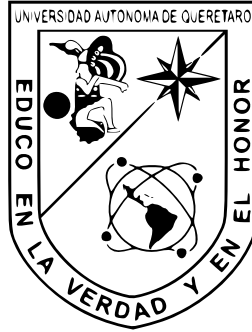


UNIVERSIDAD AUTÓNOMA DE QUERÉTARO



FACULTAD DE INGENIERÍA
MAESTRÍA EN CIENCIAS EN CONTROL EN SISTEMAS EMBEBIDOS

Control de posición de un robot SCARA neumático de 4 grados de libertad

TESIS

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias en Control en Sistemas Embebidos

Presenta:

NOEL CHÁVEZ LÓPEZ

Dirigido por:

DR. JUAN MANUEL RAMOS ARREGUÍN

Querétaro, Qro. a Octubre de 2025

La presente obra está bajo la licencia:
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>



CC BY-NC-ND 4.0 DEED

Atribución-NoComercial-SinDerivadas 4.0 Internacional

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

Bajo los siguientes términos:



Atribución — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.



NoComercial — Usted no puede hacer uso del material con [propósitos comerciales](#).



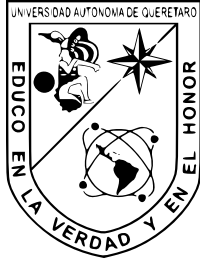
SinDerivadas — Si [remezcla, transforma o crea a partir](#) del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni [medidas tecnológicas](#) que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una [excepción o limitación](#) aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como [publicidad, privacidad, o derechos morales](#) pueden limitar la forma en que utilice el material.



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Ciencias en Control en Sistemas Embebidos

Control de posición de un robot SCARA neumático de 4 grados de libertad

TESIS

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias en Control en Sistemas Embebidos

Presenta:

Noel Chávez López

Dirigido por:

Dr. Juan Manuel Ramos Arreguín

SINODALES

Dr. Juan Manuel Ramos Arreguín

Presidente

Dr. Juvenal Rodríguez Reséndiz

Codirector

Dr. Efrén Gorrostieta Hurtado

Vocal

Dr. Jesús Carlos Pedraza Ortega

Suplente

Dr. Marco Antonio Aceves Fernández

Suplente

Centro Universitario

Querétaro, QRO

México.

Octubre 2025

Abreviaturas y siglas

ACRPPC	Control de rendimiento prescrito con tasa de convergencia ajustable (Adjustable Convergence Rate Prescribed Performance Control)
ADC	Convertidor analógico digital (Analog Digital Converter)
AIA	Algoritmo de Inteligencia Artificial
ANN	Red neuronal artificial (Artificial Neural Network)
D-H	Denavit-Hartenberg
DOF	Grados de libertad (Degrees Of Freedom)
FFC	Compensación anticipada (FeedForward Compensation)
FPGA	Matriz de compuertas programables en campo (Field Programmable Gate Array)
LQR	Regulador cuadrático lineal (Linear Quadratic Regulator)
MAE	Error absoluto medio (Mean Absolute Error)
MPC	Controlador de modelo predictivo (Model Predictive Controller)
NCD-PID	Diseño de controlador no lineal proporcional integral derivativo (Non-linear Controller Design Proportional Integral Derivative)
NN	Red neuronal (Neural Network)
PD	Proporcional Derivativo
PI	Proporcional Integral
PIB	Producto Interno Bruto
PID	Proporcional Integral Derivativo
PIDOF	Proporcional Integral Derivativo de Orden Fraccional
PMA	Musculo neumático artificial (Pneumatic Artificial Muscle)
PSO	Optimización de intercambio de partículas (Particle Swap Optimization)
SCARA	Brazo robótico articulado de cumplimiento selectivo (Selective Compliance Articulated Robot Arm)
UNESCO	Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (United Nations Educational, Scientific and Cultural Organization)

Resumen

Los sistemas con actuadores neumáticos suelen ser complejos al aplicar el control clásico, ya que su comportamiento no lineal impide el uso directo de estos controladores. Por lo tanto, es necesario aplicar técnicas de linealización o técnicas de control diferentes donde estas no linealidades no representen un problema. Los AIA, como el control difuso y un controlador basado en NN, no requieren el uso de un modelo matemático, lo que los convierte en herramientas adecuadas para un sistema no lineal. Por ello, esta investigación compara el error de posicionamiento entre un controlador PID y controladores AIA, PID difuso y NN, en un robot SCARA de cuatro grados de libertad. También se realiza una comparación entre los resultados obtenidos e investigaciones similares utilizando un controlador clásico. El objetivo de la investigación es determinar si un controlador AIA tiene un mejor rendimiento en comparación con los controladores clásicos, como lo es el PID. Los resultados experimentales muestran que, en comparación con un controlador PID, la respuesta, al utilizar un AIA, tiene un mejor rendimiento con un MAE de 5.44, -0.85 y -0.88 grados para el controlador PID, PID Difuso y NN respectivamente, para el primer grado de libertad, 2.53, 0.46 y -0.06 para el segundo grado de libertad y -4.08, 0.46 y -0.86 para el tercer grado de libertad, un máximo de 0.45 % de error de estado estacionario para el PID Difuso, 0.46 % para NN en comparación con un 2.73 % del PID y reduciendo el error de posición en al menos el 75 %. En comparación con otras investigaciones, los resultados obtenidos muestran que, comparado con un controlador PID tradicional, un controlador AIA reduce el error de posicionamiento en al menos un 30.52 %.

Abstract

Systems with pneumatic actuators are usually complicated when applying classical control, since their non-linear behavior does not allow the use of these controllers directly. Therefore, it is necessary to apply techniques to linearize or different control techniques in which these nonlinearities are not a problem. AIA such as Fuzzy control and a controller based on a NN do not require the use of a mathematical model, making them appropriate tools for a nonlinear system. This is why this research makes a comparison of the positioning error between a PID controller and AIA controllers, Fuzzy-PID and NN, in a SCARA robot with 4 DOF. Also, a comparison between the results obtained and similar researches using the PID controller. The goal of the research is to determine if an AIA controller has a better performance compared to classical controllers such as PID. The experimental results show that, compared to a PID controller, the response, when using an AIA, has a better performance with a MAE of 5.44, -0.85 and -0.88 degrees for PID, Fuzzy-PID and NN controller respectively, for the first DOF, 2.53, 0.46 and -0.06 for the second DOF and -4.08, 0.46 and -0.86 for the third DOF, a maximum of 0.45 % of steady-state error for Fuzzy-PID, 0.46 % for NN compared with a 2.73 % of PID and reducing the position error in more than 75 %. Compared with other researches, the results obtained show that compared with a traditional PID controller, an AIA controller reduces the positioning error in at least 30.52 %.

Agradecimientos

Agradezco a todas las personas que me apoyaron e hicieron posible la culminación de esta etapa. En primer lugar, agradezco a mi director de tesis, Dr. Juan Manuel Ramos Arreguín por su guía, apoyo y sugerencias durante todo el desarrollo de este trabajo. Agradezco a la Universidad Autónoma de Querétaro por brindarme las herramientas y los espacios necesarios para llevar a cabo esta investigación. Agradezco a mi familia, por su apoyo incondicional, su paciencia y por acompañarme en cada paso de este camino. Finalmente, agradezco a la SECIHTI por el apoyo económico brindado, sin el cual este proyecto no hubiera sido posible.

Índice general

1. Introducción	1
1.1. Descripción del problema	1
1.2. Justificación	3
1.3. Hipótesis y Objetivos	4
1.3.1. Hipótesis	4
1.3.2. Objetivo general	4
1.3.3. Objetivos particulares	4
1.4. Antecedentes	5
1.5. Estructura de la tesis	7
2. Marco Teórico	8
2.1. Cinemática del robot	8
2.2. Sistemas lineales	11
2.3. Sistemas no lineales	12
2.4. Controlador PID	12
2.5. Control PID Difuso	13
2.6. Red Neuronal de retropropagación	14
3. Metodología	16
3.1. Estudio de técnicas del Estado del Arte	17
3.2. Revisión del hardware del robot neumático	18
3.3. Cinemática directa del robot neumático	18
3.4. Adecuaciones al robot	19
3.5. Hardware eléctrico	21
3.6. Algoritmo de Inteligencia Artificial (IA)	21
3.6.1. Controlador PID	22
3.6.2. Controlador PID difuso	22
3.6.3. Controlador con red neuronal	25
3.7. Pruebas	27
4. Resultados	28
4.1. Control PID	28
4.2. Control PID difuso	31
4.3. Control con Red neuronal de retropropagación	33

4.4. Comparativa	35
4.5. Discusión	37
5. Conclusiones	38
Referencias	41

Índice de figuras

1.1. Instalación anual de robots industriales [1].	2
1.2. Robots instalados en 2022 [1].	3
2.1. Robot neumático.	10
2.2. Proyección geométrica.	11
3.1. Diagrama de la metodología.	16
3.2. Robot neumático.	18
3.3. Pieza impresa para acople de potenciómetros.	19
3.4. Pieza impresa para sujeción de actuadores y eslabones.	20
3.5. Comparativa entre microcontroladores.	20
3.6. Electroválvulas utilizadas.	21
3.7. Diagrama electroneumático del robot.	22
3.8. Diagrama de bloques de controlador PID.	22
3.9. Diagrama de bloques de controlador PID difuso.	23
3.10. Función de pertenencia de las entradas.	23
3.11. Función de pertenencia de las salidas K_p	24
3.12. Función de pertenencia de las salidas K_i	24
3.13. Diagrama de bloques de controlador con red neuronal de retropropagación.	25
3.14. Topología de la red neuronal de retropropagación.	26
4.1. Posición de primer grado de libertad con controlador PID.	29
4.2. Posición de segundo grado de libertad con controlador PID.	29
4.3. Posición de tercer grado de libertad con controlador PID.	29
4.4. Errores por cada grado de libertad del controlador PID.	30
4.5. Posición del efector final con controlador PID.	30
4.6. Posición de primer grado de libertad con controlador PID difuso.	31
4.7. Posición de segundo grado de libertad con controlador PID difuso.	31
4.8. Posición de tercer grado de libertad con controlador PID difuso.	32
4.9. Errores por cada grado de libertad del controlador PID difuso.	32
4.10. Posición del efector final con controlador PID Difuso.	33
4.11. Posición de primer grado de libertad con controlador NN.	33
4.12. Posición de segundo grado de libertad con controlador NN.	34
4.13. Posición de tercer grado de libertad con controlador NN.	34
4.14. Errores por cada grado de libertad del controlador con red neuronal.	35

4.15. Posición del efector final con controlador de red neuronal.	35
4.16. Comparativa de MAE de controladores por cada grado de libertad.	36

Índice de tablas

1.1. Estado del Arte	5
2.1. Variables usadas en Figura 2.1 y 2.2.	10
3.1. Componentes y materiales.	18
3.2. Componentes eléctricos.	21
3.3. Valores numéricos de funciones de pertenencia por grado de libertad.	25
3.4. Pesos iniciales y factor de aprendizaje de neurona por grado de libertad.	26
4.1. Porcentaje de error de cada controlador por cada grado de libertad.	36
4.2. Error en estado estacionario, tiempo de asentamiento y sobrepaso máximo.	36
4.3. Porcentaje de error obtenido en otras investigaciones.	37

Introducción

Los actuadores neumáticos son usados ampliamente en la industria, ya que estos tienen menores costos, un mayor rendimiento potencia-peso, son más limpios, requieren una menor cantidad de mantenimientos programados, sus diseños son más simples y no son afectados por la interferencia electromagnética en comparación con los actuadores eléctricos. El problema con este tipo de actuadores es que, al trabajar con aire, presentan no linealidades como la alta fuerza de fricción, la compresibilidad del aire y las zonas muertas de las válvulas [2][3].

Los algoritmos de inteligencia artificial, como lo es el control difuso, han sido de gran ayuda para resolver problemas de sistemas no lineales en los que un control clásico no puede dar una solución al no tener un modelo matemático. Estos algoritmos han tenido varias aplicaciones en las que ha sido de gran utilidad, y apoyados con un sistema embebido, son una herramienta ideal para un control neumático de un robot SCARA de cuatro grados de libertad [4].

En los siguientes capítulos se abarcará a mayor detalle algunas investigaciones que se han desarrollado sobre este tema analizando los antecedentes, más específicamente robots con actuadores neumáticos, y se analizarán sus contribuciones en este ámbito. Esto permitirá tener una visión más amplia de lo que se ha realizado referente a estos sistemas. Se hablará de los problemas que se pueden llegar a encontrar con el uso de actuadores neumáticos y se resaltarán los beneficios que podrían llegar a tenerse. Se definirá de una manera clara el problema que esta investigación pretende resolver y se incluirán las limitaciones y obstáculos que se podrían llegar a presentar por el uso de este tipo de actuadores. También se realizará un análisis del estado del arte para comprender el panorama actual de la investigación relacionada con actuadores neumáticos y su control.

1.1. Descripción del problema

La demanda de robots industriales va en crecimiento cada año, teniendo una pequeña caída en 2019 por cuestiones de la pandemia por COVID-19. Las cifras de instalaciones de robots industriales a nivel mundial aumentaron un 31 % de 2016 a 2022, siendo las industrias de metales, automotriz, de plásticos y eléctricas/electrónicas las de mayor crecimiento con un 45 %, 42 %, 29 % y 24 % respectivamente, en el año 2022 (Figura 1.1). Este crecimiento fue dado principalmente por China, el cual tuvo un crecimiento de 51 %, y México siendo uno de los 10 mejores a nivel mundial con un crecimiento de 61 %, pero quedando muy debajo de China, el cual obtuvo 268 mil unidades

instaladas mientras que México quedó dentro de las 5 mil unidades (Figura 1.2)[?].

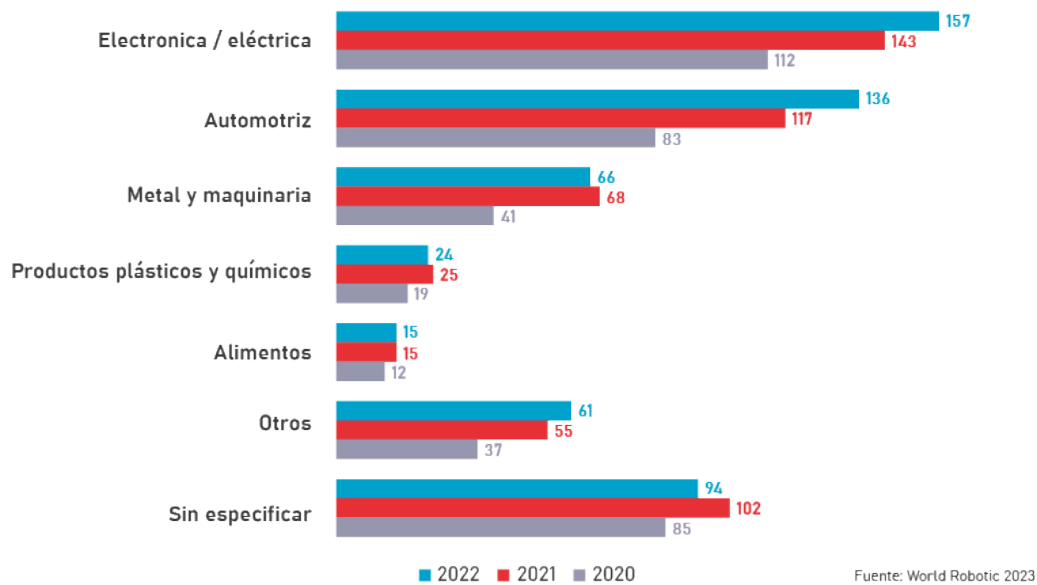


Figura 1.1: Instalación anual de robots industriales [1].

Se puede observar que la robótica en México ha crecido de una manera importante a nivel mundial. Estas cifras dan a entender la necesidad que hay a nivel mundial y a nivel nacional del desarrollo de nuevas tecnologías en cuestión de robótica.

Algunos países, como Dinamarca, ven a México como un lugar atractivo para sus productos, algunos de ellos buscando expandirse y buscando la apertura de sedes en nuestro país. México es uno de los países que más robots importa a nivel mundial, de acuerdo a la empresa ABB, pero hay que tener en cuenta que el avance que se logre en nuestro país va a estar reflejado en la cantidad de inversión que se tenga en este rubro, el cual es menos del 1 % del PIB de acuerdo a cifras de la UNESCO [5].

Los robots neumáticos, por otra parte, no tienen un crecimiento tan notorio como el caso de los robots con actuadores eléctricos. Esto se debe a que la alta compresibilidad de un fluido para transferir potencia y la presencia de fricción dificultan en gran manera la obtención de un modelo y un control preciso. Usualmente, los actuadores neumáticos son utilizados para la manipulación de cargas entre dos posiciones mediante el uso de válvulas on-off, ya que las no linealidades hacen muy difícil su control entre posiciones intermedias [6][7].

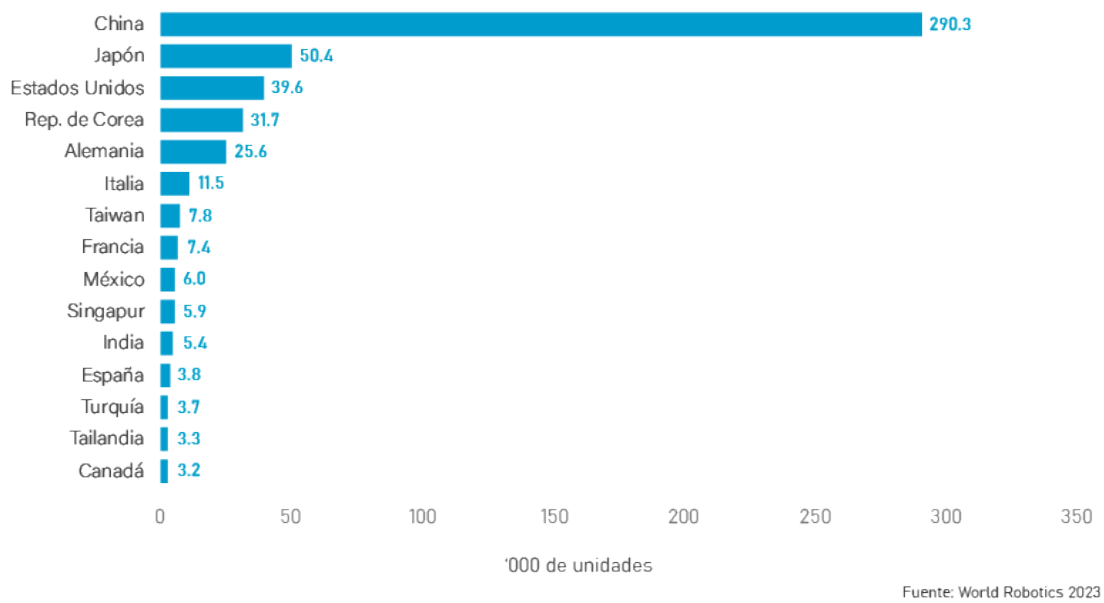


Figura 1.2: Robots instalados en 2022 [1].

1.2. Justificación

El mundo actual se está enfocando en energías limpias y mejorar el rendimiento de robots ya creados. Una opción es el uso de actuadores neumáticos, ya que estos dan un mayor rendimiento en cuestiones económicas, por la reducción de mantenimientos programados, y un mayor desempeño en cuestión potencia-peso. El problema es que, para aplicar un control en un sistema neumático, este se complica al ser un sistema no lineal en comparación con un sistema lineal como lo es el control de un sistema eléctrico [8].

Los sistemas neumáticos tienen la ventaja de ser durables, fáciles de mantener y seguros. Por otra parte, se tiene que, debido a sus no linealidades, como lo son la compresibilidad del aire, la fricción entre los componentes de los actuadores y el flujo discontinuo de las válvulas, provocan que su modelado dinámico sea complicado y que usualmente se base en suposiciones empíricas. Los llamados algoritmos inteligentes como lo son las redes neuronales y el control difuso, son atractivos, ya que no se requiere un modelo del sistema [9].

Los robots neumáticos tienen grandes ventajas, como la simplicidad en mecanismos y altas velocidades, ventajas sobre los actuadores hidráulicos, como la limpieza y el costo-efectividad, y mayor fuerza por un período prolongado de tiempo comparado con los eléctricos, los cuales están expuestos a riesgos de sobrecalentamiento. El uso de robots en la industria es cada vez mayor, por lo que, al ver el crecimiento de instalaciones de robots en el país, se busca aportar con un algoritmo de inteligencia artificial para el control de un robot, así como con un robot más amigable con el medio ambiente al usar actuadores neumáticos [10].

1.3. Hipótesis y Objetivos

En esta sección se presenta la hipótesis, objetivo general y los objetivos específicos del proyecto.

1.3.1. Hipótesis

Una metodología con base en algoritmos de inteligencia artificial implementados en la regulación del flujo de aire hacia los actuadores neumáticos controla y reduce el error de posición del actuador final en un robot neumático tipo SCARA hasta un 10 % comparado con técnicas convencionales reportadas a la fecha en el estado del arte.

1.3.2. Objetivo general

Diseñar e implementar un control inteligente de posición para un robot SCARA neumático de 4 grados de libertad.

1.3.3. Objetivos particulares

- Calcular la cinemática del robot neumático para obtener el modelo matemático mediante el método de Denavit-Hartenberg.
- Comprobar el hardware del robot para su correcto funcionamiento mediante experimentación e inspecciones visuales.
- Implementar las adecuaciones al robot neumático para mejorar su rendimiento.
- Seleccionar e implementar algoritmo de IA a utilizar para el posicionamiento de los actuadores mediante pruebas en simulación.

1.4. Antecedentes

Existen varios trabajos relacionados con el control de posición de actuadores neumáticos. En uno de ellos se realiza la comparativa entre el controlador Proporcional Integral Derivativo y un controlador Regulador Cuadrático Lineal aplicado a un músculo neumático artificial, encontrando que el RCL tiene un error en estado estacionario menor [11].

Existen comparativas entre controladores PID difusos y PIDOF, en las que se determinó que el PIDOF tiene una respuesta más rápida y con poco error en estado estacionario, esto usando un Ball and Beam como sistema [9].

Por otra parte, se presenta un robot en el que a un robot con garra se le adapta un Control de Rendimiento Prescrito de Tasa de Convergencia Ajustable a un control PIDOF, obteniendo una respuesta aún más rápida y con menor sobrepaso que solo con el control PIDOF [2].

Otra investigación en Ball and Beam con control PIDOF, a esta se suma un modelo paramétrico para determinar la función de transferencia. Se utilizó un sensor para monitorear la presión de la recámara del actuador. Esta investigación también cuenta con un modelo de control en cascada, colocando un controlador para el actuador y otro para la posición de la esfera del Ball and Beam [3].

Algunos trabajos se centran en optimizar el control difuso. Uno de ellos usa una función de activación analítica para seleccionar las reglas que rigen el control difuso; esta función determina el grado de pertenencia de las variables de entrada y la defuzzificación se realiza mediante el promedio de cada salida [12].

Por último, se tiene un estudio en el que se utilizan 2 controladores difusos, uno para el control de posición y otro para el control de la fuerza del actuador. Se hace una comparativa usando solo los dos controladores y estos mismos agregando un controlador de superficie deslizante [10].

Por otra parte, se tienen los estudios que indican numéricamente el error obtenido. En 2014 se realizó una simulación del movimiento de un brazo robótico de tipo SCARA de un grado de libertad. Este se controlaba mediante una red neuronal artificial de perceptrón multicapa, mejorando la respuesta en comparación a un control PID en 2.95 s y minimizando el error angular por 1.3° [13].

En 2016 se realizó un modelado de una plataforma de dos grados de libertad controlada por tres actuadores neumáticos. Para esto se utilizó un controlador PID en donde se observó un error de 0.2° en los puntos pico de la trayectoria senoidal [14].

En 2017 se realizó el control de posición de un manipulador serial de seis ejes usando un controlador PID difuso. Con esto se logró un error en estado estacionario de 0.4 mm [15].

Tabla 1.1: Estado del Arte

Año	Artículo	Descripción
2018	Pneumatic Rotary Actuator Position Servo System Based on ADE-PD Control [16].	Se compara el control con DE con NCD-PID y se prueba aumentando la carga inercial.

Continuación: Estado del Arte		
Año	Artículo	Descripción
2019	Development and control of a pneumatic-actuator 3-DOF translational parallel manipulator with robot vision [17].	Desarrollo y control de un manipulador paralelo de 3 grados de libertad accionado neumáticamente.
2019	Intelligent position control for pneumatic servo system based on predictive fuzzy control [18].	Control inteligente de posición para un servo neumático basado en control predictivo difuso.
2020	Positioning Control of an Antagonistic Pneumatic Muscle Actuated System using Feedforward Compensation with Cascaded Control Scheme [19].	Se presenta un control FFC para el control de posición de un PMA vertical.
2021	Self-tuning hybrid fuzzy sliding surface control for pneumatic servo system positioning [10].	Desarrollar una estrategia de control para un servo sistema neumático uniendo dos controladores lógicos difusos.
2021	A pneumatic random-access memory for controlling soft robots [20].	Control de "soft robot" mediante circuitos lógicos neumáticos.
2022	Soft pneumatic actuators: A review of design, fabrication, modeling, sensing, control and applications [21].	Revisión de actuadores neumáticos, abarcando diseño, fabricación, modelado y aplicaciones.
2022	Pneumatic soft robots: Challenges and benefits [22].	Exploración de desafíos y beneficios de "soft robots" neumáticos.
2022	Implementation of ANN-Based Auto-Adjustable for a Pneumatic Servo System Embedded on FPGA [23].	Implementación de redes neuronales para sistemas servo neumáticos.
2022	Position Control of a Pneumatic Drive Using a Fuzzy Controller with an Analytic Activation Function [12].	Controlar el posicionamiento de un sistema neumático mediante un controlador difuso usando una función de activación analítica.
2023	Soft Robots: Implementation, Modeling, and Methods of control [24].	Implementación, modelado y métodos de control de "soft robots".
2023	Recent Developments of Actuation Mechanisms for Continuum Robots: A Review [25].	Resumen de los desarrollos recientes en mecanismos de actuación para robots.
2023	Research on a Variable Pressure Driving Method for Soft Robots Based on the Electromagnetic Effect [26].	Investigación sobre un método de conducción de presión variable para "soft robots".
2023	Design, Topology Optimization, and Additive Manufacturing of a Pneumatically Actuated Lightweight Robot [27].	Diseño y optimización de robots ligeros accionados neumáticamente.
2023	Position Control of a Cost-Effective Bellow Pneumatic Actuator Using an LQR Approach [11].	Analizar la precisión de posicionamiento de un PMA utilizando un controlador PID y un LQR.

Continuación: Estado del Arte		
Año	Artículo	Descripción
2023	Adjustable Convergence Rate Prescribed Performance with Fractional-Order PID Controller for Servo Pneumatic Actuated Robot Positioning [2].	Método para rastreo óptimo de error de control de posición para un actuador servo neumático.
2023	Fuzzy Fractional Order PID Tuned via PSO for a Pneumatic Actuator with Ball Beam (PABB) System [9].	Mejorar el rendimiento del posicionamiento de un sistema neumático mediante un controlador FOPID.
2024	Design, Manufacturing, and Open-Loop Control of a Soft Pneumatic Arm [28].	Diseño, fabricación y control de un robot neumático.

1.5. Estructura de la tesis

La estructura de esta tesis es la siguiente: en el capítulo 1 se presenta la introducción donde se habla a grandes rasgos de los actuadores neumáticos y como se han utilizado los algoritmos de inteligencia artificial para atacar los problemas relacionados con las no linealidades que se presenta al usar este tipo de actuadores. También se presenta la problemática a atacar, la justificación, la hipótesis, objetivos y algunos antecedentes de proyectos similares. En el capítulo 2 se muestra el marco teórico utilizado, como lo es la cinemática del robot, qué es un sistema lineal y no lineal y, finalmente, que es un controlador PID, PID difuso y una red neuronal de retropropagación. En el capítulo 3 se menciona la metodología utilizada en esta investigación. En el capítulo 4 los resultados obtenidos para cada tipo de controlador utilizados, la comparativa entre estos y una discusión de los resultados obtenidos y, finalmente, en el capítulo 5 se presentan las conclusiones obtenidas de este proyecto.

Marco Teórico

En este capítulo se mencionará la teoría relevante para esta investigación. Se hace una mención sobre la cinemática directa e indirecta para un robot tipo SCARA y como se calcula. También se tiene una breve descripción de lo que son los sistemas lineales y no lineales y, por último, una descripción de los controladores utilizados en este proyecto.

2.1. Cinemática del robot

La cinemática es la ciencia del movimiento que estudia el movimiento sin las fuerzas externas que lo causan. Un problema básico en el estudio de manipulación mecánica es llamado cinemática directa. Esto es el problema geométrico estático de calcular la posición y orientación del efector final de un manipulador, específicamente dados los ángulos en las uniones. En cambio, la cinemática inversa, dada la posición y orientación del efector final, calcula todas las posibles combinaciones de ángulos que se necesitan para llegar a esa posición y orientación [29].

El método de Denavit-Hartenberg (D-H) para la obtención de la cinemática directa de posición propone el movimiento articulado de un robot con n articulaciones que puede ser expresado mediante los siguientes parámetros [30]:

- θ_i es el ángulo formado entre el eje x_i y el eje x_{i+1} a lo largo del eje z_i .
- d_i es la distancia entre el eje x_i y el eje x_{i+1} a lo largo del eje z_i .
- α_i es el ángulo formado entre el eje z_i y el eje z_{i+1} a lo largo del eje x_{i+1} .
- a_i es la distancia entre el eje z_i y el eje z_{i+1} a lo largo del eje x_{i+1} .

La matriz de transformación homogénea definida por T_i , representa el movimiento articulado de i a la articulación de $i + 1$ en la ecuación.

$$T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

La matriz de posición de D-H está definida por T en la ecuación

$$T_R = T_i * T_{i+1} * \dots * T_{n-1} * T_n \quad (2.2)$$

La Figura 2.1, presenta la representación asociada de la convención D-H. La matriz de posición obtenida mediante la convención D-H está dada por la ecuación:

$$T_i = \begin{bmatrix} a_x & b_x & c_x & p_x \\ a_y & b_y & c_y & p_y \\ a_z & b_z & c_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Donde

$$a_x = -\cos(q_1 + q_2 + q_3)$$

$$a_y = -\sen(q_1 + q_2 + q_3)$$

$$a_z = 0$$

$$b_x = \sen(q_1 + q_2 + q_3)$$

$$b_y = -\cos(q_1 + q_2 + q_3)$$

$$b_z = 0$$

$$c_x = 0$$

$$c_y = 0$$

$$c_z = -1$$

En las ecuaciones (4) a (6) se expresa la posición del robot.

$$p_x = l_2 \cos(q_1) + l_3 \cos(q_1 + q_2) + l_4 \cos(q_1 + q_2 + q_3) \quad (2.4)$$

$$p_y = l_2 \sen(q_1) + l_3 \sen(q_1 + q_2) + l_4 \sen(q_1 + q_2 + q_3) \quad (2.5)$$

$$p_z = l_1 - q_4 \quad (2.6)$$

Las coordenadas cartesianas p_x , p_y y p_z representan la posición del efector final. Donde el primer grado de libertad q_1 expresa el movimiento rotacional del plano generado por las variables articulares q_1 , q_2 y q_3 . Del plano generado por la Figura 2.2 consiste en proponer un ángulo q_P para la orientación del último eslabón. En las ecuaciones (7) a (17), así como en la Tabla 2.1, se presentan las variables necesarias para la solución de la cinemática inversa mediante el método geométrico.

$$x_2 = P_x - l_3 \cos(\phi) \quad (2.7)$$

$$y_2 = P_y - l_3 \sen(\phi) \quad (2.8)$$

$$\cos(q_2) = \frac{x_2^2 + y_2^2 - l_2^2 - l_3^2}{2l_2l_3} \quad (2.9)$$

$$x_2 = l_2 \cos(q_1) + l_3 \cos(q_1 + q_2) \quad (2.10)$$

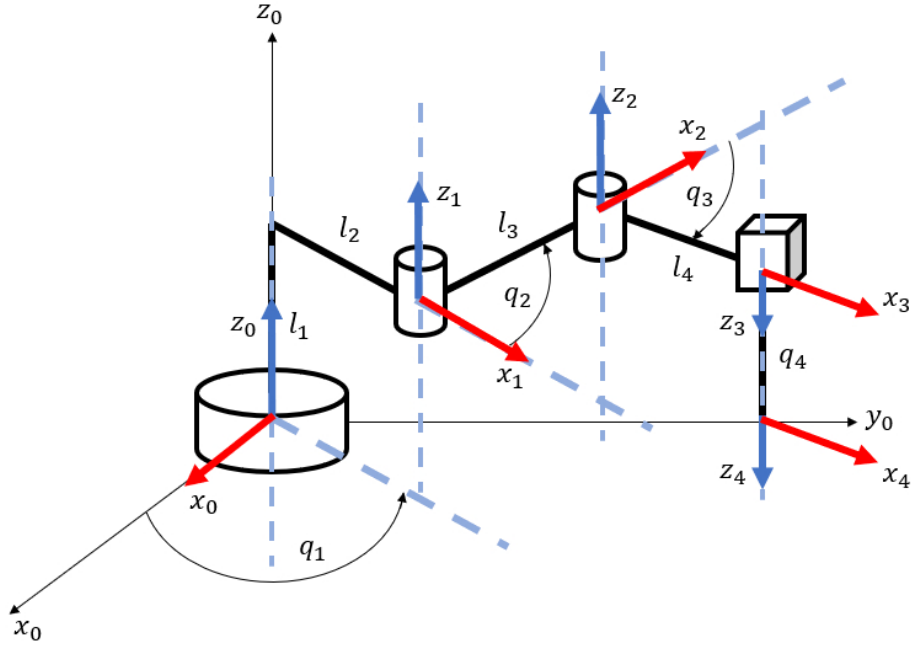


Figura 2.1: Robot neumático.

$$y_2 = l_2 \sin(q_1) + l_3 \sin(q_1 + q_2) \quad (2.11)$$

$$x_2 = \cos(q_1)(l_2 + l_3 \cos(q_2)) - \sin(q_1)(l_2 \sin(q_2)) \quad (2.12)$$

$$y_2 = \cos(q_1)(l_2 \sin(q_2)) + \sin(q_1)(l_2 + l_3 \cos(q_2)) \quad (2.13)$$

$$\cos(q_1) = \frac{(l_2 + l_3 \cos(q_2))x_2 + l_2 \sin(q_2)y_2}{x_2^2 + y_2^2} \quad (2.14)$$

$$\sin(q_1) = \frac{(l_2 + l_3 \cos(q_2))y_2 + l_2 \sin(q_2)x_2}{x_2^2 + y_2^2} \quad (2.15)$$

$$q_1 = \tan^{-1} \left(\frac{\sin(q_1)}{\cos(q_1)} \right) \quad (2.16)$$

$$q_3 = \phi - (q_1 + q_2) \quad (2.17)$$

Tabla 2.1: Variables usadas en Figura 2.1 y 2.2.

Variable	Descripción
l_1, l_2, l_3 y l_4	Distancia entre articulaciones
q_1, q_2 y q_3	Ángulos variables entre articulaciones
q_4	Distancia variable entre articulaciones

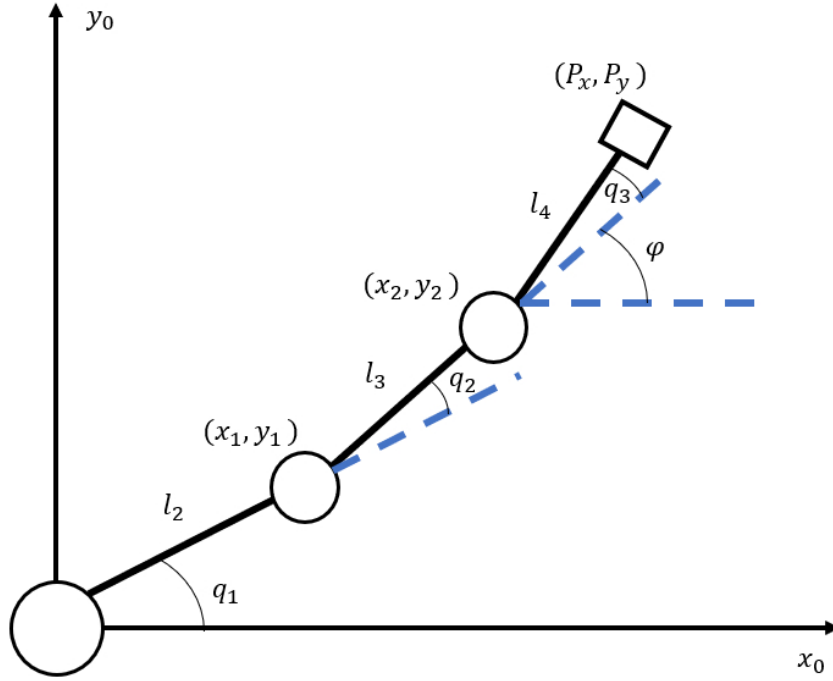


Figura 2.2: Proyección geométrica.

2.2. Sistemas lineales

El comportamiento dinámico de los sistemas se puede estudiar a partir de su descripción matemática. Las ecuaciones matemáticas se usan para describir el comportamiento de procesos y para predecir la respuesta de estos ante ciertas entradas; de esta manera es posible analizar el comportamiento de sistemas complejos y diseñar procesos para tener un comportamiento deseado. El estudio de un fenómeno usualmente inicia con el proceso de modelado, el cual requiere el uso de leyes o principios para generar las ecuaciones que describen al modelo [31].

Un sistema L es lineal cuando una entrada u_1 lleva a una salida $L(u_1)$ y una entrada u_2 a una salida $L(u_2)$ y se tiene también

$$L(c_1 u_1 + c_2 u_2) = c_1 L(u_1) + c_2 L(u_2) \quad (2.18)$$

donde c_1 y c_2 son números reales arbitrarios. Esto significa implícitamente que el espacio de las posibles entradas es cerrado bajo combinación lineal; es decir, $c_1 u_1 + c_2 u_2$ debe pertenecer al espacio si u_1 y u_2 lo hacen. O en otras palabras, un sistema se dice que es lineal cuando satisface el principio de superposición [32][33].

Un sistema lineal entonces, es aquel que cumple con dos propiedades; la primera es la propiedad de superposición y la segunda es la propiedad de homogeneidad. El principio de superposición nos dice que la respuesta producida por la aplicación simultánea de dos funciones de entrada diferentes

es la suma de las dos respuestas individuales. El principio de homogeneidad establece que si la entrada de un sistema lineal se escala por un factor a , entonces la salida también se escala por el mismo factor a [34][35].

2.3. Sistemas no lineales

Los sistemas lineales generalmente son resultado de la linealización de sistemas no lineales, o el resultado de modelar el proceso del sistema físico en el que se eliminan o se ignoran los efectos no lineales [31].

En el sentido matemático, un sistema no lineal se refiere a un grupo de ecuaciones no lineales, ya sea algebraicas, diferenciales, integrales, funcionales, y operadores, o una combinación de estos. Un sistema no lineal es usado para describir un proceso o dispositivo físico que de otra manera no podría ser bien definido con el uso de ecuaciones lineales, aunque un sistema lineal es considerado un caso especial de sistema no lineal. Es decir, un sistema no lineal no cumple las propiedades de superposición y de homogeneidad; la relación entre la entrada y la salida no sigue una forma lineal. Los sistemas no lineales pueden presentar comportamiento más complejo como caos o bifurcaciones, y son mucho más difíciles de analizar que los sistemas lineales. Un sistema se considera no lineal si las ecuaciones que lo describen son no lineales, por ejemplo, ecuaciones que involucran productos de variables o potencias, o si la salida no es proporcional a la entrada [36][37].

2.4. Controlador PID

Un controlador PID es un tipo de sistema de control automático que utiliza tres componentes fundamentales para ajustar la salida de un sistema y minimizar el error entre el valor deseado y el valor real de la variable controlada. Estas componentes son la proporcional, la cual ajusta la salida en función del error actual. La magnitud del ajuste es directamente proporcional al error y su objetivo es reducir el error de manera rápida.

$$P = K_p \cdot e(t) \quad (2.19)$$

Donde K_p es la ganancia proporcional y $e(t)$ es el error en el tiempo t .

El término integral, el cual ajusta la salida en función de la acumulación del error a lo largo del tiempo, asegura que el sistema llegue al valor deseado a largo plazo.

$$I = K_i \int e(t)dt \quad (2.20)$$

Donde K_i es la ganancia integral y $\int e(t)dt$ es la integral del error.

Finalmente, la componente derivativa que ajusta la salida en función de la tasa de cambio del error. Su objetivo es anticipar el comportamiento del error y disminuir la oscilación o el sobrepaso del sistema.

$$D = K_d \frac{de(t)}{dt} \quad (2.21)$$

Donde K_d es la ganancia derivativa y $\frac{de(t)}{dt}$ es la derivada del error.

La salida total del controlador PID es la suma de estos tres términos.

$$u(t) = K_p \cdot e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \quad (2.22)$$

Donde $u(t)$ es la señal de control que se envía a la planta [34][35].

Considérese que el controlador PID consiste en una parte PI conectada en cascada con una parte PD. La función de transferencia del controlador PID se escribe como:

$$G_c(s) = K_P + K_D s + \frac{K_I}{s} = (1 + K_{D1}s) \left(K_{P2} + \frac{K_{I2}}{s} \right) \quad (2.23)$$

La constante proporcional de la parte PD se hace unitaria, ya que sólo se necesitan tres parámetros en el controlador PID. Al igualar ambos miembros de la ecuación 2.23, se tiene:

$$K_P = K_{P2} + K_{D1}K_{I2}$$

$$K_D = K_{D1}K_{P2}$$

$$K_I = K_{I2}$$

Considere que sólo la parte PD está operando. Seleccione el valor de K_{D1} para lograr una parte de estabilidad relativa deseada. En el dominio del tiempo, esta estabilidad relativa se puede medir mediante el sobrepaso máximo, y en el dominio de la frecuencia con el margen de fase.

Seleccione los parámetros K_{I2} y K_{P2} para que el requisito de la estabilidad relativa sea satisfecho.

Como una opción, la porción PI del controlador se puede diseñar primero para una parte del requisito sobre la estabilidad relativa y, finalmente, se diseña la parte PD [38].

2.5. Control PID Difuso

De acuerdo con [39] el control difuso usa sentencias en forma de reglas para controlar un proceso. El controlador puede tomar muchas entradas, y la ventaja del control difuso es la habilidad de incluir conocimiento experto. La interfaz con el controlador es más o menos lenguaje natural, y eso es lo que distingue al control difuso de otros métodos de control. Generalmente, es un control no lineal.

Convencionalmente, los programas de computadora toman decisiones rígidas de sí o no por medio de reglas de decisión basadas en dos valores lógicos: verdadero/falso, sí/no, o uno/cero. Un ejemplo es un aire acondicionado con un controlador termostático que reconoce solo dos estados: encima de la temperatura deseada o por debajo de la temperatura deseada. La lógica difusa, por otra parte, permite valores de verdad intermedios entre verdadero y falso [39].

Si el control PID es inadecuado, el control difuso es una opción. Hay varias razones posibles:

- Ya que la estrategia de control consiste en reglas si-entonces, es fácil de leer para un operador de procesos. Las reglas pueden construirse con un vocabulario que contenga palabras del día a día tales como ‘alto’, ‘bajo’, y ‘aumentando’. Los operadores pueden incluir su experiencia directamente.
- El controlador difuso puede contar con varias entradas y varias salidas. Las variables pueden ser combinadas en una regla si-entonces con los conectores and y or. Las reglas son ejecutadas en paralelo, implicando una acción recomendada para cada una. Las recomendaciones pueden entrar en conflicto, pero el controlador resuelve conflictos.

Establecer métodos de diseño tales como colocación de polos, control óptimo, y respuesta de frecuencia solo aplica para sistemas lineales, mientras que el control difuso generalmente es no lineal. Ya que nuestro conocimiento del comportamiento de sistemas no lineales es limitado, comparado con una situación en el dominio lineal, el siguiente procedimiento está basado en control lineal [39].

1. Diseñar un controlador PID.
2. Reemplazarlo con un controlador difuso lineal.
3. Hacerlo no lineal.
4. Hacer ajustes.

2.6. Red Neuronal de retropropagación

Una red neuronal de retropropagación es un tipo de red neuronal artificial que se entrena ajustando los pesos de la red para minimizar el error entre la salida deseada y la salida que la red predice. Este ajuste se realiza mediante un algoritmo llamado retropropagación del error, el cual distribuye el error hacia atrás a través de la red para actualizar los pesos y mejorar el rendimiento del modelo.

El proceso comienza cuando las entradas x_i son alimentadas a la capa de entrada de la red. Cada neurona en las capas ocultas y de salida procesa la entrada de acuerdo a una función de activación f y produce una salida.

$$z_j^{(l)} = \sum_i w_{ij}^{(l)} x_i^{(l-1)} + b_j^{(l)} \quad (2.24)$$

Donde:

- $w_{ij}^{(l)}$ es el peso entre la neurona i de la capa anterior y la neurona j de la capa actual.
- $x_i^{(l-1)}$ es la salida de la neurona i de la capa anterior.
- $b_j^{(l)}$ es el sesgo de la neurona j .

El cálculo de la salida de la neurona j está dado por

$$a_j^{(l)} = f(z_j^{(l)}) \quad (2.25)$$

Donde $f(z_j^{(l)})$ es la función de activación que puede ser, por ejemplo, lineal, sigmoide o ReLU.

Una vez que la red ha producido una salida, se calcula el error comparando la salida obtenida con la salida esperada. Este error se utiliza para ajustar los pesos de la red.

$$\delta_j^{(L)} = (a_j^{(L)} - y_j) f'(z_j^{(L)}) \quad (2.26)$$

Donde:

- $\delta_j^{(L)}$ es el error de la neurona j en la capa de salida.
- y_j es el valor esperado.
- $f'(z_j^{(L)})$ es la derivada de la función de activación.

El error en las capas ocultas se calcula utilizando la propagación del error desde la capa siguiente:

$$\delta_j^{(l)} = \sum_k \delta_k^{(l+1)} w_{kj}^{(l+1)} f'(z_j^{(l)}) \quad (2.27)$$

Donde:

- $\delta_k^{(l+1)}$ es el error de la capa siguiente.
- $w_{kj}^{(l+1)}$ es el peso entre la neurona k de la capa siguiente y la neurona j de la capa actual.
- $f'(z_j^{(l)})$ es la derivada de la función de activación en la capa l .

Una vez calculado el error, se ajustan los pesos utilizando el algoritmo de gradiente descendente. Los pesos se actualizan en la dirección que minimiza el error de la red, es decir, en la dirección del gradiente negativo.

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \delta_j^{(l)} a_i^{(l-1)} \quad (2.28)$$

Donde:

- η es la tasa de aprendizaje.
- $\delta_j^{(l)}$ es el error de la neurona j en la capa l .
- $a_i^{(l-1)}$ es la salida de la neurona i en la capa anterior.

La actualización del sesgo se realiza de manera similar.

$$b_j^{(l)} \leftarrow b_j^{(l)} - \eta \delta_j^{(l)} \quad (2.29)$$

Este proceso se repite muchas veces hasta que el error sea lo suficientemente pequeño o el entrenamiento se detenga [40][41].

Metodología

En este capítulo se mencionará la metodología utilizada para la elaboración de este proyecto. Se analizará como es que se obtuvo la cinemática directa del robot, los componentes utilizados para la fabricación y las adecuaciones realizadas, el hardware eléctrico utilizado y el diseño de los controladores utilizados.

En la Figura 3.1 se muestra el diagrama de bloques de la metodología usada para el desarrollo de este trabajo. A continuación se hará una explicación breve de cada bloque dentro del diagrama.

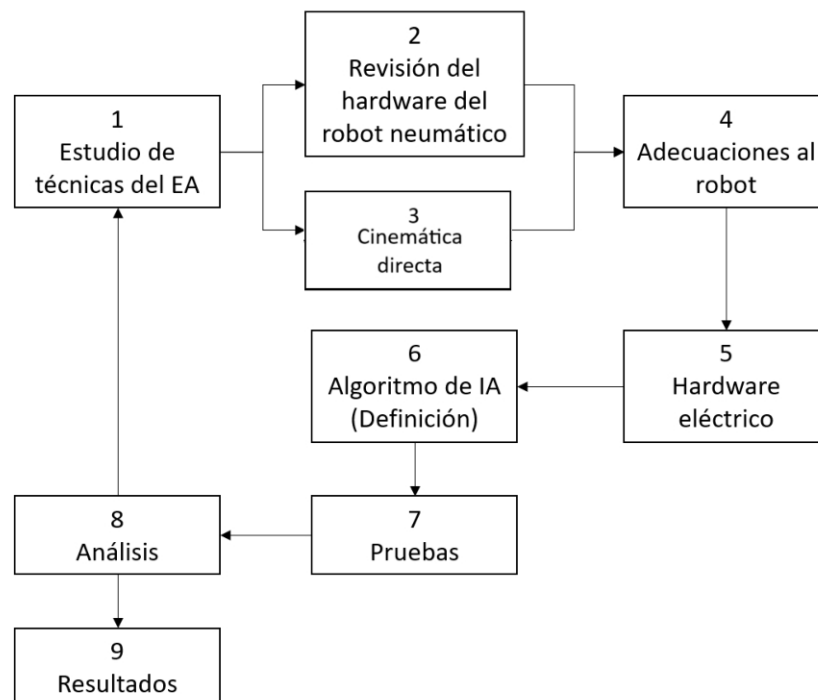


Figura 3.1: Diagrama de la metodología.

1. El primer paso consta de una búsqueda de artículos referentes a la investigación. Se tomarán los artículos más recientes como el estado del arte mientras que los artículos con más tiempo serán tomados como antecedentes.
2. El siguiente paso es la revisión del hardware del robot neumático. En este paso se analizará el rango de trabajo de los motores, es decir, el ángulo mínimo y máximo que los motores pueden alcanzar.
3. Para el siguiente paso, que se realizará en paralelo junto con el paso anterior, se analizará la cinemática del robot lo que implica la cinemática directa e inversa del robot neumático.
4. Para el cuarto paso se realizarán las adecuaciones necesarias al robot neumático, se añadirá el sensor de posición para poder determinar la posición actual de los eslabones.
5. El quinto paso consistirá en la revisión del hardware eléctrico como lo es la etapa de potencia que controlará las electroválvulas y el ADC que tomará los datos el sensor de posición. También se decidirá el sistema en el que se realizará la etapa de control, es decir, se realizarán pruebas para determinar el tipo de sistema embebido para la implementación del algoritmo a desarrollar. Se tiene contemplado el uso de un microcontrolador, un DSP o algún otro sistema embebido.
6. Para el paso 6 se analizarán distintos métodos de control para compararlos y elegir el de mejor desempeño. Se analizará el controlador difuso y algunas técnicas con redes neuronales para el controlador.
7. En el paso 7 se realizarán varias pruebas de los distintos controladores en el robot neumático para obtener datos para su análisis.
8. En este paso se analizarán los datos obtenidos en el paso anterior para ver con cual controlador el robot tiene un mejor desempeño, en caso de no tener datos satisfactorios se regresará al paso uno para plantear otra estrategia para el controlador.
9. Finalmente, en caso de que los análisis sean satisfactorios, se procederá a escribir los resultados obtenidos.

3.1. Estudio de técnicas del Estado del Arte

Revisando la literatura, se buscaron técnicas en las que no fuera necesario un modelo matemático. Se decidió tener un controlador PID para comparar los resultados de las técnicas con algoritmos de inteligencia artificial. Se optó por utilizar un controlador PID difuso y un controlador con una red neuronal de retropropagación; en ambos casos se puede realizar el control sin la necesidad de contar con un modelo matemático y son controladores utilizados ampliamente para casos en los que la dinámica del sistema es no lineal.

3.2. Revisión del hardware del robot neumático

Los actuadores utilizados, así como los materiales, se encuentran en la Tabla 3.1.

Tabla 3.1: Componentes y materiales.

Cantidad	Componente	Modelo	Parámetros
1	Motor Neumático	CRB1BW100-270S	Rango de movimiento 270°
2	Motor Neumático	CRB2BW15-270S	Rango de movimiento 270°
2	Tubo PVC 3/4"	-	0.75m de longitud
2	Tubo PVC 3/4"	-	0.50m de longitud
2	Tubo PVC 3/4"	-	0.25m de longitud
1	Válvula de estrangulación	GS04402	0 - 1 MPa
2	Rodamiento		

En la Figura 3.2 se muestra una foto del robot neumático con sus 4 grados de libertad.



Figura 3.2: Robot neumático.

3.3. Cinemática directa del robot neumático

Dado que el microcontrolador está programado para enviar la posición de los actuadores en grados, se utilizará la cinemática inversa para determinar la posición del actuador final en sus coordenadas x, y y z , las cuales están dadas por las ecuaciones (2.4), (2.5) y (2.6). Sustituyendo los valores de l_1, l_2, l_3 y l_4 por las dimensiones de los eslabones, de las cuales l_2, l_3 y l_4 están dadas por las dimensiones de los tubos de PVC de la Tabla 3.1. Teniendo un valor de l_1 de 0.6m, tenemos que la cinemática directa del robot está dada por las ecuaciones siguientes:

$$p_x = 0.75\cos(q_1) + 0.5\cos(q_1 + q_2) + 0.25\cos(q_1 + q_2 + q_3) \quad (3.1)$$

$$p_y = 0.75\sin(q_1) + 0.5\sin(q_1 + q_2) + 0.25\sin(q_1 + q_2 + q_3) \quad (3.2)$$

$$p_z = 0.6 - q_4 \quad (3.3)$$

Los valores obtenidos del microcontrolador serán después sustituidos en los valores de q_1 , q_2 y q_3 . Para q_4 , el valor será constante, ya que el efector final solo tiene dos posibles posiciones en el eje z , de las cuales en la posición final siempre será con el actuador final extendido.

3.4. Adecuaciones al robot

Se sustituyeron los encoders por potenciómetros para determinar la posición del motor neumático. Para esto se diseñaron piezas para acoplar los potenciómetros a los actuadores como la mostrada en la Figura 3.3.

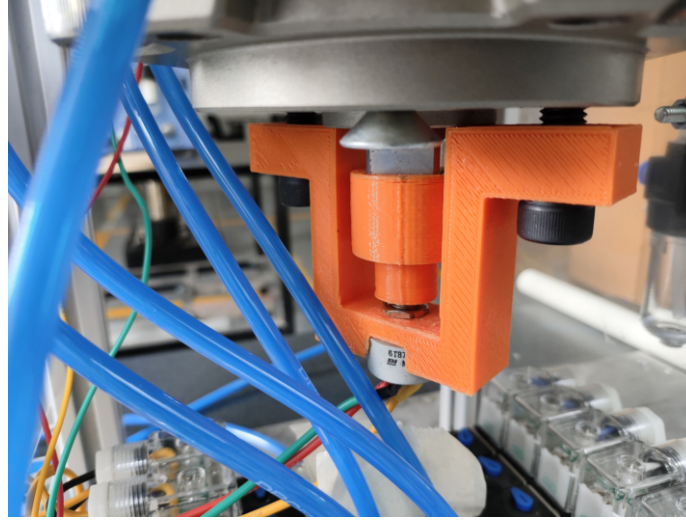


Figura 3.3: Pieza impresa para acople de potenciómetros.

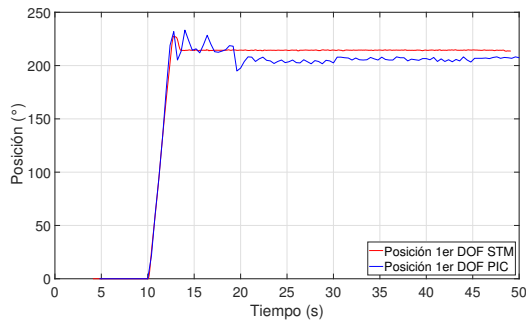
Se diseñaron piezas en impresión 3D para soportar tanto los actuadores como los tubos de PVC que forman los eslabones, mostrados en la Figura 3.4. A estos últimos se les agregó un contrapeso para disminuir el efecto de la fricción en los rodamientos. La posición de los eslabones se fue modificando a lo largo de la experimentación para tratar de equilibrar el peso que generaban los elementos unidos al eslabón con el contrapeso, para tener el menor efecto de fricción posible en los rodamientos. En la experimentación se usaron dos microcontroladores distintos para determinar cuál de ellos tenía un mejor rendimiento.

Inicialmente se utilizó un microcontrolador PIC18F4550 a una velocidad de 8 MHz con un ADC de 10 bits con un tiempo de muestreo de 100 ms al usar un grado de libertad y de 400 ms al controlar dos grados de libertad. Después se realizaron pruebas con un STM32F303RE a una velocidad de 72 MHz con un ADC de 12 bits y un tiempo de muestreo de 50 ms con los 3 grados de libertad. La razón del cambio de 50 a 400 ms en el PIC18F4550 fue debido a que al disminuir la velocidad de muestreo se comenzaron a tener problemas en el control del motor a pasos, haciendo que su movimiento fuera más lento y con pasos irregulares.

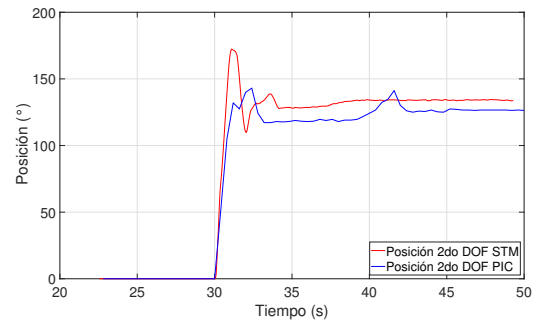


Figura 3.4: Pieza impresa para sujeción de actuadores y eslabones.

En la Figura 3.5 se pueden observar las gráficas comparativas entre el desempeño del microcontrolador PIC18F4550 y el STM32F303RE. Se observa que las gráficas son más suaves en el STM32F303RE por la diferencia en el tiempo de muestreo y también hay una disminución en el error de posición y un menor tiempo de asentamiento. Las pruebas solo se realizaron para dos grados de libertad y con esto se decidió usar el STM32F303RE como microcontrolador de este proyecto.



(a) 1er Grado de libertad.



(b) 2do Grado de libertad.

Figura 3.5: Comparativa entre microcontroladores.

3.5. Hardware eléctrico

El hardware eléctrico utilizado para el robot se muestra en la Tabla 3.2.

Tabla 3.2: Componentes eléctricos.

Cantidad	Componente	Modelo	Parámetros
1	Tarjeta NUCLEO	NUCLEO-F303RE	STM32F303RE
2	Potenciómetros	1V 15KOH	15 k Ω
1	Potenciómetro	6I87R10K	10 k Ω
8	Optoacopladores	4N32	60 mA
8	Transistores	TIP31C	100 V 3 A
1	Puente H	DBH-12V	Doble canal 30 A
8	Electroválvula	4V230C-08	0.15 - 0.8 MPa

En la Figura ?? se pueden observar las electroválvulas utilizadas en este proyecto.

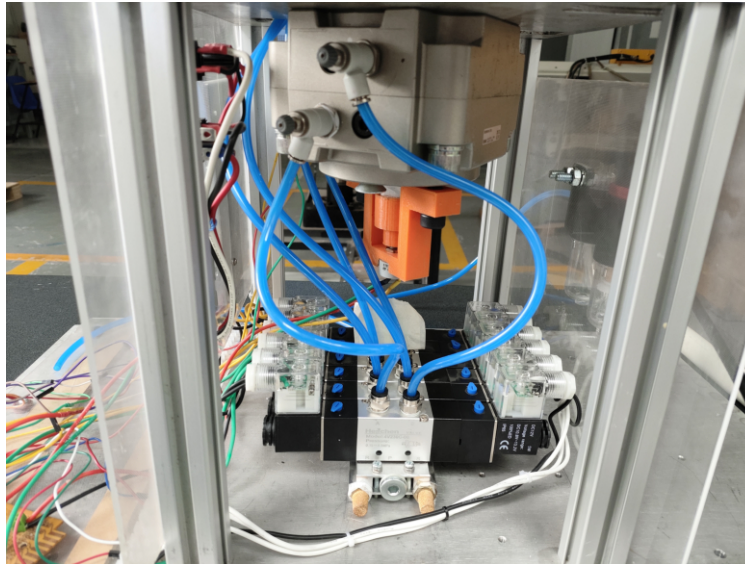


Figura 3.6: Electroválvulas utilizadas.

3.6. Algoritmo de Inteligencia Artificial (IA)

Después de los pasos anteriores, se llegó al diagrama electroneumático de la Figura 3.7. En esta se observan los actuadores utilizados, los sensores, la etapa de potencia, el sistema de control de flujo y el microcontrolador usado para todo el proceso, así como las conexiones eléctricas, mostradas con una línea punteada, y las conexiones neumáticas con una línea sólida.

Es importante mencionar que el control de flujo de aire se está realizando mediante una señal de control enviada por el microcontrolador. Esta señal de control consta de los pasos que tiene que dar el motor a pasos para abrir o cerrar una válvula de estrangulamiento, la cual dará la apertura de aire ideal para el movimiento de los actuadores.

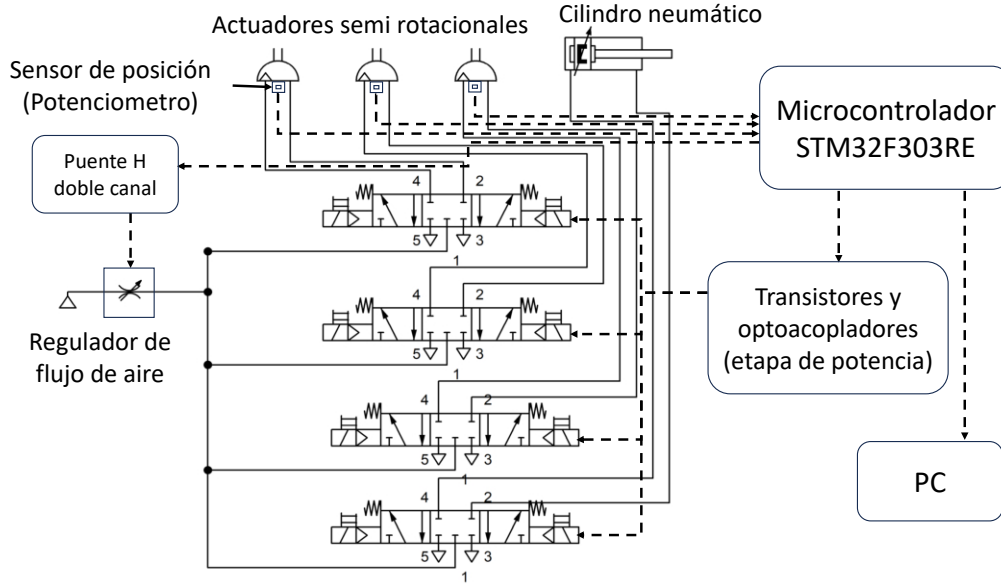


Figura 3.7: Diagrama electroneumático del robot.

3.6.1. Controlador PID

En la Figura 3.8 se muestra el diagrama de bloques del controlador PID. Este no es considerado como un controlador con AIA ya que sus ganancias son fijas, por lo que no hay una adaptación al comportamiento de la planta.

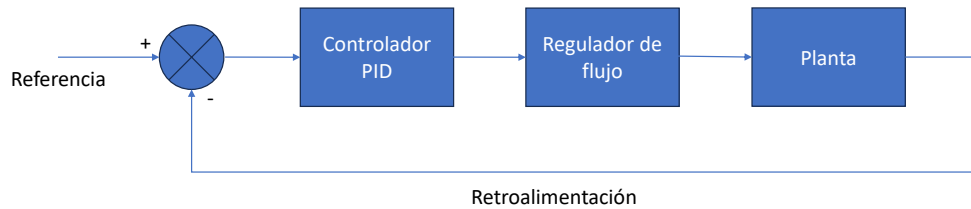


Figura 3.8: Diagrama de bloques de controlador PID.

3.6.2. Controlador PID difuso

Para los controladores con algoritmos de inteligencia artificial se decidió utilizar un controlador PID difuso para compensar las zonas en las que las no linealidades afectaban el desempeño del controlador PID. Al tener un controlador difuso, la entrada puede determinar el valor de la ganancia que se administrará al controlador PID, reduciendo así los efectos de algunas de las no linealidades. En la Figura 3.9 se puede observar el diagrama de bloques del controlador PID difuso.

El controlador PID difuso requiere de unas funciones de pertenencia y reglas de inferencia para su funcionamiento. Para las funciones de pertenencia de las entradas se tienen unas funciones de pertenencia triangulares para cada grado de libertad, las cuales se muestran en la Figura 3.10. Se usaron funciones triangulares debido a su simplicidad y bajo costo computacional.

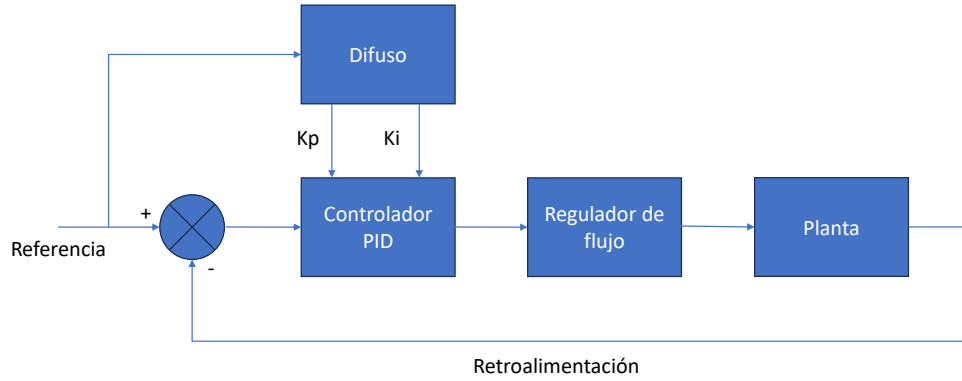
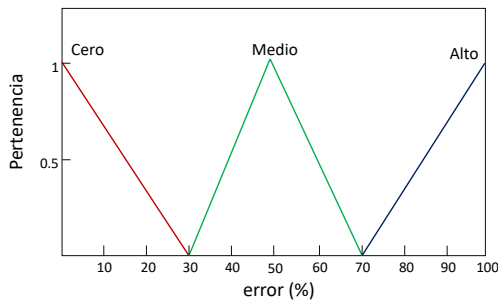
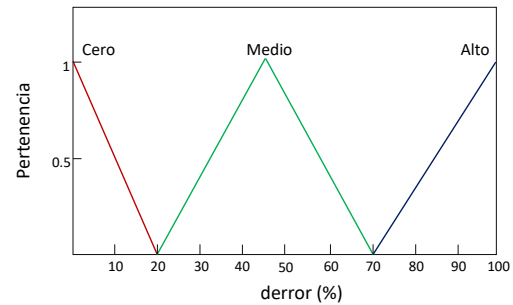


Figura 3.9: Diagrama de bloques de controlador PID difuso.

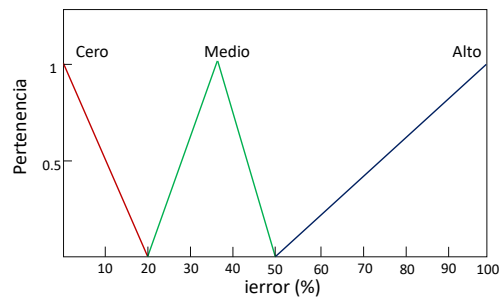
La Figura 3.10a muestra la pertenencia de la entrada error para el primer grado de libertad, mientras que las Figuras 3.10b y 3.10c muestran el segundo y tercero, respectivamente. Estas mismas pertenencias se usan para la entrada derivada del error.



(a) 1er Grado de libertad.



(b) 2do Grado de libertad.

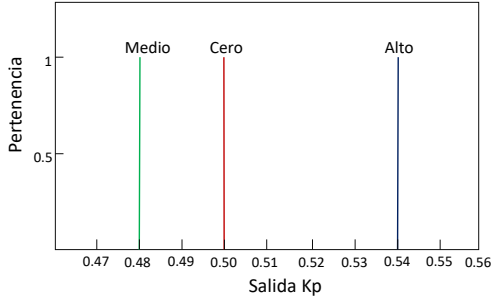


(c) 3er Grado de libertad.

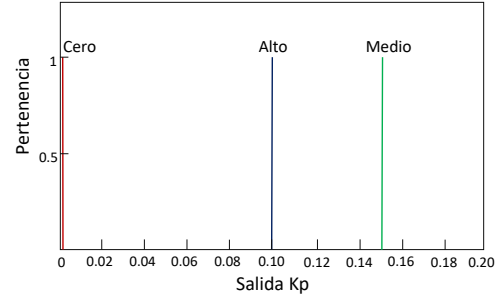
Figura 3.10: Función de pertenencia de las entradas.

Para la salida tenemos funciones de pertenencia singleton, de igual manera que para las entradas se tiene una salida K_p por cada grado de libertad, como se muestra en la Figura 3.11. Las funciones singleton se usaron debido a que el valor de la salida es uno solo que depende de la entrada y no

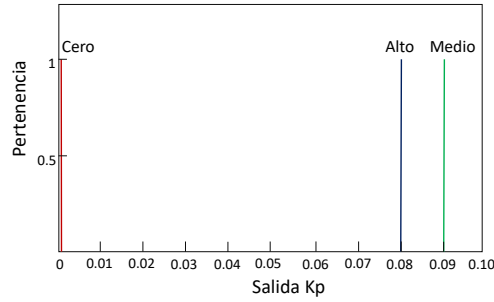
de un rango de valores.



(a) 1er Grado de libertad.



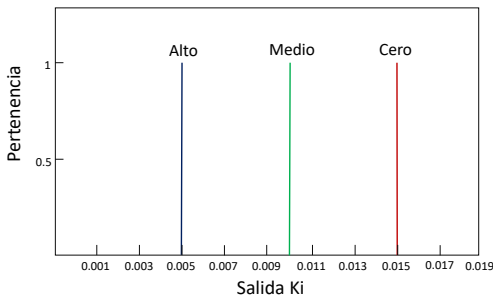
(b) 2do Grado de libertad.



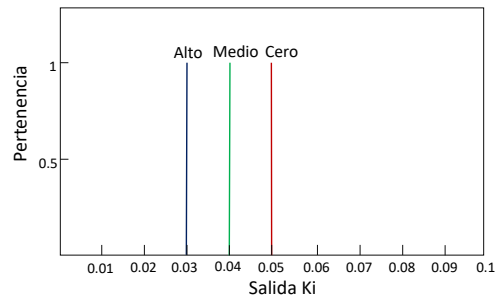
(c) 3er Grado de libertad.

Figura 3.11: Función de pertenencia de las salidas K_p .

Para el caso de la salida K_i se tiene una función de pertenencia de salida para el primer y segundo grado de libertad y otra para el tercer grado, como se muestra en la Figura 3.12. De igual manera que para la salida K_p , la función de pertenencia es de tipo singleton.



(a) 1er y 2do Grado de libertad.



(b) 3er Grado de libertad.

Figura 3.12: Función de pertenencia de las salidas K_i .

Finalmente, para determinar qué salida se obtiene a partir de cierta entrada, se tienen las reglas de inferencia, las cuales están listadas a continuación:

- Si error es "Cero", entonces K_p es "Cero"
- Si error es "Medio", entonces K_p es "Medio"
- Si error es "Alto", entonces K_p es "Alto"
- Si ierror es "Cero", entonces K_i es "Cero"
- Si ierror es "Medio", entonces K_i es "Medio"
- Si ierror es "Alto", entonces K_i es "Alto"

Los valores numéricos para cada grado de libertad se encuentran listados en la Tabla 3.3.

Tabla 3.3: Valores numéricos de funciones de pertenencia por grado de libertad.

DOF	Entrada (%)	Salida K_p	Salida K_i	Salida K_d
1ero	0-30	0.5	0.015	0.13
	30-70	0.48	0.010	
	70-100	0.54	0.005	
2do	0-20	0.001	0.015	0.001
	20-70	0.15	0.010	
	70-100	0.1	0.005	
3ero	0-20	0.001	0.05	0.0005
	20-50	0.09	0.04	
	50-100	0.08	0.03	

3.6.3. Controlador con red neuronal

Finalmente, se optó por usar un controlador basado en una red neuronal de retropropagación. Con esto, a partir de los datos obtenidos por la planta, la red aprenderá de los datos obtenidos modificando las ganancias. La red neuronal está programada de tal forma que sustituya al controlador PID, pero modificando las ganancias de este cada vez que se obtiene un dato nuevo de la planta. El diagrama de bloques del controlador con red neuronal de retropropagación se muestra en la Figura 3.13.

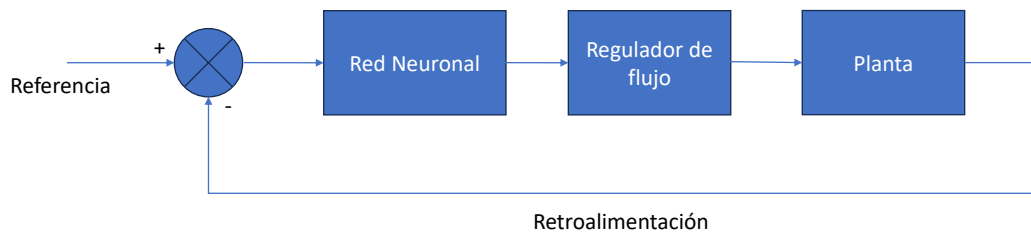


Figura 3.13: Diagrama de bloques de controlador con red neuronal de retropropagación.

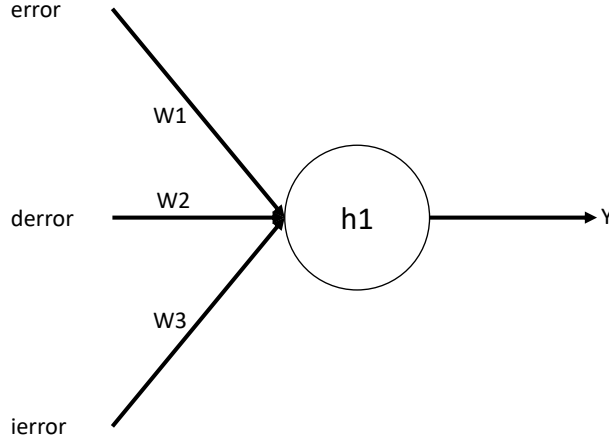


Figura 3.14: Topología de la red neuronal de retropropagación.

El controlador de red neuronal de retropropagación utiliza una sola neurona, como la que se muestra en la Figura 3.14, para cada grado de libertad.

Los pesos iniciales por grado de libertad, así como el factor de aprendizaje, se encuentran listados en la Tabla 3.4. Para este caso, al no tener los datos de entrada normalizados y tener un ADC de 12 bits, así como cambiar de un valor cercano a 0 a un valor cercano a 4000 en un tiempo muy corto, el factor de aprendizaje tiene que ser pequeño para no desbordar la variable del microcontrolador.

Tabla 3.4: Pesos iniciales y factor de aprendizaje de neurona por grado de libertad.

DOF	W_1	W_2	W_3	\mathbf{n}
1ero	0.5	0.13	0.008	2×10^{-12}
2do	0.2	0.001	0.008	1×10^{-12}
3ero	0.001	0.0005	0.03	5×10^{-13}

La red neuronal tiene una función de activación lineal ya que al tener alguna otra aumentaría mucho el costo computacional al tener que realizar operaciones más complejas. Las ecuaciones utilizadas para la actualización de los pesos son las siguientes:

$$Y = w_1 + w_2 + w_3 \quad (3.4)$$

$$E = Y_d - Y \quad (3.5)$$

$$\Delta w_1 = w_1 * E * \delta \quad (3.6)$$

$$\Delta w_2 = w_2 * E * \delta \quad (3.7)$$

$$\Delta w_3 = w_3 * E * \delta \quad (3.8)$$

$$w_1 = w_1 + \Delta w_1 \quad (3.9)$$

$$w_2 = w_2 + \Delta w_2 \quad (3.10)$$

$$w_3 = w_3 + \Delta w_3 \quad (3.11)$$

3.7. Pruebas

Las pruebas realizadas fueron llevadas a cabo de la siguiente manera.

1. Prueba con controlador PID en un grado de libertad en PIC18F4550.
2. Ajustes de ganancias de controlador PID.
3. Prueba con controlador PID difuso en un grado de libertad.
4. Ajustes de salidas de logica difusa.
5. Prueba con controlador con red neuronal de retropropagación en un grado de libertad.
6. Ajustes de pesos iniciales y factor de aprendizaje.
7. Repetir pruebas anteriores para 3 grados de libertad con los 3 controladores.
8. Ajustes en controladores de siguientes grados de libertad.
9. Repetir pruebas para un grado de libertad en STM32F303RE.
10. Ajustes de ganancias en STM32F303RE.
11. Repetir pruebas para 3 grados de libertad.
12. Comparación de resultados.

Resultados

En este capítulo se describirán los resultados obtenidos para cada tipo de controlador utilizado, una comparativa entre ellos y entre el estado del arte del proyecto, así como una discusión de los resultados obtenidos.

Los resultados obtenidos para un grado de libertad se separan en:

- Control PID.
- Control PID-difuso.
- Control con Red neuronal de retropropagación.

4.1. Control PID

Para las pruebas con control PID se tienen unas ganancias fijas para el primer grado de libertad de 0.5, 0.13 y 0.008 para la ganancia proporcional, la derivativa y la integral, respectivamente. Para el segundo grado de libertad se tienen unas ganancias de 0.2, 0.001 y 0.008 para la ganancia proporcional, la derivativa y la integral, respectivamente. Por último, para el tercer grado de libertad se tiene una ganancia proporcional de 0.1, una derivativa de 0.0005 y una integral de 0.03. El cuarto grado de libertad, en esta etapa de la investigación, solo cuenta con dos posiciones deseadas que, al ser un cilindro neumático o pistón, son con el vástago retraído y el vástago extendido, por lo que no cuenta con un controlador y se dejó como on-off. El tiempo de muestreo es de 50 ms.

Se observa en la Figura 4.1 que, para los 3 experimentos realizados, la posición del actuador es muy cercana al setpoint, teniendo un sobrepaso máximo cercano al 9.1 % y un error en estado estacionario menor a 8° con un tiempo de asentamiento cerca de los 14 s.

Por otra parte, en la Figura 4.2, se observa el comportamiento del actuador del segundo grado de libertad; esta muestra que el error en estado estacionario mayor, para los 3 experimentos realizados, es menor de 5° con un tiempo de asentamiento cerca de los 14.3 s y teniendo un sobrepaso máximo cerca del 40 %.

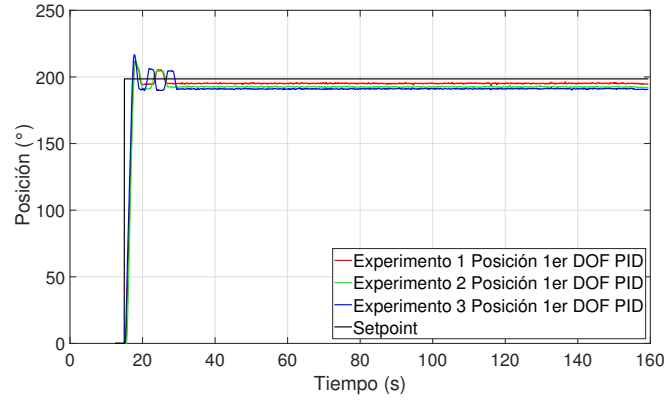


Figura 4.1: Posición de primer grado de libertad con controlador PID.

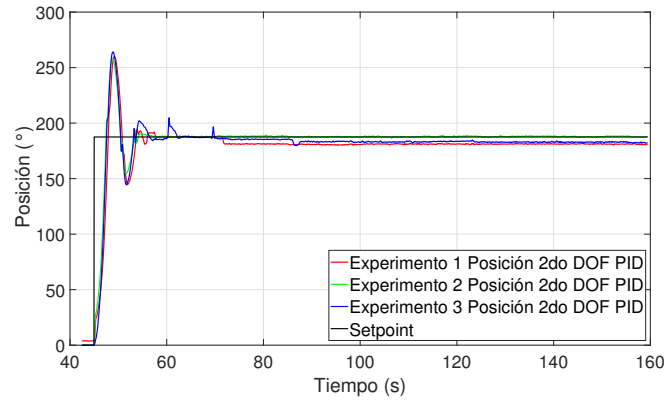


Figura 4.2: Posición de segundo grado de libertad con controlador PID.

Finalmente, para el tercer grado de libertad, la Figura 4.3 muestra que se tiene un sobrepaso máximo cerca del 55 % y un error en estado estacionario de aproximadamente 13° con un tiempo de asentamiento menor a 80 s.

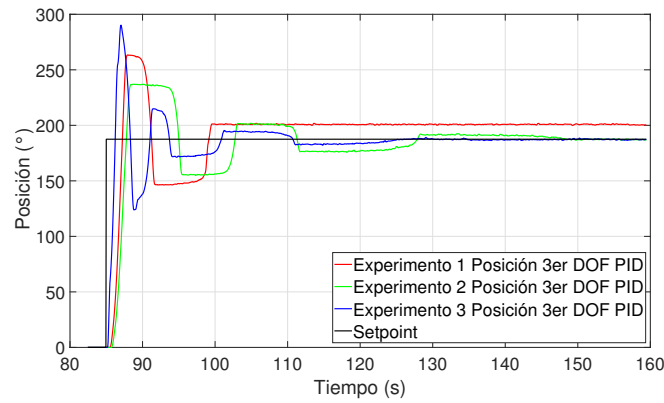
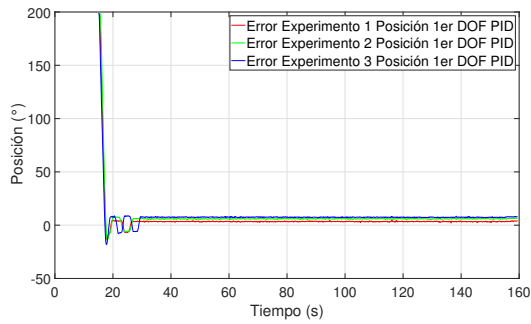
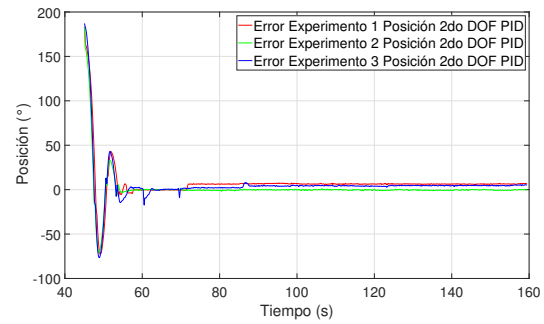


Figura 4.3: Posición de tercer grado de libertad con controlador PID.

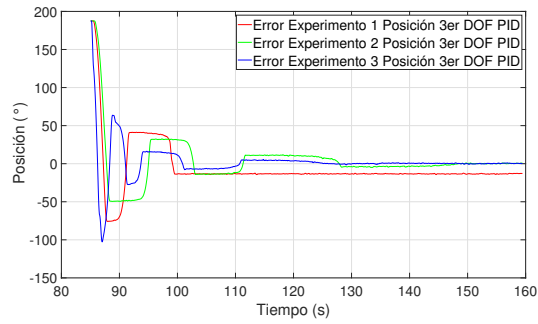
En la Figura 4.4 se puede observar la gráfica de los errores de cada experimento para cada grado de libertad.



(a) 1er Grado de libertad.



(b) 2do Grado de libertad.



(c) 3er Grado de libertad.

Figura 4.4: Errores por cada grado de libertad del controlador PID.

Al usar la cinemática directa con los resultados obtenidos, se obtuvieron las coordenadas en el eje x y en el eje y para la posición del efector final del robot. Con estas coordenadas se obtuvieron los valores del origen al efector final, los cuales se muestran en la Figura 4.5 para los tres experimentos realizados.

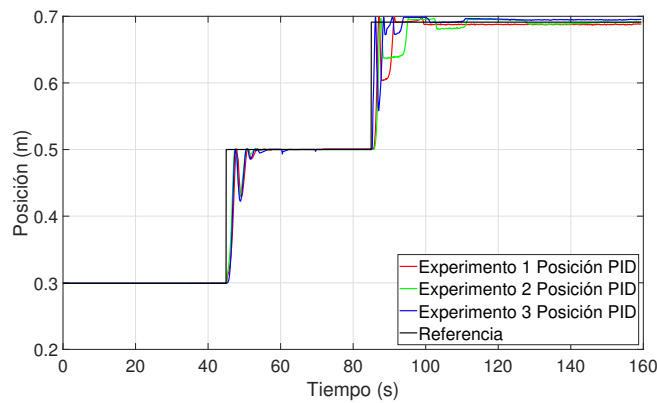


Figura 4.5: Posición del efector final con controlador PID.

4.2. Control PID difuso

Para las pruebas con control PID difuso se tienen unas ganancias para el primer grado de libertad fijas de 0.5, 0.13 y 0.008 para la ganancia proporcional, la ganancia derivativa y la ganancia integral, respectivamente. Para el segundo grado de libertad se tienen unas ganancias de 0.2, 0.001 y 0.008 para la ganancia proporcional, la ganancia derivativa y la ganancia integral, respectivamente. Por último, para el tercer grado de libertad se tiene una ganancia proporcional de 0.1, una derivativa de 0.0005 y una integral de 0.03. El cuarto grado de libertad, el cual solo tiene la función de bajar y subir el efector final, no cuenta con un controlador. El tiempo de muestreo es de 10 ms.

Se observa en la Figura 4.6 que, para los 3 experimentos realizados, la posición del actuador es muy cercana al setpoint, teniendo un sobrepaso máximo cercano al 9.6% y un error en estado estacionario menor a 2° con un tiempo de asentamiento cercano a los 4.3 s.

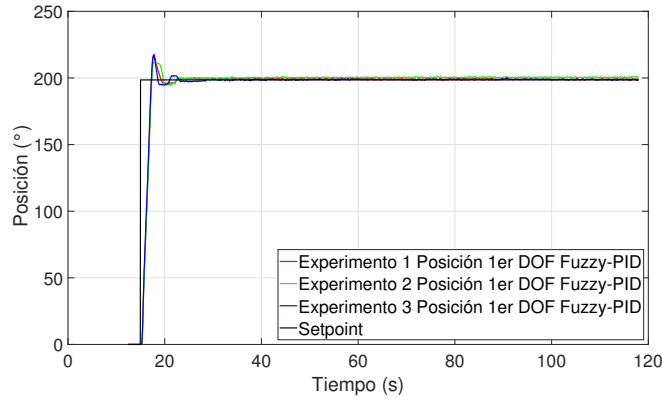


Figura 4.6: Posición de primer grado de libertad con controlador PID difuso.

Por otra parte, en la Figura 4.7, se observa el comportamiento del actuador del segundo grado de libertad; esta muestra un error en estado estacionario cerca de 3° con un tiempo de asentamiento menor a 12 s y teniendo un sobrepaso máximo cerca del 39%.

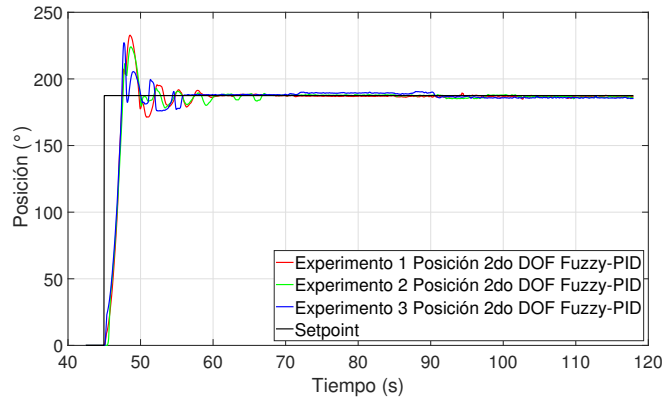


Figura 4.7: Posición de segundo grado de libertad con controlador PID difuso.

Finalmente, para el tercer grado de libertad, la Figura 4.8 muestra que se tiene un sobrepaso máximo cerca del 34 % y un error en estado estacionario menor a 3.5° con un tiempo de asentamiento menor a 42 s.

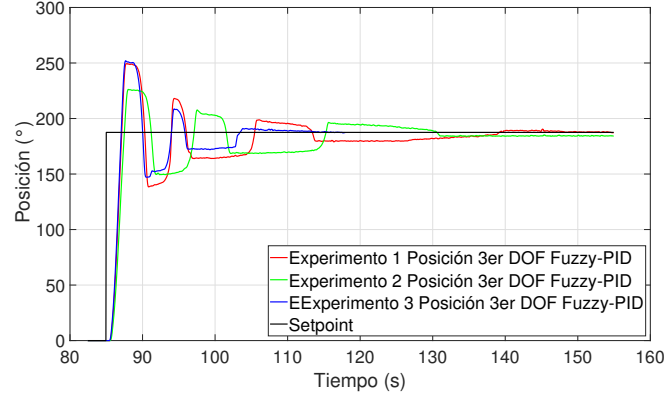
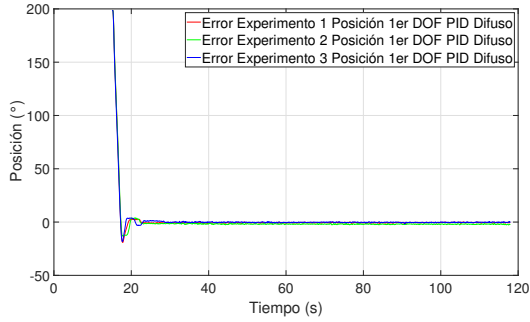
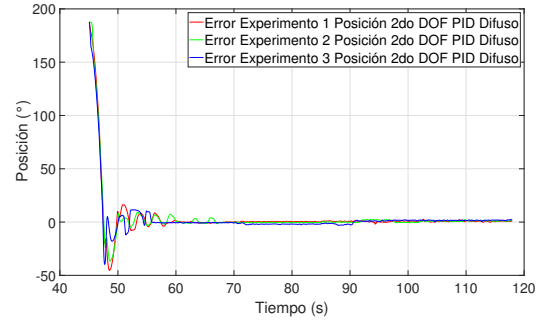


Figura 4.8: Posición de tercer grado de libertad con controlador PID difuso.

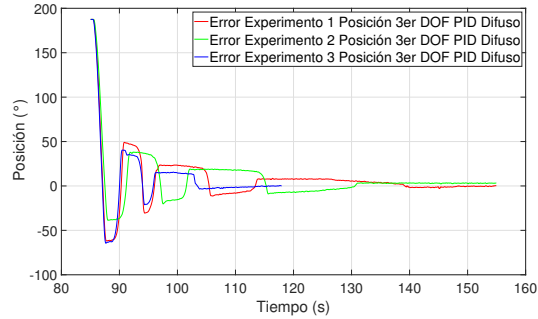
En la Figura 4.9 se puede observar la gráfica de los errores de cada experimento para cada grado de libertad.



(a) 1er Grado de libertad.



(b) 2do Grado de libertad.



(c) 3er Grado de libertad.

Figura 4.9: Errores por cada grado de libertad del controlador PID difuso.

De igual manera que con el controlador PID, se obtuvieron las posiciones del efector final del robot para los tres experimentos realizados, los cuales se muestran en la Figura 4.10.

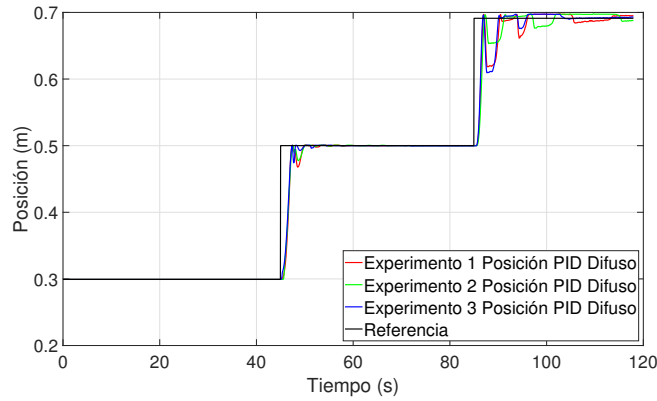


Figura 4.10: Posición del efector final con controlador PID Difuso.

4.3. Control con Red neuronal de retropropagación

Para las pruebas con control con red neuronal de retropropagación se tienen unas ganancias variables, las cuales se modifican dependiendo de la posición actual del controlador, mediante una neurona en la que entran los valores del error, la derivada del error y la integral del error. Estas entradas se procesan en la neurona y dan como salida la señal de control para el control del flujo de aire.

Se observa en la Figura 4.11 que, para los 3 experimentos realizados, la posición del actuador es muy cercana al setpoint, teniendo un sobrepaso máximo cercano al 1 % y un error en estado estacionario menor a 1.5° con un tiempo de asentamiento cerca de los 3.6 s.

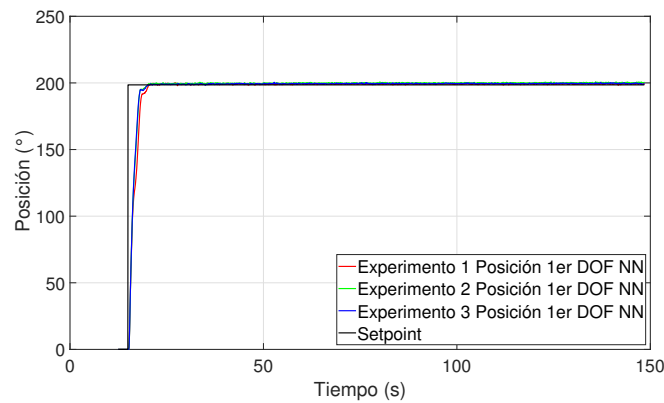


Figura 4.11: Posición de primer grado de libertad con controlador NN.

Por otra parte, en la Figura 4.12, se observa el comportamiento del actuador del segundo grado de libertad. El error en estado estacionario es cerca de 1° con un tiempo de asentamiento cerca de los 22.5 s y teniendo un sobrepaso máximo cerca del 38 %.

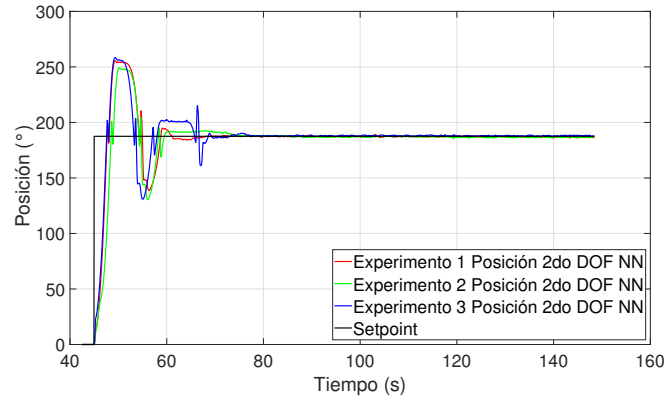


Figura 4.12: Posición de segundo grado de libertad con controlador NN.

Finalmente, para el tercer grado de libertad, la Figura 4.13 muestra que se tiene un sobrepaso máximo cerca del 61 % y un error en estado estacionario de aproximadamente 2.5° con un tiempo de asentamiento menor a 52.4 s.

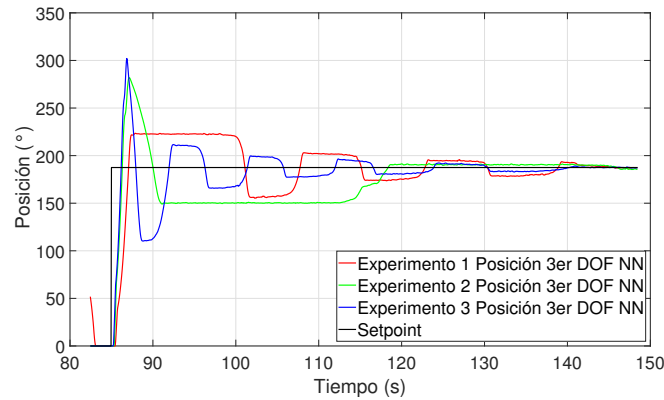
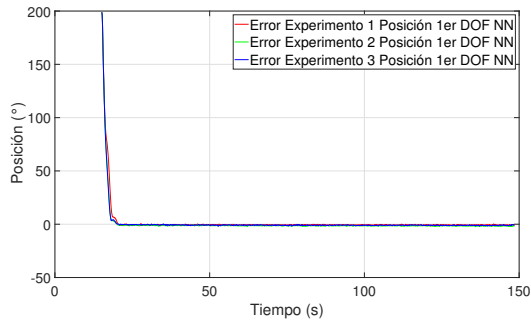


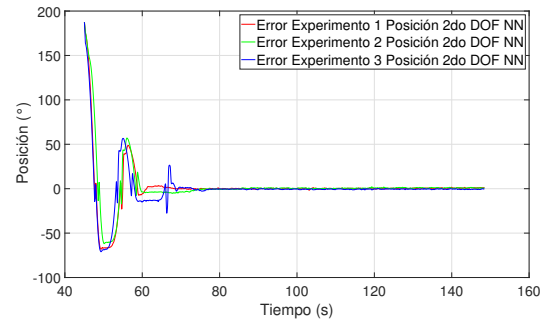
Figura 4.13: Posición de tercer grado de libertad con controlador NN.

En la Figura 4.14 se puede observar la gráfica de los errores de cada experimento para cada grado de libertad.

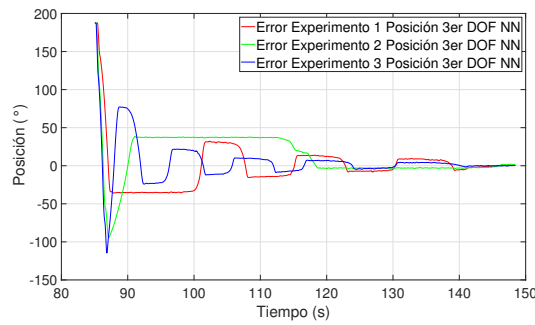
Finalmente, al igual que con el controlador PID y PID Difuso, se obtuvieron las posiciones del efector final del robot para los tres experimentos realizados, los cuales se muestran en la Figura 4.15.



(a) 1er Grado de libertad.



(b) 2do Grado de libertad.



(c) 3er Grado de libertad.

Figura 4.14: Errores por cada grado de libertad del controlador con red neuronal.

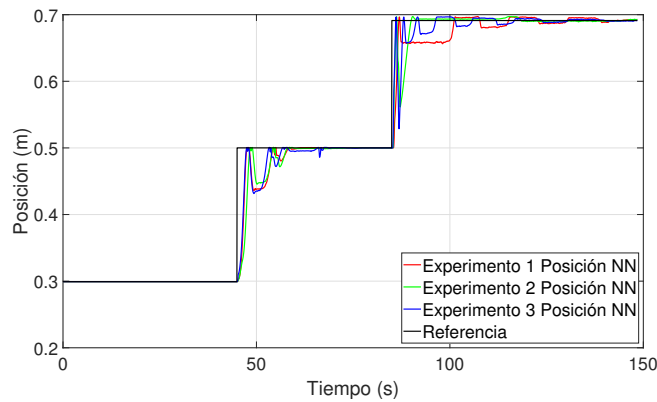


Figura 4.15: Posición del efector final con controlador de red neuronal.

4.4. Comparativa

En la Tabla 4.1 se pueden observar los porcentajes de error de cada uno de los controladores por grado de libertad. Esto se calculó obteniendo el promedio de error de cada experimento una vez alcanzado el estado estacionario, el cual se definió como el 96 % de la señal de entrada, y después promediando el error de todos los experimentos y calculando el porcentaje respecto al valor de referencia.

Tabla 4.1: Porcentaje de error de cada controlador por cada grado de libertad.

DOF	PID	PID difuso	NN
1er	2.7391	0.4539	0.4421
2do	1.2763	0.2456	0.0351
3er	2.0535	0.2042	0.4586

En la Tabla 4.2 se pueden observar los valores de error en estado estacionario (eee), sobrepaso (OS) y tiempo de asentamiento (t_s) máximos de las pruebas realizadas para cada controlador por grado de libertad.

Tabla 4.2: Error en estado estacionario, tiempo de asentamiento y sobrepaso máximo.

	DOF	eee (°)	t_s (s)	OS (%)
PID	1er	7.4876	13.95	9.11
PID difuso		-1.8905	19.35	8.79
NN		-1.3224	18.55	1.16
PID	2do	5.2294	14.35	40.84
PID difuso		0.9189	11.65	38.96
NN		-0.4131	22.5	37.84
PID	3er	-13.2616	79.9	54.76
PID difuso		1.4232	41.6	34.40
NN		-1.3209	52.35	61.06

La Figura 4.16 muestra la comparativa del error promedio absoluto de cada controlador por cada grado de libertad.

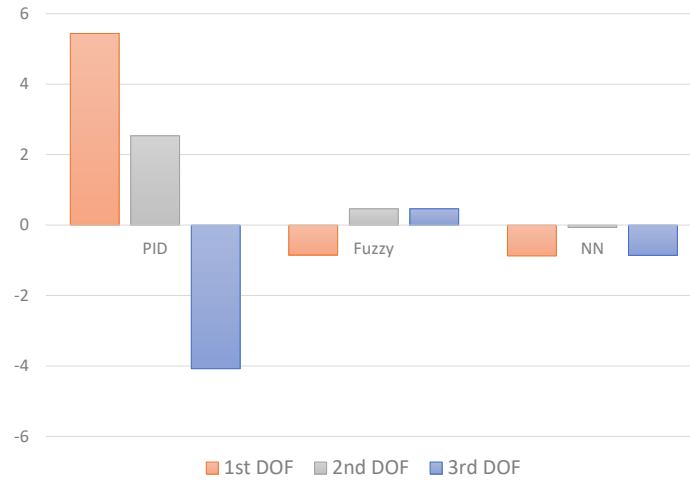


Figura 4.16: Comparativa de MAE de controladores por cada grado de libertad.

4.5. Discusión

En la Figura 4.16 se puede observar que el controlador con el peor desempeño es el controlador PID, ya que su error en estado estacionario, para los 3 grados de libertad controlados, es mucho mayor que el caso del controlador PID difuso y el de red neuronal.

De la Tabla 4.2 se puede observar que de los errores máximos obtenidos el controlador PID difuso mejoró en un 74.75 %, un 82.43 % y un 89.27 % para el primero, segundo y tercer grado de libertad, respectivamente, mientras que para el controlador con red neuronal mejoró en un 82.34 %, un 92.10 % y un 90.04 %, esto en comparación con el controlador PID. De igual forma, se puede determinar que el controlador con red neuronal mejoró en comparación con el PID difuso en un 30.05 %, 55.04 % y 7.19 %.

En cuanto a la comparación con otras investigaciones, se compararon los resultados obtenidos con un controlador PID y se compararon con los obtenidos con el controlador PID difuso y el de red neuronal. Esto es algo complicado de comparar, ya que, ni los actuadores, ni los sensores, ni las válvulas, ni la aplicación, ni las pruebas son iguales; se realizó una comparativa entre el porcentaje de error alcanzado por estas investigaciones y el porcentaje de error obtenido en esta. Es importante también mencionar que en las investigaciones realizadas no siempre se tiene un controlador PID puro, ya que en algunas de ellas se utilizó algún método de optimización. En la Tabla 4.1 se tienen los porcentajes obtenidos en esta investigación y en la Tabla 4.3 se tienen los obtenidos en otras investigaciones donde podemos ver que el posicionamiento mejoró por lo menos en un 31.23 % en el caso del controlador PID difuso y un 30.52 % en el controlador con red neuronal.

Tabla 4.3: Porcentaje de error obtenido en otras investigaciones.

Referencia	Error (%)	Tipo de controlador	Mejora con PID difuso (%)	Mejora con NN (%)
[16]	1.50	NCD-PID	69.74	69.43
[19]	10	FFPID	95.46	95.41
[3]	0.66	MPC	31.23	30.52
[2]	16.60	ACRPPC-FOPID	97.27	97.24
[9]	1.35	FOPID	66.38	66.03
[11]	3.05	PID	85.12	84.96

Conclusiones

Se puede concluir que el uso de un AIA mejora el posicionamiento de los actuadores de un robot SCARA de 4 grados de libertad. Esto queda demostrado tanto con las pruebas realizadas en la presente investigación en las que la comparación entre el controlador PID y los que usan algún AIA mejora en por lo menos un 74 % el posicionamiento de los actuadores. Por otra parte, también se demostró que en comparación con investigaciones previas, que usan un controlador PID, por lo menos se mejora en un 30.52 % el porcentaje de error en el posicionamiento de los actuadores.

Es importante mencionar que varias de estas investigaciones usan válvulas proporcionales en comparación con esta en la que se usan solenoides para la activación del flujo de aire, el cual previamente se regula con el uso de una válvula de estrangulamiento en conjunto con un motor a pasos. Esto reduce el costo de materiales en gran medida, pero en caso de necesitarse una mayor precisión se puede optar por el uso de válvulas proporcionales para determinar si esto es un factor importante en el control de flujo de aire. Por otra parte, en esta investigación se usaron solo sensores de posición, por lo que el uso de sensores de presión aunados a los de posición podría resultar en un mejor posicionamiento de los actuadores.

La adquisición de datos también juega un papel importante para el posicionamiento; si bien, el microcontrolador usado tiene un convertidor ADC de 12 bits, lo que nos da una buena resolución de la posición de los actuadores, el tiempo de muestreo aún se puede mejorar con componentes más especializados en altas velocidades. Esto podría resultar en una respuesta más rápida, lo que podría reducir el tiempo de asentamiento y el sobrepaso.

En cuestiones mecánicas, se tuvieron varios problemas con las piezas que unían los actuadores con los eslabones, ya que estas fueron construidas con impresión 3D lo que resultaba en un desgaste muy rápido de estas piezas. Como mejora, se podrían sustituir estas piezas por unas construidas con un material más duradero. Otro problema que se tuvo que enfrentar fue la fricción en los rodamientos causada por el desbalanceo de los eslabones, esto se trató de solucionar colocando contrapesos en los eslabones pero no se pudo eliminar por completo. Los resultados de la investigación resultaron satisfactorios aunque aún se tengan algunos puntos a mejorar.

Bibliografía

- [1] IFR, “World robotics industrial robots 2023,” 2023.
- [2] M. I. P. Azahar, A. Irawan, and R. R. Ismail, “Adjustable convergence rate prescribed performance with fractional-order pid controller for servo pneumatic actuated robot positioning,” *Cognitive Robotics*, vol. 3, pp. 93–106, 2023.
- [3] M. N. Muftah, A. A. M. Faudzi, S. Sahlan, and S. Mohamaddan, “Intelligent position control for intelligent pneumatic actuator with ball-beam (ipabb) system,” *Applied Sciences*, vol. 12, no. 21, p. 11089, 2022.
- [4] J.-M. Ramos-Arreguin, E. Guillen-Garcia, S. Canchola-Magdaleno, J.-C. Pedraza-Ortega, E. Gorrostieta-Hurtado, M.-A. Aceves-Fernández, and C.-A. Ramos-Arreguin, “Fuzzy logic hardware implementation for pneumatic control of one dof pneumatic robot,” in *Advances in Soft Computing: 9th Mexican International Conference on Artificial Intelligence, MICAI 2010, Pachuca, Mexico, November 8-13, 2010, Proceedings, Part II 9*, pp. 500–511, Springer, 2010.
- [5] E. G. Islas, “La era de los robots,” 2019.
- [6] B. Rouzbeh, G. M. Bone, and G. Ashby, “High-accuracy position control of a rotary pneumatic actuator,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2774–2781, 2018.
- [7] Z. Situm, D. Pavkovic, and B. Novakovic, “Servo pneumatic position control using fuzzy pid gain scheduling,” *J. Dyn. Sys., Meas., Control*, vol. 126, no. 2, pp. 376–387, 2004.
- [8] J.-M. Ramos-Arreguin, J. C. P. Ortega, E. Gorrostieta, and R. de Jesus Romero-Troncoso, “Artificial intelligence applied into pneumatic flexible manipulator,” in *2008 Seventh Mexican International Conference on Artificial Intelligence*, pp. 339–345, IEEE, 2008.
- [9] M. N. Muftah, A. A. M. Faudzi, S. Sahlan, and S. Mohamaddan, “Fuzzy fractional order pid tuned via pso for a pneumatic actuator with ball beam (pabb) system,” *Fractal and Fractional*, vol. 7, no. 6, p. 416, 2023.
- [10] M. I. P. Azahar, A. Irawan, and R. R. Ismail, “Self-tuning hybrid fuzzy sliding surface control for pneumatic servo system positioning,” *Control Engineering Practice*, vol. 113, p. 104838, 2021.
- [11] G. Gregov, S. Pincin, A. Šoljić, and E. Kamenar, “Position control of a cost-effective bellow pneumatic actuator using an lqr approach,” in *Actuators*, vol. 12, p. 73, MDPI, 2023.

- [12] Ž. Šitum and D. Čorić, “Position control of a pneumatic drive using a fuzzy controller with an analytic activation function,” *Sensors*, vol. 22, no. 3, p. 1004, 2022.
- [13] F. Escobar, S. Díaz, C. Gutiérrez, Y. Ledeneva, C. Hernández, D. Rodríguez, and R. Lemus, “Simulation of control of a scara robot actuated by pneumatic artificial muscles using rnapm,” *Journal of applied research and technology*, vol. 12, no. 5, pp. 939–946, 2014.
- [14] D. Bou Saba, E. Bideaux, X. Brun, and P. Massioni, “A complete model of a two degree of freedom platform actuated by three pneumatic muscles elaborated for control synthesis,” in *Fluid Power Systems Technology*, vol. 50060, p. V001T01A004, American Society of Mechanical Engineers, 2016.
- [15] W. Tuvayanond and M. Parnichkun, “Position control of a pneumatic surgical robot using pso based 2-dof h loop shaping structured controller,” *Mechatronics*, vol. 43, pp. 40–55, 2017.
- [16] Y. Zhang, K. Li, S. Wei, and G. Wang, “Pneumatic rotary actuator position servo system based on ade-pd control,” *Applied Sciences*, vol. 8, no. 3, p. 406, 2018.
- [17] L.-W. Lee, H.-H. Chiang, and I.-H. Li, “Development and control of a pneumatic-actuator 3-dof translational parallel manipulator with robot vision,” *Sensors*, vol. 19, no. 6, p. 1459, 2019.
- [18] S. Mu, S. Goto, S. Shibata, and T. Yamamoto, “Intelligent position control for pneumatic servo system based on predictive fuzzy control,” *Computers & Electrical Engineering*, vol. 75, pp. 112–122, 2019.
- [19] C. Chan, S. Chong, S. Loh, A. Alias, and H. Kasdirin, “Positioning control of an antagonistic pneumatic muscle actuated system using feedforward compensation with cascaded control scheme,” *International Journal of Integrated Engineering*, vol. 12, no. 2, pp. 70–74, 2020.
- [20] S. Hoang, K. Karydis, P. Brisk, and W. H. Grover, “A pneumatic random-access memory for controlling soft robots,” *Plos one*, vol. 16, no. 7, p. e0254524, 2021.
- [21] M. S. Xavier, C. D. Tawk, A. Zolfagharian, J. Pinski, D. Howard, T. Young, J. Lai, S. M. Harrison, Y. K. Yong, M. Bodaghi, *et al.*, “Soft pneumatic actuators: A review of design, fabrication, modeling, sensing, control and applications,” *IEEE Access*, vol. 10, pp. 59442–59485, 2022.
- [22] H. Su, X. Hou, X. Zhang, W. Qi, S. Cai, X. Xiong, and J. Guo, “Pneumatic soft robots: Challenges and benefits,” in *Actuators*, vol. 11, p. 92, MDPI, 2022.
- [23] M.-A. Cabrera-Rufino, J.-M. Ramos-Arreguín, J. Rodríguez-Reséndiz, E. Gorrostieta-Hurtado, and M.-A. Aceves-Fernandez, “Implementation of ann-based auto-adjustable for a pneumatic servo system embedded on fpga,” *Micromachines*, vol. 13, no. 6, p. 890, 2022.
- [24] S. A. Al-Ibadi, L. A. T. Al Abeach, and M. A. A. Al-Ibadi, “Soft robots: Implementation, modeling, and methods of control,” *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, vol. 11, no. 1, pp. 194–209, 2023.

- [25] I. A. Seleem, H. El-Hussieny, and H. Ishii, “Recent developments of actuation mechanisms for continuum robots: A review,” *International Journal of Control, Automation and Systems*, vol. 21, no. 5, pp. 1592–1609, 2023.
- [26] Z. Zhang, L. Zhang, M. Guan, S. Zhang, and T. Jiao, “Research on a variable pressure driving method for soft robots based on the electromagnetic effect,” *Sensors*, vol. 23, no. 14, p. 6341, 2023.
- [27] G. Dämmer, S. Gablenz, R. Neumann, and Z. Major, “Design, topology optimization, and additive manufacturing of a pneumatically actuated lightweight robot,” in *Actuators*, vol. 12, p. 266, MDPI, 2023.
- [28] J. F. García-Samartín, A. Rieker, and A. Barrientos, “Design, manufacturing, and open-loop control of a soft pneumatic arm,” in *Actuators*, vol. 13, p. 36, MDPI, 2024.
- [29] J. J. Craig, *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009.
- [30] J. A. P. Delgado, J. M. R. Arreguín, G. I. P. Soto, and E. G. Hurtado, “Diseño, modelo cinemático y simulación de un robot neumático de 4 dof (design, kinematic model and simulation of a 4 dof pneumatic robot),” *Pistas Educativas*, vol. 40, no. 130, 2018.
- [31] P. J. Antsaklis and A. N. Michel, *Linear systems*, vol. 8. Springer, 1997.
- [32] T. Kailath, *Linear systems*, vol. 156. Prentice-Hall Englewood Cliffs, NJ, 1980.
- [33] H. Broulès and G. K. Kwan, *Linear systems*. John Wiley & Sons, 2013.
- [34] K. Ogata *et al.*, *Modern control engineering*. Prentice Hall India, 2009.
- [35] N. S. Nise, *Control systems engineering*. John Wiley & Sons, 2020.
- [36] G. Chen, *Nonlinear Systems: Stability, Dynamics and Control*. World Scientific, 2024.
- [37] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.
- [38] B. C. Kuo, *Sistemas de control automático*. Prentice Hall, 7 ed., 1996.
- [39] J. Jantzen, *Foundations of fuzzy control: a practical approach*. John Wiley & Sons, 2013.
- [40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [41] S. Haykin, *Neural networks and learning machines, 3/E*. Pearson Education India, 2009.