabajos relevantes relacionados a la investigación propuesta.....	11
Tabla 2. Variables utilizadas en el proyecto.....	25
Tabla 3. Características de funcionamiento del motor modelo GW4058-555.....	36
Tabla 4. Descripción de los pines del IBT_2.....	41
Tabla 5. Etiquetas y expresiones lingüísticas utilizadas en la tesis.....	50
Tabla 6. Reglas difusas utilizadas en el proyecto de tesis.....	50
Tabla 7. Comparación del desempeño del controlador propuesto con técnicas similares previas.....	70

Índice de Figuras

Figura 1. Hoja de ruta de la tesis.	3
Figura 2. Comparación de un controlador difuso-PID respecto a un PID y un controlador difuso convencional (Ibrahim & Al Akkad, 2016).	7
Figura 3. Comportamiento controlador difuso (Varshney, Gupta, & Dwivedi, 2017).	8
Figura 4. Comportamiento un controlador híbrido (Camci, Kripalani, Ma, Kayacan, & Khanesar, 2018).	9
Figura 5. Diagrama de un lazo de control general.	12
Figura 6. Historia del uso de diferentes técnica de IA (Gevarter, 1987).	13
Figura 7. Diagrama de bloques de un lazo difuso general.	17
Figura 8. Diagrama de flujo general para algoritmos genéticos.	21
Figura 9. Diagrama de bloques del lazo de control propuesto para la tesis.	25
Figura 10. Representación de de PWM para una señal de 3.3 V de amplitud y 5 HZ de frecuencia.	27
Figura 11. Diagrama de pines del dsPIC utilizado en el proyecto.	29
Figura 12. Conexión del optoacoplador 4n28. (Imagen obtenida de hoja de datos).	30
Figura 13. Diseño esquemático de la conexión del dsPIC para el DAQ.	31
Figura 14. Diagrama de conexión de los optoacopladores.	32
Figura 15. Diagrama de flujo para las acciones del dsPIC.	33
Figura 16. Diseño del PCB para la tarjeta de adquisición.	34
Figura 17. Vista de la tarjeta sin componentes.	35
Figura 18. Vista de la tarjeta de adquisición en 3 dimensiones con los componentes utilizados.	35
Figura 19. Características físicas del motor utilizado.	37
Figura 20. Vista inferior del encoder e identificación de pines.	38
Figura 21. Diagrama de bloques del medio puente H. (Imagen obtenida de hoja de datos).	40
Figura 22. Pines de control en el puente H.	41
Figura 23. Conexión entre el motor y la impresora 3D.	42
Figura 24. Representación de la función triangular.	44
Figura 25. Representación de la función trapezoidal.	44
Figura 26. Representación de la función gaussiana.	45
Figura 27. Representación de la función tipo L.	46
Figura 28. Representación función tipo L.	46
Figura 29. Funciones de membresía para la variable error.	47
Figura 30. Funciones de membresía para la variable de peso.	48
Figura 31. Funciones de membresía para la variable de salida.	49

Figura 32. Representación de un individuo en forma de vector para el algoritmo genético.....	53
Figura 33. Demostración de la crucea PMX con dos individuos.....	54
Figura 34. Conexión entre el encoder y el DAQ.....	56
Figura 35. Conexión del puente H y el DAQ.	58
Figura 36. Conexión entre el puente H y el motor.	59
Figura 37. Conexión entre la computadora y el DAQ.	60
Figura 38. Diagrama de bloques para la conexión de los elementos en el proyecto.	61
Figura 39. Diagrama de flujo general de trabajo para el proyecto.	62
Figura 40. Imagen del DAQ real.	63
Figura 41. Plataforma de experimentación real utilizada para el proyecto.	64
Figura 42. Desempeño del controlador en el escenario 1 y escenario 3.	65
Figura 43. Errores obtenidos en el escenario 1 y escenario 3.....	66
Figura 44. Señal de control calculada para escenario 1 y escenario 3.....	66
Figura 45. Desempeño del controlador en el escenario 2 y escenario 4.	67
Figura 46. Errores obtenidos en el escenario 2 y escenario 4.....	67
Figura 47. Señal de control calculada para escenario 2 y escenario 4.....	68
Figura 48. Superficie de control obtenida para el controlador difuso.....	68

Abreviaturas y siglas

PID: *Proportional-Integrative-Derivativo*. (Controlador) Proporcional integral derivativo.

AG: Algoritmo genético.

IoT: *Internet of Things*. Internet de las cosas.

ANN: *Artificial Neural Networks*. Redes neuronales artificiales (RNA).

ACO: *Ant Colony Optimization*. Optimización por colonia de hormigas.

SVR: *Support Vector Regression*. Regresión por vectores de soporte.

PSO: *Particle Swarm Optimization*. Optimización por cúmulo de partículas.

IA: Inteligencia artificial.

ANFIS: *Artificial Neural Fuzzy Inference System*. Sistema de inferencia difusa neuronal artificial.

ADC: *Analogic-Digital Converter*. Convertidor analógico digital.

PWM: *Pulse-Wide-Modulation*. Modulación por ancho de pulso

RPM: Revoluciones por minuto.

PIC: *Pheriperal-Interface Controller*. Controlador de interfaz periférica.

CTS: *Clear to send*. Limpiar para enviar.

RTS: *Request to send*. Petición para enviar.

FIFO: *First input first output*. Primera entrada primera salida.

MIMO: *Multiple input multiple output*. Múltiple entrada múltiple salida.

SISO: *Single input single output*. Una entrada una salida.

UART: *Universal Asynchronous Receiver-Transmitter*. Receptor-transmisor asíncrono universal.

DAQ: *Data-Aquisition*. Adquisición de datos.

IDE: *Integrated development environment*. Ambiente de desarrollo integrado.

PCB: *Printed circuit board*. Placa de circuito impresa.

PMX: *Partial mapped crossover*. Cruza de mapeo parcial.

Resumen

El control se utiliza en procesos donde se requiera que una variable siga una referencia. Métodos convencionales como controladores PID son ampliamente preferidos debido a la gran variedad de herramientas que existen para su implementación, sin embargo presentan limitaciones en sistemas variantes en el tiempo o que presenten comportamientos no lineales por lo que se han desarrollado técnicas que combinen controladores clásicos con algoritmos de inteligencia artificial para mejorar el desempeño del sistema. El objetivo de este trabajo de tesis es desarrollar un sistema de servo control el cual utiliza lógica difusa para controlar la posición en una plataforma de experimentación compuesta por un motor eléctrico, la etapa de potencia para accionar este motor, la etapa de adquisición de datos y finalmente la estructura de una impresora 3D. La relevancia de este trabajo es que se agrega inteligencia artificial extra para optimizar los parámetros en la lógica difusa mediante un algoritmo genético. Para someter al controlador ante distintas condiciones se plantearon 4 escenarios de experimentación en donde se modifica la trayectoria y la carga agregada a la impresora 3D. Los resultados obtenidos en cada uno de los experimentos mostraron que utilizando el controlador propuesto se mejoró el tiempo de subida y el tiempo de asentamiento del sistema cuando son comparados con los que se obtuvieron en investigaciones previas con técnicas similares. La investigación demostró que el método propuesto alcanza los parámetros de control deseados ante modificaciones en las condiciones iniciales en la plataforma de experimentación.

Abstract

The control is used in processes where a variable is required to follow a reference. Conventional methods such as PID controllers are highly preferred due to the wide variety of tools that exist for their implementation, however they have limitations in time-varying systems or that have non-linear behaviors, so techniques have been developed that combine conventional controllers with algorithms based on artificial intelligence to improve system performance. The objective of this thesis work is to develop a servo control system which uses fuzzy logic to control the position of an experimentation platform consisting of an electric motor, the power stage to drive this motor, the data acquisition stage, and finally the structure of a 3D printer. The relevance of this work is that additional artificial intelligence is added to optimize the parameters in the fuzzy logic using a genetic algorithm. To subject the controller to different conditions, 4 experimentation scenarios were proposed where the trajectory and the added load to the 3D printer were modified. The results obtained in each of the experiments showed that by using the proposed controller, the rise time and settling time of the system were improved when compared to those obtained in previous research with similar techniques. The research demonstrated that the proposed method achieves the desired control parameters when modifications are made to the initial conditions on the experimentation platform.

1 INTRODUCCIÓN

Los controladores de posición se implementan en sistemas donde se requiere llegar a un punto específico o seguir una ruta predefinida dentro de un espacio de trabajo. El principal objetivo de regular la posición es realizar desplazamientos controlados dentro del sistema, reduciendo así la diferencia entre el valor deseado y el valor actual de la variable de interés. Entre las muchas aplicaciones de estos controladores se encuentran cintas transportadoras en sistemas de fabricación, soldadoras industriales, etiquetado automático de productos y sistemas robóticos.

Los controladores PID son los preferidos en la industria debido a su robustez, simplicidad de diseño e implementación (Abushawish, Hamadeh, & Nassif, 2020; Levine, 1996; Y. Zhang, Huang, & Wang, 2022). A pesar de lo anterior, estas técnicas presentan inconvenientes que muestran un bajo rendimiento en sistemas donde el modelo matemático contiene un integrador o una función de transferencia inestable. Son sensibles a las perturbaciones y resonancias externas que se producen durante el proceso (Rodriguez-Abreo, Rodriguez-Resendiz, Fuentes-Silva, Hernandez-Alvarado, & Falcon, 2021; Rodriguez-Abreo, Rodriguez-Resendiz, et al., 2021). Para resolver los problemas antes mencionados se han desarrollado estrategias de control que combinan métodos convencionales con inteligencia artificial, como lógica difusa, algoritmos genéticos y redes neuronales (Latah & Toker, 2019). Estas tecnologías híbridas han demostrado ser efectivas, mejorando significativamente el rendimiento de los sistemas, reduciendo el tiempo necesario para alcanzar el valor deseado y reduciendo los costos de energía.

La plataforma de prueba consiste en una base de impresora 3D comercial, cuyo actuador principal ha sido cambiado por un motor de corriente continua. También es necesario añadir los elementos de adquisición de datos y la etapa de potencia. La experimentación está diseñada en cuatro escenarios donde se modifica tanto la trayectoria como el peso que debe mover la impresora, verificando así la eficiencia

del controlador cuando se modifican las condiciones iniciales. Los resultados mostraron que la técnica aquí propuesta logra alcanzar los valores de posición deseados en todos los escenarios en los que fue probada. Además, estos resultados se comparan con trabajos previos en la literatura que utilizan técnicas similares y se encontró que el desempeño del controlador mejora los tiempos de subida y de asentamiento. Si bien existen trabajos previos con el mismo enfoque, la propuesta presentada ofrece los siguientes aportes:

- Un controlador difuso que utiliza un algoritmo genético para optimizar sus funciones de pertenencia.
- Una plataforma de experimentación real que fomenta el conocimiento en electrónica y programación.
- Buenos resultados de control en términos de tiempo de subida y de estabilización y sobrepaso.
- El controlador se adapta a las variaciones de las condiciones iniciales.
- Mejor comportamiento en comparación con técnicas similares utilizadas previamente en la literatura.

La estructura de este trabajo contiene las secciones: Introducción; los antecedentes brindan una revisión literaria de trabajos similares y además una fundamentación teórica de algoritmos genéticos y lógica difusa; hipótesis; objetivos; la metodología muestra el desarrollo de la plataforma de experimentación así como del controlador híbrido; resultados; discusión y finalmente las conclusiones. La **Figura 1** muestra la hoja de ruta que ayuda a visualizar cómo se divide el trabajo.

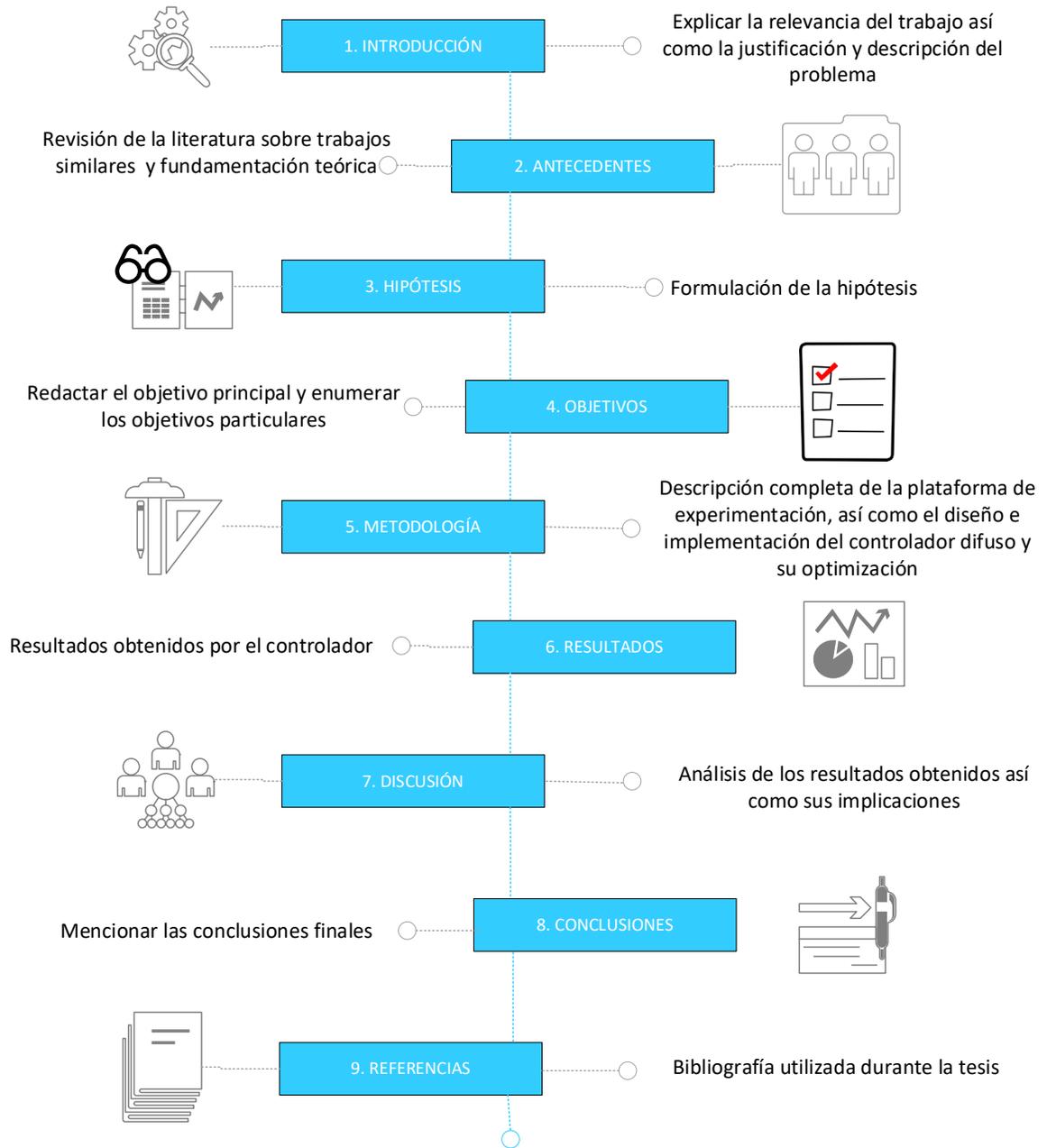


Figura 1. Hoja de ruta de la tesis.

1.1 JUSTIFICACIÓN Y PLANTAMIENTO DEL PROBLEMA

La motivación para realizar la investigación presentada en este artículo radica en encontrar una solución que utilice las técnicas antes mencionadas para alcanzar los valores deseados en procesos donde se modifican las condiciones iniciales. El objetivo principal es el diseño de un servo controlador de posición y su implementación en una plataforma de experimentación real consistente en una impresora 3D. El problema que se plantea es que las ganancias de los controladores tradicionales permanecen fijas incluso cuando cambian las condiciones del proceso para el que fueron ajustados. Por tanto, la principal aportación que se propone en este trabajo es una técnica mediante la cual se mejora el comportamiento de los controladores difusos utilizando inteligencia artificial, concretamente un AG, para optimizarlos. Cada individuo en el AG se compone de los parámetros de las funciones de pertenencia del controlador difuso, buscando los mejores valores para ellos. Por tanto, estos valores se modifican si también se modifican las condiciones del sistema. Además, también es posible reducir los errores humanos en el proceso de fusificación.

2 ANTECEDENTES

2.1 REVISIÓN LITERARIA

Esta sección menciona investigaciones previas relacionadas con los métodos de control utilizados para resolver problemas relacionados con sistemas inestables, perturbaciones y comportamiento no lineal (Lu et al., 2022; Tadic et al., 2021). En la literatura se han desarrollado diferentes técnicas para resolver los problemas de los controladores clásicos utilizando fenómenos recientes como el IoT. Estas técnicas han dado resultados sobresalientes ya que el IoT permite conectar diferentes sensores y actuadores, logrando el diseño de sistemas más completos (Chegini & Mahanti, 2019; Chegini, Naha, Mahanti, & Thulasiraman, 2021). Se han presentado

propuestas novedosas que buscan eliminar activamente las perturbaciones externas en el sistema mediante el uso de controladores no frágiles, que consideran varios problemas conocidos dentro de su diseño, entre los que se encuentran la incertidumbre de los parámetros del modelo, fallas en los elementos del sistema y errores de medición. Estos controladores no frágiles han demostrado superar cada uno de los obstáculos anteriores (Liu, Yue, & Yang, 2021).

Trabajos recientes han demostrado que la lógica difusa mejora el desempeño de los controladores convencionales en presencia de perturbaciones y comportamiento no lineal (Baier-Fuentes, Cascón-Katchadourian, Sánchez, Herrera-Viedma, & Merigó, 2018; Odry, Tadic, & Odry, 2021; Rodriguez-Abreo, Hernandez-Paredes, Rangel, Fuentes-Silva, & Velasquez, 2021). Se ha encontrado que la acción de los actuadores se puede reducir hasta en un 50 % ejecutando un controlador difuso en lugar de un controlador PID, lo que reduce el consumo de energía y los costos asociados (Mahapatra, Daniel, Dey, & Nayak, 2015; Yang, Wang, & Fan, 2022). Los investigadores han utilizado la lógica difusa tanto para identificar los parámetros del sistema como para modificar las ganancias del controlador porque su implementación es relativamente simple. Debido a lo anterior, la lógica difusa se ha convertido en una de las técnicas preferidas utilizadas en procesos donde se requiere control (Chegini, Beltran, & Mahanti, 2021; Xiao-Jun Ma, Zeng-Qi Sun, & Yan-Yan He, 1998). Sin embargo, también tiene varias limitaciones, como la dificultad de interpretar valores difusos. Requiere múltiples reglas y, finalmente, se necesita una persona que tenga un alto grado de conocimiento sobre el comportamiento del sistema. La última limitación mencionada es importante ya que la posibilidad de errores humanos en el proceso de diseño del controlador es alta.

Los controladores híbridos se han desarrollado para superar las limitaciones de la lógica difusa. Estos controladores agregan algoritmos adicionales que han

demostrado ser efectivos y han mejorado el rendimiento del sistema en múltiples objetivos (Veselý & Ilka, 2013). La inteligencia artificial ha desempeñado un papel esencial en la creación de técnicas híbridas, incluidos los algoritmos ANN, ACO, SVR y PSO. Estos algoritmos han obtenido grandes mejoras en aspectos como suavizar las trayectorias de los actuadores, reducir el consumo energético y conseguir tiempos de funcionamiento más cortos (Jain, Mahanti, Shamsolmoali, & Manikandan, 2020; Panda & Padhy, 2008; Rodríguez-Abreo, Rodríguez-Reséndiz, Montoya-Santiyanes, & Álvarez-Alvarado, 2021; Tiwari, Bhatt, Unni, Singh, & Ongsakul, 2018; Young-Moon Park, Myeon-Song Choi, & Lee, 1996).

Investigaciones recientes combinan el controlador clásico PID con lógica difusa logrando alcanzar mejores resultados, puesto que se realizan trayectorias más suaves mediante la reducción de vibraciones, haciendo que el error en estado alcance el valor de cero, y además reduciendo los tiempos de asentamiento y de subida comparándolo con controladores que trabajen de manera individual.

En un estudio previo se implementó un control híbrido combinando lógica difusa y un controlador PID convencional entregando mejores resultados. La técnica propuesta redujo los daños que pueden sufrir los sistemas debido a que se eliminan vibraciones en el motor, se reduce el error en estado estacionario lo que mejora el tiempo de respuesta que muestran los controladores difusos y los controladores PID cuando se utilizan individualmente. La **Figura 2** muestra la comparación entre un controlador difuso, un controlador PID y un controlador difuso-PID. Estos resultados fueron obtenidos en un trabajo realizado en 2016 donde se aprecia que el comportamiento del controlador híbrido mejora considerablemente en cuanto a la reducción de oscilaciones y tiempo de asentamiento.

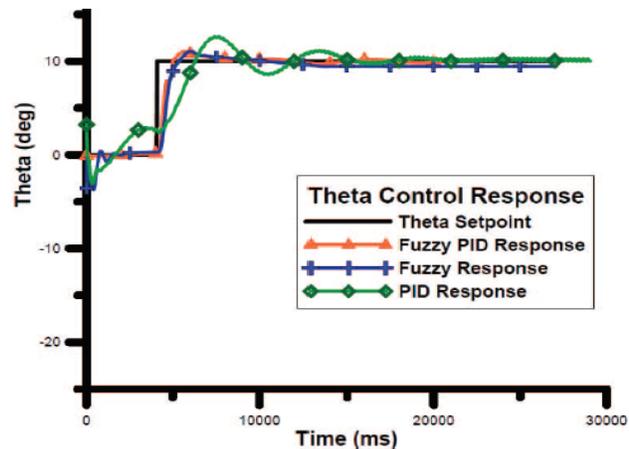


Figura 2. Comparación de un controlador difuso-PID respecto a un PID y un controlador difuso convencional (Ibrahim & Al Akkad, 2016).

Técnicas similares en las que se combina controladores PID y lógica difusa se presentaron en un trabajo de 2017 en el cual se pone a prueba el desempeño del controlador en un motor de corriente directa sin escobillas. Los resultados obtenidos por este trabajo demostraron que se alcanza un mejor rendimiento si la carga en el motor se modifica de manera gradual. En adición a lo anterior se encontró que la respuesta transitoria del sistema, específicamente el sobrepaso, error en estado estacionario y tiempo de subida mejoran con el uso de un PID difuso. El trabajo presentado en 2018 comprobó que las sacudidas y tirones experimentadas por los motores reducen en un 50% en el caso de la eliminación repentina de la carga y en aproximadamente el 80% en el caso de la eliminación gradual de la carga mediante la implementación del controlador difuso en comparación de un controlador PID convencional. La **Figura 3** muestra la comparación realizada por los autores del trabajo descrito.

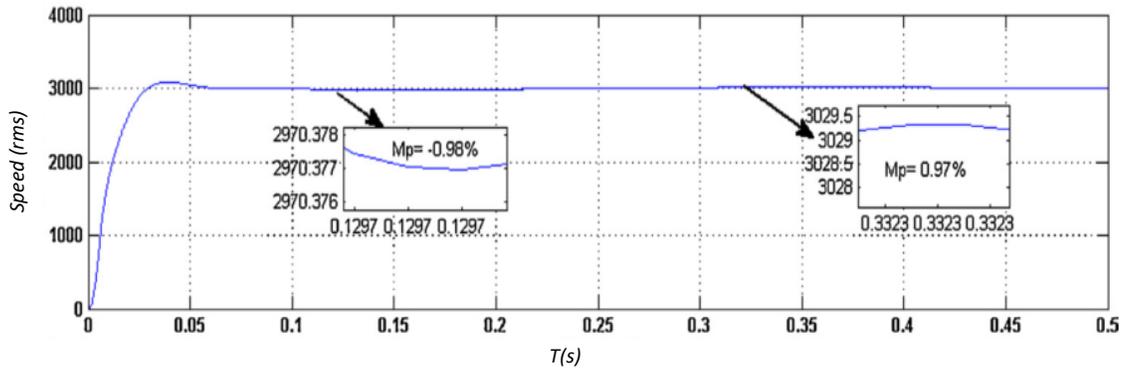


Figura 3. Comportamiento controlador difuso (Varshney, Gupta, & Dwivedi, 2017).

Se han presentado variaciones en las técnicas encontradas en la literatura. En 2017 se utilizó un controlador difuso tipo 2 con la intención de lograr un seguimiento de trayectorias en un espacio definido. Los autores concluyeron que el esquema propuesto es más robusto y efectivo que el controlador PID convencional para el seguimiento de la trayectoria. La técnica propuesta fue eficaz para reducir los efectos indeseados debido a las incertidumbres en el modelo matemático así como de las perturbaciones externas. Por lo tanto, el diseño del controlador propuesto también se puede usar en industrias donde se requieran movimientos precisos tales como soldadura y pintura en donde los manipuladores robóticos están bajo un entorno en donde ocurren grandes variaciones de parámetros debido a perturbaciones externas. La investigación muestra que el controlador es capaz de ajustar los parámetros en modelos matemáticos no lineales y con alta susceptibilidad a variaciones iniciales como por ejemplo en plantas nucleares (Kumar & Rana, 2017).

Técnicas más complejas han sido desarrolladas las cuales tienen la ventaja de reducir el error y las oscilaciones presentadas por el sistema. En contraparte aumentan el costo computacional y el tiempo de procesamiento. Ejemplos de estos controladores híbridos se presentaron en 2018 en el cual se implementa una

combinación de lógica difusa, PID y redes neuronales para regular la posición en un cuadricóptero. Los autores comprobaron que su propuesta redujo el ruido y las incertidumbres en la medición del seguimiento de trayectoria obteniendo movimientos más suaves y precisos en comparación con controladores PID convencionales y PID difusos. Este resultado prueba de que este tipo de controladores son una opción para ser implementados en procesos en tiempo real en donde la influencia del ruido es inevitable. La **Figura 4** muestra los resultados obtenidos.

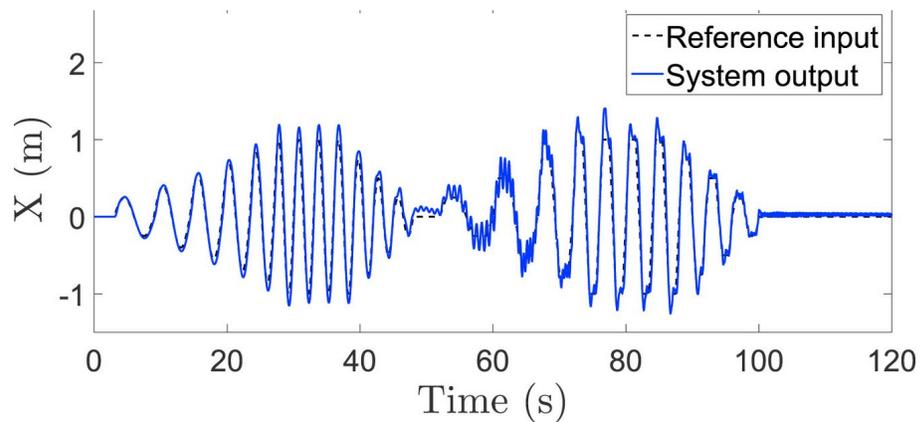


Figura 4. Comportamiento un controlador híbrido (Camci, Kripalani, Ma, Kayacan, & Khanesar, 2018).

Previamente se han presentado propuestas en las que se diseña un controlador híbrido con lógica difusa y AG, las cuales han demostrado lograr los objetivos deseados en los sistemas y mejorar su desempeño frente a comportamientos no lineales (Masoudi, Soltanpour, & Abdollahi, 2018; Sierra-García & Santos, 2021; Tabrez et al., 2021). Estas técnicas también han demostrado ser efectivas para contrarrestar perturbaciones y cambios en las condiciones iniciales del proceso. Estos logros le permiten al controlador modificar sus ganancias si hay cambios en

el proceso original (Odeh, Mora, Moreno, & Merelo, 2015; Ponticelli, Guarino, Tagliaferri, & Giannini, 2019; Rath et al., 2019). Como se puede apreciar, los métodos anteriores obtuvieron excelentes resultados y son muy similares al propuesto en este artículo. Sin embargo, la principal motivación que inspiró esta investigación es el deseo de diseñar un controlador e implementarlo en una plataforma de experimentación real, a diferencia de trabajos anteriores que prueban su desempeño a través de simulaciones. Además, nuestra propuesta proporciona tiempos de subida y bajada competitivos en comparación con otras técnicas. La

Tabla 1 muestra una recopilación con trabajos relacionados que son relevantes en la literatura por su aporte y técnica utilizada.

Referencia	Técnica	Aporte
(Mahapatra et al., 2015)	PSO-ANFIS	Los resultados mostraron que una técnica híbrida donde se utiliza un algoritmo PSO y ANFIS es más eficiente al regular la velocidad de un motor, su comportamiento dinámico y costo computacional.
(Hernández-Alvarado, 2016)	ANN	Los autores demostraron que la sintonización automática un controlador PID a través de ANN mejora el desempeño y reduce el costo de energía al posicionar un mini-robot acuático.
(Demir, Bayir, & Duran, 2016)	Lógica difusa	Este trabajo presenta un controlador PID autoajutable en tiempo real a través de lógica difusa. Los resultados han mostrado estabilizar el sistema, controlar la altitud y seguir la trayectoria de un quadrotor.
(Suman & Giri, 2016)	Algoritmo genético	La técnica utilizada actualiza las ganancias de un controlador PID mediante algoritmos genéticos. Los

		resultados mostraron un mejor rendimiento al reducir el sobreimpulso y el error de estado estable cuando se regula la velocidad.
(Jigang, Jie, & Hui, 2017)	Lógica difusa	Este trabajo regula la velocidad de un motor al combinar un controlador PID y uno difuso. Los resultados mostraron que se logra que el motor funcione más suavemente.

Tabla 1. Trabajos relevantes relacionados a la investigación propuesta.

2.2 ANTECEDENTES TEÓRICOS

Esta sección del documento otorga los fundamentos teóricos necesarios para desarrollar el proyecto planteado para la tesis. Se describe de manera breve los conceptos que se retoman más adelante en la implementación.

2.2.1 SISTEMAS DE CONTROL

En la **Figura 5** se muestra el diagrama de bloques de un lazo genérico de control donde “ r ” representa el valor deseado, “ y ” corresponde al valor real o actual, “ e ” representa el error y la señal de control se identifica con la letra “ u ”. El sistema a controlar está conformado por un actuador que en este caso es un servomotor y una planta que es la estructura de la impresora 3D, lo anterior se muestra en la parte punteada dentro del diagrama. Para el diseño del controlador generalmente se realiza la identificación del sistema, obteniendo el modelo matemático de ésta. Como se han mencionado anteriormente se han desarrollado técnicas que facilitan

la obtención del modelo, llegando incluso en algunos casos a prescindir de éste. La inteligencia artificial es el componente principal para lograr sustituir al control convencional por lo que se otorga un breve contexto teórico de esta.

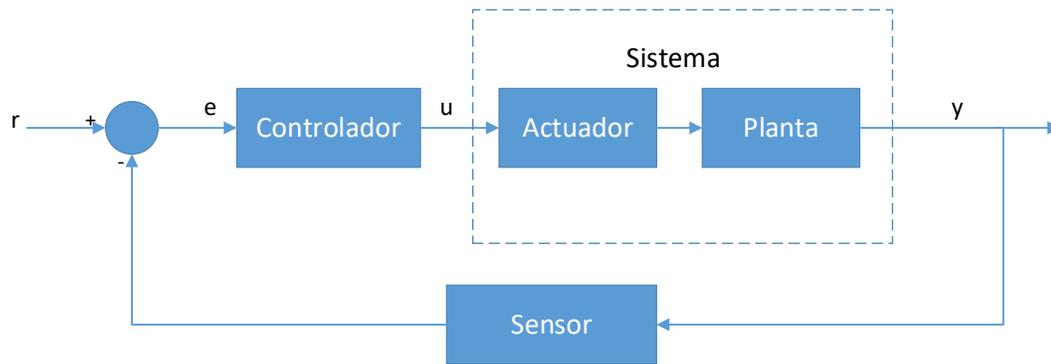


Figura 5. Diagrama de un lazo de control general.

2.2.2 INTELIGENCIA ARTIFICIAL

La inteligencia artificial tiene el objetivo de entender el concepto de inteligencia y el de construir sistemas inteligentes. Actualmente resulta difícil encasillar la definición a un concepto robusto ya que existen una multitud de subdisciplinas que están contenidas en la definición de inteligencia artificial. Se debe empezar por la definición de inteligencia, la cual en sí misma es difícil de definir, se considera a la inteligencia a la capacidad de tomar decisiones para mejorar algún proceso, método, situación que influyan en el resultado final de éste (Gevarter, 1987). La segunda parte que es la definición de artificial, consiste en algo que no es creado de manera natural, es decir que el ser humano influye para su desarrollo.

Elaine Rich definió de manera elegante a la inteligencia artificial como “el estudio sobre cómo lograr que las computadoras hagan cosas en las cuales, al momento, las personas son mejores”.

Por la definición la inteligencia artificial incluye una gran variedad de estudios los cuales se pueden ver en la **Figura 6**.

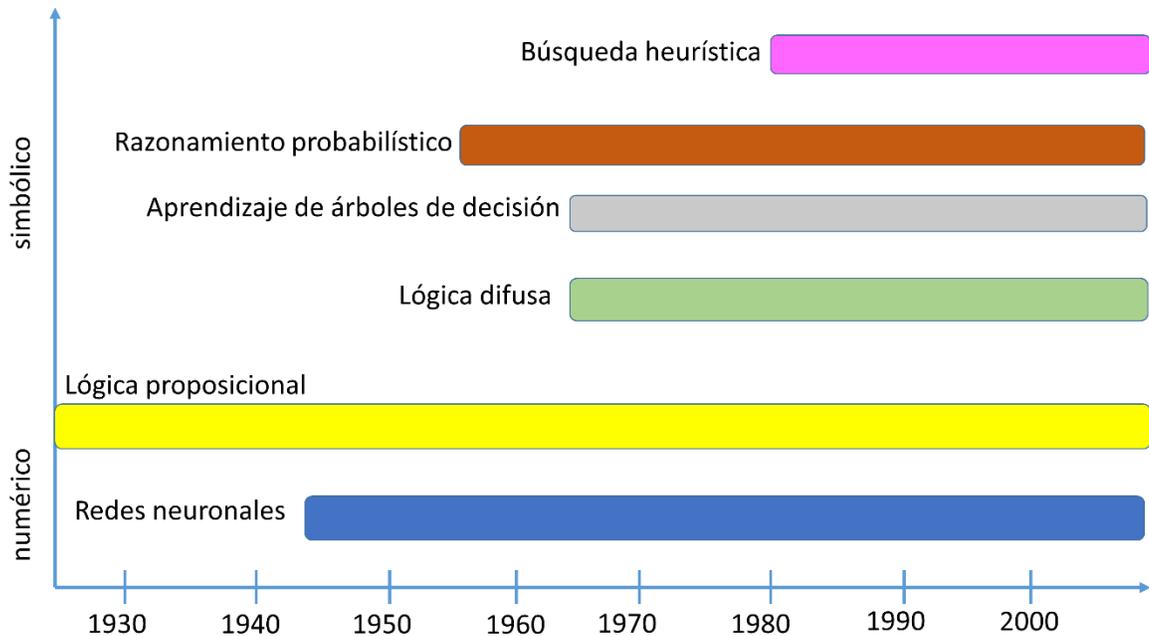


Figura 6. Historia del uso de diferentes técnica de IA (Gevarter, 1987).

2.2.2.1 Lógica difusa

El padre de la lógica difusa, Lofti A. Zadeh, presentó el trabajo "conjuntos borrosos" en 1965, en el que se definía la pertenencia de los objetos a través de clases (Zadeh, 1965). Los conjuntos borrosos permitían clasificar a las personas en conjuntos como 'la clase de los hombres bajos' o 'la clase de las mujeres jóvenes' y, por esta razón, Zadeh consideró que los conjuntos que definió estaban más cerca del razonamiento humano, principalmente en lo que respecta a cuestiones de comunicación o abstracción de información y reconocimiento de patrones. Otras ventajas que se encontraron con esta técnica son las siguientes:

- Los identificadores difusos son sencillos de implementar.
- Se pueden agregar múltiples entradas y salidas a los sistemas a diferencia del control convencional.

- Tiene un costo computacional más bajo que otras técnicas de inteligencia artificial.
- Debido a un menor número de operaciones el cálculo señal de control es menor.

Para predecir el estado futuro del sistema y de esta manera ajustar los parámetros del controlador no es necesario contar con un modelo matemático exacto pero se requiere el funcionamiento detallado (Amendola & Gonzaga, 2007). Los dos tipos de sistemas de inferencia difusos más ampliamente implementados son los llamados métodos Mamdani (Mamdani, 1974) y Takagi--Sugeno--Kang (TSK) (Takagi & Sugeno, 1985). En este artículo, se selecciona el método Mamdani, que puede expresarse mejor como un control que emplea un lenguaje natural que trabaja con expresiones, en lugar de números u oraciones en lugar de ecuaciones. Sin embargo, las variables del proceso no se miden en lenguaje natural, sino con números. Por lo tanto, es necesario realizar una transformación antes de ingresar el estado de la variable al controlador; esta etapa se llama fusificación (Kouro & Musalem, 2002). Luego, el controlador difuso, de acuerdo con una base de reglas, realiza una acción que también es una forma de expresión lingüística que finalmente se convierte nuevamente en un número en la etapa de defusificación (Rafa et al., 2014).

Los conceptos importantes para comprender el proceso de lógica difusa se describen a continuación:

- El universo de discurso se define como el conjunto X de todos los valores posibles que puede tomar una determinada variable x .
- Un conjunto borroso es un conjunto que contiene elementos parcialmente, esto significa que la capacidad de que un elemento x pertenece al conjunto A , ($x \in A$) es verdadera solo con un porcentaje de verdad.

- Un conjunto nítido es el conjunto convencional y solo considera si un elemento le pertenece o no, lo que significa que es un conjunto binario.
- Sea A un conjunto difuso y sea $(x \in X)$ un valor del conjunto universal; la función de membresía también conocida como función de pertenencia descrita en la Ecuación (1), indica el grado de pertenencia de dicho valor a un conjunto borroso.

$$\mu A(x) : X \rightarrow [0,1] \quad (1)$$

Por lo tanto un conjunto difuso es definido en la Ecuación (2).

$$A = (x, \mu A(x)) : x \in X, \mu A(x) : X \rightarrow [0,1] \quad (2)$$

Operaciones con conjuntos difusos

Ciertas operaciones se pueden realizar entre este tipo de conjuntos. Los principales son la igualdad, la inclusión, la unión, la intersección y el complemento. Estas operaciones funcionan de manera similar a las operaciones en conjuntos clásicos. La operación de igualdad descrita en la Ecuación (3) establece que dos conjuntos borrosos A y B , definidos en el mismo universo X , son iguales si su función de membresía es la misma. La inclusión establece que, dados dos conjuntos borrosos A y B , A es un subconjunto de B ($A \subseteq B$) si su función de membresía toma valores más pequeños. Esto se describe en Ecuación (4).

$$\mu A(x) = \mu B(x), \quad \forall x \in X \quad (3)$$

$$\mu A(x) \leq \mu B(x), \quad \forall x \in X \quad (4)$$

Para la operación de unión se toma la forma generalizada, que es una conorma triangular, la conocida T-conorma (\perp), y se puede expresar de acuerdo a la Ecuación (5).

$$\mu_{A \cup B}(x) = \perp(\mu_A(x), \mu_B(x)) \quad (5)$$

La T-conorma debe satisfacer las siguientes propiedades $\forall a, b, c, d \in [0, 1]$:

Conmutatividad: $\perp(a, b) = \perp(b, a)$.

Monotocidad: $\perp(a, b) < \perp(c, d)$ if $a \leq c$ and $b \leq d$.

Elemento identidad: $\perp(a, 0) = a$.

Asociatividad: $\perp(a, \perp(b, c)) = \perp(\perp(a, b), c)$.

La T-conorma que corresponde a conjuntos difusos es el máximo por lo que se reescribe la Ecuación (5) y se obtiene la Ecuación (6).

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (6)$$

La operación intersección se describe en la Ecuación (7), donde T es la norma triangular, mejor conocida como la T-norma.

$$\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x)) \quad (7)$$

De la misma manera la T-norma debe satisfacer las siguientes propiedades $\forall a, b, c, d \in [0, 1]$:

Conmutatividad: $T(a, b) = T(b, a)$.

Monotocidad: $T(a, b) < T(c, d)$ if $a \leq c$ and $b \leq d$.

Elemento identidad: $T(a, 0) = a$.

Asociatividad: $T(a, T(b, c)) = T(T(a, b), c)$.

La T-norma utilizada en conjuntos difusos es el mínimo. La Ecuación (8) se obtiene una vez que la T-norma es sustituida en Ecuación (7).

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (8)$$

Para la operación de complemento ($\mu_{\bar{A}}$), los siguientes axiomas se deben satisfacer.

μ_A es una función continua.

Monotonocidad: si $a \leq b$, entonces $\mu_A \geq \mu_B$.

El complemento puede ser definido de la misma manera que en los conjuntos clásicos y se describe en la Ecuación (9).

$$\mu_{\bar{A}} = 1 - \mu_A(x) \quad (9)$$

Un lazo de control difuso genérico se muestra en la **Figura 7**. Debido a que se trabaja con computadoras o dispositivos digitales es necesario la utilización de un convertidor analógico-digital para transformar la señal que se obtiene de la planta. Una vez que se realizan las operaciones del controlador la salida de éste vuelve a ser transformada para regresar a la planta, para lo anterior se utiliza un convertidor digital-analógico (Rafa et al., 2014). Las tres etapas mostradas en el lazo difuso con descritas a continuación.

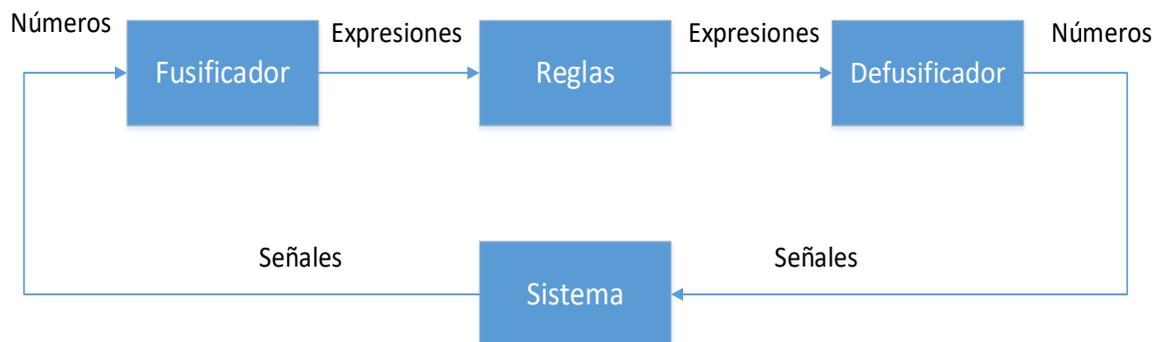


Figura 7. Diagrama de bloques de un lazo difuso general.

Fusificación

La etapa de fusificación transforma un número en una expresión lingüística previamente definida. Cada número en esta etapa se asigna a las funciones de

membresía descritas en esta etapa, lo que otorga un grado de membresía a una o más de las funciones de membresía que pueden tener forma triangular, trapezoidal o gaussiana, según la selección del usuario. El grado de pertenencia está en el rango de 0 a 1, y el número puede asignarse a más de una función que tenga diferentes rangos con diferentes grados cada uno.

Reglas

La lógica difusa utiliza reglas para la toma de decisiones. Estas reglas se componen de entradas difusas, también conocidas como precedentes, y salidas, también llamadas consecuencias. Los precedentes y las consecuencias están asociados con declaraciones si-entonces (Mejía Ramírez et al., 2021).

Las reglas difusas generalmente se agrupan en tablas, aunque la representación puede ser difícil cuando las variables de entrada son más de dos. Las diferentes variables de entrada se asocian mediante los operadores lógicos *and*, *or* y *not*. Los dos grupos principales de reglas difusas son las reglas de Takagi-Sugeno y las reglas de Mandani. Cada uno de estos grupos tiene diferentes ventajas, y la principal diferencia entre ellos es que la salida de la inferencia difusa en el método Mandani sigue siendo una variable difusa, mientras que en las reglas de Takagi-Sugeno, la variable de salida ya es un valor nítido (Dhimish, Holmes, Mehrdadi, & Dales, 2018).

Defusificación

Dado que la salida del motor de inferencia sigue siendo una expresión lingüística, es necesario volver a convertirla en un número. Así, se lleva a cabo el proceso de defusificación, que permite asociar un valor numérico a un conjunto borroso realizado para calcular el valor de salida de los sistemas difusos. El motor de inferencia difusa concluye la información de entrada pero se expresa en variables difusas. Esta conclusión o salida difusa se obtiene mediante las reglas borrosas, pero el valor de salida del sistema debe ser un número real y tiene que ser representativos de todo el conjunto, por lo que existen diferentes métodos de

defusificación. Para este proyecto, se utiliza el método del centroide, que es uno de los métodos preferidos para implementar. Este proceso, por lo tanto, realiza el procedimiento inverso a la etapa de fusificación

2.2.2.2 Algoritmos de optimización

La optimización es el procedimiento de hacer que algo sea mejor (Martins, Santos, & Santos, 2012). En otras palabras, la optimización es el proceso de manipular las entradas en un sistema para encontrar un rendimiento ideal. La entrada está conformada por variables y la salida es representada por el costo o la aptitud obtenida por cada solución encontrada.

Los algoritmos de optimización pueden estar inspirados en la naturaleza, como PSO y ACO que imitan el comportamiento presente en insectos para lograr objetivos de aumentar o disminuir el valor deseado del proceso (Rezaee Jordehi, Jasni, Abd Wahab, Kadir, & Javadi, 2015), (Kaveh, Bakhshpoori, & Afshari, 2014); mientras que los algoritmos genéticos imitan la reproducción genética natural para mejorar comunidades por medio de pasar características llamadas genes a través de distintas generaciones de individuos (Orozco & Montoya, 2010).

Algoritmos genéticos

Un algoritmo genético es un método que se utiliza principalmente para resolver problemas de optimización. Están inspirados en el comportamiento biológico de los seres vivos, específicamente en su reproducción. El objetivo de los AG es imitar la evolución biológica para obtener mejores soluciones en mejores tiempos. En general, los AG entran en la categoría de inteligencia artificial debido a que resuelven problemas a través de algoritmos computacionales que imitan el comportamiento de la inteligencia natural (Liang, Zou, Zuo, & Khan, 2020).

Los AG se presentan como una técnica de optimización de búsqueda global. Exploran una gran variedad de posibles soluciones para el problema que intentan

resolver y, por tanto, consiguen evitar soluciones locales óptimas para ir en busca de soluciones globales óptimas (Mirjalili, Mirjalili, Saremi, Faris, & Aljarah, 2018).

Los siguientes términos describen conceptos básicos para problemas de optimización.

- La función objetivo es una función que busca minimizar o maximizar su valor y también puede contener múltiples variables.
- Una solución local óptima se define como el máximo o mínimo solo para una región delimitada de todo el espacio de solución disponible.
- La solución global óptima representa el máximo o mínimo para todo el espacio de solución de la función objetivo.

Los AG trabajan con una población de individuos o un conjunto de posibles soluciones al problema. Este conjunto está sujeto a acciones aleatorias, similares a las que ocurren en la evolución biológica, para mejorar la población. Las mutaciones o recombinaciones genéticas se encuentran entre estas acciones. La evolución también ocurre cuando se hace una selección de acuerdo a un criterio que establece cuáles son los individuos más aptos, cuáles sobrevivirán, y cuáles son los menos aptos, lo cual se descarta (Liang et al., 2020). En la **Figura 8** se muestra un diagrama de flujo de las etapas de un algoritmo genético convencional. Y cada etapa se describe a continuación.

Inicializar población

Se genera aleatoriamente una población inicial que contiene posibles soluciones al problema. Estas soluciones se denominan individuales. Cada individuo debe estar codificado en un cromosoma, a su vez este cromosoma está formado por genes que representan los parámetros del proceso que se busca optimizar. Para que una computadora entienda los cromosomas, deben estar codificados en una cadena; es decir, una sucesión de letras, números o una combinación de ambos.

La correcta elección de la codificación es uno de los puntos críticos para obtener una excelente solución al problema en estudio. Por lo tanto, existen varias formas de llevar a cabo este proceso, como la codificación binaria, la codificación mediante caracteres o utilización de números enteros (T. Zhang, Liu, Rao, Li, & Zhao, 2020).

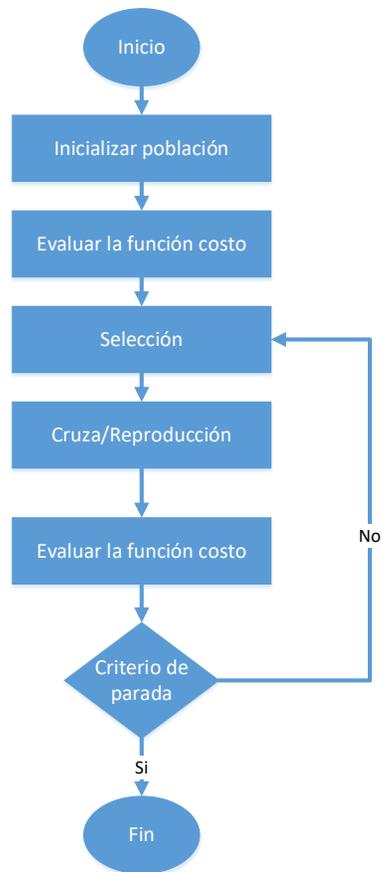


Figura 8. Diagrama de flujo general para algoritmos genéticos.

Evaluar la función objetivo.

Cada uno de los individuos generados anteriormente es evaluado por la función objetivo, generalmente una ecuación matemática. Esta función objetivo otorga una puntuación a los individuos que aportan las mejores soluciones. Aunque la mayoría

de estas soluciones no funcionan, algunas pueden ser favorables, pueden mostrar parte de la solución, incluso si es pequeña e imperfecta (Chin & Lin, 2018).

Selección.

Una vez que los individuos han sido probados, la AG debe decidir cuáles deben ser elegidos para criar. Existen diferentes técnicas para llevar a cabo este proceso. Estas técnicas utilizan una combinación de la probabilidad y aptitud de cada individuo (T. Zhang et al., 2020). Las principales técnicas de selección son:

- Selección elitista: se busca a los individuos con mejores características de cada generación.
- Selección proporcional a la aptitud: los mejores individuos tienen más posibilidades de ser seleccionados. Sin embargo, también existe la probabilidad de que se elijan individuos menos adecuados.
- Selección de la rueda de la ruleta: se selecciona a los individuos mediante probabilidades por lo que no se permite a individuos de baja aptitud poder reproducirse.
- Selección por torneo: los individuos de la población compiten directamente entre sí, y solo se elige uno de cada competencia.

Reproducción

En la reproducción, el algoritmo busca seguir el modelo natural, donde la diversidad genética está presente a través de la reproducción sexual. En AG, un par de individuos cruzan sus genes y generan dos hijos que combinan las características de ambos padres. El cruce o reproducción se puede entender como la unión entre dos individuos de distinto sexo (Chin & Lin, 2018).

Para realizar la reproducción se pueden utilizar técnicas como las que se describen a continuación:

- Cruce de un punto: se define aleatoriamente un punto de intercambio en los cromosomas de los dos individuos. El primer individuo aporta todos sus genes previos hasta ese punto, y el otro individuo aporta sus genes a partir de ese momento.
- Cruce en dos puntos: Esta técnica es similar a la anterior, pero ahora los individuos intercambian genes a intervalos delimitados por dos puntos.

Criterio de parada

Idealmente, el AG debería parar su funcionamiento cuando se alcanza la solución óptima, sin embargo esta solución se desconoce en la mayoría de los casos. Por tanto, se utilizan varios criterios de parada: que la función objetivo alcance un valor aceptable previamente definido, que el algoritmo cumpla un determinado número de iteraciones, o que alcance un tiempo de funcionamiento determinado.

3 HIPOTESIS

Un controlador digital con base en lógica difusa orientado a trayectorias en procesos robotizados otorga los valores deseados de velocidad y posición aun cuando la trayectoria y la carga original cambien.

4 OBJETIVOS

4.1 OBJETIVO GENERAL

Diseñar un sistema de control usando lógica difusa en una impresora 3D como método para regular la velocidad y obtener valores deseados de posición.

4.2 OBJETIVOS PARTICULARES

- Construir la estructura de experimentación utilizando una impresora 3D.
- Desarrollar los dispositivos necesarios para leer, adaptar y enviar las señales requeridas utilizando microcontroladores.
- Diseñar e implementar un controlador difuso en la impresora 3D.
- Optimizar el controlador difuso utilizando un algoritmo genético.

5 METODOLOGÍA

Como se ha mencionado anteriormente el objetivo de este trabajo es controlar la posición de una impresora 3D. La metodología está enfocada en desarrollar a detalle cada bloque que se muestra en a Figura 9 la cual muestra el diagrama del lazo de control diseñado para alcanzar el objetivo anterior. El bloque del controlador está constituido por una técnica híbrida entre lógica difusa y algoritmos genéticos; el DAQ es el encargado de convertir las señales a valores adecuados para que la computadora pueda comprender las mediciones; la etapa de potencia eleva los niveles de voltaje y corrientes que entrega el DAQ para que el sistema sea capaz de mover el motor seleccionado, este motor a su vez, es el actuador que modifica la posición de la base en la impresora. En la **Tabla 2** se muestran las variables utilizadas en el lazo de control.

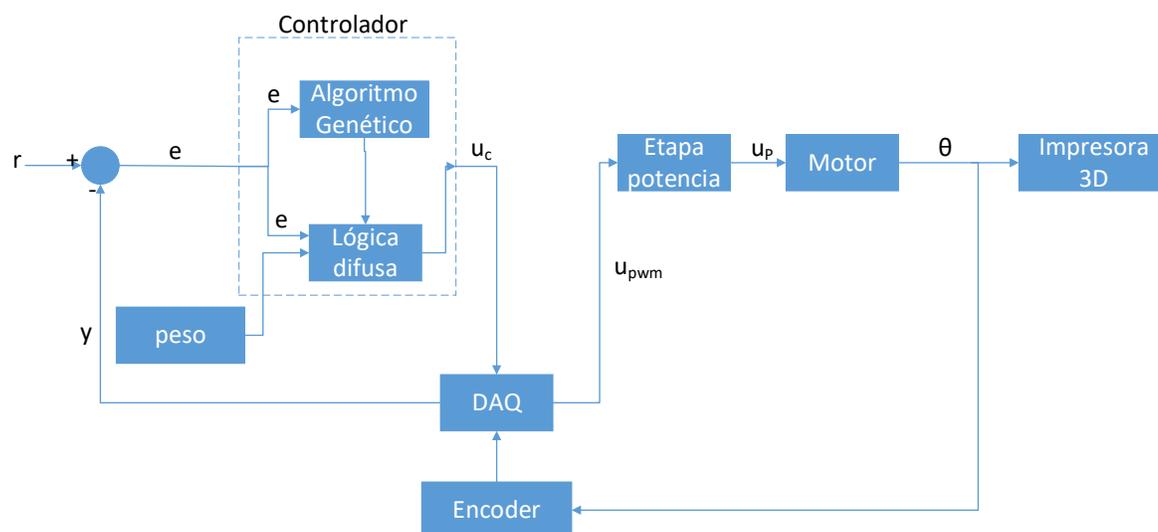


Figura 9. Diagrama de bloques del lazo de control propuesto para la tesis.

Variable	Significado
r	Posición deseada impresora
y	Posición real impresora
e	Error
\dot{e}	Derivada del error
u_c	Señal de control
u_{pwm}	Señal de PWM
u_p	Señal de potencia
θ	Posición angular del motor

Tabla 2. Variables utilizadas en el proyecto.

5.1 TARJETA DE ADQUISICIÓN (DAQ)

El elemento encargado de leer la posición que entrega el encoder es la tarjeta de adquisición de datos, la cual envía la posición actual como retroalimentación al lazo del controlador. En este bucle, el controlador realiza las operaciones necesarias y genera la señal u_c que se envía de vuelta a la tarjeta de adquisición de datos. Una vez de vuelta, la señal se convierte en un valor PWM (u_{pwm}) el cual se explica más adelante. Luego, la etapa de potencia, donde se encuentra el puente H, modifica los valores de voltaje (u_p) para el movimiento del eje del motor.

La tarjeta de adquisición tiene varias funciones fundamentales dentro de la plataforma. Es el elemento encargado de vincular todos los componentes que son necesarios para el funcionamiento del proyecto. Debido a su alta velocidad de procesamiento, se seleccionó un DSC como dispositivo principal en el DAQ. El tiempo de muestreo T_s es de 5 milisegundos. Las tareas de la tarjeta son las siguientes:

- Lee la posición del motor a través del encoder acoplado a su eje.
- Convierte los pulsos entregados por el encoder a un desplazamiento lineal y .
- Envía la posición al controlador.
- Recibe la señal de control.
- Convierte la señal de control a PWM y la envía al puente H.

La señal PWM significa modulación por ancho de pulso y consiste en generar una señal proporcional al valor deseado, en este caso específico un valor de voltaje. Es decir, la señal u_c tiene un rango de 0 a 3.3 volts. Se busca generar una señal proporcional a ese valor, el cual consiste en ondas de pulsos cuadrados. En la **Figura 10** se muestra el funcionamiento para tres valores distintos de ancho de pulso: 25%, 50% y 100 % y su equivalencia en volts de la amplitud total de la señal cuadrada.

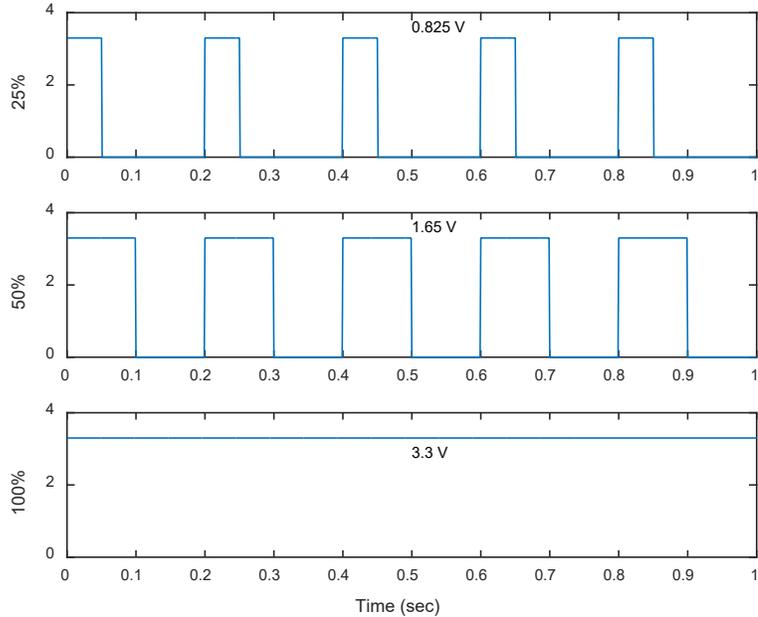


Figura 10. Representación de de PWM para una señal de 3.3 V de amplitud y 5 HZ de frecuencia.

Los componentes utilizados en la tarjeta de adquisición tienen que contar con módulos que puedan generar señales PWM y tener comunicación serial nativa, además se necesita que el microcontrolador tenga cuadratura para lectura de codificadores rotatorios.

Se seleccionó un microcontrolador de la familia Microchip, específicamente el modelo dsPIC33FJ12MC, este modelo pertenece a la subfamilia dsPIC (*Digital Signal Peripheral Interface Controller*) controlador de interfaz periférico de señal digital, estos presentan una velocidad de procesamiento superior a la de los PIC's convencionales, y gracias a lo anterior la cuenta de los pulsos del encoder se realiza sin problemas.

Las especificaciones de funcionamiento del dsPIC se enlistan a continuación:

- Voltaje de alimentación de 3.0 V a 3.6 V.
- Rango de temperatura de -40 °C a +85 °C.

- Memoria de programa lineal que trata hasta 4M palabras de instrucción.
- Direccionamiento de memoria de datos lineal de hasta 64 Kbytes.
- PWM de hasta 16-bits.
- 118 vectores de interrupción.
- Hasta 3 interrupciones externas.

El dsPIC cuenta además con funciones que permiten el control de motores eléctricos en ambas direcciones lo cual es un requisito fundamental para el proyecto.

- 6 canales para control de motores por PWM.
- 3 generadores de ciclo de trabajo.
- Tiempo muerto programable.
- Disparador para conversiones de ADC.

Para leer la señal del encoder el dsPIC cuenta con las siguientes características:

- Módulo de interface de cuadratura de encoder.
- Fase A, Fase B.
- Contador de posición de 16 bits.
- Filtros de ruido programables en las entradas.

Respecto al apartado de comunicación serial este dsPIC presenta las siguientes cualidades.

- Interrupción en la detección de bits de dirección.
- Interrupción con error UART.
- Despertar en el bit de inicio desde el modo de Suspensión.

- Buffers TX y RX FIFO de 4 caracteres.
- Codificación y decodificación en hardware.
- Modo Baudio de alta velocidad.
- Control de flujo de hardware con CTS y RTS

La conversión de la señal u_c a u_{pwm} está dada por la Ecuación (10)

$$u_{pwm} = \frac{u_c}{3.3} \quad (255) \quad (10)$$

La **Figura 11** muestra la distribución de los pines para el microcontrolador utilizado en la tesis.

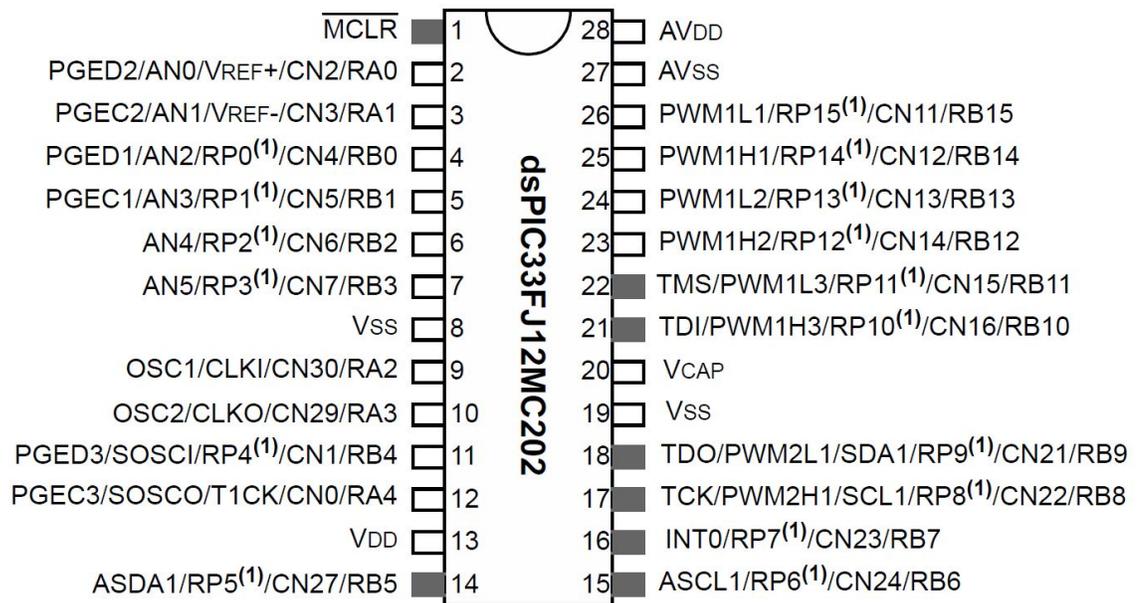


Figura 11. Diagrama de pines del dsPIC utilizado en el proyecto.

Para agregar protección a los dispositivos de control se requiere añadir aislamiento mediante elementos ópticos, específicamente optoacopladores que tienen la función de realizar una conexión mediante un haz de luz sin necesidad de

conexiones directas, por lo tanto aísla ópticamente dos o más dispositivos que operan con distintas fuentes de alimentación pero permitiendo a la vez su conexión mediante un led y un fototransistor.

La **Figura 12** muestra el diagrama de distribución de pines del optoacoplador utilizado en el proyecto. El pin 1 es el ánodo, el pin 2 es corresponde al cátodo, el pin 3 no lleva conexión, el pin 4 es el emisor, el pin 5 es el colector y finalmente el pin 6 es la base del dispositivo.

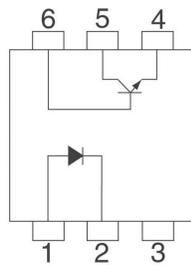


Figura 12. Conexión del optoacoplador 4n28. (Imagen obtenida de hoja de datos).

5.1.1 DISEÑO ESQUEMÁTICO DEL DAQ

La **Figura 13** presenta el circuito necesario para el funcionamiento del dsPIC. Como puede verse, se requieren dos capacitores, C1 y C2 con un valor de 100 pF; estos condensadores están conectados a los pines 9 y 10 del dsPIC y en paralelo con el cristal X1 con un valor de 20 MHz. Para reiniciar el dsPIC se emplea un botón pulsador, que requiere de una resistencia R_MRCL con un valor de 10 kohms. Para protección contra corrientes y voltajes, se agregan protecciones a la SALIDA1 y SALIDA2. El codificador está conectado a los terminales QE_A y QE_B que corresponden a los pines 24 y 26.

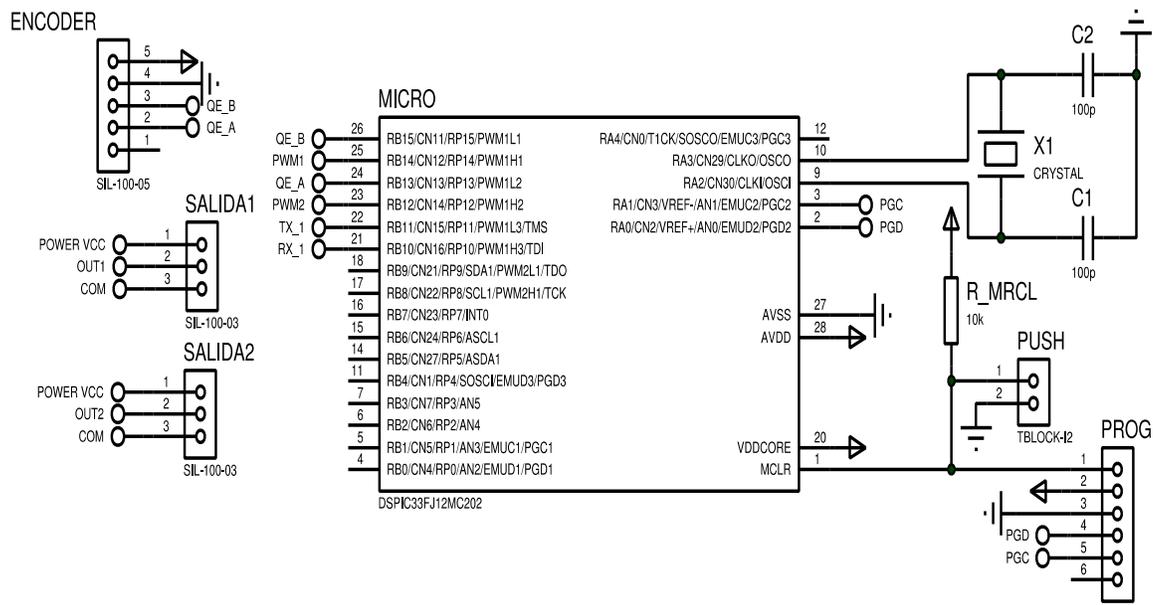


Figura 13. Diseño esquemático de la conexión del dsPIC para el DAQ.

La conexión de los optoacopladores se muestra en la **Figura 14**. Se necesitan de dos resistencias de 330 ohms, una para cada señal de PWM proveniente del dsPIC, que corresponden a los pines 25 y 26. Dos resistencias de 1 kohm para conectarlas a tierra, indicadas al pin 4 de los optoacopladores. El pin 4 es también la salida del DAQ hacia la etapa de potencia. Como se mencionó anteriormente estos dispositivos sirven para proteger la tarjeta de adquisición evitar sobrecorrientes en ésta.

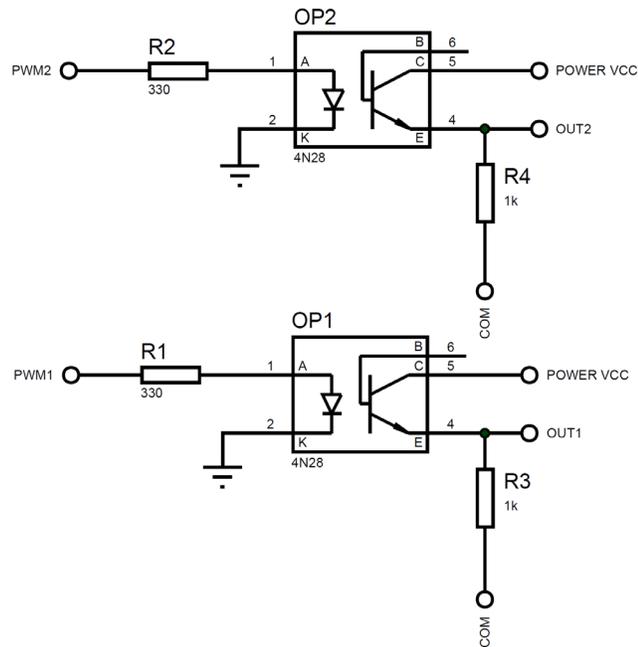


Figura 14. Diagrama de conexión de los optoacopladores.

5.1.2 PROGRAMACIÓN DEL DSPIC

El Entorno de Desarrollo Integrado (IDE) seleccionado para programar el PIC es PIC C que ha sido creado por PIC CMU. La razón de usar este IDE es porque tiene diferentes librerías y drivers que facilitan la programación. Además, PIC C usa el lenguaje C en lugar del lenguaje ensamblador. La **Figura 15** presenta los pasos a seguir para poner en marcha el dsPIC. Se deben especificar las cabeceras para el tipo de PIC, las librerías requeridas y la dirección del puerto B que se adopta como interrupción para la lectura del encoder.

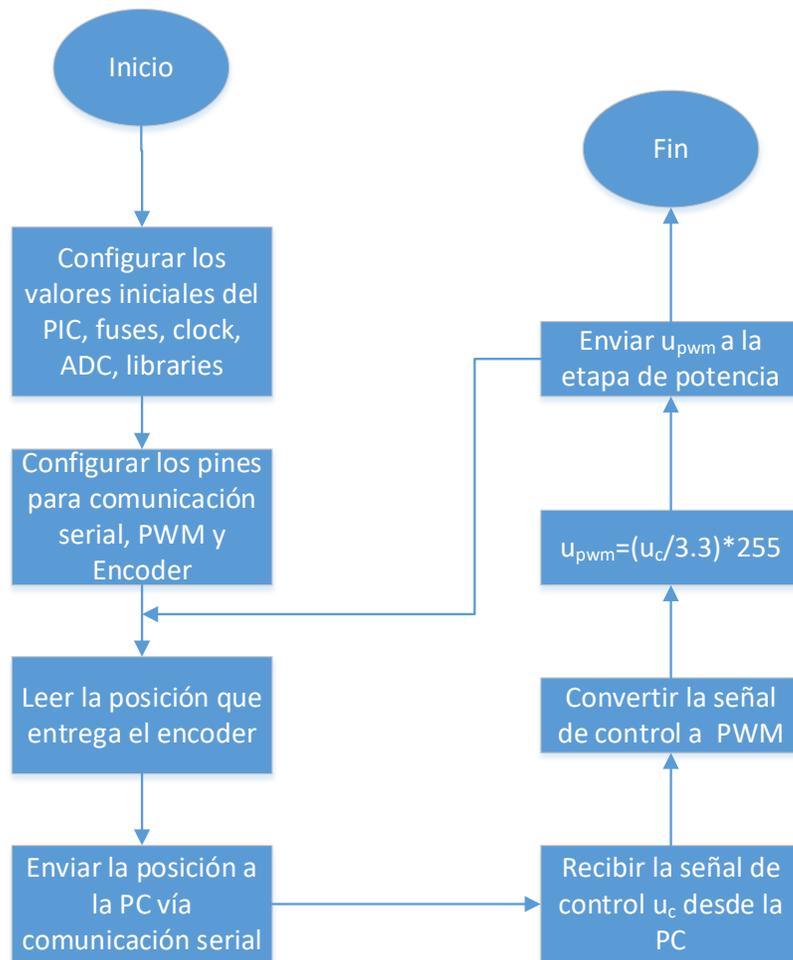


Figura 15. Diagrama de flujo para las acciones del dsPIC.

5.1.3 DISEÑO DE LA TARJETA IMPRESA

Para la tarjeta impresa se utilizó un diseño a doble cara para reducir el tamaño requerido por los componentes. En la **Figura 16** se puede observar el diseño que se imprimió físicamente para la tarjeta de adquisición en dónde los elementos en color turquesa representan a los componentes utilizados para el DAQ y el color rosa los pines para cada uno de estos componentes. Las pistas de conexión para la tarjeta se muestran en líneas de color rojo para la cara superior y líneas en color

azul para la cara inferior. Se utilizó una placa de cobre para grabar cada elemento de este diseño.

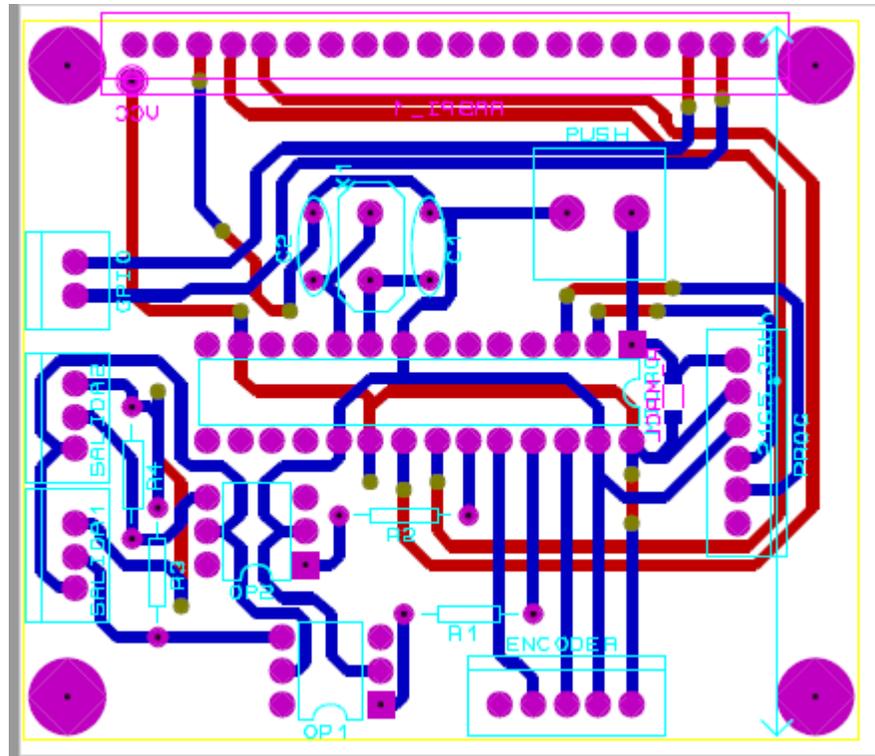


Figura 16. Diseño del PCB para la tarjeta de adquisición.

5.1.4 MODELADO DE LA TARJETA DE ADQUISICIÓN

El modelado del DAQ se muestra en la **Figura 17** para el PCB sin componentes y en la **Figura 18** para la vista en 3 dimensiones con los componentes incluidos. En estas figuras se puede observar de mejor manera la distribución total de cada elemento utilizado para la adquisición de datos en el proyecto.

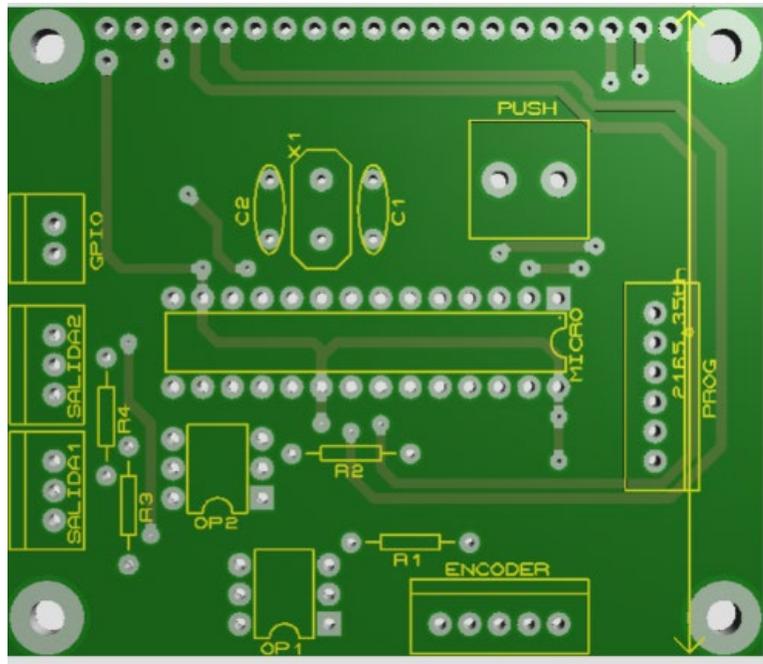


Figura 17. Vista de la tarjeta sin componentes.

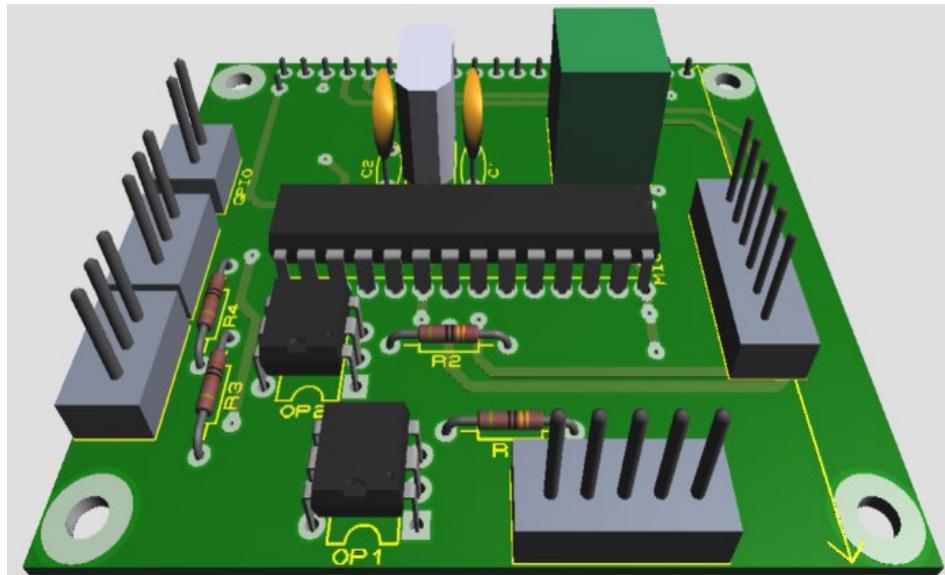


Figura 18. Vista de la tarjeta de adquisición en 3 dimensiones con los componentes utilizados.

5.2 MOTOR

Los motores eléctricos son elementos ampliamente utilizados como actuadores en la industria. Su funcionamiento depende de la acción que genera un campo magnético en la estructura del motor. El motor de corriente continua es un dispositivo que transforma la energía eléctrica en mecánica, provocando un movimiento rotatorio.

El motor que se utilizó en el proyecto fue el modelo GW4058-555 y sus características de funcionamiento se muestran en la **Tabla 3**.

Parámetro	Valor
Voltaje nominal	12 V
Velocidad al vacío	222 rpm
Velocidad con carga	105 rpm
Corriente al vacío	200 mA
Torque	20 kgcm
Corriente nominal	< 3A
Corriente de arranque	< 5A

Tabla 3. Características de funcionamiento del motor modelo GW4058-555.

La **Figura 19** muestra las características y medidas físicas del motor GW4058-555. Estas medidas son importantes ya que el motor se debe acoplar a la base de la impresora sin interferir con los otros componentes requeridos para el funcionamiento integral del proyecto.

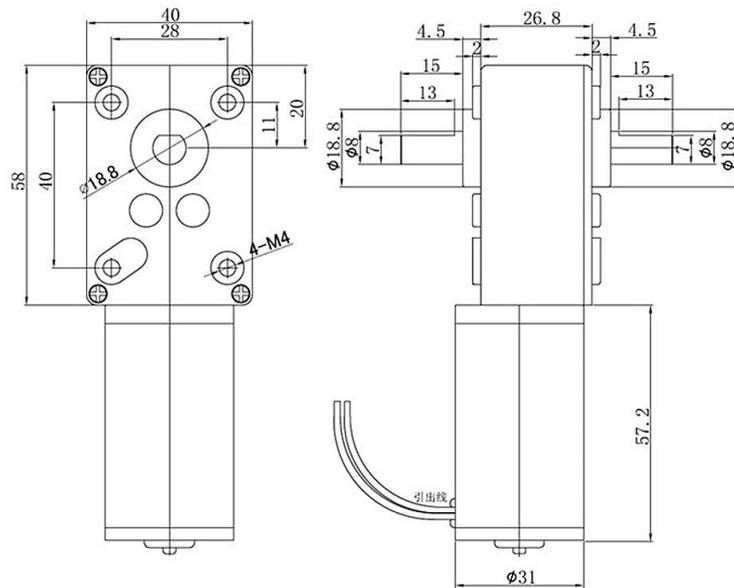


Figura 19. Características físicas del motor utilizado.

5.3 ENCODER

El sensor encargado de medir la posición angular del motor (θ) es un codificador rotatorio, conocido simplemente como encoder. El funcionamiento del encoder utiliza métodos como la tecnología óptica, magnética o de resistencia para medir el movimiento en los motores. El encoder utilizado en este proyecto es un codificador rotatorio incremental que está acoplado directamente al motor y es de tecnología óptica. El funcionamiento específico consiste en generar luz mediante un LED que pasa a través de un disco el cual alterna entre ranuras y áreas opacas formando un código. A medida que el motor gira la luz emitida por el LED se interrumpe cada vez que un área opaca del disco se interpone entre el LED y un elemento receptor,

frecuentemente un fototransistor. Lo anterior genera una señal compuesta por pulsos, encendido cuando el fototransistor capta la luz del LED y apagado cuando no detecta la luz. Esta señal es enviada a un elemento de control encargado de contar el número de pulsos y determinar la posición. En la Figura 20 se muestran los pines de conexión del motor y encoder.

Las características del encoder utilizado se mencionan a continuación:

- Voltaje de operación de 5 V de corriente directa.
- Resolución de 256 PPR.
- Protección contra cortocircuitos.
- Configuración de salida de colector abierto.
- Máxima frecuencia de respuesta 30 kHz.
- Consumo máximo de corriente 30 mA.
- Dos señales de salida A y B

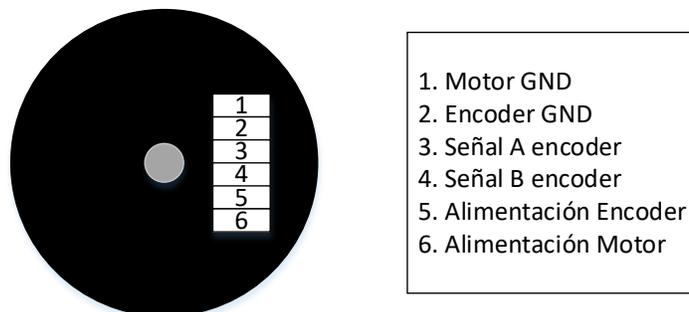


Figura 20. Vista inferior del encoder e identificación de pines.

Debido a que la resolución del encoder es de 256 PPR la conversión el movimiento angular del motor (θ) está determinado por la Ecuación (11).

$$1 \text{ revolución} = 360^\circ = 256 \text{ pulsos} \quad (11)$$

5.4 ETAPA DE POTENCIA

Como muestra la **Tabla 3** de acuerdo a los requerimientos del motor seleccionado para el proyecto, una corriente de arranque 5 A es necesaria para accionar el motor. El diseño de la etapa de potencia debe soportar esta corriente además de trabajar con 12 V, que es la tensión de alimentación del motor. Esta etapa también debe ser capaz de trabajar en sincronía con las otras etapas del sistema por lo que debe manejar velocidades de conmutación suficientes para interactuar con el microcontrolador. Se seleccionó un puente H modelo BTS7960B para satisfacer las necesidades de energía. Se recomienda emplear un circuito comercial ya ensamblado como el IBT_2 para la plataforma Arduino, que internamente contiene el puente H modelo BTS7960B y componentes de control que le permiten tener las siguientes características:

- Voltaje de alimentación para elementos de potencia de 0 hasta 27 V.
- Voltaje de alimentación para los circuitos de control de 0 hasta 5.3 V.
- Soporta corrientes de hasta 43 Amperes.
- Maneja señales PWM con frecuencias de hasta 25 kHz.
- Circuito interno de control con niveles de entrada lógicos.

En la **Figura 21** se observa el diagrama del circuito utilizado.

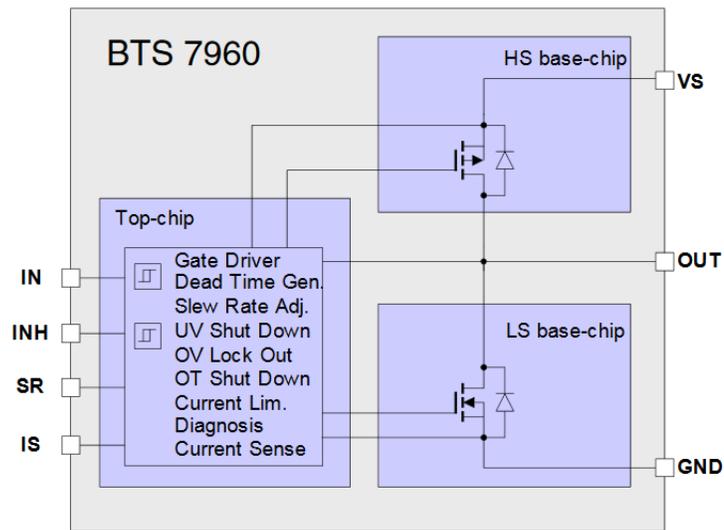


Figura 21. Diagrama de bloques del medio puente H. (Imagen obtenida de hoja de datos).

La limitación que tiene el circuito anterior es que únicamente es medio puente H, por lo que se requieren de dos de ellos para manejar ambos sentidos en un motor, además de un controlador que administre las funciones de éste. Debido a la limitación anterior se utilizó un circuito comercial ya ensamblado y encapsulado, específicamente el puente H modelo IBT_2. Este puente es originalmente diseñado para la plataforma Arduino pero se adaptó al proyecto de tesis ya que cuenta con componentes agregados que le otorgan las siguientes características.

- Voltaje de alimentación de 5 a 27 V.
- Control de velocidad de motores por medio de PWM con frecuencias de hasta 25 kHz.
- Control en ambas direcciones del motor.
- Máxima corriente soportada de hasta 30 amperes.
- Peso de 66 gramos.

En la **Figura 22** y en la **Tabla 4** se muestra los pines de conexión de este puente H a la tarjeta de adquisición.

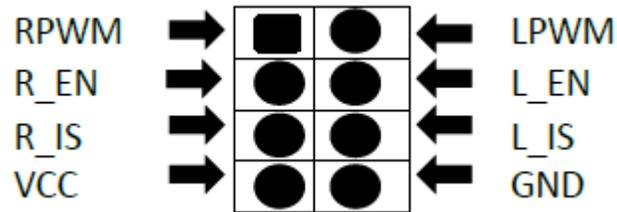


Figura 22. Pines de control en el puente H.

Pin	Nombre	Función
1	RPWM	PWM de entrada para girar a la derecha
2	LPWM	PWM de entrada para girar a la izquierda
3	R_EN	Activar giro a la derecha
4	L_EN	Activar giro a la izquierda
5	R_IS	Alarma de corriente
6	L_IS	Alarma de corriente
7	VCC	Voltaje alimentación
8	GND	Tierra

Tabla 4. Descripción de los pines del IBT_2.

La conversión de la señal de potencia u_{pwm} que entrega el DAQ a u_p que es la que entrega la etapa de potencia y es la requerida por el motor se da de acuerdo a la Ecuación (12).

$$u_p = \frac{u_{pwm}}{3.3} \quad (12)$$

5.5 IMPRESORA 3D

En esta investigación se utiliza la base de una impresora 3D modelo Anet A8 para probar el desempeño del controlador. El motor tiene un acoplamiento dentado añadido a su eje. Este acoplamiento tiene la función de sujetar una banda elástica que se conecta directamente a la base de la impresora tal como se muestra en la **Figura 23**.

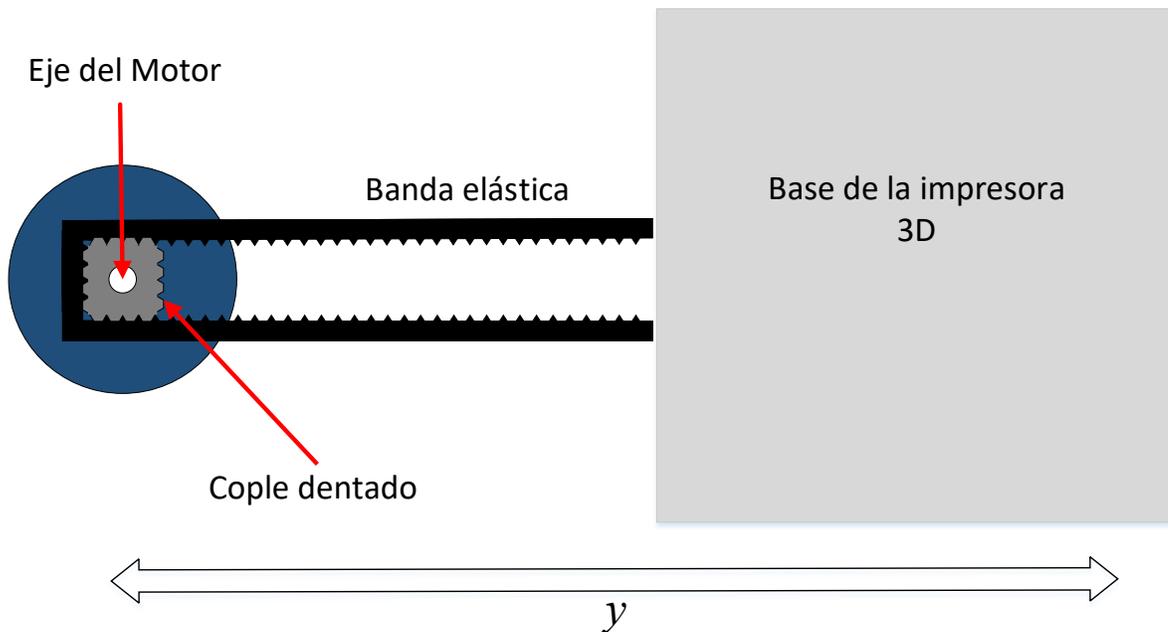


Figura 23. Conexión entre el motor y la impresora 3D.

Cada rotación del eje del motor corresponde a un movimiento lineal y en la plataforma. La relación que describe este movimiento se muestra en la Ecuación (13).

$$1 \text{ revolution} = 360^\circ = 1.5 \text{ cm} = 256 \text{ PPR} \quad (13)$$

La variable y tiene un rango completo de 30 cm, el cual está deliberadamente restringido para dejar un margen de seguridad de 3 cm a cada lado.

5.6 CONTROLADOR DIFUSO

Esta sección tiene como objetivo desarrollar las etapas para la lógica difusa, por lo que se define subsecciones para cada una de ellas: Fusificación, inferencia difusa y defusificación, tal como lo muestra la **Figura 7**.

5.6.1 FUSIFICACIÓN

Se utilizaron funciones de pertenencia de tipo triangular, trapezoidal, tipo L y tipo R para transformar los valores numéricos de las variables en expresiones lingüísticas. Las ecuaciones utilizadas para obtener el grado de pertenencia se muestran a continuación:

La función triangular está definida en la Ecuación (14) y en la **Figura 24**.

$$\mu_A(x) = \begin{cases} 0, & \text{si } (x \leq a) \\ \frac{x - a}{m - a}, & \text{si } (a < x \leq m) \\ \frac{b - x}{b - m}, & \text{si } (m < x < b) \\ 0, & \text{si } (x \geq b) \end{cases} \quad (14)$$

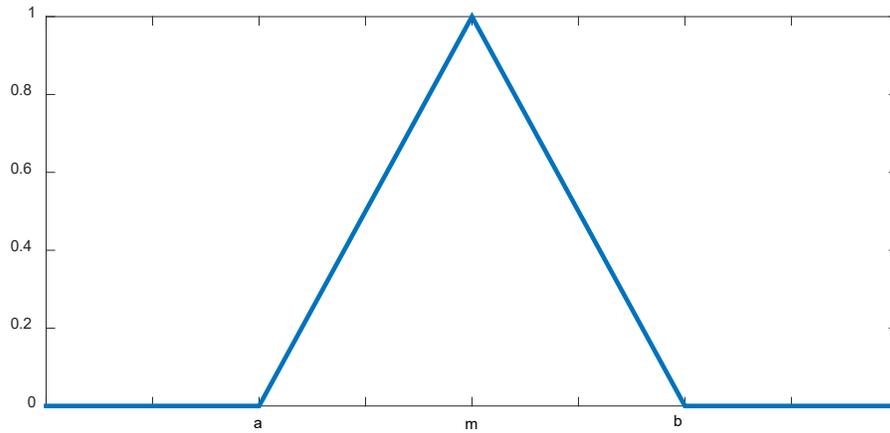


Figura 24. Representación de la función triangular.

La Ecuación (15) y la **Figura 25** describen a la función trapezoidal.

$$\mu A(x) = \begin{cases} 0, & \text{si } (x < a) \text{ o } (x > d) \\ \frac{x - a}{b - a}, & \text{si } (a \leq x \leq b) \\ 1, & \text{si } (b \leq x < c) \\ \frac{d - x}{d - c}, & \text{si } (c \leq x \leq d) \end{cases} \quad (15)$$

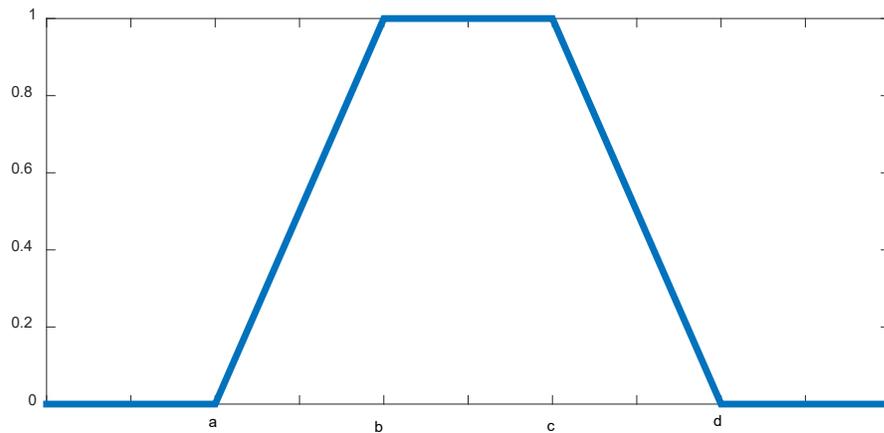


Figura 25. Representación de la función trapezoidal.

La función gaussiana se muestra en la Ecuación (16) y en la **Figura 26**.

$$\mu A(x) = e^{-\frac{(x-m)^2}{2k^2}} \quad (16)$$

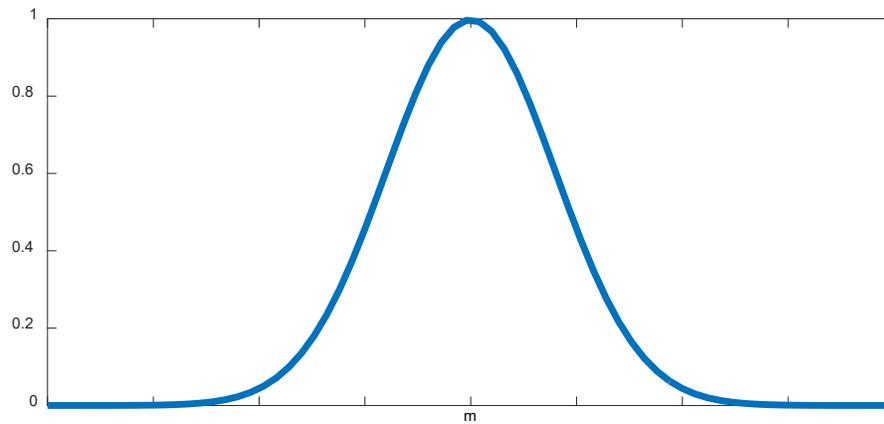


Figura 26. Representación de la función gaussiana.

La función tipo R está descrita en la Ecuación (17) y en la **Figura 27**.

$$\mu A(x) = \begin{cases} 0, & \text{si } (x > d) \\ \frac{d-x}{d-c}, & \text{si } (c \leq x \leq d) \\ 1, & \text{si } (x < c) \end{cases} \quad (17)$$

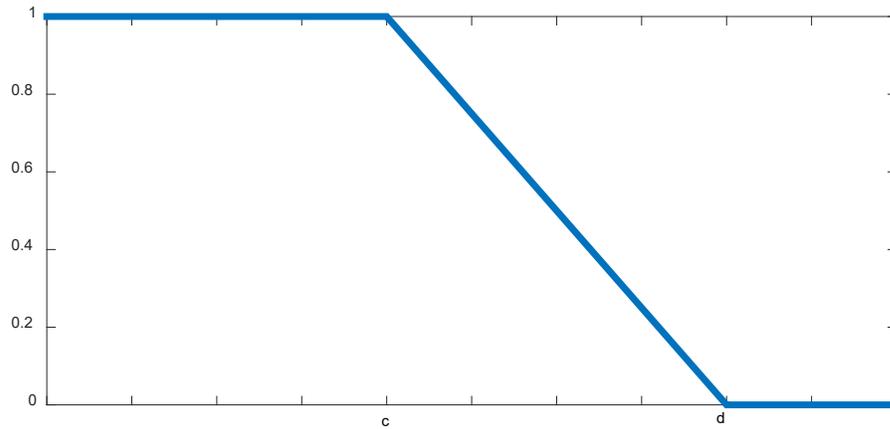


Figura 27. Representación de la función tipo L.

La Ecuación (18) y la **Figura 28** definen a la función tipo L.

$$\mu A(x) = \begin{cases} 0, & \text{si } (x < a) \\ \frac{x - a}{b - a}, & \text{si } (a \leq x \leq b) \\ 1, & \text{si } (x > b) \end{cases} \quad (18)$$

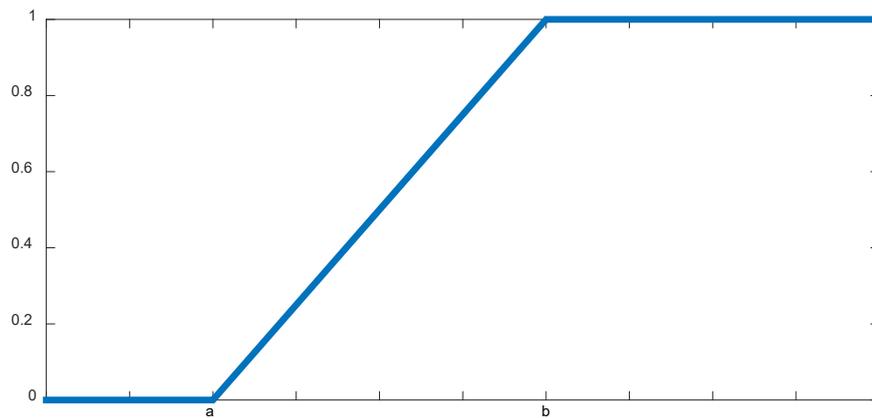


Figura 28. Representación función tipo L.

Como se muestra en la Figura 9 existe un valor de posición deseado r y se retroalimenta el valor real medido por el encoder y . El error e se obtiene de la diferencia entre los dos valores anteriores tal como se muestra en la ecuación. Ecuación (19).

$$e = y - r \quad (19)$$

De la misma manera se puede observar en la Figura 9 que e es la primera variable de entrada para el controlador difuso y sus funciones de membresía están definidas en la **Figura 29**, que consta de tres funciones triangulares, una función R y una función L, con un universo del discurso de -15 a 15 cm. Es importante hacer énfasis en que esta distribución es solo inicial y que estás variable es el enlace con la optimización mediante algoritmos genéticos que se explica más adelante en este documento.

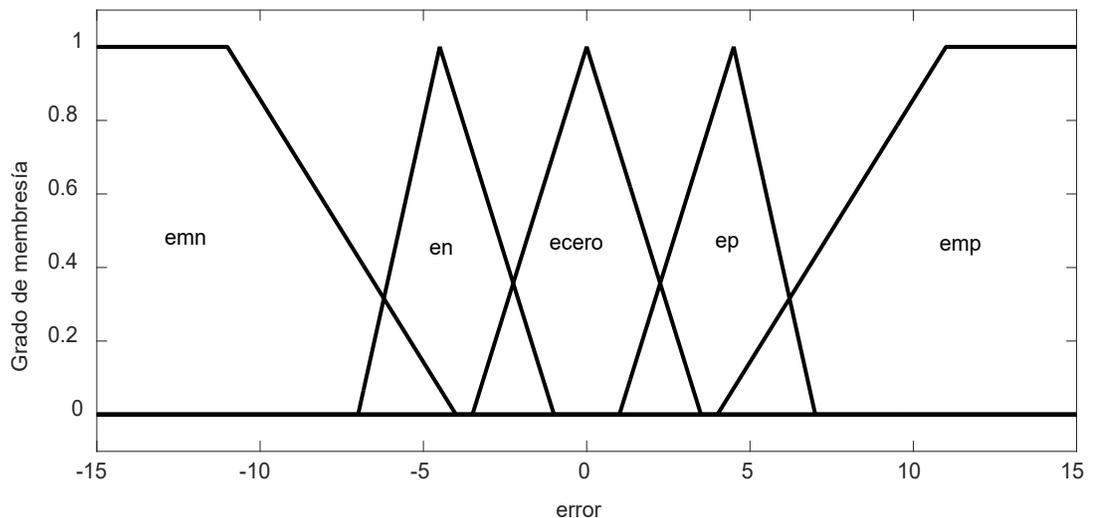


Figura 29. Funciones de membresía para la variable error.

Para este proyecto se seleccionó otra variable de entrada que es el peso que se le agrega al eje del motor. Sus funciones de pertenencia consisten en una función una función tipo L, una función tipo R y tres funciones triangulares. El rango de esta variable es de 0 a 20 libras. Esta distribución se muestra en la **Figura 30**.

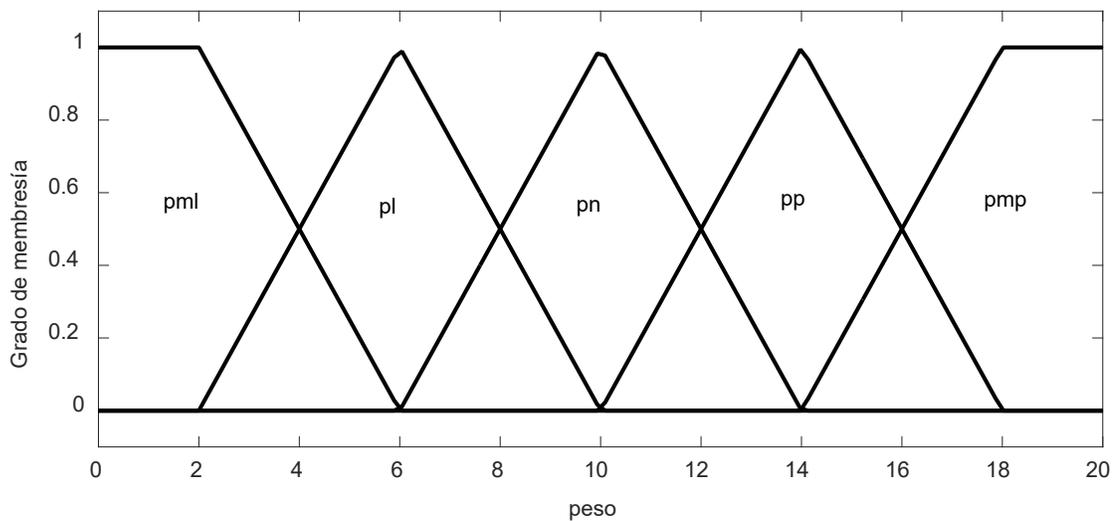


Figura 30. Funciones de membresía para la variable de peso.

Hay una única variable de salida y se utilizan cinco funciones de pertenencia de tipo triangular para transformar su valor numérico en expresión lingüística. Las funciones triangulares tienen un universo de discurso con un rango de -10 a 10 V. La figura **Figura 31** muestra las distribuciones de estas funciones de pertenencia.

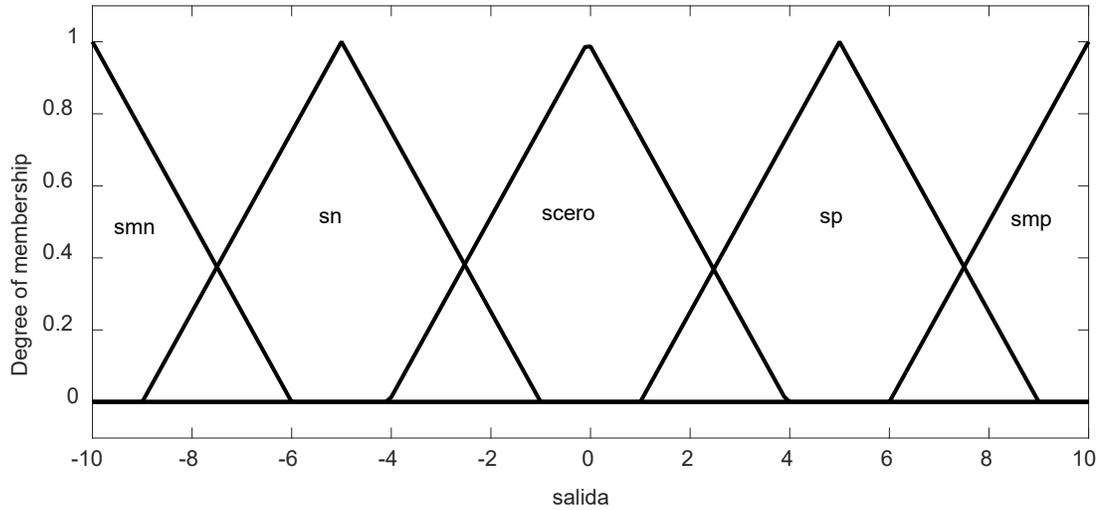


Figura 31. Funciones de membresía para la variable de salida.

Las etiquetas y su correspondiente expresión lingüística utilizada para las funciones de pertenencia de cada una de las variables utilizadas se muestran en la **Tabla 5**.

Etiqueta	Expresión lingüística
emn	Error muy negativo
en	Error negativo
ecero	Error cero
ep	Error positivo
emp	Error muy positivo
pml	Peso muy ligero
pl	Peso ligero
pn	Peso normal
pp	Peso pesado
pmp	Peso muy pesado
smn	Salida muy negativa
sn	Salida negativa

scero	Salida cero
sp	Salida positiva
smp	Salida muy positiva

Tabla 5. Etiquetas y expresiones lingüísticas utilizadas en la tesis.

5.6.2 REGLAS DIFUSAS

Las reglas difusas se definieron con base en conocimientos previos sobre el comportamiento del sistema, tomando como referencia una investigación previa realizada anteriormente (Torres-Salinas et al., 2020). Este conocimiento se obtuvo a través de múltiples experimentos en la plataforma de prueba, en los que se implementaron diferentes tipos de controladores ante distintos comportamientos. Como resultado de lo anterior, se establecieron 25 reglas, las cuales se muestran en la **Tabla 6**.

error/peso	emn	en	ecero	ep	emp
pmp	sp	sn	sn	smn	smn
pp	sp	scero	sn	smn	smn
pn	smp	sp	scero	sn	smn
pl	smp	smp	sp	scero	sn
pml	smp	smp	sp	sp	sn

Tabla 6. Reglas difusas utilizadas en el proyecto de tesis.

El conector lógico AND correspondiente a la operación de intersección descrita en Ecuación (8) se usa para vincular las dos variables de entrada en las reglas difusas.

Para entender la manera de interpretar las reglas anteriores se brindan dos ejemplos a continuación:

Si $error = en$ AND $peso = pn$, entonces $salida = sp$

Si $error = emp$ AND $peso = pml$, entonces $salida = sn$

5.6.3 DEFUSIFICACIÓN

Después de aplicar las reglas difusas se sigue teniendo una salida que está en forma de expresión lingüística, por lo que esta debe ser transformada nuevamente a un valor numérico. El método CoA se utiliza en el proceso de defusificación. Con esta técnica, el área de las funciones de membresía, en la que la salida difusa obtuvo pertenencia, se calcula según la Ecuación (20). Esta es la etapa final del controlador y su salida es u_c donde x es el valor de la variable lingüística y su dominio está representado por x_{min} y x_{max} .

$$CoA = \frac{\int_{x_{min}}^{x_{max}} f(x)x dx}{\int_{x_{min}}^{x_{max}} f(x) dx} = u_c \quad (20)$$

Los pasos simplificados para el controlador difuso se muestran en el siguiente pseudocódigo en donde el tiempo de muestro $t_s = 5$ milisegundos y T es variable dependiendo de los escenarios de experimentación en 2 segundos cuando es un solo valor deseado o 15 segundos cuando son múltiples valores deseados.

Inicialización;

Tiempo total = T ;

Establecer la posición deseada r ;

Establecer los valores iniciales para las funciones de membresía;

while $t < T$ **do**

 Obtener el valor de posición real y proveniente del DAQ;
 Calcular e de acuerdo a la Ecuación (19);
 Computar la variable de peso;
 Calcular el grado de membresía para el error mostrado en la Figura 29 utilizando las Ecuaciones (14) a (18);
 Calcular el grado de membresía para la variable de peso de la Figura 30 utilizando las Ecuaciones (14) a (18);
 Obtener la salida difusa usando las reglas de la Tabla 6 y la Ecuación (8);
 Calcular u_c usando la Ecuación (20);
 Enviar u_c al DAQ;
 $t = t + t_s$;

end

5.7 OPTIMIZACIÓN POR ALGORITMOS GENÉTICOS

Como se mencionó anteriormente, el vínculo entre el controlador difuso y la optimización con algoritmos genéticos es la variable de entrada del error. Específicamente, el objetivo es optimizar los valores para las funciones triangulares de la **Figura 29** que corresponden a las expresiones lingüísticas en , $ecero$ y ep .

Tal como se puede ver en la **Figura 25** y en la Ecuación (15) la función triangular está compuesta tres valores que son necesarios para representada de manera correcta: la base izquierda del triángulo representada por la letra a , el punto más alto del triángulo m y finalmente la base del triángulo b . La **Figura 32** muestra que cada individuo de la población en los algoritmos genéticos se compone de la combinación de estos tres puntos para cada una de las tres funciones triangulares que se van a optimizar.

a_{en}	m_{en}	b_{en}	a_{ecero}	m_{ecero}	b_{ecero}	a_{ep}	m_{ep}	b_{ep}
----------	----------	----------	-------------	-------------	-------------	----------	----------	----------

Figura 32. Representación de un individuo en forma de vector para el algoritmo genético.

En la inicialización se genera de manera aleatoria una población de cuarenta individuos que tienen la estructura mostrada en la **Figura 32**. Para mantener la forma triangular de las funciones de membresía así como para lograr una distribución uniforme y finalmente evitar que se sobrepongan los valores medios de cada una de éstas funciones se establecen una serie de restricciones descritas en las Ecuaciones (21) a (23). Los valores numéricos de -15 y 15 corresponden a los límites de trabajo de la base de la impresora 3D definidos en la **Figura 23**.

$$-15 \leq a_{en} < m_{en} < b_{en} \leq 0 \quad (21)$$

$$-5 \leq a_{ecero} < m_{ecero} < b_{ecero} \leq 5 \quad (22)$$

$$0 \leq a_{ep} < m_{ep} < b_{ep} \leq 15 \quad (23)$$

Luego de generar la población inicial, se evalúa su aptitud de acuerdo a la Ecuación (24) donde $r(k)$ y $y(k)$ son el valor deseado y el valor medido en la iteración k , respectivamente, y m es el conteo de iteraciones.

$$J = \frac{1}{m} \sum_{k=1}^m [r(k) - y(k)]^2 \quad (24)$$

El siguiente paso es el proceso de selección. Para este paso se utiliza el método del torneo, que consiste en comparar aleatoriamente dos individuos de la población inicial. El individuo con la mejor aptitud es elegido para reproducirse.

Una vez seleccionados los individuos se utiliza el cruce PMX, como se muestra en la **Figura 33**. En este tipo de cruce, dos individuos, conocidos como padres, comparten genes según un índice H . El primer nuevo individuo adquiere todos los genes del primer padre hasta el índice H y, a partir de este índice, adquiere los genes del progenitor 2. El segundo de los individuos adquiere los genes del segundo padre hasta el índice H y los genes del primer padre después del índice. El proceso anterior se repite hasta generar una nueva población con el mismo número de individuos que la población inicial.

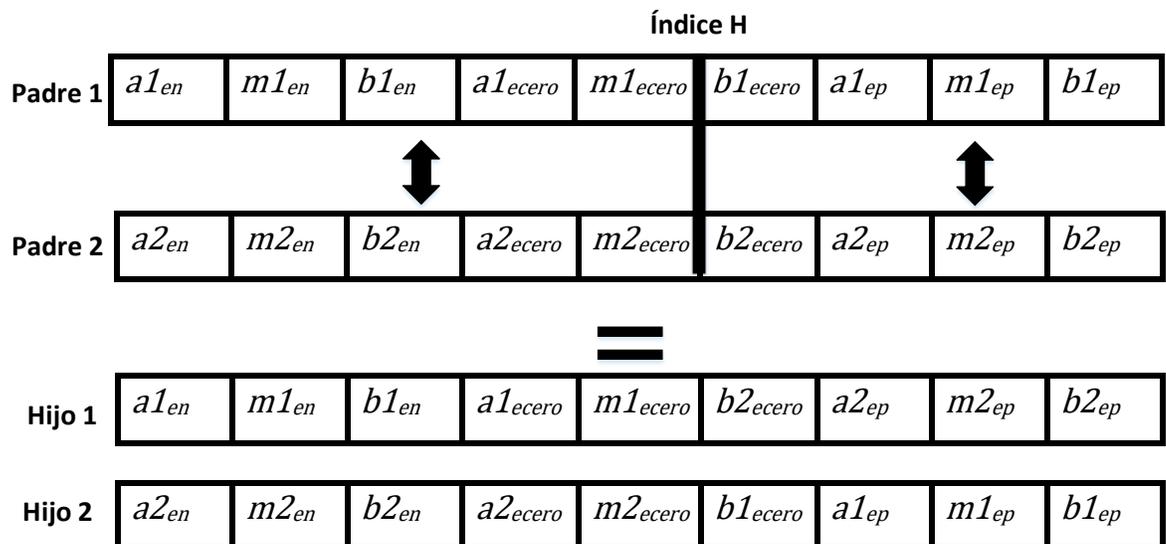


Figura 33. Demostración de la cruce PMX con dos individuos.

Una vez que se ha creado una nueva generación, se debe volver a evaluar su aptitud. El proceso continúa hasta que se cumple el criterio de parada, que consiste en completar 2 s de iteraciones para los experimentos con un solo valor deseado y 12 s para los experimentos con múltiples valores deseados.

A continuación se muestra el pseudocódigo simplificado para el algoritmo genético.

Inicialización;

Codificar los valores de las funciones de membresía a vectores como en la Figura 32;

Definir el tamaño de población $P = 40$;

Establecer los valores iniciales para las funciones de membresía;

Restringir los valores para los individuos de acuerdo a las Ecuaciones (21) a (23);

Evaluar la aptitud de los individuos de acuerdo a la Ecuación (24).

while $t < T$ do

 Seleccionar los padres de acuerdo al método del torneo;

 Realizar la cruce por PMX de acuerdo a la Figura 33;

 Seleccionar aleatoriamente individuos para mutación;

 Realizar una mutación uniforme en dos genes para los individuos;

 Evaluar la aptitud de los individuos en la nueva generación;

 Enviar los valores al controlador difuso;

end

5.8 INTEGRACIÓN E IMPLEMENTACIÓN

Esta sección tiene como objetivo describir la manera en la cual se integran todos los componentes descritos anteriormente, específicamente las conexiones entre ellos.

5.8.1 CONEXIONES Y CABLEADO

La conexión entre el encoder y el DAQ se realiza de acuerdo a las especificaciones de la distribución de pines dadas en Figura 20. Tal como se muestra en la **Figura 34** la alimentación (VCC) se conecta en el pin 5 tanto del DAQ como en el encoder, la tierra (GND) se conecta en el pin 4 del DAQ y en el pin 2 del encoder, la señal A se conecta en los pines marcados con el número 3 tanto en el DAQ como en el encoder y finalmente la señal B se conecta en el pin 2 del DAQ y en el pin 4 del encoder.

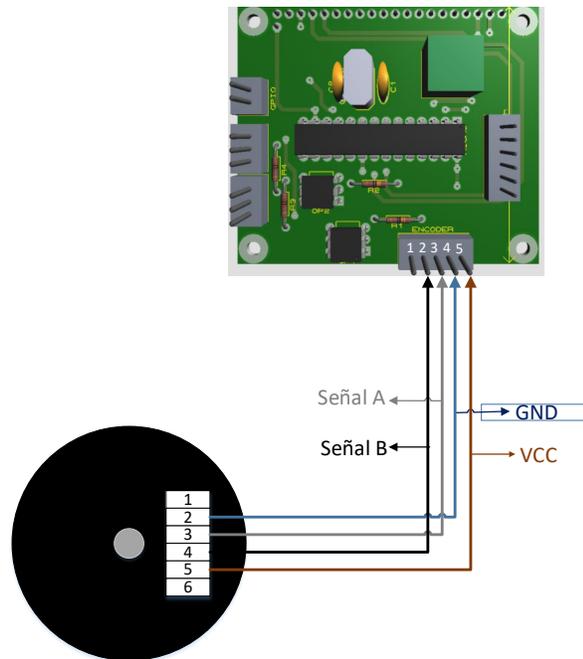


Figura 34. Conexión entre el encoder y el DAQ.

Para etapa de potencia, específicamente el puente H, se utiliza la descripción de pines mostrada en la Tabla 4. La manera en la que se realiza la conexión entre la etapa de potencia y el DAQ se muestra en la **Figura 35** en donde el bloque azul es una representación amplificada, para facilitar su visualización, de los pines 1 a 8 del dispositivo IBT_2. Se puede observar que los pines correspondientes al número 5 y

6 del circuito no se requiere que se conecten, lo anterior es debido a que su función es únicamente activar alarmas de corriente las cuales no se contemplaron en este proyecto de tesis.

En el pin 1 del puente H se conecta a una señal de PWM del DAQ correspondiente al pin 3 para activar el giro a la izquierda cuando el controlador lo indique.

El pin 2 del puente H está conectado de igual manera a otra señal de PWM del DAQ en este caso el pin 7 el cual tiene la función de activar el giro a la derecha.

El pin 3 y 4 del puente H tienen la función de habilitar la función tanto del giro a la izquierda como a la derecha, estos se requieren que estén siempre habilitados por lo que se conectan directamente a alimentación de voltaje. Esta alimentación la otorga el DAQ en el pin 3. El objetivo de que los giros estén siempre activos es porque gracias a eso solo se necesita de la señal de PWM en los pines 1 o 2 para funcionar y de esta manera se evita una instrucción más en el programa de control.

La alimentación del circuito del puente H se realiza en el pin 7 y el pin 6 del DAQ, mientras que el pin de tierra esta marcado con el pin 8 tanto en el DAQ como en el del puente H.

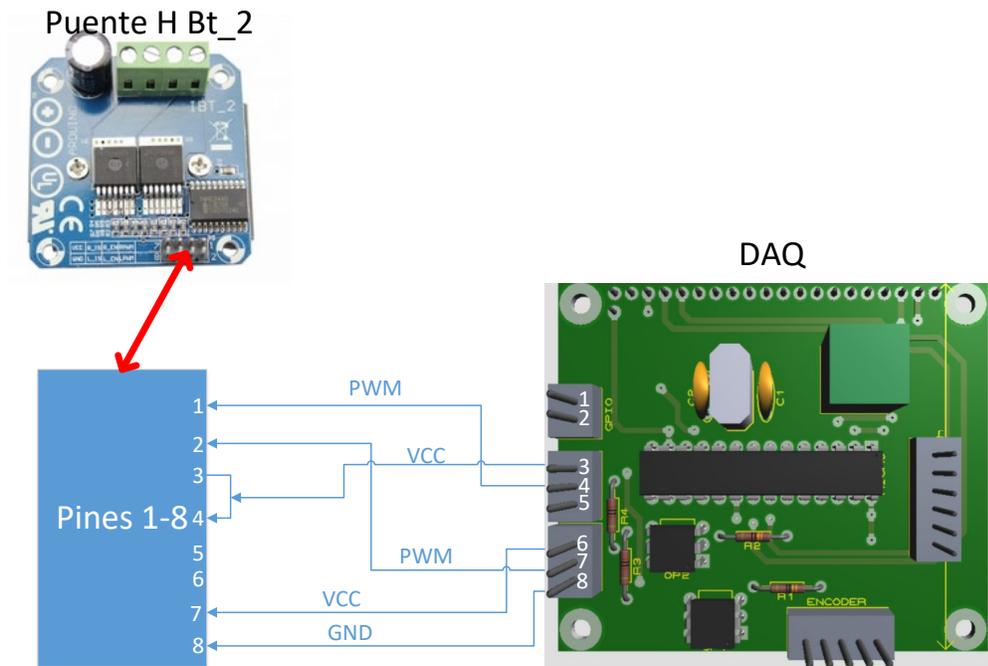


Figura 35. Conexión del puente H y el DAQ.

Para la conexión entre el puente H y el motor utilizado se necesita de una fuente externa de 12 V para que logre funcionar. En la **Figura 36** se muestra la manera en que se cablea el puente H y el motor eléctrico, la cual es muy sencilla de realizar. La terminal positiva de la fuente de alimentación se conecta al pin 9 del puente H mientras que la terminal negativa se conecta al pin 10. De la misma manera el pin 11 del puente H alimenta al motor a través de su pin 1 y finalmente el pin 12 del puente H corresponde a la tierra del motor en el pin 6.

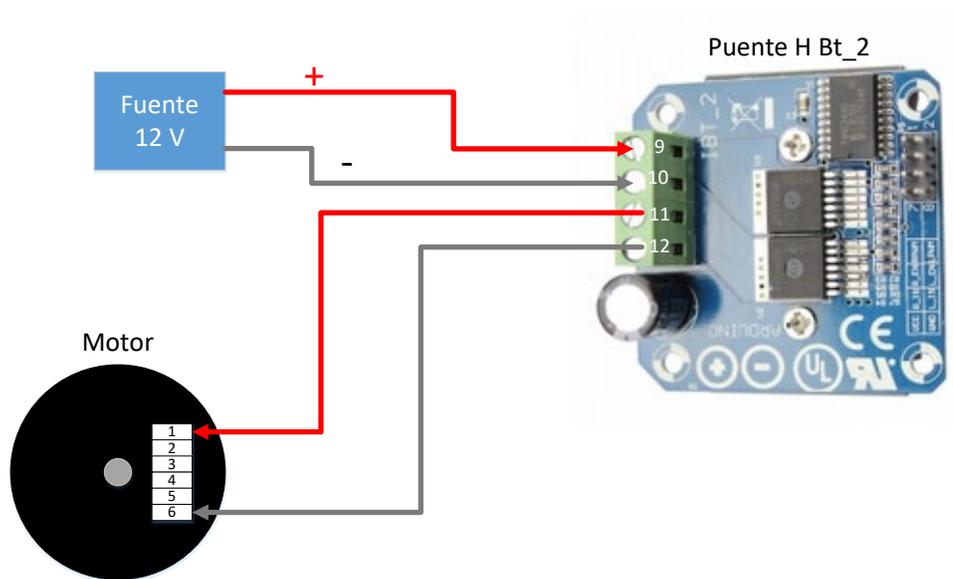


Figura 36. Conexión entre el puente H y el motor.

La elaboración del código principal, el cual se recomienda sea desarrollado en lenguaje C, es realizado en una computadora portátil que está encargada de realizar todos los cálculos del controlador descritos anteriormente.

Se seleccionó una computadora HP, modelo Pavilion G4 con un procesador Intel Pentium Inside B950. Las especificaciones del procesador son las siguientes:

- Dos núcleos.
- Dos hilos.
- Frecuencia base del reloj de 2 GHz.
- Cache de 2 MB L3.
- Velocidad del bus de 5 GT/s.

Además de lo anterior, el sistema operativo de la computadora era Windows 7, y la memoria RAM es de 6 GB. Es importante señalar que no es necesario utilizar el mismo ordenador. Sin embargo, se recomienda elegir uno con al menos las mismas

características para asegurar el correcto funcionamiento del proyecto. Para realizar la conexión entre la computadora y el DAQ se utilizó un convertidor USB-Serial. En la **Figura 37** se muestra que el pin de alimentación de 3.3 volt proveniente del convertidor se conecta al pin 1 de la parte derecha del DAQ. El pin TX del convertidor se conecta al pin 4 del DAQ que en ese caso es RX. El pin Rx del convertidor se conecta al pin TX del DAQ correspondiente al pin 5 y finalmente el pin GND se conecta al pin 3 del DAQ. El convertido se conecta mediante un puerto USB a la computadora.

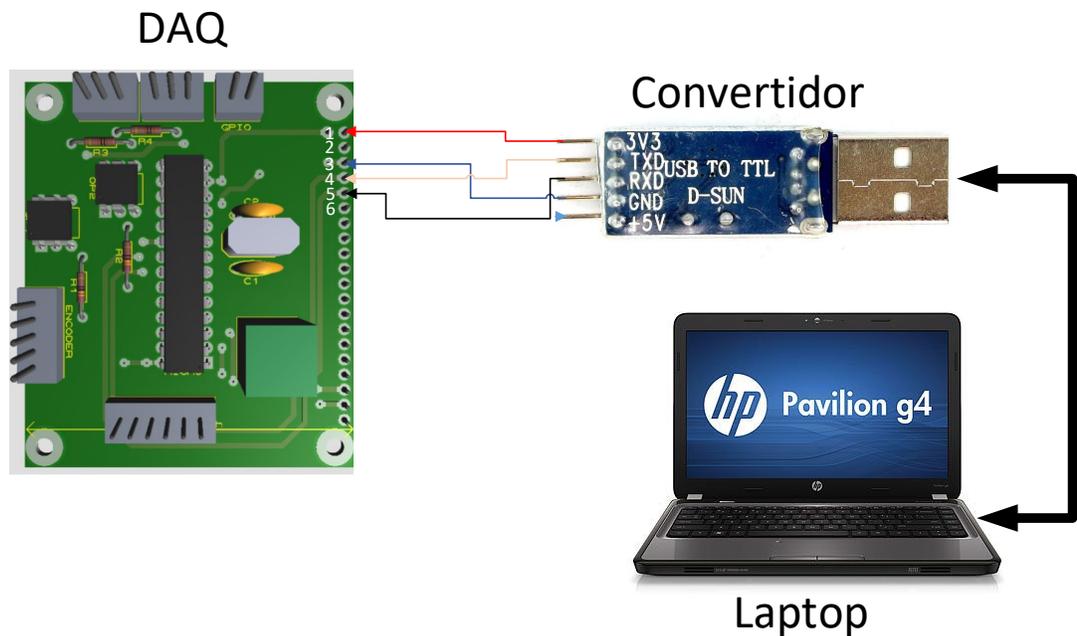


Figura 37. Conexión entre la computadora y el DAQ.

La conexión entre todos los dispositivos se muestra en el diagrama de bloques de la **Figura 38**.

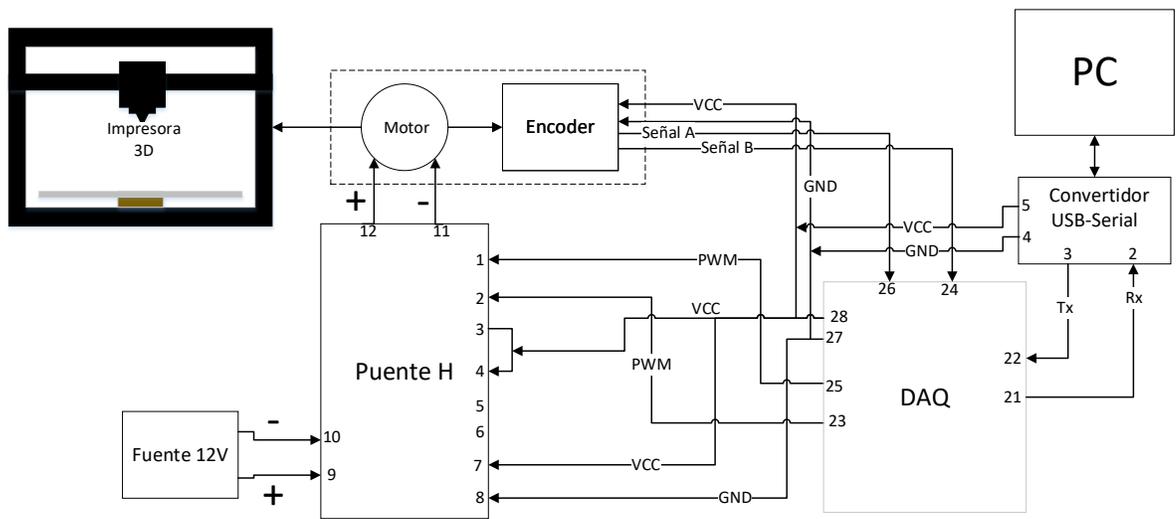


Figura 38. Diagrama de bloques para la conexión de los elementos en el proyecto.

5.8.2 DIAGRAMA DE FLUJO GENERAL

La forma en que actúan cada una de las variables utilizada en el proyecto de tesis se muestra en el diagrama de flujo de la **Figura 39**.

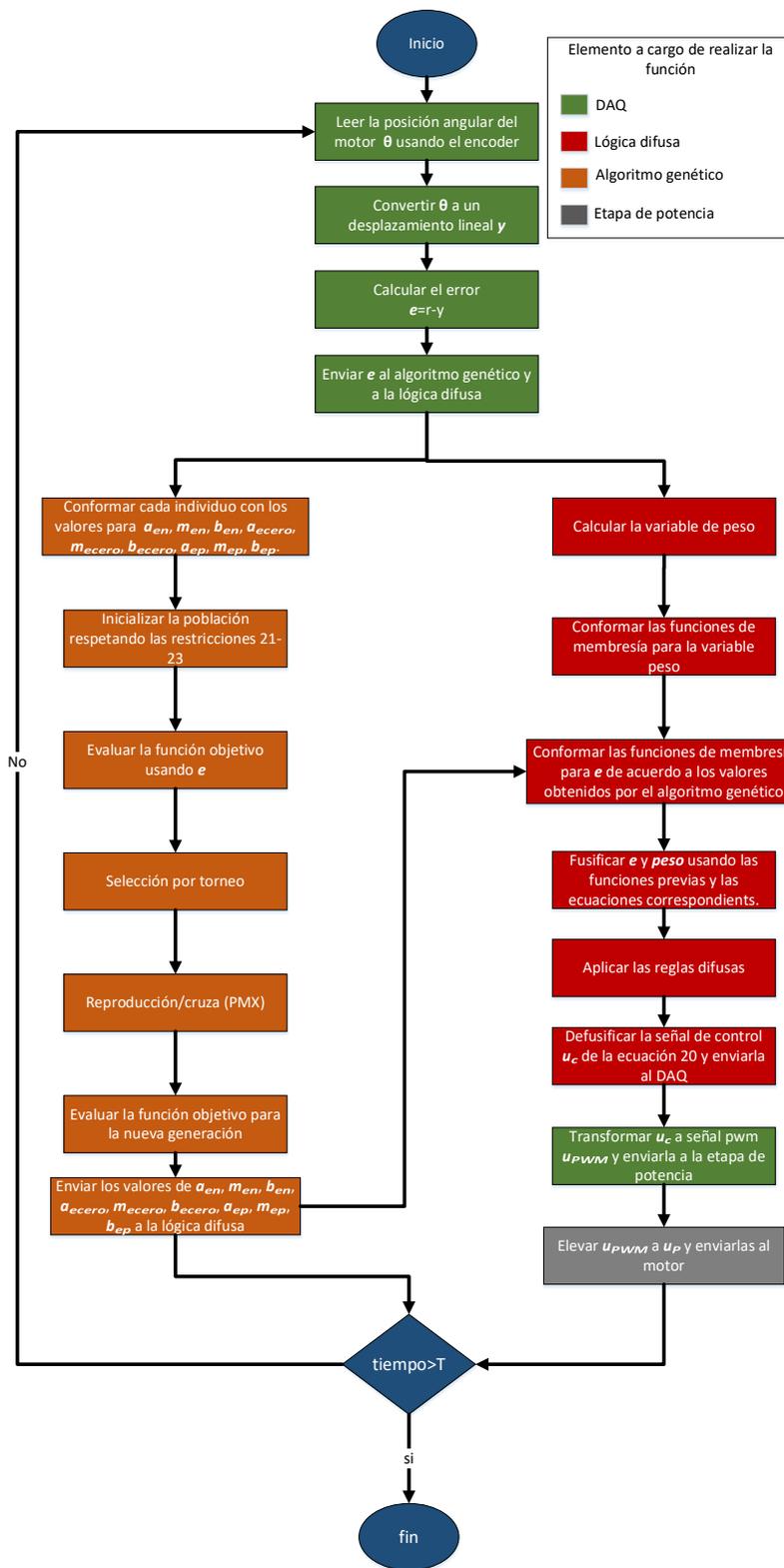


Figura 39. Diagrama de flujo general de trabajo para el proyecto.

6 RESULTADOS

Esta sección reporta los productos obtenidos derivados de la investigación de tesis así como el comportamiento del controlador en la plataforma de experimentación. La sección se divide en dos partes la primera está enfocada a mostrar evidencia de los dispositivos obtenidos y de los artículos que se lograron publicar. La segunda está enfocada en el comportamiento del controlador ante distintos escenarios.

6.1 PRODUCTOS OBTENIDOS

El DAQ fue desarrollado totalmente para el proyecto de investigación, pero debido a sus características puede ser utilizado para múltiples proyectos. La **Figura 40** muestra el DAQ en su forma real.

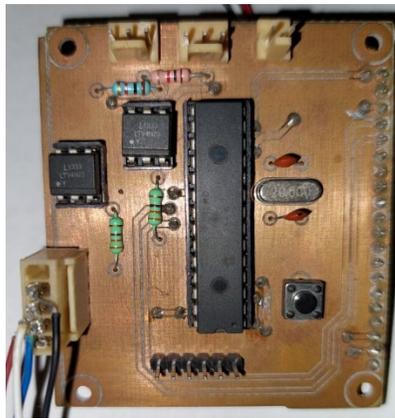


Figura 40. Imagen del DAQ real.

El segundo producto obtenido fue la plataforma de experimentación con todos sus elementos ya integrados. La **Figura 41** muestra la impresora 3D utilizada y todos los elementos que completan la plataforma de experimentación.

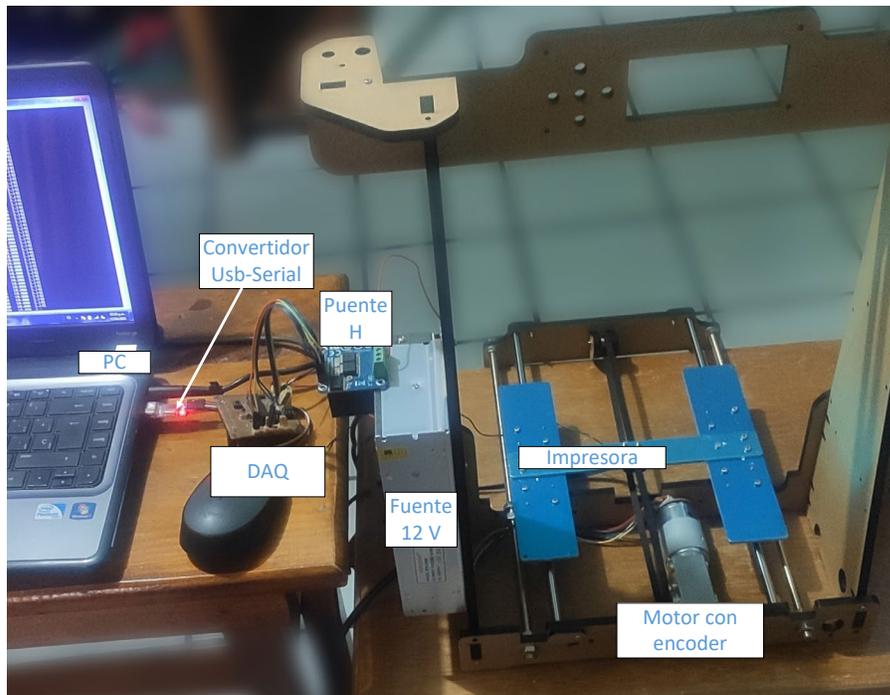


Figura 41. Plataforma de experimentación real utilizada para el proyecto.

Artículos publicados

Durante el periodo de investigación se publicaron 4 artículos relacionados con el tema de tesis, dos de ellos en revistas indexadas. Los títulos, lugar y fecha de publicación se muestran a continuación.

- Comparación Control PID Clásico y Neuro PID. 2° Congreso Nacional de Computación y Tecnología Educativa. Noviembre de 2018, Juriquilla, Querétaro.
- Fuzzy Controller Oriented to Direct Current Motors. XVII Congreso Internacional de Ingeniería (CONIIN). Mayo 2021, Cerro de las Campanas, Querétaro.
- A Hands-On Laboratory for Intelligent Control Courses. Appl. Sci. 2020, 10, 9070 (Torres-Salinas et al., 2020).

- Fuzzy Logic and Genetic-Based Algorithm for a Servo Control System. *Micromachines* 2022, 13, 586 (Torres-Salinas, Rodríguez-Reséndiz, Cruz-Miguel, & Ángeles-Hurtado, 2022).

6.2 RESULTADOS OBTENIDOS POR EL CONTROLADOR

El controlador se sometió a cuatro escenarios distintos para comprobar su desempeño los cuales fueron:

- Escenario 1: sin peso en la base y un único valor deseado.
- Escenario 2: sin peso en la base y varios valores deseados.
- Escenario 3: peso de 5 libras en la base y un único valor deseado.
- Escenario 4: peso de 5 libras en la base y múltiples valores deseados.

Los resultados de posición obtenidos por el controlador para el escenario 1 y 3 se muestran en la **Figura 42**. Los errores que se obtuvieron para estos mismos escenarios se muestran en la **Figura 43** y finalmente la señal de control se muestra en la **Figura 44**.

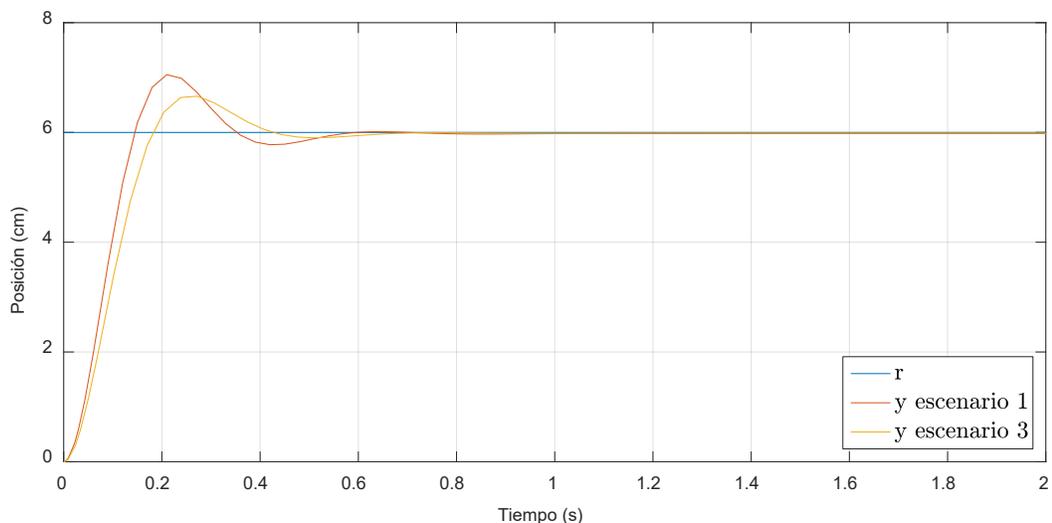


Figura 42. Desempeño del controlador en el escenario 1 y escenario 3.

De la misma manera los resultados de posición para el escenario 2 y 4 se exponen en la **Figura 45**. La **Figura 46** muestra los errores obtenidos y finalmente la señal de control se exhibe en la **Figura 47**.

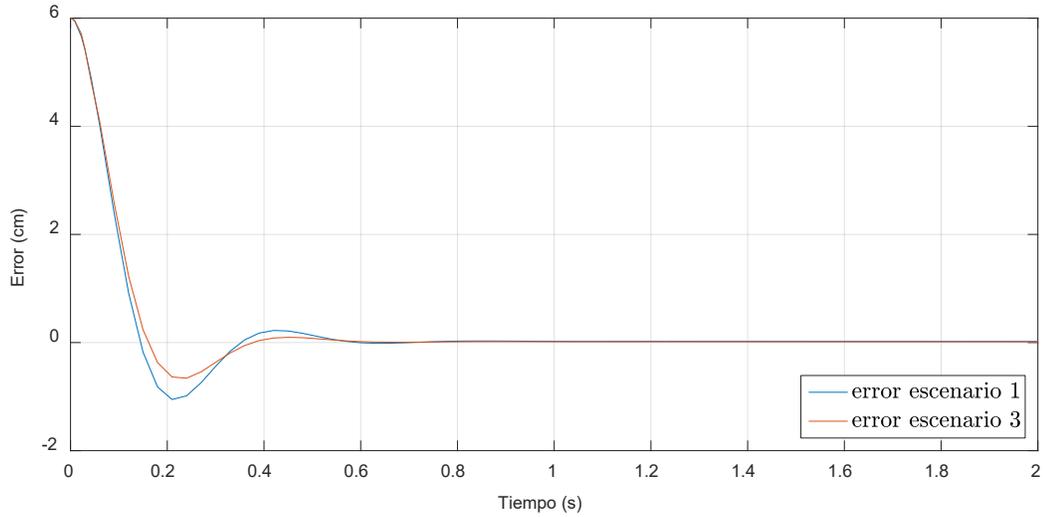


Figura 43. Errores obtenidos en el escenario 1 y escenario 3.

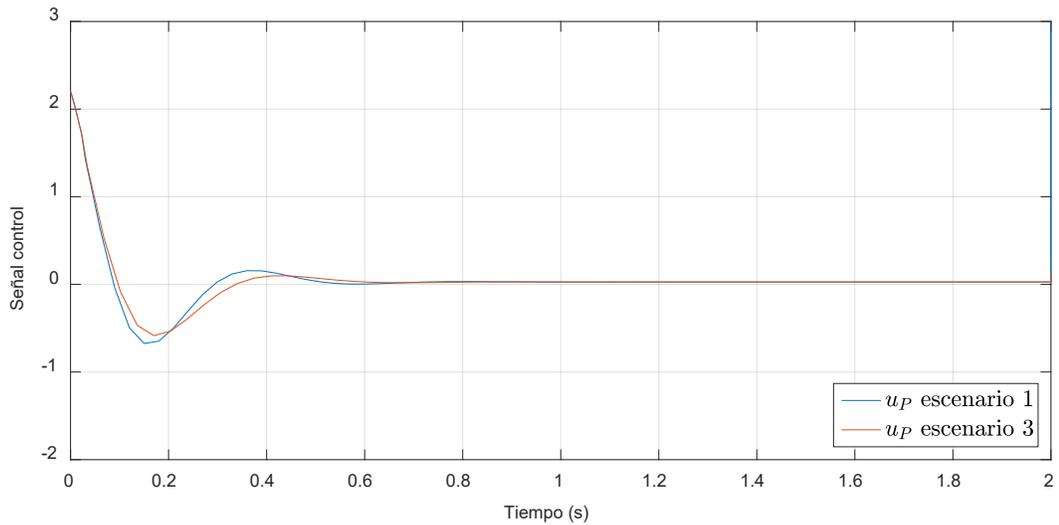


Figura 44. Señal de control calculada para escenario 1 y escenario 3.

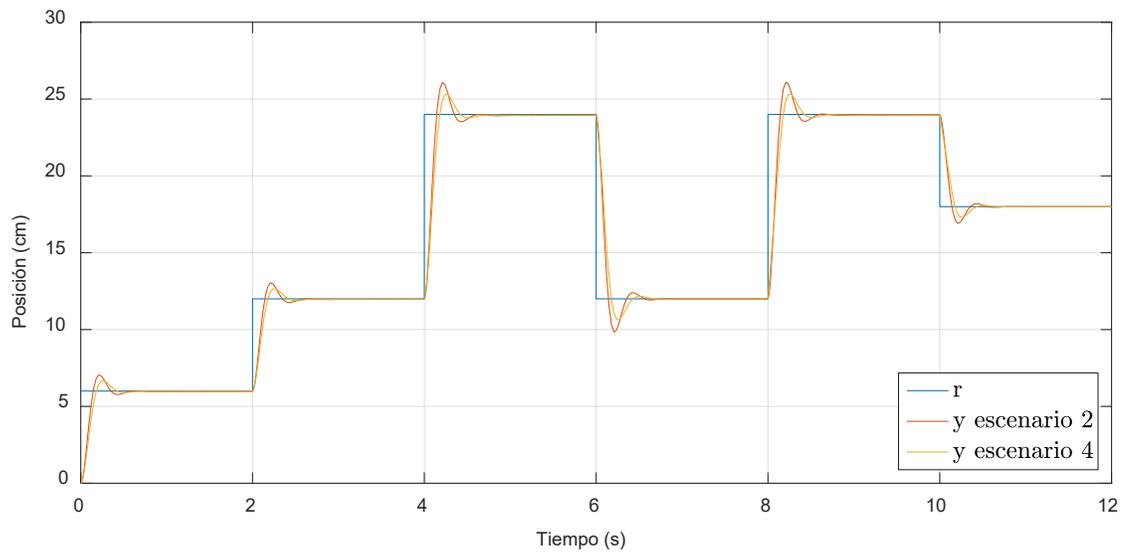


Figura 45. Desempeño del controlador en el escenario 2 y escenario 4.

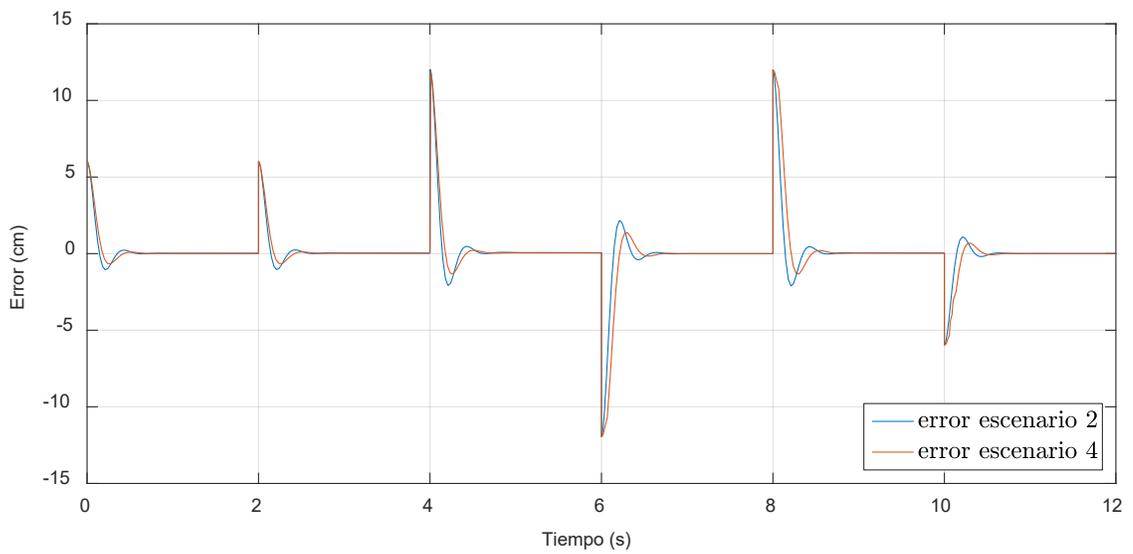


Figura 46. Errores obtenidos en el escenario 2 y escenario 4.

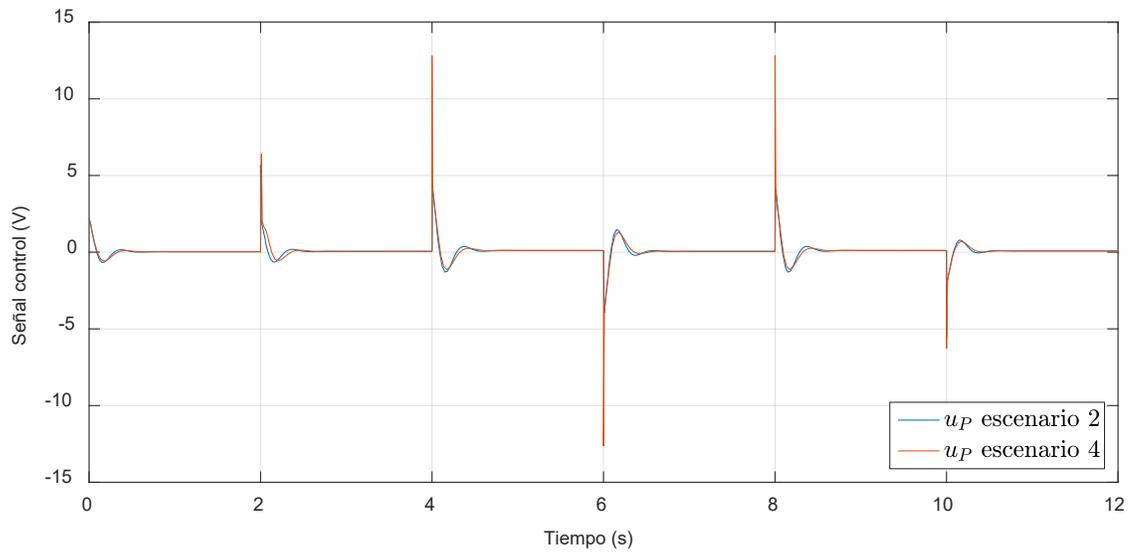


Figura 47. Señal de control calculada para escenario 2 y escenario 4.

La superficie de control obtenida por el controlador difuso se muestra en la **Figura 48**.

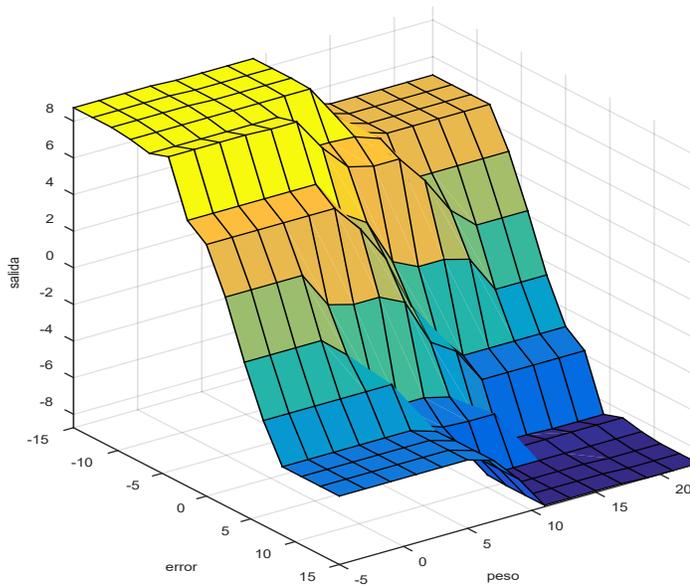


Figura 48. Superficie de control obtenida para el controlador difuso.

7 DISCUSIÓN

Como se muestra en la **Figura 42** y en la **Figura 45**, el controlador obtiene el valor deseado y se estabiliza, cumpliendo así el objetivo principal en cada uno de los escenarios de experimentación. La **Figura 42** también muestra que en el escenario 1 el tiempo de subida se alcanza más rápido que en el escenario 3, esta diferencia se debe al peso agregado a la base de la impresora 3D, lo que hace que el sistema se mueva un poco más lento. En ambos casos, el sobrepaso es inferior al 15% y el número de oscilaciones es pequeño, lo que implica que el consumo energético del sistema es adecuado.

La **Figura 45** muestra que el controlador logra seguir una trayectoria predefinida, alcanzando el valor de referencia dentro del tiempo establecido en cada caso. Este resultado demuestra la efectividad del controlador ante variaciones en las condiciones iniciales del proceso.

La **Figura 43** y la **Figura 46** muestran la estimación de los errores obtenidos para cada escenario, reduciéndolos en cada caso. Las oscilaciones presentadas en estas figuras son representativas de los todos los escenarios en donde todos los errores alcanzan el valor deseado en menos de 2 s. Del mismo modo, se elimina el error en los casos en que el objetivo es seguir un camino previamente definido.

La señal de control mostrada en la **Figura 44** y en la **Figura 47** hace referencia al voltaje que entrega el controlador por lo que es indicativo directo de la energía que se gasta en el proceso del sistema. Estas gráficas muestran que la señal de control es proporcional a los errores obtenidos anteriormente y se puede observar que comienza con valores altos, con un número de oscilaciones pequeño y estabilizándose antes de 2 s.

Finalmente la **Figura 48** muestra la superficie de control. Esta gráfica es resultado obtenido entre la relación de las variables difusas, las reglas de inferencia y su transformación. Esta gráfica proporciona una visión rápida del comportamiento

general del controlador diseñado. Es una herramienta esencial que permite ver que la selección de parámetros es adecuada.

La **Tabla 7** muestra los resultados obtenidos en trabajos que utilizan técnicas similares. El escenario con el que se compara es el 1, ya que es el que más semejanza tiene con otras investigaciones en cuanto a las condiciones a las que se somete el controlador.

Trabajo	Tiempo de subida (s)	Sobrepaso (%)	Tiempo asentamiento (s)	Técnica
Propuesta	0.15	15	0.7	Lógica difusa y algoritmos genéticos
(Khan et al., 2008)	0.155	4.84	0.252	Lógica difusa y algoritmos genéticos
(Bouallègue et al., 2012)	0.21	15	0.64	Lógica difusa y algoritmo PSO
(Peng et al., 2011)	0.4	12.1	0.62	PID y algoritmos genéticos
(Bassi et al., 2011)	0.418	17.4	3.17	Controlador PID y algoritmo PSO

Tabla 7. Comparación del desempeño del controlador propuesto con técnicas similares previas.

Los principales aportes de obtenidos en los resultados se enlistan a continuación:

- Se logró implementar el controlador en una plataforma de experimentación real compuesta por una impresora 3D y elementos de adquisición de datos y potencia.

- El controlador alcanza los valores deseados incluso cuando cambian las condiciones iniciales del sistema.
- El error se reduce a 0 en cada uno de los escenarios propuestos.
- Los tiempos de subida y de asentamiento son competitivos en comparación con técnicas similares en la literatura.

La relevancia de los resultados implica que la propuesta presentada en esta investigación es una excelente alternativa a las diseñadas anteriormente ya que promueve la formación de conocimientos en IA, electrónica, programación y control. Además, los resultados obtenidos muestran que la técnica propuesta supera efectivamente las limitaciones que motivaron esta investigación ya que aunque se modifiquen las condiciones iniciales del sistema, el controlador aún alcanza las referencias predefinidas.

8 CONCLUSIONES

El proyecto de tesis consistió en desarrollar un sistema de controlador inteligente para regular la posición. El controlador implementado está basado en lógica difusa y se usa un algoritmo genético para optimizar las funciones de membresía y, debido a que se orientó el trabajo a motores eléctricos, se implementó en una impresora 3D.

Se puede concluir que el controlador híbrido propuesto es una excelente opción para obtener los valores deseados. Aunque mostró comportamientos ligeramente diferentes, puede alcanzar el valor deseado incluso cuando se modifican las condiciones iniciales como el peso o la trayectoria. La lógica difusa tiene la limitación de que una vez definidos sus parámetros es imposible modificarlos aunque cambie la dinámica del sistema. Añadiendo inteligencia artificial se consigue superar la limitación anterior.

El trabajo propuesto desarrolla los conocimientos necesarios para implementar métodos de control en plataformas que requieren habilidades en electrónica de potencia, electrónica digital y programación de microcontroladores. El controlador híbrido implementado en este artículo ofrece una técnica novedosa al optimizar, a través de algoritmos genéticos, las funciones de membresía utilizadas en la fusificación del sistema. Además, los resultados obtenidos mostraron que los tiempos de subida y de sedimentación son mejores que los obtenidos por técnicas similares previamente mostradas en la literatura. Por lo anterior, esta investigación es una excelente alternativa a la gran variedad de técnicas previamente estudiadas.

El trabajo futuro buscará incluir algoritmos más complejos y así lograr una técnica más robusta y agregar simulaciones para verificar su desempeño. Además, futuros trabajos también buscarán optimizar las funciones de pertenencia para la derivada del error de forma que cada individuo esté formado por los genes correspondientes al error y su derivada. Finalmente, para futuras previsiones, se espera que el controlador, en su parte difusa, incluya una variable que corresponda al peso añadido a la plataforma de experimentación.

9 BIBLIOGRAFÍA

- Abushawish, A., Hamadeh, M., & Nassif, A. B. (2020). PID Controller Gains Tuning Using Metaheuristic Optimization Methods: A survey. *International Journal of Computers*, 14, 87–95. <https://doi.org/10.46300/9108.2020.14.14>
- Amendola, C. A. M., & Gonzaga, D. P. (2007). Fuzzy-Logic Control System of a Variable-Speed Variable-Pitch Wind-Turbine and a Double-Fed Induction

Generator. In *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)* (pp. 252–257). IEEE.

<https://doi.org/10.1109/ISDA.2007.110>

Baier-Fuentes, H., Cascón-Katchadourian, J., Sánchez, Á. M., Herrera-Viedma, E., & Merigó, J. (2018). A Bibliometric Overview of the International Journal of Interactive Multimedia and Artificial Intelligence. *International Journal of Interactive Multimedia and Artificial Intelligence*, 5(3), 9.

<https://doi.org/10.9781/ijimai.2018.12.003>

Bassi, Mishra, & Omizegba. (2011). Automatic Tuning Of Proportional-Integral-Derivative (Pid) Controller Using Particle Swarm Optimization (Pso) Algorithm. *International Journal of Artificial Intelligence & Applications*, 2(4), 25–34.

<https://doi.org/10.5121/ijaia.2011.2403>

Bouallègue, S., Haggège, J., Ayadi, M., & Benrejeb, M. (2012). PID-type fuzzy logic controller tuning based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 25(3), 484–493.

<https://doi.org/10.1016/j.engappai.2011.09.018>

Camci, E., Kripalani, D. R., Ma, L., Kayacan, E., & Khanesar, M. A. (2018). An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm. *Swarm and Evolutionary Computation*, 41, 1–8.

<https://doi.org/10.1016/j.swevo.2017.10.003>

Chegini, H., Beltran, F., & Mahanti, A. (2021). Fuzzy Logic Based Pasture Assessment Using Weed and Bare Patch Detection. In *Communications in Computer and Information Science* (Vol. 1470 CCIS, pp. 1–18).

https://doi.org/10.1007/978-3-030-88259-4_1

Chegini, H., & Mahanti, A. (2019). A Framework of Automation on Context-Aware Internet of Things (IoT) Systems. In *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion - UCC*

- '19 Companion (pp. 157–162). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/3368235.3368848>
- Chegini, H., Naha, R. K., Mahanti, A., & Thulasiraman, P. (2021). Process Automation in an IoT–Fog–Cloud Ecosystem: A Survey and Taxonomy. *IoT*, 2(1), 92–118. <https://doi.org/10.3390/iot2010006>
- Chin, C. S., & Lin, W. P. (2018). Robust Genetic Algorithm and Fuzzy Inference Mechanism Embedded in a Sliding-Mode Controller for an Uncertain Underwater Robot. *IEEE/ASME Transactions on Mechatronics*, 23(2), 655–666. <https://doi.org/10.1109/TMECH.2018.2806389>
- Demir, B. E., Bayir, R., & Duran, F. (2016). Real-time trajectory tracking of an unmanned aerial vehicle using a self-tuning fuzzy proportional integral derivative controller. *International Journal of Micro Air Vehicles*, 8(4), 252–268. <https://doi.org/10.1177/1756829316675882>
- Dhimish, M., Holmes, V., Mehrdadi, B., & Dales, M. (2018). Comparing Mamdani Sugeno fuzzy logic and RBF ANN network for PV fault detection. *Renewable Energy*, 117, 257–274. <https://doi.org/10.1016/j.renene.2017.10.066>
- Gevarter, W. B. (1987). *Introduction To Artificial Intelligence. Chemical Engineering Progress* (Vol. 83). <https://doi.org/10.1097/00006216-199101530-00006>
- Hernández-Alvarado, R., García-Valdovinos, L., Salgado-Jiménez, T., Gómez-Espinosa, A., & Fonseca-Navarro, F. (2016). Neural Network-Based Self-Tuning PID Control for Underwater Vehicles. *Sensors*, 16(9), 1429. <https://doi.org/10.3390/s16091429>
- Ibrahim, I. N., & Al Akkad, M. A. (2016). Exploiting an intelligent fuzzy-PID system in nonlinear aircraft pitch control. In *2016 International Siberian Conference on Control and Communications (SIBCON)* (pp. 1–7). IEEE. <https://doi.org/10.1109/SIBCON.2016.7491828>
- Jain, D. K., Mahanti, A., Shamsolmoali, P., & Manikandan, R. (2020). Deep neural

- learning techniques with long short-term memory for gesture recognition. *Neural Computing and Applications*, 32(20), 16073–16089.
<https://doi.org/10.1007/s00521-020-04742-9>
- Jigang, H., Jie, W., & Hui, F. (2017). An anti-windup self-tuning fuzzy PID controller for speed control of brushless DC motor. *Automatika*, 58(3), 321–335.
<https://doi.org/10.1080/00051144.2018.1423724>
- Kaveh, A., Bakhshpoori, T., & Afshari, E. (2014). An efficient hybrid Particle Swarm and Swallow Swarm Optimization algorithm. *Computers and Structures*, 143, 40–49. <https://doi.org/10.1016/j.compstruc.2014.07.012>
- Khan, S., Abdulazeez, S. F., Adetunji, L. W., Alam, A. Z., Salami, M. J. E., Hameed, S. A., ... Islam, M. R. (2008). Design and Implementation of an Optimal Fuzzy Logic Controller Using Genetic Algorithm. *Journal of Computer Science*, 4(10), 799–806. <https://doi.org/10.3844/jcssp.2008.799.806>
- Kouro, S., & Musalem, R. (2002). Control mediante lógica difusa. *Técnicas Modernas Automaticas,(1-7)-7*, 1–7. Retrieved from <http://www2.elo.utfsm.cl/~elo377/documentos/Fuzzy.pdf>
- Kumar, V., & Rana, K. P. S. (2017). Comments on “Design of two-layered fractional order fuzzy logic controllers applied to robotic manipulator with variable payload.” *Applied Soft Computing Journal*, 51, 145–146.
<https://doi.org/10.1016/j.asoc.2016.11.035>
- Latah, M., & Toker, L. (2019). Artificial intelligence enabled software-defined networking: a comprehensive overview. *IET Networks*, 8(2), 79–99.
<https://doi.org/10.1049/iet-net.2018.5082>
- Levine, W. S. (1996). PID Control, The Control Handbook. In *The Control Handbook* (pp. 198–210). New Jersey, USA.: IEEE Press: Piscataway.
- Liang, H., Zou, J., Zuo, K., & Khan, M. J. (2020). An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control

- system. *Mechanical Systems and Signal Processing*, 142, 106708.
<https://doi.org/10.1016/j.ymssp.2020.106708>
- Liu, C., Yue, X., & Yang, Z. (2021). Are nonfragile controllers always better than fragile controllers in attitude control performance of post-capture flexible spacecraft? *Aerospace Science and Technology*, 118, 107053.
<https://doi.org/10.1016/j.ast.2021.107053>
- Lu, Q., Sun, Z., Zhang, J., Zhang, J., Zheng, J., & Qian, F. (2022). A Novel Remote-Controlled Vascular Interventional Robotic System Based on Hollow Ultrasonic Motor. *Micromachines*, 13(3), 410.
<https://doi.org/10.3390/mi13030410>
- Mahapatra, S., Daniel, R., Dey, D. N., & Nayak, S. K. (2015). Induction Motor Control Using PSO-ANFIS. *Procedia Computer Science*, 48, 753–768.
<https://doi.org/10.1016/j.procs.2015.04.212>
- Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121(12), 1585. <https://doi.org/10.1049/piee.1974.0328>
- Martins, C. H., Santos, R. P. B. dos, & Santos, F. L. (2012). Simplified particle swarm optimization algorithm. *Acta Scientiarum. Technology*, 34(1), 21–25.
<https://doi.org/10.4025/actascitechnol.v34i1.9679>
- Masoudi, S., Soltanpour, M. R., & Abdollahi, H. (2018). Adaptive fuzzy control method for a linear switched reluctance motor. *IET Electric Power Applications*, 12(9), 1328–1336. <https://doi.org/10.1049/iet-epa.2018.0059>
- Mejía Ramírez, C. A., Montes Rivera, M., Medina Ramírez, R. R., Ramírez Prado, R. M., Gaitán Mercado, C. M., & Ochoa-Zezzatti, A. (2021). Medicine Inventory Control System Through Fuzzy Logic and Genetic Algorithms: Applied to a Biopharmaceutical (pp. 417–435). https://doi.org/10.1007/978-3-030-68655-0_21

- Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence*, 48(4), 805–820. <https://doi.org/10.1007/s10489-017-1019-8>
- Odeh, S. M., Mora, A. M., Moreno, M. N., & Merelo, J. J. (2015). A Hybrid Fuzzy Genetic Algorithm for an Adaptive Traffic Signal System. *Advances in Fuzzy Systems*, 2015, 1–11. <https://doi.org/10.1155/2015/378156>
- Odry, A., Tadic, V. L., & Odry, P. (2021). A Stochastic Logic-Based Fuzzy Logic Controller: First Experimental Results of a Novel Architecture. *IEEE Access*, 9, 29895–29920. <https://doi.org/10.1109/ACCESS.2021.3059482>
- Orozco, A. H., & Montoya, J. O. (2010). Determinación de los parámetros del motor de inducción utilizando un algoritmo de optimización por cúmulo de partículas (PSO), 15–26. Retrieved from <http://repositorio.utp.edu.co/dspace/handle/11059/1732>
- Panda, S., & Padhy, N. P. (2008). Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design. *Applied Soft Computing*, 8(4), 1418–1427. <https://doi.org/10.1016/j.asoc.2007.10.009>
- Peng, J., & Dubay, R. (2011). Identification and adaptive neural network control of a DC motor system with dead-zone characteristics. *ISA Transactions*, 50(4), 588–598. <https://doi.org/10.1016/j.isatra.2011.06.005>
- Ponticelli, G. S., Guarino, S., Tagliaferri, V., & Giannini, O. (2019). An optimized fuzzy-genetic algorithm for metal foam manufacturing process control. *The International Journal of Advanced Manufacturing Technology*, 101(1–4), 603–614. <https://doi.org/10.1007/s00170-018-2942-5>
- Rafa, S., Larabi, A., Barazane, L., Manceur, M., Essounbouli, N., & Hamzaoui, A. (2014). Implementation of a new fuzzy vector control of induction motor. *ISA Transactions*, 53(3), 744–754. <https://doi.org/10.1016/j.isatra.2014.02.005>
- Rath, A. K., Parhi, D. R., Das, H. C., Kumar, P. B., Muni, M. K., & Salony, K.

- (2019). Path optimization for navigation of a humanoid robot using hybridized fuzzy-genetic algorithm. *International Journal of Intelligent Unmanned Systems*, 7(3), 112–119. <https://doi.org/10.1108/IJIUS-11-2018-0032>
- Rezaee Jordehi, A., Jasni, J., Abd Wahab, N., Kadir, M. Z., & Javadi, M. S. (2015). Enhanced leader PSO (ELPSO): A new algorithm for allocating distributed TCSC's in power systems. *International Journal of Electrical Power and Energy Systems*, 64, 771–784. <https://doi.org/10.1016/j.ijepes.2014.07.058>
- Rodriguez-Abreo, O., Hernandez-Paredes, J. M., Rangel, A. F., Fuentes-Silva, C., & Velasquez, F. A. C. (2021). Parameter Identification of Motors by Cuckoo Search Using Steady-State Relations. *IEEE Access*, 9, 72017–72024. <https://doi.org/10.1109/ACCESS.2021.3078578>
- Rodriguez-Abreo, O., Rodriguez-Resendiz, J., Fuentes-Silva, C., Hernandez-Alvarado, R., & Falcon, M. D. C. P. T. (2021). Self-Tuning Neural Network PID With Dynamic Response Control. *IEEE Access*, 9, 65206–65215. <https://doi.org/10.1109/ACCESS.2021.3075452>
- Rodríguez-Abreo, O., Rodríguez-Reséndiz, J., Montoya-Santiyanes, L. A., & Álvarez-Alvarado, J. M. (2021). Non-Linear Regression Models with Vibration Amplitude Optimization Algorithms in a Microturbine. *Sensors*, 22(1), 130. <https://doi.org/10.3390/s22010130>
- Sierra-García, J. E., & Santos, M. (2021). Switched learning adaptive neuro-control strategy. *Neurocomputing*, 452, 450–464. <https://doi.org/10.1016/j.neucom.2019.12.139>
- Suman, S. K., & Giri, V. K. (2016). Speed control of DC motor using optimization techniques based PID Controller. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)* (pp. 581–587). IEEE. <https://doi.org/10.1109/ICETECH.2016.7569318>
- Tabrez, M., Sadhu, P. K., Hossain Lipu, M. S., Iqbal, A., Husain, M. A., & Ansari, S.

- (2021). Power Conversion Techniques Using Multi-Phase Transformer: Configurations, Applications, Issues and Recommendations. *Machines*, 10(1), 13. <https://doi.org/10.3390/machines10010013>
- Tadic, V., Odry, A., Burkus, E., Kecskes, I., Kiraly, Z., Klincsik, M., ... Odry, P. (2021). Painting Path Planning for a Painting Robot with a RealSense Depth Sensor. *Applied Sciences*, 11(4), 1467. <https://doi.org/10.3390/app11041467>
- Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1), 116–132. <https://doi.org/10.1109/TSMC.1985.6313399>
- Tiwari, S., Bhatt, A., Unni, A. C., Singh, J. G., & Ongsakul, W. (2018). Control of DC Motor Using Genetic Algorithm Based PID Controller. In *2018 International Conference and Utility Exhibition on Green Energy for Sustainable Development (ICUE)* (pp. 1–6). IEEE. <https://doi.org/10.23919/ICUE-GESD.2018.8635662>
- Torres-Salinas, H., Rodríguez-Reséndiz, J., Cruz-Miguel, E. E., & Ángeles-Hurtado, L. A. (2022). Fuzzy Logic and Genetic-Based Algorithm for a Servo Control System. *Micromachines*, 13(4), 586. <https://doi.org/10.3390/mi13040586>
- Torres-Salinas, H., Rodríguez-Reséndiz, J., Estévez-Bén, A. A., Cruz Pérez, M. A., Sevilla-Camacho, P. Y., & Perez-Soto, G. I. (2020). A Hands-On Laboratory for Intelligent Control Courses. *Applied Sciences*, 10(24), 9070. <https://doi.org/10.3390/app10249070>
- Varshney, A., Gupta, D., & Dwivedi, B. (2017). Speed response of brushless DC motor using fuzzy PID controller under varying load condition. *Journal of Electrical Systems and Information Technology*, 4(2), 310–321. <https://doi.org/10.1016/j.jesit.2016.12.014>

- Veselý, V., & Ilka, A. (2013). Gain-scheduled PID controller design. *Journal of Process Control*, 23(8), 1141–1148.
<https://doi.org/10.1016/j.jprocont.2013.07.002>
- Xiao-Jun Ma, Zeng-Qi Sun, & Yan-Yan He. (1998). Analysis and design of fuzzy controller and fuzzy observer. *IEEE Transactions on Fuzzy Systems*, 6(1), 41–51. <https://doi.org/10.1109/91.660807>
- Yang, C., Wang, Y., & Fan, W. (2022). Long Stroke Design of Piezoelectric Walking Actuator for Wafer Probe Station. *Micromachines*, 13(3), 412.
<https://doi.org/10.3390/mi13030412>
- Young-Moon Park, Myeon-Song Choi, & Lee, K. Y. (1996). An optimal tracking neuro-controller for nonlinear dynamic systems. *IEEE Transactions on Neural Networks*, 7(5), 1099–1110. <https://doi.org/10.1109/72.536307>
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353.
[https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- Zhang, T., Liu, Y., Rao, Y., Li, X., & Zhao, Q. (2020). Optimal design of building environment with hybrid genetic algorithm, artificial neural network, multivariate regression analysis and fuzzy logic controller. *Building and Environment*, 175, 106810. <https://doi.org/10.1016/j.buildenv.2020.106810>
- Zhang, Y., Huang, Y., & Wang, Y. (2022). Research on Compound PID Control Strategy Based on Input Feedforward and Dynamic Compensation Applied in Noncircular Turning. *Micromachines*, 13(2), 341.
<https://doi.org/10.3390/mi13020341>