



INSTITUTO POLITÉCNICO NACIONAL

**CENTRO DE INVESTIGACIÓN Y DESARROLLO
DE TECNOLOGÍA DIGITAL**



MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES

**“CONTROL INTELIGENTE DE UN PÉNDULO INVERTIDO
MÓVIL”**

TESIS

**QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS EN SISTEMAS DIGITALES**

PRESENTA

Ing. Carlos Martín Magaña González

BAJO LA DIRECCIÓN DE

DR. ROBERTO SEPÚLVEDA CRUZ DR. OSCAR H. MONTIEL ROSS

Marzo 2016

TIJUANA, B.C., MÉXICO



INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de Tijuana, B.C. siendo las 11:00 horas del día 01 del mes de marzo del 2016 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CITEDI para examinar la tesis titulada:

Control inteligente de un péndulo invertido móvil.

Presentada por el alumno:

MAGAÑA

Apellido paterno

GONZÁLEZ

Apellido materno

CARLOS MARTÍN

Nombre(s)

Con registro:

A	1	4	0	8	1	4
---	---	---	---	---	---	---

aspirante de:

MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES

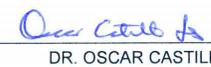
Después de intercambiar opiniones, los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de tesis


DR. ROBERTO SEPÚLVEDA CRUZ


DR. OSCAR HUMBERTO MONTIEL ROSS


DR. OSCAR CASTILLO LÓPEZ


DR. JUAN JOSÉ TAPIA ARMENTA


M. en C. ANDRÉS CALVILLO TÉLLEZ

PRESIDENTE DEL COLEGIO DE PROFESORES


DRA. MIREYA SARAI GARCÍA VÁZQUEZ




INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de Tijuana, Baja California, el día 1 de **marzo** del año **2016**, el (la) que suscribe el **Carlos Martín Magaña González** alumno (a) del Programa de **MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES** con número de registro **A140814**, adscrito al **CENTRO DE INVESTIGACIÓN Y DESARROLLO DE TECNOLOGÍA DIGITAL**, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección del (de la, de los) **Dr. Roberto Sepúlveda Cruz** y **Dr. Oscar H. Montiel Ross**, cede los derechos del trabajo titulado **“CONTROL INTELIGENTE DE UN PÉNDULO INVERTIDO MÓVIL”**, al **Instituto Politécnico Nacional** para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección **Av. Instituto Politécnico Nacional 1310, Colonia Nueva Tijuana, Baja California 22435, México**, o a la dirección electrónica: **cmagana@citedi.mx**, **rsepulvedac@ipn.mx**, **oross@ipn.mx**, **posgrado@citedi.mx**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Carlos Martín Magaña González

Nombre y firma del alumno(a)

“Aprende como si fueras a vivir toda la vida, y vive como si fueras a morir mañana.”

-Charlie Chaplin

Dedicatoria

A mi madre y a mi hermano que siempre han estado a mi lado en todo momento, gracias a su paciencia, amor y dedicación, gracias a ellos he logrado llegar tan lejos.

Agradecimientos

Quiero expresar mi gratitud a mis directores de tesis, al Dr. Roberto Sepúlveda Cruz y al Dr. Oscar Montiel Ross, por su apoyo incondicional, sus valiosos consejos y sugerencias que me ayudaron a concretar con éxito este trabajo.

A los integrantes de la comisión revisora de tesis, Dr. Juan José Tapia Armenta, Dr. Oscar Castillo López y M.C. Andrés Calvillo Téllez por sus valiosas aportaciones y sugerencias para enriquecer este trabajo de tesis.

Al Centro de Investigación y Desarrollo de Tecnología del Instituto Politécnico Nacional (CITEDI-IPN) por brindarle sus instalaciones y otorgarme la gran oportunidad de seguir creciendo profesionalmente.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado a lo largo de estos dos años, que me permitió obtener el grado de Maestro en Ciencias en Sistemas Digitales.

A todos mis amigos y compañeros de CITEDI por brindarme su gran amistad y compartir increíbles momentos, que estoy seguro formaron lazos perdurables.

Siempre estaré enormemente agradecido a mi madre que con su ímpetu, esfuerzo, dedicación y amor forjó en mí un gran sentido de la perseverancia, esfuerzo y convicción para lograr todas mis metas propuestas y crecer como persona. De igual manera quiero agradecer a mi hermano que me enseñó que uno es tan grande como sus sueños, que el coraje y la determinación son clave esencial para el éxito. Gracias a ustedes por enseñarme a siempre seguir a adelante y nunca rendirme.

Control Inteligente de un Péndulo Invertido Móvil

Resumen

En este trabajo de tesis, se presenta el control de estabilidad de un péndulo invertido móvil en base a los conceptos de inteligencia artificial. Se presenta el diseño del controlador difuso mediante la Síntesis Difusa de Lyapunov, estructurando el conjunto de Base de Reglas que conforman al Controlador por Lógica Difusa, que garantiza la estabilidad del sistema. Sin embargo esta metodología no proporciona una distribución para las Funciones de Membresía, debido a esto una sintonización manual a base de prueba y error es necesaria para obtener una respuesta aceptable del sistema. Es por eso que se hace la integración de los Algoritmos Genéticos para garantizar una distribución óptima de las funciones de membresía. A esta integración se le denomina Sistema Genético Difuso, el cual posee capacidades de aprendizaje, adaptación y permite optimizar las funciones de membresía.

Se presentan los experimentos obtenidos en base a la propuesta de solución, realizando una comparación entre la sintonización manual y por un Sistema Genético Difuso de las funciones de membresía que garantizan la estabilidad del sistema. En una primera etapa el sistema es simulado para observar su comportamiento ante los distintos tipos de sintonización de las funciones de membresía y posteriormente se realiza el mismo procedimiento en un prototipo didáctico de péndulo invertido móvil diseñado, con el fin de contrastar los datos obtenidos en simulación. Los resultados demuestran las capacidades de la propuesta de solución, cumpliendo con su objetivo principal en la corrección del error angular del péndulo invertido móvil.

Palabras Clave: Péndulo Invertido Móvil, Lógica Difusa, Síntesis Difusa de Lyapunov, Algoritmo Genético Difuso.

Intelligent Control of a Two Wheel Inverted Pendulum

Abstract

In this thesis work, the stability control of a Two Wheel Inverted Pendulum based on artificial intelligence techniques is presented. Fuzzy Controller design is proposed based on Lyapunov Fuzzy Synthesis, structuring the Rule-Base set that shape the Fuzzy Logic Controller, ensuring system stability. However, this approach does not provide a distribution for the Membership Functions, because of this a manual tuning through trial and error is required to obtain an acceptable system response. It is for this reason that the integration of Genetic Algorithms to ensure an optimal distribution of the membership functions is proposed. This integration is referred to as Genetic Fuzzy System, which has learning capabilities, adaptation and it is able to optimize the membership functions.

The experimental results obtained through the proposed solution are presented, making a comparison between manual tuning and a Genetic Fuzzy System membership functions to ensure system stability. In a first stage by simulation and then in a didactic prototype Two Wheel Inverted Pendulum designed. The results demonstrate the capabilities of the proposed solution, fulfilling its main objective in correcting the angular error of the Two Wheel Inverted Pendulum.

Keywords: Two Wheel Inverted Pendulum, Fuzzy Logic, Lyapunov Fuzzy Synthesis, Genetic Fuzzy System.

Contenido

1	Introducción	1
1.1	Antecedentes	3
1.2	Planteamiento del problema	6
1.3	Justificación	6
1.4	Hipótesis	7
1.5	Objetivos	7
1.5.1	Objetivo general	7
1.5.2	Objetivos específicos	7
1.6	Aportaciones	8
1.7	Trabajos derivados	8
1.8	Organización del documento	9
2	Marco Teórico	10
2.1	Descripción del sistema: Péndulo invertido móvil.	10
2.1.1	Modelo matemático: Péndulo invertido sobre dos ruedas.	12
2.1.2	Representación en espacio de estados	14
2.2	Lógica difusa	15
2.2.1	Conjuntos difusos	16
2.2.2	Variables lingüísticas	18
2.2.3	Sistema a base de reglas	18
2.2.4	Sistema de control difuso	18
2.2.5	Controlador por lógica difusa tipo Mamdani	19
2.2.6	Controlador por lógica difusa tipo Takagi-Sugeno (T-S)	20
2.2.7	Funciones de membresía	20
2.2.8	Función de membresía triangular	21
2.2.9	Función de membresía trapezoidal	21
2.2.10	Función de membresía Gaussiana	21
2.3	Síntesis difusa de Lyapunov	21
2.4	Algoritmos genéticos	24

2.4.1	Algoritmo genético binario	25
2.4.2	Componentes de un algoritmo genético binario	26
2.4.3	Selección de variables y función de costo	27
2.4.4	Decodificación y codificación de variables	28
2.4.5	La población	29
2.4.6	Selección natural	29
2.4.7	Apareamiento	31
2.4.8	Mutaciones	32
2.4.9	Convergencia	32
2.5	Algoritmo genético continuo	33
2.5.1	Variables y función de costo	33
2.5.2	Población inicial	34
2.5.3	Selección natural	34
2.5.4	Apareamiento	34
2.5.5	Mutaciones	35
2.5.6	Convergencia	35
2.6	Algoritmo genético difuso	35
2.6.1	Proceso de sintonización genético	37
2.6.2	Sintonización de las funciones de membresía	37
2.6.3	Sintonización de las reglas difusas	37
2.6.4	Extracción de datos	38
2.6.5	Ventajas del GFS	38
3	Metodología	40
3.1	Propuesta de solución	41
3.2	Descripción del sistema: Diagrama a bloques.	42
3.3	Descripción del sistema: Prototipo de péndulo invertido móvil	43
3.4	Descripción de los componentes	46
3.4.1	Arduino Nano	46
3.4.2	MPU-6050	48
3.4.3	Puente H (DRV8833)	50
3.4.4	Motores	51
3.5	Sistema de control y criterio de evaluación del controlador difuso	51
3.6	Diseño del controlador difuso mediante síntesis difusa de Lyapunov	54
3.7	Funcionamiento del sistema genético difuso	57
4	Experimentos y análisis de resultados	60

4.1	Simulación	61
4.2	Respuesta del ángulo: Simulación	65
4.3	Respuesta del ángulo sin perturbaciones: Prototipo didáctico	68
4.4	Respuesta del ángulo ante perturbaciones: Prototipo didáctico	72
5	Conclusiones y trabajo futuro	76
5.1	Conclusiones	76
5.2	Trabajo futuro	77

Índice de figuras

1.1	Una clasificación de robots [1]	2
2.1	Sistema de péndulo invertido móvil. Imagen tomada de [2]	11
2.2	Sistema de péndulo invertido sobre dos ruedas. Imagen tomada de [3]	11
2.3	Modelo en sistemas de coordenadas del cuerpo del péndulo invertido	12
2.4	Modelo de péndulo invertido. Imagen tomada de [4]	13
2.5	Modelo de las ruedas. Imagen tomada de [4]	13
2.6	Números difusos	16
2.7	Funciones de membresía	17
2.8	Sistema de control difuso	19
2.9	Analogía entre biología genética y algoritmo genético binario	26
2.10	Diagrama a bloques de algoritmo genético binario	27
2.11	Diagrama a bloques del cruce de cromosomas	31
2.12	Diagrama a bloques de algoritmo genético continuo	33
2.13	Taxonomía de los GFS	36
2.14	Funciones de membresía predefinidas	37
2.15	Sintonización de las funciones de membresía	38
2.16	Extracción de datos: Sistema Difuso a un cromosoma	39
3.1	Diagrama a bloques de la propuesta de solución	41
3.2	Diagrama a bloques del sistema	42
3.3	Distribución de componentes	43
3.4	Plástico utilizado	44
3.5	Mediciones del modelo	44
3.6	Modelo real del prototipo de péndulo invertido móvil: Vista Frontal	45
3.7	Modelo real del prototipo de péndulo invertido móvil: Vista Lateral	45
3.8	Esquemático: Arduino Nano.	47
3.9	Orientación de los ejes de sensibilidad y polaridad de rotación.	49
3.10	Dimensiones del chip	49
3.11	Dimensiones del puente H.	50

3.12 Sistema de Control	52
3.13 Funcionamiento del Péndulo Invertido Móvil	53
3.14 Controlador difuso: Sintonización Manual	55
3.15 Cruce de los individuos	59
4.1 Controlador difuso: Sintonización manual. La distribución de las funciones de membresía se encuentran en grados.	63
4.2 Controlador difuso: Tiempo de asentamiento, sintonización por GFS	65
4.3 Controlador difuso: Tiempo de asentamiento, sintonización manual	66
4.4 Controlador difuso: Sintonización manual y por GFS, impulso positivo	67
4.5 Controlador difuso: Sintonización manual y por GFS, impulso negativo	67
4.6 Controlador difuso: Sintonización manual y por GFS	68
4.7 Controlador difuso: Sintonización manual y por GFS (acercamiento)	69
4.8 Controlador difuso: Sintonización manual	69
4.9 Controlador difuso: Sintonización manual (acercamiento)	70
4.10 Controlador difuso: Sintonización por GFS	71
4.11 Controlador difuso: Sintonización por GFS (acercamiento)	71
4.12 Controlador difuso ante perturbaciones: Sintonización manual	72
4.13 Controlador difuso ante perturbaciones (impulso negativo): Sintonización manual	72
4.14 Controlador difuso ante perturbaciones (impulso positivo): Sintonización manual	73
4.15 Controlador difuso ante perturbaciones: Sintonización por GFS	74
4.16 Controlador difuso ante perturbaciones (impulso positivo): Sintonización por GFS	74
4.17 Controlador difuso: Sintonización manual	75

Índice de tablas

3.1	Características básicas: Arduino Nano	46
3.2	Especificaciones generales	50
3.3	Características básicas: Motores	51
4.1	Características del hardware y software utilizados, para realizar los experimentos	61
4.2	Máximos y mínimos tiempos de asentamiento t_s (<i>seg</i>) del ángulo, media, mediana, desviación estándar y tiempo de ejecución, obtenidos del algoritmo genético difuso evaluado bajo distintas generaciones y número de individuos.	62
4.3	Parámetros empíricos del sistema: Comportamiento del ángulo (grados), sin perturbaciones	68
4.4	Parámetros empíricos del sistema: comportamiento del ángulo (grados), ante perturbaciones	73

Lista de Acrónimos

AI	Inteligencia Artificial, del inglés <i>Artificial Intelligence</i>
ANFIS	Sistemas Neuro-Difusos, del inglés <i>Adaptive Network-Based Fuzzy Inference System</i>
DOF	Grados de Libertad, del inglés <i>Degrees Of Freedom</i>
EA	Algoritmos Evolutivos, del inglés <i>Evolutionary Algorithms</i>
FIS	Sistema de Inferencia Difusa, del inglés <i>Fuzzy Inference System</i>
FLC	Controlador por Lógica Difusa, del inglés <i>Fuzzy Logic Controller</i>
FRBS	Sistema basado en reglas difusas, del inglés <i>Fuzzy Rule-Base System</i>
GA	Algoritmos Genéticos, del inglés <i>Genetic Algorithms</i>
GFS	Sistemas Genético-Difuso, del inglés <i>Genetic Fuzzy Systems</i>
GFSBS	Sistemas Genético-Difuso a Base de Reglas, del inglés <i>Genetic Fuzzy Rule-Based system</i>
KB	Base de Conocimiento, del inglés <i>Knowledge Base</i>
LFS	Síntesis Difusa de Lyapunov, del inglés <i>Lyapunov Fuzzy Synthesis</i>
MF	Función de Membresía, del inglés <i>Membership Function</i>
RB	Base de Reglas, del inglés <i>Rule Base</i>
SC	Cómputo Suave, del inglés <i>Soft Computing</i>
TWIP	Péndulo invertido móvil, del inglés <i>Two Wheel Inverted Pendulum</i>

Capítulo 1

Introducción

El término robot que deriva del checo *robota* (“trabajo”) fue introducido a nuestro vocabulario por primera vez por el dramaturgo Karel Capek en el año de 1920 en su obra *Rossum’s Universal Robots* [5]. Desde entonces el término se ha aplicado en una gran variedad de dispositivos mecánicos, prácticamente cualquier mecanismo que opere con algún grado de autonomía, por lo general bajo el control de una computadora. Una definición actual de robótica la podemos encontrar en **Oxford Dictionarie** [6] que define un robot como: *Una máquina capaz de llevar a cabo una serie de acciones complejas automáticamente, especialmente aquellas programadas por computadora.*

La robótica es un campo relativamente joven de la tecnología moderna y el buen entendimiento y el desarrollo de aplicaciones de la robótica están condicionadas al conocimiento de diversas disciplinas entre las que destacan la ingeniería eléctrica, ingeniería mecánica, ingeniería industrial, economía, matemáticas y ciencias computacionales. Capek visualizo a los robots como los trabajadores perfectos, un trabajador incansable que no se queja, no necesita seguro médico y no exige un cheque de pago. Los robots son capaces de realizar tareas repetitivas de forma más rápida, barata y precisa que los seres humanos y las aplicaciones para los robots no sólo se limitan a aquellos trabajos industriales donde el robot reemplaza a un ser humano.

Existen muchas otras aplicaciones en el área de la robótica donde el ser humano es poco práctico o no deseable, entre ellas la exploración bajo la superficie marítima, exploración planetaria, recuperación y reparación de satélites, la desactivación de explosivos y el trabajo en ambientes radiactivos. Los robots generalmente están presentes en alguna de las categorías mostradas en la Figura 1.1.

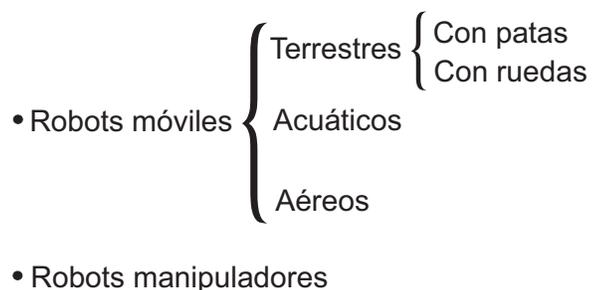


Figura 1.1: Una clasificación de robots [1]

Actualmente los robots manipuladores son comúnmente empleados en el sector industrial, donde sus principales ventajas recaen en la reducción de costos de producción, el incremento de la precisión, calidad y productividad, así como en actividades peligrosas para operadores humanos. Dentro del conjunto de plataformas que brinda la industria, las posibilidades que ofrecen los robots móviles son muy bastas.

La robótica móvil caracteriza a los robots por su capacidad para desplazarse, basados en carros o plataformas y dotados de un sistema locomotor de tipo rodante. Esto brinda la posibilidad de operarlos vía control remoto, mediante programación para entornos estáticos o controlados. Por otra parte la idea de extraer información del medio en que desarrollan su actividad mediante un conjunto de sensores y alterar su comportamiento de forma dinámica, le permite al robot realizar tareas mucho más complejas, característica atribuida a los sistemas autónomos. Esto permite el desarrollo de vehículos no tripulados capaces de interactuar con el medio que los rodea, tomar decisiones básicas y realizar una acción.

En este trabajo de tesis se aborda el problema de estabilización, el cual es un factor relevante en la robótica móvil y los sistemas autónomos, previniendo a los robots de caer y dañarse. El problema será tratado con técnicas emergentes del cómputo inteligente y aplicadas a un sistema del tipo subactuado conocido como péndulo invertido móvil (TWIP, *por sus siglas en inglés*) con el fin de probar el algoritmo de control propuesto.

1.1. Antecedentes

En la literatura existe una gran variedad de estudios referentes a la robótica, frecuentemente utilizados en la validación de modelos experimentales y eficiencia de las técnicas emergentes de control [2]. El péndulo invertido [7] es considerado uno de los sistemas robóticos más simples, con tan sólo un cuerpo rígido y una articulación, sin embargo se ha posicionado como uno de los sistemas más utilizados como punto de referencia en la validación de eficiencia y rendimiento de algoritmos de control no lineal [8].

Existen diversas variaciones del péndulo invertido, las cuales son bastante atractivas para ser analizadas bajo diferentes esquemas de control. Algunos de los más comunes son el Péndulo de Rueda Inercial [2], consta de una barra con una rueda giratoria acoplada a un actuador en su extremo libre. El par aplicado por el actuador se utiliza para controlar el sistema, el cual es provisto de sensores de movimiento que entregan la posición relativa del ángulo para la rueda de reacción y la posición angular de la barra.

El Péndulo de Furuta [9, 10], este péndulo se encuentra equipado de un brazo acoplado a un actuador que determina su posición angular y una barra en su extremidad que determina la posición angular del péndulo. Acrobot [11, 12] es un robot planar compuesto por dos articulaciones, cada una provista de actuadores que controlan la posición angular mediante el par aplicado utilizando sensores de movimiento. El robot llamado Pendubot [13, 14] es similar al anterior, sin embargo cuenta con tan solo un actuador ubicado en la primera articulación, mientras que la segunda se mueve libremente.

El péndulo invertido sobre carro [8, 2], se compone de un carro ubicado en la parte inferior como base del sistema, un péndulo y un riel como guía colocado en sus ruedas que definen la posición del carro. El péndulo se encuentra sujeto justo en el centro de gravedad del carro mediante una bisagra, lo que genera un movimiento vertical al riel. El péndulo invertido móvil o sobre dos ruedas [4, 3] consta de dos actuadores ubicados en su base, unidos de manera perpendicular al péndulo. Este sistema es equipado de sensores de movimiento que determinan su ángulo de inclinación, rotación y desplazamiento.

El sistema de péndulo invertido móvil posee las características de múltiples variables de orden superior, no lineal, fuerte acoplamiento e inestabilidad, lo que se considera como un gran desafío para las técnicas clásicas de control como los controladores PID [2]. Sin embargo desde hace varias décadas se han empleado diferentes metodologías para

el diseño de controladores [2, 8], entre las que destacan las técnicas de optimización y las técnicas de cómputo suave tales como los algoritmos genéticos, las redes neuronales, la lógica difusa y los controladores híbridos [2, 15, 16, 17].

Desde que los sistemas difusos se establecieron por primera vez en el año de 1965 en la propuesta de L. A. Zadeh [18], diversos estudios han dado lugar a la rápida evolución de la teoría de los conjuntos difusos y han demostrado su capacidad para resolver diferentes tipos de problemas en diversas aplicaciones [19, 20], entre las que destacan el diseño de controladores no lineales para sistemas de múltiples variables. Una gran cantidad de investigaciones sobre el control de sistemas de tipo péndulo invertido mediante el uso de sistemas de control difuso han sido propuestas. Margaliot [21] mostró un nuevo enfoque para determinar la estructura de controladores difusos mediante la combinación de la síntesis difusa de Lyapunov (SDL), utilizando el péndulo invertido como modelo experimental.

El interés por complementar los sistemas difusos mediante técnicas de optimización que le otorguen al sistema capacidades de adaptación y aprendizaje ha ido en aumento [22]. Los sistemas Neuro Difusos (ANFIS) [17] y Sistema Genético Difuso (GFS) [23] son considerados como los controladores híbridos más eficaces en el marco del cómputo suave. Uno de los enfoques más populares dentro de los controladores híbridos, corresponde a la unión entre la lógica difusa y los algoritmos genéticos [24, 25]. Un GFS es básicamente un sistema difuso aumentado por un proceso de aprendizaje basado en un algoritmo genético (GA). Los GA son algoritmos de búsqueda, basados en la genética, que proporcionan robustas capacidades de búsqueda en espacios complejos y por lo tanto ofrecen un enfoque válido para problemas que requieren procesos de búsqueda eficientes [15].

Los procesos de aprendizaje genéticos cubren diferentes niveles de complejidad de acuerdo con los cambios estructurales producidos por el algoritmo [26], desde el caso más simple de optimización, hasta el nivel más alto de complejidad de aprendizaje del conjunto de reglas difusas. La optimización de parámetros ha sido el enfoque utilizado para adaptar una amplia gama de sistemas difusos [27, 28]. En la literatura se muestra que los tipos más prominentes de GFS son los sistemas genético difuso a base de reglas (GFSBS) [23, 17], cuyo proceso genético aprende o sintoniza diferentes componentes de un sistema basado en reglas difusas (FRBS).

Hoy en día, varios productos comerciales, tales como lavadoras, cámaras, aparatos de

aire acondicionado y vehículos, son a menudo fabricados bajo los esquemas de controladores por lógica difusa [19] y basándose en el principio de estabilización del péndulo invertido se han realizado muchos desarrollos tecnológicos que han dado lugar distintas aplicaciones prácticas [20]. El diseño e implementación del sistema de péndulo invertido móvil se ha visto reflejado recientemente en un producto comercial conocido como Segway [8, 19, 29], el cual ha sido ampliamente reconocido como un poderoso vehículo de transporte personal.

Se construye a partir de componentes muy sofisticados y de alta calidad, como un servomotor brushless con imanes de neodimio, una caja de cambios de precisión, batería de níquel-metal hidruro, ruedas a base de sílice, un procesador de señal digital como controlador principal, controladores de motor, seis giroscopios, y varios accesorios de seguridad. En contraste con este tipo de vehículos de transporte de alto costo, muchos investigadores han presentado tecnología de sistemas auto-balanceables de bajo costo [8, 2, 20, 28]. Debido a su sencilla estructura, funcionamiento estable, alta eficiencia energética y la capacidad de adaptarse al medio en el que se encuentran, el panorama de aplicaciones prácticas comprende tanto el ámbito militar como civil.

1.2. Planteamiento del problema

Los robots son sistemas mecánicos y para conseguir controlar sus movimientos es necesario encontrar un modelo matemático que describa las dinámicas del sistema. Los sistemas convencionales de control son diseñados utilizando modelos matemáticos de sistemas físicos, por lo tanto es necesario obtener las dinámicas del sistema utilizando las leyes de la física que lo gobiernan.

La teoría de control es un campo que trata con disciplinas y métodos que nos guían a la obtención del control de un sistema. En las últimas décadas, se han empleado diferentes metodologías para el diseño de controladores y en el análisis de sistemas complejos que involucran múltiples variables, entre las que destacan las técnicas de cómputo suave como la Lógica Difusa y las técnicas de optimización como lo son las Redes Neuronales y los Algoritmos Genéticos. Estas metodologías han facilitado el diseño de controladores con capacidades de adaptación y el aprendizaje, capaces de controlar un sistema del tipo no lineal.

En la actualidad los principales logros realizados corresponden al estudio de sistemas robóticos subactuados, los cuales han dado lugar a distintas aplicaciones prácticas como los vehículos de transporte. Éstos requieren de una total comprensión del sistema para garantizar la seguridad del usuario y debido a su complejidad las técnicas emergentes del cómputo inteligente son una propuesta conveniente para atacar el problema.

1.3. Justificación

En esta investigación se busca la implementación de un Sistema Genético Difuso, con el fin de probar su eficiencia y desempeño en sistemas aplicados en la vida real, en este caso un péndulo invertido móvil, que asemeja los vehículos sobre dos ruedas que se utilizan como medio de transporte y necesitan garantizar la seguridad del usuario.

1.4. Hipótesis

Se asegura la estabilidad alrededor del punto de equilibrio de un péndulo invertido móvil al emplear la síntesis difusa de Lyapunov en el diseño de un controlador difuso para el control de estabilidad y mediante la sintonización de las funciones de membresía por medio de un algoritmo genético se optimiza el desempeño del controlador.

1.5. Objetivos

El objetivo principal de este trabajo de tesis se basa en mantener un péndulo invertido móvil (sobre dos ruedas) en su punto de equilibrio, sin la asistencia de un operador humano, mediante el diseño de un controlador adaptativo implementando algoritmos con base a los conceptos de inteligencia artificial. Para lograr lo anterior se presentan los siguientes objetivos de investigación.

1.5.1. Objetivo general

Resolver el problema de estabilización de un péndulo invertido sobre dos ruedas mediante control adaptativo, desarrollado con técnicas de control inteligente.

1.5.2. Objetivos específicos

- Diseñar y construir un prototipo de robot tipo péndulo invertido móvil.
- Verificar compatibilidad de software con el hardware, para realizar la implementación del algoritmo de control.
- Desarrollar algoritmo de control adaptativo, implementando técnicas de control inteligente.
- Aplicar el algoritmo de control adaptativo: Sistema simulado.
- Aplicar el algoritmo de control adaptativo: Prototipo didáctico de péndulo invertido móvil.
- Evaluar del algoritmo de control adaptativo: Sistema simulado.
- Evaluar del algoritmo de control adaptativo: Prototipo didáctico de péndulo invertido móvil.

1.6. Aportaciones

Durante el desarrollo de este trabajo de tesis se realizó la construcción de prototipo de robot tipo péndulo invertido móvil, en el cual se pueden probar distintas técnicas de control bajo los paradigmas del Cómputo Suave. El prototipo queda a entera disposición del laboratorio de Sistemas Inteligentes como contribución al desarrollo del área. Por otra parte, se estableció una nueva metodología para la generación poblacional en el Sistema Genético Difuso, mediante un Algoritmo de Ajuste Simple. La integración de ambas metodologías permite un mejor rendimiento del GFS.

1.7. Trabajos derivados

Como parte de la divulgación científica dentro del área de Sistemas Inteligentes se asistió y participó en la conferencia de carácter nacional TACS (Taller de Aplicaciones de Cómputo Suave) en el año 2015, con la participación de Carlos Magaña, Oscar Montiel y Roberto Sepúlveda. El artículo fue titulado: “Control de un péndulo invertido sobre dos ruedas mediante síntesis difusa de Lyapunov”.

1.8. Organización del documento

En el Capítulo 1 se presenta la introducción a este trabajo de tesis, en el cual se incluyen los antecedentes relevantes al control inteligente de un TWIP. En el Capítulo 2 se presentan los fundamentos teóricos que se involucran para lograr el control de estabilidad del TWIP, que se divide en seis apartados. El primero de ellos describe el sistema de péndulo invertido móvil que nos permite entender su comportamiento y determinar los parámetros a controlar. El segundo apartado describe los conceptos base de la FL para el diseño del controlador de estabilidad. En el tercero se revisa brevemente la LFS la cual es esencial para establecer el conjunto de reglas difusas. Posteriormente en los apartados cuarto y quinto se describen el funcionamiento de un Algoritmo Genético Binario y Algoritmo Genético Continuo respectivamente, éstos apartados contemplan la metodología para la integración del controlador por lógica difusa y un algoritmo genético, definiéndolo como tal en el sexto apartado, donde se revisa la mecánica de un Sistema Genético Difuso.

El Capítulo 3 presenta la implementación de la propuesta de este trabajo, primero se describe el diagrama a bloques del sistema completo y el diseño del prototipo de TWIP. Enseguida se explica el procedimiento para el diseño del controlador difuso mediante la LFS y por último el funcionamiento del GFS. En el Capítulo 4 los experimentos y resultados son presentados, este capítulo se divide básicamente en dos partes, en la primera se observa el comportamiento del sistema en integración del GFS en simulación y una segunda parte donde se prueba el controlador difuso optimizado en el prototipo de TWIP. Finalmente, el Capítulo 5 muestra las conclusiones de esta investigación y algunas sugerencias como trabajo futuro.

Capítulo 2

Marco Teórico

En este capítulo se exponen las bases teóricas acerca de los tópicos principales de este trabajo de tesis. Comenzando por la descripción de la dinámica del péndulo invertido móvil que nos permite entender el comportamiento del sistema y determinar los parámetros esenciales para el diseño del controlador difuso mediante la síntesis difusa de Lyapunov, por lo que se presentan los fundamentos para entender la integración de ambas metodologías que logran el control de estabilidad angular del sistema.

Posteriormente los algoritmos genéticos son descritos con la finalidad de establecer las bases necesarias para entender el funcionamiento de un sistema genético difuso, el cual se encarga de optimizar las funciones de membresía prediseñadas por la síntesis difusa de Lyapunov y mejora la respuesta del controlador difuso.

2.1. Descripción del sistema: Péndulo invertido móvil.

El péndulo invertido móvil es un sistema del tipo subactuado, es decir, posee más grados de libertad que actuadores. Se compone de un carro ubicado en la parte inferior como base del sistema, un péndulo y un riel como guía colocada en sus ruedas que definen la posición del carro [2]. El sistema completo se muestra en la Figura 2.1. El carro es guiado mediante dos vías colocadas en sus ruedas, esto le permite un desplazamiento de izquierda a derecha. El péndulo se encuentra sujeto justo en el centro de gravedad del carro mediante una bisagra lo que genera un movimiento vertical al riel.

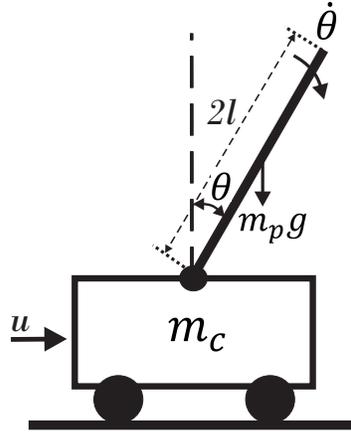


Figura 2.1: Sistema de péndulo invertido móvil. Imagen tomada de [2]

El parámetro m_c y m_p se identifican como la masa de carro y la masa de péndulo respectivamente, la fuerza de gravedad que actúa sobre el péndulo denotada por $g = 9.8m/s$. El parámetro l es la mitad de la longitud del péndulo y la variable u se propone como la fuerza aplicada horizontalmente al carro. Por último tenemos las variables θ y $\dot{\theta}$ que representan el ángulo de inclinación del péndulo y la velocidad angular respectivamente.

De manera similar existe otro modelo que representa el sistema del péndulo invertido móvil como un solo cuerpo como se muestra en [3]. La Figura 2.2 describe una masa m_L ubicada en la base justo donde se encuentran las ruedas, y m_H en el otro extremo del vehículo como un contrapeso. Una variable l que se identifica como la longitud del péndulo, φ el ángulo de inclinación y s la velocidad angular. Este nuevo sistema elimina los rieles guía obteniendo tres grados de libertad, es decir, un desplazamiento en el eje x (*roll*), rotación sobre su propio eje y (*yaw*) y el eje z (*pitch*) que es descrito por el ángulo de inclinación, como se muestra en la Figura 2.2.

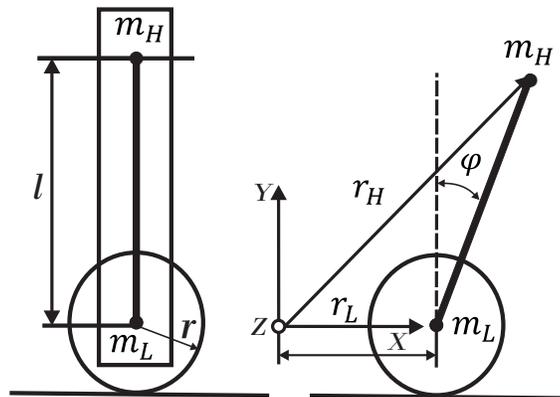


Figura 2.2: Sistema de péndulo invertido sobre dos ruedas. Imagen tomada de [3]

2.1.1. Modelo matemático: Péndulo invertido sobre dos ruedas.

En el péndulo invertido móvil (sobre dos ruedas) el número de variables aumenta, en comparación con el modelo tradicional de péndulo invertido en donde el carro es considerado como una masa M y una fuerza u que actúa sobre él. El carro presentado en la Figura 2.2 es sustituido por dos ruedas, obteniendo un total de tres grados de libertad (3-DOF). Para comenzar el análisis del sistema y obtener un modelo considerable es necesario establecer un sistema de coordenadas. Según [4] se plantea un plano en tres dimensiones del sistema de coordenadas mostrado en la Figura 2.3.

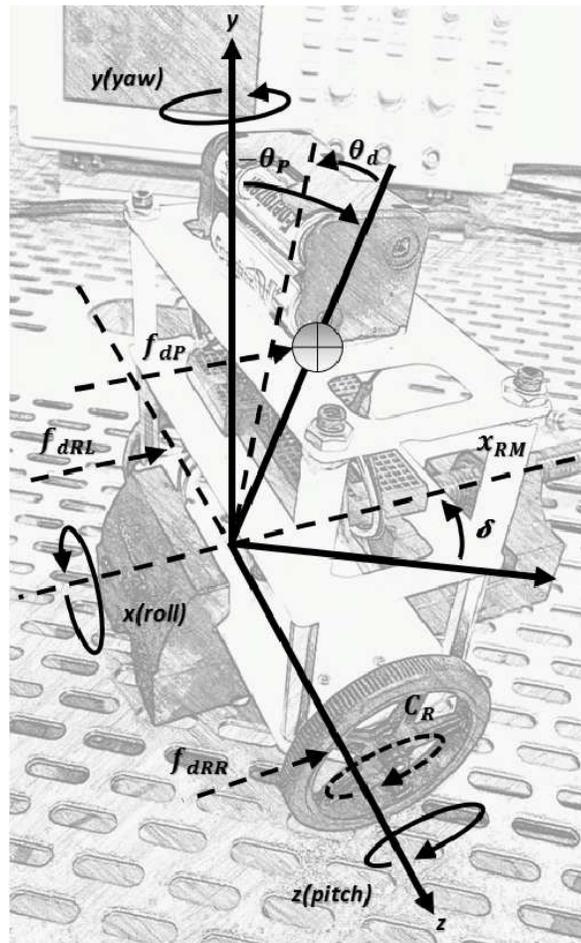


Figura 2.3: Modelo en sistemas de coordenadas del cuerpo del péndulo invertido

El eje vertical que cruza por el centro de las dos llantas se identifica por y (*yaw*) correspondiente al eje de rotación asociado a un ángulo δ y velocidad angular $\dot{\delta}$ denotados en la Figura 2.4, un eje x (*roll*) en el cual se realiza el desplazamiento caracterizado por una posición x_{RM} y la velocidad V_{RM} y un eje z (*pitch*) el que se encuentra justo sobre el eje de las ruedas, describe un ángulo θ_p y su correspondiente velocidad angular ω_p .

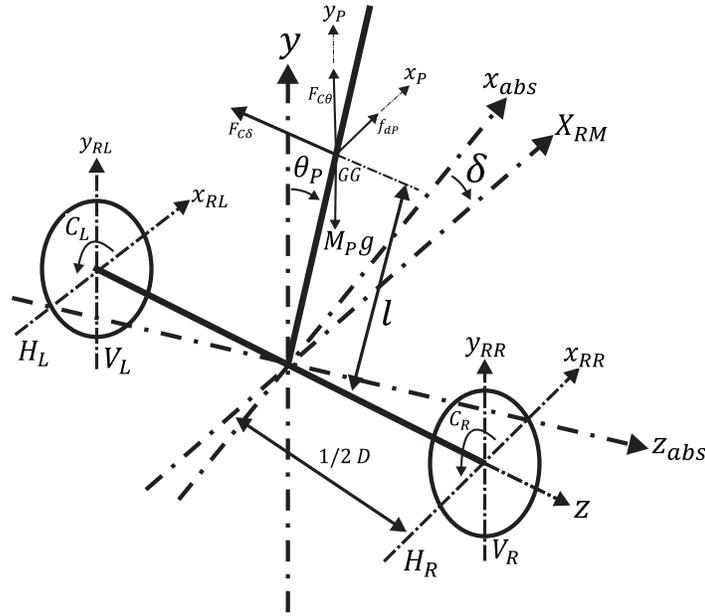


Figura 2.4: Modelo de péndulo invertido. Imagen tomada de [4]

Existen tres tipos de fuerzas que actúan sobre el péndulo, directamente sobre el centro de masa CM actúa una fuerza f_{dP} como lo puede ser una perturbación que desestabilice el vehículo, sobre el centro de la rueda izquierda una fuerza f_{dRL} y una f_{dRR} para la derecha, estas dos fuerzas son el torque aplicado para la corrección del ángulo de inclinación. Una perturbación θ_d describe el cambio de posición del centro de masa, por ejemplo, un empujón al TWIP. El péndulo es capaz de desplazarse aplicando un torque en los motores C_L para el motor izquierdo y uno C_R para el derecho.

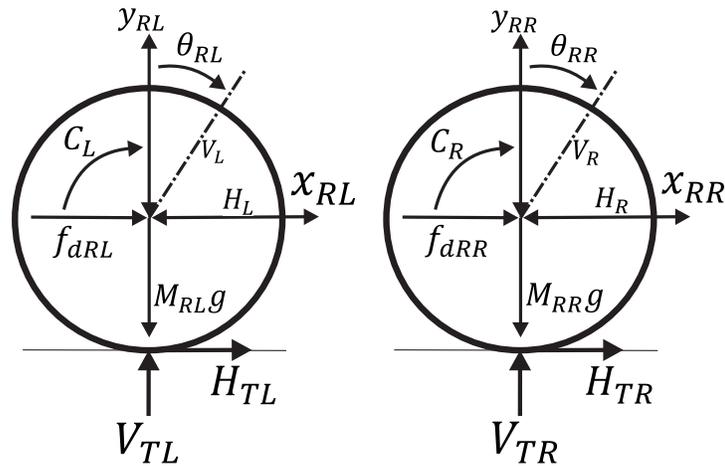


Figura 2.5: Modelo de las ruedas. Imagen tomada de [4]

Las masas que rotan junto con las ruedas se identifican como M_{RL} y M_{RR} . El momento de inercia del chasis con respecto al eje z y eje y se proponen como $J_{P\theta}$ y $J_{P\delta}$ respectivamente. Finalmente tendremos la variable M_P como la masa del chasis, el radio de las ruedas como R , D que representa la distancia entre estas dos ruedas y la distancia entre el eje z y el centro de masa CM con la variable L . De acuerdo a la Figuras 2.4 y 2.5 las ecuaciones de movimiento se definen de la siguiente manera:

$$\begin{aligned}\ddot{x}_{RR}M_{RR} &= f_{dRR} - H_R - H_{TR}, \\ \ddot{y}_{RR}M_{RR} &= V_{TR} - M_{RR}g - V_R, \\ \ddot{\theta}RI_{RR} &= C_R - H_{TR}R.\end{aligned}\tag{2.1}$$

Con efecto de simplificar sólo se presentan las ecuaciones pertenecientes a la rueda derecha, debido a que las ecuaciones de movimiento que describen la parte izquierda son representadas de la misma manera. Por otra parte las ecuaciones que describen el movimiento del chasis son las siguientes:

$$\ddot{x}M_P = f_{dP} + H_R + H_L,\tag{2.2}$$

$$\ddot{y}M_P = V_R + V_L - M_Pg + F_{C\theta},\tag{2.3}$$

$$\ddot{\theta}_PI_{P\theta} = (V_R + V_L)(L \sin \theta_P) - (H_R + H_L)(L \cos \theta_P) - (C_R + C_L),\tag{2.4}$$

$$\ddot{\delta}J_{P\delta} = (H_R - H_L)\frac{D}{2},\tag{2.5}$$

donde H_R , H_L , H_{TR} , H_{TL} , V_R , V_L , V_{TR} y V_{TL} representan el movimiento horizontal y vertical aplicado al sistema.

2.1.2. Representación en espacio de estados

El momento de inercia $J_{P\delta}$ descrito en la ecuación (2.5), depende del ángulo de inclinación θ_P , sabemos que en $\theta_P = 0$ el sistema se encuentra en equilibrio por lo que tendremos $I_{p\delta}|_{\theta_P=0}$. Para este modelo se asume que las ruedas siempre se encontrarán en contacto con la superficie sobre la que se desplace. Las ecuaciones (2.1) a (2.5) son linealizadas alrededor del punto de operación donde $x_{RM} = 0$, $\theta_P = 0$ y $\delta = 0$, de esta manera se puede establecer el sistema en espacio de estados linealizado, sin embargo el sistema puede ser

visto como dos sistemas diferentes. Un sistema describe la rotación con respecto al eje z y el segundo la rotación sobre el eje y , de esta manera se puede diseñar un controlador para cada uno de ellos con la posibilidad de asignar diferentes movimientos.

$$\begin{bmatrix} \dot{x}_{RM} \\ \ddot{x}_{RM} \\ \dot{\theta}_P \\ \ddot{\theta}_P \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & A_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & A_{43} & 0 \end{bmatrix} \begin{bmatrix} x_{RM} \\ \dot{x}_{RM} \\ \theta_P \\ \dot{\theta}_P \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \\ 0 \\ B_4 \end{bmatrix} u, \quad (2.6)$$

$$\begin{bmatrix} \dot{\delta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 \\ B_6 \end{bmatrix} [C_\delta], \quad (2.7)$$

donde A_{23} , A_{43} , B_2 , B_4 y B_6 ,

$$\begin{aligned} A_{23} &= g \left(1 - \frac{4}{3} L \frac{Mp}{E} \right), & A_{43} &= g \frac{Mp}{E}, \\ B_2 &= \frac{4LF}{3E} - \frac{1}{MpL}, & B_4 &= -\frac{E}{F}, & B_6 &= \frac{6}{(9Mr + Mp)RD}, \\ E &= \frac{1}{3} \frac{Mp(Mp + 6Mr)L}{Mp + \frac{3}{2}Mr}, & F &= \frac{Mp}{(Mp + \frac{3}{2}Mr)R} + \frac{1}{L}. \end{aligned}$$

2.2. Lógica difusa

Un controlador por lógica difusa (*FLC*, por sus siglas en inglés) se encuentra basado en la propuesta hecha por Zadeh [18, 30]. Los FLC son aptos para resolver problemas complejos que involucran múltiples variables y que son difíciles de expresar matemáticamente. Su desempeño depende mucho del conocimiento empírico acerca del sistema y el conocimiento de técnicas para interpretar la información en reglas difusas tipo *SI-ENTONCES* así como la creación de las funciones de membresía adecuadas.

En comparación a los métodos de control clásico los cuales requieren un modelo físico de la realidad, un controlador por lógica difusa no requiere un conocimiento previo de modelo matemático del sistema a controlar, lo que permite desarrollar controladores capaces de manipular sistemas de múltiples variables [31]. Durante las últimas décadas los FLC han ampliado sus fronteras en el sector industrial popularizando el método bajo la temática de sistemas inteligentes. Algunas de las problemáticas en control de sistemas donde los controladores difusos pueden ser aplicados son:

- Modelo inexacto
- Ausencia de datos
- Sistemas muy complejos
- Sistemas en los que se conoce por experiencia su comportamiento

2.2.1. Conjuntos difusos

El concepto de conjuntos difusos surgió para dar respuesta a problemas que involucran el factor de incertidumbre y en la ausencia de precisión matemática. Un número difuso es una cantidad cuyo valor es impreciso y que carece de exactitud. En la Figura 2.6 muestra una representación de números difusos, los cuales son representados por las palabras “frío”, “caliente” o “tibio” según al área que pertenecen.

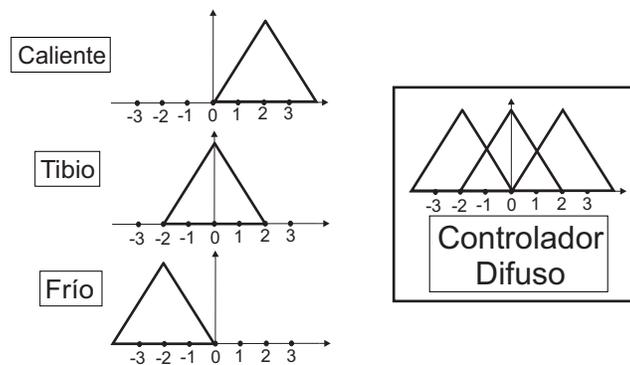


Figura 2.6: Números difusos

El grado de pertenencia x de un conjunto difuso A es asignado a un índice difuso $\mu_A(x)$ en un intervalo de $[0, 1]$, el cual es conocido como grado de membresía de x en A . Con lo descrito previamente se puede definir los siguiente:

- Un conjunto difuso puede ser definido como un conjunto de pares ordenados dado por $A = \{(x, \mu_A(x)) : x \in X\}$, donde X es un conjunto universal y $\mu_A(x)$ es el grado de membresía del objeto x en A . Por lo general $\mu_A(x)$ se encuentra en $[0, 1]$.
- Un subconjunto difuso F de un conjunto A puede ser definido como un conjunto de pares ordenados $F = \{(x, \mu_F(x)) : x \in A\}$, donde el primer elemento es un elemento del conjunto A y el segundo elemento es un valor entre los intervalos $[0, 1]$.

- Una función de membresía $\mu_A(x)$ se caracteriza por $\mu_A : x \rightarrow [0, 1]$, $x \in X$, donde x es un número real que describe un objeto o su atributo, X es el universo de discurso y $A \subset X$.

Retomando el ejemplo de la Figura 2.6, consideremos un universo T el cual denota la temperatura. Las palabras Frío, Tibio y Caliente son subconjuntos del conjunto universal T . Podemos definir los subconjuntos como

$$\begin{aligned} \text{Frío} &= \{\text{temperatura} \in T : 5^\circ\text{C} \leq \text{temperatura} < 15^\circ\text{C}\}, \\ \text{Tibio} &= \{\text{temperatura} \in T : 15^\circ\text{C} \leq \text{temperatura} < 25^\circ\text{C}\}, \\ \text{Caliente} &= \{\text{temperatura} \in T : 25^\circ\text{C} \leq \text{temperatura} < 35^\circ\text{C}\}. \end{aligned}$$

Los límites entre los distintos niveles de temperatura son muy rígidos, por ejemplo una temperatura de 24.9°C sigue siendo Tibia mientras que en 25.1°C ya es Caliente, éstos rangos de temperatura contienen restricciones muy definidas que no permiten un incremento o decremento en la temperatura más suave. Los conjuntos difusos permiten tener rangos más suaves, como se puede observar en la Figura 2.7 la temperatura 25° es miembro de dos conjuntos difusos, Tibio y Caliente con un grado de membresía de $\mu_T(\text{Tibio}) = \mu_T(\text{Caliente}) = 0.5$.

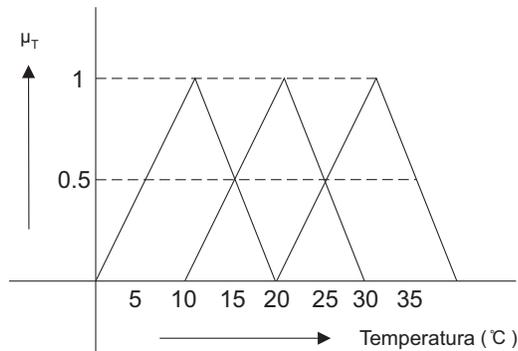


Figura 2.7: Funciones de membresía

2.2.2. Variables lingüísticas

Una variable lingüística toma palabras u oraciones como una variable, con su correspondiente peso o valor. Por ejemplo si tenemos a x como una variable lingüística con nivel “temperatura” por lo que el conjunto difuso denotado por T es descrito por

$$T = \{“muy \text{ frío}”, “frío”, “tibio”, “caliente”, “muy \text{ caliente}”\},$$

donde la temperatura es la variable base identificada como el universo discurso. En otras palabras como seres humanos somos capaces de percibir los cambios de temperatura, sin embargo cuando hacemos la pregunta, ¿cómo está el clima?, podemos responder, tibio, frío, caliente, muy frío, ésto nos permite dar una respuesta aproximada de lo que queremos expresar. Esta analogía es aplicada en la vida diaria para describir distintos universos de discurso como la edad, el apetito, la estatura, etc.

2.2.3. Sistema a base de reglas

Un sistema basado en reglas es una de las maneras más sencillas para representar el conocimiento utilizando lenguaje natural. Una manera genérica de representar lo anterior es como

$$\text{SI } \textit{antecedente}(s) \text{ ENTONCES } \textit{consecuente}(s).$$

Podemos expresar o inferir una acción por experiencia o lógica, sobre este tipo de analogía SI-ENTONCES (IF-THEN). Por ejemplo, distinguir sabores es una habilidad que adquirimos con el paso de los años, ya sea por experiencia propia o ajena mediante una descripción. Ésto permite una mayor flexibilidad en la toma de decisiones y permite aterrizar un conocimiento empírico en un sistema de control basado en lógica difusa.

2.2.4. Sistema de control difuso

Un controlador por lógica difusa puede variar según las características del sistema. El conocimiento de la planta dará origen al conjunto de reglas difusas y la estructura del controlador. En la Figura 2.8 se muestran los bloques que lo componen.

Dos de los controladores por lógica difusa más populares son conocidos como **Mamdani** [31] y **Takagi-Sugeno (T-S)** [32]. El controlador por lógica difusa tipo Mamdani [18] es un FLC adaptativo directo ya que se encuentra basado en sistema de reglas difusas para satisfacer el objetivo de control. Por otra parte el controlador por lógica difusa tipo T-S es un FLC adaptativo indirecto, es decir, el sistema o planta a

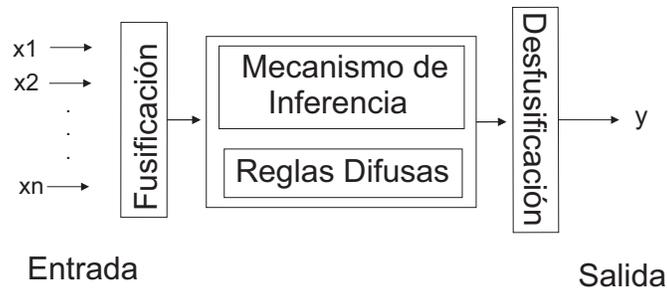


Figura 2.8: Sistema de control difuso

controlar se identifica en términos del modelo difuso descrito por T-S.

2.2.5. Controlador por lógica difusa tipo Mamdani

El FLC se compone de las siguientes etapas:

- **Fusificación:** Se refiere a la interpretación de los valores de entrada, este proceso normalmente conlleva la discretización o normalización del universo de discurso. A través de los conjuntos difusos es posible representar y procesar mediante computadoras las variables lingüísticas. En un FLC típico las variables de entrada son definidas como un error e y un cambio de error Δe , los cuales se traducen en conjuntos difusos utilizando funciones de membresía triangulares, trapezoidales, gaussianas o según convenga en el diseño del controlador.
- **Reglas Difusas:** Es una colección de reglas (SI-ENTONCES) que describen el comportamiento esperado del controlador y se encuentran directamente relacionadas con las variables de entrada y de salida.
- **Máquina de Inferencia Difusa:** Esta etapa establece la conexión lógica entre las entradas y salidas del conjunto difuso.
- **Defusificación:** Se refiere a la conversión de un valor difuso representado mediante un conjunto difuso a una variable de salida cuantificable. En general existen varios métodos para realizar el proceso de defusificación de un conjunto difuso tales como centroide de área, bisector de área, promedio de máximos, mínimo de máximos y máximo de máximos, siendo el primero el más utilizado por ser el que proporciona valores más representativos.

En general podemos observar que la estructura del controlador difuso es dictaminada por las entradas y salidas del sistema, así como el comportamiento de éste.

2.2.6. Controlador por lógica difusa tipo Takagi-Sugeno (T-S)

El controlador por lógica difusa tipo T-S es similar al tipo Mamdani en varios aspectos. Las primeras dos partes del proceso de inferencia difusa como la fusificación de las entradas y el desglose de las reglas difusas son exactamente lo mismo. Sin embargo, la principal diferencia entre ambos tipos, es que el controlador tipo T-S cuenta con funciones de membresía de salida no constantes o no lineales. Supongamos un sistema dinámico no lineal descrito por

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}),$$

donde x es un vector de estado n -dimensional, u es un vector de entrada m -dimensional. El modelo difuso tipo T-S se encuentra compuesto de r reglas, donde la regla j -ésima tiene la siguiente forma

$$\text{Regla}_j : \mathbf{SI} \ x_1 \text{ es } F_1^j \ \mathbf{Y} \ x_2 \text{ es } F_2^j \ \cdots \ \mathbf{Y} \ x_n \text{ es } F_n^j \ \mathbf{ENTONCES} \ \dot{x} = A_j x + B_j u,$$

donde $x = [x_1, x_2, \dots, x_n]^T$, $j = 1, \dots, r$. Cada regla representa una zona difusa en el espacio de estados, para un sistema en espacio de estados n -dimensional el número de dichas zonas corresponde a $r = q^n$ si cada estado es fusificado dentro de q regiones difusas. De esta manera el sistema no lineal es representado por r reglas donde cada una de ellas es asociada a un modelo lineal local. Dado un vector de estados x y un vector de entrada u , el modelo T-S para el sistema dinámico será descrito por:

$$\dot{x} = \frac{1}{\sum_{j=1}^r \mu_j} \sum_{j=1}^r \mu_j (A_j x + B_j u) \quad (2.8)$$

donde

$$\mu_j = \prod_{i=1}^n \mu_j^i(x_i) \quad (2.9)$$

2.2.7. Funciones de membresía

La función que caracteriza un conjunto difuso A en X y asocia cada punto de X con un número real en el intervalo $[0, 1]$, es conocida como función de membresía (*MF*, del inglés *Membership Function*). La elección de las funciones de membresía es por lo general un proceso heurístico y subjetivo. Las funciones de membresía más utilizadas reportadas en la literatura [33] son la triangular, trapezoidal y gaussiana.

2.2.8. Función de membresía triangular

Una función de membresía triangular se especifica por tres parámetros $\{a, b, c\}$ y se define como

$$\mu(x) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right). \quad (2.10)$$

Los parámetros $\{a, b, c\}$ deben de cumplir con $a < b < c$ los cuales determinan las coordenadas de las tres esquinas del triángulo. Las funciones triangulares pueden ser asimétricas dependiendo de la relación entre $a \leq b$ y $b \leq c$.

2.2.9. Función de membresía trapezoidal

Una función de membresía trapezoidal se especifica por cuatro parámetros $\{a, b, c, d\}$ y se define como

$$\mu(x) = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{c-x}{c-b} \right), 0 \right). \quad (2.11)$$

Los parámetros $\{a, b, c\}$ deben de cumplir con $a < b < c < d$ los cuales determinan las coordenadas de las cuatro esquinas del trapecio. Tanto las funciones triangulares y trapezoidales pueden ser asimétricas o simétricas, lo cual representa una ventaja para ciertas aplicaciones.

2.2.10. Función de membresía Gaussiana

Una función de membresía Gaussiana se especifica por dos parámetros $\{m, \sigma\}$ y se define como

$$\mu(x) = \exp \left[-\frac{1}{2} \left(\frac{x-m}{\sigma} \right)^2 \right]. \quad (2.12)$$

Los parámetros m y σ representan el centro y el ancho de la función gaussiana. No hay una regla general para la elección de las MF para resolver un problema o una aplicación en particular. La forma de las MF depende más de la aplicación, que influye en gran medida en el rendimiento del controlador difuso.

2.3. Síntesis difusa de Lyapunov

Básicamente, se sigue el método de síntesis de Lyapunov clásico mediante la construcción de una función candidata de Lyapunov $V(x)$, la cual es una función candidata si $x_0 = 0$ es un punto de equilibrio según la ecuación de análisis, si la función es definida

positiva y además su derivada parcial es de signo negativo. Entre los conceptos básicos de la teoría de Lyapunov destacan los siguientes:

1. $V(0) = 0$,
2. $V(x) > 0$, $x \in \mathfrak{R}^n$,
3. $\dot{V} = \sum_{i=1}^n \frac{\partial V}{\partial x_i} \dot{x}_i < 0$, $x \in \mathfrak{R}^n$.

Este método es empleado al contar con una descripción matemática de la planta. Se considera el siguiente sistema:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (2.13)$$

donde $f(\cdot) = (f_1(\cdot)f_2(\cdot)\dots f_n(\cdot))^T$, $f_i(\cdot)$ son funciones continuas, $x = (x_1x_2\dots x_n)^T$ es el vector de estado del sistema y $u = (u_1u_2\dots u_p)^T$ es la entrada de control. El objetivo de control es estabilizar el sistema alrededor del punto de equilibrio x_0 , en otras palabras diseñar un controlador retroalimentado $u(x)$ de manera que cero sea un punto de equilibrio estable para (2.13). Cuando se conoce el modelo exacto del sistema, una manera de lograr el objetivo de control es elegir una función candidata de Lyapunov $V(x)$, que cumpla con los conceptos mencionados. Sin embargo, cuando el modelo matemático del sistema es desconocido pero se cuenta con un conocimiento parcial, por ejemplo:

SI x_1 *es Positivo* **Y** x_2 *es Negativo* **ENTONCES** u *es Zero*,

en éste caso podemos utilizar una propuesta similar para diseñar un controlador, es decir, elegir una función candidata V y encontrar su derivada. Dado que sólo tenemos conocimiento parcial sobre el sistema, $\dot{V} \leq 0$ se convierte en una ecuación *lingüística*, cuya solución permite generar suficientes condiciones para la señal de control u .

Por lo tanto la síntesis difusa de Lyapunov es una herramienta utilizada para establecer el conjunto de reglas que conforman al controlador difuso, además de garantizar la estabilidad del sistema. Se considera la siguiente función candidata de Lyapunov para un sistema $x(t)$:

$$V(x(t)) = \frac{1}{2}(y_1^2(t) + y_2^2(t)), \quad (2.14)$$

donde $y_i(t)$ son variables del sistema y se caracterizan por dos conjuntos difusos: *chico* y *grande*. Al derivar V obtenemos:

$$\dot{V} = y_1\dot{y}_1 + y_2\dot{y}_2, \quad (2.15)$$

donde se quiere que $\dot{V} < 0$. Ahora bien se considera el primer caso donde y_1 es *chico* y y_2 es *grande*. Substituyendo éstos valores en 2.15 tenemos que:

$$\dot{V} = (\textit{chico})y_1 + (\textit{grande})y_2,$$

y segundo caso en el cual y_1 es *grande* y y_2 es *chico*, tendremos que

$$\dot{V} = (\textit{grande})y_1 + (\textit{chico})y_2.$$

De manera que las diferentes combinaciones para $y_i(t)$ serán:

- $\dot{V} = (\textit{chico})(\textit{positivo}) + (\textit{grande})(\textit{positivo})$
- $\dot{V} = (\textit{chico})(\textit{negativo}) + (\textit{grande})(\textit{negativo})$
- $\dot{V} = (\textit{chico})(\textit{negativo}) + (\textit{grande})(\textit{positivo})$
- $\dot{V} = (\textit{chico})(\textit{positivo}) + (\textit{grande})(\textit{negativo})$
- $\dot{V} = (\textit{grande})(\textit{positivo}) + (\textit{chico})(\textit{positivo})$
- $\dot{V} = (\textit{grande})(\textit{negativo}) + (\textit{chico})(\textit{negativo})$
- $\dot{V} = (\textit{grande})(\textit{negativo}) + (\textit{chico})(\textit{positivo})$
- $\dot{V} = (\textit{grande})(\textit{positivo}) + (\textit{chico})(\textit{negativo})$

Ahora podremos establecer las siguientes reglas difusas que satisfagan $\dot{V} < 0$ en todos sus posibles casos:

- **SI** y_1 es *Positivo* **Y** y_2 es *Positivo* **ENTONCES** u es *NegativoGrande*
- **SI** y_1 es *Negativo* **Y** y_2 es *Negativo* **ENTONCES** u es *PositivoGrande*
- **SI** y_1 es *Positivo* **Y** y_2 es *Negativo* **ENTONCES** u es *Zero*
- **SI** y_1 es *Negativo* **Y** y_2 es *Positivo* **ENTONCES** u es *Zero*

Finalmente tenemos el conjunto de reglas difusas que cumplen con el objetivo de control, el cual es la estabilización del sistema alrededor del punto de equilibrio $x_0 = 0$.

2.4. Algoritmos genéticos

Los algoritmos genéticos son una técnica de búsqueda y optimización basada en los principios de la selección natural y la genética [2, 15, 16]. Los GA permiten a una cierta población compuesta por varios individuos evolucionar bajo específicas reglas de selección para lograr maximizar una función de “aptitud”, la cual evaluará el desempeño del algoritmo. Este método permite moverse de una población de “cromosomas” (p. ej. cadenas de ceros y unos) a una nueva población utilizando un tipo de “selección natural” en conjunto de los operadores inspirados en la genética tales como cruce y mutación. Cada cromosoma se encuentra compuesto por “genes” (p. ej. bits), cada gen siendo una instancia de un particular “alelo o aleloide” (p. ej. 0 ó 1).

El operador de selección elige los cromosomas de la población que podrán reproducirse, donde los cromosomas más aptos se reproducen obteniendo mayor descendencia que los menos aptos. A través del cruce se intercambian subpartes de dos cromosomas, de cierta manera imitando la recombinación biológica; la mutación cambia al azar el valor del alelo de ciertas locaciones en el cromosoma. Este método fue desarrollado por John Holland en la Universidad de Michigan entre los años de 1960 y 1970. El objetivo principal de Holland no era diseñar algoritmos para resolver problemas específicos, sino estudiar el fenómeno de adaptación tal como ocurre en la naturaleza y encontrar el aporte substancial de los mecanismos de adaptación natural en los sistemas computacionales.

Holland en su libro llamado *Adaptation in Natural and Artificial Systems* [34] presenta los algoritmos genéticos como una abstracción de la evolución biológica y un marco teórico sobre la adaptación bajo la cual se rigen los algoritmos genéticos. Posteriormente en el año de 1975 el trabajo de De Jong demostró la utilidad de los algoritmos genéticos en la optimización de funciones y realizó el primer esfuerzo por optimizar los parámetros de GA [2]. Sin embargo, su estudiante Goldberg realizó contribuciones bastante importantes a través de sus exitosas aplicaciones popularizado el método [15]. Algunas de sus aplicaciones como la **optimización** hace uso de los algoritmos genéticos para resolver problemas de optimización combinatorial tales como el ruteado de circuitos y programación de horarios. La **programación automática** ha utilizado los GA para evolucionar los programas computacionales en la realización de tareas específicas y para el diseño de otras estructuras como los autómatas y clasificación de redes.

El **aprendizaje automático** involucra las máquinas con capacidades de aprendizaje en aplicaciones como la clasificación y predicción de tareas, predicción del clima, determinar los pesos adecuados en una red neuronal entre otras aplicaciones. En **economía** se modelan los procesos de innovación, el desarrollo de las estrategias de subasta y la aparición de nuevos mercados. Se pueden agregar otras como el modelado del sistema inmunológico, en ecología para modelar los fenómenos naturales, sistemas sociales como el comportamiento de las colonias de insectos. Se puede observar en los ejemplos anteriores como éstos pretenden resolver problemas de múltiples variables y además difieren entre sí, es decir, los algoritmos genéticos tienen cierta versatilidad que diversifica sus aplicaciones. Por lo que su campo se ha vuelto multidisciplinario y ha crecido rápidamente en los últimos años.

2.4.1. Algoritmo genético binario

En el siguiente ejemplo se muestra la analogía entre la evolución biológica y un algoritmo genético binario. Una manada de lobos se organiza de acuerdo a una estricta jerarquía social, la manada es liderada por un macho alfa y una hembra alfa, esta estrategia permite que la manada maximice sus oportunidades de éxito reproductivo. Supongamos que se quiere criar a un macho alfa de entre cuatro (dos hembras y dos machos) y la manera en la que se determinará el rango será a través de la actitud del lobo y no por su tamaño o fuerza.

Rasgos asociados a la actitud que caracteriza a un macho alfa se encuentran codificados en la secuencia binaria, como se observa en la Figura 2.9. Debe existir alguna manera de determinar los más aptos para obtener el título de alfa, por lo que se enfrentan entre sí. Los lobos con mayor actitud recibirán bajo costo. De esta población de lobos alfa, dos son seleccionados para ser cruzados y crear dos nuevos lobos.

Estos nuevos cachorros tienen una alta probabilidad de ser lobos alfa debido a que sus padres tienen genes que caracteriza a un lobo alfa. La nueva secuencia binaria de los cachorros contiene porciones de la secuencia binaria de ambos padres. Estos dos nuevos cachorros reemplazan a dos lobos descartados que carecen de actitud. Se generarán los cachorros necesarios para restablecer la población a su tamaño original. Realizando varias iteraciones de este proceso, nos guiará a un lobo con la mayor actitud predominante de un macho alfa.

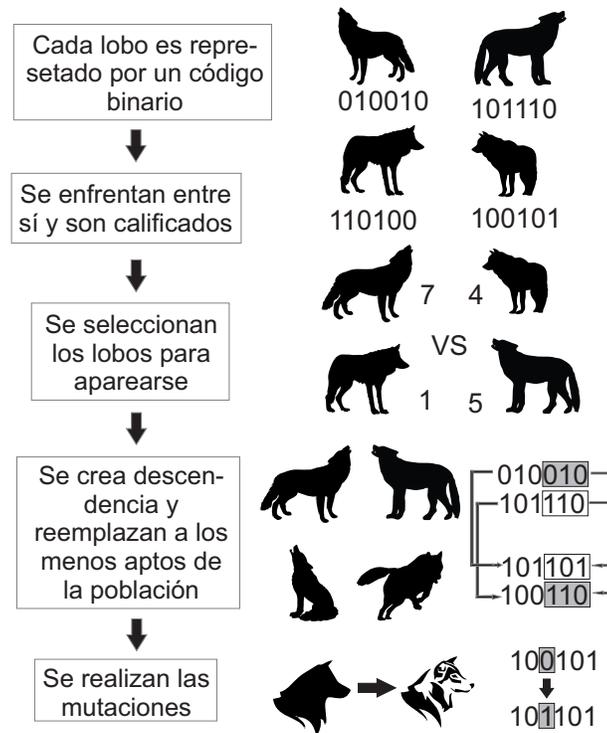


Figura 2.9: Analogía entre biología genética y algoritmo genético binario

2.4.2. Componentes de un algoritmo genético binario

En la Figura 2.10 se describe el proceso que sigue el algoritmo genético binario. Primero se define la función de costo o función aptitud, variables como porcentaje de selección, mutación, número de población y se establecen los parámetros tales como el tipo de cruce. Posteriormente se genera una población inicial de N cromosomas y éstos son decodificados. Los cromosomas son evaluados y un costo es asignado a cada uno de ellos. Según el método de selección se determinarán las parejas y los mejores individuos serán cruzados para obtener una nueva población. De acuerdo al parámetro de mutación seleccionado ciertos cromosomas serán mutados, de esta manera se previene que el algoritmo se estanque en algún punto de búsqueda. De manera iterativa el proceso se llevará a cabo hasta cumplir con los parámetros de convergencia. Finalmente se obtiene el mejor individuo.

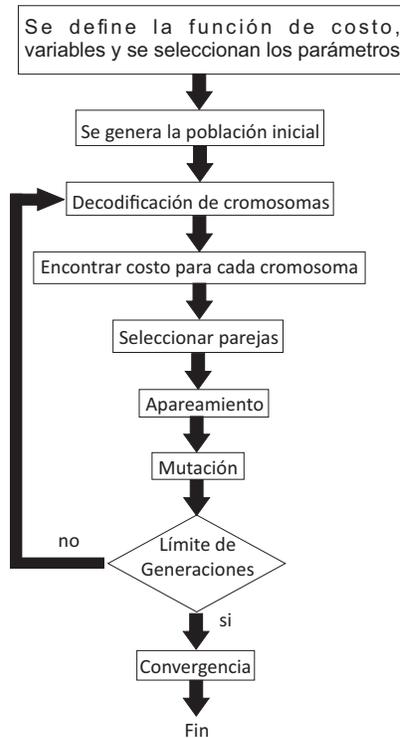


Figura 2.10: Diagrama a bloques de algoritmo genético binario

2.4.3. Selección de variables y función de costo

Una función de costo genera una salida a partir de un conjunto de variables de entrada, su objetivo principal es modificar la salida de una manera deseada encontrando valores apropiados para las variables de entrada. El costo es la diferencia entre el valor deseado y el valor actual, las variables de entrada actualizarán el valor actual de tal manera que el valor deseado sea alcanzado. Inmediatamente se puede observar como las variables de entrada y la función de costo se encuentran estrechamente relacionadas, de tal manera que la elección de los valores apropiados para ambas se reflejará en el desempeño del algoritmo. Para la selección de las variables comenzamos por definir un cromosoma o un arreglo de variables a optimizar. Un cromosoma se encuentra compuesto por N_{var} variables dadas por $p_1, p_2, \dots, p_{N_{var}}$. De esta manera podremos describir un cromosoma como un vector de N_{var} elementos.

$$cromosoma = [p_1, p_2, \dots, p_{N_{var}}]. \quad (2.16)$$

A cada cromosoma se le asignará un costo evaluando la función de costo f en sus elementos, de tal manera que

$$costo = f(cromosoma) = f(p_1, p_2, \dots, p_{N_{var}}). \quad (2.17)$$

La función de costo puede ser negativa o positiva, según lo requiera el algoritmo.

$$f(p_1, p_2, \dots, p_{N_{var}}) = \pm \text{variable} \quad \text{en} \quad (p_1, p_2, \dots, p_{N_{var}}). \quad (2.18)$$

La selección de variables conlleva un proceso analítico, en el que se evalúan cuidadosamente los elementos que componen al problema. Tener más variables no significa un mejor desempeño, sin embargo si se cuenta con las variables necesarias el algoritmo podrá ahorrar tiempo computacional y aumentar su eficiencia. Muchos de los problemas de optimización requieren de restricciones o límites, con la finalidad de acotar las variables, ya que éstas pueden tomar cualquier valor y sería algo ilógico procesar valores fuera de contexto delimitadas por el problema.

2.4.4. Decodificación y codificación de variables

El valor de las variables es representado mediante código binario por lo cual es necesario una manera de convertir un valor continuo a binario y viceversa. De manera que para su codificación tendremos que

$$p_{norm} = \frac{p_n - p_{lo}}{p_{hi} - p_{lo}}, \quad (2.19)$$

$$gen[m] = \text{round}\{p_{norm} - 2^{-m} - \sum_{p=1}^{m-1} gen[p]2^{-p}\}. \quad (2.20)$$

Para decodificar,

$$p_{quant} = \sum_{m=1}^{N_{gen}} gen[m]2^{-m} + 2^{-(M+1)}, \quad (2.21)$$

$$q_n = p_{quant}(p_{hi} - p_{lo}) + p_{lo}, \quad (2.22)$$

donde

p_{norm} = variables normalizadas, $0 \leq p_{norm} \leq 1$

p_{lo} = valor más pequeño de las variables

p_{hi} = valor más grande de las variables

$gen[m]$ = versión binaria de p_n

$\text{round}\{\cdot\}$ = redondeo del entero más cercano

p_{quant} = versión cuantificable de p_{norm}

q_n = versión cuantificable de p_n

Supongamos que la variable x tiene un valor representado por una cadena de bits correspondiente a $N_{gen} = 2$ y x tiene límites definidos por $1 \leq x \leq 4$, por lo que un gen con 2 bits tendrá $2^{N_{gen}} = 4$ posibles valores. Los valores cuantificables de las variables del gen pueden ser encontrados matemáticamente multiplicando el vector de bits por un vector con los niveles cuantificables, es decir

$$q_n = gen \times Q^T, \quad (2.23)$$

donde el $gen = [b_1 \ b_2 \dots b_{N_{gen}}]$ se compone de N_{gen} elementos, b_n es un bit binario que puede tomar valores de 1 o 0, Q es un vector cuantificable $[2^{-1} \ 2^{-2} \dots 2^{-N_{gen}}]$ y un superíndice T que denota traspuesta. En resumen un cromosoma contiene N_{var} genes, lo cuales contienen información de las variables, como en el siguiente ejemplo, cada uno codificado con $N_{var} = 10$ bits:

$$cromosoma = [\underbrace{1111001011}_{gen_1} \ \underbrace{0011011001}_{gen_2} \ \dots \ \underbrace{0110101001}_{gen_{N_{var}}}]$$

Substituyendo cada gen del cromosoma en la ecuación (2.22) produce un arreglo de la versión cuantificable de las variables. Este cromosoma cuenta con $N_{bits} = N_{gen} \times N_{var} = 10 \times N_{var} bits$.

2.4.5. La población

La población se conforma por un grupo de cromosomas. Esta población cuenta con N_{pop} cromosomas y es una matriz de dimensiones $N_{pop} \times N_{bits}$ llenada aleatoriamente con ceros y unos. Cada renglón de esta matriz corresponde a un cromosoma.

2.4.6. Selección natural

Como en la naturaleza los más aptos sobreviven, en este caso las variables con mayor o menor costo serán descartadas (según lo requiera el algoritmo). En primer lugar tendremos una población N_{pop} asociada a un cromosoma en particular, la población es ordenada de menor a mayor costo para poder seleccionar solo a los mejores. El porcentaje de la población que se mantiene está dado por $X_{rate} = 0.5$ correspondiente a un 50% de la población, el cual es un porcentaje aceptable para permitir la evolución adecuadamente.

Podemos calcular el total de cromosomas que permanecerán de toda la población mediante

$$N_{keep} = X_{rate} N_{pop}. \quad (2.24)$$

De la población seleccionada para su reproducción dos cromosomas son seleccionados para su reproducción, éstos procrearán nueva descendencia la cual reemplazará los cromosomas descartados. Existe una variedad de métodos de selección como por ejemplo:

- Tomando los cromosomas sobrevivientes del principio hasta el final, de tal manera que en la matriz los renglones serán enumerados. La madre será identificada por $ma = 1, 3, 5, \dots$ los números impar, mientras que el padre tendrá $pa = 2, 4, 6, \dots$ los números par.
- Apareamiento aleatorio, este método utiliza un generador de números aleatorios para seleccionar los cromosomas de la matriz.
- Apareamiento por pesos. La probabilidad de los cromosomas para aparearse se encuentra inversamente proporcional a su costo. Un cromosoma con el más bajo costo tiene una alta probabilidad de reproducirse, mientras que un cromosoma con un alto costo tiene una baja probabilidad de reproducirse. Existen dos técnicas para llevar esto a cabo:
 - Encuentra la probabilidad del rango n de los cromosomas mediante:

$$p_n = \frac{N_{keep} - n + 1}{\sum_{N_{keep}}^{n=1} n}. \quad (2.25)$$

Posteriormente un número de entre cero y uno es generado aleatoriamente, comenzando por el valor superior de la lista. Por lo tanto el cromosoma con un valor p_n que se acerque al número aleatorio será seleccionado y así sucesivamente hasta formar las parejas para su reproducción.

- Costo Ponderado. Un costo normalizado es calculado para cada cromosoma, substrayendo el costo más bajo de los cromosomas descartados ($C_{N_{keep}+1}$):

$$C_n = C_n - C_{N_{keep}+1}. \quad (2.26)$$

Finalmente p_n es calculado como:

$$p_n = \left| \frac{C_n}{\sum_m^{N_{keep}} C_m} \right|. \quad (2.27)$$

- Selección por torneo. Se eligen aleatoriamente del subconjunto de cromosomas (dos o tres) seleccionados para su reproducción. Los cromosomas con el más bajo costo del subconjunto se convierten en padres. El torneo se repite para cada padre y lo mejor de todo es que este proceso selecciona buenos padres sin la necesidad de ordenar la población.

Cada uno de los métodos de selección resulta en una cruce de padres distintos y por lo tanto tendremos distintos resultados. El método de selección por torneo y el de la ruleta son los más utilizados. Es difícil definir cuál de los métodos de selección es más eficiente debido a la amplia gama de aplicaciones que se pueden presentar, por lo que la experiencia juega un papel muy importante en la selección de las herramientas a utilizar.

2.4.7. Apareamiento

El apareamiento se refiere a la creación de descendencia por los padres elegidos para su reproducción a partir del método de selección. La manera más común de realizar el apareamiento consiste en seleccionar un punto de cruce aleatoriamente entre los primeros y últimos bits correspondientes a los cromosomas de los padres.

El padre heredará su lado izquierdo de su código binario delimitado por el punto de cruce y será colocado en el código binario de su primer descendiente en su lado izquierdo, mientras que su segundo descendiente heredará la parte derecha del código binario del padre y será colocado en su parte derecha. De igual manera ocurrirá con la madre, sin embargo esta intercambiará su parte izquierda con su segundo descendiente y su parte derecha con el primero. Estas dos nuevas descendencias sustituirán a los cromosomas descartados, como se aprecia en la Figura 2.11.

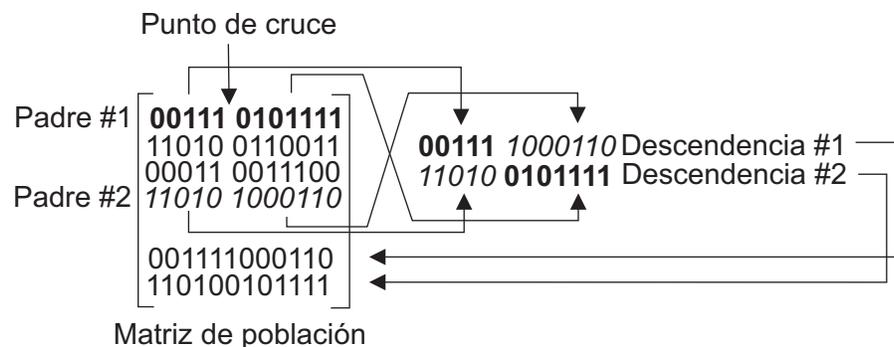


Figura 2.11: Diagrama a bloques del cruce de cromosomas

2.4.8. Mutaciones

Los algoritmos genéticos utilizan la mutación aleatoria para modificar cierto porcentaje de los bits en la matriz de cromosomas, cambiándolos de 1 a 0 y viceversa. El punto de mutación es seleccionado aleatoriamente del número total de bits en la población dado por $N_{pop} \times N_{bits}$. De esta manera el algoritmo no converge rápidamente sobre una región en específico de la superficie de costo y a su vez permite una cierta flexibilidad al algoritmo para encontrar una mejor solución. Normalmente la mutación no ocurre en las mejores soluciones, en este caso los cromosomas con el menor bajo costo, esto se conoce como elitismo. El número de mutaciones está dado por

$$\#mutaciones = \mu \times (N_{pop} - 1) \times N_{bits}. \quad (2.28)$$

La variable μ es el porcentaje de mutación, por ejemplo si se selecciona un porcentaje de mutación equivalente al 10% entonces $\mu = 0.10$. Una vez definido lo anterior se seleccionarán aleatoriamente puntos específicos en la matriz de cromosomas que serán modificados de tal manera que se tendrá algo semejante a lo siguiente

$$\begin{aligned} columns &= [10 \ 5 \ 8 \ 6 \ 8 \ 4 \ 5 \ 8 \ 7 \ 1], \\ renglones &= [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10]. \end{aligned}$$

El primer par indica (10,1) por lo que en el cromosoma número 10 será modificado su primer bit, si éste es 1 se cambiará por un 0 y viceversa.

2.4.9. Convergencia

El número de generaciones se encuentra directamente relacionado a la precisión del resultado. Una práctica muy común es proponer un cierto número de generación y observar el resultado el cual variará su precisión según el problema a resolver claro está. Otra técnica es ajustar el algoritmo para que éste se detenga cuando una solución aceptable sea alcanzada por medio de un error, ésto detendrá el algoritmo sin importar en la generación que se encuentre. Cabe mencionar que después de cierto tiempo los cromosomas y sus costos asociados tenderían a ser todos iguales si no se es implementada la mutación.

2.5. Algoritmo genético continuo

Cuando las variables son cuantificables el algoritmo genético binario es perfecto para realizar la tarea, sin embargo cuando las variables son continuas es lógico pensar representar sus valores con números de punto flotante, además que el método binario cuenta con una precisión limitada, debido a su representación binaria de los valores. Una de las ventajas del algoritmo genético continuo es que requiere menor espacio en memoria y es más rápido que su contraparte binaria, debido a que los cromosomas no tienen que ser decodificados y codificados nuevamente.

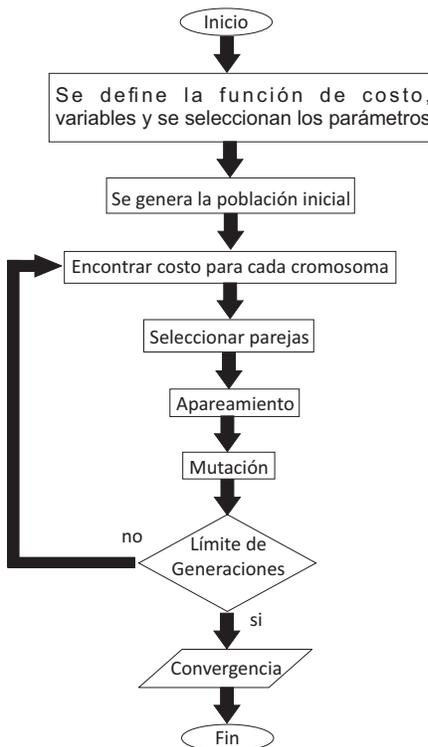


Figura 2.12: Diagrama a bloques de algoritmo genético continuo

En la Figura 2.12 se encuentra el diagrama a bloques del algoritmo genético continuo, muy parecido al binario. Sin embargo éste elimina la parte de decodificación del cromosoma, ahorrando tiempo computacional.

2.5.1. Variables y función de costo

Como ocurre en el caso binario el cromosoma que cuenta con N_{var} variables (problema de optimización N-dimensional) dado por $p_1, p_2, \dots, p_{N_{var}}$, el cromosoma se escribe como

un arreglo de $1 \times N_{var}$ elementos como sigue,

$$cromosoma = [p_1, p_2, \dots, p_{N_{var}}]. \quad (2.29)$$

En este caso las variables ya no son representadas por un número binario sino por un número de punto flotante. A cada cromosoma se le asignará un costo evaluando la función de costo f en sus elementos, de tal manera que:

$$costo = f(cromosoma) = f(p_1, p_2, \dots, p_{N_{var}}). \quad (2.30)$$

2.5.2. Población inicial

Se define una población inicial de N_{pop} cromosomas. Mediante una matriz se representará la población donde cada renglón de la matriz corresponde a un arreglo (cromosoma) de $1 \times N_{var}$ valores continuos. Por lo tanto una población inicial de N_{pop} cromosomas, será representada en una matriz conformada por $N_{pop} \times N_{var}$ valores aleatorios.

2.5.3. Selección natural

Una vez establecida nuestra población inicial, se elegirán los cromosomas más aptos para reproducirse y dar lugar a la nueva descendencia. Tal y como se realizó en la versión binaria, los costos de N_{pop} asociada a los cromosomas son ordenados según el problema de optimización. Este proceso de selección natural debe de ocurrir en cada iteración del algoritmo, permitiendo que la población de cromosomas evolucione en cada generación.

2.5.4. Apareamiento

Como en el algoritmo genético binario, dos padres son seleccionados y la descendencia proviene de ciertas combinaciones entre éstos padres. Imitando un poco las ventajas que ofrece el esquema de cruce binario, se comienza seleccionando al azar una variable en el primer par de padres para ser el punto de cruce

$$\alpha = \text{redondeo}\{\text{aleatorio} * N_{var}\}, \quad (2.31)$$

y además

$$\begin{aligned} padre_1 &= [p_{m1} p_{m2} \dots p_{m\alpha} \dots p_{mN_{var}}], \\ padre_2 &= [p_{d1} p_{d2} \dots p_{d\alpha} \dots p_{dN_{var}}], \end{aligned}$$

donde m y d son los subíndices para identificar al padre (d) y la madre (m). Enseguida se combinan las variables seleccionadas para formar la nueva descendencia de tal manera que

$$\begin{aligned} p_{new1} &= p_{m\alpha} - \beta[p_{m\alpha} - p_{d\alpha}], \\ p_{new2} &= p_{d\alpha} + \beta[p_{m\alpha} - p_{d\alpha}], \end{aligned}$$

donde β es un número aleatorio entre 0 y 1. El paso final consiste en completar el cruce con los cromosomas restantes:

$$\begin{aligned} descendencia_1 &= [p_{m1}p_{m2} \dots p_{new1} \dots p_{dN_{var}}] \\ descendencia_2 &= [p_{d1}p_{d2} \dots p_{new2} \dots p_{mN_{var}}] \end{aligned}$$

A partir de la variable seleccionada p_{new_n} los valores del nuevo descendiente serán intercambiados por los valores de la madre o el padre según corresponda.

2.5.5. Mutaciones

La mutación tiene la finalidad de prevenir al algoritmo de converger sobre un mínimo o máximo local en específico, en caso de que la función evaluada cuente con varios mínimos o máximos locales. Para romper con esta tendencia hay que forzar al algoritmo a explorar otras áreas sobre la superficie de costo introduciendo cambios aleatorios o mutaciones. Principalmente se selecciona el porcentaje de mutación y multiplicando este porcentaje por el número total de variables que pueden ser mutadas dentro de la población tal que $\%mutacion \times \#variables \times N_{pop} = \#mutaciones$, obtendremos el número de mutaciones dentro de la población. Posteriormente números aleatorios son seleccionados para identificar el renglón y la columna de las variables que serán mutadas.

2.5.6. Convergencia

Al igual que en el método binario un número de generaciones es propuesto o a través de una condición de paro. El número de generaciones se encuentra directamente relacionado a la precisión del resultado, por lo tanto por medio de ciertas pruebas podemos elegir los mejores parámetros para distintos tipos de funciones.

2.6. Algoritmo genético difuso

Los sistemas genético difuso o algoritmo genético difuso (*GFS*, por sus siglas en inglés) han sido ampliamente investigados y podemos encontrar varias propuestas en la

literatura [35, 36, 37, 38]. Un GFS consiste en un sistema difuso complementado por un proceso de aprendizaje evolutivo. Uno de los GFS más extensos corresponde al sistema genético difuso a base de reglas (*GFRBS*, por sus siglas en inglés) [2].

Uno de los inconvenientes de los sistemas a base de reglas difusas (*FRBS*, por sus siglas en inglés) es que no son capaces de aprender, pero requieren de una base de conocimiento a base de reglas que se deriva de conocimiento experto. La clave es la implementación de un proceso de aprendizaje evolutivo para automatizar el diseño de un FRBS. Esta definición automatizada de las FRBS puede ser visto como un problema de optimización o de búsqueda; los GA cuentan con la habilidad de encontrar posibles soluciones óptimas en complejos espacios de búsqueda, su estructura general y sus características de rendimiento los hacen candidatos ideales en la incorporación de conocimiento predefinido.

En el caso de los FRBS el conocimiento predefinido se encuentra en forma de variables lingüísticas, parámetros de las funciones de membresía, reglas difusas, número de reglas, etc. En [39] el autor clasifica a los GFS dentro de dos categorías de acuerdo a el uso del algoritmo, éstos son:

- Aprendizaje genético de los componentes
- Sintonización genético de los componentes

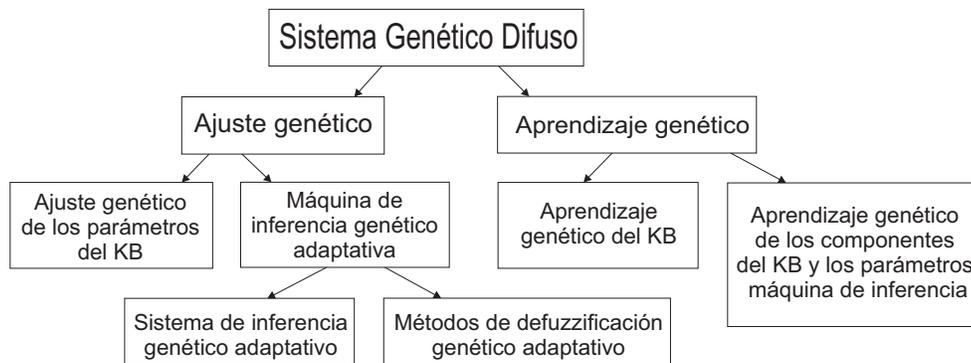


Figura 2.13: Taxonomía de los GFS

La Figura 2.13 describe una primera vertiente donde se realiza una sintonización de los parámetros predefinidos, los cuales serán ajustados para encontrar el conjunto óptimo de parámetros de las variables involucradas en la discusión, como lo son las funciones de membresía, reglas difusas, número de reglas, etc. La segunda vertiente es un poco

más compleja, ésta involucra el proceso de aprendizaje por lo que realizará una búsqueda más elaborada en el espacio de posibles base de RB o KB sin depender de una base de parámetros predefinidos, por lo que tendrá que comenzar desde cero.

2.6.1. Proceso de sintonización genético

El objetivo del proceso de sintonización genético es el adaptar los parámetros de un sistema difuso dado y obtener como resultado un FRBS que demuestre un mejor rendimiento. Los siguientes componentes del KB son candidatos potenciales para ser optimizados:

- Los parámetros de las funciones de membresía, y
- Las reglas difusas (RB)

2.6.2. Sintonización de las funciones de membresía

Se asume un RB predefinido el cual representa el conocimiento estructurado del sistema expresado de una manera lingüística. El desempeño del FRBS puede mejorar solamente sintonizando los conjuntos difusos sin necesidad de modificar el RB. A continuación se muestra una analogía de la distribución de las funciones de membresía antes y después de la sintonización de sus parámetros.

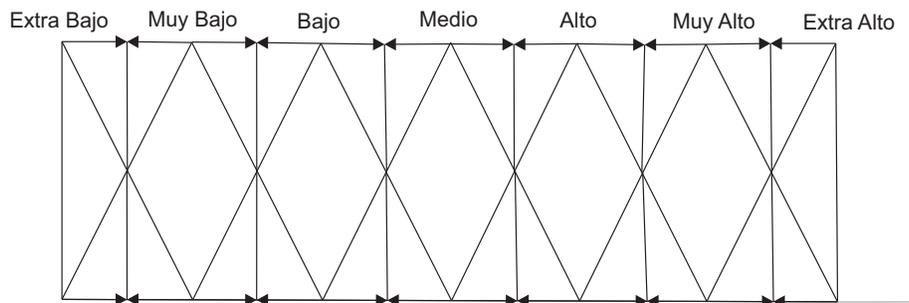


Figura 2.14: Funciones de membresía predefinidas

En la Figura 2.14 podemos observar un conjunto de funciones de membresía predefinidas, las cuales serán sintonizadas mediante un GA. Por otra parte en la Figura 2.15 tenemos tres funciones de membresía predefinidas representadas por la línea continua y su posible sintonización identificadas por la línea punteada.

2.6.3. Sintonización de las reglas difusas

La idea principal en este tipo de sintonización es identificar en la RB las reglas que causan un bajo rendimiento en el controlador y reemplazarlas por mejores reglas. Al re-

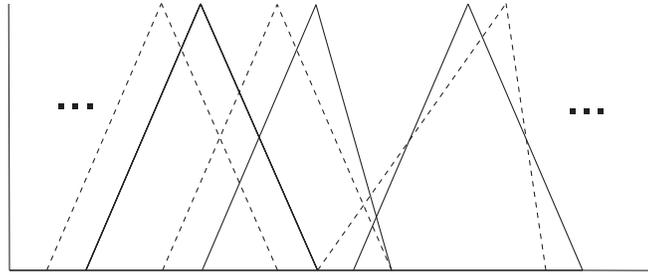


Figura 2.15: Sintonización de las funciones de membresía

emplazar solo una regla consecuente su efecto será del tipo local, sin embargo al modificar tanto el consecuente y antecedente tendrá un mayor impacto sobre el comportamiento del controlador difuso. El algoritmo tiene la capacidad de eliminar o modificar las reglas difusas con la finalidad de incrementar el desempeño del controlador.

2.6.4. Extracción de datos

Retomando la metodología implementada en los algoritmos genéticos, un cromosoma se encuentra compuesto por N_{var} variables dadas por $p_1, p_2, \dots, p_{N_{var}}$, descrita en la ecuación (2.29). En este caso el cromosoma será llenado con los valores correspondientes a los conjuntos difusos y reglas difusas. La Figura 2.16 describe el procedimiento para la extracción de los datos, el vaciado de los valores se realiza de izquierda a derecha colocando en orden todos los puntos correspondientes a las funciones de membresía. En la parte derecha del cromosoma podemos encontrar los valores de las reglas difusas, con efecto de simplificar la imagen solamente fue agregado el número de regla, sin embargo podemos agregar toda la información del controlador, claro solo si es necesario de otra forma es recomendable solo vaciar los datos que serán utilizados.

2.6.5. Ventajas del GFS

Como ya sabemos, los algoritmos genéticos son capaces de almacenar en un cromosoma una diversidad de información que posteriormente puede ser utilizada. Para el GFS se codifican diferentes componentes del sistema difuso como lo son:

- Funciones de membresía
- Formas de las funciones (triangulares, trapezoidales, gaussianas, etc)
- Formas óptimas de las funciones de membresía
- Número de niveles por variable

- Reglas difusas

De esta manera se pueden definir múltiples mecanismos con diferentes características y diversificar el método.

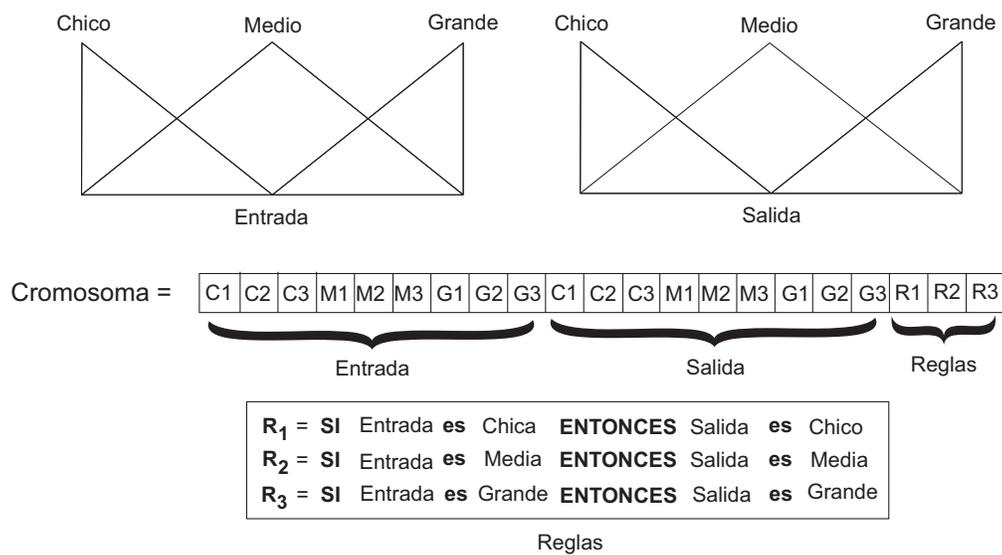


Figura 2.16: Extracción de datos: Sistema Difuso a un cromosoma

Capítulo 3

Metodología

En este capítulo se presenta la metodología empleada para resolver el problema de estabilización de un péndulo invertido móvil, mediante la propuesta de solución derivada de los tópicos ya analizados en el capítulo anterior. Se describe el diagrama a bloques del sistema y el prototipo didáctico de péndulo invertido móvil construido, en el cual se integra el controlador difuso optimizado con el objetivo de probar su rendimiento.

Se detalla el procedimiento para el diseño del controlador difuso mediante la Síntesis Difusa de Lyapunov [21], en la cual se analiza el modelo dinámico del péndulo invertido móvil con la finalidad de extraer las variables a controlar, obteniendo de esta manera el conjunto de reglas difusas que conforman parte del controlador difuso y la distribución de las funciones de membresía sintonizadas de manera manual. Finalmente se proporciona la metodología seguida para el funcionamiento del algoritmo genético difuso que optimiza las funciones de membresía prediseñadas, con el objetivo principal de mejorar el rendimiento del controlador difuso en la corrección del error angular.

3.1. Propuesta de solución

La propuesta que se hace para resolver el problema de estabilización del sistema consta de dos etapas. En la primera se propone el diseño del controlador difuso mediante la síntesis difusa de Lyapunov la cual es una herramienta utilizada para establecer el conjunto de reglas que conforman al controlador difuso, además de garantizar la estabilidad del sistema. Sin embargo esta metodología no proporciona la distribución de las funciones de membresía, es decir, es necesario realizar una sintonización manual hasta obtener una respuesta aceptable del sistema. Es por eso que se hace la integración de los GA para garantizar una distribución óptima de las funciones de membresía. En la Figura 3.1 se muestra el diagrama a bloques de la propuesta de solución.

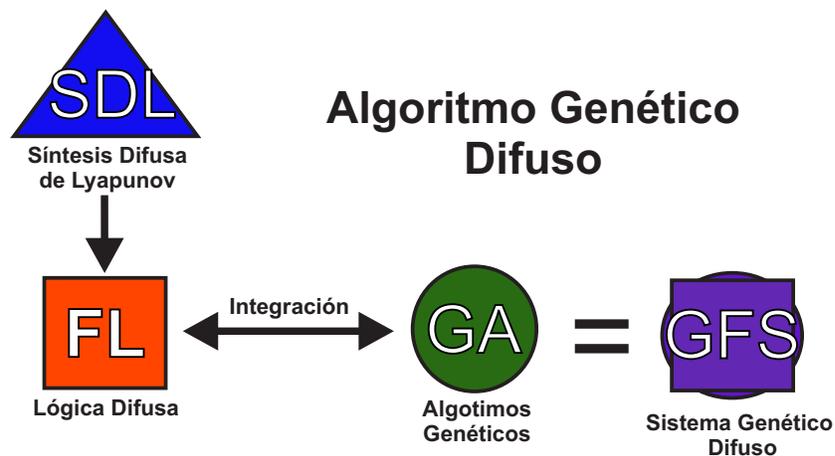


Figura 3.1: Diagrama a bloques de la propuesta de solución

La integración de la lógica difusa y los algoritmos genéticos da origen a lo que se conoce como un GFS, el cual posee capacidades de aprendizaje y adaptación. En específico para este trabajo de tesis se utilizará únicamente el parámetro de sintonización de las funciones de membresía. Este proceso requiere de un controlador difuso predefinido, previamente diseñado en base a la síntesis difusa de Lyapunov como lo muestra Margalioy y Gideon en [21]. Finalmente obtendremos un controlador difuso que consiga el objetivo de control, en este caso permitirle al vehículo permanecer en el punto de equilibrio.

3.2. Descripción del sistema: Diagrama a bloques.

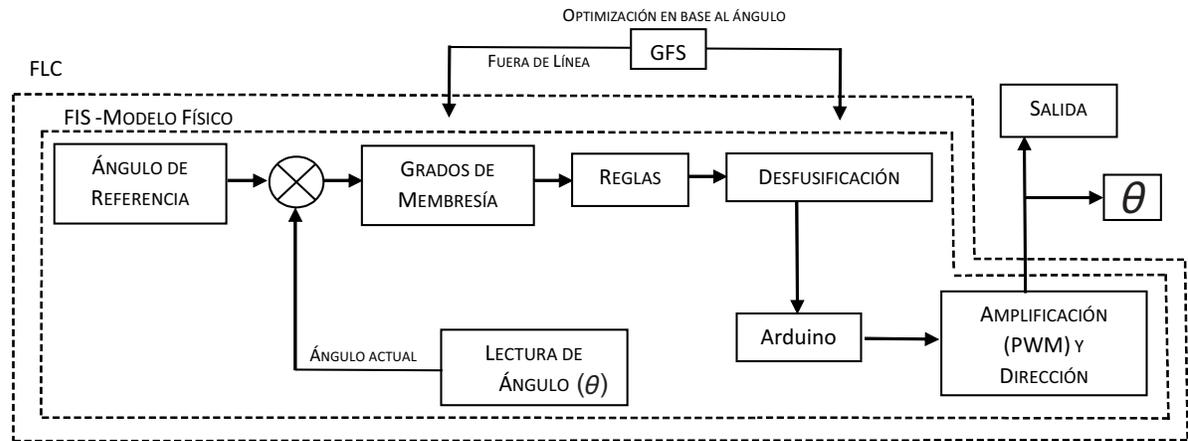


Figura 3.2: Diagrama a bloques del sistema

La arquitectura general de FLC se presenta en la Figura 3.2, en ella se observa el Sistema de Inferencia Difusa (*FIS*, por sus siglas en inglés) como unidad central y bloques complementarios. El FIS se encuentra en el dispositivo Arduino, en el cual se realizarán las operaciones matemáticas correspondientes al FIS donde se incluirán los grados de membresía de entrada (ángulo y velocidad angular), las reglas utilizadas y la defusificación.

El GFS será implementado fuera de línea, es decir, el algoritmo GFS no es programado en el microcontrolador debido a las capacidades limitadas en procesamiento de este. En esta etapa la dinámica del sistema analizada en el capítulo 2 es simulada, agregando el controlador en conjunto con el GFS. El controlador optimizado es programado en el microcontrolador y posteriormente se realizan las pruebas requeridas para determinar la eficiencia de este.

Básicamente se genera un FLC mediante la síntesis difusa de Lyapunov y el sistema completo es simulado para optimizar el controlador diseñado. El controlador recibirá lecturas del ángulo actual, esta lectura generará un error del cual podemos obtener el error y cambio de error (ángulo y velocidad angular) correspondientes a las entradas al controlador difuso. La salida de controlador difuso será interpretada como una señal PWM y según la orientación del giróscopo se determinará la dirección de giro de los motores. De esta manera tendremos una retroalimentación del ángulo, permitiendo al sistema corregir el error angular y conseguir la estabilidad del vehículo.

3.3. Descripción del sistema: Prototipo de péndulo invertido móvil

El robot consta de un chasis de plástico equipado con dos motores de DC ubicados en la parte inferior, cada uno con ruedas de 6.0 *cm* de diámetro. Para realizar el control de dirección y velocidad se utilizó un arreglo de puente H (DRV8833) ubicado en el centro del robot. El ángulo de inclinación del robot es medido mediante un giróscopo (sensor MPU-6050) colocado en el centro de masa del nivel inferior y los datos provenientes de este son procesados por un microcontrolador Arduino Nano en el cual es programado el controlador difuso optimizado. Tanto los motores, el micro controlador, el sensor y el puente H son alimentados mediante cuatro baterías de 1.2 volts, éstas fueron colocadas en el tercer nivel del vehículo con el fin de aumentar el momento de inercia del sistema tal como se muestra en la Figura 3.3.

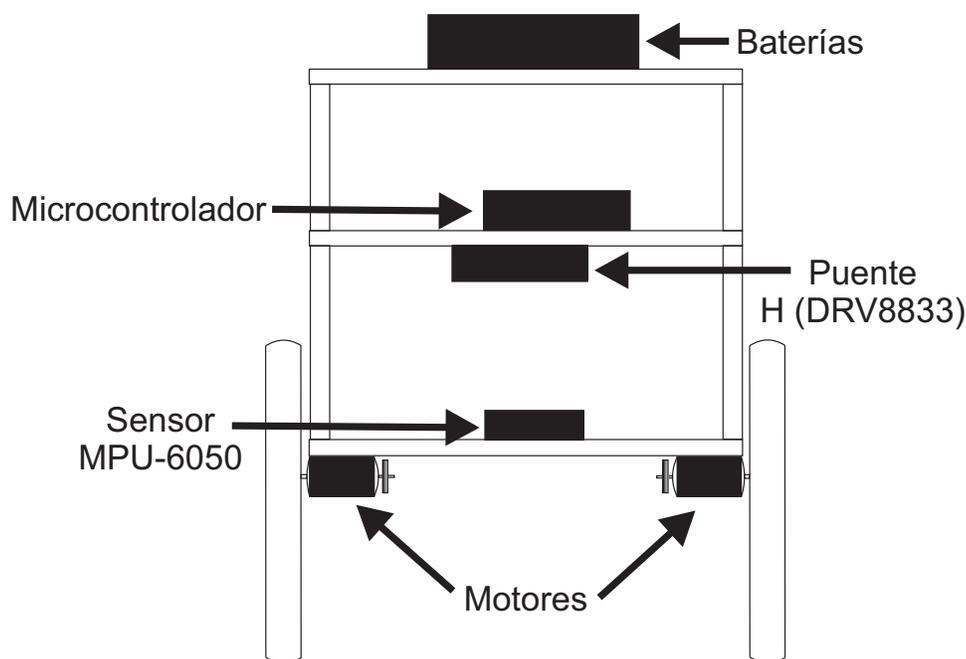


Figura 3.3: Distribución de componentes

Es recomendable colocar el giróscopo en el centro de masa del vehículo, esto permite que las aceleraciones angulares de los tres ejes se referencien en el centro y la simetría del sistema se conserve. En este caso se colocó en el primer nivel de la estructura, en cruce por el eje de los actuadores como se muestra a en la Figura 3.3. El material utilizado es un derivado del plástico pero mucho mas resistente y ligero, se le conoce como Delry y es fácilmente maquinable en cualquier CNC. Las medidas de las tres capas del chasis son

130x60mm, Figura 3.4.

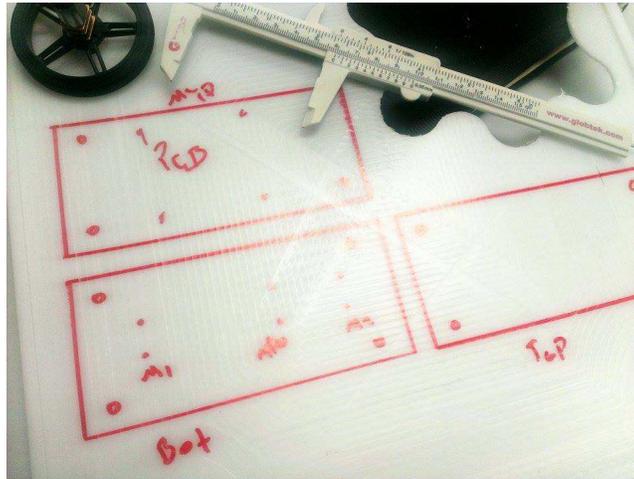


Figura 3.4: Plástico utilizado

La parte de potencia es separada del microcontrolador para evitar inducción electromagnética que se pueda presentar por el ruido eléctrico de los motores al cambiar rápidamente de dirección y velocidad. Por lo tanto el driver de los motores fue aislado en una pastilla independiente. Por otra parte el peso de las cuatro baterías de 1.2 volts juntas supera cualquier otra pieza del sistema en su peso y por esta razón se coloca en la parte superior del chasis para crear un momento de inercia mayor que disminuye la inestabilidad del equilibrio del sistema, similar al péndulo invertido con una masa mayor al final del péndulo.

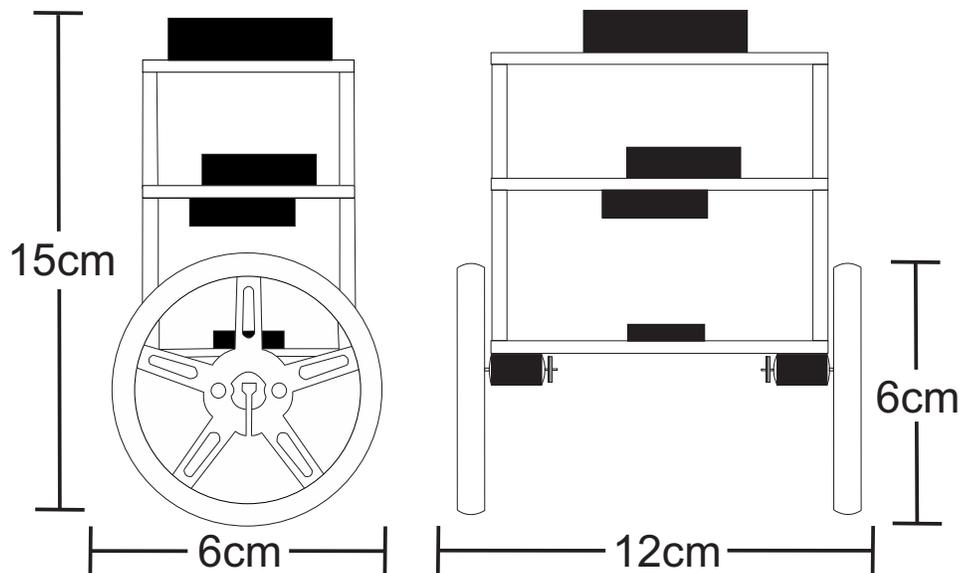


Figura 3.5: Mediciones del modelo

Las dimensiones del modelo se muestran en la Figura 3.5. Con una altura de 15 *cm*, 8 *cm* de ancho y 12 *cm* de largo permite a las ruedas cargar toda la estructura y realizar cambios de dirección y velocidad rápidos cuando el controlador así lo requiere. El modelo real del prototipo de péndulo invertido móvil se muestra en las Figuras 3.6 y 3.7.

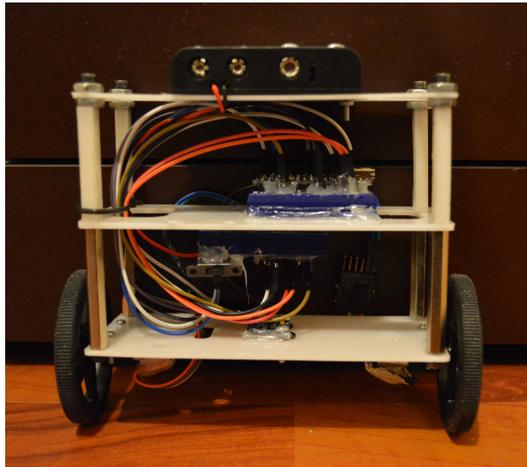


Figura 3.6: Modelo real del prototipo de péndulo invertido móvil: Vista Frontal

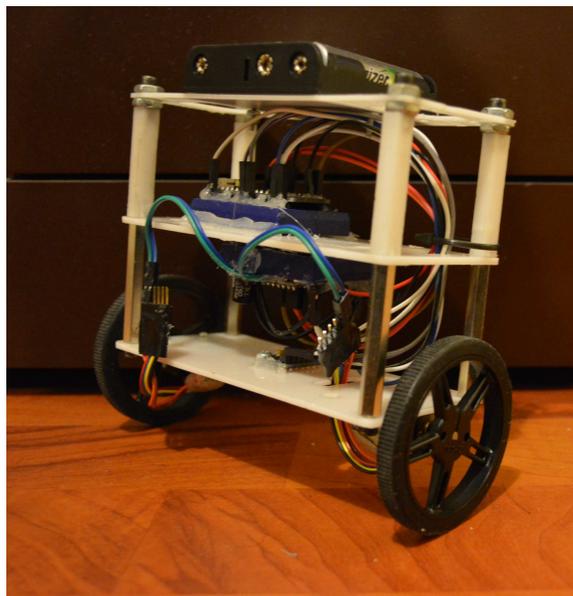


Figura 3.7: Modelo real del prototipo de péndulo invertido móvil: Vista Lateral

3.4. Descripción de los componentes

Uno de los objetivos de este trabajo de tesis consiste en la implementación de hardware adecuado con el cual sea posible obtener un buen desempeño del controlador. Los componentes utilizados en el robot se describen a continuación.

3.4.1. Arduino Nano

Arduino Nano es un microcontrolador ATmega328. Cuenta con 14 terminales de salida/entrada digitales de las cuales 6 pueden ser utilizadas como salidas PWM (Modulación por Ancho de Pulsos, *del ingles Pulse With Modulation*), ocho entradas analógicas, un botón de reset. Algunas de sus características son las siguientes:

Tabla 3.1: Características básicas: Arduino Nano

Nombre	Atributo
Microcontrolador	ATmega328
Voltaje de operación	de 3.3 V a 5 V
Voltaje de entrada	5 V a 12 V
Pines I/O Digitales	14 (6 PWM)
Pines Analógicos	8
Corriente de CD I/O	40 mA
Memoria Flash	16 KB
SRAM	1 KB
EEPROM	512 bytes
Velocidad de Reloj	16 MHz

Alimentación

La Arduino Nano puede ser alimentada por una fuente externa de 3.3 V o 5 V en el pin de VCC. Cuenta con un regulador empotrado por lo que puede aceptar voltajes de hasta 12 V de CD. Si el dispositivo es alimentado mediante una fuente no regulada es conveniente conectarlo al pin RAW y no en VCC.

Salidas y Entradas

Cada una de las 14 terminales digitales en la Arduino Nano pueden ser utilizados como entrada y salida, algunas de las funciones como “*pinMode()*”, “*digitalWrite()*” y “*digitalRead()*”. Cada pin es capaz de proveer o recibir una corriente máxima de 40mA,

cuenta con una resistencia de “pull-up” interna la cual puede activarse o desactivarse mediante programación. Algunos de los pines cuentan con ciertas características especiales:

- **Puerto Serial 0(RX) y 1(TX):** Recibir datos mediante la terminal RX y transmitir con TX.
- **Interrupciones Externas:** Estas terminales pueden ser configurados para ser disparados ante valores bajos/altos, o mediante un cambio de valor.
- **PWM:** Mediante el comando “*analogWrite()*” es posible obtener una señal PWM de 8-bit por en las terminales 3, 5, 6, 9, 10 y 11.
- **LED:** Cuenta con un LED empotrado conectado directamente en la terminal digital 13. Cuando la terminal se encuentra en estado alto, el LED se encuentra encendido, cuando se encuentra en estado bajo el LED estará apagado.
- **I²C :** Los pines A4(*SDA*) y A5(*SCL*) soportan comunicación I²C.

Características Físicas

Unas de las principales cualidades por las que se eligió este dispositivo, fue principalmente a sus dimensiones, ver Figura 3.8 y capacidad de procesamiento, ya que en la actualidad esta herramienta cuenta con distintas librerías las cuales permiten al dispositivo interactuar sencillamente con una gran variedad de sensores.

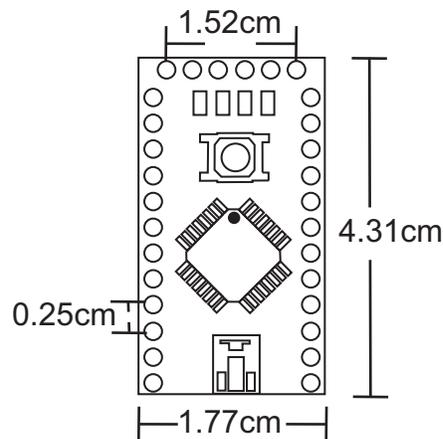


Figura 3.8: Esquemático: Arduino Nano.

3.4.2. MPU-6050

Las Unidades de Medida Inercial están compuestas por un conjunto de sensores que miden aceleración, giro, y campo magnético. El objetivo de éstos sensores en general es medir el movimiento en tres ejes, para lograr la estabilización de los vehículos autónomos. Estas unidades tienen múltiples aplicaciones en robótica, aeronaves, vehículos aéreos no tripulados, satélites, entre muchos otros.

El sensor MPU-6050 contiene un giroscopio de 3-ejes, un acelerómetro de 3-ejes y un procesador digital de movimiento (DMP-Digital Motion Processor). El sensor cuenta con un convertidor analógico a digital de 16-bit para la conversión de información proveniente del giroscopio y acelerómetro. El integrado se comunica a través de la interfaz I^2C a 400kHz, además cuenta con un sensor de temperatura integrado con un error de $\pm 1\%$. El dispositivo opera entre $2.375 V - 3.46 V$, proporciona un pin de referencia VLOGIC el cual establece el nivel lógico para la interfaz I^2C .

Características: Giroscopio

- Salida-Digital ejes X, Y y Z sensores de velocidad angular, con un rango programable de $\pm 250, \pm 500, \pm 1000, \pm 2000^\circ/sec$.
- Sincronización externa conectado al pin FSYNC el cual soporta imágenes, video y sincronización GPS.
- Contiene un ADC de 16-bit activo simultáneamente con el giroscopio.
- Corriente de operación: $3.6mA$.
- Corriente en modo espera: $5\mu A$.

Características: Acelerómetro

- Salidas digitales de tres ejes con un rango programable de $\pm 2g, \pm 4g, \pm 8g$ y $\pm 16g$.
- ADC itegrado de 16-bit en funcion simultanea con el acelerómetro sin necesidad de un multiplexor externo.
- Operación en modo bajo consumo: $10\mu A$ a $1.25Hz$, $20\mu A$ a $5Hz$, $60\mu A$ a $20Hz$, $110\mu A$ a $40Hz$.

Características Físicas

El diagrama presentado en la Figura 3.9 muestra la orientación de los ejes de sensibilidad y polaridad de rotación. Para obtener las lecturas correctas es indispensable identificar la posición en la que se encuentra el dispositivo, es decir, sobre que eje se encuentra el movimiento del sensor.

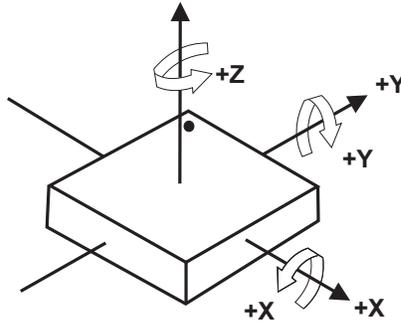


Figura 3.9: Orientación de los ejes de sensibilidad y polaridad de rotación.

Las dimensiones del giróscopio se muestran en la Figura 3.10. Las terminales utilizadas del dispositivo corresponden únicamente a la alimentación (VCC) del integrado, tierra (GND), la señal de reloj (SCL) para iniciar la comunicación I^2C y SDA la cual corresponde a la información serial de salida, es decir las lecturas del sensor.

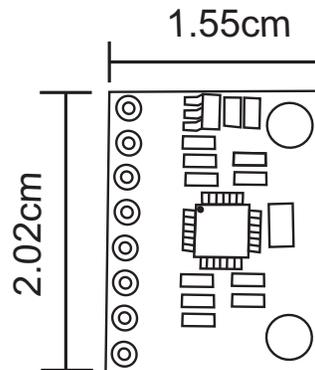


Figura 3.10: Dimensiones del chip

3.4.3. Puente H (DRV8833)

Este pequeño dispositivo (DRV8833) es capaz de controlar simultáneamente de forma independiente dos motores de CD en ambos sentidos (adelante/atrás). El voltaje de operación oscila entre 2.7 V y 10.8 V.

Tabla 3.2: Especificaciones generales

Tamaño	0.19 cm x 0.31 cm
Driver del motor	DRV8833
Peso	1.0 g
Canales	2
Voltaje mínimo	2.7 V
Voltaje máximo	10.8 V
Corriente continua por canal	1.2 A
Corriente máxima	2 A
Corriente continua en paralelo	2.4 A
Max. frecuencia PWM	100 kHz

Este dispositivo es un puente H dual, el cual puede ser utilizado para el control bidireccional de dos motores de corriente directa con un rango operativo de 2.7 V hasta 10.8 V. Es capaz de tolerar una corriente de 1.2 A por canal de manera continua con picos máximos de corriente de hasta 2 A. Incluye protección contra voltajes en reversa, sobre corriente y temperatura.

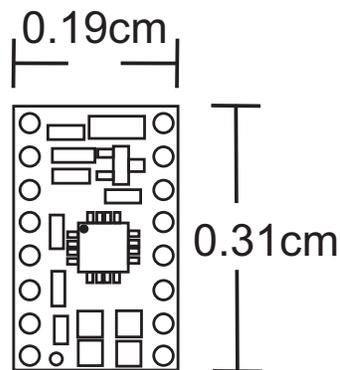


Figura 3.11: Dimensiones del puente H.

3.4.4. Motores

Los motores utilizados trabajan en un rango de $3 V$ a $9 V$, cuentan con dimensiones de $(2.4 \times 1.0 \times 1.2 \text{ cm})$, un peso aproximado de $10 g$ y con un diámetro del eje del motor de 3 cm . Gracias a sus características físicas, éstos motores permiten al controlador difuso realizar los cambios de dirección y velocidad necesarios para el control de estabilidad del péndulo invertido móvil, ya que su tiempo de respuesta en conjunto con el controlador permite al robot permanecer de pie. Algunas de las especificaciones generales del motor son las siguientes:

Tabla 3.3: Características básicas: Motores

Relación de engrane	100.37:1
Velocidad @ $6 V$	320 rpm
Corriente @ $6 V$	80 mA
Corriente Max.	1600 mA
Torque	30 oz * in

3.5. Sistema de control y criterio de evaluación del controlador difuso

Un sistema de control es un conjunto de dispositivos encargados de regular el comportamiento de otro sistema con el fin de obtener los resultados deseados y reducir las probabilidades de error. El objetivo principal es proponer un controlador que estabilice asintóticamente el sistema alrededor del punto de equilibrio, suponiendo que el péndulo se encuentra inicialmente en su posición inicial $\theta = 0$.

Retomando la Figura 2.3 mostrada en el capítulo dos, una fuerza f_{dp} (considerada una perturbación en el sistema) será aplicada en el robot, por lo tanto tendremos un $-\theta_p$ o θ_p según la dirección en la que sea aplicada. El controlador difuso realizará la acción de control θ_d para la corrección del ángulo del péndulo invertido móvil (TWIP, por sus siglas en inglés), siendo este el principal objetivo de control, tal como se muestra en la ecuación 3.1.

$$\theta_d = \theta_d - \theta_p. \quad (3.1)$$

Se tiene un sistema de control de lazo cerrado, en el que la acción de control esta en función de la señal de salida como se muestra en la Figura 3.12. Tendremos dos señales,

un ángulo de inclinación θ , una velocidad angular $\dot{\theta}$, para lograr un cambio en la dirección y velocidad del robot es necesario aplicar un torque en C_L y C_R que corresponden a la rueda izquierda y derecha del robot respectivamente. El sensor es retroalimentado hasta finalmente conseguir estabilizar el robot.

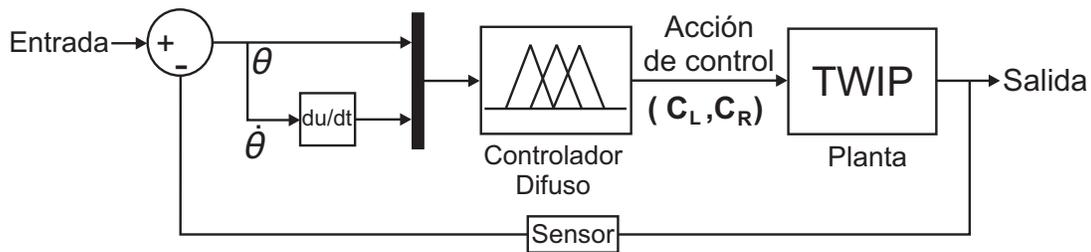


Figura 3.12: Sistema de Control

Ahora bien, hay ciertos puntos que es necesario considerar, la posición inicial del robot será $\theta = 0$ y según la dirección en la que sea aplicada una perturbación, el ángulo será interpretado como negativo o positivo como se muestra en la Figura 3.13. Por lo tanto si aplicamos una fuerza del lado derecho estando el robot en su posición inicial, el ángulo será negativo y se realizará la acción de control contrarrestando con un ángulo más positivo en la dirección contraria, hasta estabilizar el robot y aplicando una fuerza del lado izquierdo se seguirá el mismo procedimiento.

Para poder analizar y decir que nuestro controlador difuso es mejor con respecto a otro se establecieron los siguientes criterios:

- El tiempo de asentamiento del ángulo de inclinación θ , los picos máximos y amplitud de las oscilaciones en presencia del impulso o alguna perturbación implementando el controlador difuso sintonizado por el GFS debe de ser menor que el sintonizado manualmente.
- Los periodos de recuperación del ángulo de inclinación θ a su estado estable provocados por una perturbación implementando el controlador difuso sintonizado por el GFS debe de ser menor que el sintonizado manualmente.
- Todos los experimentos en simulación se realizaron en un procesador Intel Core *i7 - 4700HQ* con una frecuencia de *2.4 Ghz*. Una memoria RAM de *8 GB*. El tipo de sistema operativo Windows 8.1.

- Realizar 50 veces los experimento bajo simulación y obtener datos probabilísticos que nos ayuden a determinar cual de los controladores difusos bajo ciertos parámetros obtienen el menor tiempo de asentamiento.
- Los parámetros del controlador difuso no sufrirán cambios, es decir, permanecerá a lo largo de los experimentos el mismo número funciones de membresía y reglas difusas.
- Los experimentos realizados con el prototipo didáctico de péndulo invertido móvil, serán realizados bajo la misma superficie y aplicando el mismo tipo de perturbación para el controlador sintonizado manualmente y el sintonizado por GFS.
- Para determinar cual es más eficiente, se realizará un mapeo de los datos obtenidos y determinar máximos y mínimos sobre impulsos, una media, mediana y desviación estándar.

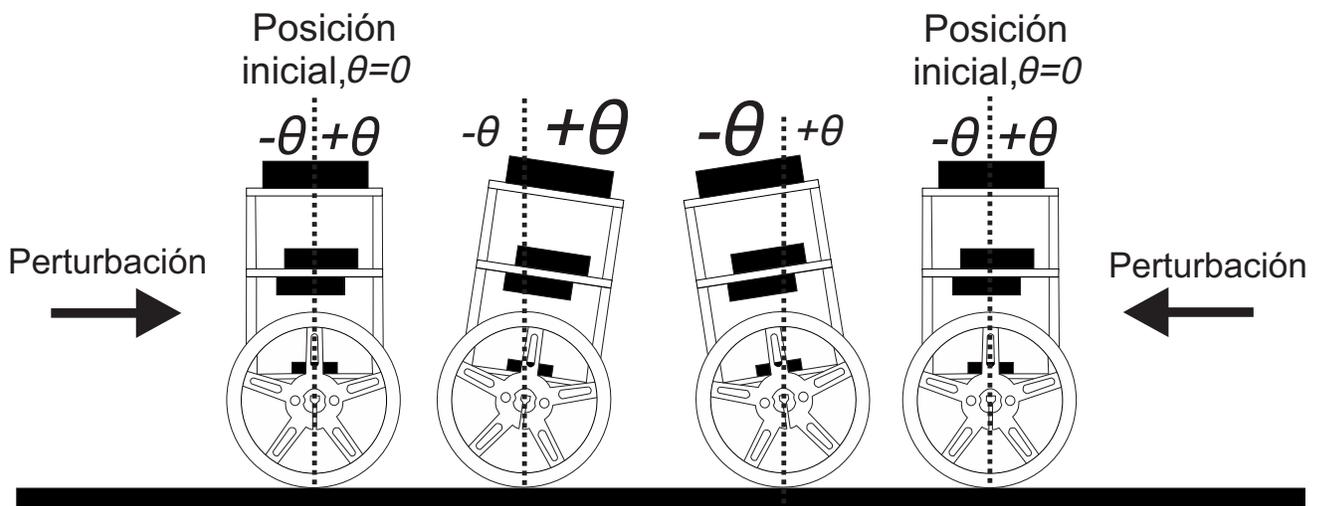


Figura 3.13: Funcionamiento del Péndulo Invertido Móvil

3.6. Diseño del controlador difuso mediante síntesis difusa de Lyapunov

El modelo dinámico del péndulo invertido móvil (2.6) y (2.7) sólo es tomado como referencia para observar las variables a controlar. En este caso se diseñará un controlador difuso para controlar el ángulo de inclinación θ correspondiente al eje z (pitch), por lo tanto tendremos que:

$$x_1 = \theta, \quad x_2 = \dot{\theta}, \quad (3.2)$$

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = A(x_1, x_2) + B(x_1, x_2)u, \quad (3.3)$$

donde,

$$A(x_1, x_2) = A_{43}, \quad B(x_1, x_2) = B_4.$$

Las variables x_1 y x_2 son las variables de estado que se utilizarán para desarrollar el controlador difuso mediante la síntesis de Lyapunov [21]. Para implementar este método en el diseño del controlador difuso es necesario asumir que la dinámica del sistema es desconocida y solo contamos con el conocimiento empírico de la planta, de esta manera podemos establecer que:

- Los estados relevantes de la planta son las variables $x_1 = \theta$ y $x_2 = \dot{\theta}$.
- \dot{x}_2 es proporcional a u , es decir, cuando u incrementa/drecrementa \dot{x}_2 incrementa/decrementa.

El objetivo principal es construir $u(x_1, x_2)$ tal que $(0, 0)^T$, el cual será el punto de equilibrio para la ecuación (3.3). Se propone la siguiente función candidata de Lyapunov $V(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2)$ que satisface claramente las condiciones 1 y 2 de Lyapunov vistas en el capítulo anterior referentes a los conceptos básicos de la teoría de Lyapunov. Ahora bien concentrémonos únicamente en la condición 3 donde tenemos

$$\dot{V} = x_1\dot{x}_1 + x_2\dot{x}_2 = x_1x_2 + x_2\dot{x}_2, \quad (3.4)$$

para satisfacer la condición 3 se tiene:

$$x_1x_2 + x_2\dot{x}_2 < 0. \quad (3.5)$$

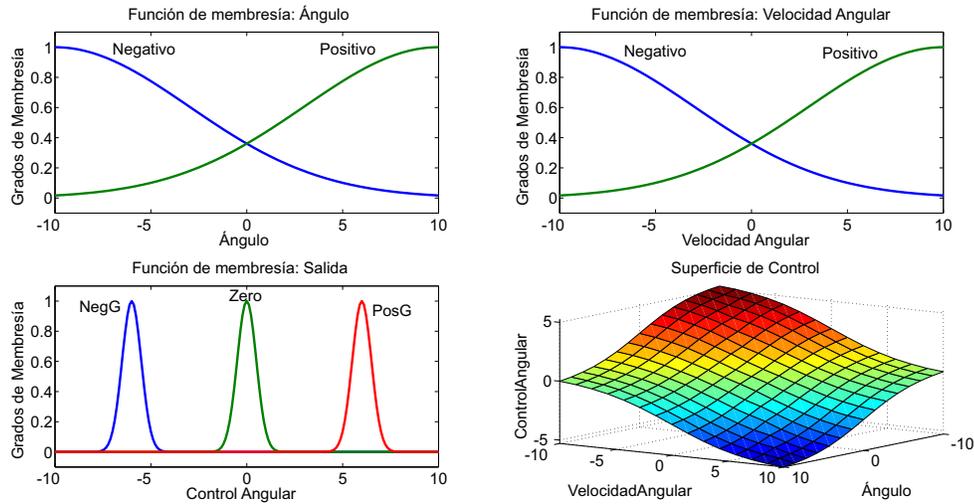


Figura 3.14: Controlador difuso: Sintonización Manual

en algún vecindario cerca a $(0,0)^T$. A partir de esto podemos generar suficientes condiciones para que la ecuación (3.5) se cumpla. Ahora bien si x_1 y x_2 tiene signo positivo entonces $\dot{x}_2 < -x$ (NegG-Negativo Grande), por lo tanto la ecuación (3.5) se cumple; si x_1 y x_2 tiene signo opuesto, entonces $x_1 x_2 < 0$ y la ecuación (3.5) se cumplirá si $\dot{x}_2 = 0$ (Zero-Cero); y si x_1 y x_2 es de signo negativo, entonces la ecuación (3.5) se cumplirá si $\dot{x}_2 > x_1$ (PosG-Positivo Grande). Utilizando estas observaciones y teniendo presente que \dot{x}_2 es proporcional a u , podemos deducir el siguiente conjunto de reglas difusas para el controlador:

- **SI** x_1 es *Positivo* **Y** x_2 es *Positivo* **ENTONCES** u es *NegG*
- **SI** x_1 es *Negativo* **Y** x_2 es *Negativo* **ENTONCES** u es *PosG*
- **SI** x_1 es *Positivo* **Y** x_2 es *Negativo* **ENTONCES** u es *Zero*
- **SI** x_1 es *Negativo* **Y** x_2 es *Positivo* **ENTONCES** u es *Zero*

De acuerdo a este criterio se diseñó el controlador tipo Mamdani compuesto por dos entradas $(\theta, \dot{\theta})$, una salida (u) y cuatro reglas difusas. La salida es calculada mediante el método del centroide. La superficie de control se muestra en la Figura 3.14, la cual presenta cambios suaves en la corrección de la inclinación del ángulo. Esto se logra gracias al conjunto difuso de salida el cual permite una respuesta uniforme.

En las ecuaciones (3.6-3.12) se presenta la distribución de las funciones de membresía del controlador difuso. Las MF fueron sintonizadas manualmente hasta obtener una

respuesta optima del sistema.

Entrada 1: Ángulo

$$\mu_{Negativo}(x) = \exp \left[-\frac{1}{2} \left(\frac{x+10}{7} \right)^2 \right], \quad (3.6)$$

$$\mu_{Positivo}(x) = \exp \left[-\frac{1}{2} \left(\frac{x-10}{7} \right)^2 \right]. \quad (3.7)$$

Entrada 2: Velocidad Angular

$$\mu_{Negativo}(x) = \exp \left[-\frac{1}{2} \left(\frac{x+10}{7} \right)^2 \right], \quad (3.8)$$

$$\mu_{Positivo}(x) = \exp \left[-\frac{1}{2} \left(\frac{x-10}{7} \right)^2 \right]. \quad (3.9)$$

Salida: Control Angular

$$\mu_{NegG}(x) = \exp \left[-\frac{1}{2} \left(\frac{x+6}{0.5} \right)^2 \right], \quad (3.10)$$

$$\mu_{Zero}(x) = \exp \left[-\frac{1}{2} \left(\frac{x-0}{0.5} \right)^2 \right], \quad (3.11)$$

$$\mu_{PosG}(x) = \exp \left[-\frac{1}{2} \left(\frac{x-6}{0.5} \right)^2 \right]. \quad (3.12)$$

3.7. Funcionamiento del sistema genético difuso

Los FLC son aptos para resolver problemas complejos y que son difíciles de expresar matemáticamente. Su desempeño depende mucho del conocimiento empírico acerca del sistema y el conocimiento de técnicas para interpretar la información en forma de reglas difusas y la distribución de las funciones de membresía. Una de las virtudes más sobresalientes de esta técnica, es que no requiere de un conocimiento completo del modelo matemático del sistema, por lo que es posible desarrollar controladores difusos capaces de manipular sistemas de múltiples variables.

Sin embargo, un controlador difuso exige una estructura sólida en cuanto a la selección de las reglas, es decir, el conocimiento empírico aplicado que dará paso al número de reglas en el diseño del controlador difuso, este proceso puede ser un poco tardado y laborioso debido a una formulación de prueba y error en la sintonización de las funciones de membresía.

Una solución viable son los controladores híbridos, como la integración de un Controlador Difuso y un Algoritmo Genético, obteniendo un Sistema Genético Difuso. Los GA se encuentran basados en los principios de la genética y selección natural, esto permite que una población compuesta por muchos individuos evolucione bajo específicas reglas de selección. Básicamente los GFS son capaces de sintonizar los parámetros del controlador difuso propuesto. Algunas de las ventajas son:

- Optimiza de igual forma variables continuas y discretas.
- Permite al sistema aprender y evolucionar.
- Es capaz de manejar múltiples variables.
- Provee de múltiples soluciones óptimas.
- Trabaja con información numérica, experimental o funciones analíticas.

En el Algoritmo 1, se muestra el pseudocódigo de la implementación en forma secuencial del Sistema Genético Difuso. Primero se establece el controlador difuso prediseñado Figura 3.14 y los parámetros de entrada como lo son: el número de población N_{pop} , el porcentaje de selección N_{Sel} , número de generaciones N_{Gen} y una función de costo t_s , en este caso corresponde al tiempo de asentamiento del ángulo y se obtiene evaluando los distintos individuos en la planta. Posteriormente se genera una población inicial

en un arreglo de variables $[C_{N_{var} \times N_{var}}, \dots, C_{N_{pop} \times N_{var}}]$ que conformarán los cromosomas codificados con distintos controladores difusos.

Algoritmo 1 Algoritmo Genético Difuso

```

1:  $FIS \leftarrow$  Controlador difuso predefinido
2:  $(N_{pop}, N_{Sel}, N_{Gen}) \leftarrow$  Variables
3:  $t_s \leftarrow$  Tiempo de asentamiento
4:  $[C_{N_{pop} \times N_{var}}, \dots, C_{N_{pop} \times N_{var}}] \leftarrow$  Población inicial de cromosomas
5:  $i \leftarrow 0$ 
6: while  $i \leq N_{Gen}$  do
7:   for  $N_{pop}$  do
8:      $t_s \leftarrow$  Evaluación de la planta
9:      $[C_{N_{pop} \times N_{var}}, \dots, C_{N_{pop} \times N_{var}}] \leftarrow$  Actualización de cromosomas
10:     $[C_{N_{Sel} \times N_{var}}, \dots, C_{N_{Sel} \times N_{var}}] \leftarrow$  Cruce de mejores cromosomas de MF-Entrada
11:   end for
12:   for  $N_{pop}$  do
13:      $t_s \leftarrow$  Evaluación de la planta
14:      $[C_{N_{pop} \times N_{var}}, \dots, C_{N_{pop} \times N_{var}}] \leftarrow$  Actualización de cromosomas
15:      $[C_{N_{Sel} \times N_{var}}, \dots, C_{N_{Sel} \times N_{var}}] \leftarrow$  Cruce de mejores cromosomas de MF-Salida
16:   end for
17:    $t \leftarrow t+1$ 
18: end while
19: return  $[C_{N_1 \times N_{var}}, \dots, C_{N_1 \times N_{var}}] \leftarrow$  El mejor cromosoma

```

La propuesta que se hace para la creación de la población inicial es tomar un FRBS predefinido, y aplicar un Algoritmo de Ajuste Simple [40] (*STA*, por sus siglas en inglés). El STA se encuentra basado en las propiedades de la superficie de control, esto permite manipular los rangos de las funciones de membresía de las variables de entrada y salida. De esta manera las funciones de membresía se contraen y expanden gracias al factor de ajuste k el cual se le asigna un valor aleatorio de entre 0 y 1. Cabe mencionar que la técnica presentada se encuentra reportada para lógica difusa tipo 1, sin embargo un trabajo más recientes [41] muestra el empleo de la técnica del STA en lógica difusa tipo 2.

Una vez establecida la población inicial, todos los cromosomas son evaluados en el sistema. En este caso la función de costo se encuentra basada en el menor tiempo de asentamiento t_s , correspondiente a la respuesta del sistema ante una función impulso. Se actualizan los cromosomas $[C_{N_{pop} \times N_{var}}, \dots, C_{N_{pop} \times N_{var}}]$ según su costo y se ordenan de manera ascendente. Los mejores individuos para su cruce son seleccionados mediante un peso, donde la probabilidad de los cromosomas para aparearse se encuentra inversamente

proporcional a su costo. La probabilidad de un rango n de los cromosomas mediante

$$p_n = \frac{N_{keep} - n + 1}{\sum_{N_{keep}}^{n=1} n}. \quad (3.13)$$

Posteriormente un número de entre cero y uno es generado aleatoriamente, comenzando por el valor superior de la lista. Por lo tanto el cromosoma con un valor p_n que se acerque al número aleatorio será seleccionado y así sucesivamente hasta formar las parejas para su reproducción. Los parámetros de las funciones de membresía se conforman por valores codependientes, como lo son los rangos entre cada función de membresía y las restricciones asignadas a cada tipo de MF (triangular, trapezoidal, gaussiana, etc.). Es por eso que para realizar el cruce entre dos cromosomas se propone lo siguiente:

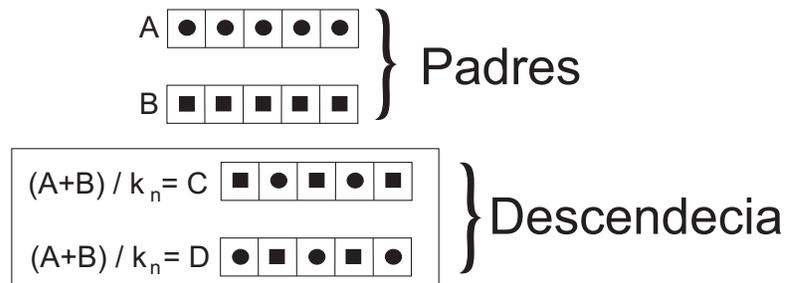


Figura 3.15: Cruce de los individuos

La nueva descendencia C y D son generadas a partir de la suma de los padres seleccionados A y B , y divididos entre un factor k_n que define el intervalo de expansión o contracción de las funciones de membresía Figura 3.15. Mediante esta técnica la nueva descendencia puede sintonizar un valor apropiado para las funciones de membresía de modo que se pueda encontrar el mínimo t_s . El factor k_n puede ser asignado de manera aleatoria para diversificar el método. Acto seguido se actualizan los cromosomas $[C_{N_{Sel} \times N_{var}}, \dots, C_{N_{Sel} \times N_{var}}]$.

El cruce de los cromosomas se realiza principalmente para las funciones de membresía de entrada y posteriormente para las de salida. Los procesos de evaluación, selección y cruce se realizarán de manera iterativa hasta culminar con el número de generaciones N_{Gen} establecido. Finalmente se obtendrá el mejor cromosoma $[C_{N_1 \times N_{var}}, \dots, C_{N_1 \times N_{var}}]$.

Capítulo 4

Experimentos y análisis de resultados

En este capítulo se presentan los experimentos realizados y los resultados obtenidos. De manera estructurada se analizó el comportamiento de los controladores, tratando como punto principal el tiempo de asentamiento del ángulo t_s , tomando como base un muestreo probabilístico para el dictamen de una conclusión concisa de los resultados. El primer experimento consistió en evaluar el desempeño del GFS bajo distintos parámetros para determinar el controlador difuso óptimo a implementar. El segundo experimento consistió en observar la respuesta al impulso unitario del sistema en lazo cerrado del controlador difuso sintonizado manualmente y el sintonizado por el GFS, en simulación.

En el tercer experimento se implementó el controlador difuso sintonizado manualmente y por GFS en el prototipo diseñado, realizando un muestreo del ángulo de primera instancia sin perturbaciones y posteriormente con perturbaciones con la finalidad de comparar los resultados obtenidos en la simulación y poder analizar más a fondo los resultados contrastados con la teoría.

4.1. Simulación

El experimento consistió en evaluar el GFS bajo ciertos parámetros y establecer una métrica que nos aporte resultados substanciales y permita la elección de un controlador difuso óptimo. El experimento se realizó 50 veces obteniendo el valor máximo y mínimo, la media, mediana y desviación estándar del tiempo de asentamiento t_s , y el tiempo de ejecución del algoritmo en promedio para cada caso. Se estableció un rango de entre 10 a 50 generaciones, con 16, 32 y 64 individuos. El hardware y software utilizados a lo largo de este trabajo de tesis se muestran en la Tabla 4.1.

Tabla 4.1: Características del hardware y software utilizados, para realizar los experimentos

Hardware y software	
Procesador	Intel Core i7-4700HQ 2.4 Ghz
RAM	8 GB
Sist. Ope.	Windows 8.1
Simulación	MATLAB 2013A

En base a los datos experimentales mostrados en la Tabla 4.2, se seleccionó el controlador difuso. Se establecieron 16, 32 y 64 individuos, con un rango de entre 10 a 50 generaciones. El experimento fue realizado 50 veces, del cual se obtuvo los máximos y mínimos tiempos de asentamiento t_s (*seg*), una media, mediana, desviación estándar y tiempo de ejecución del algoritmo en promedio. Podemos observar que a medida que el número de generaciones e individuos aumenta, el tiempo de asentamiento del ángulo disminuye, sin embargo el tiempo de ejecución incrementa. Por lo tanto se decidió tomar un valor intermedio, donde el número de generaciones e individuos nos permita obtener un tiempo de asentamiento y un tiempo de ejecución razonables.

El controlador difuso seleccionado fue aquel con un $t_s = 3.8384$ (*seg*), bajo 30 generaciones, 64 individuos y un tiempo de ejecución de 51.22 (*min*). Los valores correspondientes a las funciones de membresía se muestran en las ecuaciones 4.1-4.7.

Tabla 4.2: Máximos y mínimos tiempos de asentamiento t_s (*seg*) del ángulo, media, mediana, desviación estándar y tiempo de ejecución, obtenidos del algoritmo genético difuso evaluado bajo distintas generaciones y número de individuos.

Parámetros		Tiempo de asentamiento T_s					
Individuos	Generaciones	Máy.	Mín.	Media	Mediana	D. Est.	T. Ejec.
64	50	4.0169	3.8024	3.8218	3.8898	0.0661	84.17
	40	4.0177	3.8016	3.8249	3.9276	0.0714	69.01
	30	4.0211	3.8298	3.8384	3.9536	0.0460	51.22
	20	4.1118	3.9006	3.9856	4.0208	0.0674	35.29
	10	4.1979	3.9580	4.0558	4.0955	0.0778	19.05
32	50	4.0184	3.8517	3.9316	3.9594	0.0640	42.20
	40	4.0268	3.9064	3.9694	3.9895	0.0529	33.21
	30	4.1553	3.9564	4.0494	4.0826	0.0765	26.46
	20	4.1843	3.9860	4.0869	4.1205	0.0844	14.10
	10	4.2689	4.0118	4.1586	4.1158	0.0825	9.22
16	50	4.1124	3.9146	4.0204	4.0533	0.0893	20.21
	40	4.2181	4.0421	4.1687	4.1394	0.0826	15.50
	30	4.2560	4.0689	4.1732	4.0244	0.0887	12.57
	20	4.2866	4.0547	4.1740	4.1554	0.0814	11.39
	10	4.2983	4.1120	4.1934	4.2244	0.0958	5.24

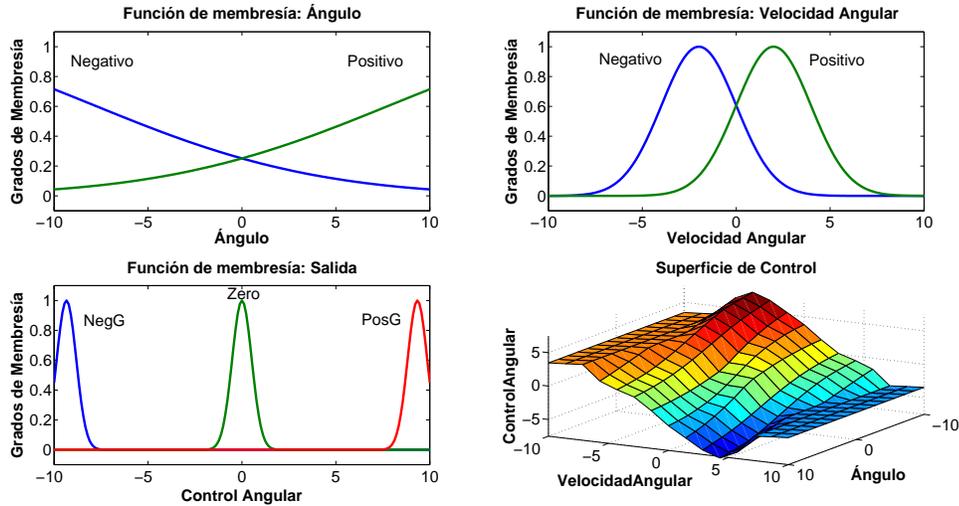


Figura 4.1: Controlador difuso: Sintonización manual. La distribución de las funciones de membresía se encuentran en grados.

Haciendo una comparación entre el controlador original (sintonización manual) mostrado en la Figura 3.14 y el controlador sintonizado por el GFS mostrada en la Figura 4.1 podemos observar como la distribución de las funciones de membresía del controlador original ha cambiado y además se logra el objetivo de control, el cual consiste en realizar el control de estabilidad en el menor tiempo posible t_s . Los valores sintonizados por el GFS, hubiesen sido obtenidos mediante la sintonización manual, pero se considera poco práctico el caso de prueba y error, además del tiempo invertido en encontrar los valores indicados podría superar a los tiempos de ejecución del algoritmo. Sin embargo gracias a las capacidades del GFS los valores sintonizados mostraron cambios sustanciales en las pruebas realizadas.

Entrada 1: Ángulo

$$\mu_{Positivo}(x) = \exp \left[-\frac{1}{2} \left(\frac{x + 19.73}{11.87} \right)^2 \right], \quad (4.1)$$

$$\mu_{Negativo}(x) = \exp \left[-\frac{1}{2} \left(\frac{x - 19.70}{11.9} \right)^2 \right]. \quad (4.2)$$

Entrada 2: Velocidad Angular

$$\mu_{Positivo}(x) = \exp \left[-\frac{1}{2} \left(\frac{x + 1.99}{1.99} \right)^2 \right], \quad (4.3)$$

$$\mu_{Negativo}(x) = \exp \left[-\frac{1}{2} \left(\frac{x - 1.99}{1.99} \right)^2 \right]. \quad (4.4)$$

Salida: Control Angular

$$\mu_{NegG}(x) = \exp \left[-\frac{1}{2} \left(\frac{x + 9.342}{0.5207} \right)^2 \right], \quad (4.5)$$

$$\mu_{Zero}(x) = \exp \left[-\frac{1}{2} \left(\frac{x - 0}{0.5207} \right)^2 \right], \quad (4.6)$$

$$\mu_{PosG}(x) = \exp \left[-\frac{1}{2} \left(\frac{x - 9.342}{0.5207} \right)^2 \right]. \quad (4.7)$$

4.2. Respuesta del ángulo: Simulación

En las Figuras 4.2 y 4.3 podemos observar en color rojo la respuesta del ángulo mediante la sintonización manual, la cual presenta oscilaciones con una amplitud mínima de 0.01 (rad/s) , un tiempo de asentamiento de 6.24 (seg) y un pico máximo de 0.6 (rad/s) en presencia del impulso. Por otra parte en color azul tenemos la respuesta del ángulo mediante la sintonización por GFS, donde la amplitud mínima de las oscilaciones y el pico máximo en presencia del impulso es de 0.005 (rad/s) y 0.15 (rad/s) respectivamente. El tiempo de asentamiento se redujo hasta 3.83 (seg) .

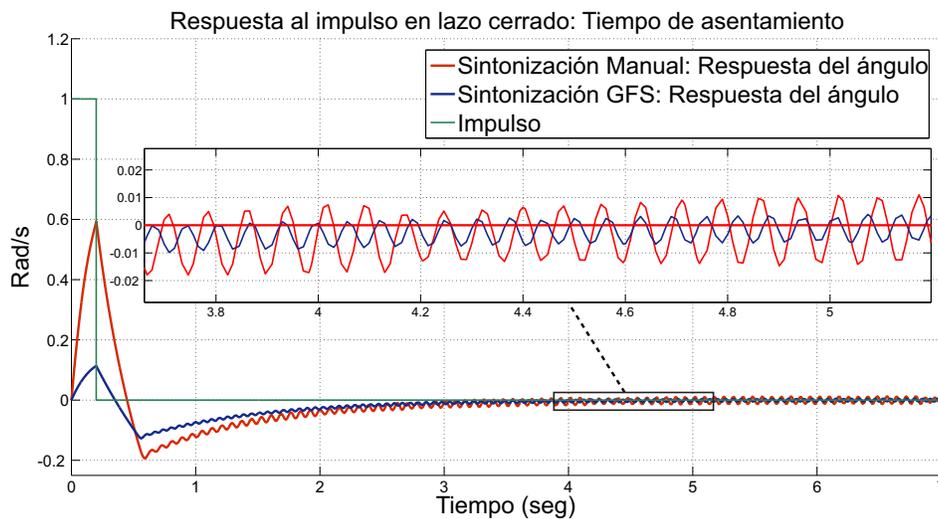


Figura 4.2: Controlador difuso: Tiempo de asentamiento, sintonización por GFS

En contraste a los resultados obtenidos se comprueba lo establecido por la síntesis difusa de Lyapunov para el diseño del controlador difuso, demostrando la estabilidad del robot alrededor del punto de equilibrio, siendo este el principal objetivo de control.

El punto de equilibrio para el sistema de péndulo invertido móvil, es cuando el vehículo se encuentra perpendicular a la superficie sobre la que se desplaza, sin embargo se pueden generar perturbaciones en diferentes direcciones, es decir, izquierda y derecha. En este caso el lado derecho corresponde a un impulso positivo y el lado izquierdo a un impulso negativo. En las Figuras 4.4 y 4.5 podemos observar las respuestas en simulación ante las perturbaciones mencionadas.

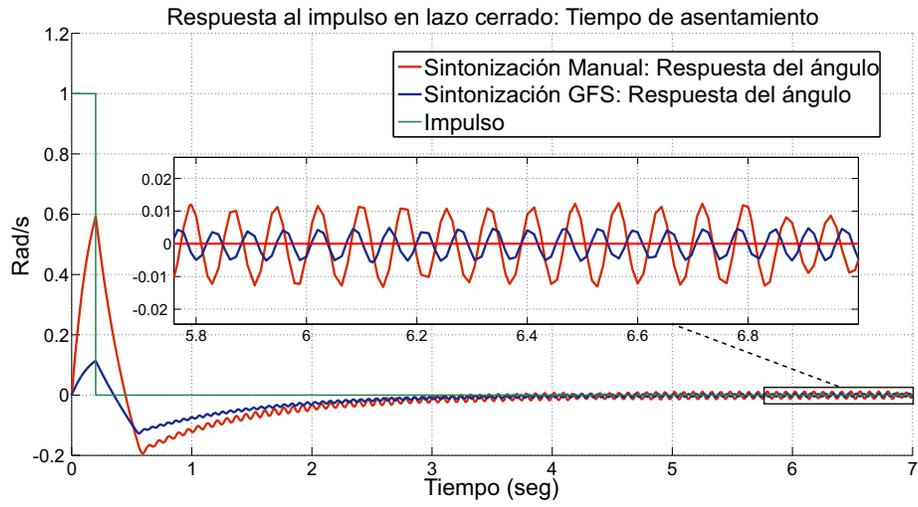


Figura 4.3: Controlador difuso: Tiempo de asentamiento, sintonización manual

Tanto el controlador sintonizado de manera manual y por GFS responden según lo esperado con tiempos de asentamiento ya mencionados, al presentarse un impulso cada 8 segundos, esto equivaldría a un empujón seguido de otro en la misma dirección. Este experimento fue realizado en el modelo físico y en simulación con la finalidad de observar la respuesta de ambos controladores.

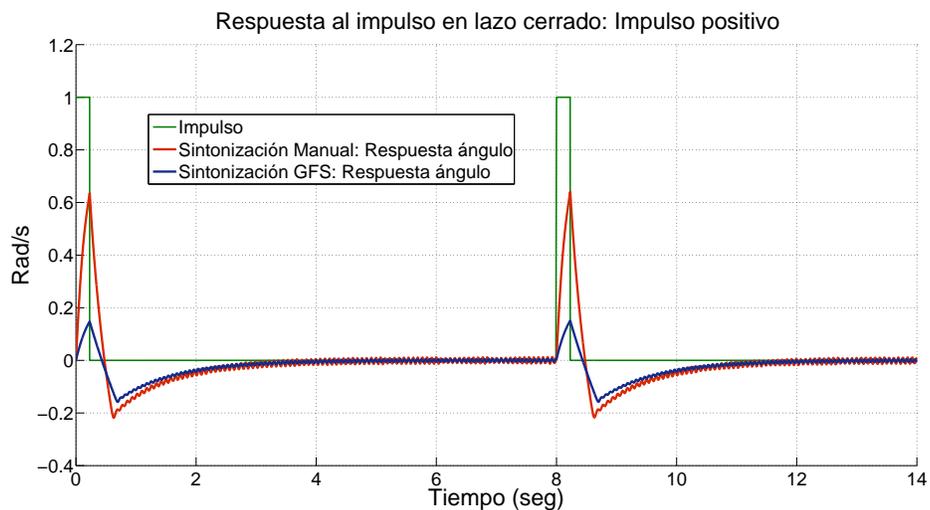


Figura 4.4: Controlador difuso: Sintonización manual y por GFS, impulso positivo

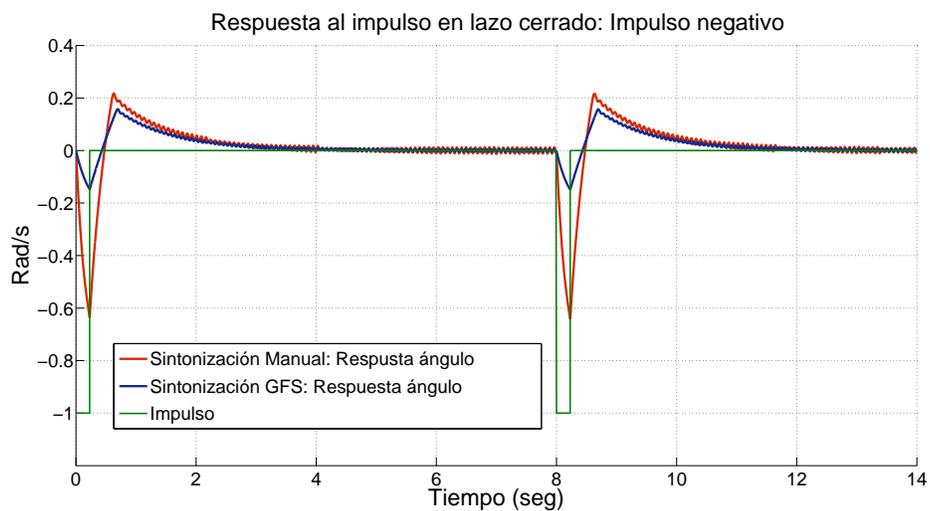


Figura 4.5: Controlador difuso: Sintonización manual y por GFS, impulso negativo

4.3. Respuesta del ángulo sin perturbaciones: Prototipo didáctico

En la Figura 4.6 podemos observar la respuesta de control del ángulo de ambos controladores en el prototipo didáctico diseñado. Nuevamente en color rojo tenemos el controlador difuso sintonizado manualmente el cual presenta mayor amplitud en sus oscilaciones que van casi hasta los 4.43° , mientras que el controlador difuso optimizado (color azul) logra disminuir las oscilaciones con picos máximos de 1.96° .

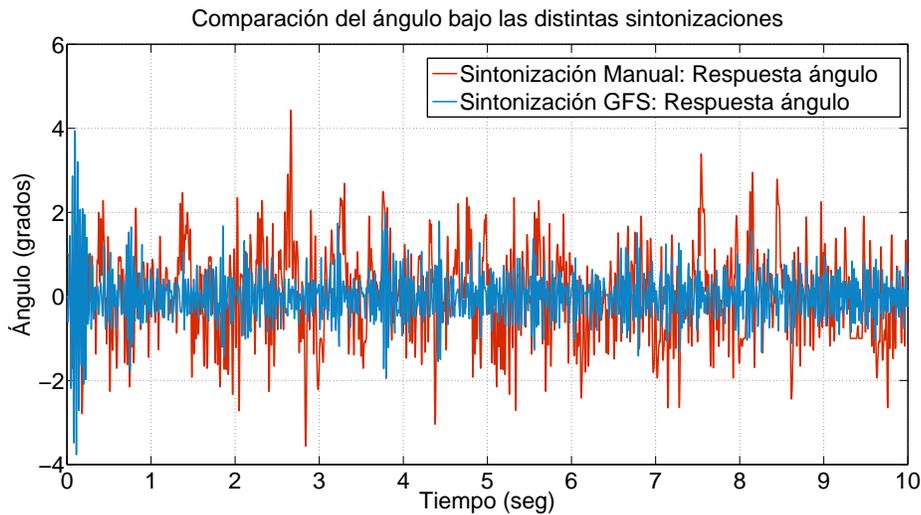


Figura 4.6: Controlador difuso: Sintonización manual y por GFS

Si hacemos un acercamiento (Figura 4.7), podremos observar el comportamiento de ambos controladores similar a las pruebas realizadas en simulación. En la Tabla 4.3 se muestra los valores empíricos del sistema real evaluado mediante los experimentos de los cuales se extrajeron datos con un muestreo 700 muestras/s.

Tabla 4.3: Parámetros empíricos del sistema: Comportamiento del ángulo (grados), sin perturbaciones

Controlador	Máx.	Mín.	Media	Mediana	D. Est.
Sintonización Manual	4.43	2.264	0.518	0.658	0.9648
Sintonización GFS	1.96	0.562	0.0115	0.030	0.5168

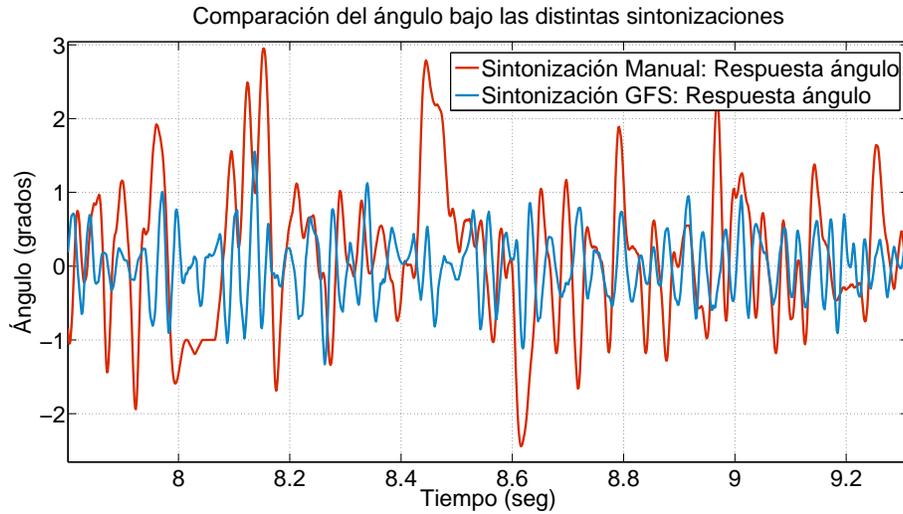


Figura 4.7: Controlador difuso: Sintonización manual y por GFS (acercamiento)

En la Figura 4.7 podremos observar mejor la corrección del ángulo del sistema sin perturbaciones durante 10 segundos. En comparación con los controladores difusos sintonizados manualmente y por GFS simulados se comprueba la teoría. El controlador en color rojo presenta mayor amplitud en sus oscilaciones que el de color azul que es más estable. En la siguiente sección se muestran valores empíricos del sistema.



Figura 4.8: Controlador difuso: Sintonización manual

Posteriormente ambos controladores fueron analizados por separado. Primero se observó la corrección del ángulo del controlador sintonizado manualmente sin perturbaciones como se muestra en las Figuras 4.8 y 4.9, el cual obtuvo una media de 0.518, con valores máximos de sobreimpulso de 4.43° y mínimos de 2.264° .

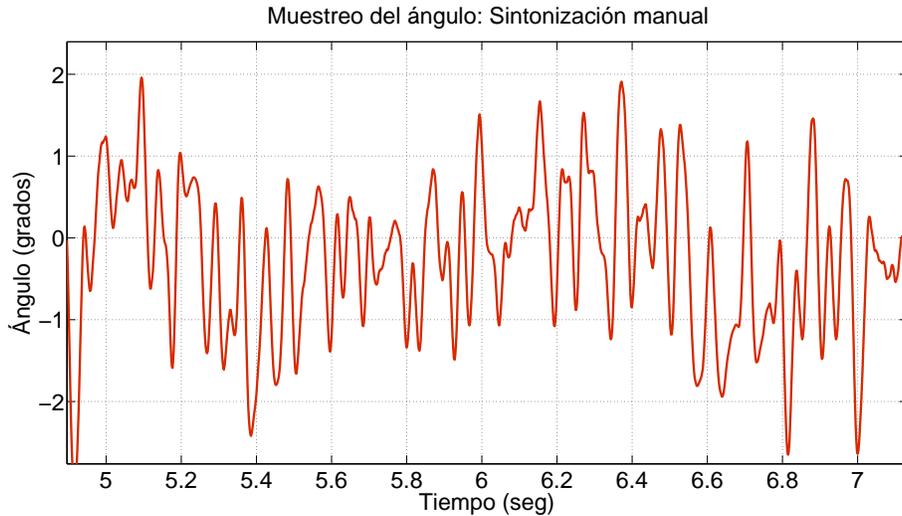


Figura 4.9: Controlador difuso: Sintonización manual (acercamiento)

Después la misma prueba fue realizada para el controlador difuso sintonizado por el GFS como se muestra en las Figuras 4.10 y 4.11, en el cual se obtuvo una media de 0.0115, con máximos sobre impulsos de 1.96° y mínimos de 0.562° . Los parámetros empíricos del sistema se muestran en la Tabla 4.3. Este controlador demostró mayor estabilidad que el anterior, disminuyendo las oscilaciones en amplitud y por ende se redujo el error del ángulo de inclinación.

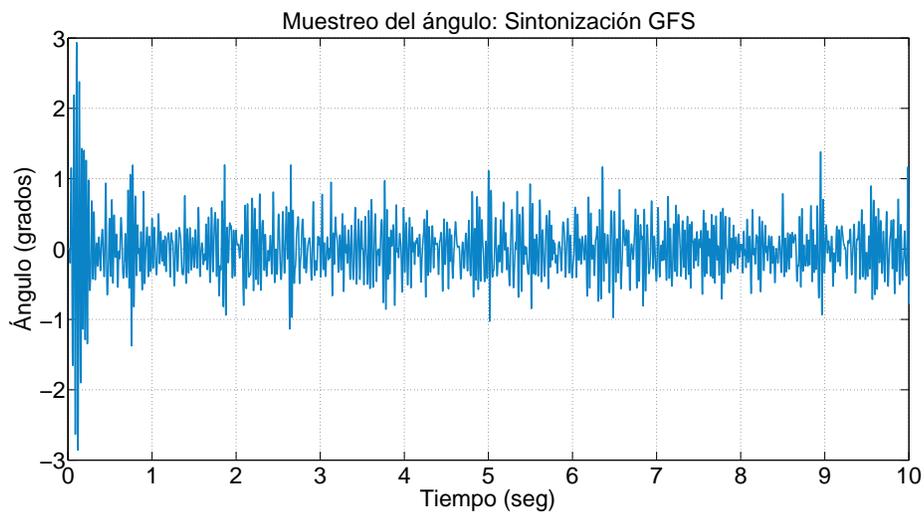


Figura 4.10: Controlador difuso: Sintonización por GFS

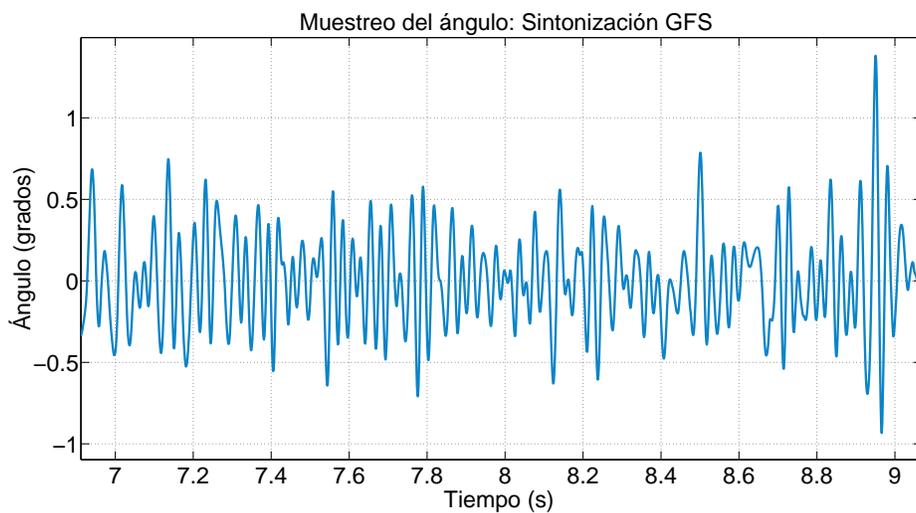


Figura 4.11: Controlador difuso: Sintonización por GFS (acercamiento)

4.4. Respuesta del ángulo ante perturbaciones: Prototipo didáctico

La segunda prueba consistió en aplicar una fuerza en ambos sentidos, es decir, un empujón del lado derecho y otro del lado izquierdo. En la Figura 4.12 se muestra la respuesta del controlador difuso sintonizado manualmente, en el cual tenemos máximos sobre impulsos de hasta 9 grados. Con tiempos de recuperación de 0.82 segundos (impulso negativo) como se observa en la Figura 4.13 y 0.24 segundos (impulso positivo) como se muestra en la Figura 4.14.

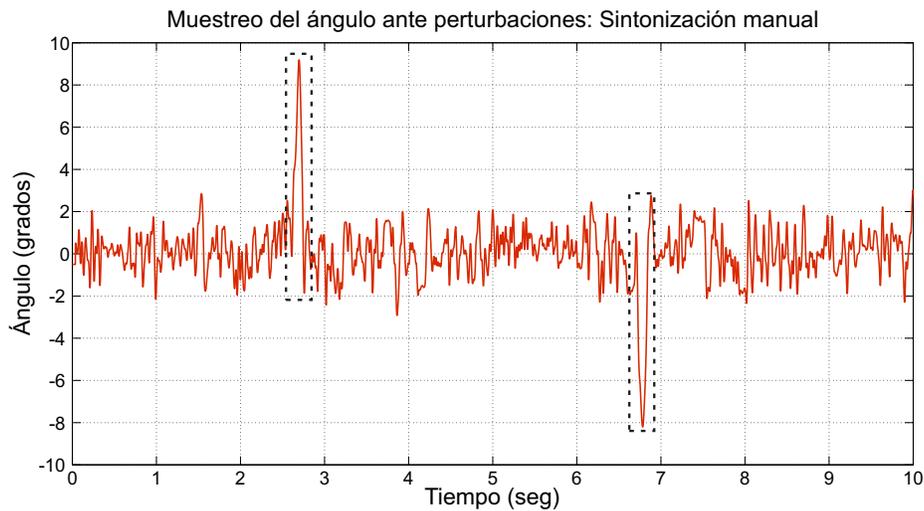


Figura 4.12: Controlador difuso ante perturbaciones: Sintonización manual

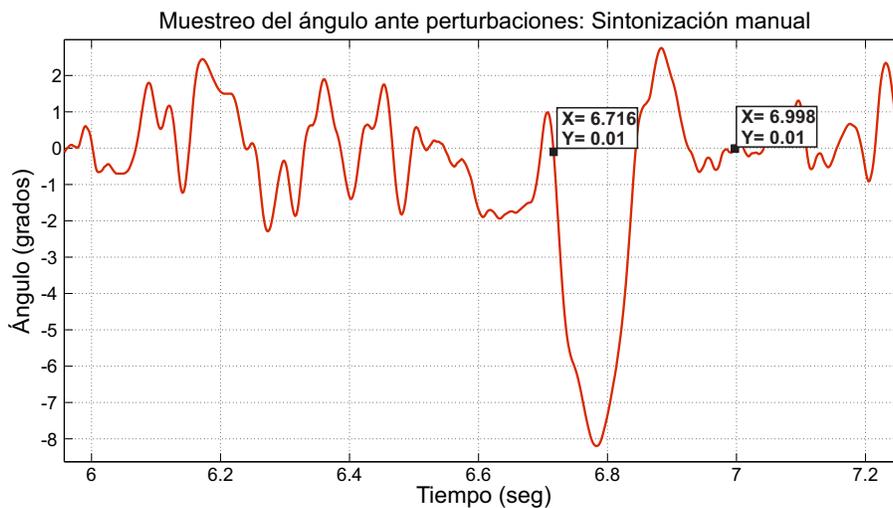


Figura 4.13: Controlador difuso ante perturbaciones (impulso negativo): Sintonización manual

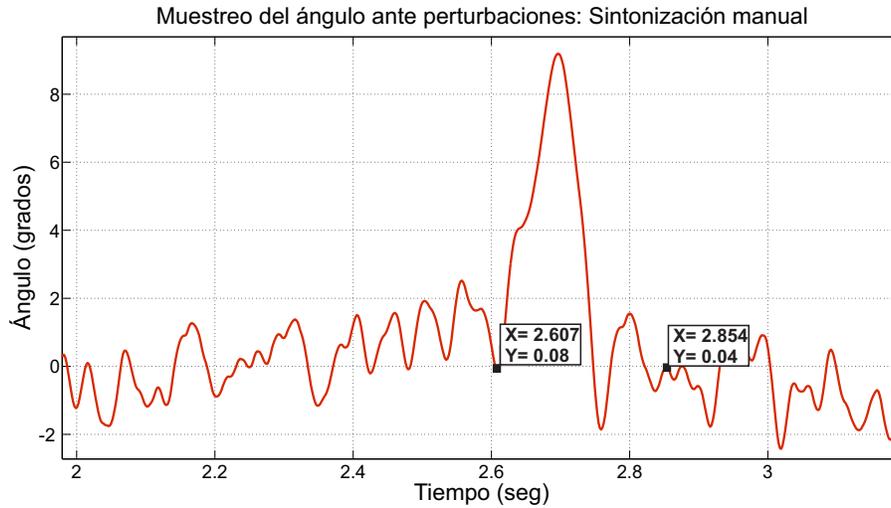


Figura 4.14: Controlador difuso ante perturbaciones (impulso positivo): Sintonización manual

Para el controlador sintonizado por GFS se obtuvo una mejor respuesta tanto en tiempo de asentamiento y máxima amplitud causada por las perturbaciones. En la Figura 4.14 y 4.15 las perturbaciones se realizaron con más frecuencia con el fin de probar la eficiencia del controlador, comenzando por pequeños empujones hasta incrementarlos de manera cuidadosa. Se presentan sobreimpulsos de hasta 8.5° en los casos más extremos. Con tiempos de recuperación de 0.43 segundos (impulso negativo) como se muestra en la Figura 4.16 y 0.23 segundos (impulso positivo) como se observa en la Figura 4.13.

Tabla 4.4: Parámetros empíricos del sistema: comportamiento del ángulo (grados), ante perturbaciones

Controlador	Máx.	Mín.	Media	Mediana	D. Est.
Sintonización Manual	9.19	2.93	0.925	0.110	1.0230
Sintonización GFS	3.74	0.571	0.024	0.042	0.3825

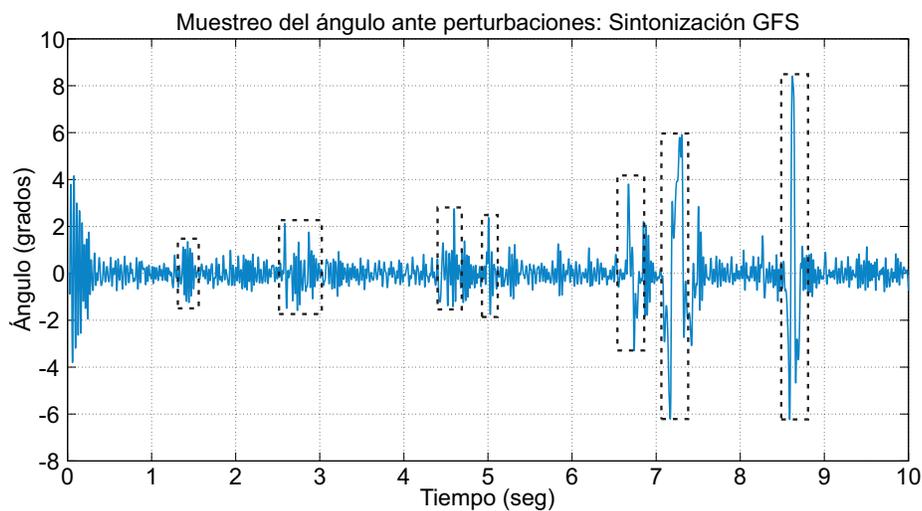


Figura 4.15: Controlador difuso ante perturbaciones: Sintonización por GFS

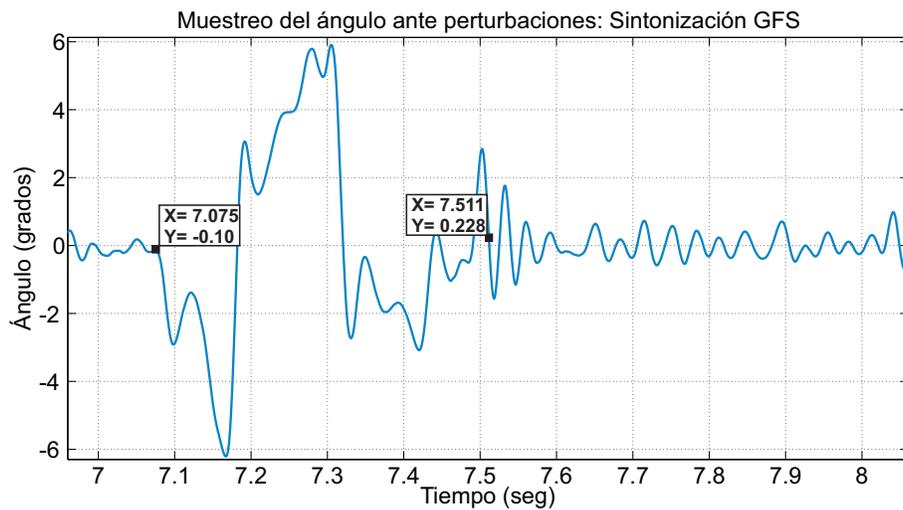


Figura 4.16: Controlador difuso ante perturbaciones (impulso positivo): Sintonización por GFS

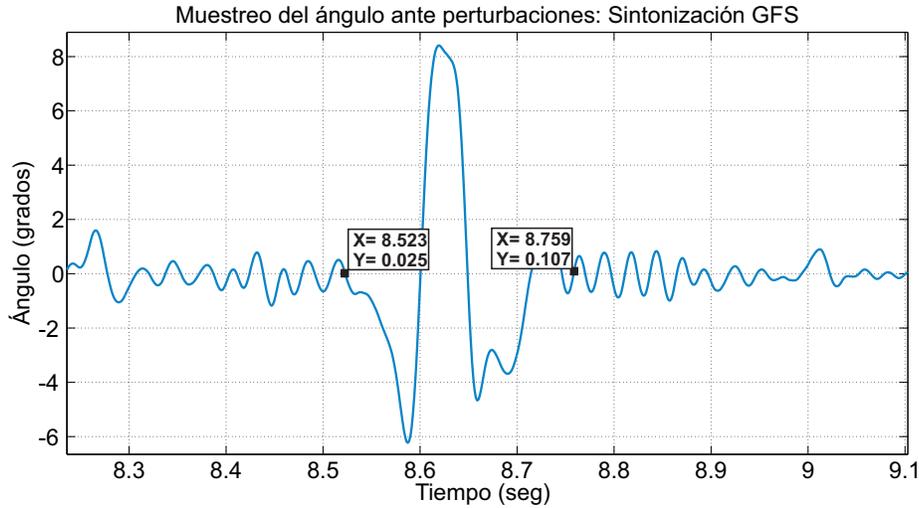


Figura 4.17: Controlador difuso: Sintonización manual

En las Tablas 4.4 y 4.3 se observa que la amplitud del error angular ante perturbaciones se reduce hasta un 40 % y el tiempo de asentamiento en un 50 %. Esto se logra gracias a la optimización previa del controlador difuso mediante el sistema genético difuso.

En la literatura la mayor parte de los controladores difusos son diseñados a base de prueba y error, con varias funciones de membresía elegidas de manera empírica [19, 42]. La metodología propuesta en este trabajo de tesis presenta gran ventaja sobre los métodos presentados en [43, 44, 24], donde se consigue el objetivo de control a costa de un alto costo computacional. Adicionalmente en este trabajo, se evitan problemas de hardware, donde el movimiento del prototipo se encuentra limitado al estar conectado a una computadora para su funcionamiento.

Capítulo 5

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones del trabajo de tesis realizado y finalmente se presentan algunas sugerencias como trabajo futuro.

5.1. Conclusiones

En este trabajo de tesis se demostró la eficiencia del método propuesto como alternativa al diseño tradicional de los controladores difusos, que permite la estabilización del sistema de Péndulo Invertido Móvil y además se consigue utilizando el menor número de reglas difusas mediante la Síntesis Difusa de Lyapunov, la cual garantiza la estabilidad del sistema al establecer el conjunto de reglas que conforman al controlador difuso. Esto se ve reflejado en las pruebas realizadas en el prototipo didáctico utilizado durante este trabajo de tesis.

Por otra parte la integración de los Algoritmos Genéticos permite sintonizar las funciones de membresía con la finalidad de mejorar la respuesta del controlador difuso. Esta propuesta denominada Sistema Genético Difuso utiliza los paradigmas del cómputo evolutivo, dotando al sistema con herramientas que le proporcionan un comportamiento semejante al de la Inteligencia Artificial.

Mediante la técnica propuesta se logró reducir adecuadamente el tiempo de asentamiento del ángulo de inclinación θ y asimismo la amplitud en las oscilaciones en la corrección del error angular. Otro aspecto notable es la implementación del Algoritmo de Ajuste Simple para la creación de población, el cual demostró ser bastante versátil, debido a que logra el mayor número de individuos aptos para formar parte de la solución sin la necesidad de evaluar a cada uno de ellos, descartar individuos y generar nueva población

en el peor de los casos, de esta manera se logró disminuir el tiempo computacional e incrementar la diversidad de los cromosomas.

Como resultado de la integración de las herramientas utilizadas a lo largo de este trabajo de tesis, se consiguió exitosamente concretar los objetivos propuestos para resolver el problema de estabilización de un Péndulo Invertido Móvil. Además, se culminó el diseño y construcción de un prototipo didáctico de TWIP, el cual puede ser utilizado en la realización de pruebas de algoritmos de cómputo inteligente.

5.2. Trabajo futuro

Algunas de las propuestas como trabajo futuro se exponen a continuación:

- La aplicación del controlador por GFS fue realizada de manera fuera de línea debido a las capacidades de procesamiento del microcontrolador utilizado. Por lo tanto se propone realizar el controlador GFS en línea, integrando el controlador en el sistema móvil, esto permitirá dotar al sistema de ciertas habilidades contempladas en los siguientes puntos.
- Como una consecuencia de lo anterior, un esquema de aprendizaje en el controlador por GFS puede ser realizada y el sistema será capaz de sintonizar las funciones de membresía bajo ciertos escenarios y situaciones al que sea expuesto, además de optimizar los parámetros del algoritmo genético difuso.
- Redimensionar el prototipo de péndulo invertido móvil, puede ser requerido una vez integrando el controlador por GFS en el sistema, principalmente por las capacidades de procesamiento que requiere el algoritmo y además, será posible evaluar el rendimiento del sistema completo como un vehículo, con la capacidad de ser tripulado.
- Una oportunidad en la que se puede indagar y sacar provecho es en la paralelización del algoritmo de GFS, permitiendo al sistema evaluar y determinar posibles soluciones en un tiempo más corto, debido a que esta técnica del cómputo paralelo ayudaría a acelerar el tiempo de cómputo del algoritmo, aumentando considerablemente su rendimiento.

Referencias

- [1] Rafael Kelly and Victor Santibáñez. *Control de movimiento de robots manipuladores*. Pearson, 2003.
- [2] Laxmidhar Behera and Indrani Kar. *Intelligent systems and control: Principles and applications*. Oxford, 2009.
- [3] Liu W. Zhen He Bin and Lv H. Feng. Mathematical model and control strategy of a two-wheeled self balancing robot. *IEEE, Institute of Electrical and Electronics Engineers*, 10:4198–4199, 2013.
- [4] Silvio Colombi Felix Grasser, Aldo D’Arrigo and Alfred Rufer. Joe: A mobile, inverted pendulum. *IEEE, Transactions on Industrial Electronics*, 49(1):107–114, 2002.
- [5] Seth Hutchinson Mark W. Spong and M. Vidyasagar. *Robot modeling and control*. Wiley, November 18, 2005.
- [6] Oxford Dictionaries. <http://www.oxforddictionaries.com>.
- [7] M. Yamakita K. Furuta and S. Kobayashi. Swing-up control of inverted pendulum using pseudo-state feedback. *Journal of Systems and Control Engineering*, 14:263–269, 1992.
- [8] Olfa Boubaker. The inverted pendulum benchmark in nonlinear control theory: A survey. *Intech, International Journal of Advanced Robotic Systems*, 10(233):1–9, 2013.
- [9] J. Á. Acosta. Furuta’s pendulum: A conservative nonlinear model for theory and practise. *Mathematical Problems in Engineering*, 10:1–30, 2010.
- [10] Victor M. Guzmán Mayra A. Cruz. Modeling, simulation, and construction of a furuta pendulum test-bed. *IEEE*, 25:72–79, 2015.

-
- [11] Min Wu Ancai Zhang, Xuzhi Lai and Jinhua She. Stabilisation of inderactuated two-link gymnast robot by using trajecyory tracking strategy. *Elsevier, Applied Mathematics and Computation*, 253:193–204, 2015.
- [12] Mark Spong. The swing up control problem for the acrobot. *IEEE, Control Systems Magazine*, 15(1):49–55, 1995.
- [13] R. Lozano I. Fantoni and Mark Spong. Energy based control of the pendubot. *IEEE, Transactions on Automatic Control*, 45(4):725–729, 2000.
- [14] Myunghwan Eom and Dongkyoung Chwa. Robust swing-up balancing control using a nonlinear disturbance observer for the pendubot system with dynamic friction. *IEEE Transactions and Robotics*, 31(2):331–343, 2015.
- [15] Randy L. Haupt and Sue Ellen Haupt. *Practical genetic algorithms*. The MIT Press, 1998.
- [16] Melanie Mitchell. *An introduction to genetic algorithms*. Wiley, November 18, 2005.
- [17] Janusz Kacprzyk and Witold Pedrycz. *Handbook of computational intelligence*. Springer, 2015.
- [18] Lotfi A. Zadeh. Outline of a new approach to the analysis of complex system and decision processes. *IEEE, Transactions Systems, Man, and Cybern*, 3(1):28–44, 1973.
- [19] Junfeng Wu and Wanying Zhang. Design of fuzzy logic controller for two-wheeled self-balancing robot. *The 6th International Forum on Strategic Technology*, 2:1266–1270, 2011.
- [20] Antonio Sala Thierry M. Guerra and Kazuo Tanaka. Fuzzy control turns 50: 10 years later. *Elsevier, Fuzzy sets and systems*, 281:168–182, 2015.
- [21] Michael Margaliot and Gideon Langholz. Fuzzy lyapunov-based approach to the desing of fuzzy controllers. *Elsevier, Fuzzy Sets and Systems*, 106:49–59, 1999.
- [22] F. Herrera F. Hoffmann O. Cordón, F. Gomide and L. Magdalena. Ten years of genetic fuzzy systems: current framework and new trends. *Elsevier, Fuzzy Sets and Systems*, 3:5–31, 2004.
- [23] Frank Hoffmann Oscar Cordón, Francisco Herrera and Luis Magdalena. *Genetic fuzzy systems: Evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific, 2001.

-
- [24] Ying Zhang Shenyong Gao, Huaixiang Zhang and Bo Zhang. Optimization to the inverted pendulum system by genetic fuzzy strategies. *IEEE*, 24:1562–1565, 2012.
- [25] Erwin Sitompul and Iksan Bukhori. A new approach in self-generation of fuzzy logic controller by means of genetic algorithm. *IEEE, 6th International Conference on Information Technology and Electrical Engineering*, 6:1–6, 2014.
- [26] Oscar Cordón. A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *Elsevier, International Journal of Approximate Reasoning*, 52:894–913, 2011.
- [27] Sumit Mukhopadhyay Shamayita Mukherjee, Shashank Pandey and Nirmal B. Hui. Digital pendulum system: Genetic fuzzy-based online tuning of pid controller. *IEEE, 8th Proceedings International Conference on Intelligent Systems and Control (ISCO)*, 8:23–28, 2014.
- [28] Marley M. Vellasco Adriano S. Koshiyama, Ttiana Escovedo and Ricardo Tanscheit. Gpfs-control: A fuzzy genetic model for control task. *IEEE, International Conference on Fuzzy Systems*, 4(3):167–179, 2014.
- [29] Segway Simply Moving. <http://www.segway.com>, 2015.
- [30] Lofti A. Zadeh. A fuzzy algorithmic approach to the definition of complex or imprecise concepts. *Int. J. Man-Machine Studies*, 8(3):249–291, 1976.
- [31] E. H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *Elsevier, International Journal of Approximate Reasoning*, 26(12):751–766, 2006.
- [32] Joabe A. Silva and Ginalber L. Serra. Takagi-sugeno fuzzy control based on robust stability specifications. *Intech, Fuzzy Controllers, Theory and Applications*, 2(65):45–68, 2011.
- [33] Nazmul Siddique and Hojjat Adeli. *Computational intelligence, synergies of fuzzy logic, neural networks and evolutionary computing*. Wiley, 2013.
- [34] John H. Holland. *Adaptation in natural and artificial systems*. Oxford, 1975.
- [35] Francisco Herrera. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Springer, Verlag*, 1:27–46, 2008.

-
- [36] Pietari Pulkkinen and Hannu Koivisto. A dynamically constrained multiobjective genetic fuzzy system for regression problems. *IEEE, Transaction on Fuzzy Systems*, 18(1):161–177, 2010.
- [37] Chen Ji-lin and Hou Yuan-long. A multi-objective genetic-based method for design fuzzy classification systems. *IJCSNS, International Journal of Computer Science and Networks Security*, 6(8):110–118, 2006.
- [38] Marcos E. Cinta and Heloisa A. Camargo. Using fuzzy formal concepts in the genetic generation of fuzzy systems. *IEEE, World Congress on Computational Intelligence*, 12:1–8, 2012.
- [39] Shiyong Li Baojiang Zhao. Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems. *Elsevier, Fuzzy Sets and Systems*, (118):235–255, 2001.
- [40] Patricia Melin Oscar Castillo Miguel A. Portal Oscar Montiel, Roberto Sepúlveda and Iliana M. Meza. Performance of a simple tuned fuzzy controller and a pid controller on a dc motor. *IEEE, Foundations of Computational Intelligence (FOCI 2007)*, 7:531–537, 2007.
- [41] H.A. Ortiz-de-la-Vega O. Castillo P. Melin J.C. Cortes-Rios, E. Gomez-Ramirez. Optimal design of interval type 2 fuzzy controllers based on a simple tuning algorithm. *Elsevier, Applied Soft Computing*, 14(23):270–285, 2014.
- [42] M. Tokhi A. Almeshal and K. Goher. Robust hybrid fuzzy logic control of a novel two-wheeled robotic vehicle with a movable payload under various operating conditions. *IEEE, International Conference on Control*, 12:747–752, 2012.
- [43] A. M. Almeshal O. K. Sayidmarie, M. O. Tokhi and S. A. Agouri. Design and real-time implementation of a fuzzy logic control system for a two-wheeled robot. *IEEE, World Congress on Computational Intelligenc*, 12:569–572, 2012.
- [44] M. O. Tokhi A. M. Almeshal, K. M. Goher and S. A. Agouri. A new configuration of a two-wheeled double inverted pendulum-like robotic vehicle with movable payload on an inclined plane. *Innovative Engineering Systems (ICIES), 2012 First International Conference on*, 12:97–102, 2012.