



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PROYECTO DE INVESTIGACIÓN

DISEÑO DE UNA RED NEURO-DIFUSA PARA LA NAVEGACIÓN DE UN ROBOT MÓVIL

PREVIO A LA OBTENCIÓN DEL TÍTULO:

INGENIERO EN MECATRÓNICA

PRESENTADO POR:

21611096

ELI SAMUEL SÁNCHEZ PORTILLO

ASESOR: ING. JOSÉ LUIS ORDOÑEZ ÁVILA

CAMPUS: SAN PEDRO SULA; ABRIL, 2021

DEDICATORIA

El siguiente trabajo va dedicado a todo aquel que formo parte de mi aprendizaje de forma directa o indirecta, los compañeros que estudiaban conmigo, los instructores de laboratorio que siempre estaban dispuestos a ayudar, los catedráticos que me formaron, mis padres y amistades que siempre me apoyaron. Gracias! Un pequeño logro mío es 100% reflejo de todo lo que me han dado.

AGRADECIMIENTO

En primer lugar, agradezco a Dios. Creo que el factor común de mi éxito y buenas obras ha sido siempre la presencia de un mando superior que rige leyes espirituales, por lo tanto, mi esfuerzo siempre está acompañado por mi fe.

Agradezco a mi familia por su espera y paciencia. Me he mantenido muy al margen de ellos en periodos tediosos en el ámbito laboral y académico, sin embargo, su atención hacia mí siempre ha estado presente.

En el ámbito académico agradezco los catedráticos en general. Me siento orgulloso del alto talento humano que cuenta nuestra universidad. Su nivel profesional y calidad humana han sido intachables hacia mí, algunas veces apoyándome una milla más de lo que su compromiso les limita.

Gracias a mis padres por tantísimo sacrificio que han hecho por mí a través de los años y poder dar la mejor educación que ellos pudieron.

EPÍGRAFE

En los tiempos de crisis solo la imaginación es más importante que el conocimiento.

~Albert Einstein

RESUMEN EJECUTIVO

La navegación autónoma de un robot móvil en un entorno desconocido con obstáculos muy abarrotados es un tema fundamental en la investigación de robótica móvil. Durante los últimos años, los robots o vehículos móviles autónomos se han aplicado cada vez más en diversas áreas como conducción automática, exploración, vigilancia, seguimiento del crecimiento de un incendio forestal, exploración de terrenos peligrosos u hostiles, recogida de datos topográficos en un entorno desconocido, etc. El diseño y desarrollo de robots móviles autónomos atrae más atención en la era de la navegación autónoma. Existen varios algoritmos que se utilizan en la práctica para resolver problemas de investigación relacionados con el modelo de robot y su entorno operativo. Por esta razón, en la presente investigación se diseñó un sistema de inferencia difusa basado en red adaptativa para el control de navegación de robots móviles. A través de una metodología incremental de tres incrementos que consistieron en: el diseño del sistema sensorial, diseño del sistema de control y el entrenamiento del ANFIS. Se diseñó un sistema sensorial de 4 sensores, 3 de distancia infrarrojos y un giroscopio que permite tener una mejor percepción del entorno. Cada uno representa una entrada al sistema de control. El sistema cuenta con dos salidas PWM para el control de cada uno de los motores del robot móvil. Tanto la FM como el tipo de salida influyen en el modelo a entrenar. Para el presente diseño, se determinó que los mejores resultados fueron obtenidos por el modelo ANFIS Gaussiana/Constante para el ANFIS-1 y para el ANFIS-2 el modelo gaussiana/lineal presentó los mejores resultados.

Palabras Clave: Robot móvil, evasión de obstáculos, ANFIS, navegación, aprendizaje automático

ABSTRACT

Autonomous navigation of a mobile robot in an unfamiliar environment with very crowded obstacles is a fundamental topic in mobile robotics research. During recent years, robots or autonomous mobile vehicles have been increasingly applied in various areas such as automatic driving, exploration, surveillance, monitoring the growth of a forest fire, exploring dangerous or hostile terrain, collecting topographic data in an environment unknown, etc. The design and development of autonomous mobile robots attracts more attention in the age of autonomous navigation. There are several algorithms that are used in practice to solve research problems related to the robot model and its operating environment. For this reason, in the present investigation a fuzzy inference system based on an adaptive network was designed for the navigation control of mobile robots. Through an incremental methodology of three increments that consist of: the design of the sensory system, design of the control system and ANFIS training. A sensory system was designed with 4 sensors, 3 infrared distance sensors and a gyroscope that allows a better perception of the environment. Each represents an input to the control system. The system has two PWM outputs to control each of the motors of the mobile robot. Both the FM and the type of output influence the model to be trained. For the present design, it was determined that the best results were obtained by the ANFIS Gaussian / Constant model for the ANFIS-1 and for the ANFIS-2 the Gaussian / linear model presented the best results.

Keywords: Mobile robot, obstacle avoidance, ANFIS, navigation, machine learning

ÍNDICE DE CONTENIDO

CAPÍTULO I. INTRODUCCIÓN	1
CAPÍTULO II. PLANTEAMIENTO DEL PROBLEMA	2
2.1 PRECEDENTES DEL PROBLEMA	2
2.2 DEFINICIÓN DEL PROBLEMA.....	3
2.3 JUSTIFICACIÓN	3
2.4 PREGUNTAS DE INVESTIGACIÓN	4
2.5 OBJETIVOS	4
2.5.1 OBJETIVO GENERAL.....	4
2.5.2 OBJETIVOS ESPECÍFICOS	4
CAPÍTULO III. MARCO TEÓRICO	5
3.1 ANÁLISIS DE LA SITUACIÓN ACTUAL.....	5
3.2 APRENDIZAJE AUTOMÁTICO	8
3.2.1 DESCENSO DEL GRADIENTE.....	10
3.2.2 BACKPROPAGATION	13
3.2.3 ESTIMACIÓN DE MÍNIMOS CUADRADOS.....	14
3.2.4 USO DE ML PARA LA NAVEGACIÓN DE ROBOTS MÓVILES.....	15
3.2 LÓGICA DIFUSA.....	17
3.2.1 SISTEMA DE INFERENCIA DIFUSO (FIS).....	19
3.2.1.1 TIPOS DE FIS	22
3.2.2 USO DE FL PARA LA NAVEGACIÓN DE ROBOTS MÓVILES	23
3.3 SISTEMA NEURO DIFUSO	26
3.3.1 ARQUITECTURA ANFIS.....	28

3.3.1.1 CAPA 1: FUSIFICACIÓN	29
3.3.1.2 CAPA 2: REGLAS	30
3.3.1.3 CAPA 3: NORMALIZACIÓN	30
3.3.1.4 CAPA 4: DEFUSIFICACIÓN.....	30
3.3.1.5 CAPA 5: SALIDA FINAL.....	31
3.3.2 USO DE ANFIS PARA LA NAVEGACIÓN DE ROBOTS MÓVILES.....	31
CAPÍTULO IV. METODOLOGÍA	33
4.1 ENFOQUE.....	33
4.2 VARIABLES DE INVESTIGACIÓN.....	33
4.2.1 VARIABLE DEPENDIENTE	34
4.2.2 VARIABLES INDEPENDIENTES.....	35
4.2.2.1 DISTANCIA.....	35
4.2.2.2 ÁNGULO	35
4.2.2.3 PWM.....	35
4.2.2.4 RMSE	35
4.3 TÉCNICAS E INSTRUMENTOS APLICADOS	36
4.4 METODOLOGÍA DE ESTUDIO	36
4.4.1 INCREMENTO I: DISEÑO DEL SISTEMA SENSORIAL.....	37
4.4.2 INCREMENTO II: DISEÑO DEL SISTEMA DE CONTROL.....	37
4.4.2 INCREMENTO III: ENTRENAMIENTO DEL ANFIS.....	37
4.5 CRONOGRAMA DE ACTIVIDADES.....	37
CAPÍTULO V. RESULTADOS Y ANÁLISIS	39
5.1 DISEÑO DEL SISTEMA SENSORIAL	39

5.2 DISEÑO DE ANFIS	45
5.3 ENTRENAMIENTO DE ANFIS-1	52
5.3.1 ANFIS-1 TRIANGULAR/CONSTANTE.....	54
5.3.2 ANFIS-1 TRIANGULAR/LINEAL	59
5.3.3 ANFIS-1 GAUSSIANA/CONSTANTE	62
5.3.4 ANFIS-1 GAUSSIANA/LINEAL.....	66
5.4 ENTRENAMIENTO DE ANFIS-2	69
5.4.1 ANFIS-2 TRIANGULAR/CONSTANTE.....	70
5.4.2 ANFIS-2 TRIANGULAR/LINEAL	73
5.4.3 ANFIS-2 GAUSSIANA/CONSTANTE	76
5.4.4 ANFIS-1 GAUSSIANA/LINEAL.....	79
5.5 DISCUSIÓN	82
CAPÍTULO VI. CONCLUSIONES	87
6.1 CONCLUSIÓN GENERAL.....	87
6.2 CONCLUSIONES ESPECÍFICAS	87
CAPÍTULO VII. RECOMENDACIONES.....	88
BIBLIOGRAFÍA.....	89

ÍNDICE DE ILUSTRACIONES

Ilustración 1-Mercado de robots móviles por región	6
Ilustración 2-Clasificación de aprendizaje automático	9
Ilustración 3-Descenso del gradiente.....	11
Ilustración 4-Sistema de inferencia difusa.....	20
Ilustración 5-Tipos de FIS.....	23
Ilustración 6-Arquitectura ANFIS	29
Ilustración 7-Variables de investigación.....	34
Ilustración 8-Metodlogía incremental.....	36
Ilustración 9-Sensor IR SHARP	40
Ilustración 10-MPU-6050	41
Ilustración 11-Distribución de los sensores del robot móvil.....	42
Ilustración 12-Diseño del controlador ANFIS.....	45
Ilustración 13-Arquitectura del controlador ANFIS.....	47
Ilustración 14-Ejemplo de comportamientos reactivos.....	50
Ilustración 15-Distribución de datos de entrenamiento para ANFIS-1	54
Ilustración 16-Funciones de membresía triangular antes del entrenamiento	55
Ilustración 17-RMSE de ANFIS-1 triangular/constante.....	56
Ilustración 18-Comparación con datos de entrenamiento de ANFIS-1 triangular/constante.....	57
Ilustración 19-Comparación con datos de prueba de ANFIS-1 triangular/constante	57
Ilustración 20-FMs triangular despues del entrenamiento ANFIS con salida constante	58
Ilustración 21-RMSE de ANFIS-1 triangular/lineal.....	59
Ilustración 22-Comparación con datos de entrenamiento de ANFIS-1 triangular/lineal.....	60

Ilustración 23-Comparación con datos de prueba de ANFIS-1 triangular/lineal	61
Ilustración 24-FMs tipo triangular despues del entrenamiento ANFIS con salida lineal	62
Ilustración 25-Funciones de membresía gaussiana antes del entrenamiento.....	63
Ilustración 26-RMSE de ANFIS-1 gaussiana/constante	63
Ilustración 27-Comparación con datos de entrenamiento de ANFIS-1 gaussiana/constante	64
Ilustración 28-Comparación con datos de prueba de ANFIS-1 gaussiana/constante.....	64
Ilustración 29-FMs gaussiana después del entrenamiento con salida constante	65
Ilustración 30-RMSE de ANFIS-1 gaussiana/lineal	66
Ilustración 31-Comparación con datos de entrenamiento de ANFIS-1 gaussiana/lineal	67
Ilustración 32-Comparación con datos de prueba de ANFIS-1 gaussiana/lineal.....	68
Ilustración 33-FM gaussiana después del entrenamiento con salida lineal	68
Ilustración 34-Distribución de datos de entrenamiento para ANFIS-2	70
Ilustración 35-RMSE de ANFIS-2 triangular/constante.....	71
Ilustración 36-Comparación con datos de entrenamiento de ANFIS-2 triangular/constante.....	72
Ilustración 37-Comparación con datos de prueba de ANFIS-2 triangular/constante	72
Ilustración 38-FMs triangular después del entrenamiento con salida constante.....	73
Ilustración 39-RMSE de ANFIS-2 triangular/lineal.....	74
Ilustración 40-Comparación con datos de entrenamiento de ANFIS-2 triangular/lineal.....	75
Ilustración 41-Comparación con datos de prueba de ANFIS-2 triangular/lineal	75
Ilustración 42-FMs triangular después del entrenamiento con salida lineal.....	76
Ilustración 43-RMSE de ANFIS-2 gaussiana/constante	77
Ilustración 44-Comparación con datos de entrenamiento de ANFIS-2 gaussiana/constante	78
Ilustración 45-Comparación con datos de prueba de ANFIS-2 gaussiana/constante.....	78

Ilustración 46-FMs gaussiana después del entrenamiento con salida constante	79
Ilustración 47-RMSE de ANFIS-2 gaussiana/lineal	80
Ilustración 48-Comparación con datos de entrenamiento de ANFIS-2 gaussiana/lineal	81
Ilustración 49-Comparación con datos de prueba de ANFIS-2 gaussiana/lineal	81
Ilustración 50-FMs gaussiana después del entrenamiento con salida lineal	82
Ilustración 51-Comportamiento del sistema de navegación final.....	85

ÍNDICE DE TABLAS

Tabla 1-Cronograma de Actividades	38
Tabla 2-Especificaciones de los sensores.....	40
Tabla 3-Sistema sensorial de diversos controladores.....	43
Tabla 4-Variables lingüísticas	46
Tabla 5-Cantidad de parámetros por ANFIS	49
Tabla 6-Datos de entrenamiento.....	51
Tabla 7-Especificación de la arquitectura ANFIS.....	52
Tabla 8-Resumen de los resultados para cada modelo de ANFIS-1	83
Tabla 9-Resumen de los resultados para cada modelo de ANFIS-2	84

ÍNDICE DE ECUACIONES

Ecuación 1-LSE	14
Ecuación 2-Conjunto difuso.....	18
Ecuación 3-Intersercción o mínimo.....	18
Ecuación 4-Unión o máximo.....	18
Ecuación 5-Función de membresía triangular	19
Ecuación 6-Función de membresía trapezoidal	19
Ecuación 7-Función de membresía gaussiana	19
Ecuación 8-Fusificación.....	21
Ecuación 9-Centroide	21
Ecuación 10- Peso promedio.....	22
Ecuación 11-Capa de fusificación de ANFIS	29
Ecuación 12-Capa de reglas de ANFIS.....	30
Ecuación 13-Capa de normalización de ANFIS.....	30
Ecuación 14-Capa de defusificación de ANFIS	30
Ecuación 15-Salida final de ANFIS	31
Ecuación 16-Regla Si-Entonces con salida lineal	48
Ecuación 17-Regla Si-Entonces con salida constante.....	48
Ecuación 18-Peso promedio con salida lineal.....	49
Ecuación 19-Peso promedio con salida constante.....	49

LISTA DE SIGLAS

ANFIS	<i>Adaptive Neuro Fuzzy Inference System</i> (Sistema de Inferencia Adaptativo Neuro Difuso)
APF	<i>Artificial Potential Field</i> (Campo de Potencial Artificial)
CD	<i>Cell Decomposition</i> (Descomposición Celular)
FIS	<i>Fuzzy Inference System</i> (Sistema de Inferencia Difusa)
FL	<i>Fuzzy Logic</i> (Lógica Difusa)
FM	Función de Membresía
GA	<i>Genetic Algorithm</i> (Algoritmos Genéticos)
GAPCNN	<i>Guided Autowave Pulse Coupled Neural Network</i> (Red Neuronal Guiada de Onda Automática Acoplada por Pulsos)
MLP	<i>Multi Layer Perceptron</i> (Red de Perceptrón Multicapa)
NEFCLASS	<i>Neuro Fuzzy Classification</i> (Clasificación Neuro Difusa)
NEFCON	<i>Neural Fuzzy Controllers</i> (Controladores Neuro Difusos)
NEFPROX	<i>Neuro Fuzzy Function Approximation</i> (Función de Aproximación Neuro Difusa)
PSO	<i>Particle Swarm Optimization</i> (Optimización de Enjambre de partículas)
RA	<i>Roadmap Approach</i> (Enfoque de la Hoja De Ruta)
RNA	Red Neuronal Artificial
RNN	<i>Recurrent Neural Network</i> (Red Neuronal Recurrente)

CAPÍTULO I. INTRODUCCIÓN

El diseño y desarrollo de robots móviles autónomos atrae más atención en la era de la navegación autónoma. Existen varios algoritmos que se utilizan en la práctica para resolver problemas de investigación relacionados con el modelo de robot y su entorno operativo. Por esta razón, en la presente investigación se diseñará un sistema de inferencia difusa basado en red adaptativa para el control de navegación de robots móviles. El desarrollo de la presente investigación está seccionado en diversos capítulos.

Capítulo 2. Planteamiento del problema. Se presenta una descripción precisa sobre el problema que se busca solucionar a través de la presente propuesta de proyecto. Se detallan los precedentes del problema, así como la situación actual.

Capítulo 3. Marco teórico. Para el desarrollo de la presente investigación es indispensable establecer diversos fundamentos teóricos que permitan su realización, esto se presenta en este capítulo. Aquí se describen diversos resultados obtenidos por otros investigadores sobre el área de interés, las piezas desarrolladas a través de manufactura aditiva y su comportamiento dentro del agua.

Capítulo 4. Metodología. Con la finalidad del desarrollo adecuado de la investigación es necesario el seguimiento de ciertos lineamientos, los cuales se presentan en la metodología. La metodología utilizada es la metodología espiral.

Capítulo 5. Resultados y análisis. En el presente capítulo se documentan todos los resultados obtenidos de las diversas pruebas realizadas a las piezas fabricadas a través de la manufactura aditiva en el agua. De igual manera, se describe el respectivo análisis de los resultados obtenidos, entre ellos un análisis microscópico.

Capítulo 6. Conclusiones. En base a los resultados obtenidos se da respuesta a las preguntas de investigación formuladas en el capítulo 2.

Capítulo 7. Recomendaciones. A partir de las conclusiones obtenidas se presentan diversas recomendaciones a futuros autores en la temática de investigación.

CAPÍTULO II. PLANTEAMIENTO DEL PROBLEMA

En este capítulo se mostrarán los precedentes del problema, definiendo el problema y estableciendo los parámetros importantes que lo afectan directamente. Luego se verá en la justificación la importancia de realizar dicho estudio con argumentos válidos. Se definirán las preguntas de investigación que orientan las respuestas que se esperan obtener mediante el análisis y desarrollo de esta investigación. Por último, los objetivos que son clave para determinar el alcance y el éxito de la investigación.

2.1 PRECEDENTES DEL PROBLEMA

Durante los últimos años, los robots o vehículos móviles autónomos se han aplicado cada vez más en diversas áreas como conducción automática, exploración, vigilancia, seguimiento del crecimiento de un incendio forestal, exploración de terrenos peligrosos u hostiles, recogida de datos topográficos en un entorno desconocido, etc. Para lograr estas aplicaciones, el robot debe tener la capacidad de navegar de forma autónoma en un entorno desconocido, donde una variedad de obstáculos pueden poner en peligro la seguridad del objetivo. Por lo tanto, el vehículo autónomo se basa en un algoritmo de navegación inteligente, que consiste en la planificación e implementación del movimiento libre de colisiones en un entorno desconocido. Los métodos de planificación de movimiento basados en sensores utilizan planificación de ruta global o local, dependiendo del entorno circundante.

El objetivo principal de los sistemas de navegación es guiar al robot móvil desde el origen hasta el destino con la ayuda de sensores. Los sensores tales como sensores ultrasónicos, sensores infrarrojos, sensores basados en láser, radar, cámara, etc. Se utilizan para la detección de obstáculos y aplicaciones de alcance en robótica. Un sensor inteligente, generalmente utilizado en los sistemas avanzados, comprende un elemento controlador y circuitos de acondicionamiento de señales para la conversión efectiva de datos brutos en información significativa. El rango limitado, el campo de visión, etc., limita el funcionamiento de los sensores individuales como un módulo independiente.

A lo largo de las últimas décadas, los investigadores han desarrollado nuevas medidas, algoritmos, estudios, y metodologías para la optimización de los sistemas de control y navegación de los robots móviles. Tzafestas (2018) desarrolló una revisión global sobre ello. Presentó una visión global de las metodologías de navegación y control de robots móviles desarrolladas durante las últimas décadas. Detallaron una lista de libros sobre robots móviles autónomos y una descripción general de los artículos académicos que cubren una amplia gama de áreas de decisión, control y navegación. Similarmente, Patle *et al.* (2019) presentaron una revisión sobre estrategias de planificación de rutas para la navegación de robots móviles. Analizaron la navegación en condiciones estáticas y dinámicas (para sistemas de robot único y múltiple) y se ha observado que los enfoques reactivos (GA, RNA, FL, PSO, entre otros) son más robustos y funcionan bien en todo terreno en comparación con los enfoques clásicos (CD, RA, y APF). También se observa que los enfoques reactivos se utilizan para mejorar el rendimiento de los enfoques clásicos como un algoritmo híbrido.

2.2 DEFINICIÓN DEL PROBLEMA

La necesidad de robots móviles autónomos aumenta día a día para reemplazar a los humanos de lugares donde el factor de riesgo es más para la vida humana y para trabajar en armonía en entornos dinámicos mientras cooperan con los humanos. Para actuar de forma independiente, se debe diseñar un comportamiento autónomo para el robot, haciendo uso de diversas técnicas clásicas y reactivas. El robot móvil enfrentará un entorno estático y dinámico mientras opera en tiempo real, es decir, los obstáculos pueden estar estacionarios o en movimiento o ambos. El robot móvil debe ser capaz de detectar el camino más corto posible para llegar al destino evitando obstáculos tanto estacionarios como móviles.

2.3 JUSTIFICACIÓN

La aplicación de un robot móvil se limitaba únicamente a las industrias manufactureras. Pero hoy en día, se usa comúnmente en los campos del entretenimiento, la medicina, la minería, el rescate, la educación, el ejército, el espacio, la agricultura y muchos más. Mientras realiza la tarea de navegación, el robot está equipado con muchos equipos inteligentes que son necesarios para modelar el entorno y localizar su posición, controlar el movimiento, detectar

obstáculos y evitar obstáculos mediante el uso de técnicas de navegación. La planificación segura de la ruta (detectando y evitando los obstáculos) desde la posición inicial hasta la posición objetivo es la función más importante de cualquier técnica de navegación. Por lo tanto, la selección adecuada de la técnica de navegación es el paso más importante en la planificación de la trayectoria de un robot cuando se trabaja en un entorno simple y complejo.

2.4 PREGUNTAS DE INVESTIGACIÓN

- 1) ¿Cuáles serán las entradas y salidas del sistema de navegación del robot móvil?
- 2) ¿Qué parámetros son los requeridos para el entrenamiento y desarrollo del sistema de inferencia neuro-difuso?
- 3) ¿Cuál será el comportamiento de navegación del modelo desarrollado?

2.5 OBJETIVOS

Un objetivo de investigación es el fin o meta que se pretende alcanzar en un proyecto, estudio o trabajo de investigación. También indica el propósito por el que se realiza una investigación. (Flores, 2017)

2.5.1 OBJETIVO GENERAL

Diseñar un sistema de inferencia difusa basado en red adaptativa para el control de navegación de robots móviles.

2.5.2 OBJETIVOS ESPECÍFICOS

- 1) Determinar las entradas y salidas del sistema de navegación del robot móvil.
- 2) Establecer los parámetros del sistema de inferencia difusa basado en red adaptativa y de su entrenamiento.
- 3) Evaluar el comportamiento de navegación del modelo desarrollado.

CAPÍTULO III. MARCO TEÓRICO

En el capítulo anterior se definió la introducción, antecedentes, planteamiento y justificación del problema de investigación, por lo cual en el presente apartado se plantea la situación actual sobre la temática de la investigación, así como teoría la cual sustenta la investigación.

3.1 ANÁLISIS DE LA SITUACIÓN ACTUAL

La robótica móvil es la rama de la ciencia y la tecnología, en la que los desarrolladores crean robots móviles que pueden moverse en cualquier entorno sin necesidad de intervención humana. Se utilizan dispositivos como sensores, software y otros engranajes para controlar estos robots. La tecnología robótica, como los vehículos terrestres no tripulados, los vehículos aéreos no tripulados y los vehículos submarinos autónomos, se tienen en cuenta al estudiar el mercado de la robótica móvil a través de varias verticales de la industria.

La tecnología robótica está experimentando una mayor adopción en varios entornos comerciales e industriales. Por ejemplo, los hospitales ahora están utilizando robots móviles autónomos para mover materiales y rastrear la salud de los pacientes. El sector militar y de defensa utiliza robots de servicio profesional que se despliegan para escenarios de combate. Además, la robótica móvil mejora las capacidades del soldado al dar a las tropas una ventaja a nivel del suelo. Además, el robot de campo, un robot de servicio profesional, se utiliza en este sector para una variedad de funciones, como transportar equipo pesado, operar en situaciones de peligro para la vida para proteger a los soldados y rescatar a los soldados heridos en zonas de combate.

Con los avances en la tecnología robótica y una mayor flexibilidad en el uso de la robótica, los robots cubren una amplia gama de aplicaciones en la industria de la salud. Los robots se utilizan en hospitales, clínicas, empresas farmacéuticas y por fabricantes de dispositivos médicos. La aplicación de la robótica móvil en la industria de la salud incluye el transporte de medicamentos, la entrega de muestras, el pick & place y la atención al paciente. Por ejemplo, SwissLog, una empresa alemana que diseña, desarrolla y ofrece soluciones de automatización para la industria

de la salud, ofrece productos como TransCar AGV, robots de servicio autónomo de retransmisión y mensajería Robo, para mudanzas de trabajo pesado y atención de pacientes.

Se espera que factores como el aumento de la necesidad de seguridad para la vida humana y el aumento de la demanda de robótica móvil de los minoristas en línea impulsen el crecimiento del mercado de la robótica móvil en el futuro. Sin embargo, el alto costo de adquisición inicial y los desafíos que enfrentan los robots móviles mientras trabajan en entornos no probados actúan como barreras importantes y obstaculizan el crecimiento global de la industria de la robótica móvil. Por el contrario, se espera que factores como el surgimiento de la industria 4.0 en la logística y el almacenamiento y el aumento en la adopción de la robótica móvil en el sector agrícola ofrezcan oportunidades lucrativas para el mercado a nivel mundial.

Se espera que el mercado de robots móviles crezca de USD 18,7 mil millones en 2018 a USD 54,1 mil millones en 2023, a una tasa compuesta anual de 23,71% durante el período de pronóstico (Markets and Markets, 2018), ver ilustración 1. Los robots móviles han sido testigos de un salto cuántico con la introducción de tecnologías como la realidad aumentada, el aprendizaje profundo y la inteligencia artificial (IA). Con la creciente población geriátrica la demanda de tecnologías avanzadas y asistentes virtuales, como robots de compañía, también está aumentando a medida que estas tecnologías simplifican el proceso de realización de una tarea. Sin embargo, el rendimiento en entornos no probados y la falta de una interfaz de alto nivel actúan como factores de restricción clave para el crecimiento del mercado.

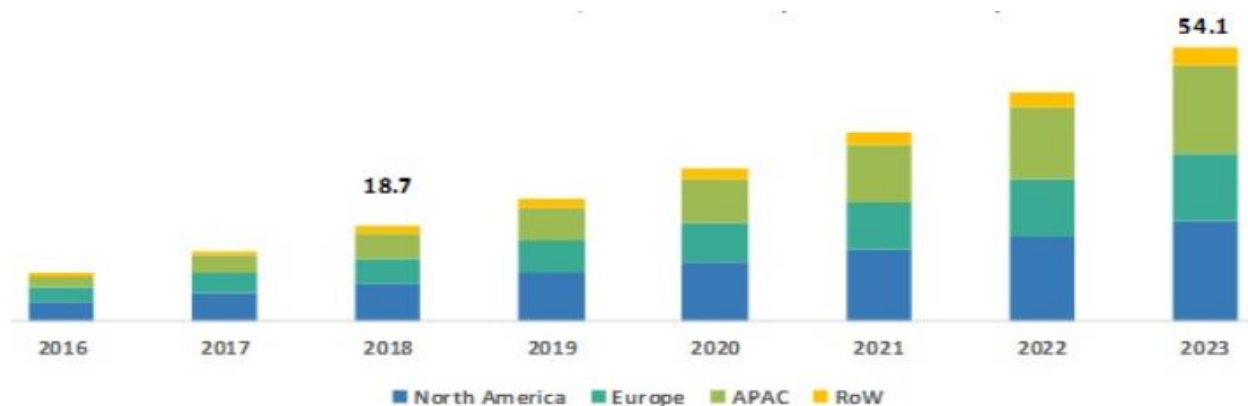


Ilustración 1-Mercado de robots móviles por región

Fuente: (Markets and Markets, 2018)

El mercado global de robótica móvil está segmentado según el producto, el componente, la aplicación y la región. Según el producto, el mercado se clasifica en vehículos terrestres no tripulados (UGV), vehículos aéreos no tripulados (UAV) y vehículos submarinos autónomos (AUV). Por componente, se divide en hardware, software y soporte y servicios. Las aplicaciones cubiertas en el estudio incluyen logística y almacén, militar y defensa, salud, hogar, entretenimiento, educación, agricultura y silvicultura, entre otras. Según la región, se analiza en América del Norte, Europa, Asia-Pacífico y LAMEA junto con sus países destacados.

El software utilizado en un sistema robótico es básicamente un conjunto de comandos codificados y una lista de instrucciones que se proponen para programar robots de acuerdo con la tarea que deben realizar. Debido a los avances tecnológicos, los OEM de robots han comenzado a integrar plataformas de análisis de datos como el análisis de big data y el análisis predictivo con el sistema robótico. El rápido crecimiento del mercado de software también se puede atribuir al aumento de la demanda de robots móviles con navegación automática y capacidad de toma de decisiones autónoma.

La adopción de robots profesionales para aplicaciones médicas, como la automatización de laboratorios y la asistencia quirúrgica, es uno de los factores clave que impulsa el mercado. Sin embargo, la automatización del almacén es una tendencia reciente observada en la industria y los AGV desempeñan un papel muy importante en el proceso de automatización del almacén. Aparte de esto, los robots ayudan a las empresas de logística a lograr un período de amortización de corto a mediano plazo y proporcionan un mayor retorno de la inversión (ROI).

El desarrollo de la inteligencia artificial y la IoT ha dado una nueva dimensión a cómo los robots pueden interactuar y trabajar con los humanos. Además, el desarrollo y la evolución de MEMS y sensores, y las mejoras en la tecnología de visualización han mejorado la eficiencia y precisión de los robots móviles. El crecimiento del mercado de robots móviles para aplicaciones domésticas está liderado por la innovación y la modernización en el campo de la robótica. Los avances tecnológicos y las actividades de investigación han llevado al desarrollo de robots de limpieza domésticos, que están ayudando a los humanos a realizar tareas mundanas.

El controlador o sistema de control es el dispositivo principal utilizado en cualquier tipo de sistema robótico. Todo robot debe tener un sistema de control para operar su sistema de accionamiento, que se utiliza para mover el brazo, la muñeca y el cuerpo de un robot y, por lo tanto, también se lo conoce como la unidad central de procesamiento (CPU) o procesador o el cerebro de los robots. . Además, con la integración de tecnologías como la inteligencia artificial (IA) y el IoT, se planteó la necesidad de sistemas de control optimizados para robots. Por lo tanto, el crecimiento del mercado de sistemas de control también se puede atribuir a la evolución de los robots autónomos y los robots integrados con IA en todo el mundo.

3.2 APRENDIZAJE AUTOMÁTICO

Aprendizaje automático o más conocido como *machine learning* (ML) es un subconjunto de la inteligencia artificial, que construye un modelo matemático basado en datos de muestra, conocido como "datos de entrenamiento", con el fin de hacer predicciones o decisiones sin estar programado explícitamente para realizar la tarea (Zhang, 2020). El aprendizaje automático se puede definir en términos generales como métodos computacionales que utilizan la experiencia para mejorar el rendimiento o realizar predicciones precisas. Aquí, la experiencia se refiere a la información pasada disponible para el alumno, que generalmente toma la forma de datos electrónicos recopilados y el modo disponible para el análisis. Estos datos podrían estar en forma de conjuntos de entrenamiento digitalizados con etiquetas humanas u otros tipos de información obtenida a través de la interacción con el medio ambiente. En todos los casos, su calidad y tamaño son cruciales para el éxito de las predicciones realizadas por el alumno. Con respecto al aprendizaje, una buena definición dada por Mitchell (1997) es:

Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y la medida de desempeño P si su desempeño en las tareas en T , medido por P , mejora con experiencia E .

Un ejemplo de un problema de aprendizaje es cómo utilizar una muestra finita de documentos seleccionados al azar, cada uno etiquetado con un tema, para predecir con precisión el tema de los documentos invisibles. Claramente, cuanto más grande es la muestra, más fácil es la tarea. Pero la dificultad de la tarea también depende de la calidad de las etiquetas asignadas a los documentos de la muestra, ya que las etiquetas pueden no ser todas correctas, y de la cantidad

de temas posibles. El aprendizaje automático consiste en diseñar algoritmos de predicción eficientes y precisos (Mohri *et al.*, 2018). Como en otras áreas de la informática, algunas medidas críticas de la calidad de estos algoritmos son su complejidad temporal y espacial. Pero, en el aprendizaje automático, necesitaremos adicionalmente una noción de complejidad de la muestra para evaluar el tamaño de muestra requerido para que el algoritmo aprenda una familia de conceptos. De forma más general, las garantías de aprendizaje teórico de un algoritmo dependen de la complejidad de las clases de conceptos consideradas y del tamaño de la muestra de formación. El aprendizaje automático tiene como objetivo establecer un regresor o clasificador a través del aprendizaje del conjunto de entrenamiento y luego evaluar el rendimiento del regresor o clasificador a través del conjunto de prueba (Hutter *et al.*, 2019). Según la naturaleza de los datos de entrenamiento, se clasifica el aprendizaje automático de la siguiente manera, como se detalla en la ilustración 2.

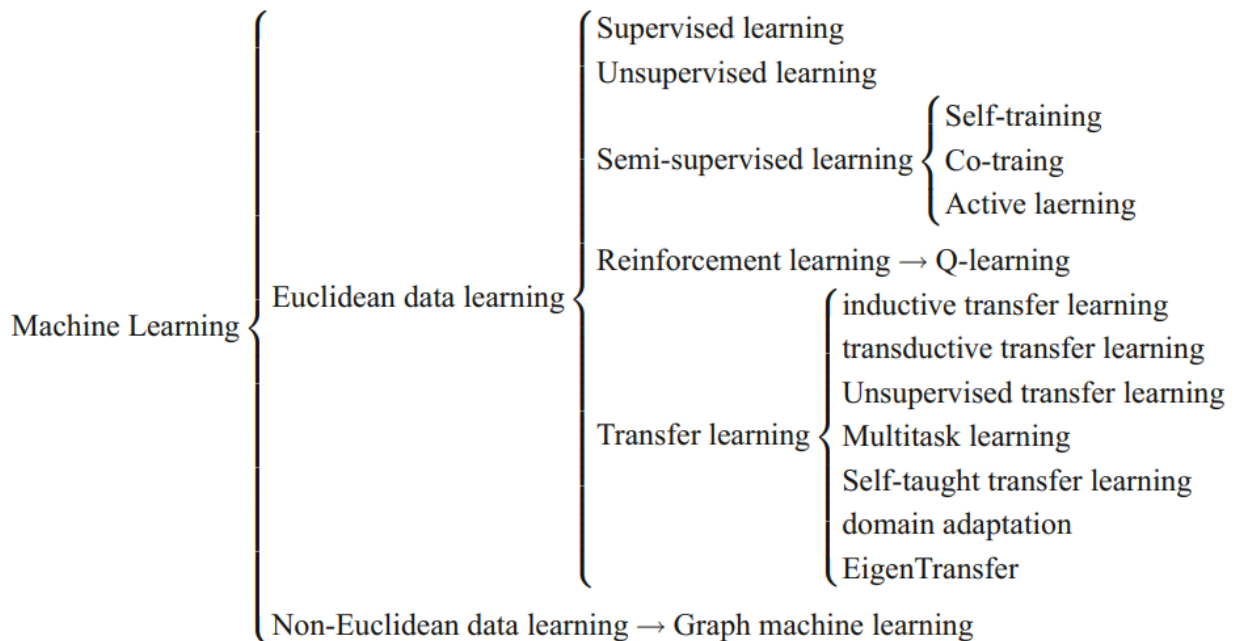


Ilustración 2-Clasificación de aprendizaje automático

Fuente: (Zhang, 2020)

Uno de los tipos más utilizados del aprendizaje automático es el aprendizaje supervisado. Este se basa un conjunto de entrenamiento que consta de datos etiquetados (es decir, entradas de

ejemplo y sus salidas deseadas), el aprendizaje supervisado aprende una regla general que asigna entradas a salidas. Esto es como un “maestro” o un supervisor (experto en etiquetado de datos) que le da a un estudiante un problema (encontrar la relación de mapeo entre entradas y salidas) y sus soluciones (datos de salida etiquetados) y le dice al estudiante que averigüe cómo resolver otros problemas similares: encontrar el mapeo de las características de muestras invisibles a sus etiquetas correctas o valores objetivo en el futuro (Carleo *et al.*, 2019). El objetivo de todos los algoritmos de aprendizaje automático supervisados es estimar mejor una función objetivo (f) que mapea los datos de entrada (X) en las variables de salida (Y). Esto describe todos los problemas de clasificación y regresión. Existen diferentes métodos o algoritmos para el proceso de aprendizaje. Uno de los métodos mayormente usados para el entrenamiento es el descenso del gradiente y *least squares estimation* (LSE).

3.2.1 DESCENSO DEL GRADIENTE

El descenso de gradiente es un algoritmo de optimización que se utiliza para encontrar los valores de los parámetros (coeficientes) de una función (f) que minimiza una función de costo (costo). El descenso de gradiente se utiliza mejor cuando los parámetros no se pueden calcular analíticamente (por ejemplo, utilizando álgebra lineal) y deben buscarse mediante un algoritmo de optimización. Este algoritmo fue propuesto inicialmente, por Werbos (1974), quién estableció que el descenso de gradiente se utiliza para minimizar alguna función moviéndose iterativamente en la dirección del descenso más pronunciado según lo definido por el negativo del gradiente. En el aprendizaje automático, usamos el descenso de gradientes para actualizar los parámetros de nuestro modelo. Los parámetros se refieren a coeficientes en regresión lineal y pesos en redes neuronales.

El procedimiento comienza con valores iniciales para el coeficiente o coeficientes de la función. Estos pueden ser cero o un valor aleatorio pequeño. El costo de los coeficientes se evalúa conectándolos a la función y calculando el costo. Se calcula la derivada del costo. La derivada es un concepto del cálculo y se refiere a la pendiente de la función en un punto dado. Necesitamos conocer la pendiente para que sepamos la dirección (signo) para mover los valores de los coeficientes a fin de obtener un costo menor en la siguiente iteración. Ahora que se sabe por la

derivada qué dirección es cuesta abajo, se puede actualizar los valores de los coeficientes. (Truong & Nguyen, 2020)

Se debe especificar un parámetro de tasa de aprendizaje que controle cuánto pueden cambiar los coeficientes en cada actualización o pasos. El tamaño de estos pasos se denomina tasa de aprendizaje (Zhou, 2018). Con una alta tasa de aprendizaje, se puede cubrir más terreno en cada paso, pero corremos el riesgo de sobrepasar el punto más bajo ya que la pendiente de la colina cambia constantemente. Con una tasa de aprendizaje muy baja, se puede mover con confianza en la dirección del gradiente negativo, ya que se recalcula con mucha frecuencia. Una tasa de aprendizaje baja es más precisa, pero calcular el gradiente lleva mucho tiempo, por lo que llevará mucho tiempo llegar al fondo o mínimo. En la ilustración 3, se muestra a gráficamente el procedimiento que realiza el algoritmo con una tasa de aprendizaje adecuada hasta alcanzar el mínimo de la función.

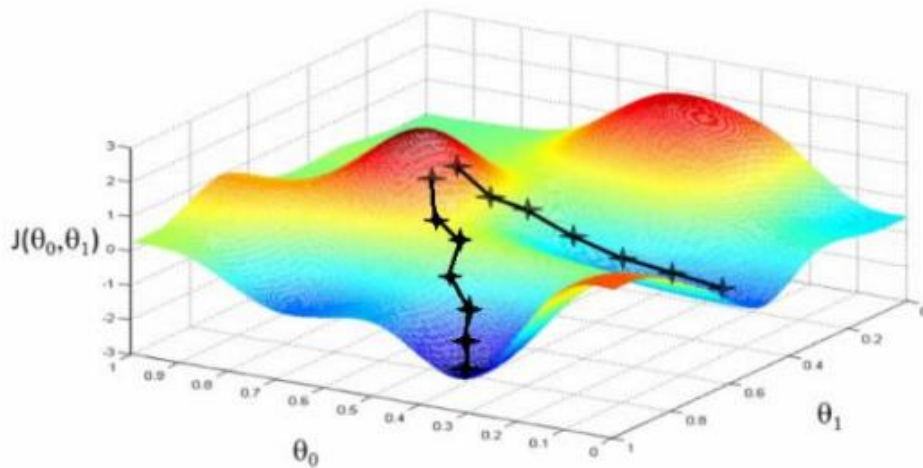


Ilustración 3-Descenso del gradiente

Fuente: (Zhou, 2018)

La evaluación de qué tan cerca se ajusta un modelo de aprendizaje automático es a través de la función de costo, se puede calcular de diferentes formas, a menudo específicas del algoritmo de aprendizaje automático. La función de costo implica evaluar los coeficientes en el modelo de aprendizaje automático calculando una predicción para el modelo para cada instancia de

entrenamiento en el conjunto de datos y comparando las predicciones con los valores de salida reales y calculando una suma o error promedio (como la suma de los residuos al cuadrado o SSR en el caso de regresión lineal) (Lee et al., 2016). Una función de costo nos dice "qué tan bueno" es nuestro modelo para hacer predicciones para un conjunto dado de parámetros. La función de costo tiene su propia curva y sus propios gradientes. La pendiente de esta curva nos dice cómo actualizar nuestros parámetros para hacer que el modelo sea más preciso.

A partir de la función de costo, se puede calcular una derivada para cada coeficiente de modo que se pueda actualizar utilizando exactamente la ecuación de actualización descrita anteriormente. El costo se calcula para un algoritmo de aprendizaje automático en todo el conjunto de datos de entrenamiento para cada iteración del algoritmo de descenso de gradiente (Lee *et al.*, 2016). Este proceso se repite hasta que el costo de los coeficientes (costo) es 0 o lo suficientemente cerca de cero para ser lo suficientemente bueno. Requiere que conozca el gradiente de su función de costo o la función que está optimizando, pero además de eso, es muy sencillo. Hutter *et al.* (2019) mencionan algunos parámetros para tomar en cuenta para aplicar el descenso del gradiente:

1) Trazar el costo en función del tiempo

Recopilar y trazar los valores de costo calculados por el algoritmo en cada iteración. La expectativa de una ejecución de descenso de gradiente con buen rendimiento es una disminución en el costo de cada iteración. Si no disminuye, se debe reducir su tasa de aprendizaje.

2) Tasa de aprendizaje:

El valor de la tasa de aprendizaje es un valor real pequeño, como 0,1, 0,001 o 0,0001. Se debe probar diferentes valores para el problema en cuestión y determinar cuál funciona mejor.

3) Reescalar entradas:

El algoritmo alcanzará el costo mínimo más rápido si la forma de la función de costo no está sesgada y distorsionada. Se puede lograr esto reescalando todas las variables de entrada (X) al mismo rango, como [0, 1] o [-1, 1].

4) Pocos pases

El descenso de gradiente estocástico a menudo no necesita más de 1 a 10 pases a través del conjunto de datos de entrenamiento para converger en coeficientes buenos o suficientemente buenos.

5) Trazar el costo medio

Las actualizaciones para cada instancia del conjunto de datos de entrenamiento pueden resultar en un gráfico ruidoso del costo a lo largo del tiempo cuando se usa el descenso de gradiente estocástico. Se puede tomar el promedio de más de 10, 100 o 1000 actualizaciones para obtener una mejor idea de la tendencia de aprendizaje del algoritmo.

3.2.2 BACKPROPAGATION

El objetivo de *backpropagation* es el de ajustar cada peso en la red en proporción a cuánto contribuye al error general. Si se reduce iterativamente el error de cada peso, eventualmente tendremos una serie de pesos que producirán mejores predicciones. Este algoritmo fue propuesto por Rumelhart *et al.* (1986), establece que la retropropagación es una aplicación de la regla de la cadena para encontrar las derivadas de costo con respecto a cualquier variable en la ecuación anidada. Se utiliza la regla de la cadena para calcular la derivada del costo con respecto a cualquier peso en la red. La regla de la cadena nos ayudará a identificar cuánto contribuye cada peso a nuestro error general y la dirección para actualizar cada peso para reducir nuestro error.

La retropropagación calcula el gradiente de la función de pérdida con respecto a los pesos de la red para un solo ejemplo de entrada-salida, y lo hace de manera eficiente, a diferencia de un cálculo directo ingenuo del gradiente con respecto a cada peso individualmente (Lee et al., 2016). Esta eficiencia hace factible el uso de métodos de gradiente para entrenar redes multicapa, actualizando pesos para minimizar las pérdidas; descenso de gradiente, o variantes como el descenso de gradiente estocástico, se utilizan comúnmente. El algoritmo de retropropagación funciona calculando el gradiente de la función de pérdida con respecto a cada peso mediante la regla de la cadena, calculando el gradiente una capa a la vez, iterando hacia

atrás desde la última capa para evitar cálculos redundantes de términos intermedios en la regla de la cadena; este es un ejemplo de programación dinámica. La retropropagación generaliza el cálculo del gradiente en la regla delta, que es la versión de una sola capa de la retropropagación, y a su vez se generaliza por diferenciación automática, donde la retropropagación es un caso especial de acumulación inversa (o “modo inverso”). (Zhou, 2018)

3.2.3 ESTIMACIÓN DE MÍNIMOS CUADRADOS

El método de “mínimos cuadrados” es una forma de análisis de regresión matemática que se utiliza para determinar la línea de mejor ajuste para un conjunto de datos, proporcionando una demostración visual de la relación entre los puntos de datos. Cada punto de los datos representa la relación entre una variable independiente conocida y una variable dependiente desconocida. El método de mínimos cuadrados consiste en estimar parámetros minimizando las discrepancias cuadráticas entre los datos observados, por un lado, y sus valores esperados, por el otro. (Banks & Joyner, 2017)

El método de mínimos cuadrados proporciona la justificación general para la ubicación de la línea de mejor ajuste entre los puntos de datos que se estudian. La aplicación más común de este método, que a veces se denomina “lineal” u “ordinario”, tiene como objetivo crear una línea recta que minimice la suma de los cuadrados de los errores que se generan por los resultados de las ecuaciones asociadas, como como los residuos cuadrados que resultan de las diferencias en el valor observado, y el valor anticipado, con base en ese modelo (Cohen & Migliorati, 2017). El criterio de mínimos cuadrados es una medida de ajuste computacionalmente conveniente. Corresponde a la estimación de máxima verosimilitud cuando el ruido se distribuye normalmente con varianzas iguales. Para escribir el estimador de mínimos cuadrados para el modelo de regresión lineal, será conveniente utilizar la notación matricial. El estimador de mínimos cuadrados viene dado por la ecuación 1, donde X son los datos de entrada en forma matricial, y Y la salida.

$$\beta = (X^T X)^{-1} X^T Y$$

Ecuación 1-LSE

3.2.4 USO DE ML PARA LA NAVEGACIÓN DE ROBOTS MÓVILES

La sección de control de los robots móviles es una de las más importantes, ya que es el que acciona la movilización o navegación del robot de una manera adecuada. Por esta razón, muchos investigadores se han dado la tarea de desarrollar controladores con diversos algoritmos para mejorar el rendimiento de la navegación del robot móvil. Uno de los algoritmos o sistemas utilizados son las redes neuronales, en conjunto con algoritmos de aprendizaje automático como ser *backpropagation*. Las características de las redes neuronales como la capacidad de generalización, el paralelismo masivo, la representación distribuida, la capacidad de aprendizaje y la tolerancia a fallas lo hacen útil en el campo de la navegación de robots móviles.

M. K. Singh & Parhi (2011) presentaron un enfoque novedoso para el diseño de un controlador inteligente para un robot móvil autónomo que utiliza una red neuronal de alimentación directa (*feed forward*) multicapa, que permite al robot navegar en un entorno dinámico del mundo real. Las entradas al controlador neuronal propuesto consisten en la distancia del obstáculo izquierdo, derecho y frontal con respecto a su posición y ángulo objetivo. La salida de la red neuronal es el ángulo de dirección. Diseñaron una red neuronal de cuatro capas para resolver el problema de optimización de ruta y tiempo de los robots móviles, que se ocupa de las tareas cognitivas como el aprendizaje, la adaptación, la generalización y la optimización. Utilizaron un algoritmo de retropropagación para el entrenamiento de la red.

Similarmente, N. H. Singh & Thongam (2019) presentaron un método para generar una ruta y una velocidad adecuada y sin colisiones para un robot móvil en un entorno dinámico que contiene obstáculos en movimiento y estáticos utilizando una red neuronal artificial. La red neuronal de perceptrón multicapa (MLP) fue utilizada para seleccionar un segmento libre de colisiones de un conjunto de cinco segmentos, y la red también controla la velocidad del robot para cada movimiento del robot. Para cada movimiento, el MLP calcula el tiempo que tarda el robot en llegar a su destino local para cada segmento libre de colisiones utilizando el tiempo que tardan los obstáculos críticos. Finalmente, el robot elige un segmento y una ruta más cercana al destino global.

Por otro lado, para mejorar el rendimiento de la RNA, Syed *et al.* (2014) han modificado la RNA básica para formar una red neuronal guiada de onda automática acoplada por pulsos (GAPCNN) para obtener una rápida convergencia de parámetros para un robot móvil que se mueve en un entorno estático y dinámico. El enfoque se modifica aplicando control direccional de ondas automáticas y disparo acelerado de neuronas en función de la estrategia de umbral dinámico.

Medina-Santiago *et al.* (2014) han introducido un sistema de control neuronal para el robot móvil en tiempo real mediante el uso de sensores ultrasónicos. La arquitectura del RNA se implementa en capacidades de procesamiento embebidas en un sistema de tiempo real, generando casos de respuesta, que no fueron tomados en cuenta a la hora de entrenar, pasando a ser un control confiable y tolerante a fallas. El control del vehículo resultó en un robusto sistema de transmisión diferencial que se utilizó.

Khnessi *et al.* (2018) presentaron un controlador de red neuronal que controla la trayectoria y la evasión de obstáculos de un robot móvil. Desarrollaron un diseño de la red neuronal basada en controlador de velocidad para el modelo cinemático de robot móvil. El controlador neuronal propuesto tiene dos entradas (velocidad lineal de x e y) y dos salidas (comando de la rueda izquierda y derecha).

Una aplicación disruptiva de los robots móviles es su uso para el cuidado del hogar. Debido a ello, diversos investigadores, se encargan de enfrentar los desafíos de navegación autónoma para el robot de atención domiciliaria, como cambios frecuentes de entorno, obstáculos y posición de la meta. ByungSoo Ko *et al.* (2017) se enfocaron en verificar el potencial de la navegación autónoma basada en redes neuronales para dispositivos móviles de atención domiciliaria. Y compararon la red neuronal recurrente (RNN) con el perceptrón multicapa en la navegación de un robot móvil autónomo. El resultado sugirió que la red neuronal recurrente puede hacer una mejor navegación del robot debido a su capacidad para manejar la dependencia temporal de una secuencia de datos. Además, muestra que la navegación basada en redes neuronales puede ser una buena alternativa, ya que tiene una capacidad de generalización decente para nuevos entornos, obstáculos y objetivos.

3.2 LÓGICA DIFUSA

La lógica difusa o también conocida como lógica borrosa es una forma matemática de lograr una representación de la incertidumbre existente en el lenguaje natural, esta forma matemática fue concebida alrededor de los años sesentas por el ingeniero eléctrico Lofti Zadeh, quien publicó el primer artículo de lógica difusa conocido como *Fuzzy Sets* (Zadeh, 1965).

En comparación con la lógica clásica, en donde solamente se permiten dos valores ya sea cero o uno, verdadero o falso, en cambio, la lógica difusa permite todos los valores dentro de cero a uno, permitiendo procesar información con un mayor grado de incertidumbre. Gracias al desarrollo de Zadeh, se fue expandiendo esta temática en los próximos años (Zadeh, 1973). Hasta el punto donde investigadores aplicaron los conceptos de lógica difusa en el control de procesos, desarrollando de tal manera, los primeros controladores difusos para el proceso de control de alguna aplicación por Ebrahim Mamdani (Mamdani & Assilian, 1975), con esta inferencia fue diseñado el controlador difuso de la presente investigación, sin embargo, otros investigadores han desarrollado otras inferencias basadas en la lógica difusa, como ser la teoría de control difuso aportada por Takagi y Sugeno (Takagi & Sugeno, 1985), siendo una alternativa del método Mamdani.

Uno de los conceptos que se debe tener bien claro sobre la lógica difusa, son los conjuntos difusos, en estos conjuntos se asocian los valores o función de pertenencia, donde se busca que un elemento forme parte de un conjunto con un determinado grado de pertenencia (Guzmán & Castaño M., 2006). Estos conjuntos difusos son los que representan las entradas y salidas que posee un controlador difuso. Por lo tanto, se deben tener en cuenta los diversos parámetros y operaciones de los conjuntos difusos.

Uno de los primeros parámetros de los conjuntos difusos es el universo de discurso, el cual se refiere a la totalidad de elementos que se están clasificando, el universo de discurso se representa con: X . Otros de los parámetros importantes del conjunto difuso es la función de membresía, esta es la medida de la pertenencia de un elemento al conjunto, el cual va a tener un dominio, que será el universo de discurso y la imagen, es un número real entre 0 y 1, donde 0 se refiere a que no hay pertenencia al conjunto y 1 hay una total de pertenencia, la función de

membresía se describe con la notación $\mu(x)$ (Zadeh, 1965). Dada una colección de objetos U, un conjunto difuso A en U se define como un conjunto de pares ordenados como se presente en la ecuación 2.

$$A = \{(x, \mu_A(x)) \mid x \in U\}$$

Ecuación 2-Conjunto difuso

Aparte de estos parámetros se deben tener en cuenta algunas operaciones que se pueden realizar con los conjuntos difusos, una de estas operaciones es la intersección (Zadeh, 1965), la cual también se puede realizar entre conjuntos clásicos, sin embargo, para los conjuntos difusos se hace uso de la ecuación 3.

$$\mu_C(x) = \mu_A(x) \wedge \mu_B(x) \text{ para } \forall x \in X$$

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x))$$

Ecuación 3-Intersercción o mínimo

La ecuación 3, es utilizada dentro del motor de inferencia en el controlador difuso, además de esta operación, en el controlador difuso también se utilizó la operación de unión de los conjuntos difusos (Zadeh, 1965), la cual también es aplicable en conjuntos clásicos, esta operación se representa con la ecuación 4.

$$\mu_C(x) = \mu_A(x) \vee \mu_B(x) \text{ para } \forall x \in X$$

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x))$$

Ecuación 4-Unión o máximo

Existen muchas más operaciones que se pueden realizar con los conjuntos difuso, donde la mayoría provienen de la teoría clásica y son aplicados en la lógica difusa, sin embargo, generalmente, estas son las operaciones utilizados dentro de un sistema de inferencia difuso. Aparte de esto, cada con junto difuso es representado por una función de membresía, las cuales sueles poseer formas geométricas, como ser una función triangular y una trapezoidal, campana, gaussiana, entre otras (Kahraman *et al.*, 2016). Sin embargo, aparte de estas, existen otras funciones de membresía con diferentes características y representación matemática, con la ecuación 5, se obtiene la función de membresía de una función triangular.

$$f(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x \geq c \end{cases}$$

Ecuación 5-Función de membresía triangular

Donde $a \leq b \leq c$, para una función trapezoidal se utilizó la ecuación 6, en donde $a \leq b \leq c \leq d$.

$$f(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x \geq d \end{cases}$$

Ecuación 6-Función de membresía trapezoidal

Otro tipo de función de membresía muy utilizada es la función de membresía campana, que su valor de membresía se obtiene a través de la ecuación 7.

$$f(x; a, b) = e^{\left(\frac{x-a}{b}\right)^2}$$

Ecuación 7-Función de membresía gaussiana

Donde a, b y c son parámetros del conjunto difuso.

3.2.1 SISTEMA DE INFERENCIA DIFUSO (FIS)

FIS se basa en reglas difusas de si-entonces. Las reglas difusas si-entonces o los enunciados condicionales difusos son expresiones de la forma SI A ENTONCES B, donde A y B son etiquetas de conjuntos difusos caracterizados por funciones de pertenencia o de membresía apropiadas. Debido a su forma concisa, las reglas difusas si-entonces se emplean a menudo para capturar los modos imprecisos de razonamiento que juegan un papel esencial en la capacidad humana para tomar decisiones en un entorno de incertidumbre e imprecisión. (Simić *et al.*, 2017)

Mediante el uso de etiquetas lingüísticas y funciones de pertenencia, una regla difusa si-entonces puede capturar fácilmente el espíritu de una "regla empírica" utilizada por los

humanos. Desde otro ángulo, debido a los calificadores en las partes de la premisa, cada regla difusa si-entonces puede verse como una descripción local del sistema en consideración. Las reglas difusas si-entonces forman una parte central del sistema de inferencia difusa, entre otros componentes que se detallan en la ilustración 4.

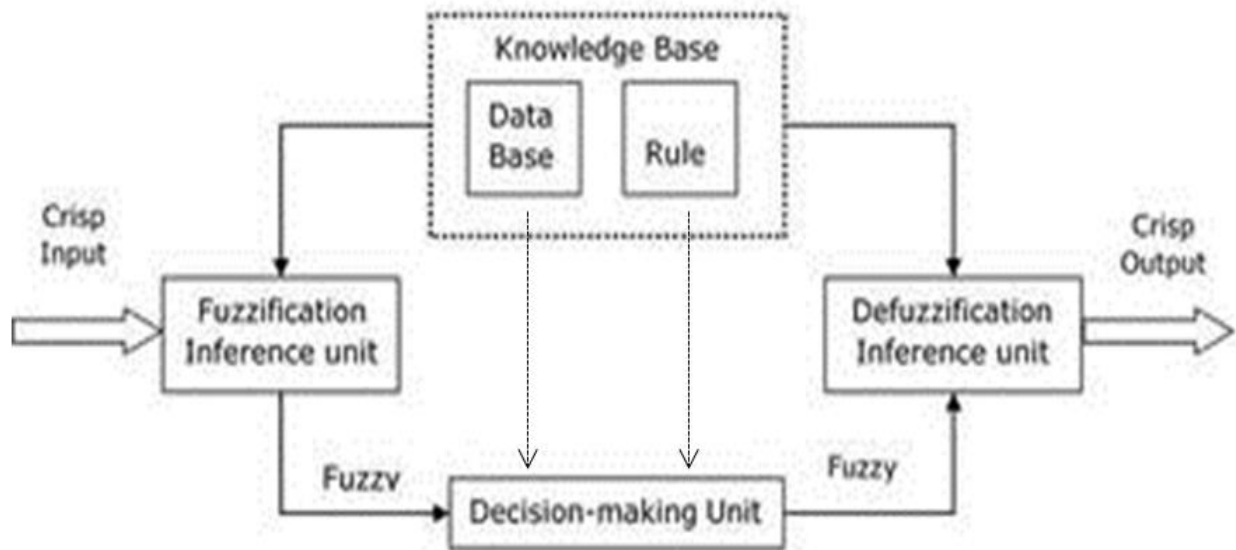


Ilustración 4-Sistema de inferencia difusa

Fuente: (Simić *et al.*, 2017)

Tal como se muestra en la ilustración 4, se inicia con el pre-procesamiento o entrada, para luego entrar al sistema de inferencia difusa que se encarga de determinar la salida, por último, se ejecuta el post-procesamiento, Simić *et al.* (2017) menciona la función de cada uno de los componentes del FIS.

1) Pre-Procesamiento (Entrada)

En la primera etapa, se acondiciona las señales antes de que estas sean introducidas en el controlador difuso, por lo general, se realizan dos funciones antes de llegar al controlador, las cuales son filtrar o amplificar la señal y muestrear la señal.

2) Fusificación

Luego, la señal entra al controlador difuso, iniciando con la fusificación de esta señal, donde se convierte la magnitud de la señal en una cantidad difusa, obteniendo el valor de pertenencia que tiene en cada uno de los valores lingüísticos. Donde se reduce a la ecuación 8.

$$\forall x [\mu_{A'}(x) \wedge \mu_{A_i}(x)] = \mu_{A_i}(x_0)$$

Ecuación 8-Fusificación

Donde A' es un conjunto singleton dentro del conjunto difuso que lo establece x_0 , el cual es el valor de entrada al FIS y A_i es la función de membresía en donde está siendo evaluada. Por lo tanto, se dice que la fusificación es la evaluación de la entrada en el conjunto difuso, variando solamente cuando el valor de entrada cambia.

3) Inferencia

La inferencia incluye la base de conocimiento del sistema de la toma de decisión. Dentro de la base de conocimiento se incluyen las reglas y la base de datos. Las reglas de control son el conjunto de reglas lingüísticas si-entonces. La base de datos define la función de membresía de los conjuntos difusos en las reglas difusas. Y la unidad de la toma de decisión, que se encarga de realizar la operación de inferencia sobre las reglas.

4) Defusificación

La última etapa del FIS es la defusificación, del paso anterior se obtiene un valor difuso resultante, por lo tanto, esto necesita ser defusificado para entregar un único valor a la salida del sistema. En la defusificación se convierte el valor difuso resultante de la inferencia en una cantidad certera para generar la señal de control. El método de defusificación varía según el método de inferencia. Para el método de inferencia Mamadani, se utiliza generalmente, el centroide del área del conjunto resultante, que se obtiene con la ecuación 9.

$$y_0 = \frac{\sum y\mu(y)}{\sum \mu(y)}$$

Ecuación 9-Centroide

Donde y_0 representa la salida en valor crisp, y $\mu(y)$ representa el valor de membresía de y . Otro de los métodos más utilizados es el peso promedio que se obtiene a partir de la ecuación 10.

$$y = \frac{w_1 y_1 + w_2 y_2 + \dots + w_n y_n}{w_1 + w_2 + \dots + w_n}$$

Ecuación 10- Peso promedio

5) Post-Procesamiento (Salida)

En la última etapa, se genera la señal de control a partir de la cantidad defusificada, donde se acondiciona la señal para el actuador.

3.2.1.1 Tipos de FIS

Kahraman *et al.* (2016) menciona que los sistemas de inferencia difusa se pueden clasificar en tres tipos, se describen a continuación, y se representan gráficamente en la ilustración 5.

1) Tsukamoto

La salida general es el promedio ponderado de la salida nítida de cada regla inducida por la fuerza de activación de la regla (el producto o mínimo de los grados de coincidencia con la parte de la premisa) y las funciones de pertenencia de la salida. Las funciones de pertenencia de salida utilizadas en este esquema deben ser funciones monótonas.

2) Mamdani

La salida difusa general se obtiene aplicando la operación máximo a las salidas difusas calificadas (cada una de las cuales es igual al mínimo de fuerza de disparo y la función de pertenencia de salida de cada regla). Se han propuesto varios esquemas para elegir la salida nítida final en función de la salida difusa general; algunos de ellos son centroide de área, ver ecuación 9, bisectriz de área, media de máximos, criterio de máximo, etc.

3) Takagi-Sugeno-Kang

La salida de cada regla es una combinación lineal de variables de entrada más un término constante, y la salida final es el promedio ponderado de la salida de cada regla, obtenido a través de la ecuación 10.

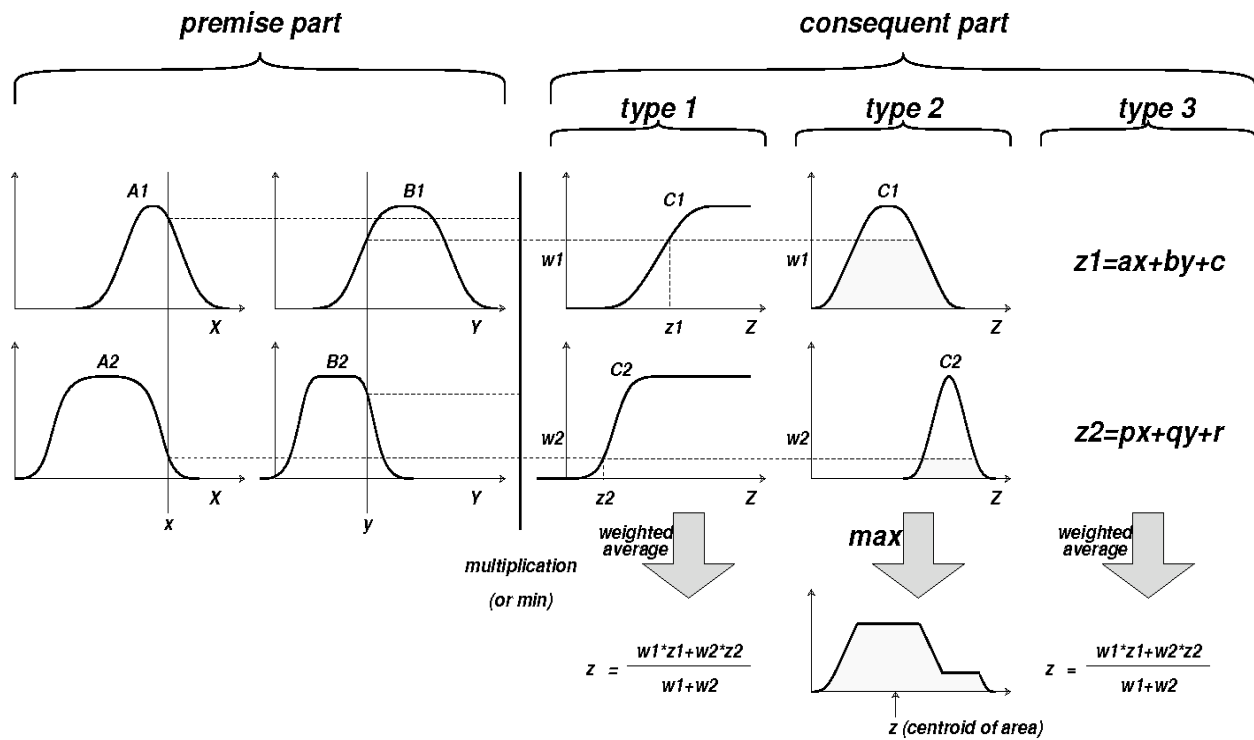


Ilustración 5-Tipos de FIS

Fuente: (Jang, 1993)

3.2.2 USO DE FL PARA LA NAVEGACIÓN DE ROBOTS MÓVILES

Desde que el concepto de FL fue propuesto por primera vez, se utilizó en muchos campos de la investigación y el desarrollo. Siendo muy apropiado para situaciones en las que existe un alto grado de incertidumbre, complejidad y no linealidad. Reconocimiento de patrones, control automático, toma de decisiones, clasificación de datos son algunos de ellos. Gracias a las reglas difusas proporcionadas por humanos, se agiliza el trabajo del diseñador del sistema y la computadora para obtener información más correcta sobre el funcionamiento de los sistemas en el mundo real. Por todo ello, es muy utilizado para la navegación de robots móviles.

Jaradat *et al.* (2012) presentó una planificación dinámica del movimiento del robot móvil autónomo utilizando un campo de potencial difuso híbrido. El campo potencial está basado en datos difusos para la planificación autónoma del movimiento de robots móviles con entornos dinámicos que incluyen objetivos y obstáculos estáticos o en movimiento. Utilizaron dos FIS, uno Mamdani y TSK para desarrollar las fuerzas totales de atracción y repulsión que actúan

sobre el robot móvil. . Las fuerzas de atracción y repulsión se estimaron utilizando cuatro entradas que representan la posición relativa y la velocidad entre el objetivo y el robot en las direcciones x e y , por un lado, y entre el obstáculo y el robot, por el otro.

Un algoritmo de lógica difusa fue desarrollado por Motlagh *et al.* (2009) para la navegación de robots móviles en entornos locales. Las entradas al controlador difuso son: las posiciones de los obstáculos en relación con el rumbo del robot y la orientación del objetivo, que se define como el ángulo entre la dirección del rumbo del robot y la dirección del robot al objetivo. El FIS se encarga de las tareas principales de evasión de obstáculos y búsqueda de objetivos, una estrategia de cambio de objetivo real-virtual resuelve el problema de los ciclos límite en cualquier tipo de callejones sin salida que se encuentren en el camino hacia el objetivo.

Para hacer frente a las incertidumbres paramétricas y no paramétricas en el modelo de robot, Abadi & Khooban (2015) desarrollaron un controlador de lógica difusa tipo Mamdani óptimo para el seguimiento de la trayectoria de los robots móviles con ruedas holonómicas. Los parámetros de las funciones de pertenencia de entrada y salida, y los coeficientes del controlador PID fueron optimizados simultáneamente mediante la optimización del enjambre de partículas de peso de inercia aleatorio.

Patle B.K. *et al.* (2016) presentaron una teoría de decisión de ruta robótica basada en un controlador difuso probabilístico. La investigación presenta la implementación eficiente de la navegación autónoma de robots móviles en el entorno complejo en presencia de obstáculos estáticos. Esto se obtiene a través de la decisión difusa a través de la probabilidad y su distribución se aplica para la función objetivo generadora sujeta a la trayectoria robótica y la evitación de obstáculos.

Los autores Abiyev *et al.* (2010) investigaron sobre la navegación de robots móviles en presencia de obstáculos. . En la navegación global, se conoce el entorno que rodea al robot y se selecciona la ruta que evita el obstáculo. Presentaron una simulación de software de problemas de navegación de un robot móvil que evita obstáculos en un entorno estático utilizando algoritmos tanto clásicos como difusos.

Aouf *et al.* (2019) desarrollaron un controlador de lógica difusa para la navegación de robots móviles de dos ruedas en entornos extraños o desconocidos. Presentaron dos comportamientos reactivos del robot móvil, a saber, "ir a la meta para evitar obstáculos" y "seguir la pared", en orden para solucionar sus problemas de navegación. Permite una planificación óptima del movimiento en términos de longitud de trayectoria y tiempo de viaje; está destinado a evitar colisiones con obstáculos convexos y cóncavos y lograr el camino más corto seguido por el robot móvil.

El desarrollo y diseño de una metodología basada en lógica difusa para controlar un robot móvil interior fue presentado por Al-mutib & Abdessemed (2017) para una navegación completa en un entorno desconocido. La metodología incorpora dos comportamientos básicos, a saber: alcanzar la meta y evitar obstáculos. El comportamiento de evasión de obstáculos se trata utilizando un esquema de seguimiento de paredes basado en una técnica de inferencia difusa de intervalo tipo 2. Esto ayuda a manejar las incertidumbres de los datos para producir un mejor rendimiento. El mecanismo de control del robot móvil utiliza algún tipo de base de conocimientos dispuesta en un conjunto de reglas difusas para implementar el comportamiento deseado que hace que el robot móvil siga el límite de un obstáculo o una pared. Se mantiene una distancia constante al obstáculo mientras el robot intenta superar esta dificultad con éxito. Una vez que el camino está despejado, el comportamiento de evasión de obstáculos se inhibe y el comportamiento de alcanzar la meta se activa mediante un controlador difuso secundario.

Rawat *et al.* (2018) presentaron un análisis e investigación de FIS Mamdani para el control y navegación de un robot móvil y exploración de diferentes técnicas de IA relacionadas con la navegación de robots. Desarrollaron un FIS Mamdani para la navegación y el control del robot durante la planificación de la ruta. Durante la investigación, la simulación y los resultados experimentales para la longitud de la ruta y el tiempo transcurrido desde el punto de origen hasta el punto de destino se han representado utilizando la metodología difusa. El controlador posee 5 entradas, estas son distancias de obstáculos desde el frente, la derecha, la parte trasera e izquierda y el ángulo de rumbo. La salida es el ángulo final calculado para la dirección del robot. Para validar la metodología desarrollada se han realizado comparaciones entre la simulación y los resultados experimentales y la desviación se encuentra dentro del 6%.

3.3 SISTEMA NEURO DIFUSO

Una red neuronal es capaz de aprender un mapeo del dominio de entrada al de salida. Un sistema difuso también implementa dicho mapeo. Entonces, si tenemos problemas para especificar los parámetros de un sistema difuso, ¿por qué no olvidarlo y simplemente usar una red neuronal? Para responder a esta pregunta, se deben considerar las situaciones típicas en las que se utilizaría un sistema difuso o una red neuronal (Jose Vieira *et al.*, 2003):

- 1) Sistema difuso: Se tiene (al menos algo) conocimiento sobre la relación entre entrada y salida, es decir, para algunas situaciones de entrada se puede especificar (vagamente) las salidas. Se puede describir este conocimiento mediante reglas difusas.
- 2) Red neuronal: Se tienen muchos datos de entrenamiento que describen la relación entrada / salida de nuestro problema, y se sabe poco o nada sobre esta relación.

Si se decide utilizar un sistema difuso y se descubre que no se puede derivar todos los parámetros a partir del conocimiento sobre el problema, entonces se puede considerar en aprenderlos. Sin embargo, si se planea utilizar técnicas de redes neuronales para hacer esto, se deben tener muchos datos de entrenamiento. Esto significa que se tiene una situación clásica para aplicar una red neuronal.

El uso de un sistema difuso tiene algunos beneficios sobre el uso de una red neuronal. Se puede interpretar un sistema difuso como un sistema de reglas lingüísticas. Esto significa que, al menos hasta cierto punto, se comprueba la plausibilidad de la solución. Una red neuronal es una caja negra para el usuario. Si se posee al menos algún conocimiento previo, se puede usarlo para inicializar un sistema difuso. Una red neuronal siempre aprende desde cero. Por tanto, tiene sentido utilizar un sistema difuso y, en lugar de ajustarlo manualmente, utilizar algún tipo de algoritmo de aprendizaje para optimizar sus parámetros (J. Vieira *et al.*, 2004). Una forma común de aplicar un algoritmo de aprendizaje a un sistema difuso es representarlo en una arquitectura especial similar a una red neuronal. Luego, se utiliza un algoritmo de aprendizaje, como, por ejemplo, retropropagación, para entrenar el sistema.

Existen diferentes enfoques de sistemas neurodifusos, sin embargo, Kruse & Nauck (1998) para enfatizar la característica común de todos estos enfoques, y para dar al término sistema neuro-

difuso un significado adecuado, menciona que estos sistemas neurodifusos poseen las siguientes propiedades:

- 1) Un sistema neuro-difuso es un sistema difuso que es entrenado por un algoritmo de aprendizaje (generalmente) derivado de la teoría de redes neuronales. El procedimiento de aprendizaje (heurístico) opera con información local y solo provoca modificaciones locales en el sistema difuso subyacente. El proceso de aprendizaje no se basa en el conocimiento, sino en los datos.
- 2) Un sistema neuro-difuso puede verse como una red neuronal de avance de 3 capas especial. Las unidades de esta red utilizan t-norms o t-conorms en lugar de las funciones de activación que se suelen utilizar en las redes neuronales. La primera capa representa las variables de entrada, la capa intermedia (oculta) representa las reglas difusas y la tercera capa representa las variables de salida. Los conjuntos difusos se codifican como pesos de conexión (difusos). Algunos modelos neuro-difusos utilizan más de 3 capas y codifican conjuntos difusos como funciones de activación. En este caso, normalmente es posible transformarlos en una arquitectura de 3 capas. Esta vista de un sistema difuso ilustra el flujo de datos dentro del sistema y su naturaleza paralela. Sin embargo, esta vista de red neuronal no es un requisito previo para aplicar un procedimiento de aprendizaje, es simplemente una conveniencia.
- 3) Un sistema neuro-difuso siempre puede (es decir, antes, durante y después del aprendizaje) ser interpretado como un sistema de reglas difusas. Es posible crear el sistema a partir de datos de entrenamiento desde cero, y es posible inicializarlo mediante conocimientos previos en forma de reglas difusas.
- 4) El procedimiento de aprendizaje de un sistema neuro-difuso tiene en cuenta las propiedades semánticas del sistema difuso subyacente. Esto da como resultado restricciones sobre las posibles modificaciones de los parámetros del sistema.
- 5) Un sistema neuro-difuso se aproxima a una función n-dimensional (desconocida) que está parcialmente dada por los datos de entrenamiento. Las reglas difusas codificadas dentro del sistema representan muestras vagas y pueden verse como prototipos vagos de los datos de

entrenamiento. Un sistema neuro-difuso no debe verse como una especie de sistema experto (difuso) y no tiene nada que ver con la lógica difusa en sentido estricto.

De igual manera, las investigaciones de los sistemas neuro-difusos han incrementado en las últimas décadas, han atraído el creciente interés de investigadores en diversas áreas científicas y de ingeniería debido a sus capacidades de aprendizaje y razonamiento efectivas. Por esta razón, los autores Shihabudheen & Pillai (2018) desarrollaron una revisión de los avances recientes en los sistemas neuro-difusos. Proponen una revisión de diferentes sistemas neuro-difusos basados en la clasificación de artículos de investigación de 2000 a 2017. Buscan ayudar a los lectores a tener una visión general del estado del arte de los sistemas neuro-difusos y fácilmente referir métodos adecuados de acuerdo con los intereses de investigación. Uno de las arquitecturas de sistemas neuro-difusos más utilizados es el ANFIS.

3.3.1 ARQUITECTURA ANFIS

Uno de los primeros sistemas neuro-difusos para la aproximación de funciones es el modelo ANFIS. Fue propuesto por Jang (1993) que representa un sistema difuso de tipo Sugeno en una arquitectura de red de alimentación directa especial. Los conjuntos difusos se modelan mediante funciones de pertenencia en forma de campana, triangular o trapezoidal, generalmente. Debido a que ANFIS usa solo funciones diferenciables, es fácil aplicar procedimientos de aprendizaje estándar de la teoría de redes neuronales.

Jang (1993) introdujo un algoritmo de aprendizaje híbrido en ANFIS para actualizar todos los parámetros (premisa y parámetros consecuentes), con el fin de minimizar el error entre la salida ANFIS y los datos de entrenamiento. El ajuste de los parámetros en ANFIS consiste en un pase hacia adelante y un pase hacia atrás. En el paso hacia adelante del algoritmo de aprendizaje híbrido, los parámetros de la premisa se fijan, las señales funcionales avanzan hasta la capa 4, y los parámetros consiguientes se identifican mediante el método de estimación de mínimos cuadrados (LSE). En el paso hacia atrás, los parámetros consiguientes son fijos y todas las tasas de error se propagan hacia atrás. Los parámetros antecedentes se actualizan mediante el método de gradiente. (Karaboga & Kaya, 2019)

El sistema neuro-difuso ANFIS se basa en las reglas difusas TSK. La arquitectura de esta sistema neuro-difuso cuenta con 5 capas como se muestra en la ilustración 6. A continuación se describe cada una de las capas. (Jang, 1993)

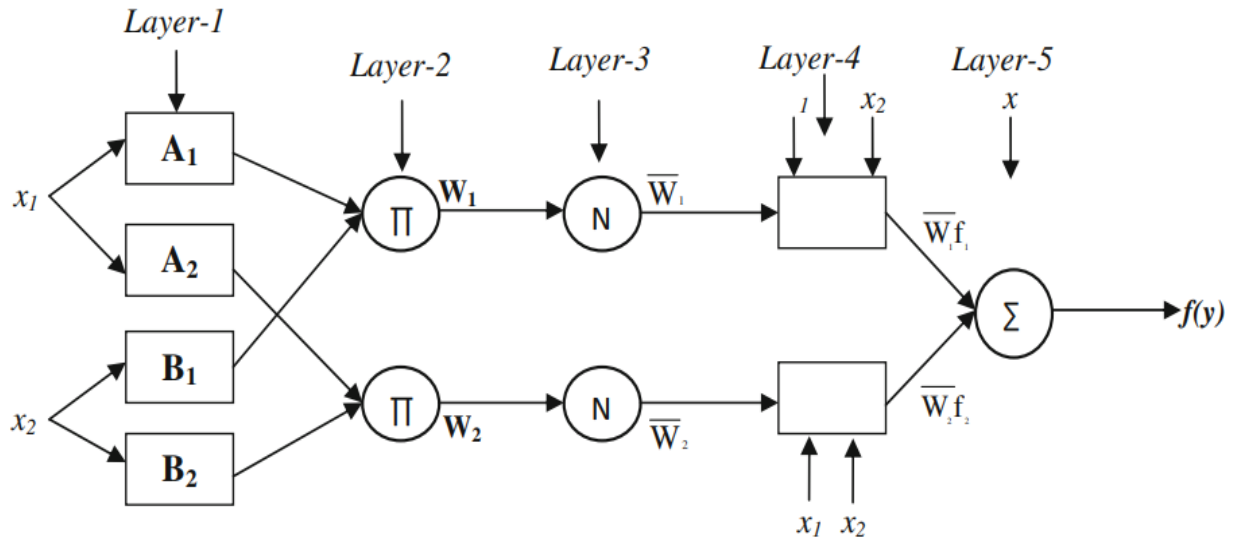


Ilustración 6-Arquitectura ANFIS

Fuente: (Jang, 1993)

3.3.1.1 Capa 1: Fusificación

También se conoce como capa de fusificación. Cada nodo de esta capa es un nodo adaptativo (nodo cuadrado) y calcula los valores de la función de pertenencia en un conjunto difuso. Las salidas de estos nodos se calculan con la ecuación 11.

$$O_{1,i} = \mu_{A_i}(x_1)$$

$$O_{1,i} = \mu_{B_i}(x_2)$$

Ecuación 11-Capa de fusificación de ANFIS

En otras palabras, $O_{1,i}$ es la función de pertenencia de A_i y B_i y especifica el grado en que x_1 y x_2 satisfacen el cuantificador A_i y B_i . La función de pertenencia difusa se puede elegir en cualquier forma, como triangular (ver ecuación 5), gaussiana, trapezoidal (ver ecuación 6), en forma de campana (ver ecuación 7), etc. Por lo general, elegimos la función en forma de campana con el valor máximo igual a 1 y el valor mínimo igual a 0.

3.3.1.2 Capa 2: Reglas

Se conoce como capa de reglas. Cada nodo de esta capa es un nodo fijo (nodo circular). Cada punto de regla recibe señales de los puntos respectivos de la capa 2 y calcula la fuerza de disparo de cada regla difusa. La salida de la capa es el producto de todas las señales entrantes (entradas), obtenida a través de la ecuación 12.

$$O_{2,i} = w_i = \mu_{A_i}(x_1) \cdot \mu_{B_i}(x_2)$$

Ecuación 12-Capa de reglas de ANFIS

Donde w_i representa fuerza de disparo de cada regla difusa.

3.3.1.3 Capa 3: Normalización

Es la capa de normalización (ver ecuación 13) y cada nodo de esta capa es un nodo fijo (nodo circular). Las salidas de esta capa se denominan fuerza de disparo normalizada. La fuerza de disparo normalizada del i -ésimo punto se calcula por la fuerza de disparo del i -ésimo nodo a la suma de la fuerza de disparo de todas las reglas.

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2$$

Ecuación 13-Capa de normalización de ANFIS

Donde \bar{w}_i representa La fuerza de disparo normalizada de una regla dada.

3.3.1.4 Capa 4: Defusificación

Es la capa de defusificación. Cada nodo de esta capa es un nodo adaptativo. Cada nodo está conectado al nodo de normalización correspondiente y también recibe las entradas iniciales x_1 y x_2 . Un nodo de defusificación determina el valor consecuente ponderado de una regla dada que se calcula con la ecuación 14.

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i), i = 1, 2$$

Ecuación 14-Capa de defusificación de ANFIS

Donde $O_{4,i}$ representa la salida de cada nodo de la capa 4, $\{p_i, q_i, r_i\}$ son los parámetros establecidos para este nodo. Estos parámetros también se denominan parámetros consecuentes.

3.3.1.5 Capa 5: Salida Final

Está representado por un único nodo de suma. Este único nodo es fijo. Este nodo calcula la salida general del sistema sumando todos los puntos de defusificación de la capa anterior. Este se calcula con la ecuación 15.

$$O_{5,i} = \sum_i \bar{w}_i f_i$$
$$O_{5,i} = \frac{\sum_i w_i f_i}{\sum_i w_i}$$
$$O_{5,i} = \frac{\sum_i w_i (p_i x_1 + q_i x_2 + r_i)}{\sum_i w_i}, \quad i = 1, 2$$

Ecuación 15-Salida final de ANFIS

Donde $O_{5,i}$ representa la salida del nodo de la capa 5.

3.3.2 USO DE ANFIS PARA LA NAVEGACIÓN DE ROBOTS MÓVILES

M. K. Singh *et al.* (2009) desarrollaron un sistema ANFIS para la navegación de robots móviles. Analizan el control de la navegación de un robot móvil utilizando un sistema de inferencia neuro-difuso adaptativo (ANFIS) en un entorno dinámico. Las entradas a la capa de lógica difusa son la distancia del obstáculo frontal, la distancia del obstáculo izquierdo, la distancia del obstáculo derecho y la dirección del objetivo. Desarrollaron un algoritmo de aprendizaje basado en la técnica de redes neuronales para ajustar los parámetros de las funciones de pertenencia difusa, que suavizan la trayectoria generada por el sistema de lógica difusa.

Para la navegación autónoma de robots móviles entre obstáculos estáticos y dinámicos Pandey Anish *et al.* (2019) desarrollaron múltiples arquitecturas ANFIS. Diseñaron e implementaron la técnica de control de sensor-actuador (motor) basada en la arquitectura del sistema de inferencia neuro-difusa adaptativa múltiple (MANFIS). Utilizaron tres sensores de rango de infrarrojos se han montado en el lado frontal, izquierdo y derecho del robot, que lee los obstáculos estáticos y dinámicos de avance, de izquierda a derecha y de derecha en el entorno.

Adarsh & Ramachandran (2018) diseñaron un algoritmo de fusión de datos de sensores para navegación de robots móviles utilizando ANFIS y su análisis en todas las funciones de

membresía. El uso combinado de redes borrosas y neuronales en ANFIS hace que el valor de la distancia medida de la covarianza residual sea consistente con su valor real. El algoritmo de fusión de datos dentro del controlador del robot móvil fusiona la entrada de los sensores ultrasónicos e infrarrojos para una mejor percepción del entorno. Los resultados obtenidos por los investigadores indican que el algoritmo de fusión de datos proporciona un error cuadrático medio mínimo (RMSE) y un error de porcentaje absoluto medio (MAPE) en comparación con el de los sensores individuales.

Para la navegación de robot móvil en entornos estáticos desconocidos los investigadores Pandey *et al.* (2016) desarrollaron un controlador ANFIS. Utilizaron diferentes sensores, como el sensor de telémetro ultrasónico y el sensor de rango infrarrojo nítido, se utilizan para detectar los obstáculos delanteros en los entornos. Las entradas del controlador ANFIS son distancias de obstáculos obtenidas de los sensores y la salida del controlador es un ángulo de dirección del robot.

CAPÍTULO IV. METODOLOGÍA

La presente investigación científica es un proyecto aplicado en la industria, referente al área de mantenimiento, específicamente, el mantenimiento predictivo o también conocido como mantenimiento basado en condición, la metodología a seguir para el desarrollo de este proyecto de investigación jugó un papel muy importante, ya que en esta, se detalló el seguimiento realizado, así como las diferentes variables que fueron analizadas a lo largo de toda la investigación, permitiendo manipularlas con el fin de obtener los mejores resultados, para esto se utilizaron diversas técnicas, instrumentos y materiales, además, se detalló este seguimiento a través de diferentes actividades que se presentaron en el cronograma de actividades, todo esto se presenta en este capítulo.

4.1 ENFOQUE

En el presente proyecto de investigación se analizaron diversas variables numéricas, encontradas en el desarrollo del modelo de control de navegación con un sistema neurodifuso, por esta razón, se estableció la presente investigación como una investigación con un enfoque cuantitativo. Hernández Sampieri, Fernández Collado, & Baptista Lucio (2010) mencionan que una investigación cuantitativa se da por aludida al ámbito del análisis numérico de las variables de investigación, se analiza una realidad objetiva con base en las mediciones numéricas y el análisis de estas, con la finalidad de determinar predicciones o patrones de comportamiento del problema planteado.

Debido a que las variables de investigación fueron manipuladas para emplear experimentos y realizar un análisis de causa-efecto para obtener la resolución del problema planteado y obtener el alcance estipulado, se determina que la presente investigación es de tipo experimental. Con esto se obtiene una investigación la cual conlleva un proceso secuencial y deductivo

4.2 VARIABLES DE INVESTIGACIÓN

Las variables de investigación juegan un papel muy importante en la investigación, ya que son manipuladas y observadas de manera experimental para posteriormente analizar su

comportamiento y determinar los efectos de la manipulación. Las variables de investigación se clasificaron en variables dependientes e independientes.

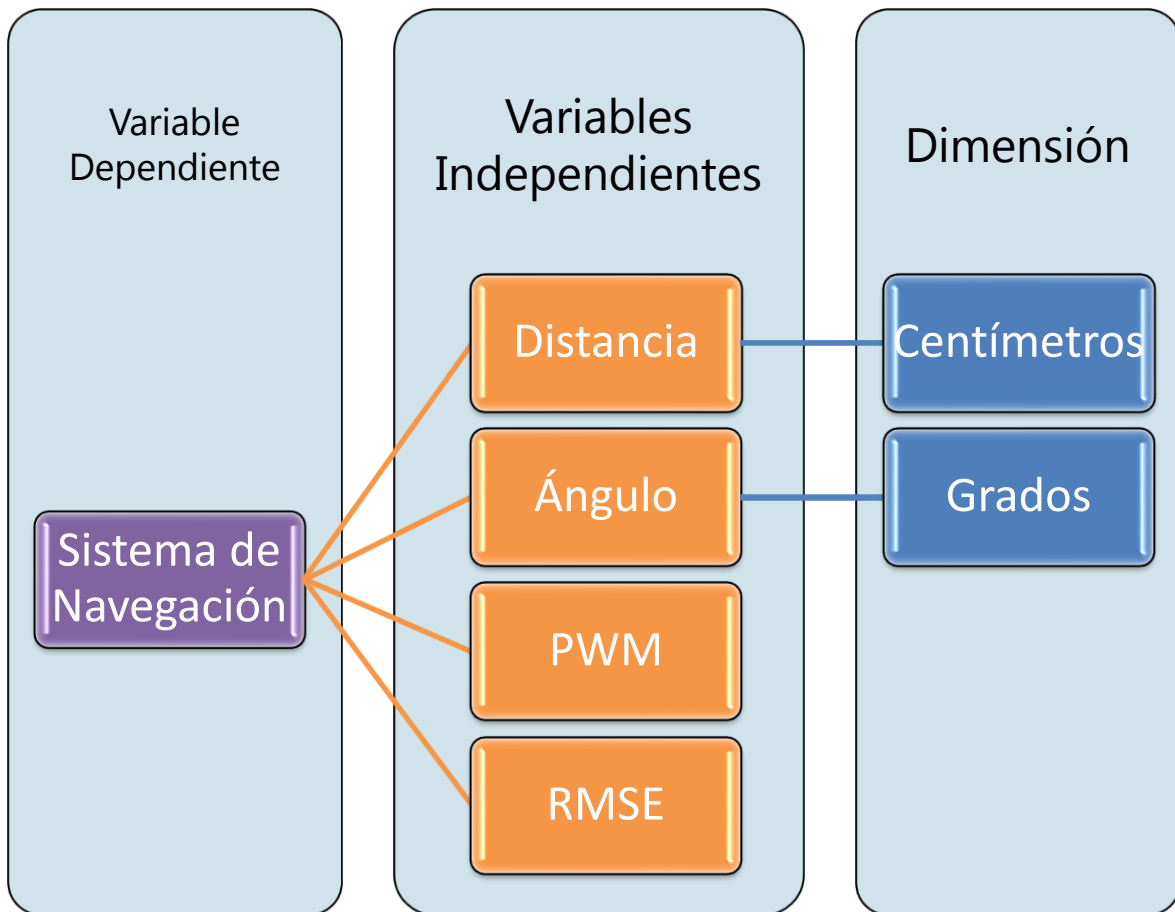


Ilustración 7-Variables de investigación

Fuente: Elaboración Propia

4.2.1 VARIABLE DEPENDIENTE

Como variable dependiente de la presente investigación se estableció el comportamiento del sistema de navegación del sistema de control. Esto se refiere a cómo se comportan cada una de las salidas del sistema de control según las variables de entrada del mismo. Cabe recalcar que el comportamiento final del sistema dependerá de igual manera del entrenamiento de la arquitectura ANFIS.

4.2.2 VARIABLES INDEPENDIENTES

Se establecieron 4 variables independientes de las cuales se dio seguimiento en la presente investigación, las cuales se discuten en las subsecciones contiguas.

4.2.2.1 Distancia

La distancia es una de las variables de mayor importancia de la presente investigación. Ya que esta se refiere a la distancia existente entre el robot y un objeto en el ambiente en el que se encuentra el robot móvil. Estos datos son retroalimentados a través de una señal analógica por medio de los sensores infrarrojos. La distancia posee unidad de centímetros. A causa de los sensores utilizados, solamente se permiten lecturas de 5 a 15 centímetros.

4.2.2.2 Ángulo

El ángulo es otra variable independiente que se considera en el sistema de navegación. Esta es retroalimentada a través de una señal obtenida del giroscopio que ayuda sobre la orientación que posee el robot en cada momento. El ángulo posee unidades de grados, que puede variar de -180 a 180 grados.

4.2.2.3 PWM

El PWM es el valor establecido en los motores que permite el ajuste de su velocidad. La unidad de esta variable es adimensional, sin embargo, posee un rango de 0 a 255 donde 0 representa el motor en reposo y 255 la velocidad máxima. Esta es una de las variables que la arquitectura ANFIS debe de controlar.

4.2.2.4 RMSE

La raíz del error cuadrático medio (RMSE) es una medida de uso frecuente de las diferencias entre los valores (valores de muestra o de población) predichos por un modelo o un estimador y los valores observados. El RMSE sirve para agregar las magnitudes de los errores en las predicciones para varias veces en una sola medida de poder predictivo. La RECM es una medida de precisión, para comparar errores de predicción de diferentes modelos para un conjunto de

datos en particular y no entre conjuntos de datos, ya que depende de la escala. Un RMSE más baja es mejor que una más alta.

4.3 TÉCNICAS E INSTRUMENTOS APLICADOS

La investigación se encuentra sustentada en diferentes fuentes confiables, que proporcionan los cimientos a la investigación y enriquecen los conceptos pertinentes, permitiendo transmitir y comunicar la información en el desarrollo de la investigación. Las fuentes confiables hacen referencia a diversos artículos de revistas académicas y libros que se centren en el tema de la investigación. Fue utilizado el software Matlab para el desarrollo y entrenamiento de cada uno de los modelos ANFIS utilizados en las pruebas de la presente investigación.

4.4 METODOLOGÍA DE ESTUDIO

Para responder al problema de investigación planteado con anterioridad, fue de suma importancia seguir una metodología de estudio, la cual describe los procedimientos, actividades con el fin obtener resultados contundentes. De esta manera, la presente investigación, fue desarrollada a través de una metodología incremental, que se presenta en la ilustración 8.

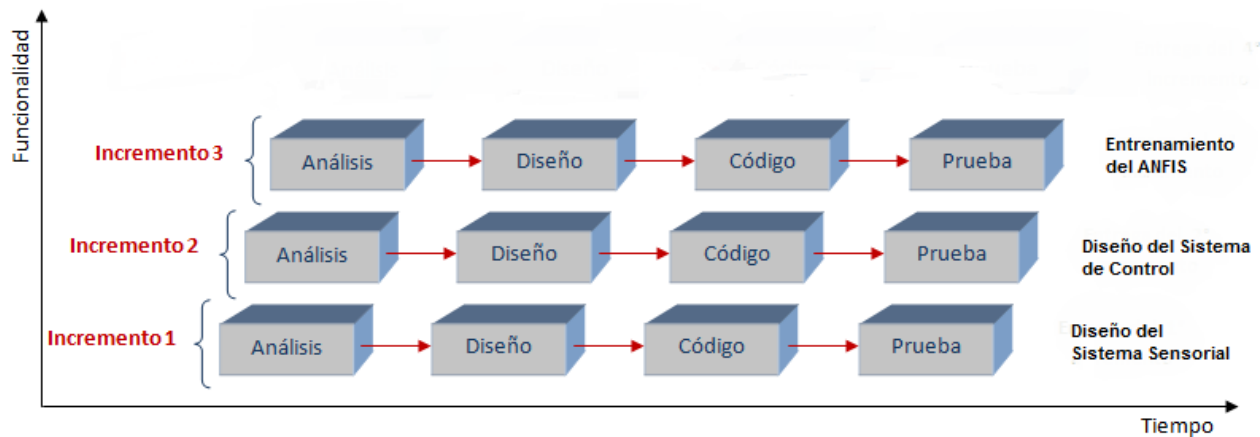


Ilustración 8-Metodología incremental

Fuente: Elaboración Propia

4.4.1 INCREMENTO I: DISEÑO DEL SISTEMA SENSORIAL

El primer incremento de la metodología de estudio consistió en el diseño del sistema sensorial que el robot móvil utilizará para la navegación. A través de este sistema sensorial, se le permitirá al robot móvil determinar la distancia entre algún objeto u obstáculo existente y el mismo robot, así como tener una mejor perspectiva del ambiente en que este se encuentra.

4.4.2 INCREMENTO II: DISEÑO DEL SISTEMA DE CONTROL

El segundo incremento consistió en el diseño del sistema de control. Aquí fueron determinadas las entradas y salidas del sistema de control. Así como los parámetros de los diferentes modelos ANFIS a investigar. Incluyendo el número de funciones de membresía que posee cada una de las entradas y las reglas difusas.

4.4.2 INCREMENTO III: ENTRENAMIENTO DEL ANFIS

El último incremento de la presente investigación consistió en el entrenamiento de cada uno de los modelos ANFIS desarrollados. Para ello fue necesario el desarrollo de una base de datos de entrenamiento. La cual fue seccionada para datos de entrenamiento y de prueba. Se obtuvo el RMSE tanto de entrenamiento como de prueba de cada uno de los modelos ANFIS y fue determinado el modelo óptimo para el sistema de navegación del robot móvil.

4.5 CRONOGRAMA DE ACTIVIDADES

El desarrollo de la presente investigación fue ejecutado a lo lapso de un tiempo determinado de manera secuencial, se realizaron diferentes actividades que permitieron la finalización adecuada de la investigación, el tiempo total fue de 10 semanas, como se presenta en la tabla 1. Se inició con la propuesta de investigación, para plantear el problema a investigar. Luego, fue necesario recopilar la teoría y conceptos que sustentaron la investigación y la metodología utilizada. Posteriormente, se presentó la metodología en la semana 3. Seguidamente, fueron desarrollados el diseño del sistema sensorial como el diseño del sistema de control. El entrenamiento de cada uno de los modelos ANFIS fue ejecutado. Obteniendo los resultados y

análisis de la presente investigación. Que permitieron formular las conclusiones y recomendaciones. Para hacer la entrega final en la semana 10.

Tabla 1-Cronograma de Actividades

Actividades Desarrolladas	Semana									
	1	2	3	4	5	6	7	8	9	10
Planteamiento del Problema	■									
Teoría de sustento		■								
Metodología			■							
Diseño del Sistema Sensorial				■	■					
Diseño del Sistema de Control						■	■			
Entrenamiento del ANFIS								■		
Resultados y Análisis									■	
Conclusiones y recomendaciones									■	
Entrega Final										■

Fuente: Elaboración Propia

CAPÍTULO V. RESULTADOS Y ANÁLISIS

En el presente capítulo, se detallan tanto los resultados obtenidos como su respectivo análisis, demostrando el origen y efecto de los resultados, se muestra tanto el diseño del sistema de navegación como el entrenamiento de los modelos ANFIS.

5.1 DISEÑO DEL SISTEMA SENSORIAL

Los robots móviles se caracterizan por su capacidad de desplazarse de manera autónoma en un entorno desconocido o solamente conocido de manera parcial. Además, cuentan con características específicas que los hacen aptas para una tarea determinada. Dicha tarea o aplicación es la que determina el diseño del robot, especialmente las particularidades sensoriales del robot. Gracias al sistema sensorial o sistema de percepción de un robot móvil, este es capaz de hacer frente a situaciones cambiantes del entorno, incluyendo una reacción rápida ante posible eventos inesperados mientras el robot se encuentra en movimiento. En consecuencia, el entorno por donde se va a movilizar el robot posee influencia sobre su diseño. Sin embargo, un diseño de sistema sensorial, puede ser aplicado a diferentes robots móviles.

Esta información del entorno debe ser abundante en calidad y cantidad de tal forma que el robot móvil realice la tarea de manera similar al operador humano. Por ello, un robot móvil rara vez va equipado con un único sensor para realizar todas sus tareas, sino que la práctica más habitual consiste en combinar dentro del sistema sensorial varios sensores que en mayor o menor medida se complementan. Si bien la variedad de sensores que pueden emplearse en un robot móvil está directamente relacionada con el campo de aplicación de éstos, hay una serie de sensores que se pueden considerar más acordes con las funciones del robot.

Para el diseño de un sistema inteligente, la selección de sensores adecuados juega un papel vital. La vida útil del sistema y sus matrices de rendimiento se asociaron en gran medida con la selección de los sensores. En este trabajo, el sensor de infrarrojo fue considerado para el diseño del sistema inteligente, ver ilustración 9.



Ilustración 9-Sensor IR SHARP

Fuente: (SHARP, 2020)

Los sensores ultrasónicos son otro tipo de sensores utilizados para robots móviles. En comparación con el sensor ultrasónico, el sensor de infrarrojos ofrece una alta resolución con un tiempo de respuesta más rápido. El sensor tiene un transmisor de infrarrojos y un detector óptico sensible a la posición, que utiliza el principio de triangulación óptica para la medición de distancias. El sensor de infrarrojos exhibe características no lineales en varias superficies. El sensor utilizado en este experimento puede proporcionar una lectura constante solo después de 5 cm. La especificación técnica del sensor infrarrojo utilizado para el experimento se da en la tabla 2.

Tabla 2-Especificaciones de los sensores

Parámetros	Sensor IR (SHARP)
Alcance del sensor	10-80 cm
Ancho del Haz	75 deg
Patrón del Haz	Estrecho
Características de la radiación EM	Longitud de onda = 870 ± 70 nm

Fuente: (SHARP, 2020)

Otro sensor utilizado dentro del sistema sensorial es el giroscopio. Un giroscopio es un dispositivo que se utiliza para medir o mantener la orientación y la velocidad angular. Es una rueda o disco giratorio en el que el eje de rotación (eje de giro) es libre de asumir cualquier orientación por sí mismo. Al girar, la orientación de este eje no se ve afectada por la inclinación

o rotación del soporte, de acuerdo con la conservación del momento angular. Por esta razón, este sensor le permite al robot móvil determinar su dirección y orientación actual. El circuito integrado de MPU-6050, ver ilustración 10, contiene un acelerómetro y giroscopio MEMS en un solo encapsulado (InvenSense, 2013). Es uno de los sensores más precisos del mercado con sus 16-bits de resolución, lo que significa que divide el rango dinámico en 65536 fracciones y cuenta con las siguientes características.

- 1) Módulo MPU-6050 (giroscopio de tres ejes + acelerómetro triaxial).
- 2) Chip: MPU-6050.
- 3) Alimentación: 3-5V (incluye un regulador interno LDO).
- 4) Comunicación: Digital I2C.
- 5) Convertidor analógico-digital de 16 bit en todos los ejes.
- 6) El giroscopio dispone de cuatro escalas programables por el usuario: ± 250 , ± 500 , ± 1000 y ± 2000 ° / seg (dps).
- 7) La escala del acelerómetro es programable por el usuario con valores de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$.
- 8) Pines de paso estándar: 2,54 mm
- 9) Tamaño: 2 x 1.6 x 0.1mm



Ilustración 10-MPU-6050

Fuente: (NoMADA, 2016)

El diseño del controlador ANFIS propuesto para la navegación del robot móvil se encarga de controlar los drivers de los motores. Por lo tanto, el sistema de control cuenta cuatro entradas y dos salidas. Las primeras tres entradas es la distancia entre el robot y algún obstáculo presente dentro de su rango de visión. Se obtiene retroalimentación de la distancia a la que se encuentran los obstáculos a través de los sensores infrarrojos ubicados enfrente, a la derecha y a la izquierda, como se muestra en la ilustración 11. Las primeras 3 entradas se denominan como Sensor Enfrente (x_1), Sensor Izquierdo (x_2), Sensor Derecho (x_3), respectivamente.

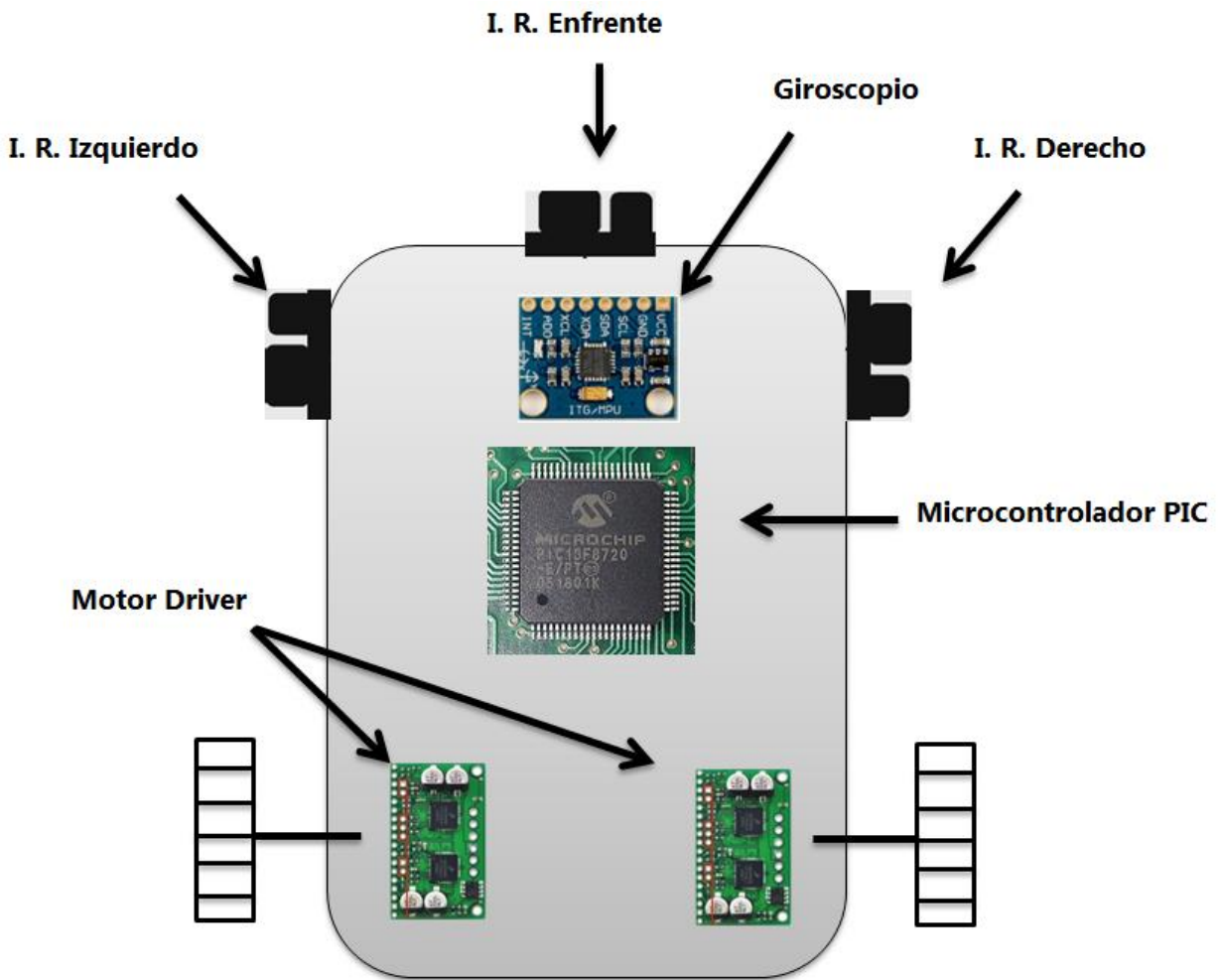


Ilustración 11-Distribución de los sensores del robot móvil

Fuente: Elaboración Propia

La cuarta entrada del sistema de control es el giroscopio (x_4), que proporciona retroalimentación de la orientación del robot móvil. Las cuatro señales de entradas son

alimentadas al microcontrolador PIC que a su vez controla las salidas a través del modelo ANFIS diseñado. Las dos salidas a controlar son los drivers o controladores de los motores, como se muestra en la ilustración 11. Cada salida representa cada una de los motores, el motor izquierdo (PWMI) y el motor derecho (PWMD).

La cantidad de entradas y salidas depende directamente del diseñador al momento de desarrollar el sistema de navegación. De igual manera, puede variar el tipo de sensores y salidas que utilizan dentro del controlador. M. K. Singh et al. (2009) en su diseño de un controlador ANFIS utilizó 9 sensores infrarrojos de proximidad y de luz ambiental 2 sensores infrarrojos de proximidad al suelo para aplicaciones de seguimiento de línea, 5 sensores ultrasónicos con rango de 0,2 a 4 metros. Sin embargo, para fines de control, el modelo ANFIS solamente contaba con un total de 3 entradas de distancia, siendo la distancia de los obstáculos enfrente, a la izquierda y derecha. Posee una cuarta entrada que es la dirección de destino. Similarmente, Chen et al. (2021) hacen uso de un total de ocho sensores ultrasónicos distribuidos con una distancia equitativa en la parte frontal del robot. A pesar de ello, el controlador que propusieron posee tres entradas de la distancia entre el robot móvil y el obstáculo. Esto lo lograron ya que propusieron que la señales de entradas al controlador fue el valor mínimo de entrada que está retroalimentando uno de los sensores en un área en específico. Además, poseen una cuarta entrada que es el ángulo entre el robot y el objetivo. En la tabla 3, se muestra diversos controladores propuestos junto con su diseño del sistema sensorial.

Tabla 3-Sistema sensorial de diversos controladores

Investigadores	Tipo de Sensores	Cantidad de Sensores	Entradas al Controlador
(M. K. Singh et al., 2009)	Infrarrojo/Ultrasónico	16	4
(Chen et al., 2021)	Ultrasónico	9	4
(Mohanty & Parhi, 2015)	Infrarrojo	8	4
(Pandey Anish et al., 2019)	Infrarrojo	3	3
(Pandey et al., 2016)	Infrarrojo/Ultrasónico	3	3
(Subbash & Chong, 2019)	Ultrasónico	3	3
(Ramkumar et al., 2018)	Ultrasónico/RP Lidar	3	2
Presente	Infrarrojo	4	4

Fuente: Elaboración Propia

Mohanty & Parhi (2015) utilizaron en su controlador propuesto 8 sensores infrarrojos de proximidad y de luz ambiental con un alcance de hasta 100 mm. Las entradas de su algoritmo de control consistían en tres entradas de obstáculos, distancia de obstáculo enfrente, a la derecha y la izquierda. Pandey Anish et al. (2019) hicieron uso de 3 sensores infrarrojos que alimentaban directamente el controlador que desarrollaron. Similarmente, fueron utilizados 3 sensores por los autores Pandey et al. (2016) que consistían dos sensores infrarrojos y uno ultrasónico. El controlador propuesto posee una cuarta entrada que es el ángulo objetivo, que es el ángulo entre el objetivo y el robot. Como salida a controlar, propusieron el ángulo de dirección. Subbash & Chong (2019) diseñaron un controlador ANFIS con 3 entradas, dichos valores eran retroalimentado por 3 sensores ultrasónicos, respectivamente para cada entrada. Las variables a controlar en eran las velocidad de cada una motores. Para una mejor percepción del medio ambiente Ramkumar *et al.* (2018) presentaron un sistema sensorial que consistía en el uso de dos sensores ultrasónico y un RP Lidar 360°. Gracias a estos sensores utilizados obtuvieron una mejor percepción del entorno.

Como se observa de la tabla 3, mayormente se utilizan los sensores más utilizados es el ultrasónico e infrarrojo. La cantidad de sensores permite tener una mejor percepción del ambiente, sin embargo, a nivel de control no se utilizan todas las señales de lectura ya que esto requeriría un mayor requerimiento de recursos computacionales. Por esta razón reducen las señales a una menor cantidad, generalmente, tres señales, que representan cada uno de los lados, enfrente, izquierda y derecha. Esto se logra de diferentes manera, en primera instancia, seleccionando el valor mínimo de lectura de diversos sensores en una área, tal como fue presentado por Chen *et al.* (2021) en su diseño de controlador para la navegación de robots móviles. En otros casos, se utiliza la fusión de datos para determinar la distancia de los obstáculos en el ambiente. Adarsh & Ramachandran (2018) diseñaron un algoritmo para la fusión de datos con un modelo ANFIS, donde utilizaron dos entradas, un sensor infrarrojo y un ultrasónico para determinar la distancia exacta de un objeto u obstáculo en el ambiente a partir de estas dos lecturas. Esto permite el uso de diferente número y tipos de sensores y determinar la distancia exacta de un objeto en el ambiente. Por lo tanto, en la presente investigación se

busca utilizar tres entradas de distancia y una de ángulo para determinar el accionamiento del robot móvil.

5.2 DISEÑO DE ANFIS

En la presente sección, se desarrolla el controlador basado en una arquitectura ANFIS utilizado para la navegación del robot móvil y para la evasión de obstáculos. ANFIS es un método para formular la relación de mapeo entre la entrada y la salida utilizando el concepto matemático de lógica difusa. Esta arquitectura ANFIS funciona con base en la técnica de aprendizaje neuroadaptativo tipo Takagi-Sugeno y este método es más compacto, computacionalmente eficiente y su capacidad para personalizar las funciones de membresía en comparación con el sistema tipo Mamdani.

La arquitectura ANFIS propuesta posee un total de cuatro entradas y dos salidas. Tres de las entradas representan la distancia de un obstáculo y el robot de cada uno de los lados del mismo, incluyendo enfrente, derecha e izquierda. Estas entradas se denominan Sensor Enfrente (x_1), Sensor Izquierdo (x_2), Sensor Derecho (x_3). La cuarta entrada es la señal retroalimentada por el giroscopio que permite determinar la orientación y dirección que el robot está siguiendo.

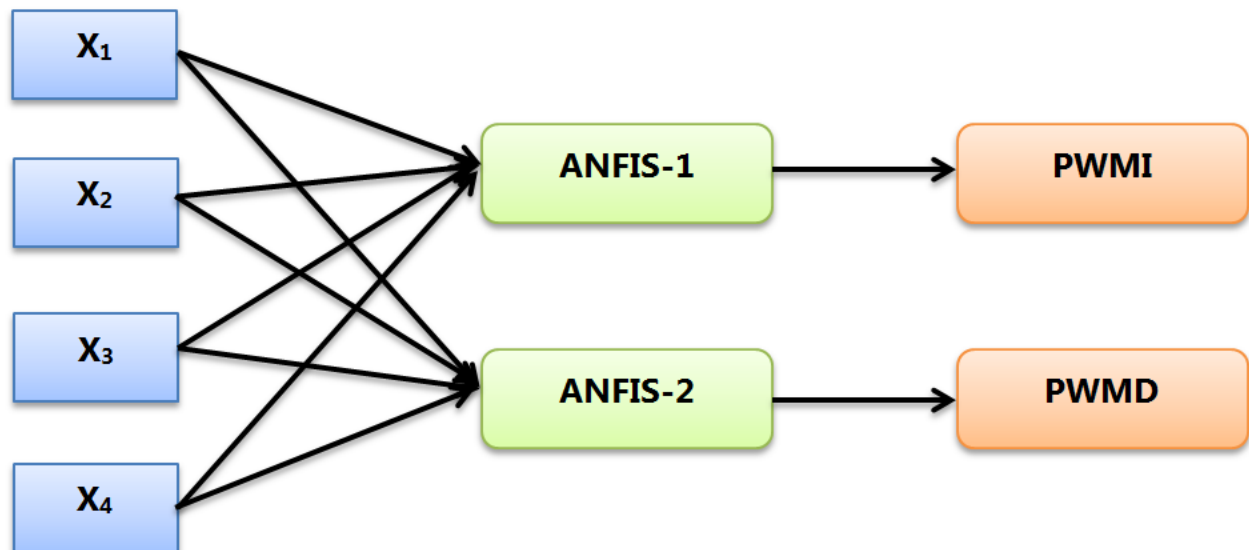


Ilustración 12-Diseño del controlador ANFIS

Fuente: Elaboración Propia

Con el modelo de control desarrollado, se busca el control sobre los motores a través del valor PWM asignado a cada uno de los motores. El robot móvil cuenta con dos motores, uno en cada lado del robot, cada uno de ellos representa una salida del controlador. Por lo tanto, la dos salidas del controlador ANFIS es el PWM del motor izquierdo (PWMI) y del motor derecho (PWMD). La arquitectura ANFIS basada en el FIS de tipo Takagi-Sugeno permite múltiples entradas, sin embargo, solamente permite una salida. En el presente sistema de navegación, se busca controlar dos diferentes salidas, una para cada motor que posee el robot móvil. En consecuencia, se desarrollaron y entrenaron dos ANFIS diferentes una para cada salida del sistema de navegación, tal como se muestra en la ilustración 12.

Por lo tanto, se tiene dos arquitecturas ANFIS diferentes. Sin embargo, poseen los mismos parámetros de diseño para el entrenamiento del modelo. Cuentan con la misma cantidad de entradas, capas y funciones de membresía de cada entrada, como se muestra en la ilustración 13. Cada una de las variables de entrada posee 3 funciones de membresía o pertenencia, Cerca (A_1), Medio (A_2) y Lejos (A_3), así respectivamente, para cada una de las entradas de la arquitectura ANFIS. En la tabla 4, se resumen las variables lingüísticas del controlador diseñado, cabe destacar que estos parámetros son simétricos para cada ANFIS que controla cada una de los motores en el sistema de navegación.

Tabla 4-VARIABLES LINGÜÍSTICAS

Variable Lingüística	Universo de Discurso	Valores Lingüístico
Sensor Enfrente (x_1)	5 a 15 cm	Cerca (A_1)
		Medio (A_2)
		Lejos (A_3)
Sensor Izquierdo (x_2)	5 a 15 cm	Cerca (B_1)
		Medio (B_2)
		Lejos (B_3)
Sensor Derecho (x_3)	5 a 15 cm	Cerca (C_1)
		Medio (C_2)
		Lejos (C_3)
Ángulo (x_4)	-180 a 180 grados	Negativo (D_1)
		Cero (D_2)
		Positivo (D_3)

Fuente: Elaboración Propia

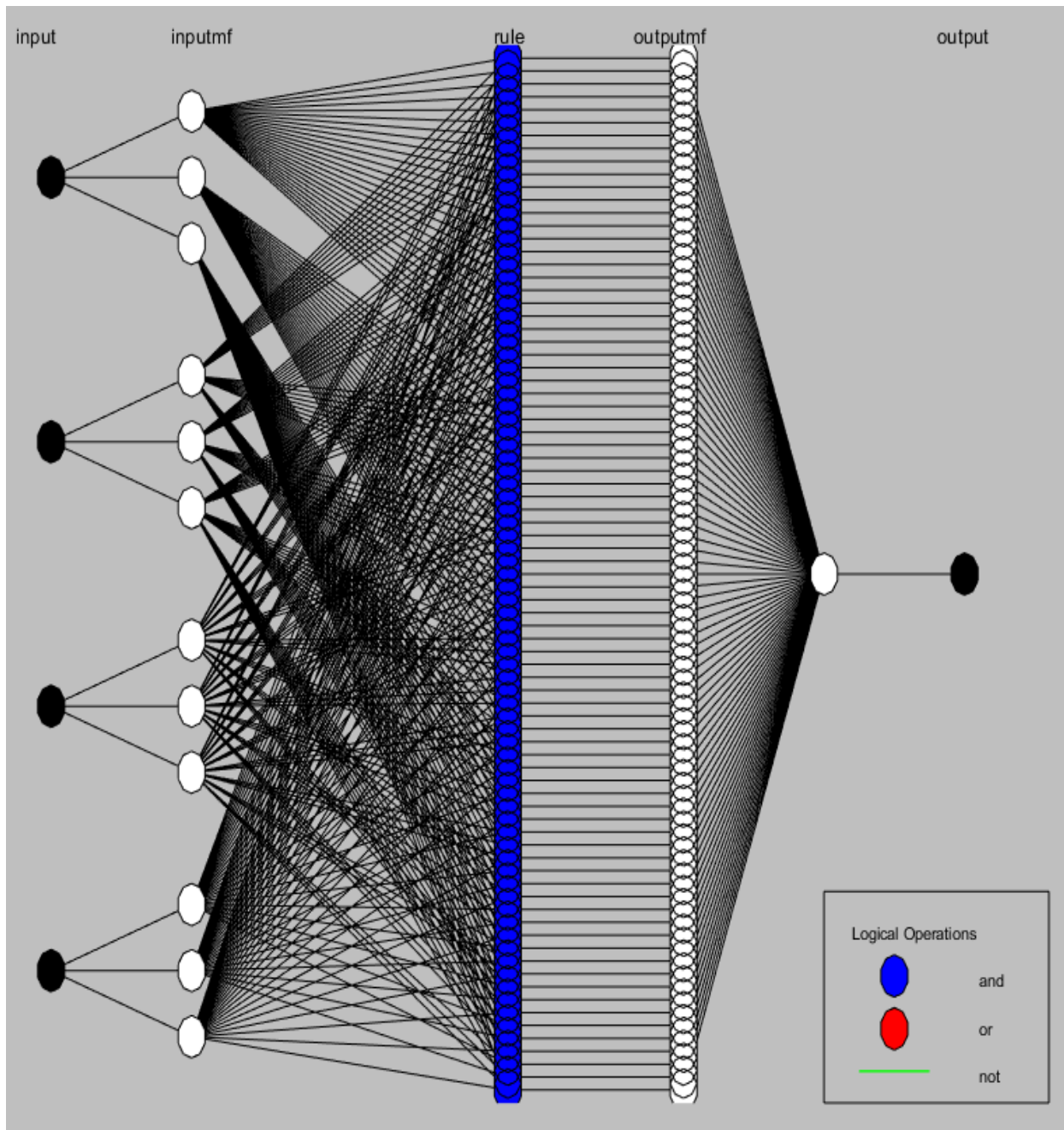


Ilustración 13-Arquitectura del controlador ANFIS

Fuente: Elaboración Propia

Posteriormente, ya que el sistema de inferencia difuso que utiliza la arquitectura ANFIS es el de Takagi-Sugeno, las reglas de si-entonces se formulan tal como se especifica en la ecuación 16, tomando en cuenta las variables lingüísticas establecidas con anterioridad, en conjunto con los

valores lingüísticos que son cada una de las funciones de pertenencia. Asumiendo que la salida es una función lineal.

$$\begin{aligned} & \text{IF } (x_1 \text{ is } A_i \text{ and } x_2 \text{ is } B_i \text{ and } x_3 \text{ is } C_i \text{ and } x_4 \text{ is } D_i) \text{ THEN } (PWM_z (f_n)) \\ & = p_n x_1 + q_n x_2 + r_n x_3 + s_n x_4 + u_n \end{aligned}$$

Ecuación 16-Regla Si-Entonces con salida lineal

Donde A_i, B_i, C_i y D_i son las funciones de membresía de cada una de las variables de entrada del controlador x_1, x_2, x_3 , y x_4 , respectivamente. Donde $i = 1, 2, 3$ y $n = 1, 2, \dots, 81$, dicho valor representa el número de reglas que contiene el modelo ANFIS que depende directamente de la cantidad de variables y valores lingüísticos. Por otro lado, p_n, q_n, r_n, s_n , y u_n son los parámetros lineales de la función de salida f_n , también conocidos como parámetros consecuentes y cambiando estos parámetros podemos ajustar la salida del controlador ANFIS. Y z representa la salida PWM ya sea para el motor izquierdo o derecho. De igual manera, en vez de una salida de función lineal en el controlador se puede tener una constante. En consecuencia, la regla Si-Entonces del FIS Takagi-Sugeno estaría formulado tal como se especifica en la ecuación 17.

$$\text{IF } (x_1 \text{ is } A_i \text{ and } x_2 \text{ is } B_i \text{ and } x_3 \text{ is } C_i \text{ and } x_4 \text{ is } D_i) \text{ THEN } (PWM_z (f_n)) = u_n$$

Ecuación 17-Regla Si-Entonces con salida constante

Donde los parámetros son los mismos descritos con anterioridad. Solamente, cambia la salida f_n a un valor constante, que sería el parámetro consecuente. Por otro lado, con respecto a los parámetros de premisa que se encuentra en la capa adaptativa 1, ver ilustración 11, se encuentra un total de 12 conjuntos $\{a_i, b_i, c_i\}$ que representa cada una de las funciones de membresía en el ANFIS propuesto para el control de navegación del robot móvil. Estos parámetros son los parámetros de las funciones de membresía, en caso que la función de membresía se de forma campana serían los parámetros de la ecuación 7, en caso que la función de membresía a utilizar fuera la triangular serían los parámetros de la ecuación 5. Por lo tanto, para el sistema de control propuesto, la cantidad de parámetros de premisa por ajustar sería 3 para cada una las 12 funciones de membresía, son un total de 36 parámetros por cada ANFIS desarrollada. La cantidad de reglas difusas para el modelo ANFIS propuesto son 81, ya que

depende de la cantidad de funciones de membresía de cada variable de entrada. El total de parámetros consecuentes depende si la salida es una función lineal o constante. En caso de que esta sea una función constante, los parámetros de salida serían 5 por cada una de las reglas, un total de 405 parámetros consecuentes, la salida se obtendría a partir de la ecuación 18.

$$PWM_z = \frac{\sum_i w_i (p_n x_1 + q_n x_2 + r_n x_3 + s_n x_4 + u_n)}{\sum_i w_i}, \quad i = 1, 2, \dots, 81$$

Ecuación 18-Peso promedio con salida lineal

Dicha ecuación se parte de la ecuación 10, ya que el tipo de FIS utilizado es el Sugeno. Por otro lado, si la salida es constante, solamente serían 81 parámetros consecuentes por ajustar, se obtiene a partir de la ecuación 19.

$$PWM_z = \frac{\sum_i w_i (u_n)}{\sum_i w_i}, \quad i = 1, 2, \dots, 81$$

Ecuación 19-Peso promedio con salida constante

Donde i representa el número de reglas de la arquitectura ANFIS diseñada. En la tabla 5, se resumen los parámetros por ajustar con los algoritmos de aprendizaje, en cada una de los casos a utilizar para el diseño de la arquitectura ANFIS.

Tabla 5-Cantidad de parámetros por ANFIS

Parámetros	Cantidad
Entradas	4
Funciones de Membresía por Entrada	3
Funciones de Membresía (Total)	12
Parámetros de Premisa (FM Triangular)	3
Parámetros de Premisa (FM Gaussiana)	2
Reglas	81
Parámetros Consecuentes (Lineal)	405
Parámetros Consecuentes (Constante)	81
Total de Parámetros por Ajustar (Lineal/FM Triangular)	441
Total de Parámetros por Ajustar (Lineal/FM Gaussiana)	429
Total de Parámetros por Ajustar (Constante/FM Triangular)	117
Total de Parámetros por Ajustar (Constante/FM Gaussiana)	105

Fuente: Elaboración Propia

Durante el proceso de aprendizaje para la navegación segura del robot móvil en varios entornos, ANFIS modifica los parámetros de la función de pertenencia de entrada y los

parámetros consecuentes con el objetivo de minimizar el error entre el entrenamiento y los valores predichos de múltiples salidas. El conjunto de datos de entrenamiento entrena la red ANFIS, mientras que el conjunto de datos de prueba verifica la precisión y la eficacia de la red ANFIS desarrollada.

La construcción de base de datos para el entrenamiento es de vital importancia para el desarrollo del ANFIS. Ya que dependiendo de estos datos se obtendrá una mejor precisión de la navegación del robot móvil. En el sistema actual de navegación se buscan abarcar diversos comportamientos reactivos, entre ellos se encuentra la evasión de obstáculos. En la ilustración 14, se muestra diversos ejemplos de diferentes comportamientos reactivos que se busca que el robot móvil ejecute en tal caso dado.

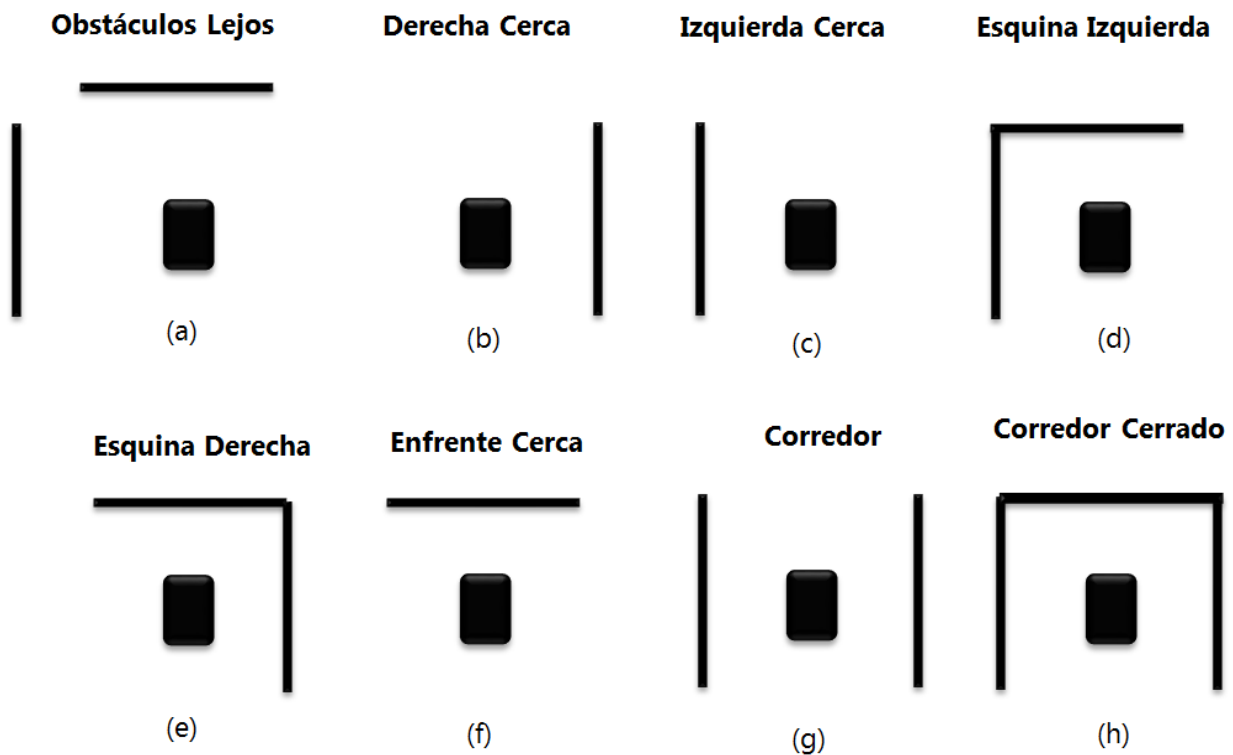


Ilustración 14-Ejemplo de comportamientos reactivos

Fuente: Elaboración Propia

Tal como se muestra en la ilustración 14, se tomaron diversos comportamientos reactivos que se busca que el sistema de navegación siga, y esto lo logra a través del entrenamiento. Para esto se creó la base de datos a partir de estos comportamientos, buscan la evasión de obstáculos

cuando se presente los diferentes casos descritos en la ilustración 12. Para la evasión de obstáculos, cuando el robot móvil detecta (por información sensorial) cualquier obstáculo en el lado frontal, izquierdo o derecho. Este comportamiento se activa para evitar colisiones con obstáculos. Cuando la información recibida de los sensores muestra la presencia de obstáculos en el lado frontal, izquierdo y derecho del robot, el robot invierte su movimiento. Esto es logrado a través de la variación de la salida PWM de cada uno de los motores que posee el robot móvil. Según cambios que se presenten en el entorno del robot móvil, las salidas reaccionan. En la tabla 6, se muestra algunos de los datos utilizados para el entrenamiento del ANFIS.

Tabla 6-Datos de entrenamiento

Sensor Enfrente (x_1)	Sensor Izquierdo (x_2)	Sensor Derecho (x_3)	Ángulo (x_4)	PWMI	PWMD
15	15	15	0	240	240
12.5	15	15	0	240	200
15	5	15	-60	240	80
10	15	15	0	240	160
7.5	15	15	0	240	120
15	5	15	-75	240	80
15	15	12.5	-75	200	240
15	15	12.5	60	200	240
15	7.5	15	0	240	120
15	15	12.5	15	200	240
15	12.5	15	75	240	200
15	15	10	15	160	240
15	15	10	-45	160	240
15	15	7.5	-15	120	240
15	10	15	-45	240	160
15	7.5	15	-45	240	120
10	15	15	-75	240	160

Fuente: Elaboración Propia

Una vez creada la base de datos se procede con el entrenamiento de la red ANFIS. Sin embargo, es necesario, con anterioridad, establecer diversos parámetros para el diseño de la red ANFIS. En primera instancia se debe determinar el tipo de partición de espacio de entrada. En un FIS, básicamente hay tres tipos de particiones de espacio de entrada: cuadrícula (*grid*), árbol (*tree*) y

particiones de dispersión (*scattering partitioning*). Para la presente investigación se seleccionó el tipo cuadrícula. Este utiliza partición de cuadrícula y genera reglas enumerando todas las combinaciones posibles de funciones de pertenencia de todas las entradas. Esto conduce a una explosión exponencial incluso cuando el número de entradas es moderadamente grande. Por ejemplo, para un FIS con 10 entradas, cada una con dos funciones de pertenencia, la partición de la cuadrícula conduce a 1024 ($=2^{10}$) reglas, que es inhibitoriamente grande para cualquier método de aprendizaje práctico, esto conlleva a lo que se conoce comúnmente como la "maldición de la dimensionalidad". Se refiere a una situación en la que el número de reglas difusas, cuando se utiliza la partición de cuadrícula, aumenta exponencialmente con el número de variables de entrada. Otros parámetros o especificación a establecer para el entrenamiento de la red ANFIS se describen en la tabla 7.

Tabla 7-Especificación de la arquitectura ANFIS

Especificación	Tipo
Tipo de FIS Generado	Sugeno
Tipo de partición de espacio de entrada	Partición de Cuadrícula
Método de Decisión para Operación Lógica Difusa AND (mínimo)	Producto
Método de Decisión para Operación Lógica Difusa OR (máximo)	Probabilístico
Método de Defusificación de Salida	Peso Promedio (Ecuación 10)
Número de Funciones de Membresía para (x_1)	3
Número de Funciones de Membresía para (x_2)	3
Número de Funciones de Membresía para (x_3)	3
Número de Funciones de Membresía para (x_4)	3
Tipo de Función de Membresía de Entrada	Gaussiana/Triangular
Tipo de Salida	Constante/Lineal
Número de Reglas	81
Método de Entrenamiento	Híbrido
Número de Épocas	Variable

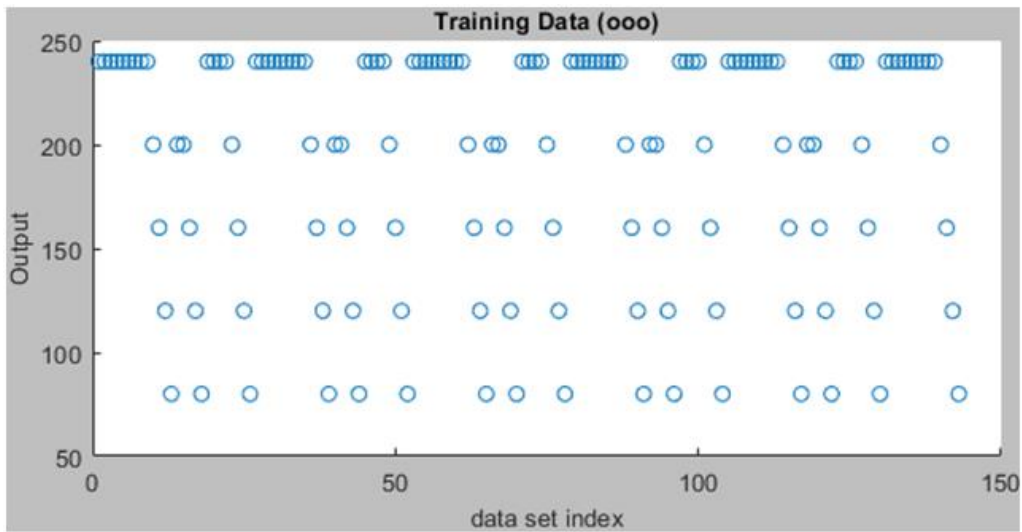
Fuente: Elaboración Propia

5.3 ENTRENAMIENTO DE ANFIS-1

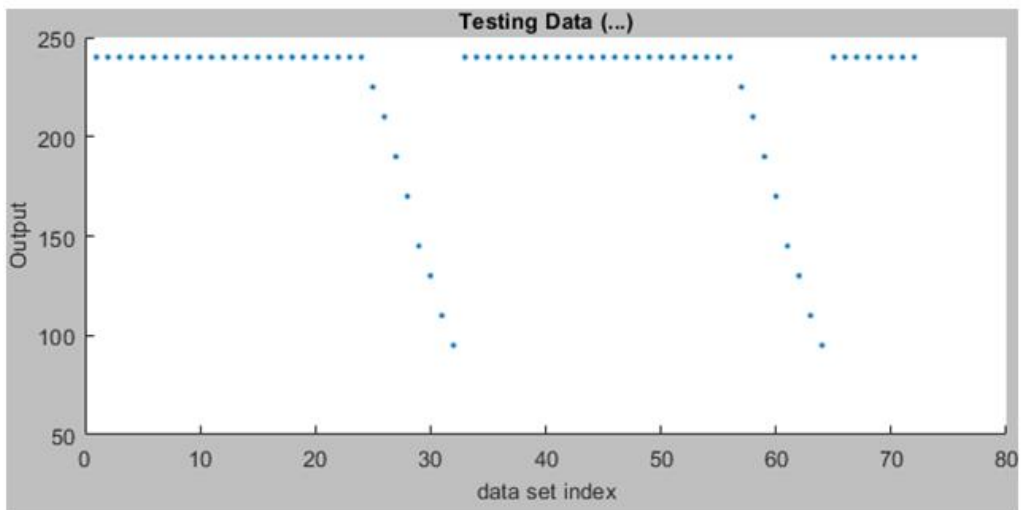
Una vez establecidos todos los parámetros del ANFIS se procede a su desarrollo y entrenamiento. En esta sección se muestran los resultados obtenidos del entrenamiento de la ANFIS-1 que se encarga del control sobre el motor ubicado en la izquierda, con la salida

denominada como PWMI. Con la finalidad de obtener el mejor modelo que describa la salida a controlar, fueron desarrollados 4 modelos ANFIS variando dos parámetros de su configuración. En primera instancia, se varió el tipo de función de membresía de entrada, que influye en la cantidad de parámetros de premisa o parámetros no lineales, se selecciona entre una función de membresía triangular (ecuación 5) y una función gaussiana (ecuación 7). Y el segundo parámetro a modificar es el tipo de salida ya sea lineal (ecuación 16) o constante (ecuación 17), esto influye significativamente en la cantidad de parámetros consecuentes que el algoritmo híbrido de aprendizaje debe de ajustar.

Inicialmente, se establecen la base de datos de entrenamiento en dos partes. Una de las partes es para el entrenamiento en sí, y la otra parte es para la prueba de la red. Para el desarrollo del ANFIS-1, se hizo de so de los datos de entrada y solamente una salida, PWMI, ver tabla 5. En la ilustración 15a se muestra los datos para entrenamiento, donde se muestra el valor de salida con respecto a cada registro de las 4 entradas (*data set index*), que represente el índice de cada registro de la base de datos. En la ilustración 15b se muestra la distribución de los datos de prueba. Ambas bases de datos mostradas en la ilustración 15, fueron utilizadas para el entrenamiento de los 4 diferentes modelos ANFIS, variando sus parámetros.



(a)



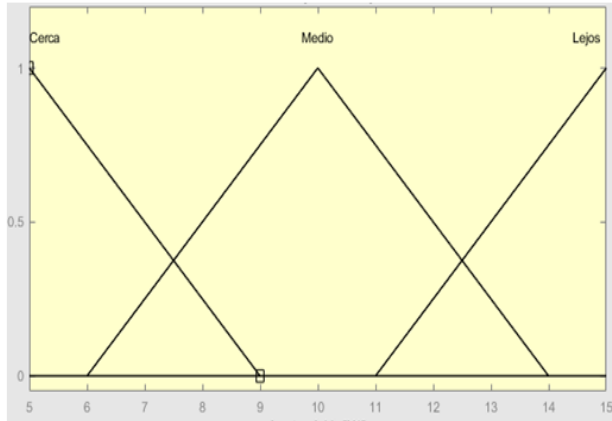
(b)

Ilustración 15-Distribución de datos de entrenamiento para ANFIS-1

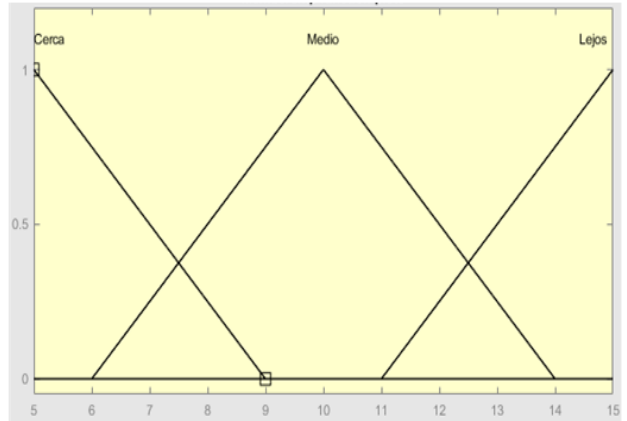
Fuente: Elaboración Propia

5.3.1 ANFIS-1 TRIANGULAR/CONSTANTE

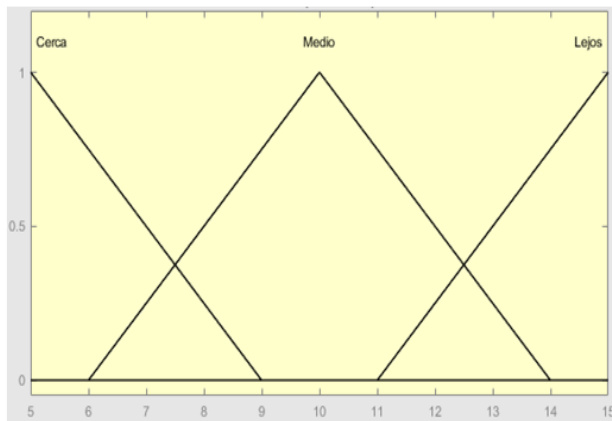
Inicialmente, se utilizaron funciones de membresía de tipo triangular. Estas FM fueron inicializadas para posteriormente, ajustar sus parámetros. En la ilustración 16, se muestra las funciones de membresía triangular utilizadas inicialmente para el modelo ANFIS. Fueron establecidas 3 funciones de membresía triangulares para cada uno de las 4 entradas.



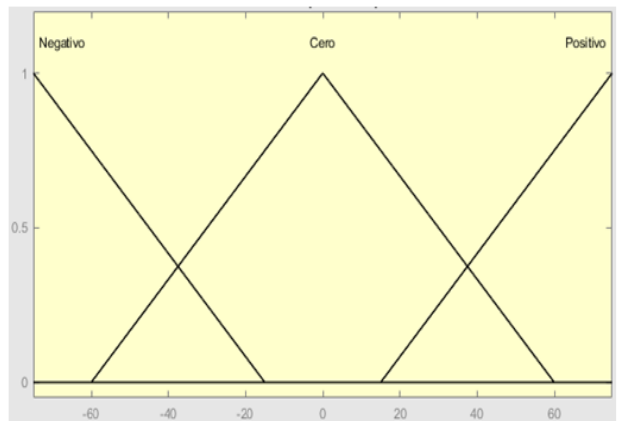
(a) Sensor Enfrente



(b) Sensor Izquierdo



(c) Sensor Derecho



(d) Ángulo

Ilustración 16-Funciones de membresía triangular antes del entrenamiento

Fuente: Elaboración Propia

El conjunto de datos obtenido en los pasos anteriores se utiliza para construir un controlador ANFIS. Se ha utilizado un método híbrido para entrenar el conjunto de datos. El error de entrenamiento de la raíz cuadrada media (RMSE) fue obtenido para determinar la precisión del modelo ANFIS desarrollado. Para el presente modelo fue seleccionada una salida del tipo constante. En la ilustración 17, se muestra el resultado del entrenamiento del modelo ANFIS, donde se detalla el cambio del error respecto a cada época.

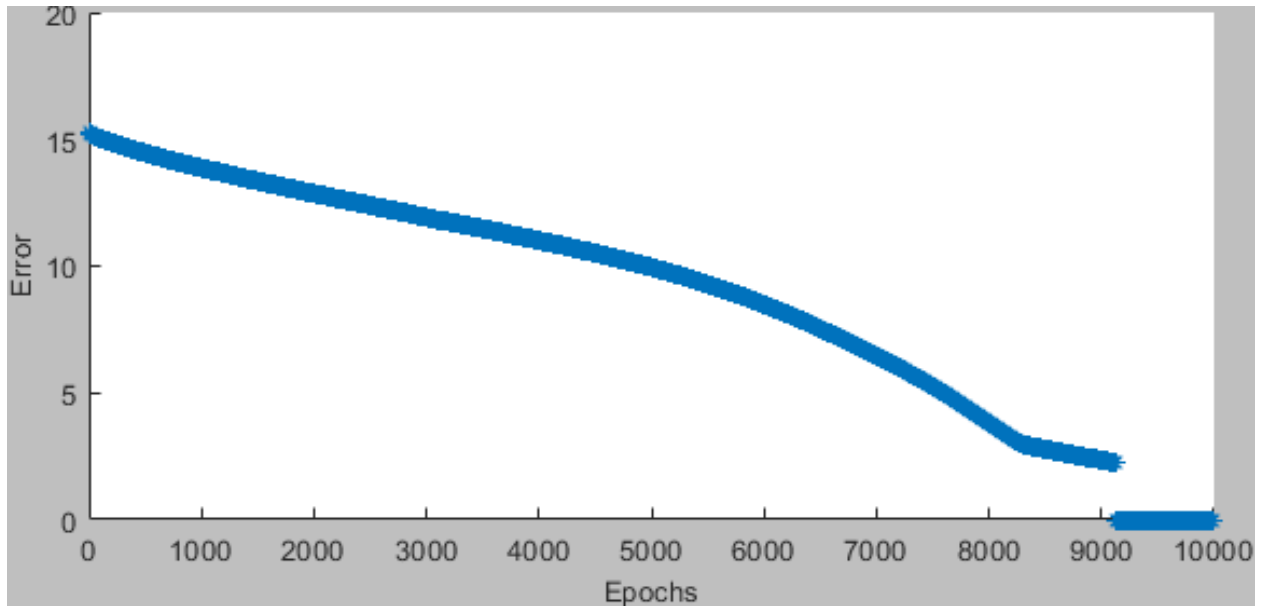


Ilustración 17-RMSE de ANFIS-1 triangular/constante

Fuente: Elaboración Propia

Tal como se muestra en la ilustración 17, para el entrenamiento del modelo ANFIS se utilizaron 10 mil épocas para obtener el menor RMSE posible. Se obtuvo una convergencia en el entrenamiento de la red ANFIS alrededor de la época 9150. El valor mínimo de RMSE obtenido fue de 2.2354 para los datos de entrenamiento. Se comprobó el modelo ANFIS desarrollado a través de la comparación de los datos de salida de la base de datos de entrenamiento, con respecto a los valores predichos por el modelo desarrollado. En la ilustración 18, se muestran los resultados obtenidos, respecto a la salida de los datos de entrenamiento respecto a los del modelo ANFIS.



Ilustración 18-Comparación con datos de entrenamiento de ANFIS-1 triangular/constante

Fuente: Elaboración Propia

Posteriormente, se compara los resultados obtenidos por el modelo entrenado con respecto a la base de datos para prueba. En la ilustración 19, se muestra los resultados obtenidos.

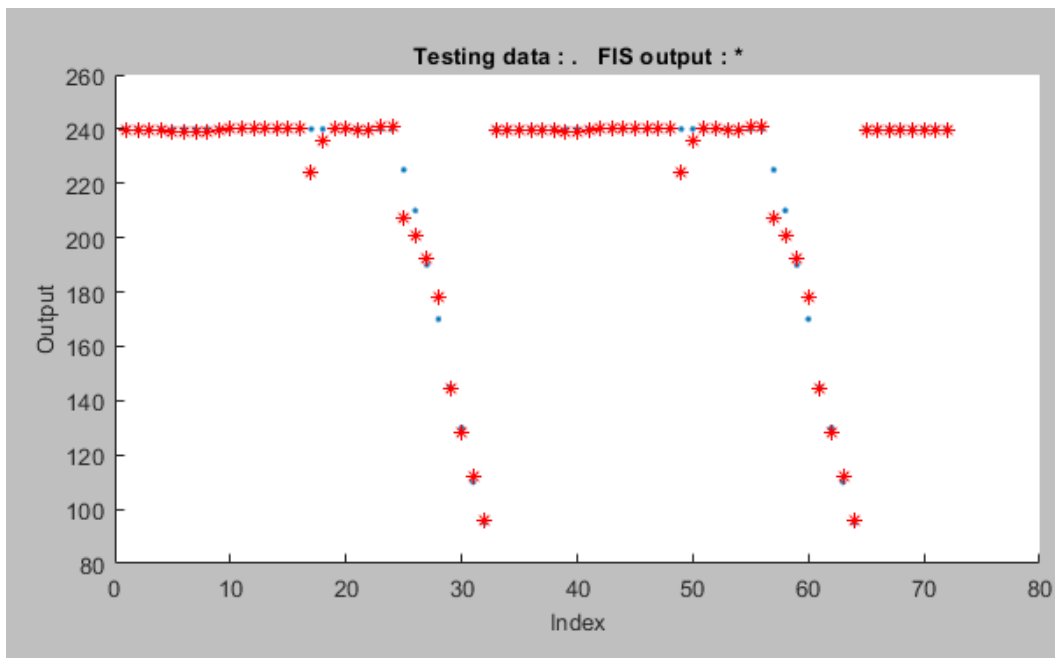


Ilustración 19-Comparación con datos de prueba de ANFIS-1 triangular/constante

Fuente: Elaboración Propia

Tal como se observa en la ilustración 19, hay diferentes registros en lo que el modelo FIS ajustado no concuerda con la salida de prueba. Por esta razón, el RMSE del entrenamiento difiere al de la prueba. En base a los resultados de prueba obtenidos, el modelo ANFIS-1 triangular/constante cuenta con un RMSE mínimo de 4.5953.

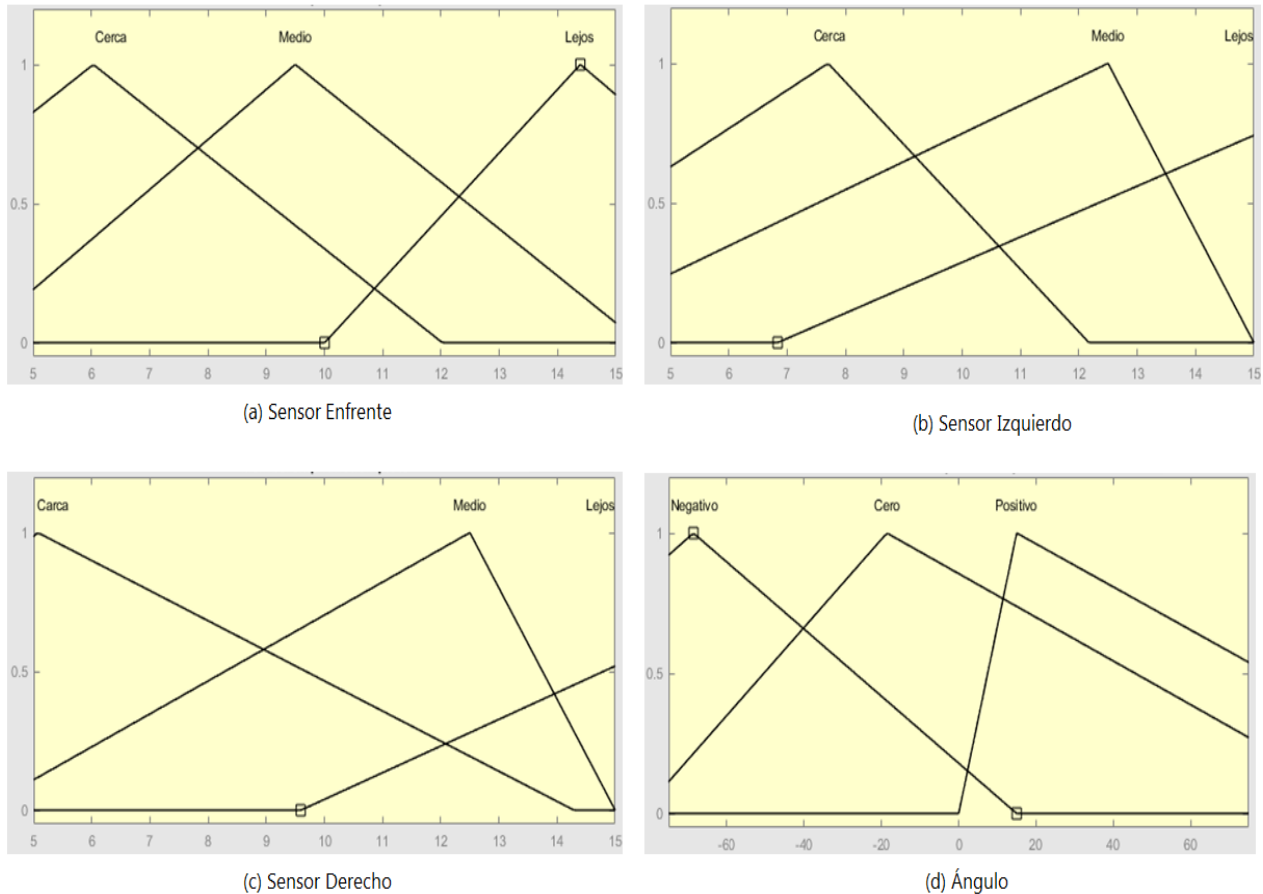


Ilustración 20-FMs triangular después del entrenamiento ANFIS con salida constante

Fuente: Elaboración Propia

Finalmente, se obtuvieron las funciones de membresía finales después del entrenamiento o ajuste de parámetros a través del algoritmo híbrido de aprendizaje. Cada una de los parámetros de dichas funciones de membresía fueron ajustados de acuerdo a los datos de entrenamiento para obtener el menor error posible. En la ilustración 20, se muestran gráficamente, las funciones de membresía del modelo FIS desarrollado y optimizado que permite la navegación del robot móvil.

5.3.2 ANFIS-1 TRIANGULAR/LINEAL

El siguiente modelo ANFIS a entrenar cuenta con funciones de membresía tipo triangular y con un tipo de salida lineal. Por lo tanto, los parámetros que el algoritmo de aprendizaje debe de ajustar son de 441, incluyendo parámetros de premisa y consecuentes. Siendo este modelo en que posee más parámetros por ajustar.

De igual manera, fueron inicializadas las funciones de membresía para el diseño de la arquitectura ANFIS. En la ilustración 16, se muestran las funciones de membresía antes del entrenamiento. Seguidamente, fueron utilizados los datos de entrenamiento, que fueron los mismos que fueron utilizados para el modelo ANFIS anterior, que se parte de los datos mostrados en la ilustración 15.

En base a dicho conjunto de entrenamiento de datos, se procedió con el entrenamiento de la red ANFIS. En la ilustración 21, se muestra la gráfica del valor de RMSE respecto a cada época del entrenamiento.

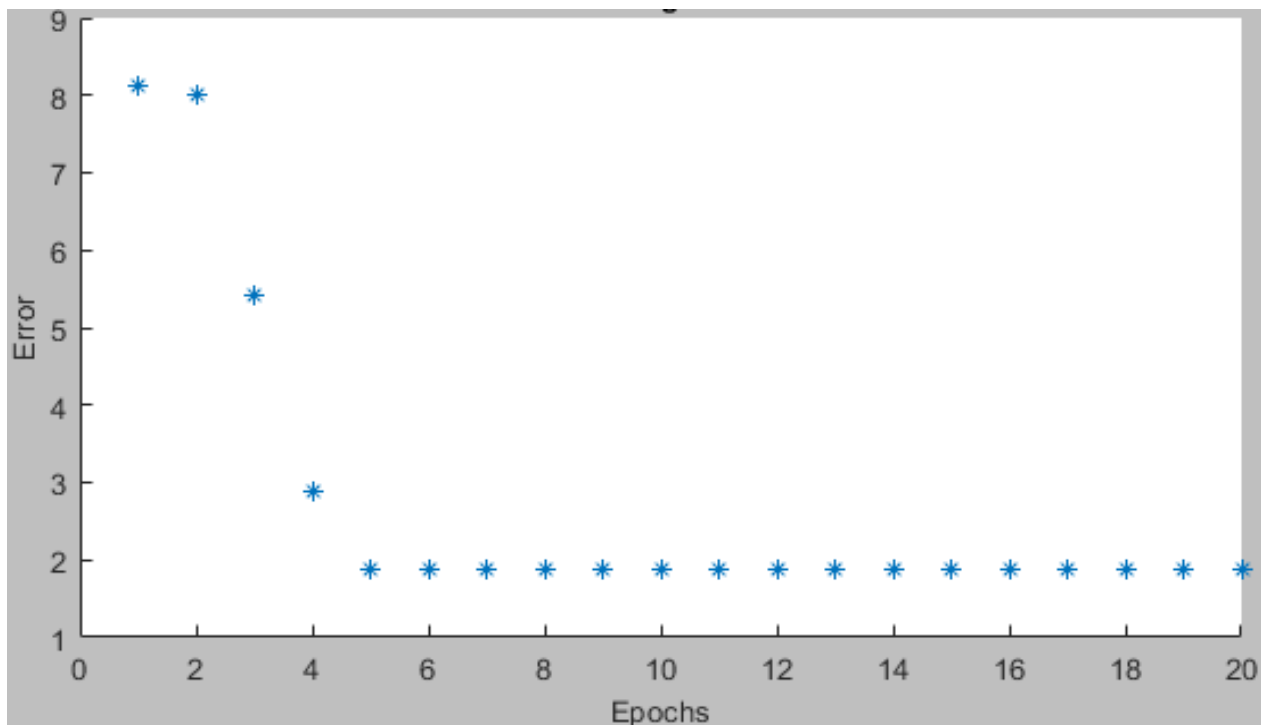


Ilustración 21-RMSE de ANFIS-1 triangular/lineal

Fuente: Elaboración Propia

Cabe destacar que a causa del mayor número de parámetros que el algoritmo de aprendizaje híbrido debe de ajustar, el tiempo de cómputo de cada época incrementa considerablemente en comparación del modelo anterior con función de membresía triangular y salida constante, ya que en este solamente fueron ajustados 117 parámetros, casi 4 veces menos. A pesar de ello, el entrenamiento de este modelo ANFIS tuvo una convergencia con un número mucho menor de épocas, solamente con 6 épocas se encontró el menor RMSE, tal como se muestra en la ilustración 19. El valor mínimo de RMSE obtenido fue de 1.865. Similarmente, el modelo fue puesto a prueba con los datos de entrenamiento, en la ilustración 22 se muestra el valor predicho por el modelo FIS optimizado con respecto al valor real de la salida.

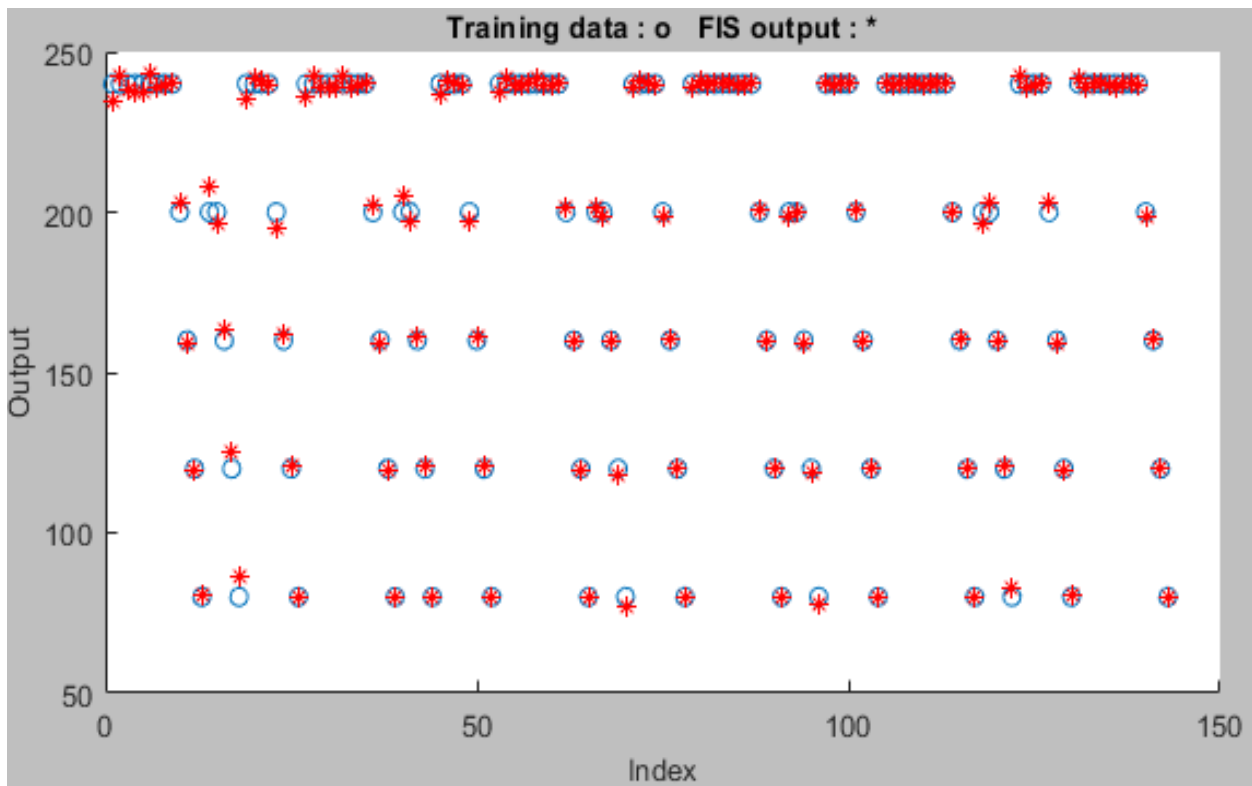


Ilustración 22-Comparación con datos de entrenamiento de ANFIS-1 triangular/lineal

Fuente: Elaboración Propia

Por otro lado, el FIS optimizado fue utilizado para la predicción de la base de datos para prueba. En la ilustración 23, se presenta el valor que se obtuvo con el modelo con respecto al valor real de la salida.

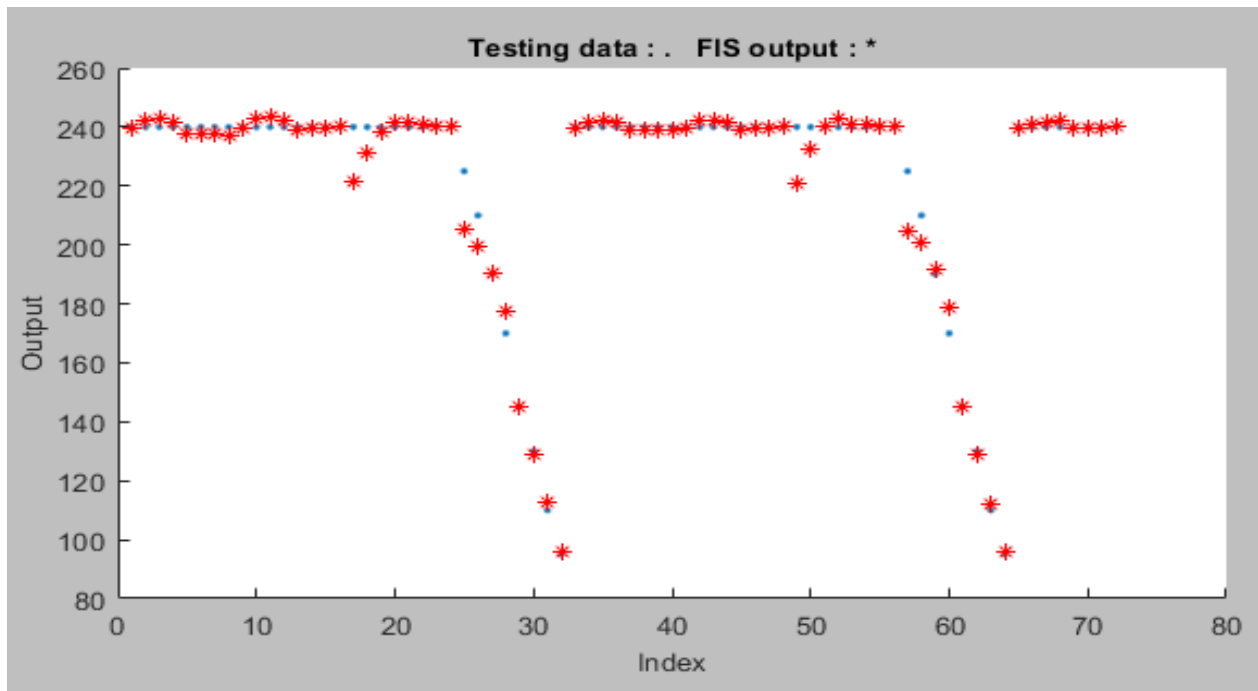


Ilustración 23-Comparación con datos de prueba de ANFIS-1 triangular/lineal

Fuente: Elaboración Propia

Cabe destacar que el RMSE varía según la base de datos utilizada. Por lo tanto, el RMSE con datos de prueba es diferente con los datos de entrenamiento. Para la base de datos de prueba, se obtuvo un RMSE mínimo de 5.4151. Generalmente, el RMSE es menor para la base de datos de entrenamiento, ya que en base a estos datos es que el algoritmo de aprendizaje va ajustando los datos.

Las funciones de membresía triangulares del modelo FIS resultado del entrenamiento se muestran gráficamente en la ilustración 24. Tal como se muestran, estas no difieren mucho en comparación como fueron inicializadas para el entrenamiento del ANFIS, ver ilustración 16. Esto se debe principalmente a que a causa de que el número de parámetros por ajustar de este modelo ANFIS es mayor, se obtuvo convergencia solamente con ajustar los parámetros consecuentes. Sin embargo, los parámetros de premisa fueron ajustados levemente, de igual manera, influyó el menor número de épocas que el modelo ANFIS requirió para obtener el menor RMSE:

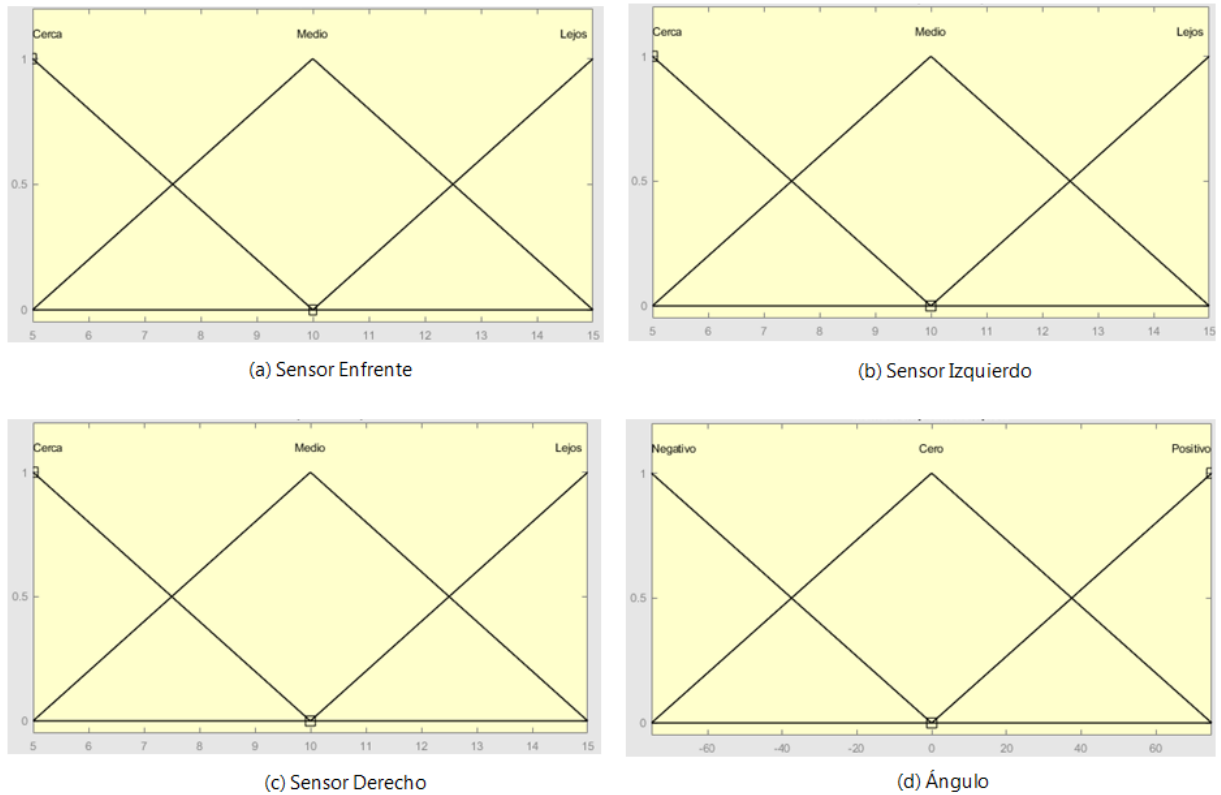


Ilustración 24-FMs tipo triangular despues del entrenamiento ANFIS con salida lineal

Fuente: Elaboración Propia

5.3.3 ANFIS-1 GAUSSIANA/CONSTANTE

El siguiente modelo ANFIS a entrenar cuenta con funciones de membresía tipo gaussiana (ecuación 7) y con un tipo de salida constante. Por lo tanto, los parámetros que el algoritmo de aprendizaje debe ajustar son de 105, incluyendo parámetros de premisa y consecuentes. Siendo este modelo en que posee menos parámetros por ajustar. Primeramente, se deben inicializar las funciones de membresía gaussiana para el entrenamiento del modelo ANFIS. En la ilustración 25, se muestran las funciones de membresía gaussiana de cada una de las entradas del ANFIS. Se establecieron un total de 3 funciones de membresía por cada una de las variables de entrada.

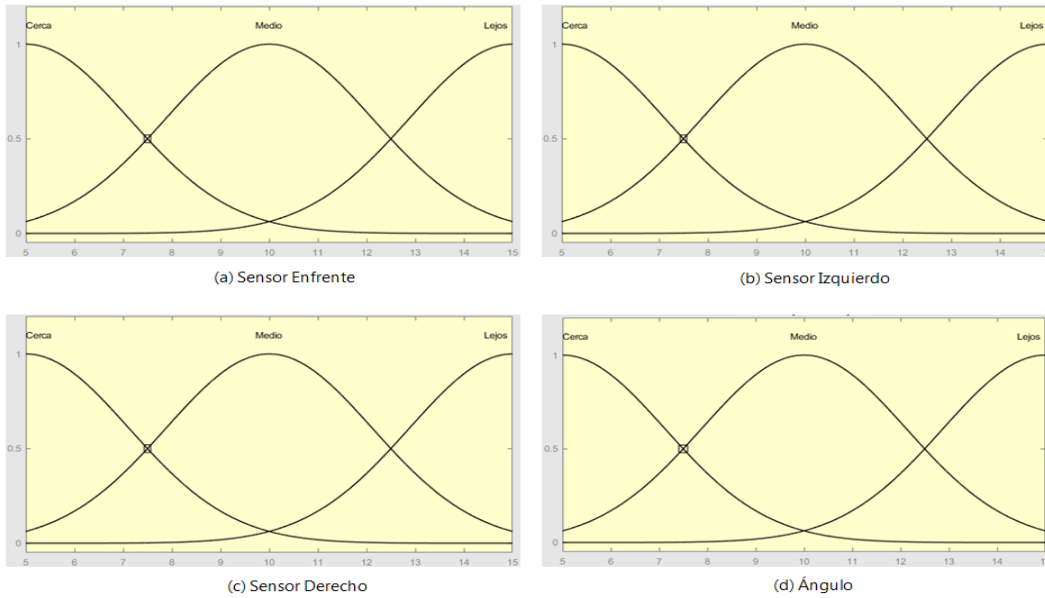


Ilustración 25-Funciones de membresía gaussiana antes del entrenamiento

Fuente: Elaboración Propia

A través del entrenamiento del modelo ANFIS el menor RMSE obtenido fue de 0.69115. Fueron evaluadas 10 mil épocas para el entrenamiento, tal como se muestra en la ilustración 26.

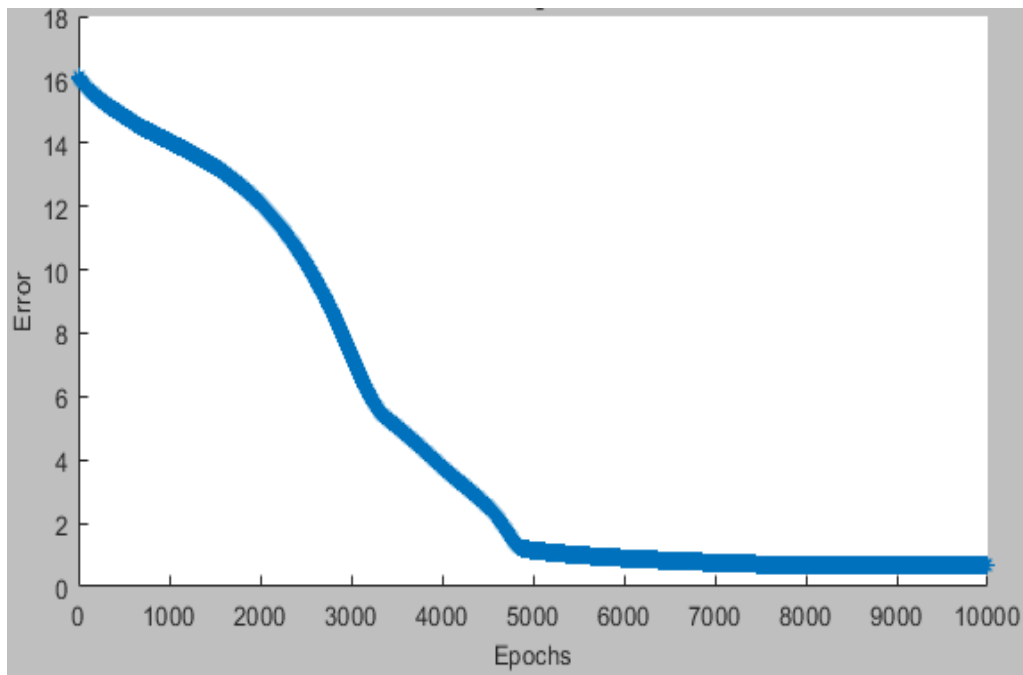


Ilustración 26-RMSE de ANFIS-1 gaussiana/constante

Fuente: Elaboración Propia

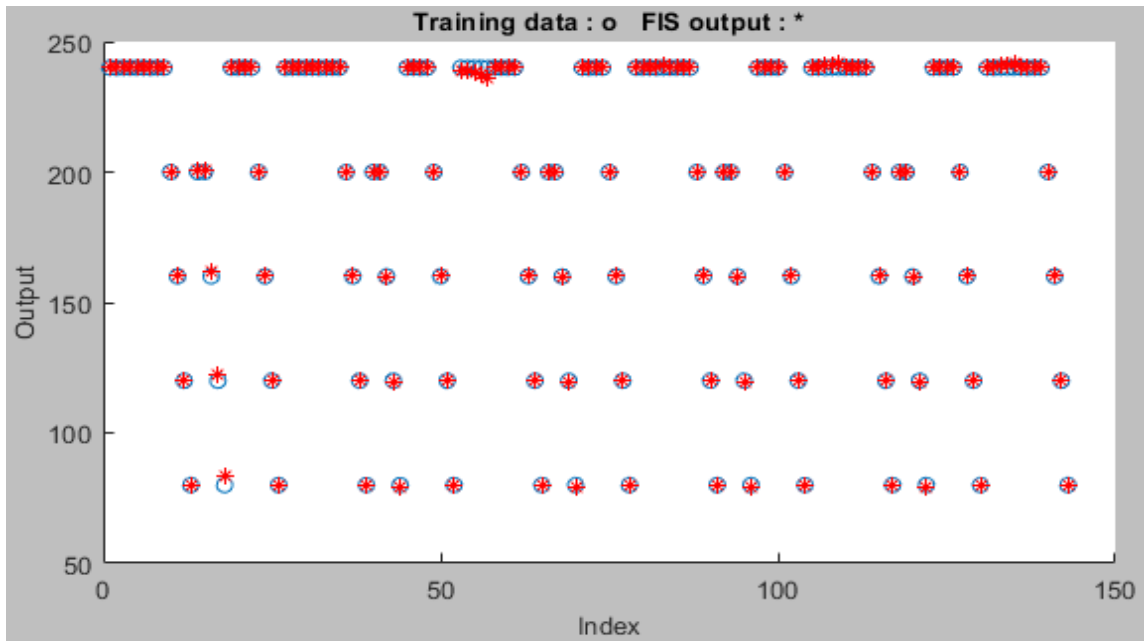


Ilustración 27-Comparación con datos de entrenamiento de ANFIS-1 gaussiana/constante

Fuente: Elaboración Propia

En la ilustración 27, se muestra la comparación de la salida de la base de datos de entrenamiento respecto a los valores predichos por el modelo FIS optimizado final.

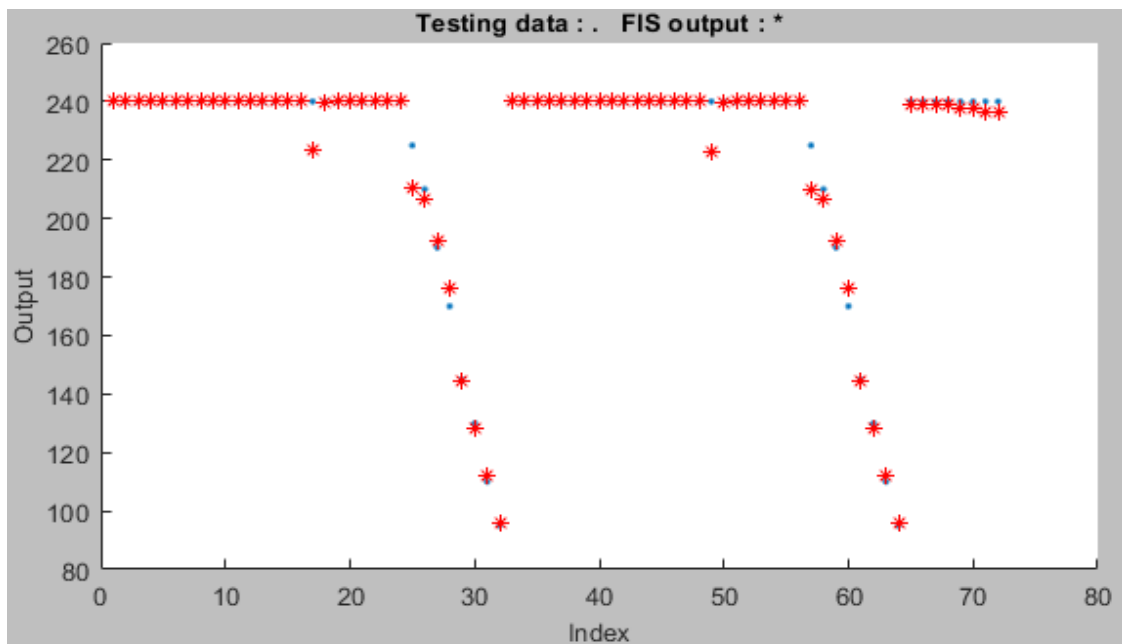


Ilustración 28-Comparación con datos de prueba de ANFIS-1 gaussiana/constante

Fuente: Elaboración Propia

Con respecto a los datos de prueba, el modelo FIS optimizado predijo los valores de manera adecuada como se muestra en la ilustración 28. De hecho, con esta base de datos, se obtuvo un RMSE mínimo de prueba de 4.0618. Menor en comparación al RMSE de la base de datos de prueba de los modelos ANFIS desarrollados con anterioridad.

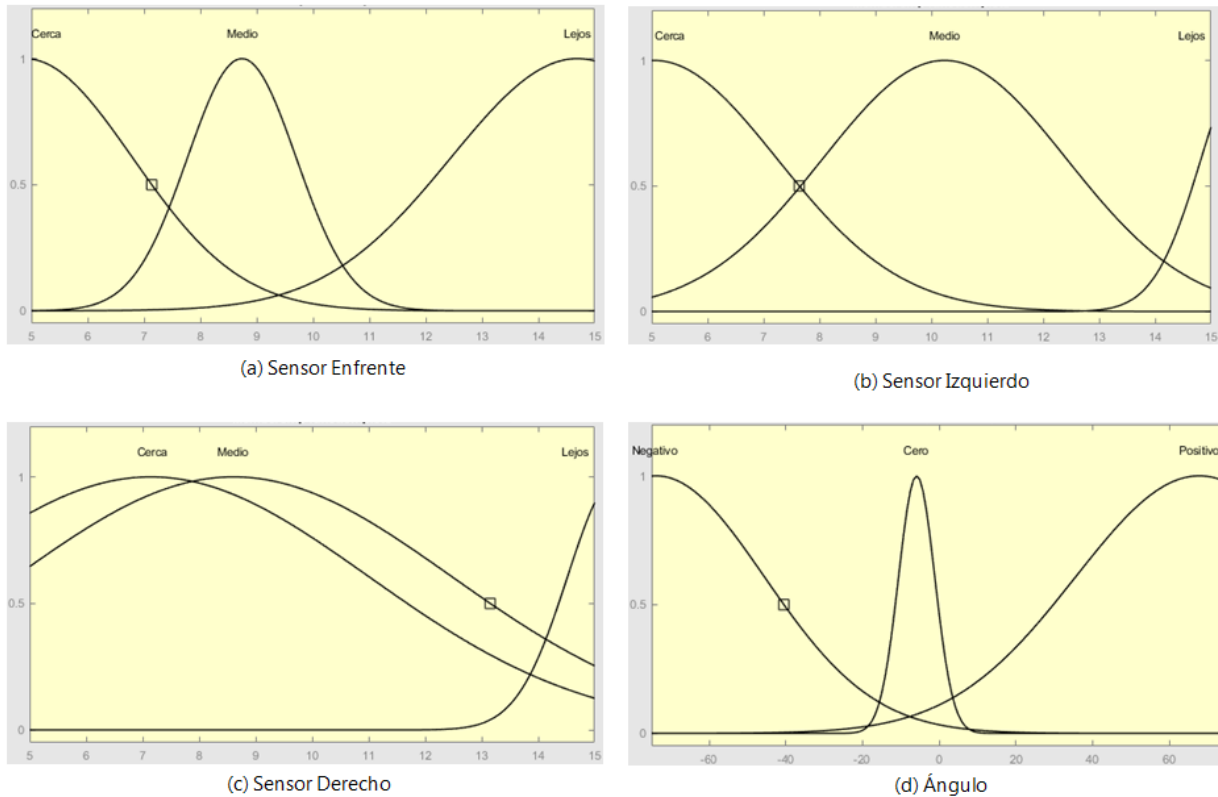


Ilustración 29-FMs gaussianas después del entrenamiento con salida constante

Fuente: Elaboración Propia

Finalmente, se obtuvo las funciones de membresía finales después del entrenamiento con sus respectivos valores de premisa ajustados. Cada una de los parámetros de dichas funciones de membresía fueron ajustados de acuerdo a los datos de entrenamiento para obtener el menor error posible. En la ilustración 29, se muestra la gráfica de cada una de las funciones de membresía de cada entrada del modelo FIS.

5.3.4 ANFIS-1 GAUSSIANA/LINEAL

El cuarto y último modelo a desarrollar para el ANFIS-1 se utilizan funciones de membresía gaussiana (ver ecuación 7) y salida lineal. En consecuencia, para el presente modelo ANFIS, el algoritmo de aprendizaje híbrido debe de ajustar un total de 429. Siendo mucho mayor en comparación de los modelos con salida constante.

Fueron inicializadas las funciones de membresía gaussiana igual que el modelo anterior con las FM mostradas en la ilustración 25. A partir de esto, en conjunto con la base de datos de entrenamiento, se procedió al ajuste de los parámetros con el algoritmo de aprendizaje. En la ilustración 28, se muestra el valor de RMSE con respecto a cada época del entrenamiento del modelo ANFIS.

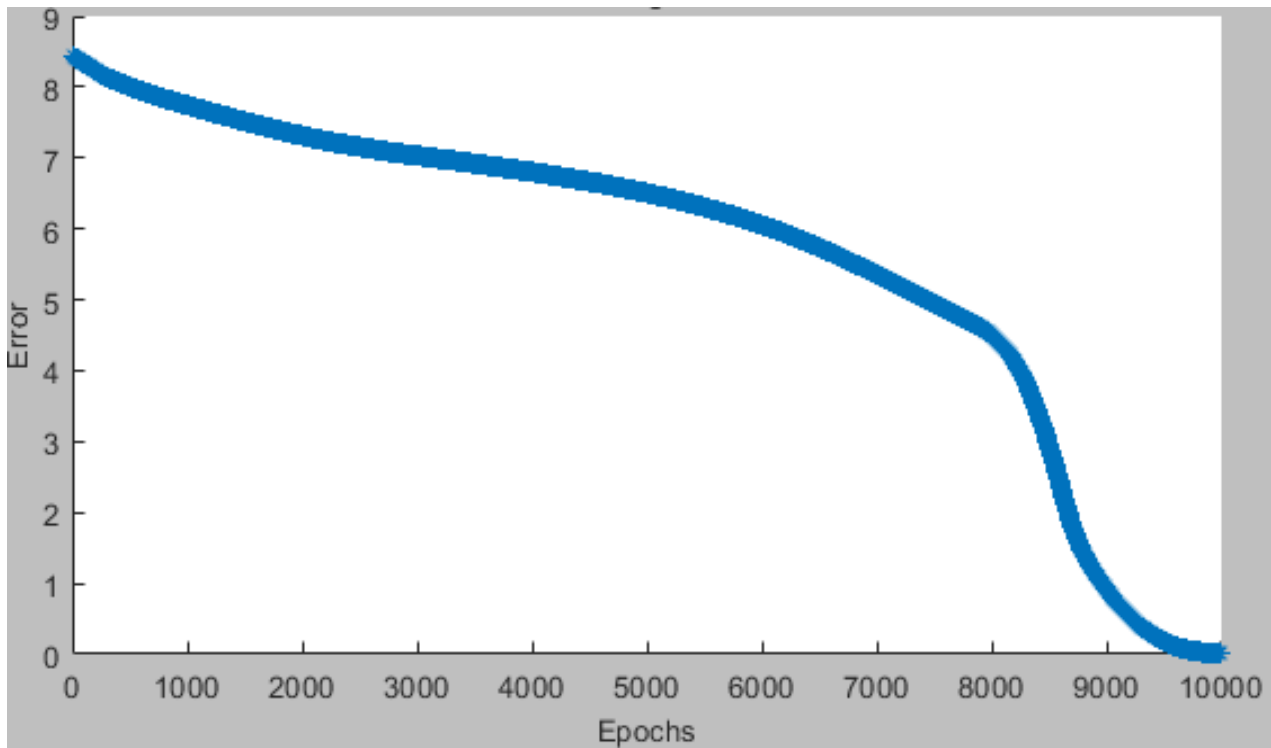


Ilustración 30-RMSE de ANFIS-1 gaussiana/lineal

Fuente: Elaboración Propia

Tal como se muestran en la ilustración 30, el valor de RMSE de entrenamiento mínimo obtenido fue de 0.0070722. Siendo mucho menor en comparación de los modelos anteriores.

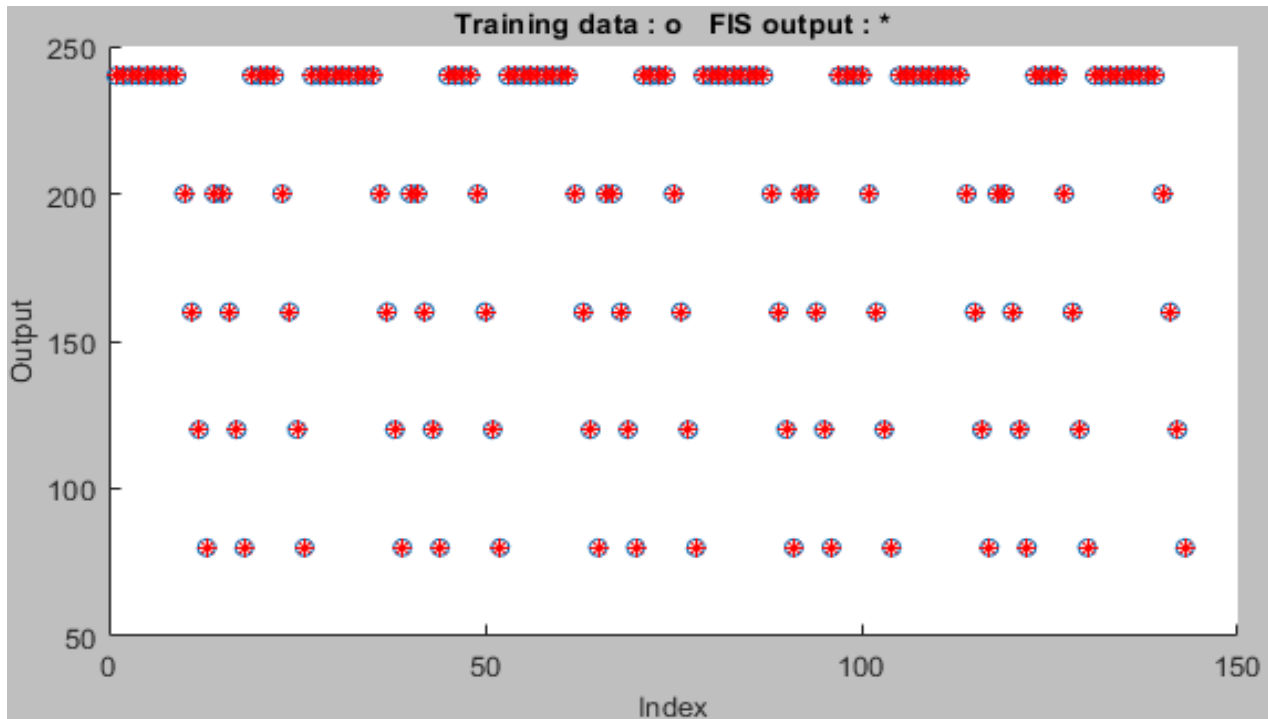


Ilustración 31-Comparación con datos de entrenamiento de ANFIS-1 gaussiana/lineal

Fuente: Elaboración Propia

Cabe destacar que los modelos con funciones de membresía gaussiana obtuvieron un menor RMSE que al utilizar funciones de membresía triangular, tanto al utilizar una salida constante como lineal. En la ilustración 31, se muestra la comparación de los valores predichos por el modelo FIS optimizado, con respecto al valor real de la base de datos de entrenamiento. Se puede observar como cada uno de los valores fueron predichos con una gran precisión.

Similarmente, para comprobar la generalización del modelo, fue utilizada una base de datos de prueba. El modelo fue utilizado para la predicción con estos datos. En la ilustración 32, se muestran los valores de salida obtenido por el modelo FIS optimizado. El RMSE mínimo de prueba obtenido fue de 4.0618. Dicho valor es mucho mayor en comparación del RMSE de entrenamiento. Esto da indicio a que el modelo optimizado muestra señales de *overfitting*, haciendo referencia a que solamente aprendió de "memoria" los valores de la base de entrenamiento. Sin embargo, a pesar de ello, aún sigue siendo un RMSE menor que algunos de los otros modelos.

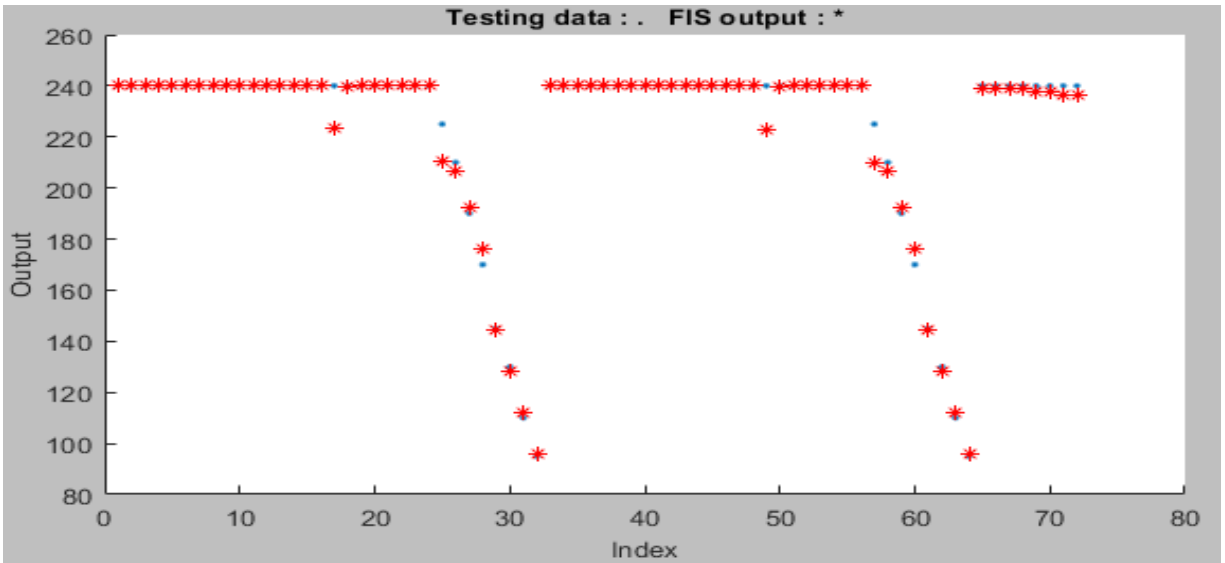


Ilustración 32-Comparación con datos de prueba de ANFIS-1 gaussiana/lineal

Fuente: Elaboración Propia

En la ilustración 33, se muestran las funciones de membresía gaussianas después del entrenamiento del modelo ANFIS.

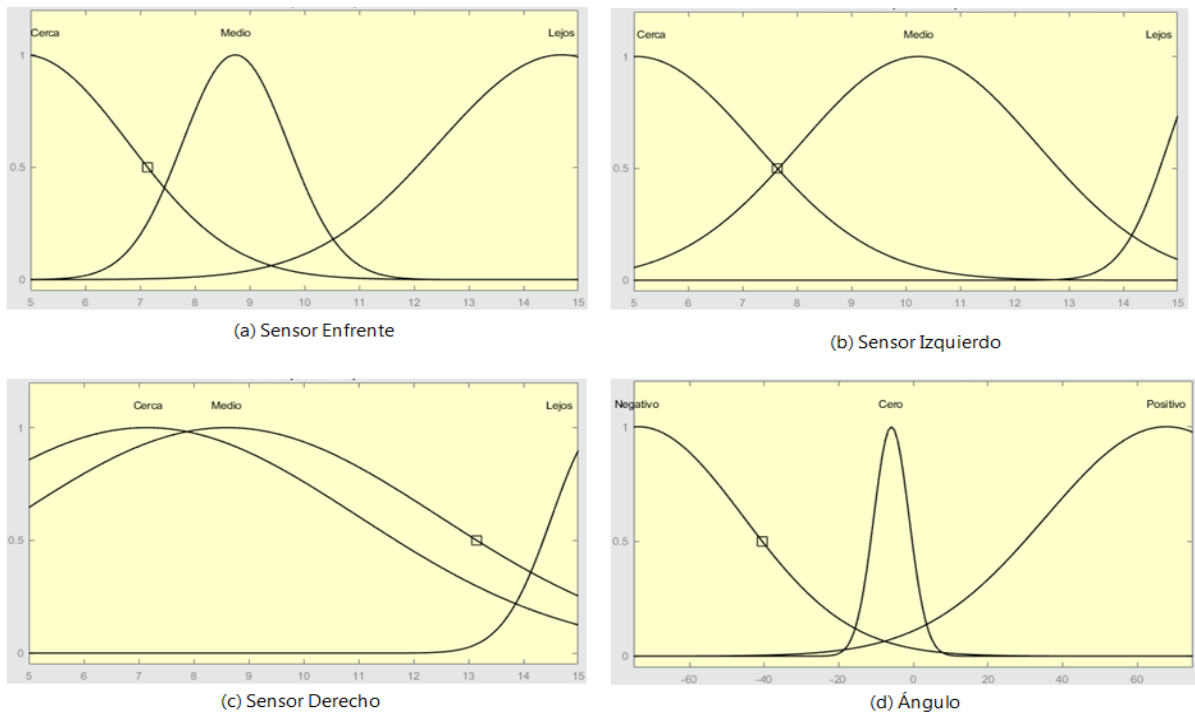


Ilustración 33-FM gaussiana después del entrenamiento con salida lineal

Fuente: Elaboración Propia

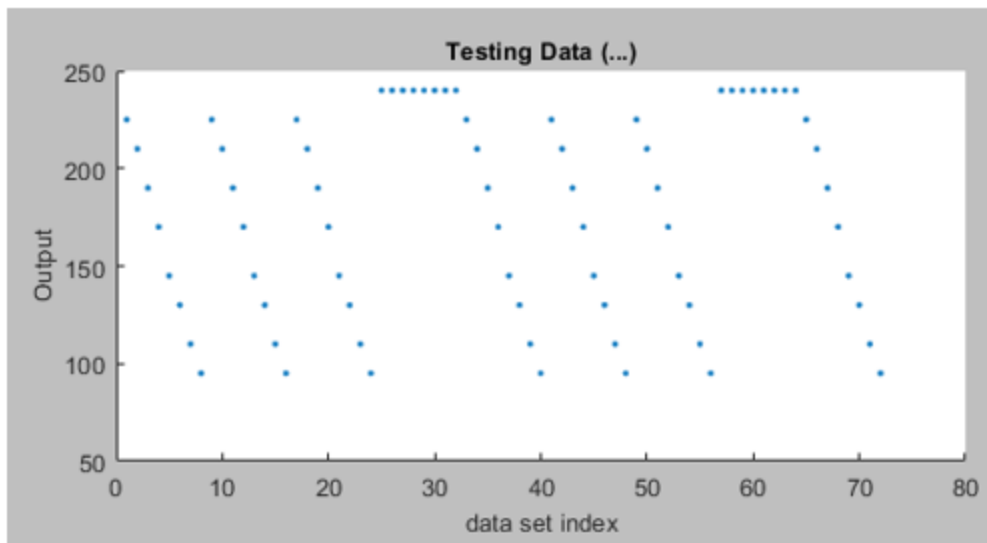
5.4 ENTRENAMIENTO DE ANFIS-2

En esta sección se muestran los resultados obtenidos del entrenamiento de la ANFIS-2 que se encarga del control sobre el motor ubicado en la derecha, con la salida denominada como PWMD. Similarmente, para obtener el mejor modelo que describa la salida a controlar, fueron desarrollados 4 modelos ANFIS variando dos parámetros de su configuración, la función de membresía y el tipo de salida. Se varió el tipo de función de membresía de entrada, que influye en la cantidad de parámetros de premisa o parámetros no lineales, se selecciona entre una función de membresía triangular (ecuación 5) y una función gaussiana (ecuación 7). Y el segundo parámetro a modificar es el tipo de salida ya sea lineal (ecuación 16) o constante (ecuación 17), esto influye significativamente en la cantidad de parámetros consecuentes que el algoritmo híbrido de aprendizaje debe de ajustar.

Inicialmente, se establecieron la base de datos de entrenamiento en dos partes. Una de las partes es para el entrenamiento en sí, y la otra parte es para la prueba de la red. Para el desarrollo del ANFIS-1, se partió de los datos de entrada y solamente una salida, PWMD, ver tabla 5. En la ilustración 34a se muestra los datos para entrenamiento, donde se muestra el valor de salida con respecto a cada registro de las 4 entradas (*data set index*), que represente el índice de cada registro de la base de datos. En la ilustración 34b se muestra la distribución de los datos de prueba. Ambas bases de datos mostradas en la ilustración 34, fueron utilizadas para el entrenamiento de los 4 diferentes modelos ANFIS, variando sus parámetros.



(a)



(b)

Ilustración 34-Distribución de datos de entrenamiento para ANFIS-2

Fuente: Elaboración Propia

5.4.1 ANFIS-2 TRIANGULAR/CONSTANTE

Para el primer modelo a entrenar del ANFIS-2 utiliza funciones de membresía triangulares y con salida de tipo constante. Las entrenadas del ANFIS-2 son las mismas que retroalimentan al modelo ANFIS-1. Por lo tanto, para la inicialización de las funciones de membresía fueron las mismas presentadas en la ilustración 16.

A partir de la base de datos de entrenamiento la red ANFIS fue ajustando sus parámetros de premisa y consecuentes cada época. En la ilustración 35, se detalla el valor del error respecto a cada época de entrenamiento de la red ANFIS. Para obtener la convergencia del entrenamiento y obtener el menor RMSE, se requirió menos de 10 mil épocas. El RMSE de entrenamiento mínimo obtenido fue de 3.1588.

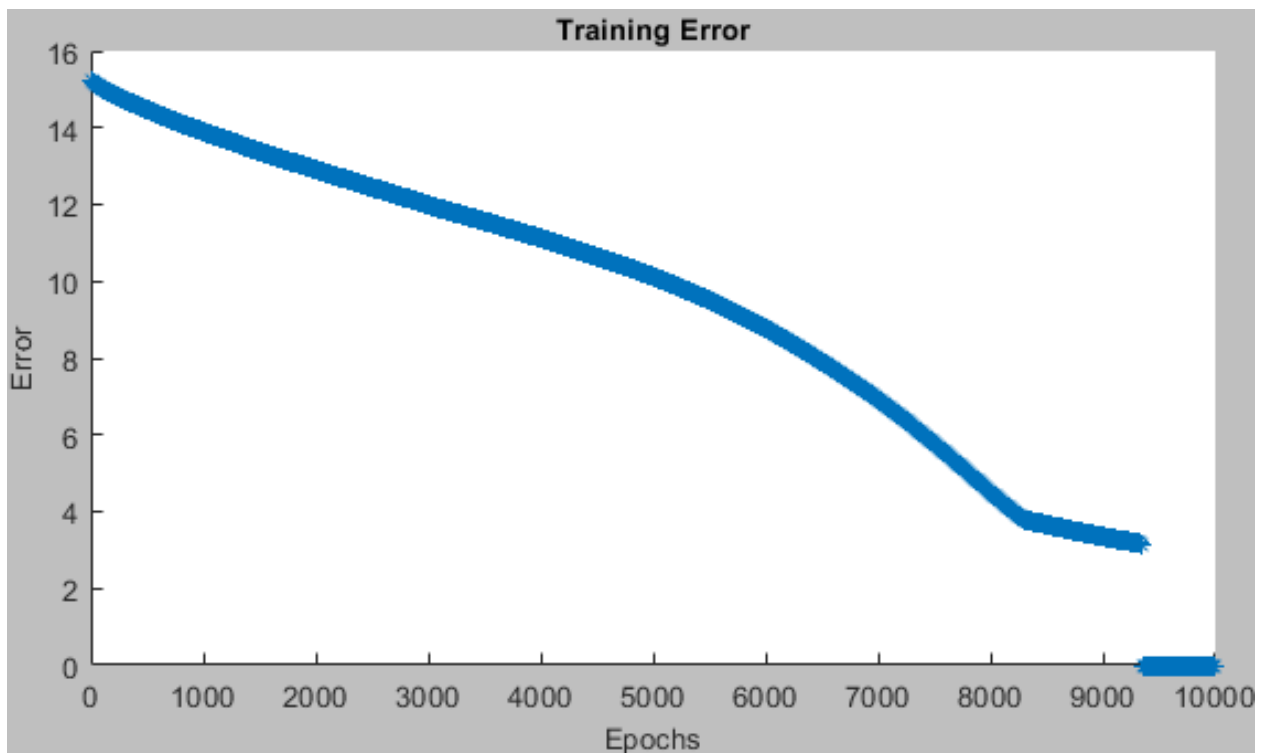


Ilustración 35-RMSE de ANFIS-2 triangular/constante

Fuente: Elaboración Propia

Se comprobó el modelo ANFIS desarrollado a través de la comparación de los datos de salida de la base de datos de entrenamiento, con respecto a los valores predichos por el modelo desarrollado. En la ilustración 36, se muestran los resultados obtenidos, respecto a la salida de los datos de entrenamiento respecto a los del modelo ANFIS. De igual manera, fue utilizada una base de datos de prueba para la verificación del modelo FIS optimizado.

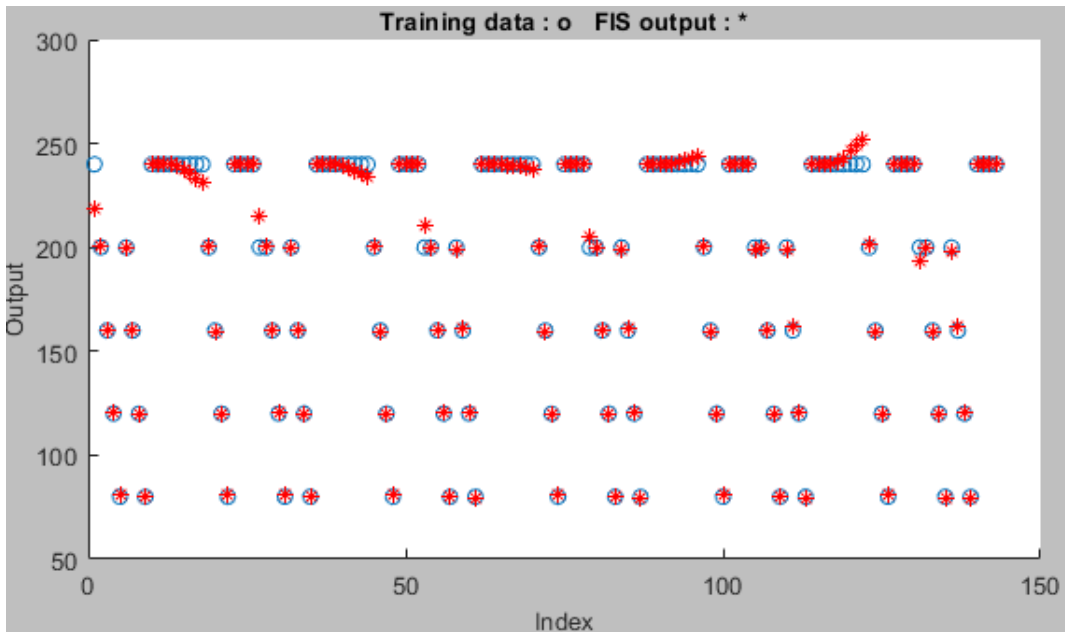


Ilustración 36-Comparación con datos de entrenamiento de ANFIS-2 triangular/constante

Fuente: Elaboración Propia

En la ilustración 37, se presenta la comparación de los valores reales de salida del PWMMD respecto a los obtenidos con el ANFIS con la base de datos de prueba.

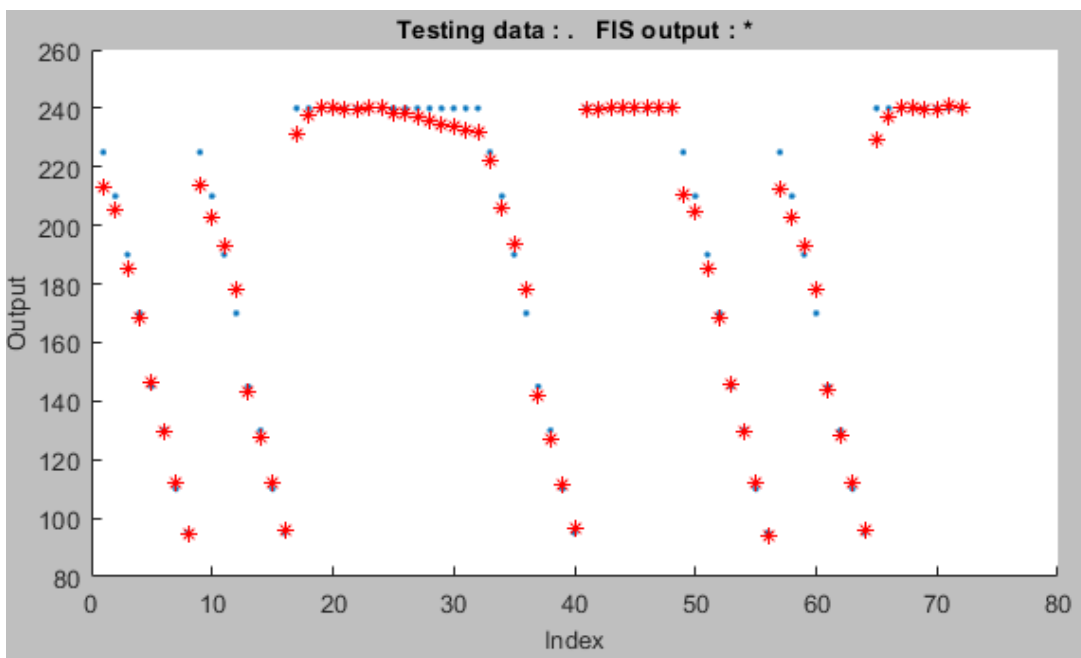


Ilustración 37-Comparación con datos de prueba de ANFIS-2 triangular/constante

Fuente: Elaboración Propia

Tal como se observa en la ilustración 37 la mayoría de los valores de la salida coinciden, sin embargo, se encontraron diversos valores que tienen un mayor error. A partir de la base de datos de prueba se obtuvo un RMSE de prueba mínimo de 4.6519. Los parámetros de premisa y consecuentes fueron ajustados época por época para disminuir el error del modelo. Las funciones de membresía triangulares con los parámetros optimizados se muestran en la ilustración 38.

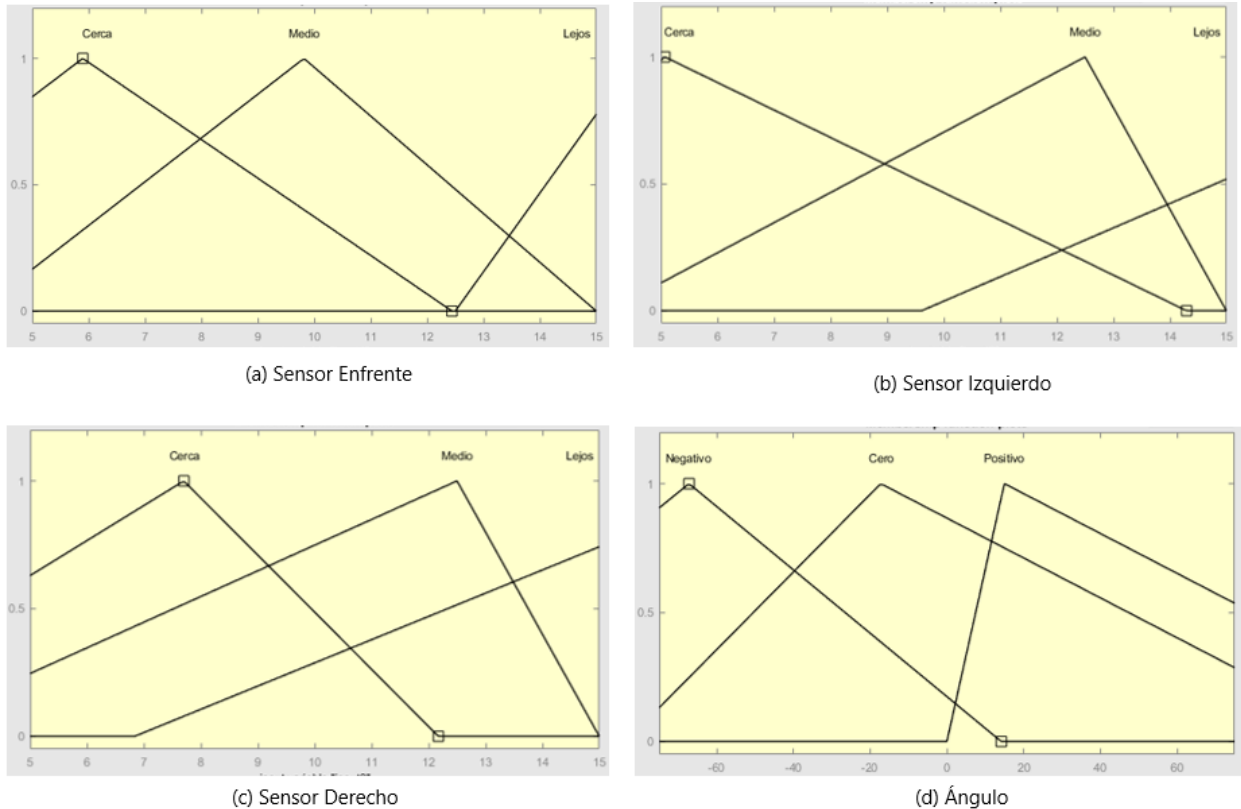


Ilustración 38-FMs triangular después del entrenamiento con salida constante

Fuente: Elaboración Propia

5.4.2 ANFIS-2 TRIANGULAR/LINEAL

El siguiente modelo ANFIS a entrenar cuenta con funciones de membresía triangular y con un tipo de salida lineal. Por lo tanto, los parámetros que el algoritmo de aprendizaje debe de ajustar son de 441, incluyendo parámetros de premisa y consecuentes. Siendo este modelo en que posee más parámetros por ajustar.

Fueron inicializadas las funciones de membresía para el diseño de la arquitectura ANFIS. En la ilustración 16, se muestran las funciones de membresía antes del entrenamiento. Seguidamente, fueron utilizados los datos de entrenamiento, que fueron los mismos que fueron utilizados para el modelo ANFIS anterior, que se parte de los datos mostrados en la ilustración 32. A partir del conjunto de entrenamiento de datos, se realizó el ajuste de parámetros de la red ANFIS. En la ilustración 39, se muestra la gráfica del valor de RMSE respecto a cada época del entrenamiento de la red ANFIS.

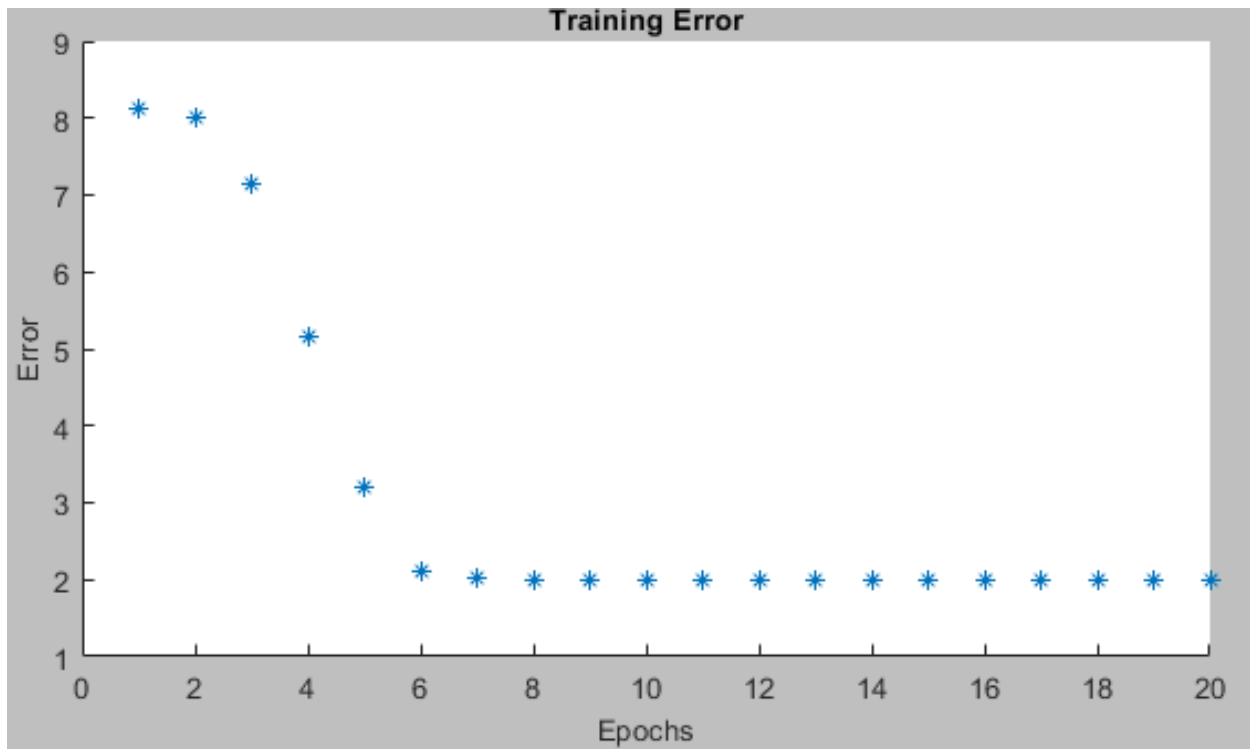


Ilustración 39-RMSE de ANFIS-2 triangular/lineal

Fuente: Elaboración Propia

A causa de más cantidad de parámetros por ajustar, el tiempo de procesamiento de cada época aumentó. Sin embargo, el presente modelo, obtuvo una convergencia u un valor RMSE mínimos en las primeras 7 épocas. El valor de RMSE de entrenamiento mínimo obtenido fue de 1.9826, tal como se muestra en la ilustración 39. La precisión de cada salida respecto al de entrenamiento se presenta en la ilustración 40.



Ilustración 40-Comparación con datos de entrenamiento de ANFIS-2 triangular/lineal

Fuente: Elaboración Propia

De igual manera, se utilizó la base de datos de prueba para la verificación del modelo entrenado, en la ilustración 41, se presenta la comparación.

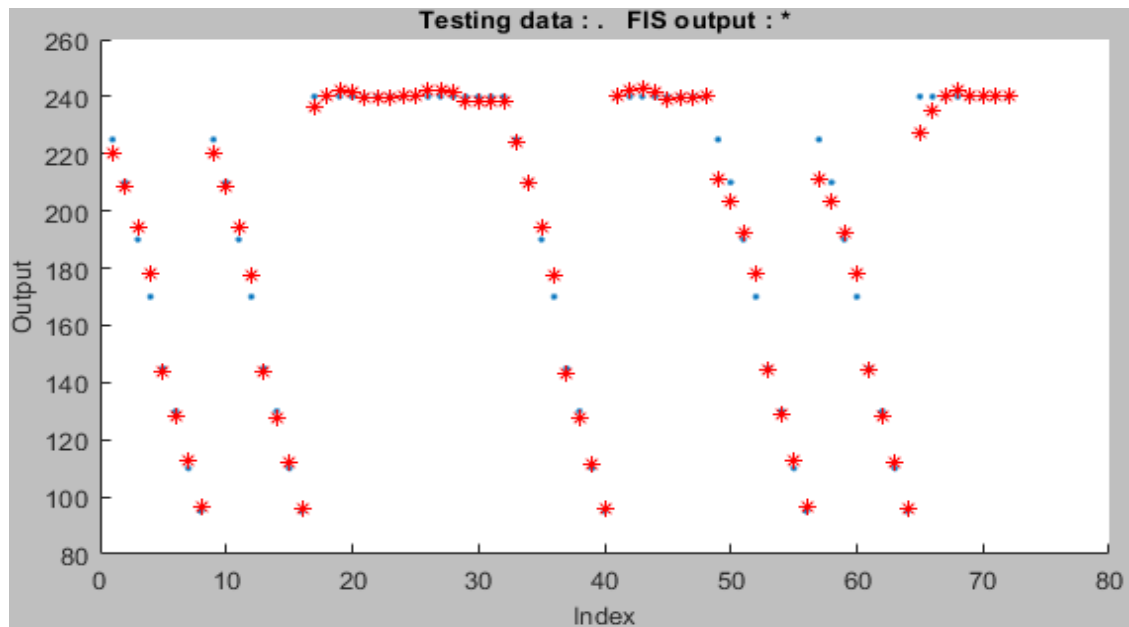


Ilustración 41-Comparación con datos de prueba de ANFIS-2 triangular/lineal

Fuente: Elaboración Propia

De acuerdo con los resultados mostrados en la ilustración 42, el modelo ANFIS aprendió a generalizar en más casos, ya que ha acertado con gran precisión muchos de las salidas de la base de datos de prueba. De hecho, se obtuvo un RMSE de prueba de 4.1054. El algoritmo de aprendizaje híbrido fue el encargado de ajustar los parámetros consecuentes y de premisa de cada una de las funciones de membresía de cada entrada del sistema. En la ilustración 40, se muestran las funciones de membresía después del entrenamiento.

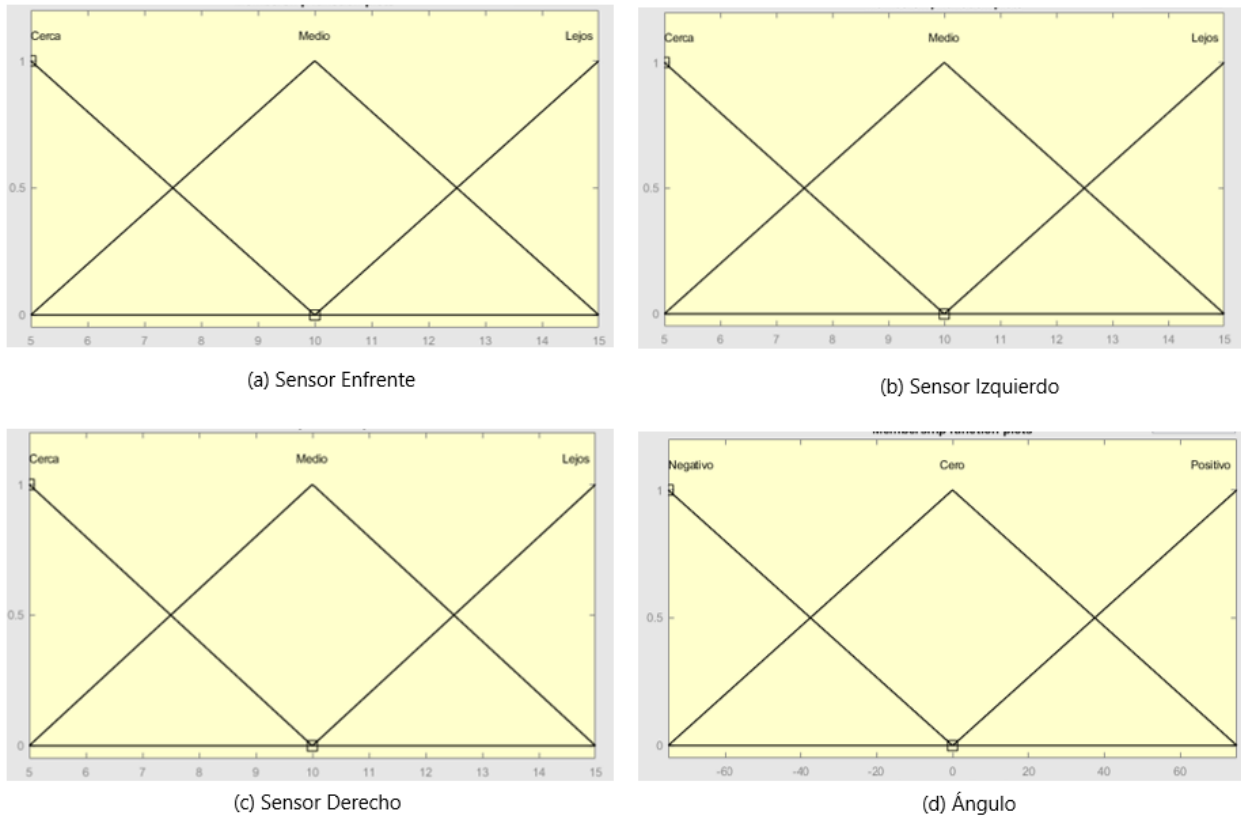


Ilustración 42-FMs triangular después del entrenamiento con salida lineal

Fuente: Elaboración Propia

5.4.3 ANFIS-2 GAUSSIANA/CONSTANTE

El tercer modelo ANFIS a entrenar cuenta con funciones de membresía gaussiana (ecuación 7) y con un tipo de salida constante. Por lo tanto, los parámetros que el algoritmo de aprendizaje que deben ajustar son 105, incluyendo parámetros de premisa y consecuentes. Fueron inicializadas las funciones de membresía gaussiana para el entrenamiento del modelo ANFIS. En

la ilustración 25, se muestran las funciones de membresía gaussiana de cada una de las entradas del ANFIS.

Se establecieron un total de 3 funciones de membresía por cada una de las variables de entrada. Seguidamente el modelo ANFIS con estos parámetros establecidos se realizó se respectivo entrenamiento con la base de datos de entrenamiento. En la ilustración 43, se muestra el progreso del entrenamiento por cada época, donde se observa que el error va decreciendo continuamente, hasta obtener un valor mínimo de RMSE de 0.91771, mucho menor que el RMSE de los modelos con funciones de membresía triangular.

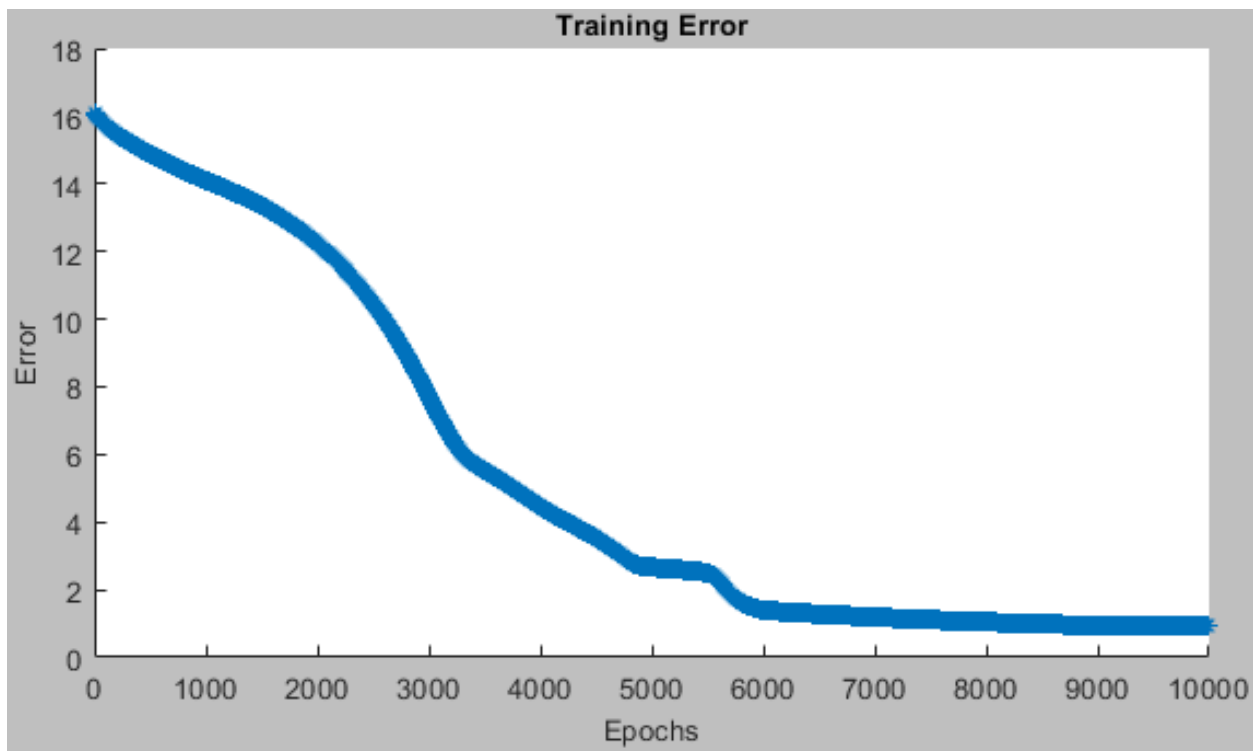


Ilustración 43-RMSE de ANFIS-2 gaussiana/constante

Fuente: Elaboración Propia

Fue comprobada la precisión del modelo ANFIS desarrollado a través de la comparación de los datos de salida de la base de datos de entrenamiento, con respecto a los valores predichos por el modelo desarrollado. En la ilustración 42, se la predicción de la salida por parte del modelo ANFIS entrenado.



Ilustración 44-Comparación con datos de entrenamiento de ANFIS-2 gaussiana/constante

Fuente: Elaboración Propia

Similarmente, se utilizó la base de datos de prueba para la verificación del modelo ANFIS desarrollado. En la ilustración 45, se muestra las predicciones del modelo.

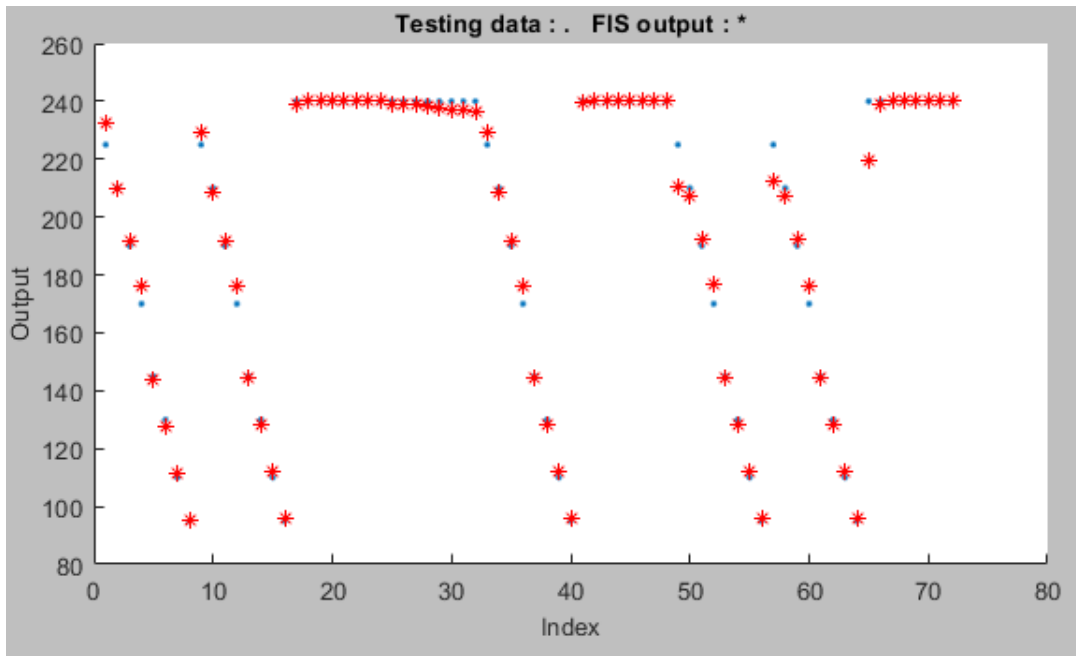


Ilustración 45-Comparación con datos de prueba de ANFIS-2 gaussiana/constante

Fuente: Elaboración Propia

En base a las predicciones realizadas por el modelo ANFIS entrenado, tanto para la base de datos de entrenamiento (ver ilustración 44) como la base de datos de prueba (ver ilustración 45) se puede observar que el modelo presenta una alta precisión. Específicamente, se obtuvo un RMSE de prueba de 4.0632. Los parámetros de premisa y consecuentes del FIS fueron ajustados. Las funciones de membresía gaussianas de cada una de las entradas del modelo después del entrenamiento se presentan en la ilustración 46.

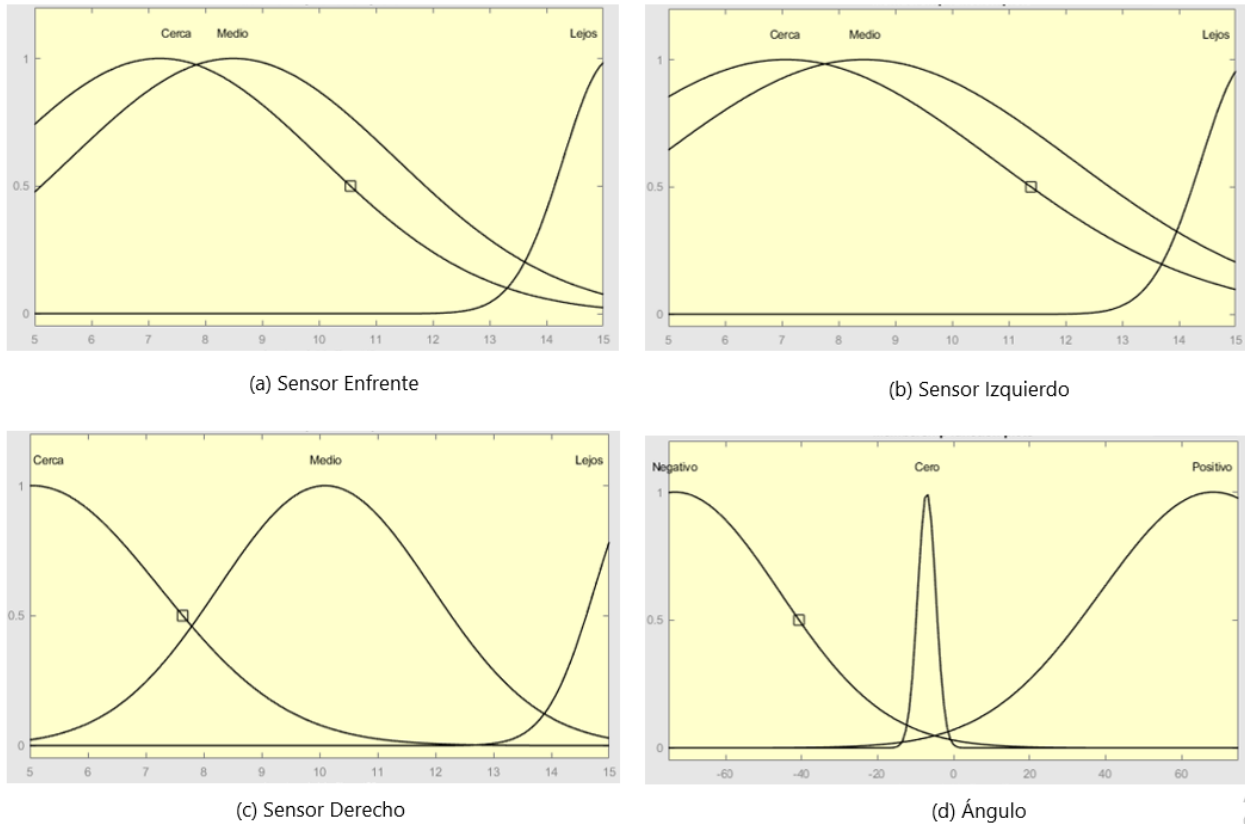


Ilustración 46-FMs gaussianas después del entrenamiento con salida constante

Fuente: Elaboración Propia

5.4.4 ANFIS-1 GAUSSIANA/LINEAL

El último modelo desarrollado para el ANFIS-2 se utilizan funciones de membresía gaussianas (ver ecuación 7) y salida lineal. Por lo tanto, para el presente modelo ANFIS, el algoritmo de aprendizaje híbrido debe de ajustar un total de 429.

Fueron inicializadas las funciones de membresía gaussiana similar que el modelo anterior con ver ilustración 25. A partir de esto, en conjunto con la base de datos de entrenamiento ver ilustración 34, se procedió al ajuste de los parámetros con el algoritmo de aprendizaje. En la ilustración 47, se muestra el progreso del entrenamiento de la arquitectura ANFIS y cómo se comporta el error respecto a cada época.

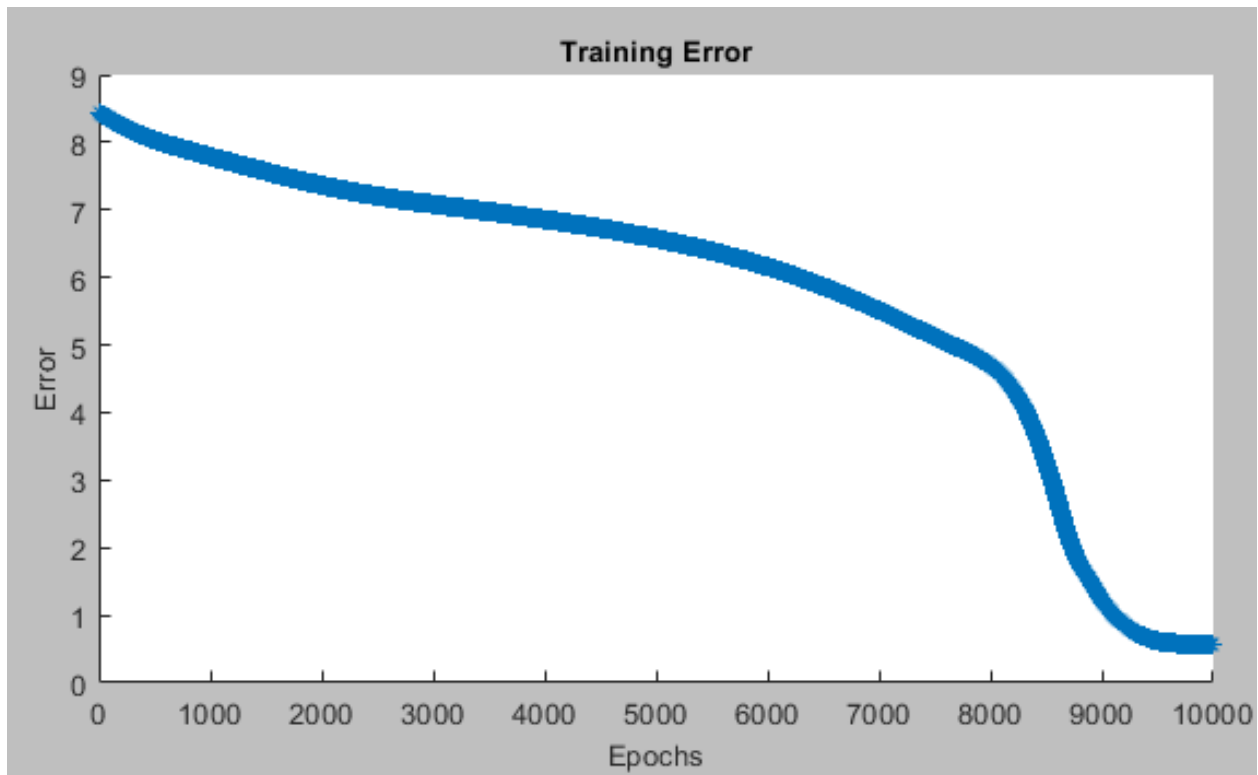


Ilustración 47-RMSE de ANFIS-2 gaussiana/lineal

Fuente: Elaboración Propia

Tal como se presenta en la ilustración 47, el RMSE va disminuyendo con cada número de época. Para el presente modelo, el RMSE de entrenamiento obtenido fue de 0.57116. Fue necesario un total de 10 mil épocas para obtener el menor RMSE. De igual manera, el modelo fue puesta a prueba y se utilizó para la predicción de la salida PWM con la base de datos de entrenamiento, los resultados se muestran en la ilustración 48.

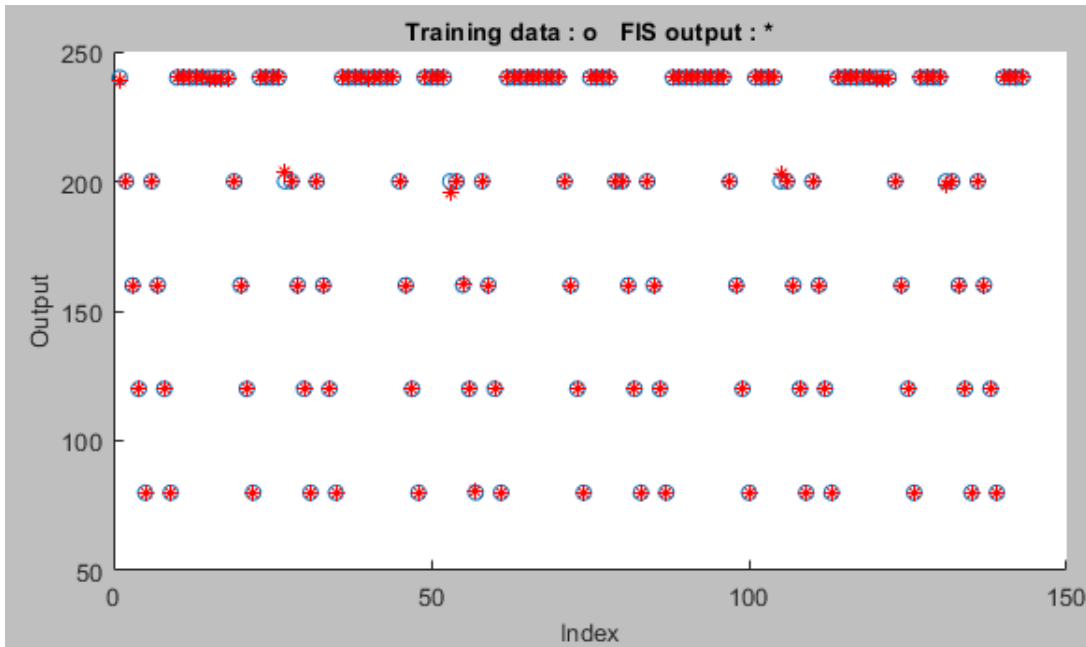


Ilustración 48-Comparación con datos de entrenamiento de ANFIS-2 gaussiana/lineal

Fuente: Elaboración Propia

Posteriormente, el modelo ANFIS fue utilizado para la predicción de la salida con la base de datos de pruebas, en la ilustración 49, se muestran los resultados obtenidos.

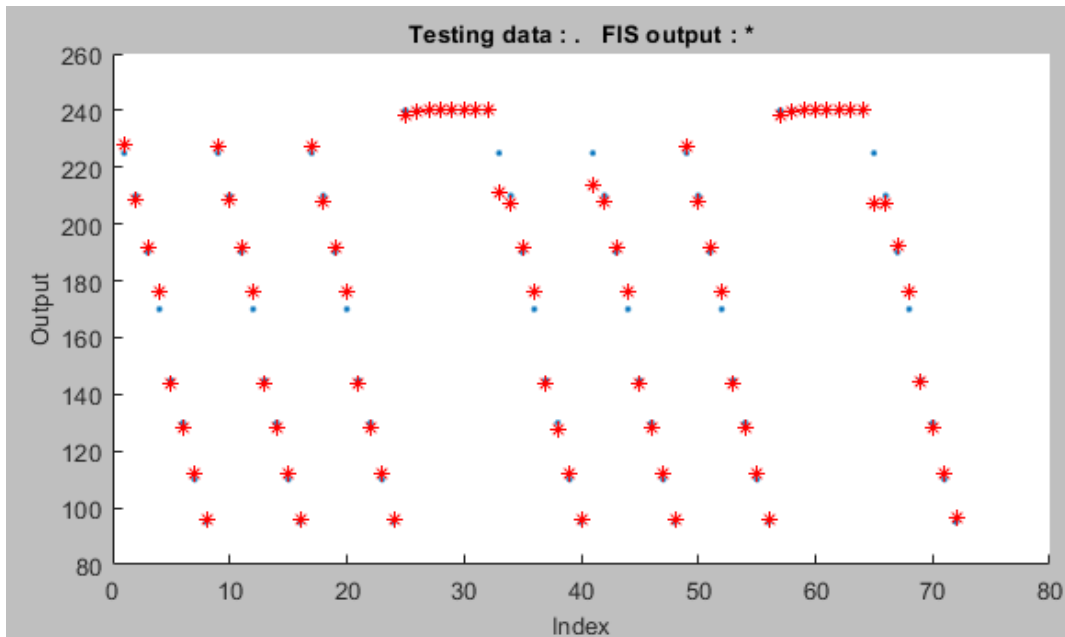


Ilustración 49-Comparación con datos de prueba de ANFIS-2 gaussiana/lineal

Fuente: Elaboración Propia

El modelo ANFIS entrenado presentó buenos resultados y precisión en la predicción de las salidas tanto para la base de datos de entrenamiento como de prueba. El RMSE de prueba obtenido fue de 3.8409. Las funciones de membresía gaussianas después del entrenamiento se muestran en la ilustración 50.

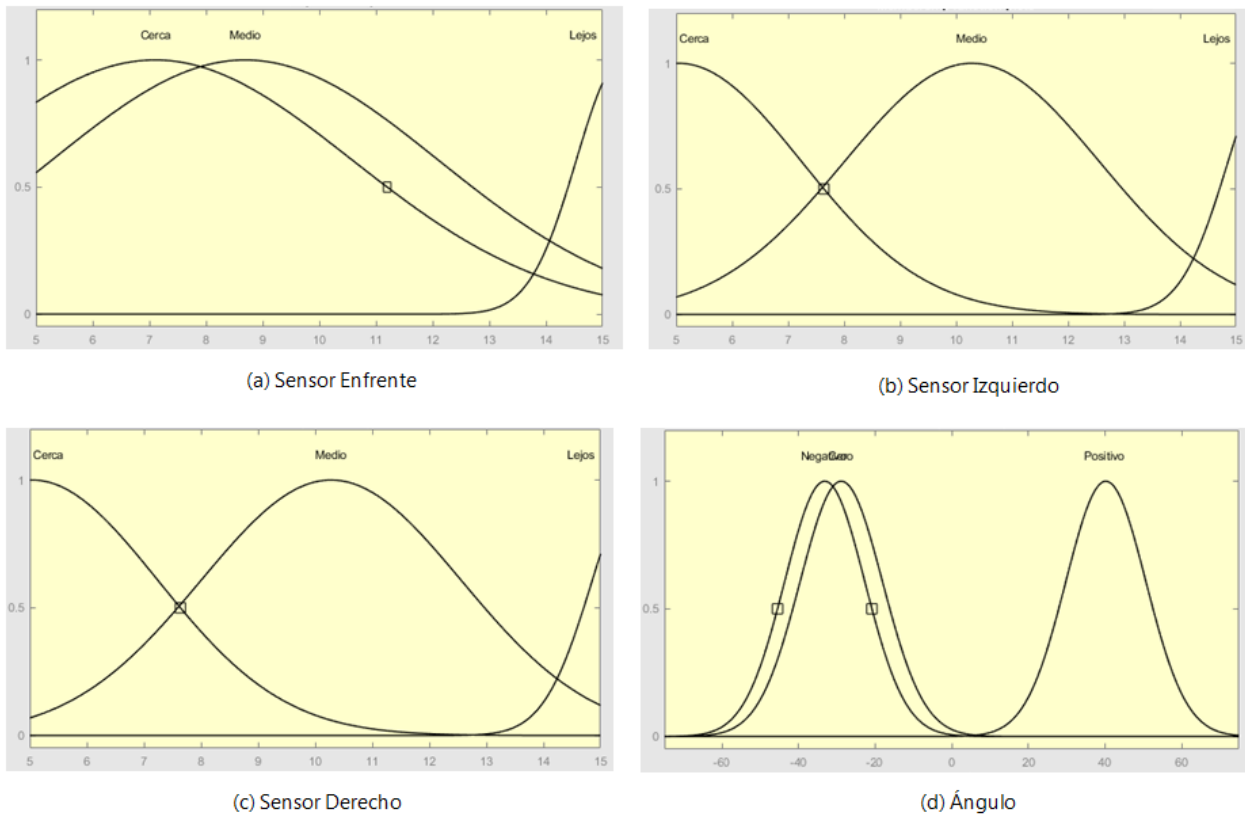


Ilustración 50-FMs gaussianas después del entrenamiento con salida lineal

Fuente: Elaboración Propia

5.5 DISCUSIÓN

Fue desarrollado un sistema de navegación basado en la arquitectura ANFIS. Esta permite múltiples entradas, sin embargo, solamente una salida. En consecuencia, fueron desarrollados dos modelos ANFIS, uno para cada salida que representa el valor PWM de un motor. Con la finalidad de obtener el mejor resultado, fueron desarrollados diferentes modelos ANFIS para cada motor con diferentes funciones de membresía (triangular y gaussiana) y con diferente tipo de salida (constante y lineal). Se determinó tanto el RMSE de entrenamiento como de prueba de

cada uno de ellos. En la tabla 8, se resumen los resultados obtenidos para el ANFIS-1 que controla la salida PWMI.

Tabla 8-Resumen de los resultados para cada modelo de ANFIS-1

Parámetros	Triangular/Constante	Triangular/Lineal	Gaussiana/Constante	Gaussiana/Lineal
RMSE de los datos de Entrenamiento	2.2354	1.865	0.69115	0.0070722
RMSE de los datos de prueba	4.5953	5.4151	4.0618	4.7471
Épocas	9136	6	7600	10000
No. De Parámetros	117	441	105	429

Fuente: Elaboración Propia

Tal como se muestra en los resultados de la tabla 8, el modelo ANFIS-1 con mayor RMSE de entrenamiento es el triangular/constante. Con una bajo número de parámetros requirió de más de 9 mil épocas para convergir. El algoritmo logró generalizar ya que el RMSE de prueba no tuvo mucha diferencia en comparación con el RMSE de entrenamiento. Por otro lado, el modelo triangular/lineal presentó un menor valor de RMSE de entrenamiento, sin embargo, el de prueba fue mayor. Esto a pesar que es el modelo con un mayor número de parámetros requirió menos de 10 épocas para su entrenamiento.

Se pudo observar en los resultados que los modelos con funciones de membresía gaussiana poseen un mejor rendimiento en comparación con los modelos que cuentan con funciones de membresía triangular. De hecho, el modelo gaussiana/lineal presentó el menor RMSE de entrenamiento. No obstante, el RMSE de prueba, demuestra un RMSE mucho mayor. Esto da referencia a que este modelo presentó *overfitting*. Por lo tanto, este es un modelo que no generalizó los datos solamente "aprendió de memoria" los datos de entrenamiento. Esto no es muy conveniente para momentos en que el algoritmo se encuentra con situaciones no esperadas. Por otro lado, el modelo gaussiana/constante es el que presentó el mejor rendimiento, con un menor RMSE tanto de entrenamiento como de prueba. Por esta razón, se determinó que el modelo óptimo para el control de la salida PWMI posee funciones de membresía gaussiana con salidas de tipo constante. Cabe recalcar que a pesar del menor

número de parámetros, se requirió más de 7 mil épocas para convergir y obtener el menor RMSE.

Tabla 9-Resumen de los resultados para cada modelo de ANFIS-2

Parámetros	Triangular/Constante	Triangular/Lineal	Gaussiana/Constante	Gaussiana/Lineal
RMSE de los datos de Entrenamiento	3.1588	1.9826	0.91771	0.57116
RMSE de los datos de prueba	4.6519	4.1054	4.0632	3.8409
Épocas	9136	6	7600	10000
No. De Parámetros	117	441	105	429

Fuente: Elaboración Propia

Por otro lado, los resultados del entrenamiento de los diferentes modelos de ANFIS-2 fueron similares, como se presenta en la tabla 9. Los modelos con funciones de membresía gaussiana poseen un mejor rendimiento en comparación con los modelos con funciones de membresía triangular. Similarmente, los modelos con salida lineal presentaron mejor resultados en contraste con la salida tipo constante. Esto indica, que a mayor número de parámetros, se le permite al algoritmo de aprendizaje en mejor ajuste entre los valores de entrada con respecto a los de salida. No obstante, a un mayor número de parámetros los recursos computacionales para el procesamiento del algoritmo incrementan considerablemente.

El modelo triangular/constante es el que presentó un RMSE de entrenamiento y de prueba mayor. El modelo triangular/lineal presentó un menor RMSE de entrenamiento y de prueba. El modelo gaussiana/constante obtuvo un RMSE de entrenamiento muy bajo, por lo tanto, fue ajustado adecuadamente a los datos de entrenamiento. Por otro lado, el modelo gaussiana/lineal es el que presentó un menor RMSE tanto de entrenamiento como de prueba. El RMSE de entrenamiento fue de 0.57116 y el RMSE de prueba fue de 3.8049. Este modelo no presentó *overfitting* como lo presentó el mismo modelo del ANFIS-1. Por lo tanto, se determinó que el modelo gaussiana/lineal es el que presentó los mejores resultados para el control de la salida PWMD.

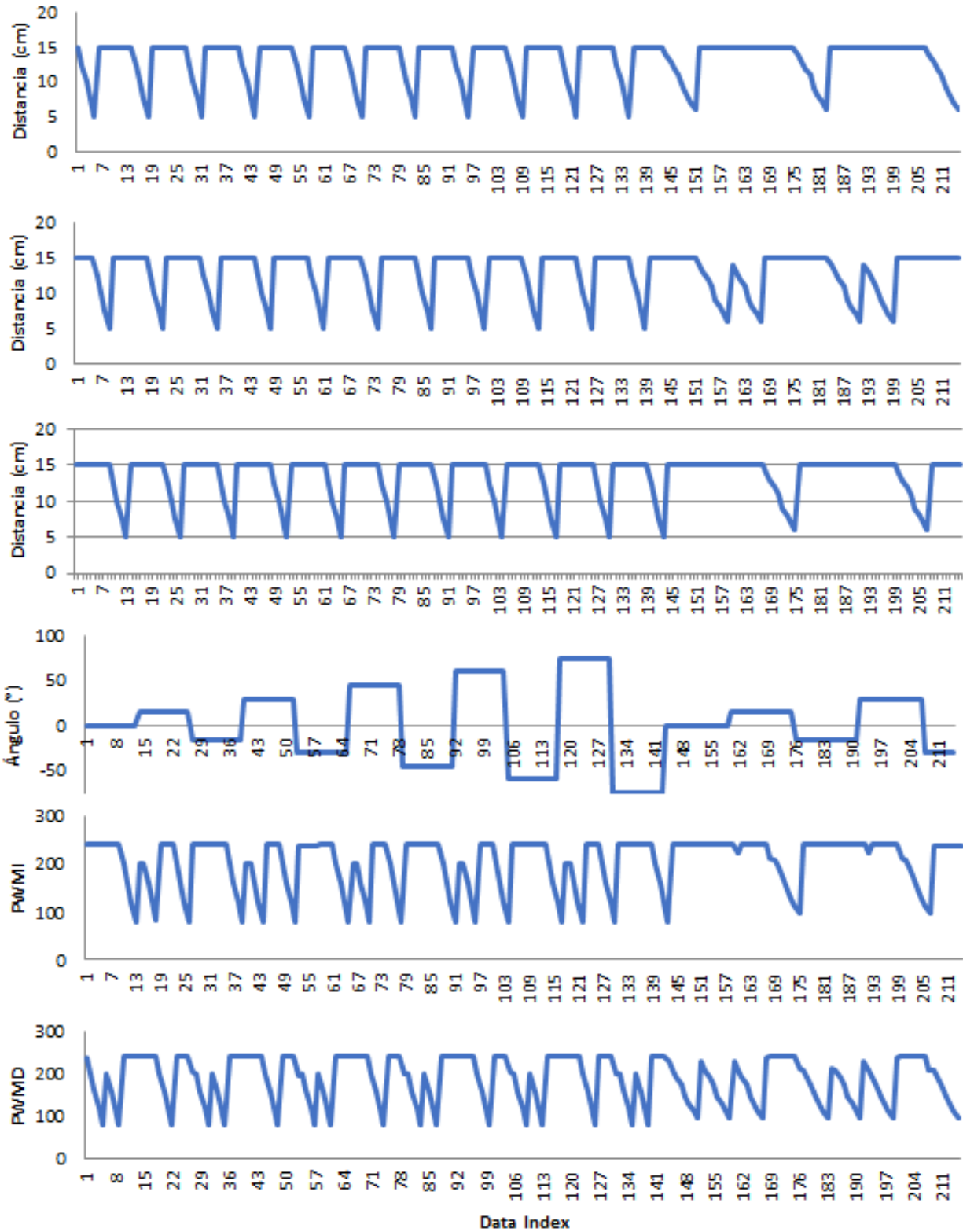


Ilustración 51-Comportamiento del sistema de navegación final

Fuente: Elaboración Propia

Para el sistema final de navegación del robot móvil, fue seleccionado el modelo gaussiana/constante para el ANFIS-1 que controla la salida PWMI. Y para salida PWMD fue seleccionado el modelo gaussiana/lineal del ANFIS-2. En la ilustración 51, se detalla el comportamiento del sistema de navegación final del robot móvil. Donde se indica el valor de cada una de las salidas, en base a cada diversas combinaciones de la 4 entradas del sistema de navegación.

CAPÍTULO VI. CONCLUSIONES

En el presente capítulo se presenta las conclusiones obtenidas las cuales estas basadas en los resultados mostrados en el capítulo anterior.

6.1 CONCLUSIÓN GENERAL

Fue desarrollado un sistema de navegación y evasión de obstáculos para un robot móvil. Se utilizó un sistema neuro-difuso, una arquitectura ANFIS para el sistema de control. Esto permitió el desarrollo de un sistema FIS que fue optimizado a través de los algoritmos de aprendizaje en conjunto con los datos de entrenamiento. Dicha base de datos fue construida a partir de los comportamientos que el robot móvil debe de seguir para su navegación.

6.2 CONCLUSIONES ESPECÍFICAS

- 1) Se diseñó un sistema sensorial de 4 sensores, 3 de distancia infrarrojos y un giroscopio que permite tener una mejor percepción del entorno. Cada uno representa una entrada al sistema de control. El sistema cuenta con dos salidas PWM para el control de cada uno de los motores del robot móvil.
- 2) Fueron desarrollados dos modelos ANFIS, uno para cada salida del sistema de navegación. A partir de diversos comportamientos que debe seguir el robot móvil fue diseñado la base de datos. Para obtener el mejor modelo ANFIS se hicieron pruebas con FM de membresía triangular y gaussiana, y salida de función constante y lineal.
- 3) Tanto la FM como el tipo de salida influyen en el modelo a entrenar. Para el presente diseño, se determinó que los mejores resultados fueron obtenidos por el modelo ANFIS Gaussiana/Constante para el ANFIS-1 y para el ANFIS-2 el modelo gaussiana/lineal presentó los mejores resultados.

CAPÍTULO VII. RECOMENDACIONES

Una vez presentadas las conclusiones de la investigación, se presentan algunas recomendaciones para futuros investigadores.

- 1) Con la finalidad de que el robot móvil posea una mejor percepción del ambiente en que se encuentra, se recomienda la retroalimentación una mayor cantidad de sensores o el uso de un sensor que permita la visualización de obstáculos a lo largo de los 360 grados.
- 2) Se recomienda a futuros investigadores el uso de diferentes algoritmos de optimización para el ajuste de parámetros para el desarrollo de un FIS, como ser algoritmos genéticos o PSO para comparar resultados del comportamiento de estos algoritmos para la navegación y evasión de obstáculos del robot móvil.

BIBLIOGRAFÍA

- Abadi, D. N. M., & Khooban, M. H. (2015). Design of optimal Mamdani-type fuzzy controller for nonholonomic wheeled mobile robots. *Journal of King Saud University - Engineering Sciences*, 27(1), 92-100. <https://doi.org/10.1016/j.jksues.2013.05.003>
- Abiyev, R., Ibrahim, D., & Erin, B. (2010). Navigation of mobile robots in the presence of obstacles. *Advances in Engineering Software*, 41(10), 1179-1186. <https://doi.org/10.1016/j.advengsoft.2010.08.001>
- Adarsh, S., & Ramachandran, K. I. (2018). Design of Sensor Data Fusion Algorithm for Mobile Robot Navigation Using ANFIS and Its Analysis Across the Membership Functions. *Automatic Control and Computer Sciences*, 52(5), 382-391. <https://doi.org/10.3103/S0146411618050036>
- Al-mutib, K., & Abdessemed, F. (2017). Indoor Mobile Robot Navigation in Unknown Environment Using Fuzzy Logic Based Behaviors. *Advances in Science, Technology and Engineering Systems Journal*, 2(3), 327-337.
- Aouf, A., Boussaid, L., & Sakly, A. (2019). Same Fuzzy Logic Controller for Two-Wheeled Mobile Robot Navigation in Strange Environments. *Journal of Robotics*, 2019, 2465219. <https://doi.org/10.1155/2019/2465219>
- Banks, H. T., & Joyner, M. L. (2017). AIC under the framework of least squares estimation. *Applied Mathematics Letters*, 74, 33-45. <https://doi.org/10.1016/j.aml.2017.05.005>
- ByungSoo Ko, Ho-Jin Choi, Hong, C., Kim, J., Oh Chul Kwon, & Yoo, C. D. (2017). Neural network-based autonomous navigation for a homecare mobile robot. *2017 IEEE International*

- Conference on Big Data and Smart Computing (BigComp)*, 403-406.
<https://doi.org/10.1109/BIGCOMP.2017.7881744>
- Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., & Zdeborová, L. (2019). Machine learning and the physical sciences. *Rev. Mod. Phys.*, *91*(4), 045002. <https://doi.org/10.1103/RevModPhys.91.045002>
- Chen, C.-H., Lin, C.-J., Jeng, S.-Y., Lin, H.-Y., & Yu, C.-Y. (2021). Using Ultrasonic Sensors and a Knowledge-Based Neural Fuzzy Controller for Mobile Robot Navigation Control. *Electronics*, *10*(4). <https://doi.org/10.3390/electronics10040466>
- Cohen, A., & Migliorati, G. (2017). Optimal weighted least-squares methods. *The SMAI Journal of Computational Mathematics*, *3*, 181-203. <https://doi.org/10.5802/smai-jcm.24>
- Flores, M. H. B. (2017). *Prerrequisitos para cursar la UDI: Haber cursado las Unidad Didáctica integrada (UDI) de Bioestadística e Informática Unidades didácticas integrales acreditadas previas a cursarla de Metodología de la investigación, Ingles, TICs, bioestadística, Epidemiología, Administración en enfermería*. 8.
- Guzmán, D. & Castaño M. (2006). La Lógica Difusa en Ingeniería: Principios, Aplicaciones y Futuro. *Ciencia y Tecnología*, *24*(2), 87-107.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). *Metodología de la investigación* (5a ed). McGraw-Hill.
- Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated Machine Learning: Methods, Systems, Challenges*. Springer Nature.

- InvenSense. (2013). MPU-6000 and MPU-6050 Product Specification: Revisión 3.4. *InvenSense Inc.*
- Jang, J.-R. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665-685. <https://doi.org/10.1109/21.256541>
- Jaradat, M. A. K., Garibeh, M. H., & Feilat, E. A. (2012). Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field. *Soft Computing*, 16(1), 153-164. <https://doi.org/10.1007/s00500-011-0742-z>
- Kahraman, C., Öztayşi, B., & Onar, S. Ç. (2016). A Comprehensive Literature Review of 50 Years of Fuzzy Set Theory. *International Journal of Computational Intelligence Systems*, 9(sup1), 3-24. <https://doi.org/10.1080/18756891.2016.1180817>
- Karaboga, D., & Kaya, E. (2019). Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey. *Artificial Intelligence Review*, 52(4), 2263-2293. <https://doi.org/10.1007/s10462-017-9610-2>
- Khnessi, K., Seddik, C., & Seddik, H. (2018). Smart Navigation of Mobile Robot Using Neural Network Controller. *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, 205-210. <https://doi.org/10.1109/SaCoNeT.2018.8585616>
- Kruse, R., & Nauck, D. (1998). Neuro-Fuzzy Systems. En O. Kaynak, L. A. Zadeh, B. Türkşen, & I. J. Rudas (Eds.), *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications* (pp. 230-259). Springer Berlin Heidelberg.
- Lee, J. D., Simchowit, M., Jordan, M. I., & Recht, B. (2016). *Gradient Descent Converges to Minimizers*. arXiv:1602.04915.

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Machine Studies*, 7(1).

Markets and Markets. (2018). *Mobile Robots Market by Operating Environment (Aerial, Ground, and Marine), Component (Control System, Sensors), Type (Professional and Personal & Domestic Robots), Application (Domestic, Military, Logistics, Field), and Geography—Global Forecast 2023*. Markets and Markets. <https://www.marketsandmarkets.com/Market-Reports/mobile-robots-market-43703276.html#:~:text=of%20each%20company-,The%20mobile%20robots%20market%20is%20expected%20to%20grow%20from%20USD,the%20growth%20of%20the%20market>.

Medina-Santiago, A., Camas-Anzueto, J. L., Vazquez-Feijoo, J. A., León, H. R. H., & Mota-Grajales, R. (2014). Neural Control System in Obstacle Avoidance in Mobile Robots Using Ultrasonic Sensors. *Journal of Applied Research and Technology*, 12(1), 104-110. [https://doi.org/10.1016/S1665-6423\(14\)71610-4](https://doi.org/10.1016/S1665-6423(14)71610-4)

Mitchell, T. M. (1997). *Machine Learning* (Vol. 45). McGraw Hill, Burr Ridge.

Mohanty, P. K., & Parhi, D. R. (2015). A new hybrid optimization algorithm for multiple mobile robots navigation based on the CS-ANFIS approach. *Memetic Computing*, 7(4), 255-273. <https://doi.org/10.1007/s12293-015-0160-3>

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning* (2nd ed.). Massachusetts Institute of Technology Press.

- Motlagh, O. R. E., Hong, T. S., & Ismail, N. (2009). Development of a new minimum avoidance system for a behavior-based mobile robot. *Fuzzy Sets and Systems*, 160(13), 1929-1946. <https://doi.org/10.1016/j.fss.2008.09.015>
- NoMADA. (2016). *Data Sheet Acelerómetro—Giroscopio [PROD 041A – Acelerómetro—Giroscopio-03/16]*. NoMADA. <https://nomada-e.com/descargas/datasheet/Aceler%c3%b3metro%20y%20Giroscopio.pdf>
- Pandey, A., Kumar, S., Pandey, K. K., & Parhi, D. R. (2016). Mobile robot navigation in unknown static environments using ANFIS controller. *Perspectives in Science*, 8, 421-423. <https://doi.org/10.1016/j.pisc.2016.04.094>
- Pandey Anish, Kashyap Abhishek Kumar, Parhi Dayal R., & Patle B.K. (2019). Autonomous mobile robot navigation between static and dynamic obstacles using multiple ANFIS architecture. *World Journal of Engineering*, 16(2), 275-286. <https://doi.org/10.1108/WJE-03-2018-0092>
- Patle, B. K., L, G. B., Pandey, A., Parhi, D. R. K., & Jagadeesh, A. (2019). A review: On path planning strategies for navigation of mobile robot. *Defence Technology*, 15(4), 582-606. <https://doi.org/10.1016/j.dt.2019.04.011>
- Patle B.K., Parhi D.R., Jagadeesh A., & Kashyap Sunil Kumar. (2016). Probabilistic fuzzy controller based robotics path decision theory. *World Journal of Engineering*, 13(2), 181-192. <https://doi.org/10.1108/WJE-04-2016-024>
- Ramkumar, R., Adarsh, S., & Ramachandran, K. I. (2018). Fusion of Ultrasonic and RP Lidar 360 sensors using ANFIS for mobile robot navigation. *2018 15th IEEE India Council*

<https://doi.org/10.1109/INDICON45594.2018.8987187>

Rawat, H., Parhi, D. R., Kumar, P. B., Pandey, K. K., & Behera, A. K. (2018). Analysis and Investigation of Mamdani Fuzzy for Control and Navigation of Mobile Robot and Exploration of different AI techniques pertaining to Robot Navigation. *International Journal of Scientific & Engineering Research*, 9(4), 158-170.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536. <https://doi.org/10.1038/323533a0>

SHARP. (2020). *GP2Y0A02YK0F Data sheet.* SHARP. https://www.sparkfun.com/datasheets/Sensors/Infrared/gp2y0a02yk_e.pdf

Shihabudheen, K. V., & Pillai, G. N. (2018). Recent advances in neuro-fuzzy system: A survey. *Knowledge-Based Systems*, 152, 136-162. <https://doi.org/10.1016/j.knosys.2018.04.014>

Simić, D., Kovačević, I., Svirčević, V., & Simić, S. (2017). 50 years of fuzzy set theory and models for supplier assessment and selection: A literature review. *Journal of Applied Logic*, 24, 85-96. <https://doi.org/10.1016/j.jal.2016.11.016>

Singh, M. K., & Parhi, D. R. (2011). Path optimisation of a mobile robot using an artificial neural network controller. *International Journal of Systems Science*, 42(1), 107-120. <https://doi.org/10.1080/00207720903470155>

Singh, M. K., Parhi, D. R., & Pothal, J. K. (2009). ANFIS Approach for Navigation of Mobile Robots. *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, 727-731. <https://doi.org/10.1109/ARTCom.2009.119>

- Singh, N. H., & Thongam, K. (2019). Neural network-based approaches for mobile robot navigation in static and moving obstacles environments. *Intelligent Service Robotics*, 12(1), 55-67. <https://doi.org/10.1007/s11370-018-0260-2>
- Subbash, P., & Chong, K. T. (2019). Adaptive network fuzzy inference system based navigation controller for mobile robot. *Frontiers of Information Technology & Electronic Engineering*, 20(2), 141-151. <https://doi.org/10.1631/FITEE.1700206>
- Syed, U. A., Kunwar, F., & Iqbal, M. (2014). Guided Autowave Pulse Coupled Neural Network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots. *Robotics and Autonomous Systems*, 62(4), 474-486. <https://doi.org/10.1016/j.robot.2013.12.004>
- Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1), 116-132.
- Truong, T. T., & Nguyen, H.-T. (2020). Backtracking Gradient Descent Method and Some Applications in Large Scale Optimisation. Part 2: Algorithms and Experiments. *Applied Mathematics & Optimization*. <https://doi.org/10.1007/s00245-020-09718-8>
- Tzafestas, S. G. (2018). Mobile Robot Control and Navigation: A Global Overview. *Journal of Intelligent & Robotic Systems*, 91(1), 35-58. <https://doi.org/10.1007/s10846-018-0805-9>
- Vieira, J., Morgado, D. F., & Mota, A. (2004). Neuro-Fuzzy Systems: A Survey. *WSEAS TRANSACTIONS on SYSTEMS archive*, 3.

- Vieira, Jose, Dias, F., & Mota, A. (2003). Comparison between artificial neural networks and neurofuzzy systems in modelling and control: A case study. *IFAC Proceedings Volumes*, 36(12), 249-255. [https://doi.org/10.1016/S1474-6670\(17\)32543-0](https://doi.org/10.1016/S1474-6670(17)32543-0)
- Werbos, P. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences* [Ph. D.]. Harvard University.
- Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, 8, 338-353.
- Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, 3, 28-44.
- Zhang, X.-D. (2020). Machine Learning. En *A Matrix Algebra Approach to Artificial Intelligence* (pp. 223-440). Springer Singapore. https://doi.org/10.1007/978-981-15-2770-8_6
- Zhou, X. (2018). Understanding the Convolutional Neural Networks with Gradient Descent and Backpropagation. *Journal of Physics: Conference Series*, 1004, 012028. <https://doi.org/10.1088/1742-6596/1004/1/012028>