



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**TEMA: REHABILITACIÓN DEL SISTEMA DE LEVITACIÓN
MAGNÉTICA MLS Y DISEÑO E IMPLEMENTACIÓN DE
ALGORITMOS DE CONTROL AVANZADO**

**AUTOR:
SR. PÉREZ VILLACÍS, PAÚL SEBASTIÁN**

DIRECTOR: ING. AGUILAR JARAMILLO, EDWIN RENÉ

SANGOLQUÍ

2017



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación: “**REHABILITACIÓN DEL SISTEMA DE LEVITACIÓN MAGNÉTICA MLS Y DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL AVANZADO**”, realizado por el señor **PAÚL SEBASTIÁN PÉREZ VILLACÍS**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, por lo tanto, me permito acreditarlo y autorizar al señor **PAÚL SEBASTIÁN PÉREZ VILLACÍS** para que lo sustente públicamente.

Sangolquí, agosto de 2017

Ing. Edwin René Aguilar Jaramillo MSc.

DIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, **PAÚL SEBASTIÁN PÉREZ VILLACÍS** con cédula de identidad N°: 180433898-4, declaro que el trabajo de titulación “**REHABILITACIÓN DEL SISTEMA DE LEVITACIÓN MAGNÉTICA MLS Y DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL AVANZADO**”, ha sido desarrollado tomando en cuenta los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente manifiesto que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance del proyecto mencionado.

Sangolquí, agosto de 2017

A handwritten signature in blue ink is positioned above a horizontal line. The signature is cursive and appears to read 'Paúl Sebastián Pérez Villacís'.

Sr. Paúl Sebastián Pérez Villacís

C.C. 180433898-4



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, **PAÚL SEBASTIÁN PÉREZ VILLACÍS**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca virtual de la institución el presente trabajo de titulación “**REHABILITACIÓN DEL SISTEMA DE LEVITACIÓN MAGNÉTICA MLS Y DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL AVANZADO**”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, agosto de 2017

Sr. Paúl Sebastián Pérez Villacís

C.C. 180433898-4

DEDICATORIA

Quiero dedicar este proyecto de investigación a mi familia, ya que gracias a su apoyo no podría haber logrado este objetivo en mi vida.

A mi madre Gladys, por apoyarme y guiarme en las decisiones que tomé a lo largo de mi camino.

A mi padre Jorge, que supo enseñarme que todo esfuerzo por los seres que amas vale la pena, y que, a pesar de no estar junto a mí, sé que me brinda fuerza para seguir adelante.

A mi hermano Jorge Luis, por ser mi ejemplo, y demostrarme que todo lo que te plantees en la vida está a tu alcance si te lo propones, con tesón y sacrificio.

AGRADECIMIENTO

A mi familia, por estar presente, por guiarme y ayudarme a darme cuenta de mis errores.

A todos mis amigos, que han estado en los buenos y malos momentos, con quienes hemos compartido experiencias de vida que me han permitido crecer como persona.

A la Universidad de las Fuerzas Armadas ESPE, a los docentes que me han compartido sus conocimientos a lo largo de mi carrera, de manera especial a los Ingenieros Víctor Proaño y Edwin Aguilar por guiarme y brindarme su ayuda en la consecución de este proyecto.

ÍNDICE DE CONTENIDO

CERTIFICACIÓN	ii
AUTORÍA DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDO	vii
RESUMEN	xiv
ABSTRACT	xv
CAPÍTULO I	1
INTRODUCCIÓN	1
1.1.Antecedentes	1
1.2.Justificación e importancia	2
1.3.Alcance del proyecto.....	3
1.4.Objetivos	3
1.4.1.Objetivo general	3
1.4.2.Objetivos específicos	3
CAPÍTULO II	5
REPARACIÓN Y MODELAMIENTO DEL SISTEMA DE LEVITACIÓN MAGNÉTICA	5
2.1.Descripción del sistema reparado	5
2.1.1.Fuente de alimentación.	5
2.1.2.Tarjeta STM32F407	6
2.1.3.Circuito de potencia	8
2.1.4.Sensor de posición	9
2.1.5.Sensor de corriente.....	9

2.2.Modelamiento del sistema	11
2.2.1.Caracterización del sensor de posición	12
2.2.2.Respuesta estática del actuador	14
2.2.3.Control mínimo	16
2.2.4.Respuesta dinámica del actuador	17
2.2.5.Modelo continuo lineal	21
CAPÍTULO III.....	24
MARCO TEÓRICO	24
3.1.Algoritmos genéticos	24
3.1.1.Población.....	25
3.1.2.Función objetivo	26
3.1.3.Selección	27
3.1.4.Cruce.....	29
3.1.5.Mutación	30
3.1.6.Reducción	31
3.2.Algoritmo de Nealder-Mead	32
3.2.1.Procedimiento	33
3.3.Control difuso	36
3.3.1.Fuzzificación.....	37
3.3.2.Evaluación de reglas	38
3.3.3.Defuzzificación	39
3.3.4.Controlador PD difuso	40
CAPÍTULO IV	42
DISEÑO Y SIMULACIÓN DE ALGORITMOS DE CONTROL.....	42
4.1.Diseño del controlador fuzzy PD	42
4.1.1.Variables lingüísticas y valores lingüísticos	42
4.1.2.Rango de las variables lingüísticas	43
4.1.3.Evaluación de reglas	44
4.1.4.Simulación del sistema de levitación magnética con controlador PD difuso ...	46

4.2.Diseño del algoritmo genético	49
4.2.1.Población inicial y codificación de individuos	50
4.2.2.Función objetivo	51
4.2.3.Selección	51
4.2.4.Cruce.....	52
4.2.5.Mutación	52
4.2.6.Simulación del sistema de levitación magnética con controlador PD difuso optimizado mediante algoritmos genéticos.....	53
4.3.Simulación del sistema de levitación magnética con controlador PD difuso optimizado mediante el algoritmo de Nealder-Mead.....	58
CAPÍTULO V.....	62
IMPLEMENTACIÓN DE LOS ALGORITMOS DE CONTROL EN EL SISTEMA FÍSICO.....	62
5.1.Implementación del controlador fuzzy PD en el sistema de levitación magnética.....	62
5.2.Implementación del controlador PD difuso en el sistema de levitación magnética, optimizado mediante algoritmos genéticos.....	65
5.3.Implementación del controlador PD difuso en el sistema de levitación magnética, optimizado mediante el algoritmo de Nealder-Mead.....	66
CAPÍTULO VI.....	69
CONCLUSIONES Y RECOMENDACIONES.....	69
6.1.Conclusiones	69
6.2.Recomendaciones	70
BIBLIOGRAFÍA.....	71

ÍNDICE DE TABLAS

Tabla 1 Características técnicas fuente FP6012.....	5
Tabla 2 Descripción de las entradas/salidas utilizadas	8
Tabla 3 Características eléctricas driver L298N	8
Tabla 4 Características eléctricas del sensorACS712	9
Tabla 5 Constantes de modelamiento	12
Tabla 6 Relación Voltaje-Distancia sensor fotoeléctrico.....	13
Tabla 7 Control mínimo.....	17
Tabla 8 Respuesta dinámica del actuador	18
Tabla 9 Valores de las constantes del sistema de levitación.....	20
Tabla 10 Rangos de variables lingüísticas	43
Tabla 11 Base de reglas	45
Tabla 12 Cadena de cromosoma de cada individuo.....	51

ÍNDICE DE FIGURAS

Figura 1. Representación de las partes funcionales del Equipo de Levitación Magnética MLS	5
Figura 2. Fuente de alimentación FP6012	6
Figura 3. Tarjeta STM32F4	7
Figura 4. Diagrama de bloques tarjeta STM32F4.....	7
Figura 5. Driver L298N	8
Figura 6. Sensor de corriente ACS712	9
Figura 7. Salida de voltaje vs corriente del sensor ACS712.....	10
Figura 8. Diagrama Sistema de Levitación Magnética rehabilitado.....	10
Figura 9. Sistema de Levitación Magnética rehabilitado.....	11
Figura 10. Señal del sensor vs posición de la esfera.....	14
Figura 11. Programa para el experimento respuesta estática del actuador	15
Figura 12. Corriente vs Duty Cycle	15
Figura 13. Programa para el experimento de control mínimo	16
Figura 14. Programa para el experimento respuesta dinámica del actuador.....	18
Figura 15. Programa de procesamiento de datos de corriente y duty cycle.....	18
Figura 16. Diagrama general de un algoritmo genético.....	25
Figura 17. Método de selección muestreo universal estocástico	28
Figura 18. Individuo visto como un disco	29
Figura 19. Operador basado en dos puntos	29
Figura 20. Operador de cruce uniforme	30
Figura 21. Máscaras de cruce para operadores con 1 y 2 puntos.....	30
Figura 22. Replicación y expansión en el algoritmo de Nealder-Mead.....	34
Figura 23. Contracción interna y externa.....	35
Figura 24. Encogimiento.....	36
Figura 25. Arquitectura de un control fuzzy	37
Figura 26. Fase de Fuzzificación	38
Figura 27. Fase de evaluación de reglas	39
Figura 28. Fase de Defuzzificación.....	40
Figura 29. Controlador difuso PD.....	41
Figura 30. Funciones de membrecía del error	43
Figura 31. Funciones de membrecía de la derivada del error	44

Figura 32. Funciones de membresía de la señal de control	44
Figura 33. Editor de reglas del FIS Editor	46
Figura 34. Simulación del sistema de levitación con control difuso	47
Figura 35. Set Point y salida del sistema	48
Figura 36. Señal de control	48
Figura 37. Error del sistema.....	49
Figura 38. Descripción de la programación del algoritmo.....	50
Figura 39. Cruce en un punto.....	52
Figura 40. Mutación de bit.....	53
Figura 41. Simulación del sistema de levitación optimizado mediante algoritmos genéticos	53
Figura 42. Desempeño vs generaciones	54
Figura 43. Set Point y salida del sistema 10 generaciones.....	55
Figura 44. Señal de control con 10 generaciones.....	55
Figura 45. Desempeño vs generaciones	56
Figura 46. Set Point y salida del sistema 20 generaciones.....	56
Figura 47. Señal de control 20 generaciones	57
Figura 48. Error del sistema.....	57
Figura 49. Simulación del sistema de levitación optimizado mediante Nealder- Mead	58
Figura 50. Set Point y salida del sistema con 10 seg. de simulación.....	59
Figura 51. Señal de control con 10 seg. de simulación.....	59
Figura 52. Set Point y salida del sistema con 4 seg. de simulación.....	60
Figura 53. Señal de control con 4 seg. de simulación.....	61
Figura 54. Error del sistema.....	61
Figura 55. Programa implementado en el sistema de levitación magnética real	62
Figura 56. Programa de recepción de datos	63
Figura 57. Salida del sistema de levitación real.....	64
Figura 58. Señal de control	64
Figura 59. Programa implementado en el sistema de levitación magnética optimizado por algoritmos genéticos	65
Figura 60. Salida del sistema de levitación real.....	65
Figura 61. Señal de control	66

Figura 62. Programa implementado en el sistema de levitación magnética optimizado por el algoritmo de Nealder-Mead	66
Figura 63. Salida del sistema de levitación real	67
Figura 64. Señal de control	67
Figura 65. Sistema físico con esfera levitando	68

RESUMEN

El proyecto titulado “Rehabilitación del Sistema de Levitación Magnética MLS y Diseño e Implementación de Algoritmos de Control Avanzado”, tiene por objeto rehabilitar el sistema de levitación magnética que posee del laboratorio de servomecanismos, para después diseñar e implementar algoritmos de control avanzados. Se implementará control difuso, cuyo comportamiento va a ser optimizado fuera de línea, mediante el uso de algoritmos genéticos y el algoritmo de Nealder-Mead. Después se simulará el sistema a través del desarrollo de un programa realizado en Matlab. En primer lugar, se efectúa el análisis de todas las partes funcionales del sistema, para después refaccionar las que no lo estén, y de esta manera poner de nuevo en funcionamiento el sistema. Se realiza el modelamiento matemático del sistema de levitación, con este fin se desarrolla toda la programación necesaria, ya que requiere una reestructuración global al ser un sistema totalmente diferente. Se efectúa el diseño del controlador difuso, del método basado en algoritmos genéticos y del algoritmo de Nealder-Mead. En base al modelo obtenido se realiza la simulación de los algoritmos diseñados, luego de lo cual, con los resultados obtenidos de cada una de las simulaciones, se implementará en el sistema físico. Para la implementación en el sistema de levitación real, se realiza la programación del controlador difuso, y de las comunicaciones necesarias para la transmisión de datos, con las ganancias del controlador obtenidas en la simulación de cada algoritmo de optimización implementado. Para el análisis de los resultados obtenidos, se analiza el error del sistema, derivado de la cada uno de los algoritmos realizados.

PALABRAS CLAVE:

- **MODELO**
- **DIFUSO**
- **ALGORITMO**
- **GENÉTICO**
- **OPTIMIZACIÓN**

ABSTRACT

The project entitled “Rehabilitation of the MLS Magnetic Levitation System and Design and Implementation of Advanced Control Algorithms”, aims to rehabilitate the magnetic levitation system of the servomechanism laboratory, and then design and implement advanced control algorithms. Fuzzy control will be implemented, whose behavior will be optimized offline, using genetic algorithms and the Neelder-Mead algorithm. Then the system will be simulated through the development of a program made in Matlab. First, the analysis of all the functional parts of the system is performed, then the parts that are not functional are replaced, and the system is put back into operation. The mathematical modeling of the levitation system is performed, for this purpose all the necessary programming is developed, since it requires a global restructuring as it is a totally different system. The design of the fuzzy controller, the method based on genetic algorithms and the Neelder-Mead algorithm is performed. Based on the model obtained, the simulation of the designed algorithms is performed, after which, with the results obtained from each of the simulations, it will be implemented in the physical system. For the implementation in the real levitation system, the programming of the fuzzy controller, and of the communications necessary for the transmission of data, is done with the gains of the controller obtained in the simulation of each optimization algorithm implemented. For the analysis of the reached results, we analyze the system error, derived from each one of the performed algorithms.

KEYWORDS:

- **MODEL**
- **FUZZY**
- **ALGORITHM**
- **GENETIC**
- **OPTIMIZATION**

CAPÍTULO I

INTRODUCCIÓN

Con el fin de cooperar con el proceso de aprendizaje del alumnado de la universidad se presenta el siguiente trabajo, en el cual se realiza la reparación del Sistema de Levitación Magnética MLS, partiendo de un análisis de las diferentes partes averiadas y de la posterior refacción o reparación en caso de ser necesario. Además, se implementa el diseño de técnicas de control avanzado para corroborar el correcto funcionamiento del sistema. Este equipo permitirá reforzar el estudio del área de control, parte primordial en la preparación de los futuros ingenieros de la carrera de Ingeniería en Electrónica, Automatización y Control.

1.1. Antecedentes

Los laboratorios constituyen una parte primordial dentro de la formación académica de todos los profesionales que se educan en la Universidad de las Fuerzas Armadas-ESPE, en este caso se plantea trabajar en el laboratorio de Servomecanismos del DECEM. Este laboratorio de Servomecanismos cuenta con plantas para implementar una serie de algoritmos de control automático, y dispone de los siguientes equipos:

- Frenos abs.
- Helicóptero.
- Sistema multitanques.
- Grúa tridimensional.
- Suspensión magnetoreológica.
- Péndulo invertido.
- Sistema servo.
- Levitador magnético.

El proyecto se realizará sobre el equipo de levitación magnética MLS. Este sistema cuyas siglas en inglés significan “Magnetic Levitation System”, traducido al español; Sistema de Levitación Magnética. Dicho sistema consta de un electromagneto, una esfera ferromagnética, un sensor de posición, un sensor de corriente. Es un sistema no lineal, dinámico sin fricción.

El principio básico de funcionamiento del sistema MLS es aplicar tensión a un electroimán para mantener levitando una esfera ferromagnética. Por otra parte, la esfera puede seguir un valor de posición deseado que varía en el tiempo. La corriente de la bobina se mide para experimentar la identificación y estrategias de control multi lazo o no lineales. Para levitar, la esfera requiere un controlador en tiempo real.

La etapa de equilibrio de dos fuerzas (la gravitacional y electromagnética) se logra gracias al controlador para mantener la esfera a la distancia deseada del imán (INTECO, INTECO, 2014). Actualmente el sistema no se encuentra funcional, es por eso que se plantea su rehabilitación y la posterior implementación de algoritmos de control avanzado para su comprobación.

1.2. Justificación e importancia

Se debe tener presente la importancia de potenciar las habilidades de los estudiantes de ingeniería electrónica a través de trabajos de laboratorios dinámicos e interactivos con aplicación concreta del conocimiento donde se privilegie el razonamiento y el análisis por encima del trabajo mecánico. Es por tal razón que se plantea este proyecto, con el fin de contribuir al enriquecimiento del aprendizaje dentro de la universidad.

Se ha visto en la actualidad que los sistemas de control de posición son un componente importante de muchos productos industriales. Ejemplos son los controladores en robótica, procesos de control y muchos otros (Pérez & Readman). Todos estos necesarios en el bagaje de los futuros profesionales de la rama de la automatización y control.

Es por eso que se ve la relevancia de la realización de este proyecto ya que permitirá tener en operación equipos que facilitarán la puesta en práctica de los conocimientos adquiridos por el cuerpo estudiantil en cada una de las materias relacionadas con la enseñanza de diferentes tipos de control, desde el control clásico hasta el control inteligente.

Además, se puede notar que se busca una mayor difusión de técnicas de control avanzado, desde su etapa inicial de diseño hasta su posterior implementación, poniendo a disposición, de esta manera una fuente de fácil acceso para el estudio de técnicas de control avanzado.

1.3. Alcance del proyecto

El presente proyecto, plantea la rehabilitación y puesta en marcha del Sistema de Levitación Magnética MLS averiado del que se dispone en el laboratorio de servomecanismos; además del diseño e implementación de algoritmos de control avanzado para el sistema.

Como primera etapa se contempla el análisis de todas las averías presentes en el sistema, para lo cual se realizará un estudio exhaustivo de cada uno de los componentes de hardware del levitador, revisando su funcionalidad tomando como referencia las especificaciones dadas por el fabricante. Después de lo que se procederá a comprobar el correcto funcionamiento del sistema.

La siguiente etapa del proyecto trata el estudio a profundidad de las técnicas de control a ser diseñadas e implementadas en el sistema, los algoritmos de control a ser analizados serán:

- Sintonización óptima de un controlador Fuzzy PD para el sistema de levitación magnética utilizando el Método simple de Neelder-Mead y Algoritmos genéticos.
- Posteriormente se procederá con el diseño de los algoritmos de control tratados, aquí se realizará las simulaciones necesarias con el modelo del sistema antes de entrar en la etapa de implementación. La última etapa del proyecto constará de la implementación tanto en hardware como en software de las técnicas de control diseñadas, se considerará pruebas de funcionamiento de cada algoritmo en el sistema de levitación, y la posterior entrega del sistema funcional para su uso en el laboratorio.

1.4. Objetivos

1.4.1. Objetivo general

- Rehabilitar el Sistema de Levitación Magnética MLS dispuesto en el laboratorio de servomecanismos e Implementar algoritmos de control avanzado para dicho sistema.

1.4.2. Objetivos específicos

- Identificar las averías existentes en el sistema MLS, y efectuar las correcciones necesarias para su adecuado funcionamiento.

- Diseñar un controlador fuzzy PD para el sistema de levitación magnética.
- Diseñar en forma óptima un controlador fuzzy PD para el levitador magnético mediante el método simple de Nelder-Meade.
- Diseñar en forma óptima un controlador fuzzy PD para el levitador magnético mediante algoritmos genéticos.
- Implementar los controladores diseñados sobre el sistema de levitación magnética.

CAPÍTULO II

REPARACIÓN Y MODELAMIENTO DEL SISTEMA DE LEVITACIÓN MAGNÉTICA

2.1. Descripción del sistema reparado

En condiciones originales el sistema constaba de cinco ítems principales que conformaban el hardware del equipo; fuente de alimentación, la tarjeta de interfaz RTDAC4/PCI, un circuito de potencia, un electroimán (actuador) y sensores de posición (sensor óptico) y de corriente. Como parte de la reparación se procedió a reemplazar la tarjeta de interfaz, el circuito de potencia y el sensor de corriente, además de sustituir la fuente de alimentación del sistema, esto después de un análisis de funcionalidad de los distintos componentes. Los componentes que se encuentran operativos y que se utilizaron para la implementación del sistema renovado son los que se muestran en la figura 1.

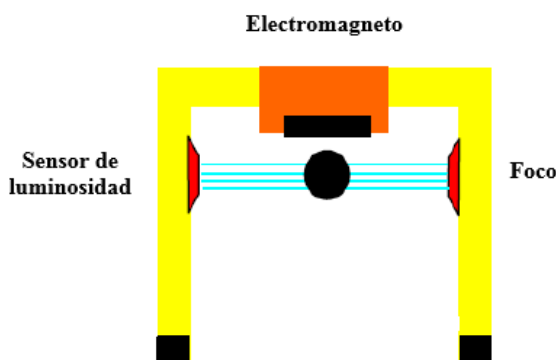


Figura 1. Representación de las partes funcionales del Equipo de Levitación Magnética MLS

2.1.1. Fuente de alimentación.

La etapa de alimentación ha sido suplida, en base a las necesidades del sistema, con una fuente modelo FP6012 mostrada en la figura 2, con las siguientes características:

Tabla 1
Características técnicas fuente FP6012

Voltaje de salida	12 Vcc 5A
--------------------------	-----------

Continua

Rango de voltaje	85-264 VAC
Rango de ajuste de salida	$\pm 10\%$ de tensión nominal de salida
Regulación de línea	$< 0.5\%$
Salida con protección contra sobrecarga	110%-150% modo de hipo, auto-recuperación
Salida con protección a sobre voltaje	115%-135% de tensión nominal de salida
Normas de seguridad	NOM, ANCE, NYCE



Figura 2. Fuente de alimentación FP6012

2.1.2. Tarjeta STM32F407

Ya que la tarjeta de interfaz original del equipo se encontraba averiada, esta fue reemplazada con la tarjeta STM32F407 de STMicroelectronics mostrada en la figura 3. El principal parámetro que se tomó en cuenta para la elección de este componente fue la frecuencia mínima necesaria de pwm para provocar la atracción suficiente de la esfera; que después de la experimentación realizada con el sistema reparado fue de 40 KHz, fácilmente proporcionado por la tarjeta seleccionada. Además de las facilidades de programación que presenta ya que básicamente se tiene que programar desde cero todos los procedimientos, tanto para el modelamiento como para la implementación de los algoritmos de control. En la figura 4. se observa el diagrama de bloques de la tarjeta STM32F407.



Figura 3. Tarjeta STM32F4

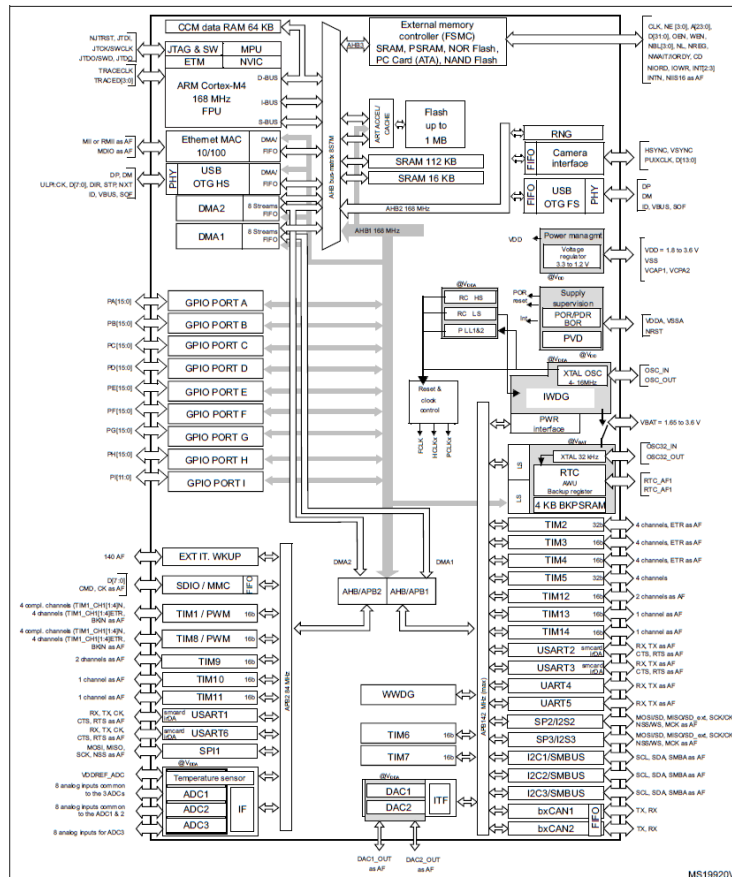


Figura 4. Diagrama de bloques tarjeta STM32F4

Las entradas y salidas que se utilizarán se describen en la tabla 2.

Tabla 2
Descripción de las entradas/salidas utilizadas

Entrada/salida	Canal
Entrada sensor de posición	PA4
Entrada sensor de corriente	PA5
Salida PWM	PA8

2.1.3. Circuito de potencia

Para el circuito de potencia (pre-actuador) se ha dispuesto de un driver L298N (figura 5), el mismo que se usa para acoplar la señal de control con el actuador, gracias a que permite variar la corriente suministrada al electroimán encargado de la atracción de la esfera mediante la variación del duty cycle del pwm proveniente de la tarjeta de interfaz. Sus características eléctricas se muestran en la tabla 3.



Figura 5. Driver L298N

Tabla 3
Características eléctricas driver L298N

Parámetro	Típico	Máx.	Unidades
Retardo de apagado de corriente de suministro	1.5	-	μs
Tiempo de caída de corriente de suministro	0.2	-	μs
Retardo de encendido de corriente de suministro	2	-	μs
Tiempo de subida de corriente de suministro	0.7	-	μs
Retardo de apagado de corriente de entrada	0.7	-	μs
Tiempo de caída de corriente de entrada	0.25	-	μs

Continua 

Retardo de encendido de corriente de entrada	1.6	-	μs
Tiempo de subida de corriente de entrada	0.2	-	μs
Frecuencia de conmutación	25	40	KHz

2.1.4. Sensor de posición

Para medir la posición se utiliza un sensor fotoeléctrico, ya que este nos permite efectuar mediciones en el rango de los milímetros dado su sensibilidad, en este caso se hace uso del sensor original ya que este no ha sufrido averías. El quipo cuenta con un foco que proporciona luz dirigida al sensor, gracias a un prisma que se encuentra empotrado por delante del foco, como se muestra en la figura 1.

2.1.5. Sensor de corriente


En la medición de corriente se utiliza un sensor ACS712, el mismo que se visualiza en la figura 6 y cuyas características eléctricas se muestran en la tabla 4. Se escogió este sensor ya que la corriente máxima con la que trabaja el actuador del sistema de levitación es de 3 amperios este sensor nos permite medir hasta 5 amperios.



Figura 6. Sensor de corriente ACS712

Tabla 4
Características eléctricas del sensor ACS712

Parámetro	Min.	Típico	Máx.	Unidades
Suministro de voltaje	4.5	5.0	5.5	V
Suministro de corriente	-	10	13	mA
Carga de capacitancia de salida	-	-	10	nF
Carga resistiva de salida	4.7	-	-	k Ω
Resistencia de conductor primario	-	1.2	-	m Ω
Tiempo de subida	-	3.5	-	μs

Continua 

Ancho de banda de frecuencia	-	80	-	KHz
No linealidad	-	1.5	-	%
Simetría	98	100	102	%
Tiempo de encendido	-	35	-	μ s

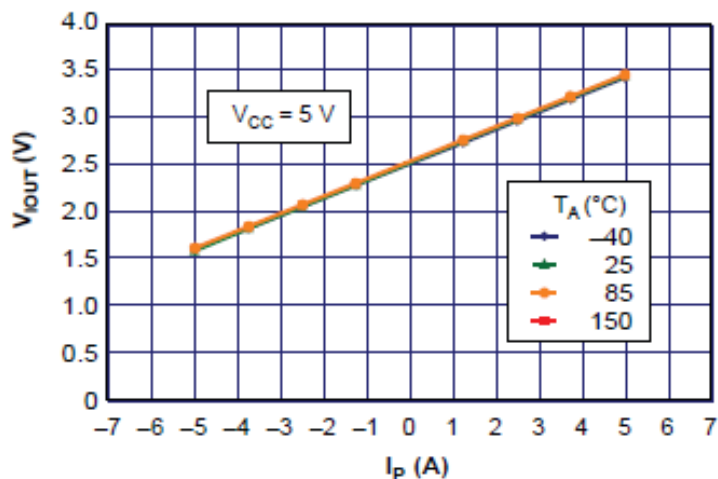


Figura 7. Salida de voltaje vs corriente del sensor ACS712

En la figura 7 se visualiza la respuesta del sensor de corriente, la salida de voltaje dependiente de la corriente. Se puede observar en la figura 8 un diagrama del sistema de levitación magnética ya rehabilitado. En la figura 9 se visualiza el sistema de levitación magnética ya rehabilitado.

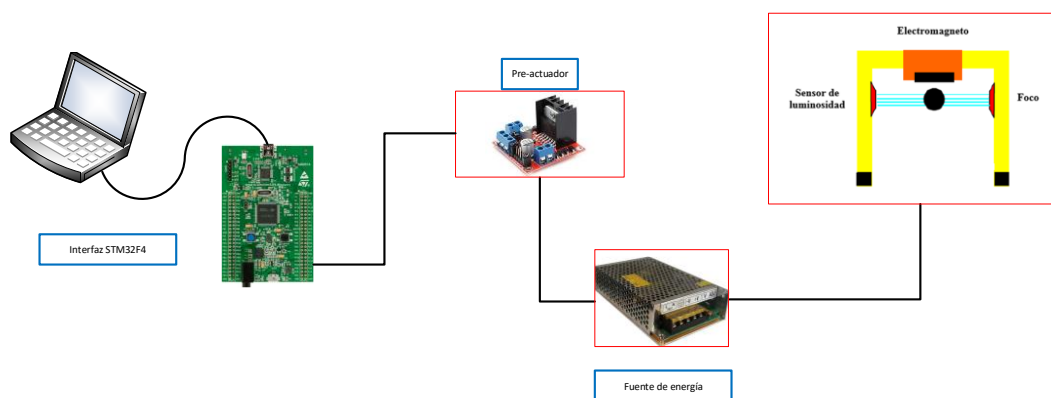


Figura 8. Diagrama Sistema de Levitación Magnética rehabilitado

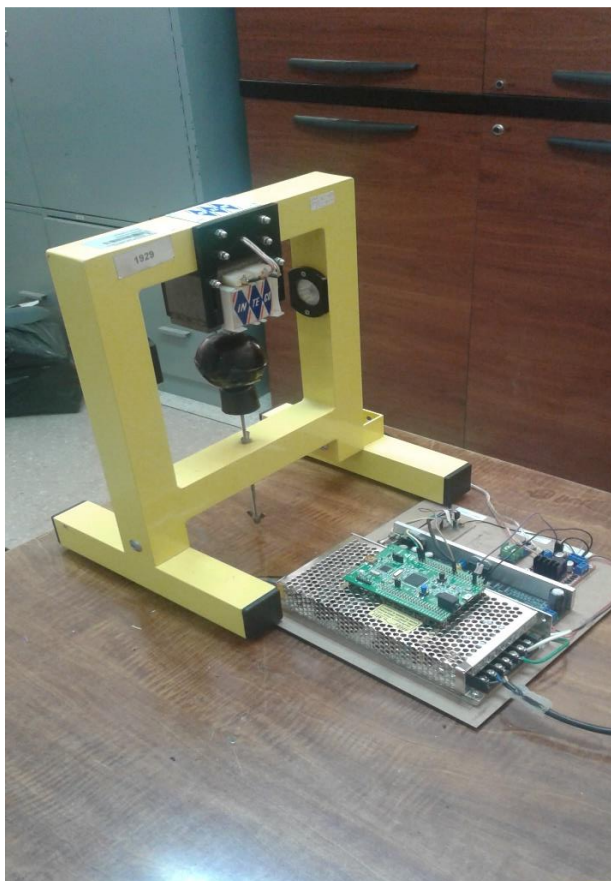


Figura 9. Sistema de Levitación Magnética rehabilitado

2.2. Modelamiento del sistema

El modelamiento del sistema de levitación magnética MLS consiste en hallar una serie de constantes que permiten llegar al modelo matemático, a través de cuatro experimentos para posteriormente definir la función de transferencia, mediante las siguientes ecuaciones:

$$\dot{x}_1 = x_2 \quad Ec. (1.1)$$

$$\dot{x}_2 = -\frac{F_{em}}{m} + g \quad Ec. (1.2)$$

$$\dot{x}_3 = \frac{1}{f_i(x_1)} (k_i u + c_i - x_3) \quad Ec. (1.3)$$

$$F_{em} = x_3^2 \frac{F_{emP1}}{F_{emP2}} \exp\left(-\frac{x_1}{F_{emP2}}\right) \quad Ec. (1.4)$$

$$f_i(x_1) = \frac{f_{iP1}}{f_{iP2}} \exp\left(-\frac{x_1}{f_{iP2}}\right) \quad \text{Ec. (1.5)}$$

Donde x_1 corresponde a la posición de la esfera, x_2 a la velocidad de movimiento de la esfera y x_3 es la corriente que fluye por el electroimán. Las distintas constantes y parámetros necesarios para la definición del modelo matemático del sistema se muestran en la tabla 5.

Tabla 5
Constantes de modelamiento

Parámetro	Definición
m	Masa de la esfera
g	Gravedad
F_{em}	Fuerza electromagnética
F_{emP1}	Constante de la ecuación de fuerza electromagnética 1.4
F_{emP2}	Constante de la ecuación de fuerza electromagnética 1.4
$f_i(x_1)$	Inductancia función de x_1
f_{iP1}	Constante de la ecuación de inductancia 1.5
f_{iP2}	Constante de la ecuación de inductancia 1.5
c_i	Constante de característica estática
k_i	Constante de característica estática
x_{3MIN}	Corriente mínima del actuador
u_{MIN}	Señal de control mínima (duty cycle)

2.2.1. Caracterización del sensor de posición

Para determinar la respuesta del sensor de posición se realiza un experimento que consiste en ir modificando la posición de la esfera en forma manual, girando el tornillo que sostiene la base sobre la que se coloca la esfera; cada vuelta del tornillo corresponde a 0.7 milímetros, se toman las medidas de voltaje obtenidas de la salida del acondicionamiento del sensor cada media vuelta de tornillo y se las muestra en la tabla 6.

Tabla 6
Relación Voltaje-Distancia sensor fotoeléctrico

Distancia (m)	Voltaje (V)	Distancia (m)	Voltaje (V)
0	3.22	0.00910	1.77
0.00035	3.22	0.00945	1.63
0.00070	3.22	0.00980	1.59
0.00105	3.21	0.01015	1.47
0.00140	3.21	0.01050	1.43
0.00175	3.21	0.01085	1.32
0.00210	3.21	0.01120	1.28
0.00245	3.20	0.01155	1.18
0.00280	3.19	0.01190	1.15
0.00315	3.18	0.01225	1.06
0.00350	3.14	0.01260	1.04
0.00385	3.10	0.01295	0.96
0.00420	3.05	0.01330	0.93
0.00455	2.98	0.01365	0.87
0.00490	2.92	0.01400	0.85
0.00525	2.82	0.01435	0.80
0.00560	2.75	0.01470	0.78
0.00595	2.63	0.01505	0.74
0.00630	2.57	0.01540	0.72
0.00665	2.43	0.01575	0.69
0.00700	2.36	0.01610	0.67
0.00735	2.22	0.01645	0.65
0.00770	2.16	0.01680	0.63
0.00805	2.02	0.01715	0.61
0.00840	1.96	0.01750	0.61
0.00875	1.82	0.01785	0.60

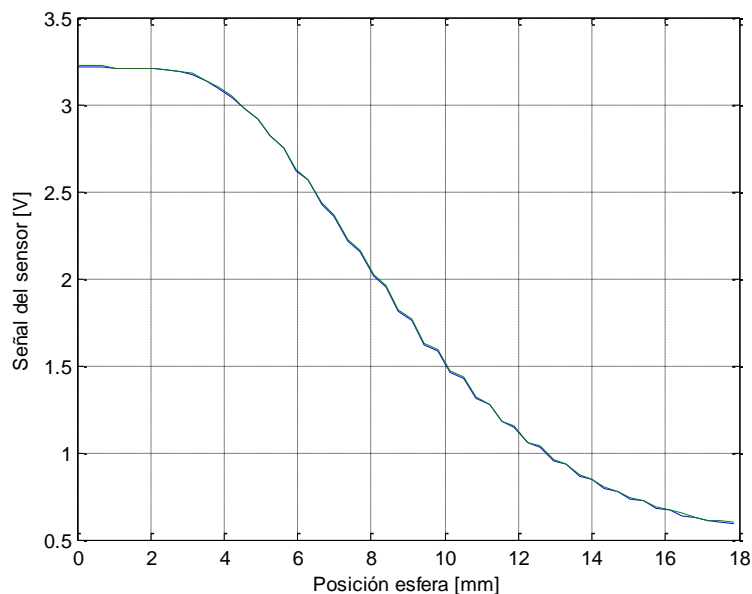


Figura 10. Señal del sensor vs posición de la esfera

Para poder utilizar los datos proporcionados por el sensor se usa la herramienta Lookup Table, en donde se ingresan los datos de voltaje y posición obtenidos de las mediciones realizadas, las cuales se pueden observar en la figura 10, y se efectúa una aproximación a una curva con el método de interpolación “Cubic splir” y el método de extrapolación “Clip”, con lo que se obtiene la respuesta del sensor.

2.2.2. Respuesta estática del actuador

Para la ejecución de todos los programas necesarios para efectuar las pruebas requeridas en el modelamiento del sistema de levitación se echó mano de las librerías de MATLAB denominadas Waijung Blockset creadas por AIMAGIN CO., LTD.

Gracias a este experimento se puede conocer las características estáticas del actuador. Se realiza sin la presencia de la esfera. Básicamente el objetivo es hallar la relación existente entre la señal de entrada al pre-actuador y la corriente que fluye a través del actuador (electroimán) para tener una noción de la capacidad que posee el electroimán de atraer a la esfera con los cambios realizados en el hardware, mencionados anteriormente.

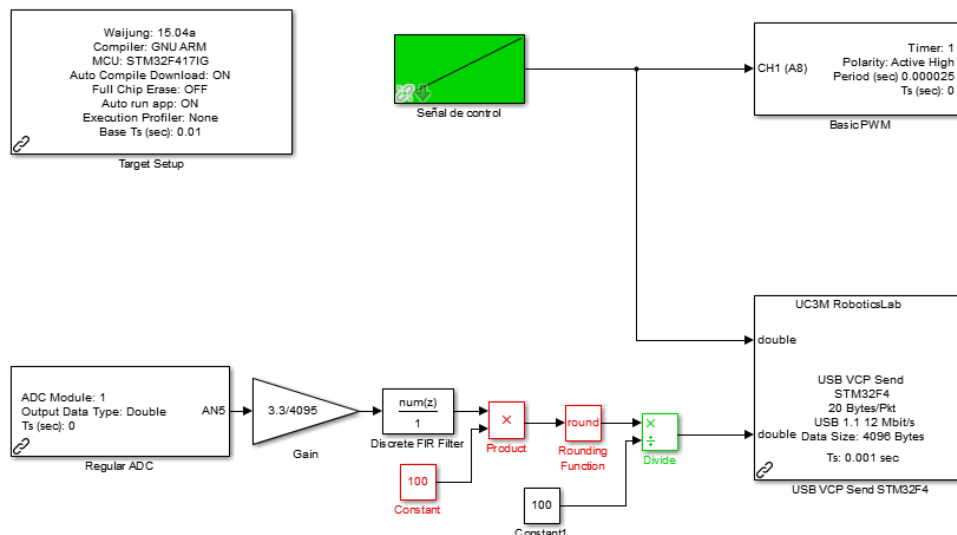


Figura 11. Programa para el experimento respuesta estática del actuador

Mediante el programa de Simulink mostrado en la figura 11 se envía una señal de entrada tipo rampa con una variación de 0.1% del duty cycle cada milisegundo desde 0 hasta 100%, y se adquiere la señal del sensor de corriente, datos con los cuales se obtiene la gráfica 12. de corriente en función del duty cycle.

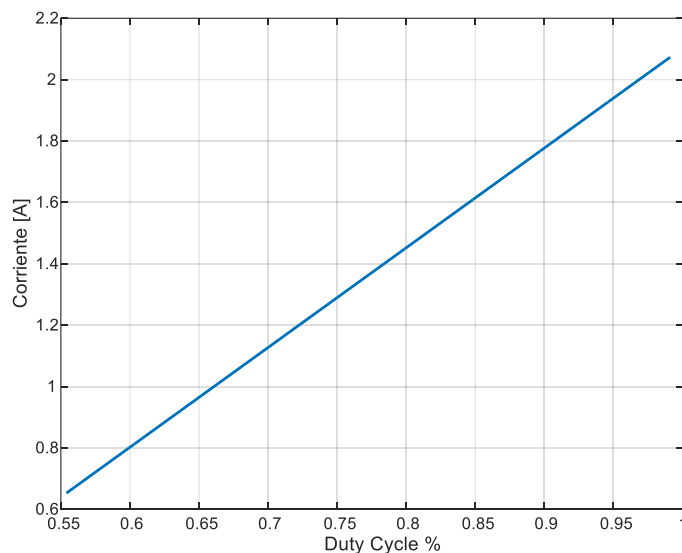


Figura 12. Corriente vs Duty Cycle

Con la relación del duty cycle vs corriente visualizada en la figura 12 se obtiene la ecuación lineal correspondiente, para determinar los parámetros k_i y c_i , además se identifican los parámetros x_{3MIN} y u_{MIN} .

$$i(u) = k_i u + c_i \quad Ec. (1.6)$$

$$k_i = 3.2499$$

$$c_i = -1.1476$$

$$x_{3MIN} = 0.6203 [A]$$

$$u_{MIN} = 0.544$$

2.2.3. Control mínimo

En este experimento se examina el control mínimo necesario para provocar un movimiento forzado de la esfera desde la base situada en la parte inferior hacia el electroimán, el objetivo no es que la esfera levite sino solamente su atracción por la acción del campo electromagnético. Se mide el porcentaje del duty cycle y por lo tanto la corriente en determinada posición mediante el programa estructurado en la figura 13.

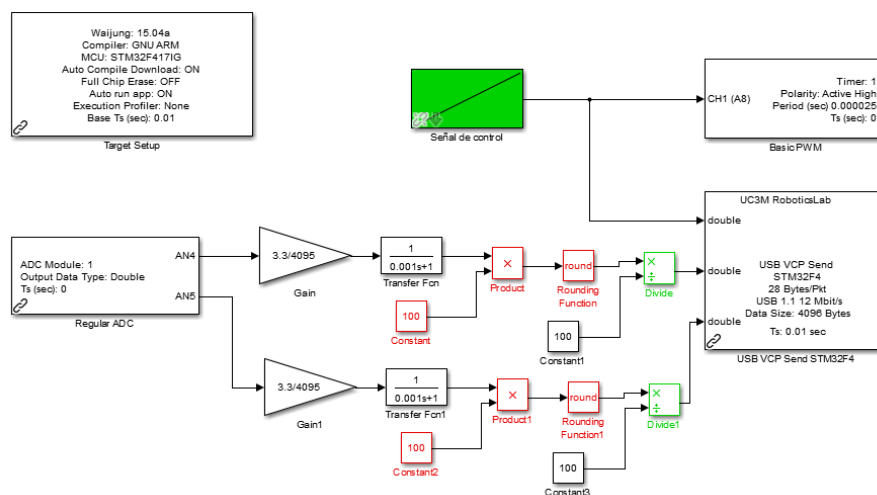


Figura 13. Programa para el experimento de control mínimo

Para poder determinar la corriente necesaria para atraer la esfera en determinada posición se procede a colocar a la esfera a cuatro vueltas de distancia del electroimán y se ejecuta el programa descrito para tomar las mediciones, después se vuelve a

ejecutar el programa cada dos vueltas; los valores de posición, duty cycle y corriente obtenidos del experimento se visualizan en la tabla 7.

Tabla 7
Control mínimo

Vueltas	Posición(mm)	Control mínimo(duty cycle)	Control Mínimo(A)
4	2,8	0,623	0,8770877
6	4,2	0,654	0,9778346
8	5,6	0,679	1,0590821
10	7	0,706	1,1468294
12	8,4	0,732	1,2313268
14	9,8	0,758	1,3158242
16	11,2	0,788	1,4133212
18	12,6	0,812	1,4913188
20	14	0,856	1,6343144
22	15,4	0,9	1,77731
24	16,8	0,928	1,8683072
26	17,5	0,976	2,0243024

Las constantes obtenidas del experimento a través de una optimización de la ecuación 1.4 son F_{emP1} y F_{emP2} .

$$F_{emP1} = 0.05316$$

$$F_{emP2} = 0.4777$$

2.2.4. Respuesta dinámica del actuador

El objeto de este experimento es analizar las características dinámicas del actuador, es decir la fuerza electromotriz generada por el movimiento de la esfera. El experimento consiste en forzar atracción sobre la esfera y adquirir la corriente que se genera con esta acción, enviando una señal de control variable como se observa en el programa de la figura 14.

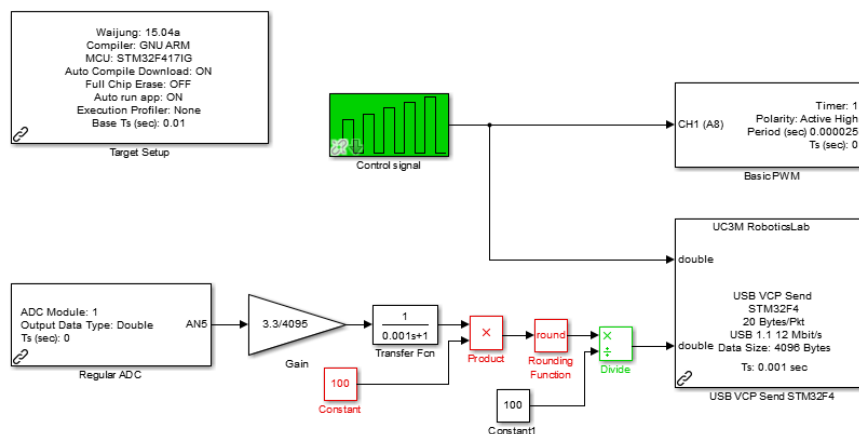


Figura 14. Programa para el experimento respuesta dinámica del actuador

Los datos adquiridos de corriente y duty cycle se procesan en el programa de la figura 15, dados los valores obtenidos de k_i y c_i del segundo experimento. Se realiza este proceso para diferentes distancias de la esfera, con lo que se extraen los valores de k_i y de f_i de la ecuación 1.5, valores mostrados en la tabla 8.

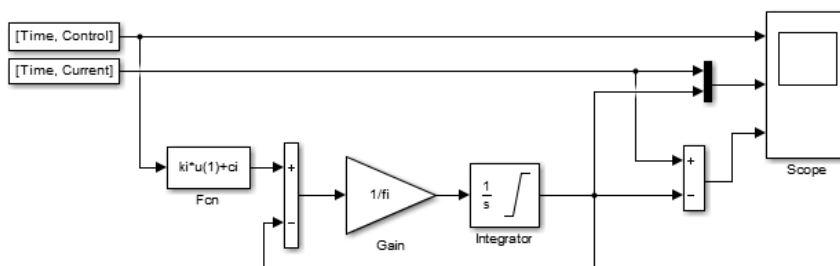


Figura 15. Programa de procesamiento de datos de corriente y duty cycle

**Tabla 8
Respuesta dinámica del actuador**

Posición(mm)	PWM(%)	Corriente(A)	k_i	f_i
2,8	0,6	0,7906	3,7865	0,0685
	0,7	1,0560		
	0,8	1,3510		
	0,9	1,6950		
	1	1,9850		
3,5	0,6	0,7912	3,8054	0,0754
	0,7	1,0620		
	0,8	1,3580		
	0,9	1,7050		
	1	1,9950		

Continua

	0,6	0,7925		
	0,7	1,0710		
	0,8	1,3720		
	0,9	1,7190		
4,2	1	2,0010	3,8349	0,0753
	0,6	0,7938		
	0,7	1,0800		
	0,8	1,3790		
	0,9	1,7250		
4,9	1	2,0120	3,8073	0,0783
	0,6	0,7945		
	0,7	1,0850		
	0,8	1,3850		
	0,9	1,7340		
5,6	1	2,0230	3,7850	0,0759
	0,6	0,7959		
	0,7	1,0920		
	0,8	1,3970		
	0,9	1,7420		
6,3	1	2,0330	3,7829	0,0750
	0,6	0,7963		
	0,7	1,0960		
	0,8	1,4050		
	0,9	1,7490		
7	1	2,0420	3,7877	0,0753
	0,6	0,7982		
	0,7	1,1010		
	0,8	1,4170		
	0,9	1,7540		
7,7	1	2,0510	3,7820	0,0734
	0,6	0,7991		
	0,7	1,1060		
	0,8	1,4380		
	0,9	1,7620		
8,4	1	2,0580	3,8211	0,0734
	0,6	0,7990		
	0,7	1,1150		
	0,8	1,4420		
	0,9	1,7710		
9,1	1	2,0640	3,8363	0,0730

Continua 

	0,6	0,8001		
	0,7	1,1190		
	0,8	1,4480		
	0,9	1,7750		
9,8	1	2,0710	3,8062	0,0727
	0,6	0,8011		
	0,7	1,1210		
	0,8	1,4510		
	0,9	1,7800		
10,5	1	2,0730	3,9598	0,0869
	0,6	0,8023		
	0,7	1,1270		
	0,8	1,4520		
	0,9	1,7840		
11,2	1	2,0750	3,7780	0,0748

Partiendo del experimento se hallan las constantes f_{iP1} y f_{iP2} mediante optimización por el método de Nealder-Mead con la herramienta `fminsearch`, de la ecuación 1.5.

$$f_{iP1} = 0.02307$$

$$f_{iP2} = 0.2534$$

Los valores obtenidos de las distintas constantes necesarias para el modelamiento de sistema se muestran en la tabla 9.

Tabla 9
Valores de las constantes del sistema de levitación

Parámetro	Valor	Unidades
m	0.04481	[Kg]
g	9.81	$\left[\frac{m}{s^2}\right]$
F_{em}	0.4396	[N]
F_{emP1}	0.05316	[H]
F_{emP2}	0.4777	[m]
f_{iP1}	0.02307	[m * s]
f_{iP2}	0.2534	[m]

Continua 

c_i	-1.1476	[A]
k_i	3.2499	[A]
x_{3MIN}	0.6203	[A]
u_{MIN}	0.544	%

2.2.5. Modelo continuo lineal

El sistema de levitación magnética (MLS) es no lineal, es así que se requiere aproximar a un modelo lineal en torno a un punto de equilibrio. Con este fin, se describe el levitador magnético en el espacio de estados con las siguientes ecuaciones:

$$\dot{x} = Ax + Bu \quad Ec. (1.7)$$

$$y = Cx \quad Ec. (1.8)$$

Partiendo de las ecuaciones 1.1, 1.2 y 1.3; se las define de la siguiente forma:

$$f_1 = \dot{x}_1 = x_2$$

$$f_2 = \dot{x}_2 = -\frac{F_{em}}{m} + g$$

$$f_3 = \dot{x}_3 = \frac{1}{f_i(x_1)} (k_i u + c_i - x_3)$$

Las matrices A y B se determinan mediante las siguientes aproximaciones:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix} \quad Ec. (1.9)$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} \\ \frac{\partial f_2}{\partial x_1} \\ \frac{\partial f_3}{\partial x_1} \end{bmatrix} \quad \text{Ec. (1.10)}$$

Se reduce las matrices a:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ a_{21} & 0 & a_{23} \\ a_{31} & 0 & a_{33} \end{bmatrix}$$

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

$$C = [1 \quad 0 \quad 0]$$

Calculando las derivadas parciales se obtiene:

$$a_{21} = \frac{x_3^2 F_{emP1}}{m F_{emP2}} \exp\left(-\frac{x_1}{F_{emP2}}\right)$$

$$a_{23} = -\frac{2x_3 F_{emP1}}{m F_{emP2}} \exp\left(-\frac{x_1}{F_{emP2}}\right)$$

$$a_{31} = (k_i u + c_i - x_3) \left(\frac{f_{iP1}}{f_{iP2}}\right) \exp\left(-\frac{x_1}{f_{iP2}}\right)$$

$$a_{33} = -\frac{x_1}{f_i(x_1)}$$

$$b_3 = \frac{k_i x_1}{f_i(x_1)}$$

El punto de equilibrio se escoge en la distancia 8mm, donde la velocidad obtenida de la ecuación 1.1 es cero. Partiendo de las ecuaciones 1.2 y 1.3 se logran obtener la corriente y el porcentaje del ciclo de trabajo. El punto de equilibrio es:

$$P_e = [x_1 \quad x_2 \quad x_3] = [0.008 \quad 0 \quad 0.6802]$$

$$u = 0.5624$$

Las matrices calculadas son:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ a_{21} & 0 & a_{23} \\ a_{31} & 0 & a_{33} \end{bmatrix}$$

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

$$C = [1 \quad 0 \quad 0]$$

Operando con las matrices en el espacio de estados, se obtiene la siguiente función de transferencia:

$$G(s) = \frac{-122.4}{s^3 + 11.33s^2 - 1.13s - 12.8}$$

CAPÍTULO III

MARCO TEÓRICO

3.1. Algoritmos genéticos

Los algoritmos genéticos (AGs) son criterios adaptativos que suelen utilizarse para solucionar problemas de optimización. Se encuentran basados en los cambios genéticos producidos durante la evolución de los organismos vivos a lo largo de las generaciones de acuerdo con los principios de la selección natural, análogamente estos algoritmos tienen la capacidad de ir creando soluciones para los problemas planteados, donde la evolución de estas soluciones hasta lograr valores óptimos es dependiente del correcto procesamiento de las mismas (Hollan, 1975).

Los AGs resultan de una analogía directa con la conducta de los distintos organismos de la naturaleza, es decir, trabajan con una cierta población de individuos, donde cada uno implica una solución posible al problema, cada individuo es valorado con una puntuación dependiendo de los beneficios dados por dicha solución, lo que equivale a la capacidad de un ser vivo de competencia por los recursos que le son necesarios. Mientras mayor sea la capacidad de adaptación al problema, será más alta la probabilidad de que el individuo sea seleccionado para cruzar su material genético con otro organismo seleccionado de la misma manera. De este cruce se tendrán nuevos individuos que comparten ciertas características de sus padres (Hollan, 1975).

Es de esta manera que se crea una nueva población de soluciones posibles que reemplaza a la anterior, proporcionando mejores resultados en comparación con la generación anterior. A medida que las generaciones avanzan las características beneficiosas se difunden a través de la población; así se propende el cruce de los individuos mejor adaptados, siendo examinadas las áreas más convenientes del espacio de búsqueda, de modo que gracias a un algoritmo bien diseñado se obtendrá una población orientada hacia una solución óptima del problema (Hollan, 1975).

El poderío de los AGs deriva de la robustez que posee, ya que se pueden usar con éxito en una gran variedad de problemáticas procedentes de distintas áreas, en donde inclusive otros métodos tienen dificultades. A pesar de que el algoritmo genético no garantiza encontrar siempre una solución óptima, puede hallar una solución bastante aceptable. Es probable en caso de que existan técnicas específicas para resolver determinados problemas, éstas superen los resultados obtenidos mediante la aplicación

de los algoritmos genéticos. Resulta susceptible de mejora si se logra combinar las técnicas específicas con los AGs, tal como se logra en la tesis presentada.

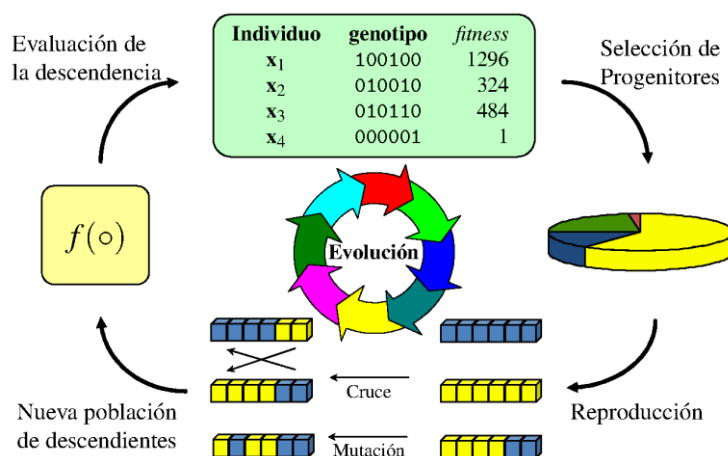


Figura 16. Diagrama general de un algoritmo genético

Fuente: (Sancho, 2016)

3.1.1. Población

3.1.1.1. Tamaño de la población

El tamaño de la población es un parámetro a considerar. Se podría intuir que si la población es pequeña puede no cubrir de una manera adecuada el espacio de búsqueda, mientras que si la población es muy grande se pueden tener limitaciones en lo que respecta al costo computacional.

En un estudio realizado por parte de (Goldberg, 1989), se concluyó que la dimensión óptima de la población para cadenas de longitud l , de codificación binaria, aumenta de manera exponencial junto con la dimensión de la cadena. Siendo así se podría pensar que no representan competencia frente a otros métodos de optimización combinatoria, sin embargo a través de evidencia empírica se sugiere que un tamaño de población entre l y $2l$ es apto para solucionar los problemas considerados (Alander, 1992).

3.1.1.2. Población inicial

Comúnmente se genera la población inicial con cadenas originadas al azar, teniendo cada gen uno de los probables valores. Se podrían obtener los individuos de la población inicial mediante técnicas de optimización local, esta inicialización no

aleatoria puede apresurar la convergencia del algoritmo genético, pero una desventaja resulta de la prematura convergencia del algoritmo a óptimos locales.

3.1.2. Función objetivo

Hay que considerar dos aspectos de gran importancia dentro de la conducta de los AGs, que son, la utilización de una adecuada función de adaptación y la codificación usada. El interés se centra en construir funciones objetivo “regulares”, esto quiere decir que, para dos individuos cercanos en el espacio de búsqueda, sus correspondientes valores en dichas funciones sean parecidos. Una dificultad que presentan los algoritmos genéticos es la aparición de óptimos locales, y el aislamiento del óptimo global.

Lo que se busca de una buena función objetivo es que ésta debe plasmar de manera real el valor del individuo, pero en muchos casos de optimización una gran parte del espacio de búsqueda presenta individuos no válidos por la existencia de varias restricciones. Para estos casos en los que los individuos están sujetos a restricciones hay varias soluciones propuestas.

La primera es de régimen *absolutista*, en donde los individuos que no verifican las restricciones no se consideran, o en su defecto se les establece una función igual a cero. Otra manera es reconstruir a éstos individuos, lo que se puede hacer a través de un operador llamado comúnmente *reparador*.

Otra posible solución se basa en la *penalización de la función de adaptación*, cuya idea consiste en dividir la función para una cantidad relacionada con las restricciones transgredidas por el individuo.

Si el procesamiento computacional de la función objetivo resulta muy complejo se puede utilizar otra técnica llamada *evaluación aproximada de la función objetivo*. En ciertos casos el uso de un número n de funciones proporciona mejores resultados que la evaluación de una sola función, siendo mínimo n veces más rápida.

El tratamiento de la función objetivo también se centra en evitar que la convergencia sea muy rápida o muy lenta. Si es muy rápida se denomina *convergencia prematura*, lo cual hace que el algoritmo converja hacia óptimos locales. Si es muy lenta se llama *convergencia lenta*. Los problemas de convergencia prematura producen individuos dominantes con adaptación muy superior al resto, dicho problema se trata a través de una transformación de la función objetivo, en este caso por medio de una

compresión del rango de la función. La convergencia lenta se corrige de la misma forma, con la diferencia de que en este caso se realiza una expansión del rango.

3.1.3. Selección

En esta operación se busca seleccionar los padres para cada generación. La llamada *función de selección proporcional a la función objetivo* es la función más utilizada, en donde la probabilidad de cada individuo de resultar seleccionado, es proporcional al valor de la función de adaptación (Brindle, 1991).

Así, la probabilidad de que el individuo sea seleccionado I_t^j viene dada por:

$$p_{j,t} = \frac{g(I_t^j)}{\sum_{j=1}^{\lambda} g(I_t^j)} \quad \text{Ec. (3.1)}$$

Esta función varía frente a un cambio de escala. Una manera de solucionar el problema mencionado anteriormente de cuando aparecen individuos dominantes, es ejecutar la *selección proporcional al rango del individuo*, produciéndose una repartición más uniforme de la probabilidad de selección del individuo (Brindle, 1991). Si se ordenan los individuos del peor al mejor, la probabilidad de selección siendo esta proporcional al rango del individuo viene dada por:

$$p_{j,t}^{rango} = \frac{rango(g(I_t^j))}{\lambda(\lambda + 1)/2} \quad \text{Ec. (3.2)}$$

Esta función basada en el rango no varía en razón al escalamiento ni a la traslación, y la constante de normalización parte de la suma de los rangos $\lambda(\lambda + 1)/2$.

Otro esquema que ha dado resultados aceptables dado por (Brindle, 1991), se llama *muestreo estocástico con reemplazamiento del resto*, entonces el número de ocasiones que el individuo escogido es:

$$n(I_t^j) = n_1(I_t^j) + n_2(I_t^j) \quad \text{Ec. (3.3)}$$

Se plantea por (Baker, 1987) un método nombrado *muestreo universal estocástico*, en donde se utiliza una ruleta cuyos sectores circulares son proporcionales a la función de adaptación.

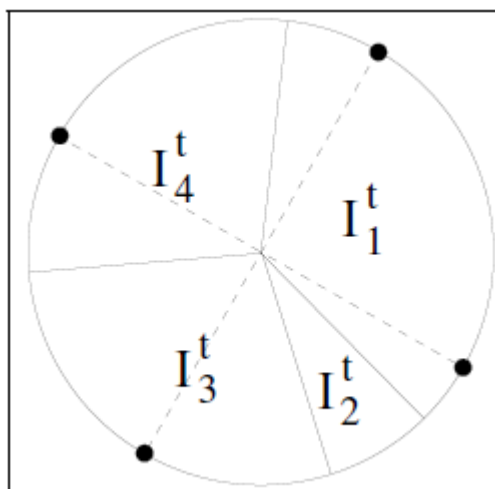


Figura 17. Método de selección muestreo universal estocástico

Fuente: (Baker, 1987)

Este método usa un muestreo sistemático, no estratificado como en la selección proporcional a la función de adaptación.

Se tiene también una *selección elitista*, en la que se fuerza la selección del mejor individuo en un tiempo t . Otro modelo es la denominada *selección por torneo*, que es una selección de padres bastante extendida, se trata de elegir al azar un número determinado de individuos, este número es el tamaño del torneo, escoger al mejor individuo de este grupo, y volver a realizar esta acción hasta que el número de individuos escogidos sea igual al tamaño de la población, usualmente el tamaño del torneo es dos (Baker, 1987).

Una buena opción es utilizar los denominados *métodos de selección dinámicos*, que se caracterizan porque las probabilidades de selección de un individuo varían en cada generación, lo que no ocurre en los métodos antes mencionados donde estas probabilidades se mantienen constantes. También se pueden dividir en método *preservativo*, el cual asigna una probabilidad distinta de cero; y el método *extintivo*, que sí presenta una probabilidad igual a cero.

3.1.4. Cruce

Se puede utilizar un *cruce basado en un punto*, como lo hace un algoritmo genético canónico, en el cual los individuos seleccionados para ser padres se recombinan a partir de un punto de corte, intercambiando después las secciones que se encuentran a la derecha de este punto.

También han sido desarrollados otros tipos de operadores de cruce con más de un punto de cruce. El *operador de cruce basado en múltiples puntos*, investigado por (De Jong, 1975), sacó a relucir que el cruce basado en dos puntos supone una mejora pero que añadir más puntos no presenta beneficio para el algoritmo. El beneficio que se obtiene al trabajar con más de un punto de cruce se radica en que permite explorar más fácilmente el espacio de búsqueda. Como se indica en la figura 18, el operador de cruce que utiliza dos puntos realiza la selección de dichos puntos de manera aleatoria.

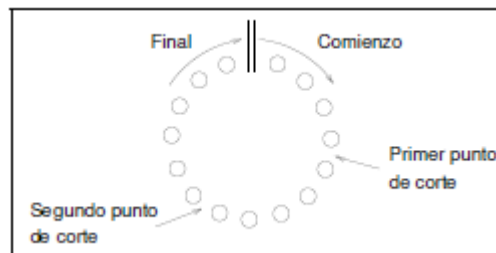


Figura 18. Individuo visto como un disco

Fuente: (De Jong, 1975)

Es así que el cruce basado en un punto puede considerarse un caso particular del basado en dos puntos, donde se encuentra fijo uno de los puntos de corte al inicio de la cadena que constituye al individuo.

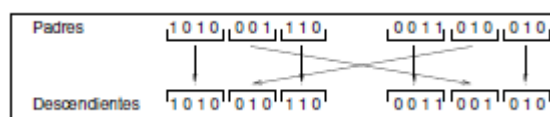


Figura 19. Operador basado en dos puntos

Fuente: (De Jong, 1975)

Existe un operador de cruce llamado *uniforme*. En este método cada gen de los individuos de la descendencia es creado copiando el gen correspondiente de uno de los padres, elegido siguiendo una máscara originada aleatoriamente (Syswerda, 1991). Dependiendo de si existe un 1 o 0 en la máscara de cruce, el gen copiado proviene del primer o segundo padre respectivamente como se muestra en la figura 19. La máscara

de cruce está formada por una muestra de longitud λ , también se podría utilizar un operador de cruce basado en la función de adaptación.

Máscara de cruce	1	0	0	1	0	0	1
Padre 1	1	1	0	1	1	0	1
	↓			↓			↓
Descendiente	1	0	0	1	1	1	1
		↑	↑		↑	↑	
Padre 2	0	0	0	1	1	1	0

Figura 20. Operador de cruce uniforme

Fuente: (Syswerda, 1991)

Máscara de cruce	1	1	1	0	0	0	0	1	1	0	0	0	1
Padre 1	1	0	1	1	0	0	1	1	0	1	1	0	0
Descendiente	1	0	1	0	1	1	1	1	0	0	0	1	0
Padre 2	1	0	0	0	1	1	1	1	0	0	0	1	1

Figura 21. Máscaras de cruce para operadores con 1 y 2 puntos

Fuente: (Syswerda, 1991)

3.1.5. Mutación

Se considera a la mutación un operador básico, necesario que provee aleatoriedad al entorno de los individuos que forman parte de la población. Este operador es el encargado de realizar la búsqueda por todo el espacio de soluciones factibles, que también cobra importancia a medida que se da la convergencia de la población de individuos. Los AGs que utilizan solamente selección y cruce se denominan de *evolución primitiva*. Se puede observar según estudios realizados (Schaffer, Caruna, & Eshelman, 1989) que los algoritmos genéticos que utilizan la mutación, además de selección y cruce, presentan mejores resultados, ya que la determinación óptima de probabilidad de mutación es más decisiva que la relacionada con el cruce.

En la búsqueda del valor óptimo de probabilidad de mutación (De Jong, 1975) aconseja el uso de una probabilidad de mutación del bit l^{-1} , l es la longitud de la cadena. En base a resultado experimentales (Schaffer, Caruna, & Eshelman, 1989)

estimaron una tasa óptima en el orden de $\frac{1}{\lambda^{0.9318}\lambda^{0.4535}}$, siendo λ el número de individuos de la población.

En la mayoría de implementaciones de algoritmos genéticos la probabilidad de cruce y mutación son constantes, sin embargo, algunos autores han modificado la probabilidad de mutación a medida que aumenta el número de iteraciones, obteniendo así, un mejor desempeño.

3.1.6. Reducción

Ya obtenidos los descendientes de la población en un tiempo t , se procede a reducir este número de descendientes al tamaño original de la población, escogiendo λ individuos de entre los individuos que son parte de la población en un tiempo determinado t y sus descendientes. Existen algunos procedimientos para efectuar la reducción, pero mencionaremos dos principales y las variantes que se les puede hacer.

La *reducción simple*, donde los λ descendientes son los que conforman la población en el tiempo $t + 1$; o sino se eligen los λ individuos más adaptados, de entre los 2λ individuos de la población, denominándose esta *reducción elitista de grado λ* (Hollan, 1975). Un concepto importante dentro de la reducción es el de la *tasa de reemplazamiento generacional* o t_{rg} , que corresponde al porcentaje de descendientes producidos con respecto al grueso de la población.

Partiendo de la idea de (Hollan, 1975), de que el reemplazo se realice de 1 en 1 con una tasa $t_{rg} = \lambda^{-1}$, normalmente se hace el reemplazo con una tasa de $t_{rg} = 1$. Con el fin de hacer un solapamiento controlado entre padres y descendientes, (De Jong, 1975) introdujo el concepto de t_{rg} . Los algoritmos genéticos en los que los hijos resultantes de la selección y cruce reemplazan a individuos de la población anterior en proporción a la t_{rg} , son conocidos como SSGA (Steady State Genetic Algorithm).

Los algoritmos genéticos han sido aplicados en conjunto con la teoría de control clásico. En controladores PID como el descrito en (Ruge & Alvis, 2009), para el diseño de sistemas de distribución de energía eléctrica como en (Bernal), aplicaciones para el diseño de controladores robustos detalladas en (Martínez, Sanchis, & Blasco, 2006), o controladores multiobjetivo especificados en (Herreros).

3.2. Algoritmo de Nealder-Mead

Este algoritmo es utilizado para encontrar el mínimo de una función definida en un espacio n -dimensional, sin necesidad de sus derivadas. No se puede disponer de las derivadas en casos como cuando el cálculo de las derivadas parciales representa un costo computacional muy alto, o cuando no se posee una forma explícita de estas derivadas parciales. Este algoritmo es un método de optimización de búsqueda directa, que es robusto y funciona aún para funciones no convergentes o que presenten discontinuidades. El algoritmo de Nealder-Mead es bastante utilizado en química, ingeniería y medicina. Desde su publicación, se ha constituido en uno de los algoritmos más utilizados para programación no lineal sin restricciones (Pilotta).

El algoritmo implementado por (Nealder & Mead, 1965), utiliza el método de bisección para ecuaciones no lineales unidimensional, que considera los valores de la función dentro de un conjunto de puntos del plano o del espacio con el objetivo de localizar el valor mínimo de la función. Aunque utilice el mismo concepto geométrico que el método simplex aplicado a la programación lineal, el algoritmo de Nealder-Mead no debe ser confundido con este.

Cada iteración empieza con un simplex en \mathbb{R}^n dado por los $n + 1$ vértices del plano y los correspondientes valores de la función (Pilotta). Un simplex se refiere a una envoltura convexa de un conjunto de $n + 1$ puntos independientes afines a un espacio de dimensión n o de mayor. Después de efectuar algunas pruebas en los valores de f en uno o dos puntos adicionales, esta iteración del método concluye con un nuevo simplex, diferente del inicial, de tal manera que los valores de la función en los nuevos vértices cumplen la condición de descenso cotejada con el simplex inicial de la iteración (Pilotta). Una de las características que distinguen a este algoritmo de los otros de búsqueda directa, es que el número de evaluaciones extra de la función que se hacen por iteración, es bajo. Se pueden hacer n evaluaciones por iteración. Se conoce que en general el algoritmo trabaja bien para dimensiones pequeñas, pero presenta inconvenientes con dimensiones grandes. Al ser un método heurístico, hasta el momento han sido probados pocos resultados de convergencia (Pilotta), sin embargo en los años recientes ha vuelto a despertar el interés de científicos y matemáticos. Es así que se busca entender sus propiedades teóricas tanto como sea posible (Pilotta).

Al inicio de cada iteración se empieza con un *simplex no degenerado* en \mathbb{R}^n y se concluye con otro simplex diferente del anterior en \mathbb{R}^n . Un simplex no degenerado en

\mathbb{R}^n es una cápsula que converge en $n + 1$ puntos no coplanares, así los vértices del simplex inicial se ordenan de la siguiente manera:

$$f_1 \leq f_2 \dots \leq f_{n+1} \quad \text{Ec. (3.4)}$$

Donde $f_i = f(x_i)$.

Al buscar el minimizador de f , se declara a x_1 como el mejor vértice y a x_{n+1} como el peor.

El diámetro del simplex viene definido por:

$$\text{diam}(S) = \max_{1 \leq i, j \leq n+1} \|x_i - x_j\| \quad \text{Ec. (3.5)}$$

3.2.1. Procedimiento

Antes se deben definir ciertos parámetros esenciales para la ejecución del método, estos parámetros son ρ , δ , γ y σ , que son usados en cada iteración y deben cumplir con las siguientes condiciones:

$$\delta > 1, \quad 0 < \rho < \delta, \quad 0 < \gamma < 1, \quad 0 < \sigma < 1$$

Los valores de estos parámetros usados por defecto son:

$$\delta = 2, \quad \rho = 1, \quad \gamma = \frac{1}{2}, \quad \sigma = \frac{1}{2}$$

3.2.1.1. Ordenamiento.

Se ordenan los $n + 1$ vértices del simplex como se muestra en la figura 22.

3.2.1.2. Reflexión.

Se procede a calcular el centroide de los mejores n puntos.

$$\hat{x} = \sum_{i=1}^n \frac{x_i}{n} \quad \text{Ec. (3.6)}$$

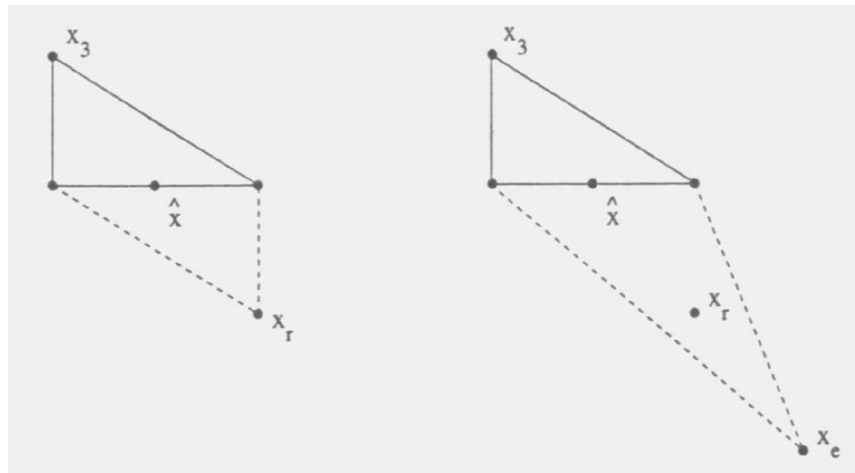


Figura 22. Replicación y expansión en el algoritmo de Nelder-Mead

Fuente: (Pilotta)

Calcular el punto de reflexión.

$$x_r = \hat{x} + \rho(\hat{x} - x_{n+1}) = (1 + \rho)\hat{x} - \rho x_{n+1} \quad \text{Ec. (3.7)}$$

Se calcula $f_r = f(x_r)$.

Si $f_1 \leq f_r < f_n$, se toma x_r como nuevo vértice del simplex, se elimina el peor vértice y se concluye la iteración.

3.2.1.3. Expansión.

Se calcula el punto de expansión si cumple que $f_r < f_1$ y se evalúa $f_e = f(x_e)$.

$$x_e = \hat{x} + \delta(x_r - \hat{x}) = \hat{x} + \rho\delta(\hat{x} - x_{n+1}) = (1 + \rho\delta)\hat{x} - \rho\delta x_{n+1} \quad \text{Ec(3.8)}$$

Si $f_e < f_r$, tomamos x_e , eliminamos el peor vértice y terminamos la iteración; si en cambio, $f_e \geq f_r$, se toma x_r , se elimina el peor vértice y se concluye la iteración.

3.2.1.4. Contracción.

Si $f_r \geq f_n$, se efectúa la contracción entre el mejor punto de x_{n+1} y x_r , con \hat{x} . Se puede realizar de dos formas.

- **Contracción externa**

Si $f_n \leq f_r < f_{n+1}$, calculamos:

$$x_{ce} = \hat{x} + \gamma(x_r - \hat{x}) = \hat{x} + \rho\gamma(\hat{x} - x_{n+1}) = (1 + \rho\gamma)\hat{x} - \rho\gamma x_{n+1} \quad \text{Ec. (3.9)}$$

Después evaluamos $f_{ce} = f(x_{ce})$.

Si $f_{ce} \leq f_r$, tomamos x_{ce} , eliminamos el peor vértice y terminamos la iteración; si no cumple la condición, continuamos al siguiente paso.

- **Contracción interna**

Si $f_r \geq f_{n+1}$, calculamos:

$$x_{ci} = \hat{x} - \gamma(\hat{x} - x_{n+1}) = (1 - \gamma)\hat{x} + \gamma x_{n+1} \quad \text{Ec. (3.10)}$$

Después evaluamos $f_{ci} = f(x_{ci})$.

Si $f_{ci} < f_{n+1}$, tomamos x_{ci} , eliminamos el peor vértice y terminamos la iteración; si no cumple la condición, continuamos al siguiente paso.

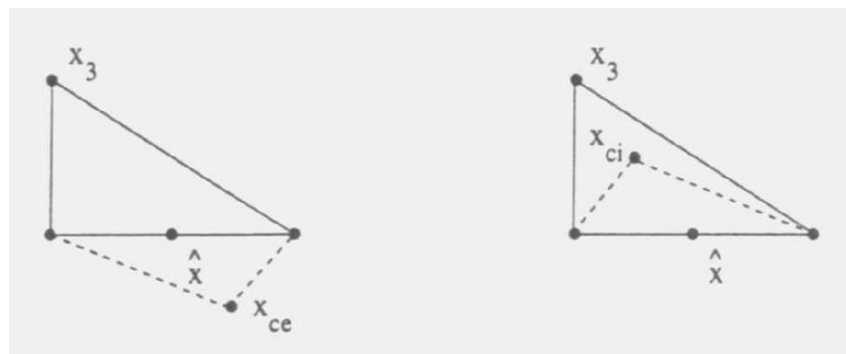


Figura 23. Contracción interna y externa

Fuente: (Pilotta)

3.2.1.5. Encogimiento.

En este paso se evalúa f en los n puntos $y_i = x_1 + \sigma(x_i - x_1)$, para $i = 2, \dots, n + 1$.

Los vértices nuevos en la siguiente iteración serán x_1, v_2, \dots, v_{n+1} .

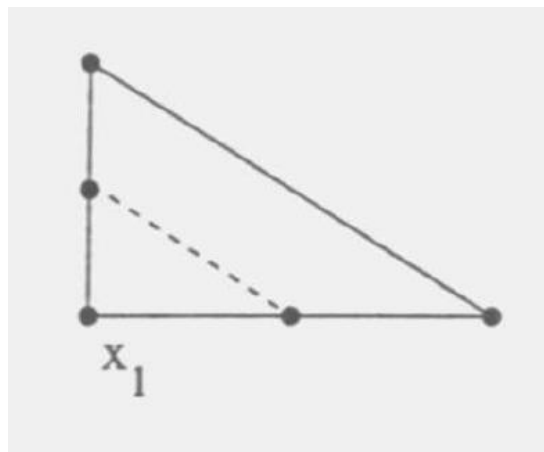


Figura 24. Encogimiento

Fuente: (Pilotta)

3.3. Control difuso

El control difuso (Fuzzy Logic) apareció como una herramienta muy útil para el control de procesos industriales con cierto grado de complejidad, sistemas de diagnóstico, para electrónica del hogar y entretenimiento, y otros sistemas especializados. El desarrollo vertiginoso de esta tecnología vino desde Japón a pesar de haber sido creada en Estados Unidos. De manera que las patentes por aplicaciones de esta lógica aumentan de manera exponencial (Montejo, s.f.).

El desarrollo del control difuso se origina en los sistemas basados en reglas para la toma de decisiones (Gómez). Estos tipos de controladores, de la misma manera que en otros, adquieren los valores de las variables de entrada, efectúan un procesamiento de estas, y modifican los valores de la salida en concordancia con este procesamiento afectando a la planta. La diferencia fundamental se presenta en que, en lugar de procesar ecuaciones o tablas, se procesan reglas para determinar el cambio de las salidas. Esto facilita su uso, debido a que las reglas se basan más en el sentido común que en modelos y ecuaciones (Gómez).

El procesamiento de reglas se encuentra relacionado con los sistemas expertos, que son resultado del estudio de la inteligencia artificial. Con estos sistemas se logró dotar del conocimiento y experiencia que posee el humano para resolver problemas. Estos sistemas utilizan este conocimiento y mecanismos de inferencia para la resolución de problemas específicos, con el fin de obtener resultados similares a los del ser humano.

El control difuso se fundamenta en una lógica multievaluada que acepta valores intermedios que nos permiten especificar evaluaciones convencionales como

pequeño/grande, blanco/negro, etc. Este tipo de evaluaciones pueden expresarse matemáticamente y por lo tanto ser procesadas computacionalmente (Montejo, s.f.).

Un sistema basado en reglas se expresa con reglas del tipo:

- **Si antecedente entonces consecuente.**
- **Si condición entonces acción.**

La parte de control de programa le corresponde al llamado *mecanismo de inferencia*, cuyo objetivo es procesar las reglas incluidas en la *base de conocimiento*, remitiéndose a los datos obtenidos y al estado del sistema, para con esta información producir cambios en el mismo, y repetir el proceso (Gómez).

La teoría de conjuntos fuzzy fue introducida por el profesor Lotfi Zadeh en la Universidad de California (Berkeley) a mediados de los años 60, partiendo de la idea de que la pertenencia de un elemento a determinado conjunto no se limitaba solamente a que pertenece o no, sino que tiene cierto grado de pertenencia. La vinculación entre el elemento y su grado de pertenencia viene dada por las denominadas *membership functions* (Gómez). Es así que, al usar la lógica difusa para evaluar reglas, según el grado de pertenencia que tengan, influirán en menor o mayor grado en la solución del problema. Los controladores difusos constan de procesos cíclicos que se dividen en tres partes primordiales, la fuzzificación, base de reglas, inferencia difusa y defuzzificación, tal como se observa en la figura 25.

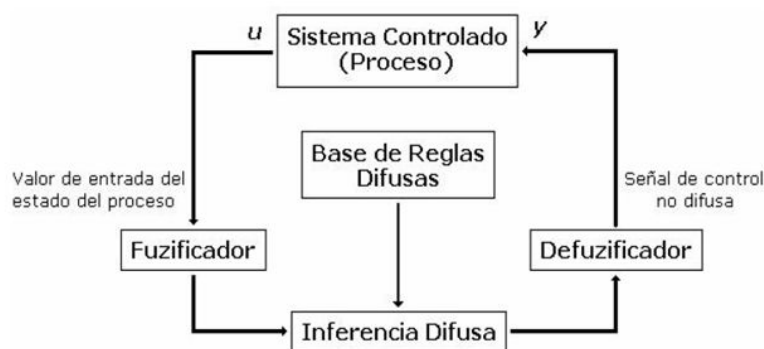


Figura 25. Arquitectura de un control fuzzy

3.3.1. Fuzzificación

Constituye la primera parte del proceso, donde después de la adquisición de los valores de las variables del sistema, se calcula el grado de pertenencia de dichas

variables a los conjuntos fuzzy (Gómez), de este proceso se obtienen las fuzzy inputs. El procedimiento se ejemplifica en la figura 26.

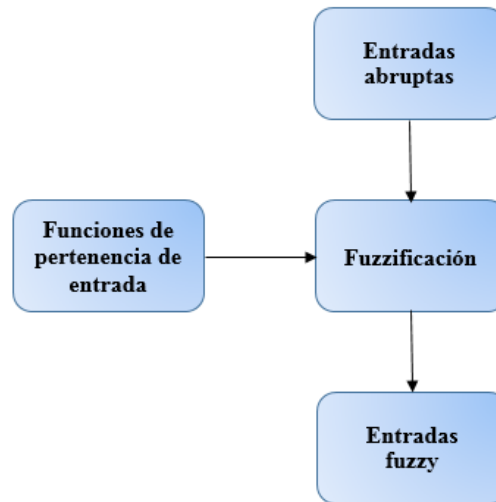


Figura 26. Fase de Fuzzificación

Para efectuar la transformación de las variables de entrada (entradas abruptas) en entradas fuzzy, primeramente se deben definir las funciones de pertenencia de la variable, después de haber hecho esto, la variable ingresada se compara con la función de pertenencia adecuada dando como resultado una entrada fuzzy (Tejada).

3.3.2. Evaluación de reglas

La evaluación de reglas engloba la base de reglas y la inferencia difusa, en esta etapa se avalúa los antecedentes de las reglas para conseguir el grado de veracidad o *peso* para cada regla. El controlador fuzzy utiliza reglas lingüísticas para representar lo obtenido de la etapa de fuzzificación (Tejada).

Las reglas difusas son sentencias de la forma si→entonces que detalla la acción que se va a tomar como consecuencia a las entradas fuzzy (Tejada). Las reglas están supeditadas a un conjunto predefinido de términos lingüísticos, que tienen la siguiente sintaxis:

$$IF \text{ antecedente}_1 \text{ AND } \text{antecedente}_2 \dots \Rightarrow \text{consecuente}_1 \text{ AND } \text{consecuente}_2 \dots$$

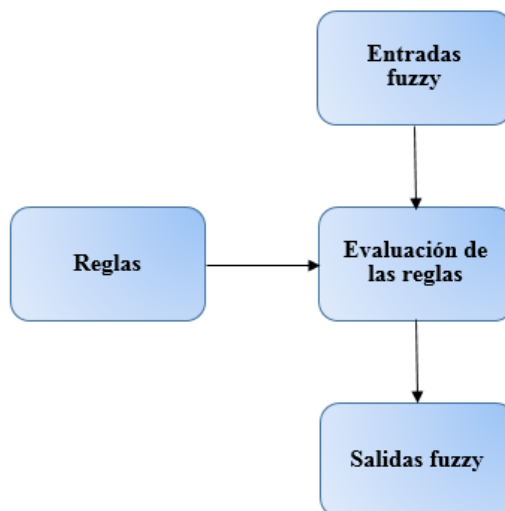


Figura 27. Fase de evaluación de reglas

Teniendo a AND como uno de los operadores del control difuso
 Los antecedentes tienen la siguiente forma:

$$\text{variable de entrada} = \text{etiqueta}$$

El formato del consecuente viene dado por:

$$\text{variable de salida} = \text{etiqueta}$$

Las reglas siguen el sentido que se les ha determinado según el comportamiento del sistema y se escriben de acuerdo a las etiquetas de las funciones de pertenencia (Tejada).

3.3.3. Defuzzificación

En este paso se busca obtener las salidas, de la misma naturaleza de las variables del sistema (salidas abruptas), con este fin, cada salida fuzzy modificará a su respectiva función de pertenencia de salida (Tejada). El proceso que se realiza para este cálculo se encuentra representado en la figura 28.

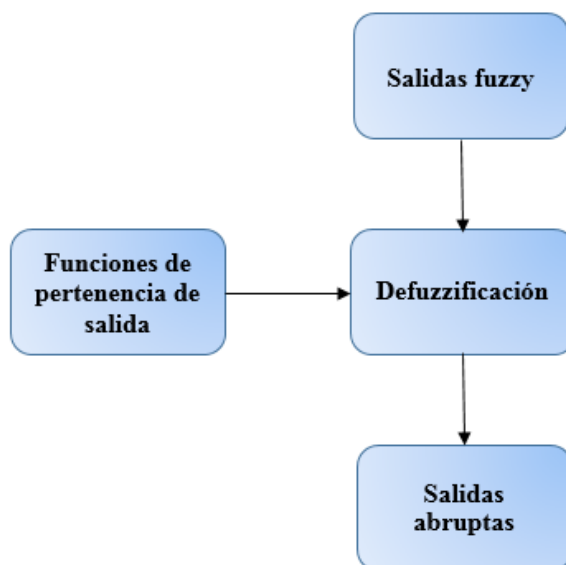


Figura 28. Fase de Defuzzificación

Los métodos de defuzzificación más utilizados son:

3.3.3.1. Método de la media ponderada

Este método calcula el promedio entre los valores de salida que se obtienen para cada uno de los conjuntos fuzzy multiplicados por la ponderación o peso de la respectiva regla o grado de pertenencia (Tejada).

3.3.3.2. Método del centro de gravedad

Se trata de determinar el centro de gravedad o centroide del conjunto de salida, que es el resultado de la unión de las contribuciones de todas las reglas válidas. El valor de la salida es la abscisa del punto del centroide (Tejada).

3.3.4. Controlador PD difuso

El diseño de un controlador fuzzy se desarrolla con una base de la teoría clásica, con arquitectura serie o paralela (Chamorro, Vladimir, & Trujillo, 2010). Ya que lo que se requiere es regular la salida de un proceso en torno a una referencia, resulta lógico considerar al error, a la derivada del error y/o a la integral del error como entradas, inclusive para la implementación de un controlador difuso.

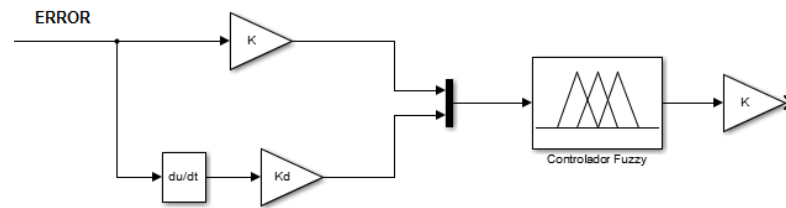


Figura 29. Controlador difuso PD

Cuando se implementa un controlador PD difuso, para una aplicación como la del levitador magnético, es complicado predecir el efecto de las distintas ganancias que intervienen en el tiempo de subida, el sobre impulso y el tiempo de establecimiento, que es donde más no linealidades se presentan (Chamorro, Vladimir, & Trujillo, 2010). Entonces las ganancias que se impongan presentan una gran importancia. El controlador PD difuso ofrece una rápida corrección de las perturbaciones, además ha sido implementado de manera satisfactoria en diferentes convertidores de potencia.

CAPÍTULO IV

DISEÑO Y SIMULACIÓN DE ALGORITMOS DE CONTROL

4.1. Diseño del controlador fuzzy PD

Como ya se denotó anteriormente este tipo de controladores utiliza una lógica multievaluada que busca dar a los sistemas computacionales la capacidad de interpretación y posterior solución de problemas que posee el raciocinio del ser humano. Se tomará como punto de partida la teoría de controladores difusos, y la ayuda del FIS Editor de MATLAB.

Se determina primero las entradas y salidas del sistema, luego el rango de funcionamiento tanto de entradas como salidas. Así, partiendo del conocimiento adquirido en los experimentos de modelamiento del sistema, se debe tener en cuenta que mientras mayor sea la tensión aplicada al electromagneto, mayor será la corriente que circule por el mismo y por lo tanto será mayor el campo electromagnético generado para atraer la esfera. Por el contrario, mientras menos campo se genere, menos atracción tendrá la esfera.

4.1.1. Variables lingüísticas y valores lingüísticos

Dado el sistema de levitación magnética (MLS) y que se implementará un controlador PD difuso, se han definido dos entradas, que parten de la señal del sensor adquirida por la tarjeta STM32F4; y una señal de salida que corresponde a la señal de control que proporciona la tarjeta a la etapa de potencia (pre-actuador). Entonces, las variables lingüísticas quedan definidas como:

- ❖ Error → entrada
- ❖ Derivada del error → entrada
- ❖ Señal de control → salida

Los valores lingüísticos determinados para las variables definidas son:

- ❖ Grande negativo (gn)
- ❖ Pequeño negativo (pn)

- ❖ Cero (z)
- ❖ Pequeño positivo (pp)
- ❖ Grande positivo (gp)

4.1.2. Rango de las variables lingüísticas

Para determinar los rangos más idóneos dentro de los cuales deberían actuar las variables lingüísticas se tomarán en cuenta los rangos de las mediciones que se pueden obtener. El rango de la señal de adquisición del sensor de posición puede ser tomado en voltios o en posición dada en milímetros. Se trataron ambas opciones, de la que mejores resultados se obtuvieron fue de la medición tomada en voltios, sin embargo, cualquiera de las dos opciones es loable. En este caso se optó por definir los rangos para la medición efectuada en voltios. Los rangos dispuestos se muestran en la tabla 10.

Tabla 10
Rangos de variables lingüísticas

Variable lingüística	Rango
Error	-1 hasta 1
Derivada del error	-1 hasta 1
Señal de control	-1.5 hasta 1.5

Las *funciones de membresía* que se utilizaron para cada variable lingüística, fueron triangulares, tanto para los valores lingüísticos medios como para las fronteras, como se visualiza en las figuras 30, 31 y 32.

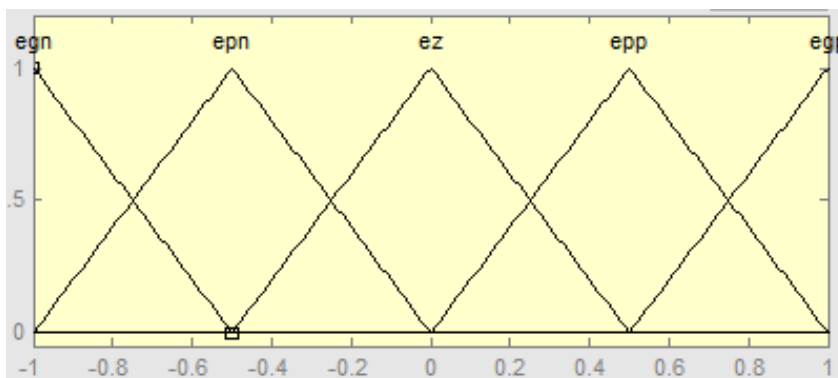


Figura 30. Funciones de membresía del error

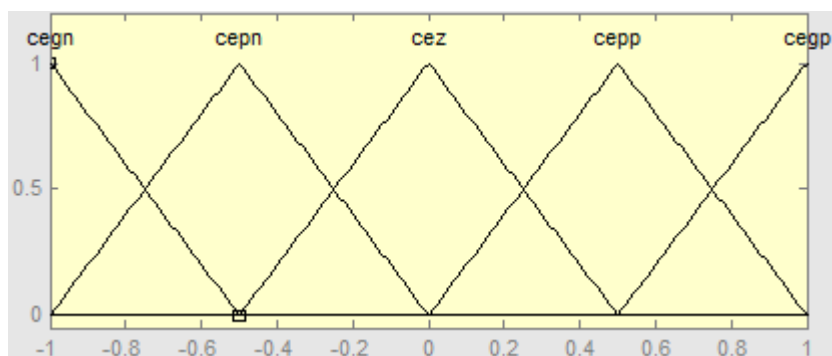


Figura 31. Funciones de membresía de la derivada del error

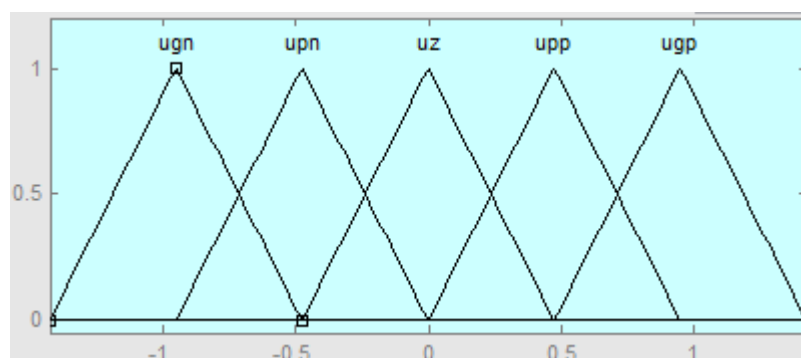


Figura 32. Funciones de membresía de la señal de control

4.1.3. Evaluación de reglas

En la evaluación de reglas, se debe definir la base de reglas que parten de las variables lingüísticas y los valores lingüísticos, que determinan la manera en que el sistema va a reaccionar de acuerdo a los valores adquiridos de la señal del sensor dependiendo de la posición en la que se encuentre la esfera. Las reglas necesarias para controlar el sistema dado que tenemos tres variables lingüísticas y cinco valores lingüísticos, son veinte y cinco en total, y se presentan en la tabla 11.


Los valores lingüísticos se representan con los siguientes acrónimos:

- **Error**
 - ❖ Error grande negativo (egn)
 - ❖ Error pequeño negativo (epn)
 - ❖ Error cero (ez)
 - ❖ Error pequeño positivo (epp)

- ❖ Error grande positivo (egp)
- **Derivada del error**
- ❖ Derivada del error grande negativa (cegn)
- ❖ Derivada del error pequeño negativa (cepn)
- ❖ Derivada del error cero (cez)
- ❖ Derivada del error pequeño positiva (cepp)
- ❖ Derivada del error grande positiva (cegp)
- **Señal de control**
- ❖ Señal de control grande negativa (ugn)
- ❖ Señal de control pequeño negativa (upn)
- ❖ Señal de control cero (uz)
- ❖ Señal de control pequeño positiva (upp)
- ❖ Señal de control grande positiva (ugp)

Tabla 11
Base de reglas

1	IF	egn	AND	cegn	THEN	ugp
2	IF	egn	AND	cepn	THEN	ugp
3	IF	egn	AND	cez	THEN	ugp
4	IF	egn	AND	cepp	THEN	upp
5	IF	egn	AND	cegp	THEN	uz
6	IF	epn	AND	cegn	THEN	ugp
7	IF	epn	AND	cepn	THEN	ugp
8	IF	epn	AND	cez	THEN	upp
9	IF	epn	AND	cepp	THEN	uz
10	IF	epn	AND	cegp	THEN	upn
11	IF	ez	AND	cegn	THEN	ugp
12	IF	ez	AND	cepn	THEN	upp
13	IF	ez	AND	cez	THEN	uz
14	IF	ez	AND	cepp	THEN	upn
15	IF	ez	AND	cegp	THEN	ugn
16	IF	epp	AND	cegn	THEN	upp
17	IF	epp	AND	cepn	THEN	uz

Continua 

18	IF	epp	AND	cez	THEN	upn
19	IF	epp	AND	cepp	THEN	ugn
20	IF	epp	AND	cegp	THEN	ugn
21	IF	egp	AND	cegn	THEN	uz
22	IF	egp	AND	cepn	THEN	upn
23	IF	egp	AND	cez	THEN	ugn
24	IF	egp	AND	cepp	THEN	ugn
25	IF	egp	AND	cegp	THEN	ugn

En la figura 33 se visualiza el editor de reglas de la herramienta de MATLAB utilizada para configurar el controlador fuzzy implementado.

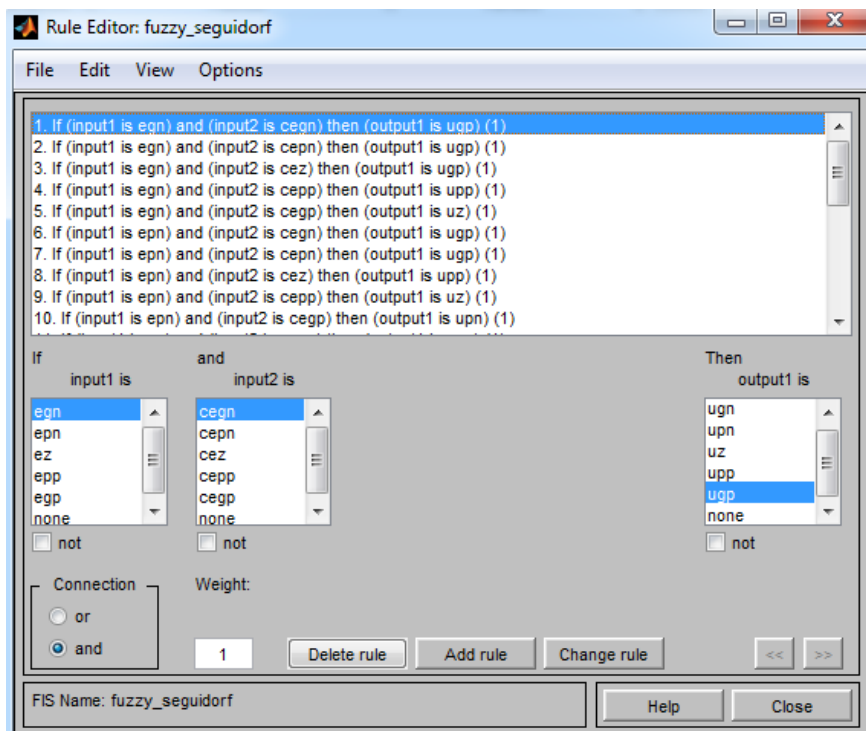


Figura 33. Editor de reglas del FIS Editor

4.1.4. Simulación del sistema de levitación magnética con controlador PD difuso

Para simular el sistema de levitación, se utilizará la función de transferencia obtenida del modelamiento de la planta realizado en el capítulo II, la simulación se efectúa en el entorno de Simulink de MATLAB.

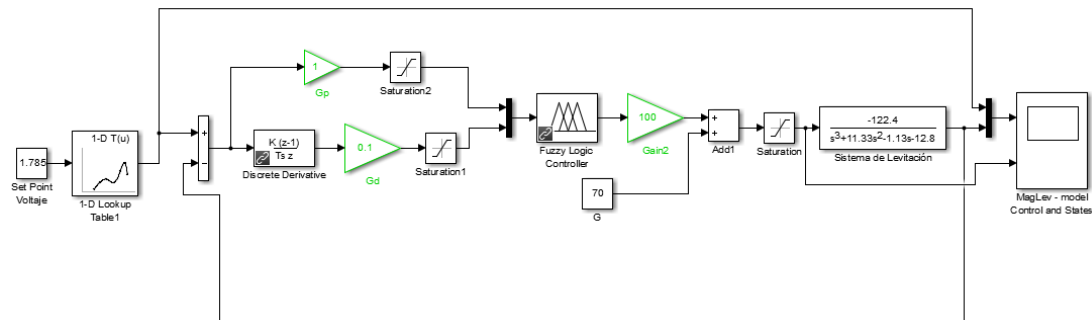


Figura 34. Simulación del sistema de levitación con control difuso

En la figura 34 se visualiza las distintas etapas que se realizan en la simulación y que posteriormente se implementarán en el sistema físico, que se describen de la siguiente forma:

- El uso del Lookup Table para convertir la señal de adquisición del sensor que se encuentra en voltaje. A la salida de esta herramienta se obtiene la medición de posición en milímetros.
- Se muestra el error y la derivada del error, señales que tienen una determinada ganancia junto con limitadores, ya que las entradas del controlador varían entre -1 y 1, esto modifica los rangos de los valores con los que ingresan al controlador fuzzy, y así como también la velocidad con la que cambian. Las ganancias se modificarán optimizando la respuesta del sistema mediante los algoritmos que se utilizarán más adelante.
- Se observa el controlador, que se realiza con la herramienta fuzzy de MATLAB y configuración en el FIS Editor, cuyo diseño se reseña en el apartado anterior.
- Después se dispone una ganancia de 100 debido a que la señal de control debe contribuir con el Duty Cycle para el pwm, y este es de 0 a 100. Se suma al ciclo de trabajo una constante que corresponde al control mínimo necesario para levantar la esfera, cuyo valor viene dado por el ciclo de trabajo mínimo que se requiere para atraer a la esfera desde la posición determinada por el set point, valores ya determinados en el experimento de control mínimo que se efectúa para el modelamiento del sistema.

- Finalmente se muestra la función de transferencia correspondiente al modelamiento efectuado, y se envía la salida a un scope para poder visualizar la salida del sistema.

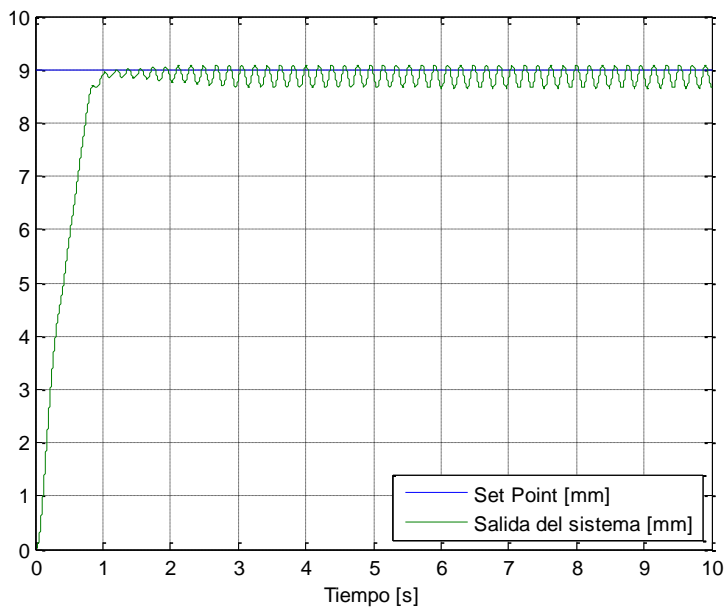


Figura 35. Set Point y salida del sistema

En la figura 35 se puede observar el set point y la salida del sistema, se aprecia unas oscilaciones las cuales se espera corregir aplicando los algoritmos de optimización presentados en este trabajo.

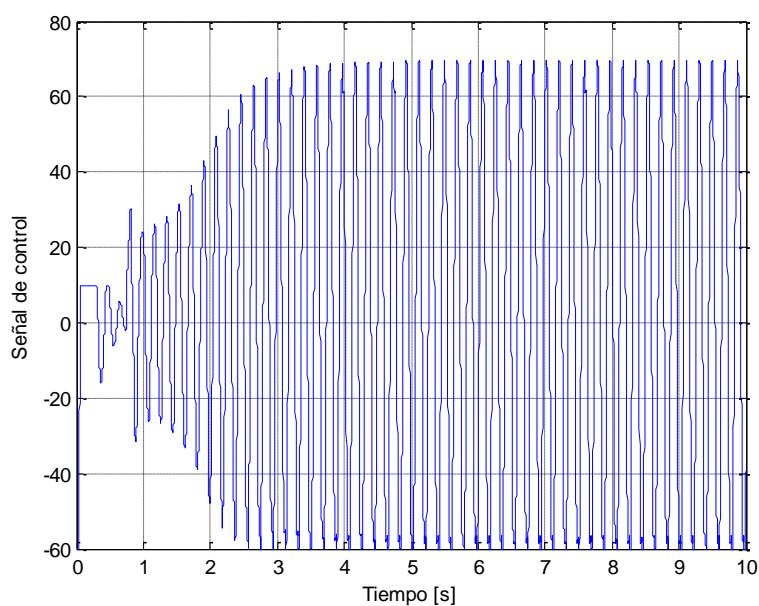


Figura 36. Señal de control

La figura 36 nos muestra la señal de control, resultante del controlador difuso después de todas las configuraciones de ganancias y limitadores que se dispusieron. El objetivo primordial de este tipo de control es lograr el establecimiento del sistema, ya que no se puede predecir ni establecer exactamente los parámetros como el tiempo de subida, el sobre impulso y el tiempo de establecimiento.

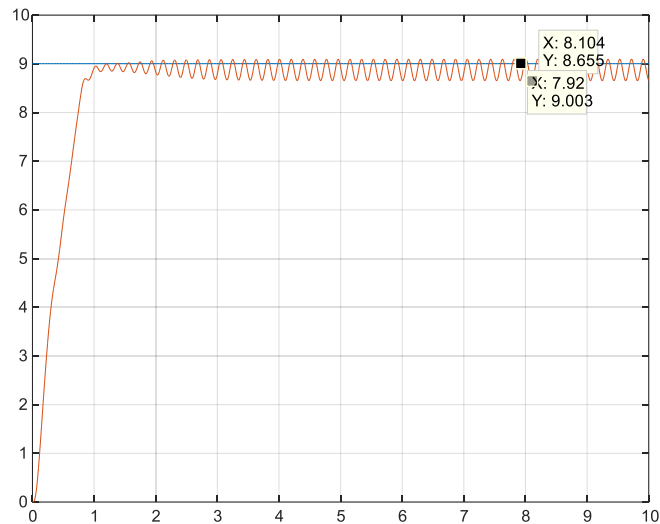


Figura 37. Error del sistema

Se puede observar que las oscilaciones en la salida no son uniformes respecto al set point. La figura 37 muestra el error que tiene el sistema, para medir este error se tomó el valor inferior de la salida con respecto al set point, ya que es el que mayor error presenta.

$$e = (9 - 8.655)mm = 0.345mm$$

$$e = 3.83\%$$

4.2. Diseño del algoritmo genético

En el proyecto realizado se ha planteado la temática del diseño de controladores en torno a la optimización de un parámetro, dicho parámetro es el error, ya que es el principal objetivo que el error sea el menor posible.

Con la aplicación de los algoritmos genéticos se busca optimizar los valores de las ganancias del error, la derivada del error y la constante de control mínimo que se suma

a la salida del controlador, para de esta manera mejorar la respuesta del sistema. Mientras mejor sea la respuesta ofrecida por determinados valores de estas constantes, mejor será la respuesta del sistema y por lo tanto menor el error que obtenga.

La programación del algoritmo se realizó en varios scripts de MATLAB, considerando a cada proceso como una función para efectuar cada etapa del algoritmo de una manera más organizada ya que implica una programación relativamente extensa, además de conservar así un orden determinado, tal como se observa en la figura 38.

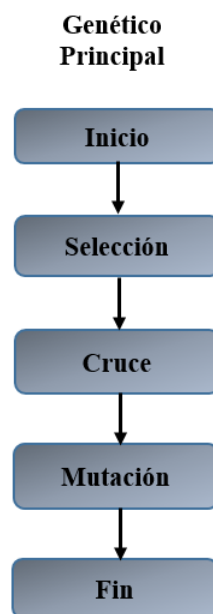


Figura 38. Descripción de la programación del algoritmo

4.2.1. Población inicial y codificación de individuos

En este apartado se define la dimensión de la cadena que representa cada individuo, así como el tamaño de la población inicial. La longitud de la cadena de cromosoma es de 28, significa que cada individuo tendrá 28 espacios de información, donde cada espacio acarrea determinada información del individuo. La población inicial que se utilizará es de 25 individuos. La información de cada individuo y la disposición de la cadena se muestra en la tabla 12.

Tabla 12
Cadena de cromosoma de cada individuo

8 bits	8 bits	8 bits	Posición 25	Posición 26	Posición 27	Posición 28
Gp	Gd	G	Gp	Gd	G	J
binario	binario	binario	decimal	decimal	decimal	

Donde se definen cada una de las constantes y el desempeño que se van a operar dentro del algoritmo, de la siguiente manera:

Gp: Ganancia proporcional del controlador

Gd: Ganancia derivativa del controlador

G: Ganancia de control mínimo

J: Desempeño del algoritmo

4.2.2. Función objetivo

La función objetivo en este caso, al ser la respuesta del sistema la que se requiere corregir, será el error cuadrático medio. Esto se da ya que mientras menor sea el error mejor respuesta presentará el sistema. Se mostrará el desempeño que se define por la siguiente expresión:

$$J = \frac{1}{\sum_{i=1}^{tam} (error^2)} \quad Ec. (4.1)$$

Donde J corresponde al desempeño de los individuos en cada generación, y tam representa el tamaño de la población. El objetivo primordial de la aplicación del algoritmo genético en este sistema será maximizar el desempeño dependiente del error que tenga, así se puede apreciar que mientras menor sea el error mayor será el desempeño, y se escogerá los valores de las constantes optimizadas que mejor desempeño ofrezcan.

4.2.3. Selección

En la selección se siguen los lineamientos básicos ya vistos en el marco teórico, se seleccionan los padres para la siguiente generación. Se analiza el desempeño y de

acuerdo a este se van escogiendo los padres para la nueva generación. Se realiza la selección de los padres siguiendo la ecuación de probabilidad de selección 3.2, que denota como ya se vio, una proporcionalidad con respecto a la función objetivo ordenando en forma ascendente los individuos del peor al mejor comparando sus desempeños como se extrajo de (Brindle, 1991).

4.2.4. Cruce

En la etapa de cruce, se utilizó un cruce basado en un punto, esto a partir de los individuos seleccionados en la etapa anterior como padres. El escogimiento del punto de cruce se realiza recorriendo aleatoriamente todos los espacios correspondientes a los bits que contienen las constantes optimizadas, que se llamarán G_p , G_d y G .

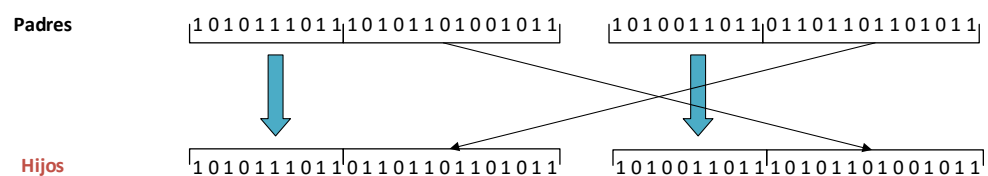


Figura 39. Cruce en un punto

En la figura 39 se observa que el punto de cruce se encuentra en la posición 10 del cromosoma y, mediante las flechas, se indica el intercambio de los genes a la derecha del punto de cruce del primer padre hacia el segundo hijo y se mantienen los genes a la izquierda del punto de cruce en el primer hijo; y de manera viceversa sucede con los genes del segundo padre.

4.2.5. Mutación

En esta etapa se realiza la mutación de los genes de los individuos originados, para efectuar dicha mutación se cambia o muta un bit de la información contenida dentro de los 24 bits que contienen los datos de las constantes a ser optimizadas. De igual manera que en la etapa de cruce, la posición del bit que muta se determina de manera aleatoria.

1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1
 ↓
 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1

Figura 40. Mutación de bit

Se puede visualizar en la figura 40 que el bit que cambia o muta es el de la posición 15. A continuación se realiza la evaluación de los individuos, en donde se procede a determinar cuál es el mejor individuo, es decir, el que presenta el mejor desempeño mediante una búsqueda dentro del espacio de soluciones obtenido.

4.2.6. Simulación del sistema de levitación magnética con controlador PD difuso optimizado mediante algoritmos genéticos

Para simular el sistema de levitación, se utiliza la función de transferencia obtenida del modelamiento del sistema de levitación realizado en el capítulo II, la simulación se efectúa en el entorno de Simulink de MATLAB.

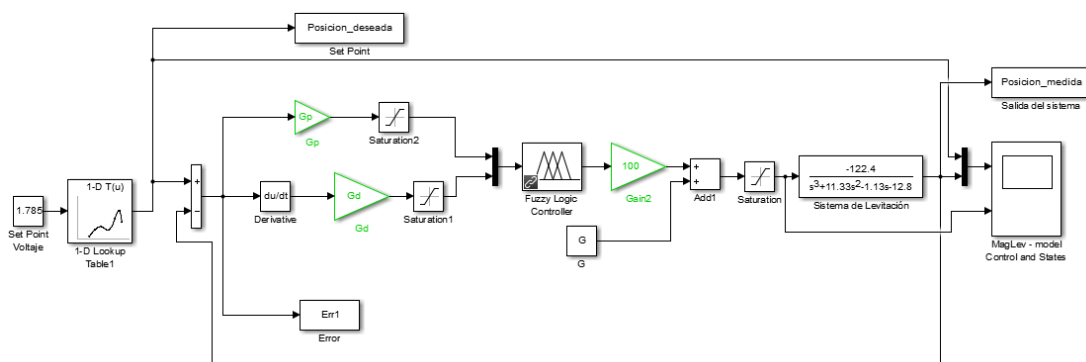


Figura 41. Simulación del sistema de levitación optimizado mediante algoritmos genéticos

En la figura 41 se visualiza las distintas etapas que se realizan en la simulación del sistema optimizado y que posteriormente se implementarán en el sistema físico, las cuales se describen a continuación:

- El uso del Lookup Table de la misma manera que en la simulación anterior.

- Se muestra el error y la derivada del error. Las ganancias son nombradas como G_p y G_d ; estas irán variando mientras se ejecuta el algoritmo genético hasta obtener los valores deseables.
- Se observa el controlador fuzzy diseñado anteriormente. Después se dispone una ganancia de 100 y el control mínimo viene dado por la ganancia G que se optimizará por el algoritmo genético.
- También se disponen de variables nombradas como Posicion_deseada, Posicion_medida y Err1, que corresponden al Set Point, la salida del sistema y el error respectivamente; variables que se envían al espacio de trabajo de MATLAB para poder operar con ellas en los programas de implementación del algoritmo genético.

Se realizaron varias pruebas para determinar el número idóneo de generaciones que se utilizarían, se mostrarán dos ejemplos, el primero parte de las pruebas preliminares, y el segundo ya con el mejor resultado presentado.

En la figura 42 se puede observar el desempeño dado por 10 generaciones.

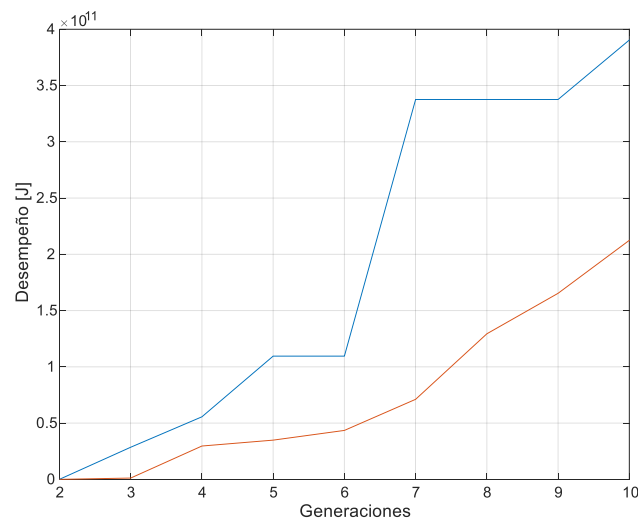


Figura 42. Desempeño vs generaciones

Los valores obtenidos después de la optimización realizada son:

$$G_p = 0.96$$

$$G_d = 0.08$$

$$G = 6.05$$

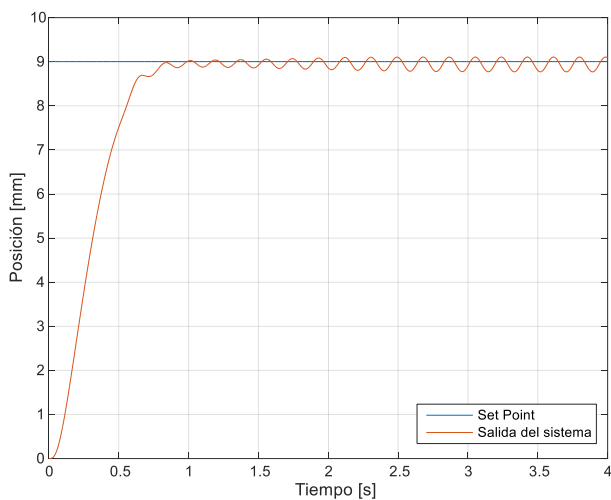


Figura 43. Set Point y salida del sistema 10 generaciones

En la figura 43 se observa el set point y la salida del sistema, se presentan todavía oscilaciones. En la figura 44 se muestra la señal de control resultante con la implementación del algoritmo genético.

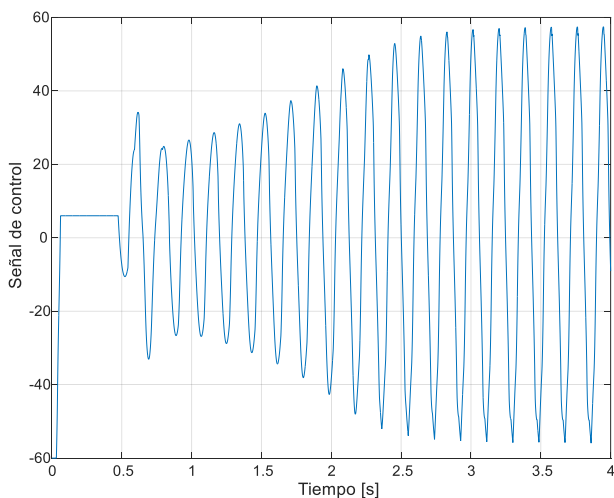


Figura 44. Señal de control con 10 generaciones

Se utilizaron finalmente 20 generaciones ya que no presentaron mucho tiempo de simulación y se obtuvieron resultados aceptables, presentando valores de desempeño como se muestra en la figura 45.

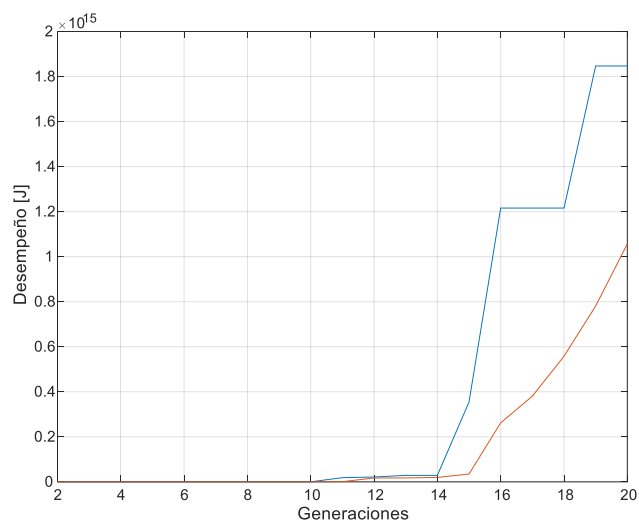


Figura 45. Desempeño vs generaciones

Los valores obtenidos después de la optimización realizada son:

$$Gp = 0.93$$

$$Gd = 0.095$$

$$G = 8.75$$

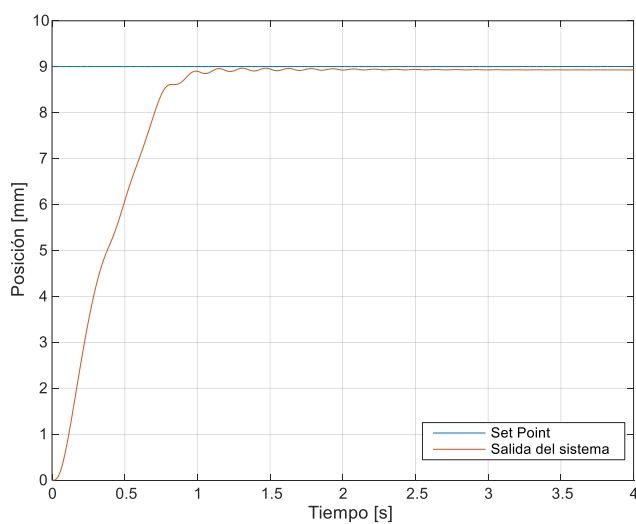


Figura 46. Set Point y salida del sistema 20 generaciones

En la figura 46 se puede observar el set point y la salida del sistema, se muestra que las oscilaciones se han reducido ostensiblemente con la aplicación del algoritmo genético y se ha logrado una optimización satisfactoria del desempeño del sistema. En la figura 47 se observa la señal de control resultante con la implementación del algoritmo genético.

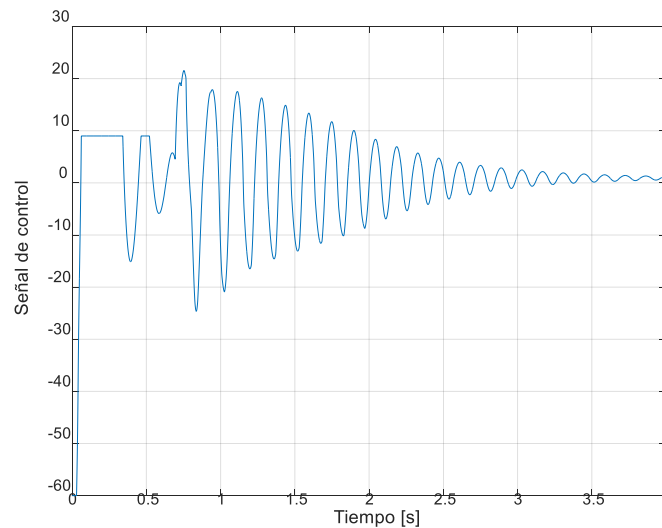


Figura 47. Señal de control 20 generaciones

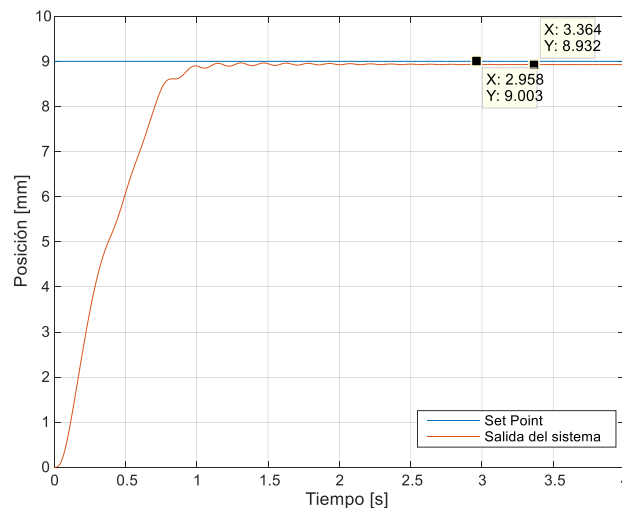


Figura 48. Error del sistema

En la figura 48 se muestra el error que tiene el sistema, dado que el sistema llega a estabilizarse en un punto sin oscilaciones la medición del error se facilita, simplemente se resta la salida, del set point.

$$e = (9 - 8.932)mm = 0.068mm$$

$$e = 0.75\%$$

4.3. Simulación del sistema de levitación magnética con controlador PD difuso optimizado mediante el algoritmo de Nealder-Mead

Para poder aplicar el algoritmo de optimización de Nealder-Mead se echó mano de la herramienta de MATLAB, denominada `fminsearch`, cuya función primordial es minimizar funciones, en este caso la función a ser minimizada es el error.

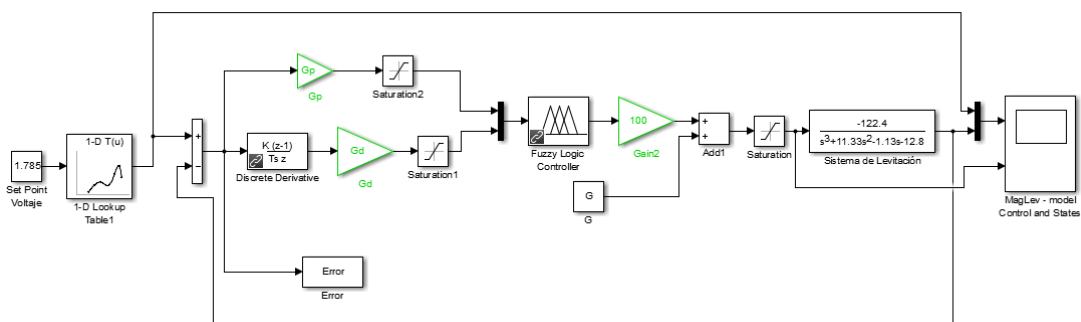


Figura 49. Simulación del sistema de levitación optimizado mediante Nealder-Mead

En la figura 49 se puede observar la programación realizada para la implementación del algoritmo de Nealder-Mead, la cual se describe a continuación:

- Se usa del Look up Table para transformar el voltaje obtenido del sensor en posición dada en mm. El Set Point es el mismo de las simulaciones anteriores.
- Se muestra el error y la derivada del error, como en las simulaciones anteriores, además se tienen las ganancias G_p y G_d , que se optimizarán mediante el algoritmo de Nealder-Mead.
- Se observa el controlador fuzzy, que es el mismo que se diseñó anteriormente. Después se coloca una ganancia de 100 y la ganancia G de control mínimo, que se optimizará por el algoritmo de Nealder-Mead.
- La variable de Err1 es el error, que es necesario para efectuar la optimización, ya que es el valor que se minimiza.

Se efectuaron varias pruebas para determinar el tiempo idóneo de simulación que se utilizaría, se mostrarán dos ejemplos, el primero parte de las pruebas preliminares, y el segundo ya con el mejor resultado obtenido.

Las ganancias que se obtuvieron con tiempo de simulación de 10 segundos son las siguientes:

$$G_p = 0.99$$

$$G_d = 0.32$$

$$G = 35.2$$

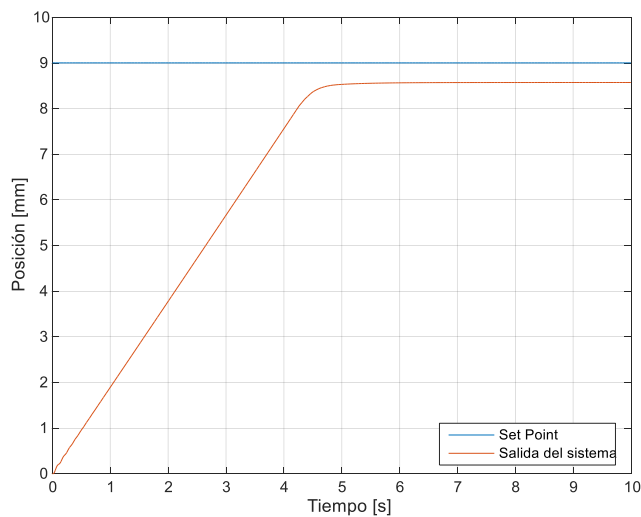


Figura 50. Set Point y salida del sistema con 10 seg. de simulación

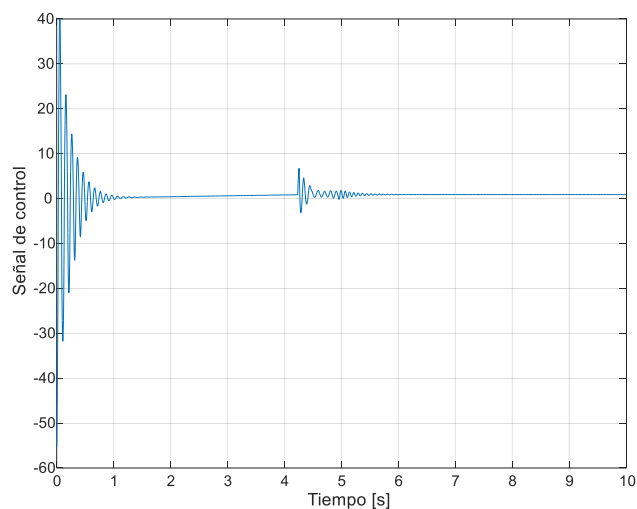


Figura 51. Señal de control con 10 seg. de simulación

En la figura 50 se observa la salida y el set point, mientras que en la figura 51, se muestra la señal de control, resultados obtenidos con 10 segundos de simulación.

Después de haber realizado las pruebas pertinentes, los mejores valores obtenidos después de la optimización realizada con 4 segundos de simulación son:

$$Gp = 0.982$$

$$Gd = 0.201$$

$$G = 9.03$$

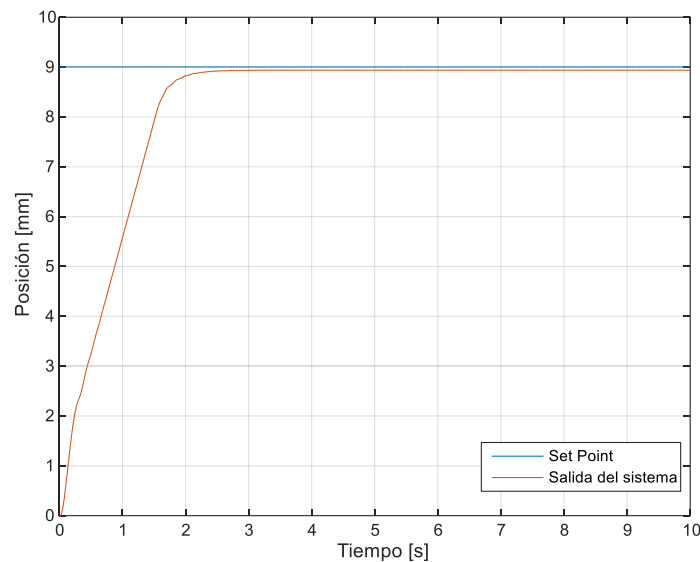


Figura 52. Set Point y salida del sistema con 4 seg. de simulación

En la figura 52 se puede observar el set point y la salida del sistema, se puede verificar que las oscilaciones han desaparecido inclusive en mayor medida que con los algoritmos genéticos. En la figura 53 se observa la señal de control obtenida con la implementación del algoritmo de Nealder-Mead.

En la figura 54 se observa el error que tiene el sistema, ya que el sistema no presenta oscilaciones la medición del error se realiza restando la salida, del set point.

$$e = (9 - 8.938)mm = 0.062mm$$

$$e = 0.68\%$$

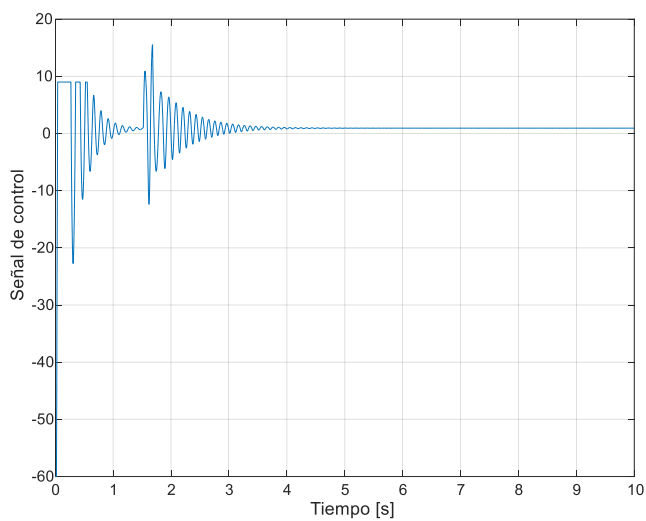


Figura 53. Señal de control con 4 seg. de simulación

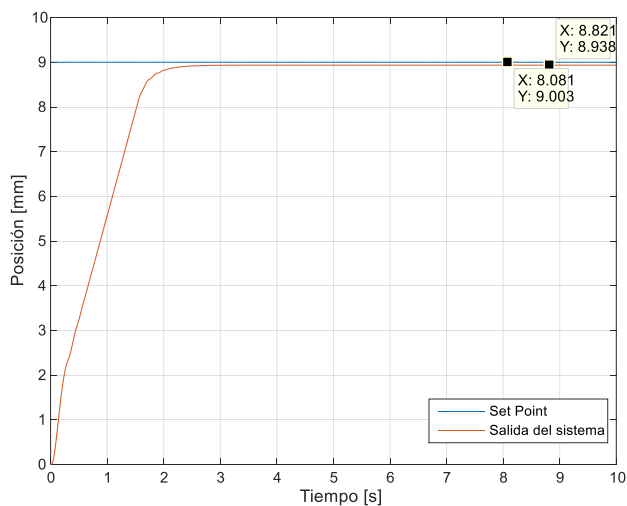


Figura 54. Error del sistema

Tabla 13
Comparación del error de cada método aplicado

Método	Error
Control Fuzzy	3.83%
Algoritmos genéticos	0.75%
Algoritmo de Nealder-Mead	0.68%

Se observa, según lo expuesto en la tabla 13, que el método que menor error proporcionó es el algoritmo de Nealder-Mead.

CAPÍTULO V

IMPLEMENTACIÓN DE LOS ALGORITMOS DE CONTROL EN EL SISTEMA FÍSICO

5.1. Implementación del controlador fuzzy PD en el sistema de levitación magnética

Para la implementación de todos los programas de implementación de los distintos controladores diseñados, en el sistema de levitación magnética se utilizó la herramienta Waijung Blockset provista por AIMAGIN CO., LTD.

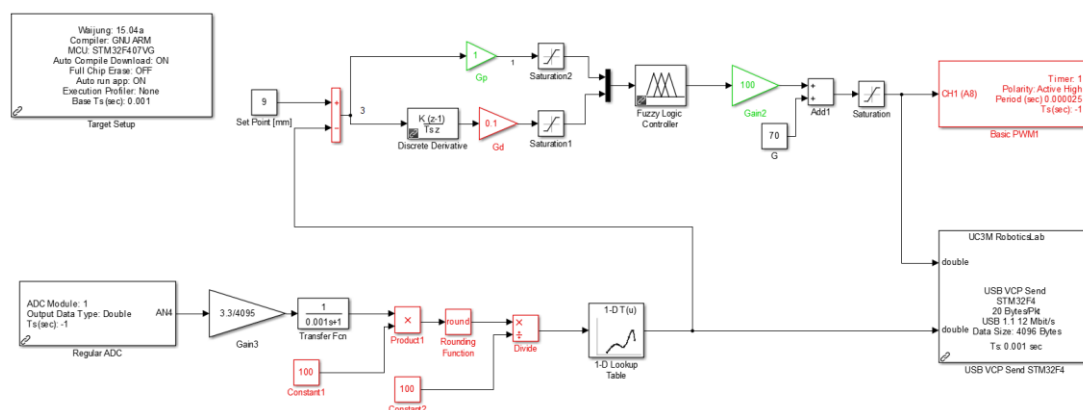


Figura 55. Programa implementado en el sistema de levitación magnética real

En la figura 55 se observa el programa que se carga en la tarjeta STM32F4, con los valores de las constantes que se determinaron en la simulación sin la aplicación de los algoritmos de optimización. Las distintas etapas y componentes que conforman el programa implementado se describen de la siguiente manera:

- Primeramente, se debe disponer del bloque de compilación de la herramienta Waijung (Target Setup), que está en la parte superior izquierda.
- El bloque del conversor analógico digital de la librería Waijung (Regular ADC).
- Se dispone de un filtro para reducir el ruido de la señal adquirida del sensor y se redondea para enviarla al Lookup Table, para convertir el voltaje en posición.

- Se coloca un bloque de envío por transmisión serial (USB VCP Send STM32F4), que se usa para el envío tanto de la señal del sensor, como de la señal de control.
- Los demás valores de ganancias, proporcional, derivativa y la de control mínimo, se disponen de la misma manera que en la simulación.
- Se utilizan el pin PA8 para la salida pwm y el pin PA4 para la adquisición de la señal del sensor.

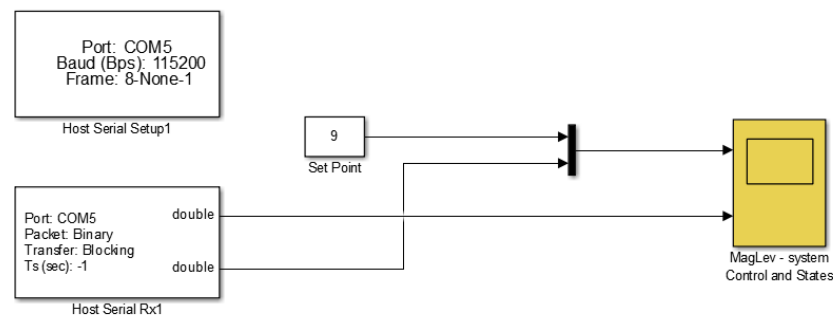


Figura 56. Programa de recepción de datos

Se necesita un programa adicional para la recepción de los datos que adquiere la tarjeta STM32F4, que se lo puede observar en la figura 56, y cuyos componentes se describen de la siguiente manera:

- Se tiene un bloque de configuración para la recepción serial (Host Serial Setup), donde se determina el puerto y la tasa de baudios.
- También se tiene un bloque de recepción (Host Serial Rx), que es el encargado de recibir las señales.
- Y finalmente se envían las señales a un scope para graficar las señales del sensor y de control. Se debe puntualizar que el programa de recepción es el mismo para la implementación de todos los algoritmos diseñados.

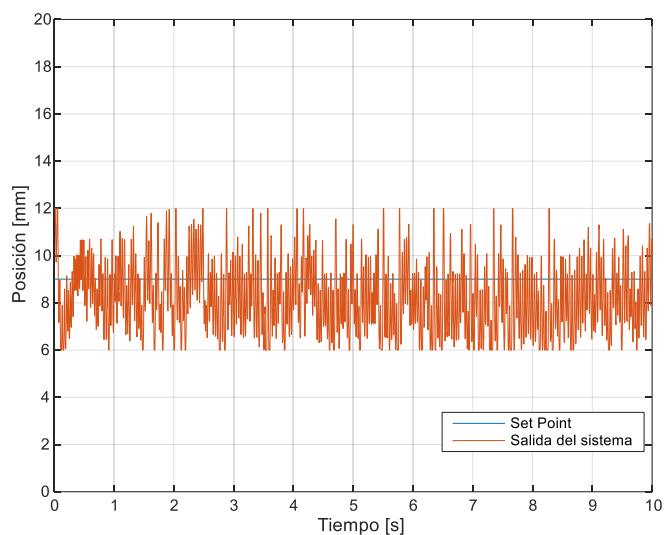


Figura 57. Salida del sistema de levitación real

En la gráfica 57 se visualiza la salida del sistema físico con el controlador fuzzy PD ya implementado en milímetros y el set point impuesto de 9 mm.

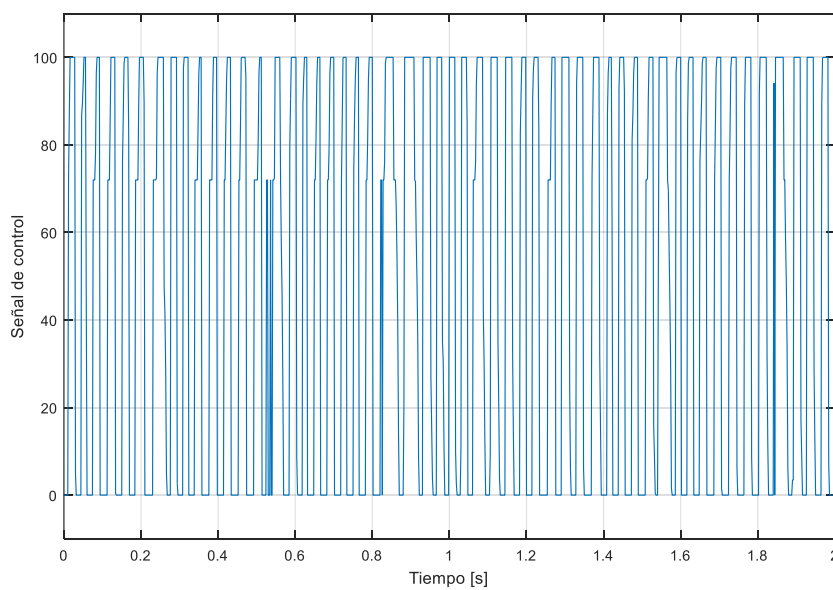


Figura 58. Señal de control

De la figura 58 se puede rescatar la señal de control enviada a través de la tarjeta STM32F4.

5.2. Implementación del controlador PD difuso en el sistema de levitación magnética, optimizado mediante algoritmos genéticos

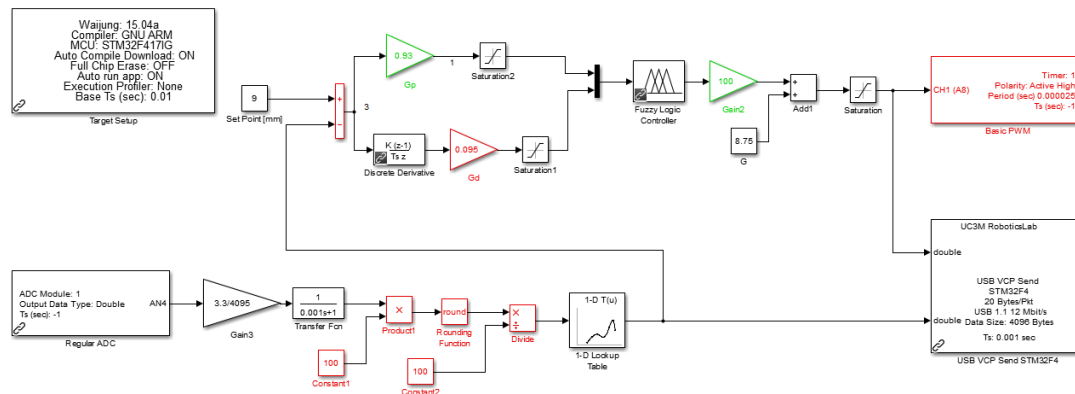


Figura 59. Programa implementado en el sistema de levitación magnética optimizado por algoritmos genéticos

En la figura 59 se observa el programa que se carga en la tarjeta STM32F4, con los valores de las constantes que se obtuvieron en la simulación aplicando el algoritmo genético diseñado para su optimización, referidas a las ganancias proporcional, derivativa y de control mínimo. Las distintas etapas y componentes que forman parte del programa implementado son los mismos descritos en el programa anterior. El programa de recepción es el mismo realizado para la implementación anterior. Para lograr una levitación más estable de la esfera se experimentó con el sistema real, se varió la ganancia G del control mínimo, siendo el valor óptimo 70.

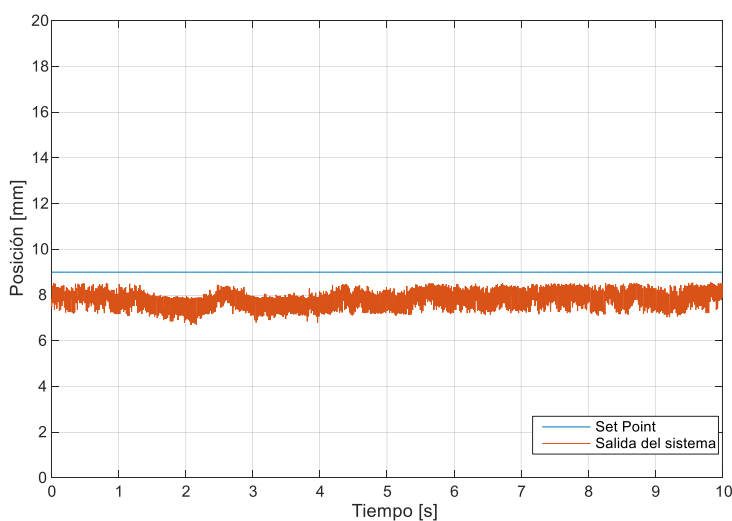


Figura 60. Salida del sistema de levitación real

En la gráfica 60 se observa la salida del sistema físico y el set point de 9 mm, con el controlador fuzzy PD optimizado por algoritmos genéticos. En la figura 61 se tiene la señal de control.

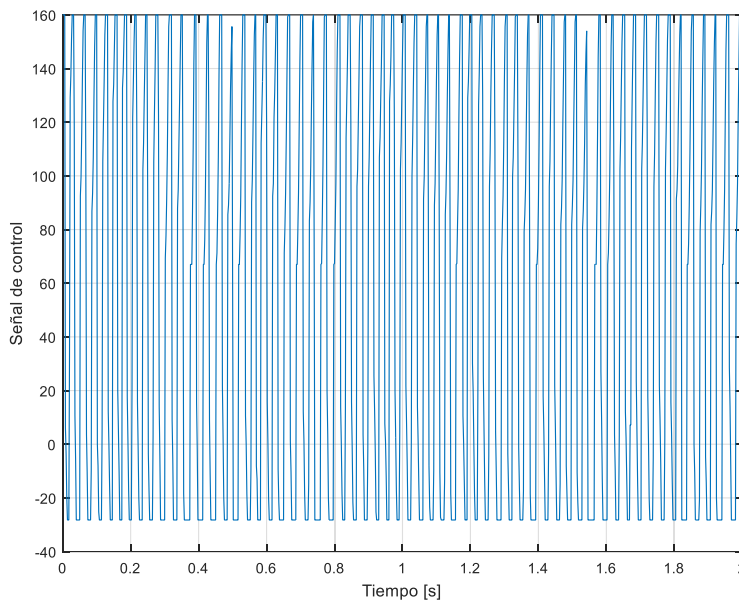


Figura 61. Señal de control

5.3. Implementación del controlador PD difuso en el sistema de levitación magnética optimizado mediante el algoritmo de Nealder-Mead

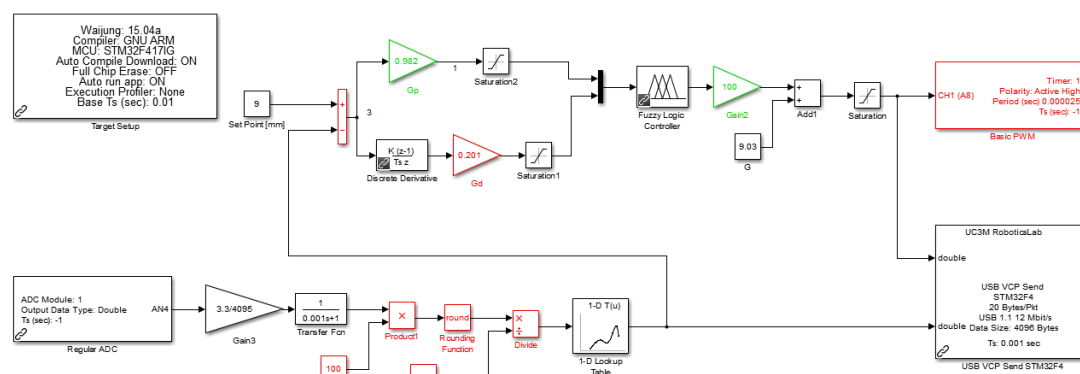


Figura 62. Programa implementado en el sistema de levitación magnética optimizado por el algoritmo de Nealder-Mead

En la figura 62 se puede observar el programa que se carga en la tarjeta STM32F4, con las constantes que se obtuvieron en la simulación aplicando el algoritmo de Nealder-Mead diseñado para su optimización, estas son las ganancias, proporcional,

derivativa y de control mínimo. Las etapas y componentes que constituyen el programa implementado son los mismos descritos en los programas anteriores. La recepción se efectúa de la misma manera que en las demás implementaciones. Para lograr una levitación más estable en la experimentación con el sistema físico, se varió la ganancia G que corresponde al control mínimo, siendo el valor óptimo el mismo que se implementó con los algoritmos genéticos que es 70.

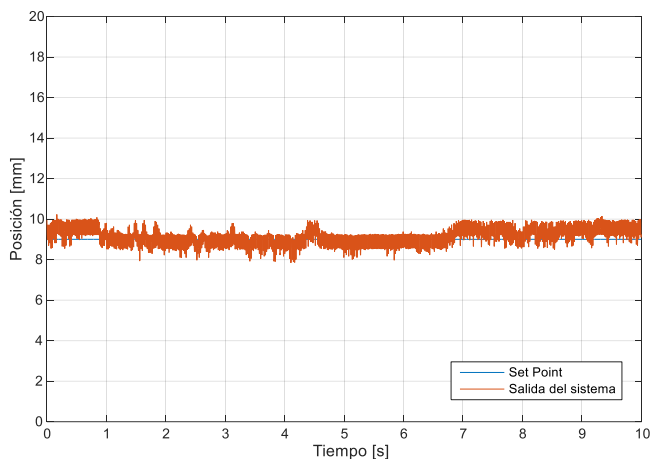


Figura 63. Salida del sistema de levitación real

De la gráfica 63 se observa la salida del sistema físico y el set point dados en milímetros, con el controlador fuzzy PD optimizado mediante el algoritmo de Neelder-Mead. En la figura 64 se visualiza la señal de control.

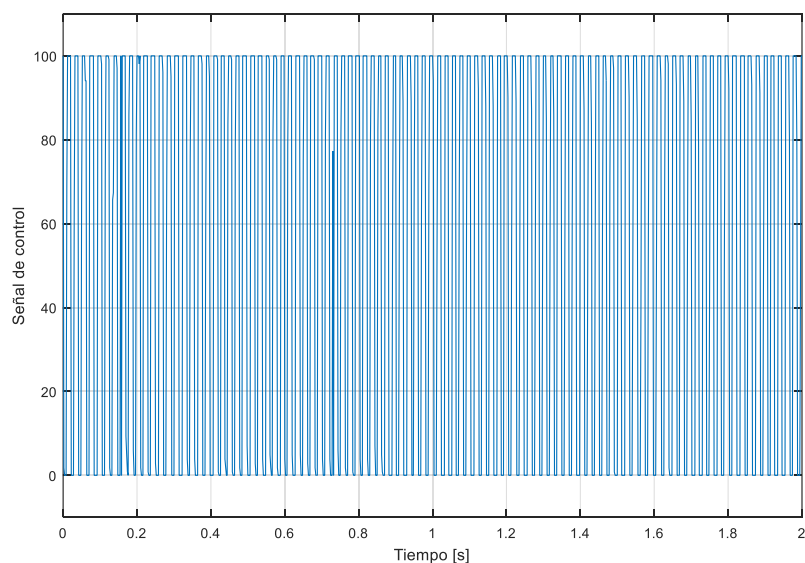


Figura 64. Señal de control

En la figura 65 se muestra el sistema de levitación ya implementado en su totalidad con la esfera levitando.

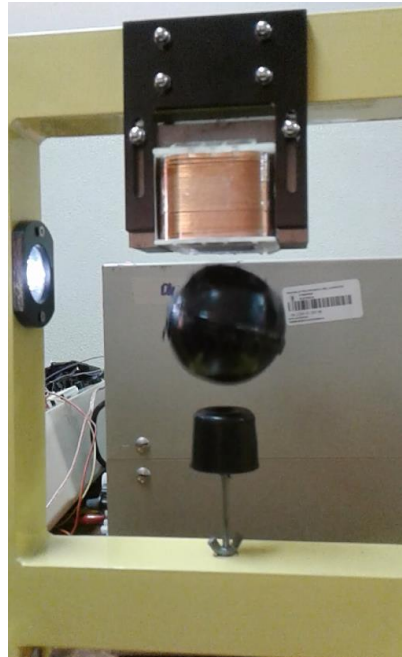


Figura 65. Sistema físico con esfera levitando

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

En lo que respecta al modelamiento, se puede bien inferir que la función de transferencia obtenida se asemeja en gran medida al sistema real, basándose en los resultados entregados especialmente de las simulaciones, además de la demostración de que en el sistema físico la esfera levita.

Se observó que, al generarse un campo electromagnético debido al electroimán implementado, este influía de manera visible en las señales adquiridas tanto del sensor de posición como del sensor de corriente, introduciendo ruido en dichas señales.

Se pudo observar claramente que al poner en marcha los algoritmos de optimización, se obtuvieron resultados ostensiblemente mejores, basados en la respuesta del sistema en torno a la salida en estado estacionario.

Una desventaja que se presenta al implementar el controlador fuzzy sin optimización, es que los parámetros definidos G_p y G_d y G , se tienen que ajustar de manera empírica, basada en los resultados que se van obteniendo con cada cambio de parámetro, lo que implica más tiempo de implementación.

Los algoritmos de optimización implementados, se presentan como una buena opción para la obtención de soluciones que mejoren la respuesta que el sistema muestra originalmente utilizando el control fuzzy.

Se ha realizado una implementación de los algoritmos genéticos para la optimización de una manera relativamente sencilla, mediante la minimización del error, incorporando una función de coste, que corresponde al error cuadrático medio, que muestra el desempeño del algoritmo sobre la respuesta del sistema.

De los resultados obtenidos a partir de la optimización con el algoritmo genético, se pudo dilucidar que el número de generaciones resulta un aspecto importante, siendo 20 el número de generaciones que se consideraron suficientes para alcanzar una optimización aceptable.

En lo referente al algoritmo de Neelder-Mead, se consideró al error en sí, como la función de coste, ya que es un método de minimización, y los parámetros a ser optimizados son G_p y G_d y G .

Se observó que los resultados presentados por el algoritmo de Nealder-Mead fueron mejores que los obtenidos aplicando los algoritmos genéticos, ya que en la simulación las oscilaciones se redujeron de forma definitiva y el error alcanzado fue menor; además que aplicados al sistema real se obtuvo un menor margen de error.

6.2. Recomendaciones

Se recomienda, para trabajos futuros el estudio de una posible implementación totalmente online de los algoritmos de optimización diseñados, ya que al ser en teoría métodos recursivos, tiene como uno de sus objetivos su uso en línea.

En lo que respecta a los algoritmos genéticos, es recomendable tomar en consideración el número de generaciones, dependiendo del tiempo de simulación que se tenga, se debe encontrar un equilibrio de acuerdo a los resultados que se obtengan del sistema, no se debe disponer de un número excesivo de generaciones ya que se desperdiciaría tiempo innecesariamente para encontrar un resultado óptimo.

Para poder reducir el tiempo de establecimiento se recomienda que en cualquiera de los dos métodos de optimización utilizados, especialmente en el algoritmo de Nealder-Mead, se reduzca el tiempo de simulación, ya que de esta manera se fuerza al sistema a estabilizarse con mayor rapidez, esto debido a que, al no ser multiobjetivo la implementación y ser considerado el error solamente como parte de la función de coste, no se toman en cuenta las demás especificaciones consideradas en las técnicas de control clásico.

Se recomienda que, para la etapa de potencia, se implemente un preactuador con mayor frecuencia de conmutación; ya que se requiere de una velocidad de respuesta muy grande a partir de 1 MHz, para así tener una mejora en la levitación de la esfera.

Debido a que se genera un campo electromagnético, se debe tener en cuenta el ruido que se puede inducir hacia las demás etapas que componen el sistema, tomando medidas de protección del cableado para contrarrestar los efectos de ruido, especialmente en la adquisición de la señal del sensor de posición.

BIBLIOGRAFÍA

- Alander, J. (1992). Computer Systems and Software Engineering. *6th Annual European Computer Conference*, pp. 65-70.
- Baker, J. (1987). Reducing bias and inefficiency in the selection algorithm. *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, (pp. 14-21).
- Bernal, J. L. (n.d.). *Aplicación de algoritmos genéticos al diseño óptimo de sistemas de distribución de energía eléctrica*. Universidad de Zaragoza, Zaragoza.
- Brindle, A. (1991). *Genetic algorithms for function optimization*. Universidad de Alberta, Alberta.
- Chamorro, H., Vladimir, B., & Trujillo, C. (2010). Diseño y simulación de un controlador PD difuso para el control de la velocidad de un motor de inducción. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal*, 73-74.
- De Jong, K. (1975). *An analysis of the behaviour of a class of genetic adaptive systems*. Universidad de Michigan, Michigan.
- Fogel, D. (2000). *Evolutionary Computation Basic Algorithms and Operators*. Philadelphia: Institute of Physics Publishing.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
- Gómez, J. C. (n.d.). *Fuzzy Control*. Universidad Tecnológica Nacional, Buenos Aires.
- Herreros, A. (s.f.). *Diseño de controladores robustos multiobjetivo por medio de algoritmos genéticos*. Universidad de Valladolid, Valladolid.
- Hollan, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.
- INTECO. (2014). *Magnetic Levitation Systems*. Obtenido de <http://www.inteco.com.pl/products/magnetic-levitation-systems/>
- Martínez, M., Sanchis, J., & Blasco, X. (2006). *Algoritmos genéticos aplicados al diseño de controladores robustos*. *Revista Iberoamericana Automática e Informática Industrial*, 39-51.
- Montejo, M. (s.f.). *Breve curso de lógica difusa y control difuso*. Obtenido de <http://redeya.bytemaniacos.com/electronica/cursos/fuzzy/fuzzy.htm>
- Nealder, A., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 308-313.
- Pérez, M., & Readman, M. (n.d.). *infoPLC*. Obtenido de <http://www.infopl.net/documentacion/210-motion-control-servos-mecatronica/1217-sistemas-de-servo-control-servomecanismos-digitales>
- Pilotta, E. (n.d.). *El método de Nealder-Mead para minimización irrestricta sin derivada*. Universidad Nacional de Córdoba, Córdoba.
- Ruge, I., & Alvis, M. (2009). Aplicación de los algoritmos genéticos para el diseño de un controlador PID adaptativo. *Red de Revistas de América Latina, el Caribe, España y Portugal*, 82-87.
- Sancho, F. (2016). *Algoritmos Genéticos y Computación Evolutiva*. Obtenido de <http://www.cs.us.es/~fsancho/?e=65>
- Schaffer, J., Caruna, R., & Eshelman, L. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, (pp. 51-60).

- Syswerda, G. (1991). Schedule optimization using genetic algorithms. *Handbook of Genetic Algorithms*, (pp. 332-349). New York.
- Tejada, G. (n.d.). *Tutorial de Lógica Fuzzy*. Universidad Nacional Mayor de San Marcos, Lima.