



UNIVERSIDAD CATÓLICA DEL NORTE  
FACULTAD DE INGENIERÍA Y CIENCIAS GEOLÓGICAS  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y  
COMPUTACIÓN

**CONTROL NEURONAL–DIFUSO DE NAVEGACIÓN REACTIVA  
CONTINÚA INCLUYENDO LA EVASIÓN DE OBSTÁCULOS  
PRESENTES Y EMERGENTES.**

Tesis para optar al grado de Magister en Ingeniería Informática.

**FELIPE ANDRÉS CORREA SANTIS**

Profesor Guía: Dr. José Gallardo

Antofagasta, Chile

2013

Dedicado a:  
Mi esposa y mi Hija, por su apoyo.

## Tabla de Contenidos

Con formato: Español (Chile)

<i>Resumen</i> .....	<i>ix</i>
<i>1 Introducción</i> .....	<i>1</i>
<i>2 Estado del Arte</i> .....	<i>4</i>
<i>3 Investigación propuesta</i> .....	<i>40</i>
<i>4 Diseño y Experimentación</i> .....	<i>44</i>
<i>5 Análisis de resultados</i> .....	<i>64</i>
<i>6 Conclusiones y Trabajo futuro</i> .....	<i>69</i>
<i>7 Anexos</i> .....	<i>71</i>
<i>8 Trabajos citados</i> .....	<i>86</i>

# INDICE DE MATERIAS

Con formato: Español (Chile)

<i>Resumen</i> .....	<i>ix</i>
<b>1</b> <i>Introducción</i> .....	<b>1</b>
<b>1.1</b> <b>Introducción</b> .....	<b>1</b>
<b>1.2</b> <b>Objetivo de la Tesis</b> .....	<b>2</b>
<b>1.3</b> <b>Estructura General</b> .....	<b>2</b>
<b>2</b> <i>Estado del Arte</i> .....	<b>4</b>
<b>2.1</b> <b>Marco Teórico</b> .....	<b>4</b>
2.1.1 Robótica Móvil.....	4
2.1.2 Inteligencia Artificial.....	7
2.1.3 Ingeniería Neuronal.....	8
2.1.4 Lógica Difusa.....	14
<b>2.2</b> <b>Discusión bibliográfica</b> .....	<b>24</b>
2.2.1 Introducción.....	24
2.2.2 Sistemas de Control.....	26
2.2.3 Sistemas Inteligentes.....	32
2.2.4 Sistemas basados en campos de potencial.....	37
2.2.5 Conclusiones del Capítulo.....	38
<b>3</b> <i>Investigación propuesta</i> .....	<b>40</b>
<b>3.1</b> <b>Problemática</b> .....	<b>40</b>
<b>3.2</b> <b>Preguntas de Investigación</b> .....	<b>40</b>
<b>3.3</b> <b>Hipótesis de Trabajo</b> .....	<b>41</b>
<b>3.4</b> <b>Objetivo General y Objetivos Específicos</b> .....	<b>41</b>
3.4.1 Objetivo General.....	41
3.4.2 Objetivos Específicos.....	42
<b>3.5</b> <b>Resultados Esperados</b> .....	<b>42</b>

<b>3.6</b>	<b>Justificación .....</b>	<b>42</b>
<b>4</b>	<b><i>Diseño y Experimentacion.....</i></b>	<b>44</b>
<b>4.1</b>	<b>Introducción.....</b>	<b>44</b>
<b>4.2</b>	<b>Diseño de la solución .....</b>	<b>48</b>
4.2.1	Redes neuronales .....	49
4.2.2	Lógica difusa .....	52
4.2.3	Controlador Neuro – Difuso. ....	59
4.2.4	Métricas .....	61
<b>4.3</b>	<b>Implementación y experimentación.....</b>	<b>62</b>
<b>5</b>	<b><i>Analisis de resultados .....</i></b>	<b>64</b>
<b>5.1</b>	<b>Simulaciones ejecutadas .....</b>	<b>65</b>
<b>5.2</b>	<b>Análisis cuantitativo de los resultados. ....</b>	<b>67</b>
<b>6</b>	<b><i>Conclusiones y Trabajo futuro .....</i></b>	<b>69</b>
<b>6.1</b>	<b>Conclusiones finales. ....</b>	<b>69</b>
<b>6.2</b>	<b>Trabajo Futuro. ....</b>	<b>70</b>
<b>7</b>	<b><i>Anexos .....</i></b>	<b>71</b>
<b>7.1</b>	<b>Anexo A: Controlador Difuso. ....</b>	<b>71</b>
7.1.1	Funciones de pertenencia. ....	71
7.1.2	Reglas difusas .....	73
<b>7.2</b>	<b>Anexo B: Red neuronal.....</b>	<b>77</b>
<b>7.3</b>	<b>Anexo C: Análisis estadístico. ....</b>	<b>78</b>
7.3.1	Análisis ANOVA para la métrica TB <sub>E</sub> .....	79
7.3.2	Análisis ANOVA para la métrica BE .....	80
7.3.3	Análisis ANOVA para la métrica velocidad Promedio. ....	81
7.3.4	Análisis ANOVA para la métrica SM1.....	82
7.3.5	Análisis ANOVA para la métrica SM2.....	84
<b>8</b>	<b><i>Trabajos citados .....</i></b>	<b>86</b>

## Figuras

Figura 2-1 Conjunto Nítido.....	15
Figura 2-2 Conjunto difuso.....	15
Figura 2-3 Ejemplos de funciones de pertenencia .....	17
Figura 2-4 Intersección entre conjuntos difusos. ....	19
Figura 2-5 Unión de conjunto difuso. ....	19
Figura 2-6 Operación negación o complemento a uno. ....	20
Figura 2-7 Variable lingüística edad.....	21
Figura 2-8 Sistema de inferencia difuso. ....	22
Figura 2-9 Ejemplo de inferencia.....	23
Figura 4-1 Distribución del anillo ultrasónico.....	46
Figura 4-2 Esquema Sistema Completo.....	49
Figura 4-3 Red neuronal.....	50
Figura 4-4 Transformación entorno. ....	52
Figura 4-5 Función de Pertenencia Z.....	54
Figura 4-6 Función de Pertenencia Triangular. ....	55
Figura 4-7 Función de Pertenencia Gaussiana.....	55
Figura 4-8 Función de Pertenencia S. ....	56
Figura 4-9 Funciones de pertenencia. ....	57
Figura 4-10 Reglas difusa modelo 5 variables lingüísticas. ....	58
Figura 4-11 Método de desfusificación. ....	59
Figura 4-12 Esquema Sistema Completo.....	60
Figura 5-1 Entorno de simulación.....	64
Figura 5-2 Arquitectura reactiva pura. ....	65
Figura 5-3 Arquitectura reactiva neuronal.....	66

Figura 5-4 Arquitectura reactiva difusa. ....	66
Figura 5-5 Arquitectura reactiva neuro – difusa. ....	67
Figura 7-1 Funciones de pertenencia variables de entrada. ....	71
Figura 7-2 Funciones de pertenencia variables de salida. ....	72

## Tablas

Tabla 5-1 Resultados de las simulaciones. _____	68
Tabla 7-1 funciones y parámetros funciones de pertenencia _____	72
Tabla 7-2 Reglas difusas _____	73
Tabla 7-3 Pesos sinápticos IN - HIDE 1 _____	77
Tabla 7-4 Pesos sinápticos HIDE 1 - HIDE 2 _____	78
Tabla 7-5 Pesos sinápticos HIDE 2 - OUT _____	78
Tabla 7-6 TBe método de Tukey. _____	80
Tabla 7-7 Be método de Tukey. _____	81
Tabla 7-8 Velocidad promedio método de Tukey. _____	82
Tabla 7-9 SM1 método de Tukey. _____	83
Tabla 7-10 SM2 método de Tukey. _____	85

## Ecuaciones

Ecuación 2-1.....	10
Ecuación 2-2.....	11
Ecuación 2-3.....	12
Ecuación 2-4.....	12
Ecuación 2-5.....	12
Ecuación 2-6.....	16
Ecuación 2-7.....	18
Ecuación 2-8.....	18
Ecuación 2-9.....	18
Ecuación 2-10.....	19
Ecuación 2-11.....	20
Ecuación 4-1.....	46

**Con formato:** Revisar la ortografía y la gramática

Ecuación 4-2.....	48
Ecuación 4-3.....	54
Ecuación 4-4.....	54
Ecuación 4-5.....	55
Ecuación 4-6.....	56
Ecuación 4-7.....	59
Ecuación 4-8.....	62
Ecuación 4-9.....	62
Ecuación 4-10.....	62
Ecuación 7-1.....	77

## RESUMEN.

El proyecto de tesis presentado para la obtención del grado de Magister en Ingeniería Informática, es una investigación desarrollada en el ámbito de la robótica y más específicamente en robótica móvil.

Esta investigación parte desde los antecedentes históricos de la evolución de la idea y desarrollos primitivos, pasando por la evolución que tienen los robots actuales y cómo la ciencia y la tecnología han evolucionado y transformando los paradigmas que enfrenta el hombre en el largo camino de crear artilugios autónomos que imiten o reproduzcan sus mismas capacidades.

En este sentido, uno de los aspectos más influyentes es el poder dotar a los robots de la capacidad de tomar decisiones, lo que está fuertemente ligado a la inteligencia humana. Es por ello, que la investigación de la inteligencia artificial en este trabajo, tiene una importancia más relevante, así como la teoría de *Lógica Difusa*, que permite modelar variables lingüísticas, o en otras palabras que el robot pueda procesar conceptos que no tienen una precisión matemática formal.

Lo anterior con el objetivo de aplicar estos conocimientos en una experiencia práctica, en la cual se pretende mejorar la performance de una plataforma robótica definida. Para ello, se parte del postulado de que aplicando algoritmos de lógica difusa y redes neuronales, en el control de navegación de un robot móvil, se puede lograr una mejor evasión de obstáculos tácitos y emergentes generando una labor más efectiva, al permitirle al robot modificar su velocidad en los momentos que no tiene como

prioridad evadir obstáculos, haciendo de éste un agente más rápido y eficiente, la comparación se realizara en función de porcentaje alcanzado en razón de la velocidad teórica máxima, y la cantidad de colisiones.

Para cumplir con el objetivo anteriormente mencionado, se realizaran diferentes pruebas en distintos entornos, y cada uno con diferente nivel de complejidad, rescatando de cada uno de estas pruebas datos representativos para ser comparados.

# 1 INTRODUCCIÓN.

## 1.1 Introducción.

El presente trabajo de tesis consiste en el desarrollo de un controlador Neuro-Difuso como mecanismo de control de navegación continua de una plataforma robótica reactiva móvil autónoma.

Se busca que el robot sea capaz de desplazarse en un ambiente cambiante, eludiendo los obstáculos que enfrenta y en ausencia de estos aumente su velocidad de desplazamiento, a fin de que ejecute su tarea de forma más eficiente y efectiva.

En la evolución que han tenido los robots a través del tiempo, se han ido empleando diferentes mecanismos de control para definir su comportamiento, lo más reciente es el empleo de técnicas de inteligencia artificial, que aumentan las capacidades adaptativas de los robots.

El uso de la lógica difusa permitirá que el robot opere variables lingüísticas, tales como cerca o lejos, lo cual representa una ventaja sobre la lógica matemática convencional, ya que no impone límites taxativos (on-off) en las decisiones. Estas variables permiten el acercamiento de la lógica borrosa al lenguaje natural, facilitando así su utilización.

Por otra parte, las redes neuronales, son capaces de adaptar su funcionamiento a distintos entornos, modificando sus conexiones entre neuronas. De esta manera, pueden aprender de la experiencia y generalizar

conceptos. Por consiguiente, se espera que con el uso de éstas técnicas, el robot pueda tener una respuesta más adaptativa a entornos cambiantes, estructurados o no (ambiente real).

## **1.2 Objetivo de la Tesis.**

La presente tesis, primordialmente, persigue optimizar el comportamiento de una plataforma robótica existente, mediante la aplicación de las técnicas mencionadas, a saber: lógica difusa y redes neuronales. Por lo que mediante el algoritmo Neuro-Difuso, se espera que el robot sea capaz de eludir mejor y más eficientemente los obstáculos que se le presenten en el camino y ejecutar su tarea en el menor tiempo, logrando un desempeño más eficaz y un uso más eficiente de energía.

Por otra parte, se considera pertinente señalar además que éste trabajo aportará como base de conocimiento al Departamento de Ingeniería de Sistemas y Computación de la Universidad, para las investigaciones futuras en este campo del saber y consecuentemente puede considerarse como un objetivo secundario o de más largo plazo.

## **1.3 Estructura General.**

En el siguiente apartado, se presentará de forma sucinta una descripción de los tópicos presentados en esta propuesta de tesis, para dar al lector una idea más precisa de los temas señalados en la tabla de contenido.

**Capítulo 2:** Es una presentación de los antecedentes teóricos que sustentan el desarrollo de la tesis, lo cual se exhibirá por medio de un marco teórico. Junto con esto se revisa la literatura existente frente al tema de investigación, para abrir paso al siguiente capítulo.

**Capítulo 3:** En este capítulo se expone el problema que se desea resolver, por medio de este proyecto de tesis. Así mismo, se presentan antecedentes generales del problema identificado, la hipótesis principal, los objetivos (generales y específicos), los resultados que se pretenden lograr y su justificación.

**Capítulo 4:** Este capítulo detalla el proceso de diseño, implementación y experimentación del controlador neuro-difuso. Además se presentan las métricas que serán usadas para medir y comparar la veracidad de la Hipótesis de tesis.

**Capítulo 5:** En este capítulo se compara el controlador neuro – difuso respecto de otras alternativas, analizando los resultados obtenidos en las pruebas realizadas.

**Capítulo 6:** Se presentan las conclusiones de la investigación realizada y también se comentan posibles mejoras para el modelo.

**Capítulo 7:** Se presentan los anexos, con información más detallada del proceso de diseño.

**Capítulo 8:** Se presentan las fuentes que fueron consultadas y citadas, que sustentan esta tesis.

## **2 ESTADO DEL ARTE.**

### **2.1 Marco Teórico.**

La robótica es una ciencia presente en la humanidad desde la antigüedad, la cual ha evolucionado en el tiempo. En este apartado se profundizará en la robótica móvil y en inteligencia artificial, haciendo especial énfasis en dos técnicas, redes neurales y lógica difusa, las cuales serán utilizadas para dotar al robot móvil de comportamiento inteligente.

#### **2.1.1 Robótica Móvil.**

En la actualidad, según su morfología, los robots se pueden clasificar en cuatro tipos: Robot industrial, Robot móviles, Androides y Zoomórficos. Cuando un robot puede desplazarse autónomamente por cualquier medio, se habla de robótica móvil, este tipo de robots pueden desplazarse gracias a su morfología la cual puede poseer, ruedas, patas, propulsores, hélices entre otros, además de poseer un sistema de suministro energético propio, ya sea por almacenamiento en baterías o paneles solares. Aunque estas características les permiten moverse de forma autónoma, esto por sí solo no es suficiente para desplazarse, requieren poseer sensores externos que les permitan observar su entorno y algoritmos que conformen su lógica interna (Ruiz de Garibay Pascual, 2007).

Los robots móviles pueden operar en diversos ambientes, algunos peligrosos para el humano. Realizando tareas repetitivas que requieren precisión y apoyando las tareas de búsqueda y rescate así como también

sirviendo a personas con capacidades disminuidas (Ruiz de Garibay Pascual, 2007).

Las principales líneas investigativas en robótica móvil se centran en mejorar los diseños físicos, estudiando los componentes, la cinemática y dinámica para aprovechar de mejor manera las fuerzas intrínsecas, mejorar la gestión energética, algoritmos que permitan la toma de decisiones inteligentes y utilización de sensores tanto para la navegación como para la medición de factores ambientales (Ruiz de Garibay Pascual, 2007).

Como los robots móviles operan en un ambiente de alta complejidad, se ha recurrido a la utilización de técnicas de control más avanzadas, que le posibilitan trabajar en condiciones de imprecisión e incertidumbre, reaccionando de forma rápida frente a los obstáculos, y también poder definir, de manera autónoma sub tareas que contribuyan a alcanzar su objetivo primario (Vallejo Rodríguez, 2004).

#### 2.1.1.1 Paradigmas.

Para establecer una base formal, que permita exponer de forma clara y precisa el control de navegación de robots móviles, se exponen los siguientes términos: Arquitectura y Paradigma. Si bien la **Arquitectura** representa la estructura de un robot tanto en el aspecto físico como conceptual del sistema de control, el **Paradigma** apunta hacia una filosofía que aborda los problemas, la elección y el uso de las herramientas necesarias para su solución (Vallejo Rodríguez, 2004).

Existen tres primitivas generalmente aceptadas en robótica. **La percepción**, representa las tareas en las cuales se obtiene información del ambiente. **Planificación**, es el proceso que convierte la información obtenida en tareas para ser realizadas por el robot. Y **Acciones**, son las operaciones que generan acción en los actuadores del robot (Vallejo Rodríguez, 2004).

De acuerdo a la forma en que se organizan estas tres primitivas se originan cuatro paradigmas en la robótica. El **Paradigma Deliberativo**, se basa en la concepción que para la realización de una acción esta requiere una planificación previa. El **Paradigma Reactivo**, utiliza solamente dos de las primitivas mencionadas, percepción y acción. Este paradigma será revisado en extenso en el punto siguiente. **Paradigma Híbrido**, nace de la unión de los dos paradigmas anteriormente mencionados, el cual constituye un bloque reactivo bajo uno deliberativo, el cual planifica tareas que deben realizarse de forma reactiva. **Paradigma con Aprendizaje**, este paradigma incorpora al aprendizaje como una cuarta primitiva (Vallejo Rodríguez, 2004).

El **Paradigma Reactivo**, tiene una relación directa de las entradas de los sensores con la salida de sus actuadores, sin que exista un bloque de planificación, este paradigma puede resumirse en una relación bilateral entre percibir – actuar (Vallejo Rodríguez, 2004). La ventaja que provee esta configuración, es la rápida respuesta que presenta el modelo frente al entorno y los cambios en él (ambiente dinámico). Sin embargo como desventaja, carece de planificación global que le permita realizar tareas complejas. En la Figura 2-1 puede observarse el esquema de este paradigma.

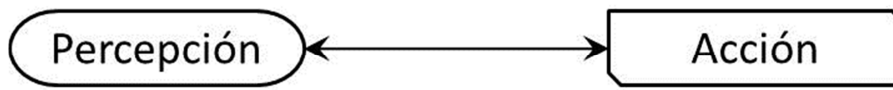


Figura 2-1 Arquitectura reactiva

### 2.1.2 Inteligencia Artificial

La Inteligencia Artificial (IA) se enfoca en emular entidades inteligentes, sintetizando y automatizando las tareas que requieren uso del intelecto humano (J. Russell & Norvig, 2004).

En la IA a “cualquier cosa capaz de percibir su medioambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores” se le denomina **Agente**. El comportamiento de éste se describe en termino ~~matematicos~~matemáticos a ~~trave~~través de la ~~funcion~~función del agente y el programa del agente, sobre el cual la función es implementada (J. Russell & Norvig, 2004).

Es común confundir estos *dos conceptos*. La diferencia entre ambos recae en términos temporales. El programa del agente trabaja con la percepción – que es la capacidad para sentir su ambiente - actual del agente, mientras que la función del agente actúa sobre el conjunto datos histórico sentidos (J. Russell & Norvig, 2004).

*“un agente tomará una decisión en un momento dado dependiendo de la secuencia completa de percepciones hasta ese instante”* (J. Russell & Norvig, 2004).

**Con formato:** Sangría: Izquierda: 0 cm, Sangría francesa: 1,25 cm

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

La función que define al programa del agente se representa mediante una tabla la cual debe contener todas las posibles entradas. Por ejemplo, para diseñar un agente que opere sobre un tablero de ajedrez, se necesitaría contar con al menos  $10^{150}$  entradas (J. Russell & Norvig, 2004). Pero un tamaño tan desmesurado genera los siguientes problemas.

1. No existe agente físico en el mundo capaz de almacenar una tabla de esas dimensiones (J. Russell & Norvig, 2004).
2. El tiempo necesario para la creación de una tabla así, excederá las capacidades temporales del diseñador (J. Russell & Norvig, 2004).
3. El agente no será capaz de aprender todas las entradas de la tabla en función de su experiencia (J. Russell & Norvig, 2004).
4. El diseñador no contará con alguien que pueda asesorarlo para el correcto llenado de la tabla (J. Russell & Norvig, 2004).

Por lo tanto, el desafío de la IA es crear programas, que con una reducida cantidad de códigos, puedan ser capaces de reproducir el comportamiento racional, en vez de necesitar tablas de inconmensurable dimensión para cumplir con sus metas (J. Russell & Norvig, 2004).

Para alcanzar este objetivo se han desarrollado variadas técnicas de inteligencia artificial. Dentro de las más **usadas** en la navegación autónoma destacan ingeniería neuronal y lógica difusa.

### 2.1.3 Ingeniería Neuronal

El objetivo principal de la ingeniería neuronal es la emulación del comportamiento del cerebro humano, a través de modelos matemáticos

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

denominados *redes neuronales artificiales* (RNA), al igual que todo modelo, es una abstracción de la realidad, tomando sus características principales (Bertona, 2005).

Las RNA son una simplificación del cerebro, y al igual que este, establece como elemento básico de procesamiento la neurona, la cual se agrupa en estructuras denominadas capas, y el conjunto de estas conforman la red neuronal. Esta estructura dota a las RNA de procesamiento paralelo, información distribuida y adaptabilidad (Bertona, 2005).

#### 2.1.3.1.1 Modelo de una neurona artificial.

Las neuronas artificiales pueden ser clasificadas, según su funcionalidad, en tres tipos:

1. Las que reciben información directamente desde el exterior, llamadas neuronas de entrada.
2. Las que reciben información desde otras neuronas artificiales, llamadas neuronas ocultas. En estas se almacena la representación de la información.
3. Las que reciben información procesada y las devuelven al exterior. Llamadas neuronas de salida.

La clasificación de las neuronas, solamente distingue la posición que poseen estas en la red neuronal artificial debido a que todas poseen la misma estructura:

1. **Conjunto de entradas  $x_j(t)$ .** Estas pueden ser provenientes del exterior o de otras neuronas artificiales.

2. **Pesos sinápticos  $W_{ij}$ .** Representa el grado de comunicación entre la neurona artificial  $j$  y la neurona artificial  $i$ . pueden ser excitadores o inhibidores.
3. **Regla de propagación  $\sigma_i(W_{ij}, x_j(t))$ .** Integra la información proveniente de las distintas neuronas artificiales y proporciona el valor del potencial post sináptico de la neurona  $i$ .
4. **Funciones de activación  $f_i(a_i(t-1), h_i(t))$ .** Provee el estado de activación actual de la neurona  $i$ .
5. **Función de salida  $F_i(a_i(t))$ .** Representa la salida actual de la neurona  $i$ .

De esta forma, la salida producida por una neurona  $i$ , para un determinado instante de tiempo  $t$  puede ser escrita en forma general como se muestra en la Ecuación 2-1, (Bertona, 2005).

$$y_i(t) = F_i \left( f_i \left[ a_i(t-1), \sigma_i \left( w_{ij}, x_j(t) \right) \right] \right)$$

Ecuación 2-1

#### 2.1.3.1.2 Entradas y salidas.

La transmisión de datos en las neuronas biológicas es representada en las RNAs como valores numéricos, pudiendo estos ser binarios o continuos. Si la RNA admite valores binarios, se denominará **red neuronal artificial digital** y utilizará el alfabeto  $\{0,1\}$  o  $\{-1,1\}$ . Pero si la RNA trabaja con valores continuos, se hablará de **red neuronal artificial análoga**, y habitualmente se utiliza el alfabeto  $[-1,1]$  (Bertona, 2005).

#### 2.1.3.1.3 Pesos sinápticos.

La representación matemática de la intensidad existente entre dos neuronas biológicas, se denomina peso sináptico y es denotado como  $W_{ij}$ . El subíndice  $i$  representa la neurona transmisora y  $j$  representa la receptora. Y al igual que el sistema que representa, este toma valores positivos en el caso de ser excitador, valores negativos cuando es inhibidor y valores nulos cuando no existe esa conexión. El ajuste de los pesos sinápticos de la red es lo que permite que las RNA, al igual que el sistema biológico, puedan adaptarse a distintos entornos y realizar tareas específicas (Bertona, 2005).

#### 2.1.3.1.4 Reglas de propagación.

Las reglas de propagación son las que permiten obtener el resultado de la neurona  $i$  con  $N$  neuronas vecinas. Como las RNAs son un modelo matemático, a este valor se designa como  $h_i$ . Existen varias reglas de propagación pero la más utilizada y simple al mismo tiempo, es la sumatoria de todas las entradas, las cuales se ponderan con sus pesos sinápticos, como se puede observar en la Ecuación 2-2 (Bertona, 2005).

$$h_i(t) = \sum_j W_{ij} \cdot x_j(t)$$

Ecuación 2-2

#### 2.1.3.1.5 Función de activación.

La función de activación se encarga de establecer el estado de activación que posee una neurona en el instante actual. Esta función se denota  $a_i(t)$ , y depende de la correspondencia entre el potencial resultado  $h_i$  y el estado de activación anterior de la neuronal  $a_i(t - 1)$ , como se puede ver en la **¡Error!**

Con formato: Revisar la ortografía y la gramática

Con formato: Español (Chile), Resaltar

Con formato: Español (Chile)

~~No se encuentra el origen de la referencia.~~ Aunque comúnmente se suele omitir el valor del estado anterior de la neurona, dependiendo el estado de activación solamente del potencial resultante, como se muestra en la Ecuación 2-3 y Ecuación 2-4 (Bertona, 2005).

$$a_i(t) = f_i(a_1(t-1), h_i(t))$$

Ecuación 2-3

$$a_i(t) = f_i(h_i(t))$$

Ecuación 2-4

#### 2.1.3.1.6 Función de salida.

El valor de la función de salida se calcula en función del estado de activación de la neurona. Comúnmente se ocupa la función identidad como se puede ver en la Ecuación 2-5 (Bertona, 2005).

$$y_i(t) = F_i(a_i(t)) = a_i(t)$$

Ecuación 2-5

#### 2.1.3.2 Topología de la Red Neuronal

Cuando se desea implementar una RNA, es necesario considerar, además del tipo de neuronas a utilizar, la **topología** de la misma. Se entiende por topología a la organización que tendrán las neuronas dentro de la red, determinado número de capas y la cantidad de neuronas que tendrán. Otro punto importante a definir, es el grado y tipo de conexión que estas tendrán (Bertona, 2005).

Existen dos tipos de conexión presentes en las RNAs. Las conexiones de una neurona hacia neuronas de otra capa, lo cual se denominan conexiones *inter – capa*. Mientras que si son en la misma capa se llaman *intra – capa*. Además, pueden existir conexiones contrarias al flujo in – out (entrada – salida) denominadas *conexiones recurrentes* (Bertona, 2005).

### 2.1.3.3 Aprendizaje

Una vez determinada la topología de la RNA, se comienza con la etapa de aprendizaje. En esta etapa se entrena la red para realizar determinado comportamiento, hasta que se logre un nivel deseado de entrenamiento (Bertona, 2005).

El proceso de aprendizaje de una RNA se inicia asignando de forma aleatoria valores a los pesos sinápticos, los que serán modificados durante el transcurso de esta fase, con el fin de lograr un nivel operacional deseado. Existen tres tipos de aprendizaje con los cuales una RNA puede ser entrenadas (Bertona, 2005):

1. **Aprendizaje supervisado:** para el entrenamiento, se prepara un conjunto de valores de entrada y sus salidas esperadas. Durante el proceso, los pesos sinápticos se ajustan para cumplir con las condiciones presentadas (Bertona, 2005).
2. **Aprendizaje no supervisado:** Como su nombre lo indica, no se genera un conjunto de entrenamiento, y es la red por sí sola, que ajusta sus pesos sinápticos en función de la correlación existente entre los datos de entrada (Bertona, 2005).

3. **Aprendizaje por refuerzo:** este modelo de entrenamiento se puede situar en un punto medio de los anteriores. Por medio del refuerzo positivo y negativo se le indica a la red, si el resultado generado para una entrada dada es correcto o no (Bertona, 2005).

#### 2.1.3.4 Fase de operación.

Concluida la fase de entrenamiento la RNA ya está preparada para ser utilizada en la tarea para la cual fue diseñada. La característica más importante que posee es la capacidad que de generalizar conceptos. Esto le permite a la RNA operar con datos que no fueron presentados durante la fase de entrenamiento (Bertona, 2005).

### 2.1.4 *Lógica Difusa*

La Lógica Difusa es una técnica de IA, que por su naturaleza multivaluada ha permitido un impulso significativo de la inteligencia artificial en corto tiempo, ya que permite trabajar la imprecisión de forma matemática. Esto se debe a que su planteamiento no radica en determinar la pertenencia de un elemento a un conjunto, sino por el contrario mide el nivel de pertenencia que tiene el elemento a dicho conjunto (Barragán Piña, 2009).

#### 2.1.4.1 Fundamentos de los Sistemas Difusos

##### 2.1.4.1.1 Conjuntos Difusos y funciones de pertenencia

La diferencia entre la lógica difusa y la clásica, radica en el concepto de pertenencia. En un conjunto nítido la pertenencia de un elemento a un conjunto está completamente definida, es decir, su propiedad es nítida. Mientras que en un conjunto difuso la pertenencia esta expresada en términos

ambiguos; por lo cual, expresar con certeza la pertenencia de los elementos al conjunto es imposible (Barragán Piña, 2009).

En consecuencia, la diferencia entre los conjuntos difusos y los nítidos reside en la misma definición de los límites de estos. En la lógica nítida, la pertenencia del elemento al conjunto se determina en parámetros booleanos (verdadero – falso), mientras que la lógica difusa, es la función de pertenencia que determina el grado de pertenencia que tiene un elemento al conjunto. En la ~~s~~ Figura 2-1 ~~;~~ ~~Error! No se encuentra el origen de la referencia.~~ y Figura 2-2 ~~;~~ ~~Error! No se encuentra el origen de la referencia.~~ se ilustra visualmente la diferencia en los límites de los conjuntos nítidos y difusos. Se considera en color, todos los elementos que pertenecen al conjunto y en blanco aquellos que no. De esta forma se distingue claramente que los valores de pertenencia a un conjunto nítido - Figura 2-1 - pueden ser 0 y 1, es decir pertenece o no pertenece. Mientras que el caso del conjunto difuso - Figura 2-2, los valores de pertenencia se encuentran en el intervalo [0,1] (Barragán Piña, 2009).

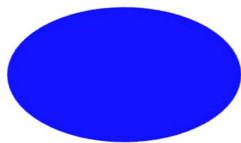


Figura 2-1 Conjunto Nítido

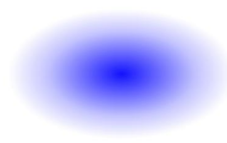


Figura 2-2 Conjunto difuso.

Para poder saber si un elemento pertenece a un conjunto difuso se utiliza una **función de pertenencia**. Ésta, denotada con la letra griega  $\mu$ , determina el grado de pertenencia que posee un elemento dicho conjunto.

Código de campo cambiado

Con formato: Español (Chile)

Código de campo cambiado

Código de campo cambiado

Código de campo cambiado

Con formato: Español (Chile)

Código de campo cambiado

Código de campo cambiado

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

*x no pertenece a A si  $\mu_A(x) \geq \alpha$*   
*x pertenece a A si  $\mu_A(x) \leq \beta$*   
*x tiene un estatus indeterminado con respecto a A si  $\beta < \mu_A(x) < \alpha$*

Es importante señalar, que el grado de pertenencia de un elemento a un conjunto difuso, no es igual al concepto de la probabilidad de que el mismo elemento pertenezca a un conjunto nítido. Por muy similar que pueden parecer existen importantes diferencias entre ambos conceptos.

Para ejemplificar de forma tangible la diferencia entre ambos, se puede suponer que se observa una botella, la cual tiene tabulado que posee un grado de pertenencia de 0.01 al conjunto de venenos mortales. El rotulado no significa que la botella posea una probabilidad del 1% de ser veneno mortal, sino, muy por el contrario afirma que posee una muy pequeña cantidad de algún veneno mortal, por lo mismo implicaría solamente un riesgo para la salud (Barragán Piña, 2009).

#### 2.1.4.1.2 Definiciones básicas

**Conjunto difuso:** Sea  $X$  un conjunto no vacío, un conjunto difuso  $A$  está caracterizado por la Ecuación 2-6 .

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

Ecuación 2-6

Donde  $\mu_A(x)$  representa el grado de pertenencia del elemento  $x$  sobre el conjunto difuso  $A$  para cada  $x \in X$ .  $A$  está completamente determinado por el conjunto de pares. Esta ecuación muestra que los conjuntos difusos quedan caracterizados por su función de pertenencia, la cual puede poseer una

definición discreta o continua, siendo esta última la más comúnmente utilizada (Barragán Piña, 2009).

En la Figura 2-3 se exponen algunos ejemplos de funciones de pertenencia, todas ellas con un *universo de discurso* de  $X \in [-5,5]$ . Se puede apreciar que la primera de las funciones  $\mu_A(x)$ , es una función nítida, mientras que el resto de ellas son funciones de pertenencia difusas.

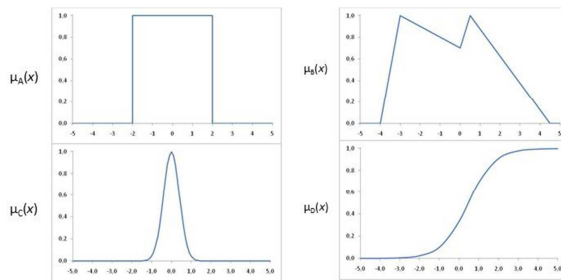


Figura 2-3 Ejemplos de funciones de pertenencia

Con formato: Revisar la ortografía y la gramática

#### 2.1.4.1.3 Operaciones sobre conjuntos difusos.

Aun con las visibles diferencias que existen entre lógica difusa y lógica nítida, existen operaciones que pueden ser aplicadas indistintamente a ambos tipos de lógica. Tales como unión, intersección y complemento. A continuación se ~~revisaran~~ revisarán algunas de las operaciones y propiedades básicas de la lógica difusa (Barragán Piña, 2009).

**Igualdad:** Se define como igualdad de conjuntos difusos a aquellos que poseen funciones de pertenencia iguales para todo el universo de discurso (Barragán Piña, 2009).

$$A = B \Leftrightarrow \mu_A(x) = \mu_B(x), \forall x \in X$$

Ecuación 2-7

**Inclusión:** Se define como inclusión de conjuntos difusos cuando un conjunto está contenido en otro, esto es equivalente a decir que un conjunto difuso es subconjunto de otro (Barragán Piña, 2009).

$$\left. \begin{array}{l} A \subset B \\ A \leq B \end{array} \right\} \Leftrightarrow \mu_A(x) \leq \mu_B(x), \forall x \in X$$

Ecuación 2-8

**Intersección:** Se define como intersección de conjuntos difusos, al área en común existente entre dos funciones de pertenencia distintas, además se cumple que para dos conjuntos difusos distintos y ambos convexos, la intersección también será convexa. Para el caso de que la intersección forme un conjunto vacío, se dice que ambos son disjuntos. La Ecuación 2-9 y Figura 2-4 se presenta un ejemplo y formula (Barragán Piña, 2009).

$$\begin{aligned} \mu_C(x) = \mu_{A \cap B}(x) &= \min\{\mu_A(x), \mu_B(x)\}, \forall x \in X \\ \mu_C &= \mu_A \wedge \mu_B \end{aligned}$$

Ecuación 2-9

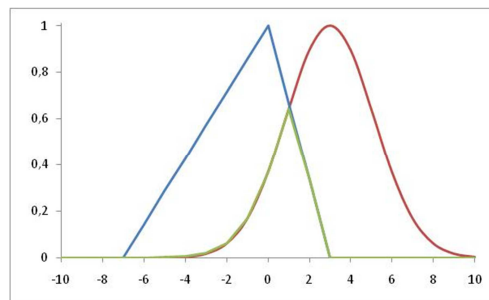


Figura 2-4 Intersección entre conjuntos difusos.

**Unión:** Se define como unión de dos conjuntos difusos, al área conjunta que ocupan ambas funciones de pertenencia. En la Ecuación 2-10 y Figura 2-5 se presenta un ejemplo y fórmula (Barragán Piña, 2009).

$$\mu_C(x) = \mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \forall x \in X$$

$$\mu_C = \mu_A \vee \mu_B$$

Ecuación 2-10

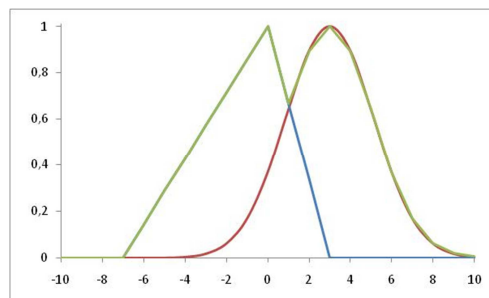


Figura 2-5 Unión de conjunto difuso.

Con formato: Revisar la ortografía y la gramática

**Negación o complemento a uno:** se define como negación o complemento a uno de un conjunto difuso, a la función de pertenencia inversa a la del conjunto. En la Ecuación 2-11 y Figura 2-6 se presenta un ejemplo y fórmula (Barragán Piña, 2009).

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

Ecuación 2-11

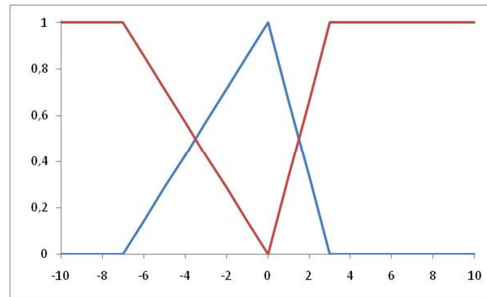


Figura 2-6 Operación negación o complemento a uno.

**Con formato:** Revisar la ortografía y la gramática

Se han presentado las operaciones y propiedades más básicas de la lógica difusa, considerando que este proyecto de tesis está enfocado a la aplicación de esta estrategia y no al estudio y validación axiomática de esta herramienta matemática, por ende también se han obviado las operaciones avanzadas presentes en la literatura.

#### 2.1.4.1.4 Variables Lingüísticas.

Una variable lingüística es aquella que posee valores que son palabras o sentencias en un lenguaje natural o artificial y permiten la transformación de este a lógica difusa. De esta manera se facilita la utilización lógica del lenguaje común, sin poseer las deficiencias del lenguaje preciso (Barragán Piña, 2009).

Hay que tener en cuenta que, mientras más complejo sea un sistema, será más difícil poder hacer afirmaciones precisas y significativas sobre algún comportamiento. Llegando incluso a niveles tan elevados en que los conceptos de precisión y relevancia sean mutuamente excluyentes (Barragán Piña, 2009).

Como se puede apreciar en la Figura 2-7, se han representado distintas funciones de pertenencia, en la que cada una de ellas hace referencia a un término lingüístico, que para este ejemplo se escogió la variable lingüística edad, y se ha representado en etiquetas lingüísticas como *niño*, *joven*, *adulto* y *anciano* (Barragán Piña, 2009).

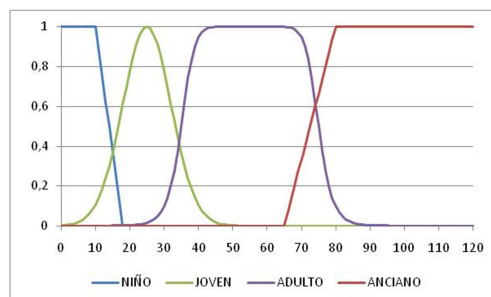


Figura 2-7 Variable lingüística edad.

Con formato: Revisar la ortografía y la gramática

#### 2.1.4.2 Sistema de Inferencia Difuso.

Se denomina sistema de inferencia difuso a todo sistema que esté basado en lógica difusa y en el razonamiento aproximado. Estos sistemas se dividen en dos componentes, el *motor de inferencia difusa* y *base de conocimientos*. Dependiendo de la aplicación esperada del sistema, puede ser necesario incorporar dos elementos, *fuzzyficador* y *defuzzyficador*, que cumplen con la función de vincular al sistema de inferencia difuso con el mundo real. La Figura 2-8 muestra la configuración típica de un sistema de inferencia difusa que está conectado con el mundo real (Barragán Piña, 2009).

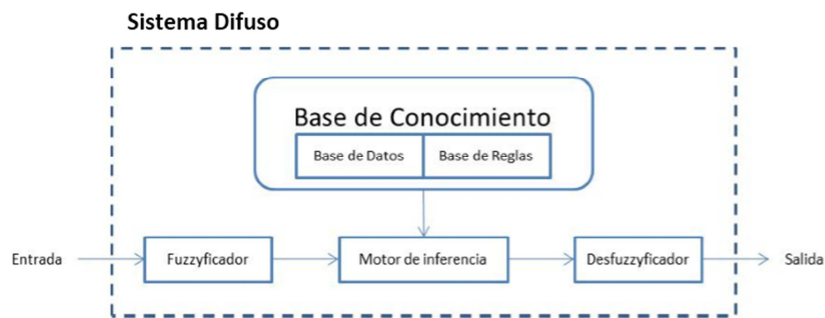


Figura 2-8 Sistema de inferencia difuso.

**Con formato:** Revisar la ortografía y la gramática

#### 2.1.4.2.1 Fuzzyficador

En general las entradas que posee un sistema difuso, son valores numéricos, los cuales son provistos por sensores. Por esta razón se hace necesario incorporar un componente capaz de procesar los valores de entradas y traducirlos en valores difusos, para ser procesados por el motor de inferencia. El componente que realiza esta tarea es denominado fuzzyficador (Barragán Piña, 2009).

#### 2.1.4.2.2 Inferencia difusa.

El motor de inferencia difusa es el encargado de generar las salidas del sistema, en función de las reglas y las entradas, las cuales pueden ser difusas o haber sido fuzzyficadas. Los valores de salida generados dependerán de los requerimientos para los cuales fueron diseñados (Barragán Piña, 2009).

Otra perspectiva que permite comprender el funcionamiento del motor de inferencia difusa, es considerarlo como un modelo matemático de interpolación no lineal, el cual permite combinar la información de las reglas en una conclusión única. Este resultado se logra por la composición de los consecuentes de las reglas ponderadas con una función de grado del cumplimiento de cada una de ellas (Barragán Piña, 2009).

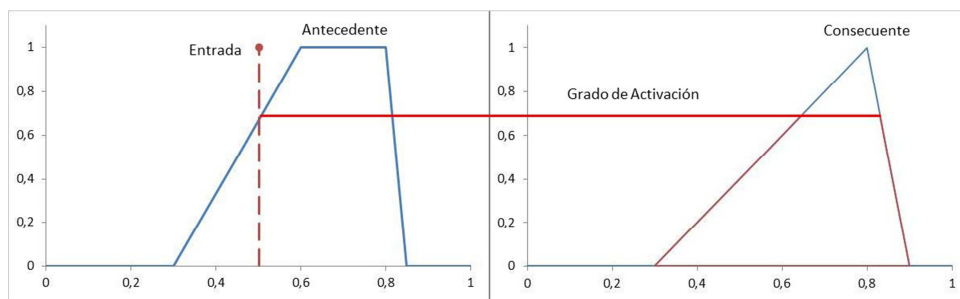


Figura 2-9 Ejemplo de inferencia.

Se puede observar en la Figura 2-9, un ejemplo gráfico de inferencia difusa, en el cual el grado de activación de la regla se obtiene evaluando la función de pertenencia del antecedente en el punto otorgado por la entrada. Este grado de activación es propagado hacia su consecuente.

#### 2.1.4.2.3 Base de conocimiento.

La base de conocimiento, es el componente que almacena las relaciones entre las entradas y salidas. Este componente se puede analizar en dos partes (Barragán Piña, 2009):

1. **La base de datos** contiene las funciones de pertenencia que definen cada una de las etiquetas lingüísticas.
2. **La base de reglas** contiene todas las reglas lingüísticas del sistema.

Con formato: Revisar la ortografía y la gramática

#### 2.1.4.2.4 Defuzzificador.

Como se mencionó con anterioridad, existen casos en que se requiere que el sistema de inferencia difuso entregue valores nítidos, por ende, se requiere emplear una interfaz que transforme el valor difuso resultante del motor de inferencia en un valor numérico. El componente encargado de realizar esta tarea es el defuzzificador (Barragán Piña, 2009).

## 2.2 Discusión bibliográfica.

### 2.2.1 Introducción.

En la literatura revisada, se puede apreciar la existencia de distintos enfoques para tratar la problemática de la robótica móvil autónoma, esta gran gama comprende desde sistemas que trabajan en la optimización de rutas ocupando teoría de control y cálculos vectoriales, hasta sistemas que se enfocan en lograr comportamientos reactivos ocupando algoritmos inteligentes. En este capítulo se hará énfasis en los enfoques reactivos inteligentes.

La complejidad en la robótica móvil autónoma proviene de tener que enfrentar la naturaleza dinámica del entorno, es por esto que se hace necesario ~~dotar~~ incorporar en el robot, de mecanismos reactivos para sortear obstáculos no previstos en el mapa (navegación planificada). Pero el objetivo más importante en la robótica móvil autónoma, es la capacidad de explorar ambientes nuevos o desconocidos (Tsankova, 2010), evitando colisiones y completar la misión encomendada, lo cual se logra dotándolo de una capacidad adaptativa.

La capacidad adaptativa del robot le permitirá enfrentar situaciones de mayor complejidad, como ambientes dinámicos y/o desconocidos. Para esto el robot debe poseer un grado de inteligencia que le permita una adecuada toma de decisiones, pero esto no basta, ya que además debe poseer la capacidad del correcto aprendizaje para enfrentar nuevas condiciones. (Alhaj Ali, Ghaffari, Liao, & Hall, 2006)

Algunos de los sistemas y métodos usados de recolección de datos para la navegación adaptativa del robot móvil son (Alhaj Ali, Ghaffari, Liao, & Hall, 2006):

- Navegación basada en sensores.
- Navegación basada en visión.
- Sistema de navegación de faros activos.
- Navegación basada en lugares de referencia.
- Posicionamiento basado en mapas.
- Sistema de posicionamiento global (GPS).

Estos métodos y sistemas constituyen formas de recolección de datos que el robot deberá procesar para desplazarse en el entorno. La lógica a usar será la que definirá la capacidad de operar en forma autónoma y la habilidad de resolver situaciones más o menos complejas. Existen también diferentes enfoques para enfrentar este aspecto del problema, desde sistemas determinísticos a sistemas adaptativos, estos últimos permiten mayor flexibilidad de respuesta frente a ambientes cambiantes.

En general, la literatura relativa al tema de investigación puede dividirse en dos grupos, que permitirán un mejor flujo y orientación para el tema a tratar en este proyecto de tesis. Estos grupos son: *sistemas de control* y *sistemas inteligentes*. En el primer grupo se considerarán a todos aquellos que no correspondan al segundo grupo. Mientras que en el segundo se considerarán a todos aquellos que ocupen técnicas inteligentes para la navegación.

### **2.2.2 Sistemas de Control.**

En el presente apartado se exponen algunas técnicas de control de navegación, aun cuando no guardan relación directa con el tema de investigación, en el ámbito de la metodología a ocupar, aportan para contrastar con las técnicas inteligentes que se expondrán en el siguiente apartado, sirviendo de referencia frente a la navegación misma.

Las metodologías basadas en teoría de control, fueron las primeras utilizadas para la navegación autónoma, y aún se sigue investigando en este campo, las cuales se han mejorado en la aplicaciones de nuevas estrategias con la inclusión de funciones matemáticas y operatoria matricial y vectorial, esto ha sido posible gracias a los avances en el área de la informática y computación, proveyendo lenguajes y compiladores que permiten realizar operaciones más complejas, y la mejora en las microtecnologías, las cuales permiten introducir estos programas en pequeños chips, de alto poder de procesamiento, velocidad y memoria.

Estas metodologías basadas en la teoría de control, apuntan a controlar al robot móvil, en función de un circuito de lazo cerrado, en el cual

constantemente se corrige el error generado frente a la ruta trazada por el modelo bajo el cual fue diseñado. Este modelo se podría dividir en tres grupos, los cuales no necesariamente son disjuntos, pues en varios trabajos se utilizan en forma sinérgica, esta división se puede realizar en función del *cómo* generan la ruta del robots.

#### 2.2.2.1 Guiado Externo.

Una metodología comúnmente ocupada, especialmente para líneas de transporte en la industria, es la de proveer al robot móvil de un sistema de guiado externo, lo cual permite al robot poder seguir una trayectoria ya definida.

Los robots que navegan de esta forma, se pueden catalogar en tres tipos. Trenes sin conductor, que son vehículos capaces de tirar varios remolques y su función recae en mover grandes cargas por almacenes. El segundo tipo son los traslapes autónomos, el cual genera una sinergia hombre máquina, ya que el operador humano guía la operación de izaje de carga paletizada, luego programa su destino y el vehículo de forma autónoma se dirige a la zona señalada. Por último están los portadores de unidad de carga, los cuales cumplen la función de transportar la carga de una estación a otra, generalmente utilizados para cadenas de montaje. (Medina Varela, Restrepo Correa, & Cruz Trejos, 2009)

Pero para poder realizar estas tareas, este tipo de robots móviles debe conocer la ruta que debe seguir, y es aquí donde entra el guiado externo; el cual requiere de elementos externos que guíen y orienten al robot, estos comúnmente son tres tipos:

**Cables enterrados**, este tipo de solución, ocupa cables de guía enterrados, los cuales son alimentados con generadores de frecuencia, por lo que los robots que utilizan esta metodología, poseen bobinas a ambos lados de él, éstas actúan como sensores para medir el campo magnético que genera el cable. Por lo cual, si el robot se desvía de la ruta definida el campo magnético medido en sus bobinas cambiará, permitiendo corregir el rumbo. El inconveniente que presentan estos sistemas, se acota a las zonas que poseen ruido eléctrico, lo cual afecta las lecturas de los sensores. (Medina Varela, Restrepo Correa, & Cruz Trejos, 2009)

**Banda reflectante o de pintura**, la cual se basa en el seguimiento de una línea trazada en el suelo, la cual demarca el camino al robot, también se pueden incluir marcas codificadas sobre éstas, para poder determinar la posición del robot durante su recorrido como se describe en (Uhrhan, Jaramillo, Plata, Buss, & Mojica, 2000), la desventaja de esta técnica es el desgaste que presenta, además de necesitar mantenerse limpia y ser contantemente re pintada. (Medina Varela, Restrepo Correa, & Cruz Trejos, 2009)

**Balizas**, a diferencia de los dos métodos anteriores, este último no utiliza rutas definidas, ya que simplemente se disponen por todo el entorno estas balizas, las cuales son identificadas por el robot para guiarse y orientarse. Las balizas en si pueden ser de varios tipos, como códigos de barra instalados en las paredes, los cuales son leídos y procesados. Pero en definitiva, las balizas permiten una mayor flexibilidad en la navegación de los robots ya que no existe una ruta trazada física, debido a que ésta se crea por software. (Medina Varela, Restrepo Correa, & Cruz Trejos, 2009)

### 2.2.2.2 Planificación de Trayectorias.

A diferencia de la metodología anteriormente mencionada, ésta puede ser aplicada en todo tipo de ambientes, pero con la salvedad de poseer previamente el mapa, ya que esta metodología se basa en generar un plan de navegación, en el cual la trayectoria es optimizada, es decir, encontrar la combinación de segmentos que permitan obtener la ruta más corta.

La optimización de ruta, se logra al poseer una posición inicial desde donde comienza el recorrido del robot y una final donde termina, y se pretende encontrar alguna ruta por la cual un robot pueda desplazarse de forma que no sufra colisiones con los obstáculos, y al mismo tiempo el recorrido seleccionado a seguir posea la menor distancia en comparación al resto de las posibles soluciones.

Para lograr esto, se emplean distintas metodologías de control, las de mayor utilización en robots móviles operan con sistemas PID. Para controlar el error generado con la trayectoria trazada; como en (Gutiérrez Zea, Jurko Vásquez , & Zuluaga, 2000) donde se construyó un robot subacuático que posee los comportamientos de mantenerse a una profundidad determinada o seguir una ruta definida por un operador humano, para lograr que el robot pudiera realizar estas tareas se le incorporó un sistema PID, el cual le permite de forma progresiva y suavizada corregir el error entre su posición y la trayectoria trazada.

### 2.2.2.3 SLAM.

SLAM es otra metodología que desarrolla el paradigma para la navegación móvil autónoma, a pesar de ser muy similar al caso anterior es necesario comentarlo aparte, en general esta metodología también apunta a la optimización de rutas y navegación planificada, pero la diferencia recae en que se trabaja sin mapas del entorno, y es el mismo robot quien mientras explora comienza a generar el mapa del entorno.

La contracción SLAM proviene de las siglas en inglés de “*Simultaneous Localization and Mapping*”, lo que se presenta como un paradigma recursivo entre localización y mapeo; el robot se sitúa y mapea de forma simultánea, permitiendo generar mapas de lugares no explorados y planificar de forma óptima la trayectoria a zonas del entorno ya visitadas y por ende mapeadas.

Otra característica importante de esta metodología, es que el robot puede decidir cuál será su propio objetivo, o sea, mientras explora el entorno puede determinar si existen hitos importantes que desee realizar o si ya están todas las tareas posibles realizadas; un ejemplo de ello, se puede observar en (Auat Cheeín, di Sciascio, & Carelli, 2008) donde esta técnica es aplicada y con la tarea de explorar las zonas de mayor interés, determinando si el sector es o no conocido, por ende el robot móvil se desplaza por el ambiente en busca de zonas no exploradas, trazando rutas óptimas entre las zonas ya mapeadas.

#### 2.2.2.4 Algoritmos Matemáticos.

En las tres metodologías anteriormente mencionadas, se excluyeron intencionalmente la condición de que el robot se encontrase con algún obstáculo no previsto y la confección de su ruta en función del entorno.

Cuanto un robot móvil, realiza una planificación de ruta, o se encuentra con un obstáculo no previsto, debe reaccionar de forma que evite colisionar con él, lo cual genera que éste desvíe su trayectoria inicial. Para corregir esta situación, deberá volver a su ruta, en el caso de los robots con sistemas de guías. En la situación que la trayectoria sea planificada deberá crear una nueva ruta.

Estas operaciones, se realizan con la ocupación de funciones matemáticas, vectoriales y matriciales. Que le permiten al robot móvil, crear una ruta nueva para lograr su objetivo sin afectar significativamente a su desempeño, en algunos casos tomando la actual posición como nuevo punto inicial, y en otros casos generando una trayectoria corta que le permita volver a su ruta predefinida.

Existe un gran número de operaciones matemáticas, vectoriales y matriciales aplicadas al campo de la robótica móvil, y debido a que este proyecto de tesis aborda tan sólo la temática de navegación inteligente reactiva, se presentaran sólo algunas investigaciones a modo de ejemplo, intentando generalizar el comportamiento comúnmente realizado por los robots móviles al evadir obstáculos.

En el caso de los robots que navegan con un sistema de guiado externo, luego de encontrarse con un obstáculo que interfiere con su recorrido establecido, realizan acciones evasivas acordes con una operatoria previamente programada. Un ejemplo de esto sería esquivar hacia el lado más cercano del obstáculo, tomando como referencia la ruta predefinida (Uhrhan, Jaramillo, Plata, Buss, & Mojica, 2000).

Pero luego de realizar esta acción evasiva debe volver a la ruta definida por la guía, lo que generalmente efectúan, trazando una curva a través de alguna función matemática. Por ejemplo, en (Uhrhan, Jaramillo, Plata, Buss, & Mojica, 2000), el robot, luego de realizar una acción evasiva realiza una curva generada por interpolación polinomial de splines cúbicos, lo cual genera una curva suavizada que le permite volver a su ruta inicial.

En el caso de los robots que planifican su ruta, sin la necesidad de guías externas, generan trayectorias que les permiten navegar por el entorno de forma óptima, al implementar funciones matemáticas con la información del entorno, de esta forma pueden trazar una ruta conformada por segmentos y/o curvas que genere una trayectoria de longitud mínima para lograr su objetivo, como lo es en el caso de (Pedroza Reyes, Sanches Lopez, & Rodriguez Marmolejo, 2007), en el cual ocupan curvas paramétricas del tipo Hermite, ya que este tipo de curvas poseen un punto inicial y final con sus respectivas orientaciones, generando una familia de curvas suaves entre ambos puntos.

### ***2.2.3 Sistemas Inteligentes.***

Otro enfoque para la navegación autónoma es dotar al robot, mediante el uso de las redes neuronales y la lógica difusa, de la inteligencia suficiente que

le permita reaccionar en forma oportuna ante la aparición de obstáculos no previstos, sin que por ello pierda de vista su tarea objetivo, realizando los cambios de ruta en forma eficiente, estas técnicas permiten aumentar la capacidad de adaptarse a entornos cambiantes y debido a su capacidad de aprendizaje, establecer patrones de comportamiento, por su capacidad de generalizar conceptos.

Estas técnicas permiten hacer robots que sean capaces de realizar las actividades relacionadas con la navegación, es decir, pueden desplazarse, evadir obstáculos e incluso atravesar puertas, lo anterior sin la necesidad de poseer un plano o instalar sistemas guías en el entorno. A su vez, estas tecnologías permiten a los robots autónomos ser altamente adaptables a cualquier ambiente, y a partir de consideraciones básicas pueden generar comportamientos más complejos.

#### 2.2.3.1 Redes Neuronales.

Las redes neuronales son una de las técnicas de la inteligencia artificial utilizadas en numerosos campos de investigación, entre ellos se encuentran la robótica móvil, gracias a las ventajas que presentan, como son el procesamiento paralelo, la generalización y el bajo costo computacional en su ejecución.

Cabe destacar que la utilización de esta técnica de inteligencia artificial, es aplicable a distintas áreas de la navegación y su topología de diseño no es única, al contrario podemos apreciar diferentes morfologías. Las tareas en que más comúnmente se aplica esta tecnología son fusión sensorial, clasificación de superficies, generación de trayectorias y la navegación del robot.

**Fusión sensorial**, esta estrategia se aplica generalmente, en robots móviles que poseen más de un tipo de sensores de distancia, pero cada uno tiene carencias que él o los otros sensores son capaces de suplir. Se entrena una red, para que al tomar cada tupla de medición genere una única salida de mayor confiabilidad y certeza. Disminuyendo de esta forma los errores de lectura que generarían cada sensor por separado. Como en el caso de (Zamora, Tomás-Balibrea, Martínez, & Skarmeta) en el que realizan esta operación entre un sensor ultrasónico y uno infrarrojo, generando una medida más precisa que la del ultrasonido y de mayor alcance que la del infrarrojo.

**Clasificación de superficies**, otra manera de apoyar la navegación autónoma, generalmente la conducida por *lógica difusa*, es utilizar una red neuronal que permita al robot, en función de sus señales de entrada, poder clasificar el entorno a través de las lecturas de sus sensores y alimentar al controlador de navegación, para poder aportar mayor información al proceso de decisión de la navegación, en (Rios G., Bueno L., & Sanchez A., 2008) se presenta un clasificador de superficie que apoya directamente al controlador de navegación a través del reconocimiento de superficies como paredes y esquinas.

**Generación de trayectorias**, otra aplicación para las redes neuronales, es la de generar trayectorias, es decir, en función del ambiente inmediato, pueden tomar la decisión de optar por una nueva ruta más segura. Como es el caso de (Rios G., Bueno L., & Sanchez A., 2008), que además de clasificar las superficies existentes en las inmediaciones del robot, determina la ruta más óptima para que él la siga.

**La Navegación**, se habla de navegación por redes neuronales cuando éstas se encargan de realizar todas las operaciones relacionadas con la navegación, como determinar la ruta a seguir, la evasión de los obstáculos y controlar el resto de los periféricos necesarios para su funcionamiento. Por ejemplo en (Gonzales Acevedo & Mella Castañeda, 2007) se entrena una red neuronal de base radial, la cual es capaz de decidir las acciones necesarias para lograr llegar al objetivo y evadir los obstáculos, o como en el caso de (Cordoba G.) en el que se inserta un control neuronal como operador inteligente de un vehículo aéreo no tripulado, el cual se encarga de controlar todas las variables y sensores de sus componentes abordo y llegar al objetivo definido.

#### 2.2.3.2 Lógica Difusa.

La lógica difusa es una de las técnicas de la inteligencia artificial que se basa en trabajar con conjuntos que no tienen una definición clara en sus límites y también con variables lingüísticas, estos dos conceptos abren un abanico de utilidades, dado que antes de la concepción de esta técnica no se podía dar una connotación matemática a una variable lingüística imprecisa, como los conceptos de mucho o cerca, entre otros.

La aplicación de la lógica difusa a la robótica móvil autónoma, ha permitido solucionar algunos problemas que en general presentan dificultades al tratarlo con la matemática tradicional, como lo es el navegar en un entorno real y dinámico, del cual se posee información incompleta, aunque exista un mapa, y los sensores pueden ser afectados por las condiciones ambientales propias del entorno.

Dentro de las características que es importante destacar de la aplicación de esta técnica inteligente a la robótica móvil, es la tolerancia que presenta frente a los errores del sensado, debido a su naturaleza difusa estos errores no presentan una gran magnitud de influencia en las decisiones que se toman, además al estar basado en reglas que funcionan sobre variables lingüísticas, permite un diseño más intuitivo del controlador de navegación, ya que las mismas le otorgan un sentido natural. Y por último y no menos importante, permite generar transiciones paulatinas en los cambios de estados, lo cual evita que el robot presente movimientos bruscos o cambios de velocidad muy violentos.

Aunque en la literatura se puede ver que la aplicación de esta técnica apunta hacia lo mismo, que es controlar la navegación y evasión de obstáculos, son muchos los trabajos que han utilizado esta técnica, los que se consideran pertinentes destacar son (Mosso Solano & Vivas Alban, 2000), (Gonzales Acevedo & Mella Castañeda, 2007), (Gómez Skarmeta, Martínez Barberá, & García López, 2001), (Baturone & Gersnoviez, 2010) y (Gersnoviez & Baturone, 2008), ya que en estos trabajos, a pesar de ocupar la misma técnica, exponen diferentes aspectos de ella, por ejemplo en (Mosso Solano & Vivas Alban, 2000), se presentan las formas para generar un modelo difuso, en una publicación que relata las experiencias obtenidas por alumnos de un Master. En las otras publicaciones se presentan distintas formas de abordar el problema en función a cómo generar el controlador y las reglas definidas, como también las funciones de pertenencia que se ocupan, lo cual representa un factor importante el momento de diseñar un control difuso.

#### *2.2.4 Sistemas basados en campos de potencial.*

Otro modelo de solución utilizada para sortear este tipo de problemas, radica en la ocupación de campos potenciales artificiales, este tipo de solución se caracteriza por ser simple y agilidad de cálculo, permitiéndole la navegación en tiempo real (Mora & Tornero, 2007).

Este campo de potencial se construye con la superposición del campo de atracción originado por la posición final deseada o meta, y los campos de repulsión generados por los obstáculos. (Rodríguez, 1998) (Mora & Tornero, 2007). Y la forma más común de generar la trayectoria del robot móvil, es suponer que existe, un resorte que une al robot con su meta, sin embargo, esta alternativa, genera que la fuerza de atracción que crea la meta, disminuya a medida que el robot se acerca a su destino, esta es la razón, por la cual la mayoría de los autores, solamente utilizan la dirección de esta fuerza resultante pero mantienen constante la magnitud. No así, con el potencial repulsivo, ya que esta magnitud debe tener, un mayor valor mientras esté más cerca a los obstáculos (Rodríguez, 1998).

La evasión de los obstáculos móviles, generalmente asumiendo la posición de los obstáculos de forma determinísticas en un instante determinado. Pero también existe otro planteamiento, que debe ser considerado. En el cual se estima la posición de los obstáculos móvil a través de un modelo matemático que genera como resultado un elipsoide, donde el centro representa el valor estimado de la posición, mientras que, el volumen de esta, indica zona probable en que se encuentre (Mora & Tornero, 2007).

### *2.2.5 Conclusiones del Capítulo*

Después de investigar el marco teórico y revisar numerosas publicaciones, se posee un nivel mayor de entendimiento y conocimiento de la robótica móvil y de las estrategias que se utilizan para el desarrollo de ésta.

Como se puede ver en este capítulo las soluciones planteadas a través de sistemas guías externos, requieren por obligación ejecutarse en el entorno para el cual fueron diseñados, en caso de necesitar implementar este tipo de sistemas en un entorno distinto se presenta la complejidad de tener que implementar todo el sistema guía y reprogramar al robot para el nuevo esquema.

Los sistemas de control basados en planificación de trayectorias son muy óptimos en cuanto a las rutas generadas y la complicación de implementarlos en otros entornos no es tan alta como los sistemas de guía externa, pero demandan un elevado costo computacional y presentan un alto nivel deliberativo, teniendo que incluir controles reactivos para poder responder frente a la naturaleza dinámica del entorno en que operan.

Pero a diferencia de estos dos tipos de sistemas de control, los que están basados en inteligencia artificial presentan un nivel adaptativo excepcional, pudiendo incluso realizar sus tareas en cualquier entorno, que obviamente sus sensores le permitan.

A pesar de no generar rutas siempre óptimas logran resultados efectivos, en cuanto a trayectorias y cumplimiento de metas, lo anterior a un bajo costo

computacional y en tiempo real, permitiéndoles de esta forma reaccionar adecuadamente a la naturaleza dinámica del entorno.

Por último, el modelo de campos de potencial artificial, permite una navegación segura por el entorno, además de estos, por las características intrínsecas de este modelo, el robot móvil se ve influenciado por los campos resultantes a modificar su velocidad de desplazamiento. Sin embargo, la presencia de vanos o discontinuidades en las paredes, no representaran un factor de repulsor, mas por el contrario puede que sea una espacio deseable para la navegación.

### **3 INVESTIGACIÓN PROPUESTA.**

#### **3.1 Problemática**

En el campo de la robótica móvil, existen numerosos trabajos que emplean diferentes técnicas para lograr una efectiva evasión de obstáculos, pero varios casos sacrifican la velocidad de desplazamiento en consideración al tiempo de respuesta y en beneficio de la resolución del sensado. Además, esta configuración de velocidad permite despreciar los efectos de las fuerzas dinámicas que afectan al robot móvil durante la navegación.

Este tipo de soluciones asume a priori que siempre existirá la presencia de obstáculos en la trayectoria del robot. Por este motivo, se encuentra que en la mayoría de los desarrollos actuales en robótica móvil el desplazamiento es a velocidad constante y/o reducida.

La mayoría de los robots móviles, no cuenta con un sistema energético alternativo que le permita recargar sus baterías, lo cual limita su tiempo de operación, y debido a que en general trabajan a una baja velocidad de desplazamiento, reducen considerablemente su área posible de operación y alcance, restringiendo de esta forma el número posible de tareas que puedan ejecutar.

#### **3.2 Preguntas de Investigación.**

1. ¿Es factible que al incorporar un variador de velocidad, siga eludiendo los obstáculos de forma eficiente?

2. ¿La incorporación de un variador de velocidad permitirá una mayor área de operación?
3. ¿Se mejora la eficiencia en cuanto a tiempo de operación?
4. ¿Cómo afectará a la navegación del robot móvil la variación controlada de la resolución de sensado?
5. ¿Se logra un aumento significativo de la velocidad de desplazamiento?
6. ¿Las técnicas de ingeniería neuronal y lógica difusa serán apropiadas para lograr una efectiva evasión de obstáculos y variación de velocidad?

### **3.3 Hipótesis de Trabajo**

A través de las técnicas de ingeniería neuronal y lógica difusa se pretende lograr, un controlador inteligente que permita al robot móvil realizar acciones de evasión tempranas frente a obstáculos presentes y emergentes, variando su velocidad de desplazamiento, en función de la composición del entorno inmediato y distante. Con lo cual se conseguirá una mayor eficiencia en razón del tiempo de ejecución de una tarea.

### **3.4 Objetivo General y Objetivos Específicos**

#### ***3.4.1 Objetivo General***

Utilizando técnicas de *Ingeniería Neuronal* y *lógica difusa*, generar un controlador inteligente adaptativo, que en ambientes estructurados y dinámicos, dote al robot móvil de la capacidad de variar su velocidad de desplazamiento, en virtud de la composición del entorno cercano y lejano, para la evasión de obstáculos presentes y la detección de vanos y pasillos, para evitar colisiones con posibles obstáculos emergentes.

### **3.4.2 Objetivos Específicos**

1. Realizar acciones preventivas tempranas para evasión de obstáculos.
2. Dotar al robot de la capacidad de detectar vanos y pasillos, para realizar acciones preventivas frente a la posibilidad de aparición de un obstáculo emergente.
3. Mejorar el tiempo de ejecución de una tarea mediante un controlador que le permita incrementar su velocidad, en ambientes libres de obstáculos.
4. Priorizar el cambio de ruta en vez de la desaceleración, en la evasión de obstáculos.
5. Detección de caminos sin salida, para evitar bucles infinitos.

### **3.5 Resultados Esperados**

1. Controlador Neuronal capaz de detectar e identificar distintos hitos (paredes, obstáculos, vanos y pasillos) en el entorno cercano y lejano del robot móvil.
2. Controlador Difuso capaz de variar la velocidad de navegación del robot móvil, en función de las entradas de control que permite la evasión de obstáculos.
3. Integración de ambos controladores.
4. Realizar pruebas en un espacio simulado estructurado y dinámico.
5. Publicación de resultados en revistas y/o congresos pertinentes.

### **3.6 Justificación**

La realización de este proyecto, tiene como propósito y motivación, el postular y demostrar empíricamente una nueva alternativa de solución a los

problemas de robótica móvil, ya que en general las técnicas revisadas en la literatura, trabajan a velocidad constante y/o reducida. Lo anterior, para tener una clara percepción del entorno, y evitar colisionar con los obstáculos presentes en el medio.

El planteamiento aquí propuesto apunta a dotar al robot de un comportamiento un grado más inteligente, ya que él podrá decidir, si el entorno es lo suficientemente *Seguro*, como para aumentar la velocidad de desplazamiento o disminuirla en caso contrario.

El destacar el término *Seguro* tiene dos intenciones, por una parte la posibilidad que al no existir obstáculos en el entorno del robot, no limite la velocidad de desplazamiento de éste. Y la segunda intención, que posee una prioridad mayor que la primera, es que en un entorno libre de obstáculos, existe la incertidumbre que a través de un vano o por un pasillo aparezca fortuitamente un obstáculo nuevo en el entorno.

He aquí el nuevo nivel de inteligencia que se pretende dotar al robot, o sea, que ya no sólo evada obstáculos, sino que también pueda variar su velocidad, en forma positiva o negativa, y que además entienda que un vano o pasillo significa un potencial obstáculo, el cual puede o no materializarse.

## 4 DISEÑO Y EXPERIMENTACION

### 4.1 Introducción

Este capítulo describe el proceso de diseño, implementación y experimentación de la solución.

El proceso de desarrollo, se enmarca en dos etapas. La primera es el diseño de la solución, donde se establecieron las técnicas y parámetros a utilizar. Segundo, la experimentación, la cual se realizó en un entorno simulado, para de esta manera eludir variables no controladas que posee el entorno real.

Dentro del proceso de investigación se evaluaron diferentes plataformas robóticas existentes, con el objetivo de seleccionar la más idónea para el desarrollo propuesto, y al mismo tiempo que sirviese de base de conocimiento para desarrollos futuros del Departamento de Ingeniería de sistemas y Computación.

Se seleccionó la plataforma MobileSim debido a que esta posee un simulador de robots móviles que permite realizar simulaciones muy cercanas a la realidad. Además de poseer una amplia gama de robots (Mobile Robots, 2012).

Junto con esto, MobileSim permite trabajar con lenguajes de programación de alto nivel, como C++ y JAVA, para la confección de algoritmos en el robot. Esto es posible gracias a la Liberia ARIA.

La librería ARIA es una interfaz de comunicación entre el programa, escrito en un lenguaje de alto nivel, y el robot. Esta librería puede ser ocupada en diferentes sistemas operativos, lo cual vuelve flexible al diseño e implementación de los sistemas on board.

Un programa anexo al MobileSim, el cual permite la creación de escenarios es Mapper. Estos son utilizados por el simulador como entornos para la navegación del robot (Mobile Robots, 2012).

De los robots que posee el simulador MobileSim, se optó por utilizar el robot Pioneer 3-DX. Este robot móvil diferencial es muy versátil y utilizado en la realización de diferentes experiencias de laboratorios, debido a que es posible anexarles distintos accesorios para operación. Físicamente este robot cuenta con 2 ruedas motrices y una volante, 3 baterías como fuente de alimentación que le permiten una autonomía de 8 a 10 horas, un anillo ultrasónico de 16 sensores, puede alcanzar una velocidad de 1.2 metros por segundo y transportar cargas de hasta 8 kilogramos (Mobile Robots, 2012).

El Pioneer 3-DX cuenta con un sistema de locomoción diferencial, con dos ruedas laterales controladas por motores independientes, lo cual permite el desplazamiento lineal, realizar arcos y rotar sobre su propio eje. Y una rueda volante que cumple la función de estabilizar al robot (Ruiz Del Solar & Salazar).

El anillo ultrasónico de 16 sensores que posee el Pioneer 3-DX está configurado con 8 sensores ubicados en la mitad frontal y los otros 8 ubicados en la mitad trasera, como se muestra en la Figura 4-1.

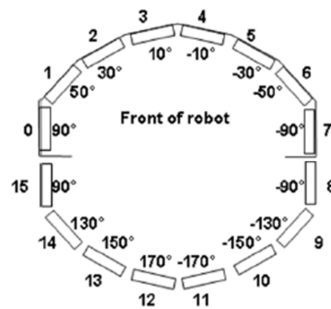


Figura 4-1 Distribución del anillo ultrasónico.

Con formato: Revisar la ortografía y la gramática

Los sensores ultrasónicos funcionan emitiendo una onda de sonido a una frecuencia de 40 kHz, la cual es superior a la del rango audible por el humano que va desde 16 Hz hasta 20 kHz.

Este sistema posee un emisor de ultrasonido y un receptor, el cual recibe la reflexión del pulso emitido. La medición de la distancia, se calcula en función de la diferencia del tiempo entre la emisión del pulso y su recepción, como se muestra en Ecuación 4-1, donde “V” es la velocidad del sonido en el aire y “t” es la diferencia del tiempo entre la emisión y la recepción.

$$d = \frac{1}{2} V \cdot t$$

Ecuación 4-1

Los sensores ultrasónicos, son ampliamente utilizados en distintos tipos de experimentos y configuraciones de robots. Sin embargo poseen deficiencias, las cuales no pueden ser ignoradas al momento de trabajar con ellos:

1. Los sensores ultrasónicos generan un área de medición con forma cónica. El eco que recibe indica la presencia del objeto más cercado dentro del área. Sin embargo no precisa su posición angular dentro del cono, es por esto, que se asume que se encuentra en el eje central de este.
2. Otro error de medición importante, recae en las características físicas del objeto que genera el eco, si las irregularidades de este son comparables a la longitud de onda, la energía acústica refleja disminuye.
3. Algunos sensores ultrasónicos usan el mismo transductor como emisor y receptor. Después de generar un pulso, se espera un determinado tiempo para que las vibraciones en el sensor se disipen y así, el sensor esté preparado para recibir el eco generado por el obstáculo. Esto genera que exista una distancia mínima de medición, en función del tiempo de relajación del sensor. Por lo tanto, todo valor recibido por debajo de esta medición, se considera como la marca mínima.
4. La velocidad de propagación en el aire de la onda ultrasónica se ve fuertemente afectada por la temperatura, ya que esta modifica la densidad del aire. En la Ecuación 4-2 se muestra la fórmula para calcular la velocidad de propagación del sonido en el aire a distintas temperaturas, donde  $V_{so}$  es la velocidad del sonido a  $0^{\circ}$  C, y T es la temperatura en grado Kelvin.

$$V_s = V_{so} \cdot \sqrt{1 + \frac{T}{273}}$$

Ecuación 4-2

5. Otro factor que afecta las mediciones, son los denominados ecos falsos, estos se generan cuando un pulso ultrasónico, incide en varias superficies antes de volver al transductor, generando mediciones erróneas.
6. Las ondas ultrasónicas son regidas por las leyes de reflexión de ondas, es decir, la obtención del eco del objeto dependerá de la orientación de la superficie con respecto al eje del sensor.

## 4.2 Diseño de la solución

En esta etapa se describe el proceso de diseño del controlador Neuro – Difuso. Este fue diseñado para poder identificar hitos (paredes, obstáculos y puertas) en el entorno y variar su velocidad en función a la presencia de estos.

Se seleccionó la técnica de lógica difusa debido a que permite trabajar con ambigüedad y cambios suaves de estados. Mientras que las redes neuronales, tienen la capacidad para la detección de patrones y el bajo costo computacional en su implementación. Por esto se utilizaron ambas técnicas en forma sinérgica.

El controlador Neuro – Difuso integrara en forma sinérgica ambas técnicas de inteligencia artificial. En la Figura 4-2, se ilustra el esquema completo del sistema.

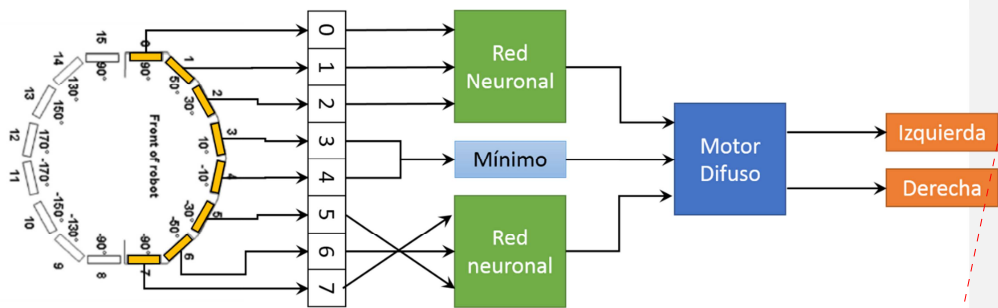


Figura 4-2 Esquema Sistema Completo.

**Con formato:** Revisar la ortografía y la gramática

A continuación se presenta el proceso de diseño e integración de ambas técnicas ya mencionadas, la selección de las métricas a utilizar y la implementación de estas.

#### 4.2.1 Redes neuronales

Se utilizó redes neuronales con el fin de identificar los hitos y en función de estos, generar una transformación del entorno que permita al robot, una navegación más segura.

**Comentario [FCS1]:** Cambiar redacción del párrafo

El tipo de red seleccionada es una Feed-forward con entrenamiento Backpropagation, ya que este tipo de red es ampliamente utilizada en la detección de patrones, como en este caso (Pino, Gómez, & de Abajo, 2001).

La RNA (red neuronal artificial) diseñada, posee la siguiente topología:

1. Capa de entrada: la cantidad de neuronas en esta capa está definida por el número de entradas que, para este caso, es la información proveniente de 3 sensores, que representan la información lateral del robot.
2. Esta RNA, está diseñada con dos capas ocultas

- a. Capa Oculta 1: es esta capa se compone de 9 neuronas, pese a que existen reglas que recomiendan no superar el doble de las neuronas de entrada (Flórez & Fernández, 2008), la red no lograba generalizar los conceptos presentados.
  - b. Capa Oculta 2: en esta capa se siguió el teorema anteriormente mencionada, y se implementó con 6 neuronas.
3. Capa de Salida: Ésta se define por la cantidad de salidas esperadas del sistema. La configuración de la solución, posee solo una salida, ya que la red trabajara con la información lateral del robot, para así alimentar al controlador difuso.

En la Figura 4-3 se ilustra gráficamente la red neuronal diseñada.

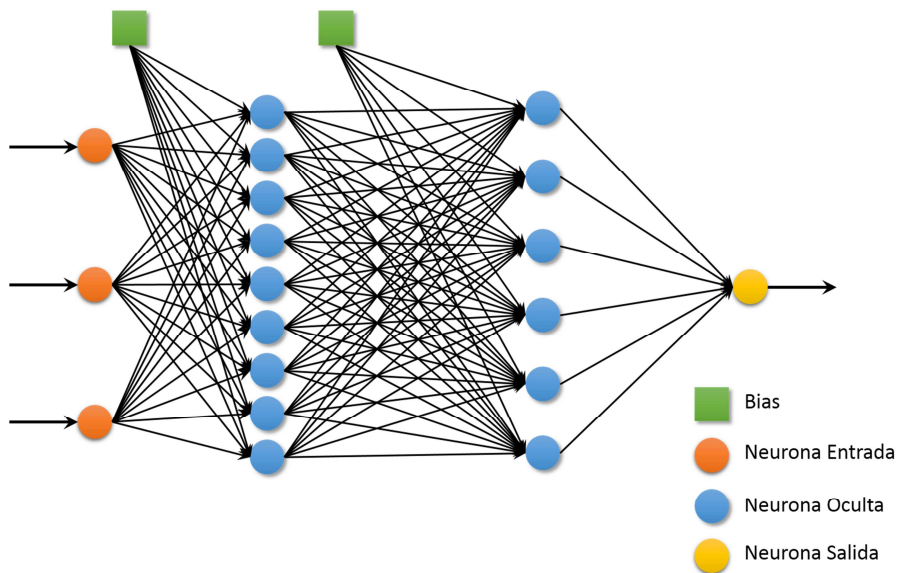


Figura 4-3 Red neuronal.

Para seleccionar la cantidad de patrones que fueron presentados a la red en la fase entrenamiento, se determinó mediante el número total de neuronas de la red dividido por el inverso del error esperado (Flórez & Fernández, 2008). En este caso son: 19 neuronas y un error esperado del 0.5%. El volumen del conjunto de entrenamiento resultante es de 3.800 entradas.

Para entrenar la red neuronal se utilizó el programa SPSS Clementine 8.1, el cual, dentro de sus múltiples funcionalidades permite el entrenamiento de las redes neuronales. El programa da la posibilidad al usuario de seleccionar los parámetros de la topología, y determina de forma automática la función de activación.

Para poder operar, las RNA requieren que sus valores de entrada estén en el alfabeto [0,1]. Por esto se requiere transformar las lecturas de los sensores ultrasónicos a este alfabeto. Por esta razón las salidas de la RNA, que están en el alfabeto [0,1] requieren ser destransformadas para poder ser utilizadas, en este caso por el Controlador Difuso.

Para la transformación de los valores de entrada, se ~~multiplio~~ multiplicó por el inverso del valor máximo alcanzable, 5.000.

Para el caso de salida, el valor obtenido de la red se multiplicó por 5.000.

La función de activación determinada por el SPSS Clementine 8.1 fue la logística. La cual se expresa en la ecuación:  $f(x) = 1/(1 + e^{-x})$ .

La RNA fue diseñada para determinar patrones en función de las discontinuidades del entorno obtenidas por el anillo ultrasónico. Dado esto,

proporciona valores de salida (valores de distancia) que permiten al robot navegar de forma más segura. En la Figura 4-4 se ilustra esta transformación.

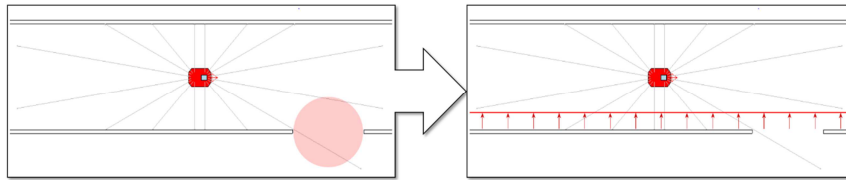


Figura 4-4 Transformación entorno.

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

#### 4.2.2 Lógica difusa

La lógica difusa se utilizó para controlar el comportamiento del robot, es decir, implementar el variador de velocidad suavizado.

El primer paso para la aplicación de esta técnica es determinar el universo de discurso, el cual para las entradas está acotado por los parámetros de lectura del sensor ultrasónico y para las salidas por los valores esperados.

Para el caso de las variables de entrada (Left, Front y Right) el universo de discurso está determinado por los valores máximos y mínimos resultantes de la red neuronal, los cual van desde 0 a 5 metros.

La variable lingüística, utilizada para la entrada fue “Distancia”, la cual se tipificó en 5 términos lingüísticos: muy cerca, cerca, medio, lejos y muy lejos.

Para el caso de las variables de salida (Left y Right) el universo de discurso está determinado por los valores esperados. Estos representan una ponderación de la velocidad máxima del robot, la cual, va desde -0.5 a 1.

La variable lingüística utilizada para la salida fue “Velocidad”, la cual se tipificó en 6 términos lingüísticos: extremo, muy lento, lento, normal, rápido, muy rápido.

Como las variables de entrada provienen de la red neuronal, hay que fusificarlos, y para esta tarea es necesario contar con funciones de pertenencia que calculen el grado de pertenencia de la entrada a los términos lingüísticos. Estas funciones también se utilizan para poder defusificar las variables de salida, es decir convertir el valor difuso resultante en un valor nítido.

Las funciones de pertenencia que se seleccionaron para determinar el grado de pertenencia de los valores (entrada y salida) a los términos lingüísticos fueron cuatro: triangular, S, Z y Gauss.

La función Z se utiliza para las variables del borde inferior, muy cerca y muy lento, esta función presenta una curva al aumentar en el universo de discurso. La función Z se define por la Ecuación 4-3 y en la Figura 4-5 se presenta un ejemplo.

$$Z[a, b](x) = \begin{cases} 0 & \text{si } x \leq a \\ 1 - 2 \left( \frac{x-a}{b-a} \right)^2 & \text{si } a < x \leq \frac{a+b}{2} \\ 2 \left( \frac{b-x}{b-a} \right)^2 & \text{si } \frac{a+b}{2} \leq x < b \\ 1 & \text{si } x \geq b \end{cases}$$

Ecuación 4-3

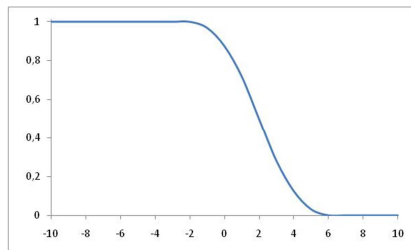


Figura 4-5 Función de Pertenencia Z

La función triangular, fue utilizada para definir los conceptos intermedios, cerca, lejos, lento y rápido. Y como caso especial también para el concepto extremo. Tiene como característica permitir transiciones lineales entre los estados. La función triangular se define por la Ecuación 4-4 y en la Figura 4-6 se presenta un ejemplo.

$$Tri_{abc}[a, b, c](x) = \begin{cases} \frac{x-a}{b-a} & \text{si } a < x \leq b \\ \frac{c-x}{c-b} & \text{si } b < x < c \\ 0 & \text{en otro caso} \end{cases}$$

Ecuación 4-4

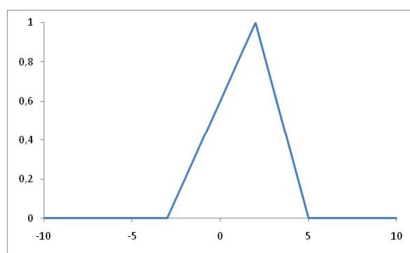


Figura 4-6 Función de Pertenencia Triangular.

La función gaussiana, fue utilizada para representar el concepto intermedio de las variables de entrada y salida, normal y medio. Como esta función tiene forma de campana, permite transiciones suaves hacia ambos lados del centro del universo de discurso. La función gaussiana se define por la Ecuación 4-5 y en la Figura 4-7 se presenta un ejemplo.

$$Gauss[c, \beta](x) = e^{-\left(\frac{x-c}{\beta}\right)^2}$$

Ecuación 4-5

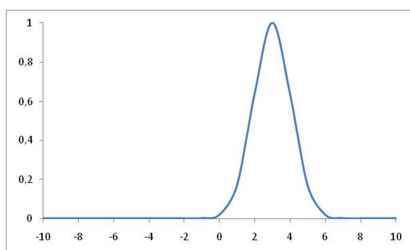


Figura 4-7 Función de Pertenencia Gaussiana.

Por último, la función S es el reflejo de la función de pertenencia Z, y fue utilizada para representar los valores del borde superior de las variables lingüísticas, representando los conceptos de muy lejos y muy rápido. La función S se define por la Ecuación 4-6 y en la Figura 4-8 se presenta un ejemplo.

$$S[a,b](x) = \begin{cases} 0 & \text{si } x \leq a \\ 2 \left( \frac{x-a}{b-a} \right)^2 & \text{si } a < x \leq \frac{a+b}{2} \\ 1 - 2 \left( \frac{b-x}{b-a} \right)^2 & \text{si } \frac{a+b}{2} < x < b \\ 1 & \text{si } x \geq b \end{cases}$$

Ecuación 4-6

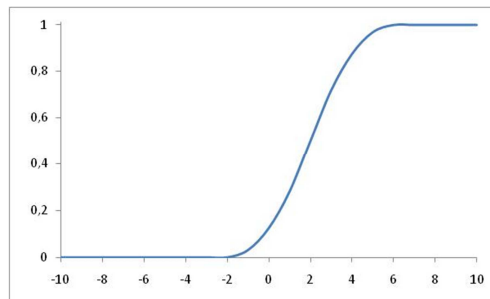


Figura 4-8 Función de Pertenencia S.

El sistema Difuso utilizado es el modelo Mamdani. Se seleccionó este modelo por la libertad que presenta al momento de seleccionar los métodos de fusificación, inferencia y defusificación. Permite también, tratar con facilidad las entradas y salidas del sistema.

La base de conocimiento está conformada por las funciones de pertenencia mencionadas anteriormente y por las reglas difusas que describen el comportamiento del robot.

En las Figura 4-9 se muestran las funciones de pertenencia utilizadas, en las variables de entrada y salida, para los términos lingüísticos en el universo de discurso.

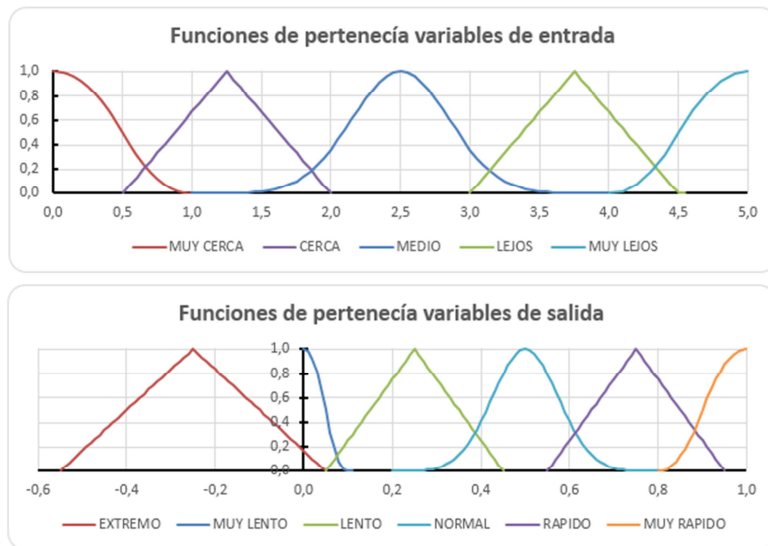


Figura 4-9 Funciones de pertenencia.

La cantidad de reglas para un sistema difuso, está determinada por la cantidad de variables de entrada y la tipificación de las variables lingüísticas. En este caso, se cuenta con 3 variables de entrada tipificadas en 5 términos lingüísticos, lo cual genera 125 reglas.

Las reglas se confeccionan de forma similar a una tabla lógica, tal de conseguir todas las combinaciones posibles. Las salidas se completan evaluando las entradas y en función de la respuesta deseada para esa condición. La Figura 4-10 muestra un fragmento de la tabla de reglas difusas. Para revisar en extenso esta tabla refiérase al punto 7.1.2, Anexos.

	ENTRADA			SALIDA	
	FRENTE	IZQUIERDA	DERECHA	IZQUIERDA	DERECHA
1	MUY CERCA	MUY CERCA	MUY CERCA	EXTREMO	EXTREMO
2	MUY CERCA	MUY CERCA	CERCA	LENTO	EXTREMO
3	MUY CERCA	MUY CERCA	MEDIO	LENTO	EXTREMO
4	MUY CERCA	MUY CERCA	LEJOS	LENTO	EXTREMO
5	MUY CERCA	MUY CERCA	MUY LEJOS	LENTO	EXTREMO
6	MUY CERCA	CERCA	MUY CERCA	EXTREMO	LENTO
7	MUY CERCA	CERCA	CERCA	LENTO	EXTREMO
8	MUY CERCA	CERCA	MEDIO	LENTO	EXTREMO
9	MUY CERCA	CERCA	LEJOS	LENTO	EXTREMO
10	MUY CERCA	CERCA	MUY LEJOS	LENTO	EXTREMO
11	MUY CERCA	MEDIO	MUY CERCA	EXTREMO	LENTO
12	MUY CERCA	MEDIO	CERCA	EXTREMO	LENTO
13	MUY CERCA	MEDIO	MEDIO	LENTO	EXTREMO
14	MUY CERCA	MEDIO	LEJOS	LENTO	EXTREMO
15	MUY CERCA	MEDIO	MUY LEJOS	LENTO	EXTREMO
16	MUY CERCA	LEJOS	MUY CERCA	EXTREMO	LENTO
17	MUY CERCA	LEJOS	CERCA	EXTREMO	LENTO
18	MUY CERCA	LEJOS	MEDIO	EXTREMO	LENTO
19	MUY CERCA	LEJOS	LEJOS	LENTO	EXTREMO
20	MUY CERCA	LEJOS	MUY LEJOS	LENTO	EXTREMO
21	MUY CERCA	MUY LEJOS	MUY CERCA	EXTREMO	LENTO
22	MUY CERCA	MUY LEJOS	CERCA	EXTREMO	LENTO
23	MUY CERCA	MUY LEJOS	MEDIO	EXTREMO	LENTO
	MUY CERCA	MUY LEJOS	LEJOS	EXTREMO	
	MUY CERCA	MUY LEJOS	MUY LEJOS	EXTREMO	

Con formato: Revisar la ortografía y la gramática

Figura 4-10 Reglas difusas modelo 5 variables lingüísticas.

El motor de inferencia difusa, de ahora en más FIS (Fuzzy Inference Solve), es el encargado inferir el comportamiento ad hoc en función de su base de conocimiento. Para realizar esto, el FIS evalúa la activación de cada antecedente y componen las reglas en función de su definición, la cual está presente en la base de conocimiento. Posteriormente este calcula el predicado de cada regla y realiza la unión de estos, obteniendo así un valor difuso resultante. Como el robot debe trabajar con un valor nítido es necesario desfusificar las variables de salida.

El método utilizado para la desfusificación, fue el método del centro de gravedad, este método tiene la ventaja de poseer un costo computacional de  $O(0)$ . Lo cual implica que este no depende de la complejidad de las funciones activadas o requiere ciclos para su cálculo.

El centro de gravedad se calcula como el cociente entre la suma ponderada de los centros de gravedad y su grado de activación, con la suma de todos los grados de activación. Como se muestra en la Ecuación 4-7.

$$\frac{y_1 \cdot Cg_1 + y_2 \cdot Cg_2 + \dots + y_n \cdot Cg_n}{y_1 + y_2 + \dots + y_n}$$

Ecuación 4-7

Gráficamente la Figura 4-11 muestra la defusificación la función de pertenencia Gaussiana para el término lingüístico normal de las variables de salida.

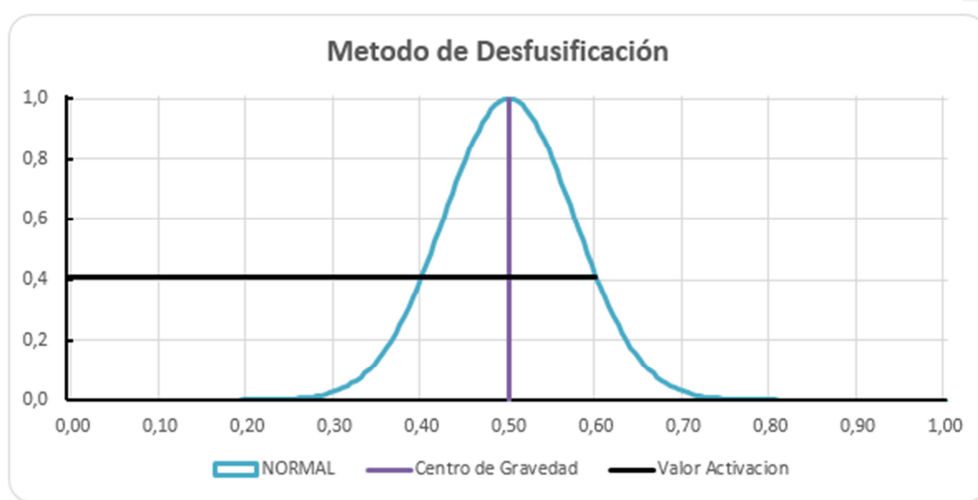


Figura 4-11 Método de defusificación.

### 4.2.3 Controlador Neuro – Difuso.

La implementación completa del sistema neuro – difuso, se puede apreciar en la Figura 4-12, en esta se presenta la conexión existente desde la entrada de los sensores, hasta la salida de velocidades ya procesadas.

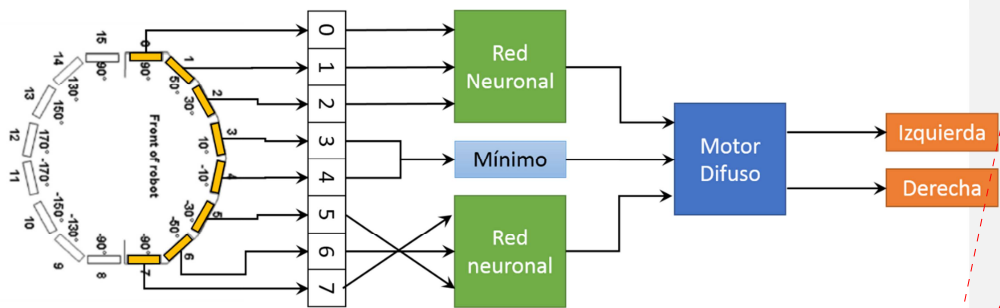


Figura 4-12 Esquema Sistema Completo.

Con formato: Revisar la ortografía y la gramática

Los sensores ultrasónicos de cada flanco del robot Pioneer 3-DX, miden la distancia existente entre el robot y su entorno 5 veces por segundo. Estos valores numéricos alimentan las dos redes neuronales.

Las redes neuronales procesan las lecturas provenientes de los sensores ultrasónicos, transformando el entorno sentido en uno más seguro. Esto significa, modificar las percepciones del robot 3-DX con respecto a su entorno. Como resultado de este proceso, la red neuronal alimenta al controlador difuso con valores de distancia modificados.

El controlador difuso, después de recibir las distancias modificadas evalúa, en función de su base de conocimiento, el ambiente presentado e infiere el comportamiento deseado para la condición actual. Como resultado de esto se obtienen la ponderación de la velocidad máxima a la cual el Pioneer 3-DX puede desplazarse.

Como resultado de este proceso el Pioneer 3-DX, debiese navegar por cualquier entorno que se le presente, siendo capaz de modificar su trayectoria en presencia de obstáculos, sin colisionar y variando su velocidad.

#### 4.2.4 Métricas

La información que se desea comparar es: seguridad, distancia, tiempo, suavidad de trayectoria, velocidad máxima y promedio

Se planteó generar un algoritmo que mejorase el tiempo y distancia recorrida en la navegación, redujese el consumo energético y al mismo tiempo generase una navegación segura. Dado esto, las siguientes métricas fueron seleccionadas:

1. **Medida de Seguridad-1 (SM1):** Es la distancia promedio desde el robot a los obstáculos, durante toda la navegación. Esta métrica tendrá un valor alto cuando la navegación haya ocurrido por una zona libre de obstáculos (Muñoz, Valencia, & Londoño, 2010).
2. **Medida de Seguridad-2 (SM2):** Es la distancia mínima Promedio desde el robot a los obstáculos. Esta métrica da una idea general de riesgo global existente durante la navegación (Muñoz, Valencia, & Londoño, 2010).
3. **Distancia Mínima (Min):** Es distancia mínima medida por cualquier sensor durante toda la trayectoria. Esta métrica entrega el riesgo máximo que se presentó durante la navegación (Muñoz, Valencia, & Londoño, 2010).
4. **Energía de Curvatura (Bending Energy  $B_E$ ):** esta métrica evalúa la suavidad de la curvatura del robot. Esto se calcula como el cuadrado de la curvatura  $k$  sobre la trayectoria (Muñoz, Valencia, & Londoño, 2010). En la Ecuación 4-8 se presenta la fórmula de  $k$  y en la Ecuación 4-9 la fórmula de  $B_E$ .

$$k(x_i) = \frac{f''(x_i)}{(1 + (f'(x_i))^2)^{\frac{3}{2}}}$$

Ecuación 4-8

$$B_E = \frac{1}{n} \sum_{i=1}^n k^2(x_i, f(x_i))$$

Ecuación 4-9

5. **T<sub>B<sub>E</sub></sub>**: esta métrica considera tanto la suavidad como la longitud de la trayectoria (Muñoz, Valencia, & Londoño, 2010). En la Ecuación 4-10 se presenta la fórmula para calcular la métrica T<sub>B<sub>E</sub></sub>.

$$TB_E = \sum_{i=1}^n k^2(x_i, f(x_i))$$

Ecuación 4-10

Adicionalmente a estas métricas, se comparan las distancias totales recorridas, el tiempo que duró la simulación y las velocidades máximas y medias alcanzadas por el Pioneer 3-DX

### 4.3 Implementación y experimentación.

La implementación de la solución se realizó en el lenguaje de programación Java. Principalmente por pericia profesional.

Con este lenguaje se implantó una aplicación, la cual controla la navegación del robot y al mismo tiempo almacena la información que será usada para el cálculo de las métricas.

Esta aplicación fue diseñada con orientación a objetos, permitiendo así, separar los controladores difusos y neuronal para la realización de las pruebas.

Las pruebas realizadas sobre el robot Pioneer 3-DX, en la plataforma de simulación MobileSim fueron: navegación reactiva pura, navegación reactiva con red neuronal, navegación difusa y navegación neurodifusa. Todas estas con una ventana de tiempo de 5 minutos o hasta la primera colisión.

## 5 ANALISIS DE RESULTADOS

Para el análisis de resultados y comprobación de la veracidad de la hipótesis planteada se comparó el desempeño de cuatro arquitecturas distintas. Estas son: navegación reactiva pura, navegación reactiva neuronal, navegación difusa y navegación neuro – difusa, siendo esta última el postulado de la tesis.

Estas simulación fueron ejecutadas en la plataforma MobileSim, con el robot Pioneer 3-DX, durante un tiempo máximo de 5 minutos o bien hasta que colisionará.

Las simulaciones fueron ejecutadas en un entorno simular a un departamento, el cual fue modificado con el fin de presentar configuraciones complejas. Esto permitió probar la robustez de las diferentes arquitecturas de navegación. En la Figura 5-1 se puede ver el entorno empleado.

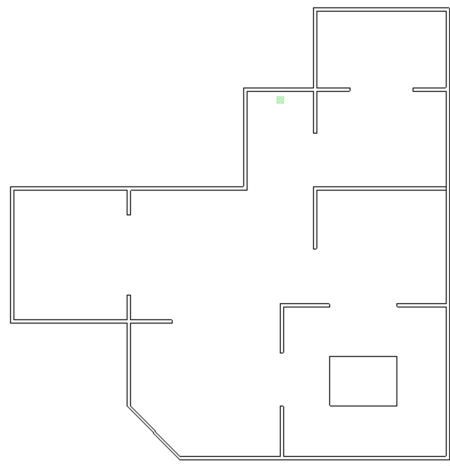


Figura 5-1 Entorno de simulación.

**Con formato:** Revisar la ortografía y la gramática

## 5.1 Simulaciones ejecutadas

Las pruebas realizadas a las distintas arquitecturas presentaron distintos patrones de comportamiento dentro del entorno. A continuación se presentara, a modo de ejemplo una captura de las simulaciones realidad para cada arquitectura.

En la Figura 5-2 se aprecia la navegación de la arquitectura reactiva pura, la cual durante un largo periodo de tiempo, quedo atrapada dentro de una de las habitaciones trampa del entorno.

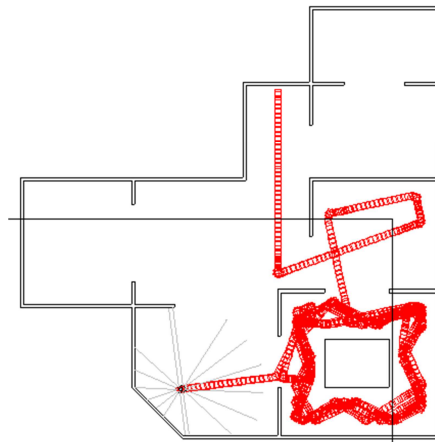


Figura 5-2 Arquitectura reactiva pura.

En la Figura 5-3 se aprecia la navegación de la arquitectura reactiva neuronal, la cual presento un mejor desempeño, sorteando en mejor tiempo la salida de una de las habitaciones trampa del entorno.

**Con formato:** Revisar la ortografía y la gramática

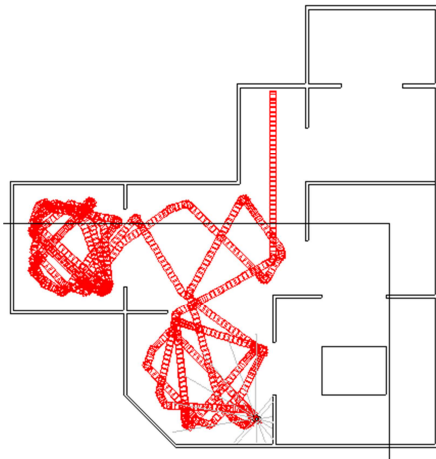


Figura 5-3 Arquitectura reactiva neuronal.

En la Figura 5-4 se aprecia la navegación de la arquitectura difusa, la cual colisiono a los 3 minutos y 50 segundos. Durante el tiempo de simulación esta arquitectura logro superar todas las habitaciones trampas en las que entro, a excepción de la última.

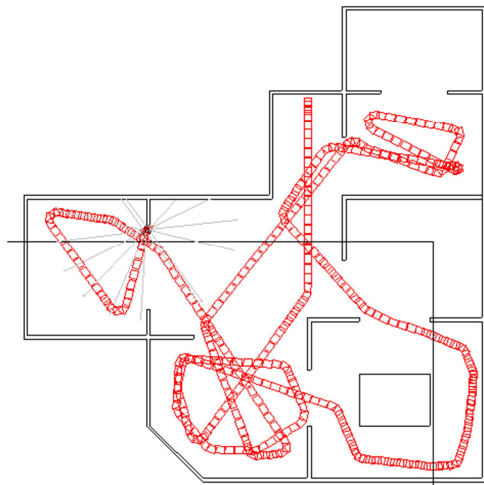


Figura 5-4 Arquitectura reactiva difusa.

**Con formato:** Revisar la ortografía y la gramática

**Con formato:** Revisar la ortografía y la gramática

En la Figura 5-5 se aprecia la navegación de la arquitectura neuro – difusa entro y salió de cuatro habitaciones trampas sin colisionar durante toda la simulación. Junto con esto, esta arquitectura navego la mayor parte del tiempo por zonas catalogadas como menos inseguras.

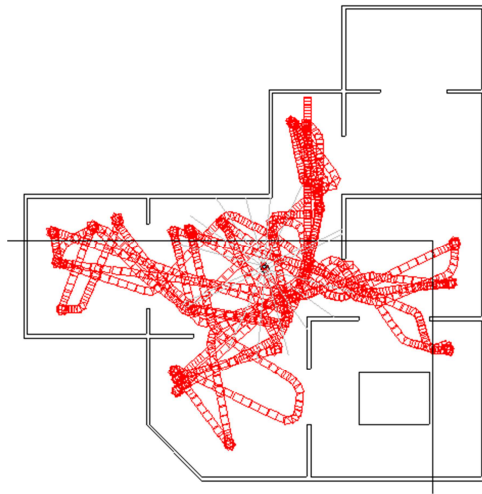


Figura 5-5 Arquitectura reactiva neuro – difusa.

## 5.2 Análisis cuantitativo de los resultados.

El análisis comparativo de las cuatro diferentes arquitecturas fue realizado utilizando las métricas descritas en punto 4.2.4. La Tabla 5-1 muestra los valores promedios de las métricas definidas para comparar el rendimiento y performance de los cuatro tipos de arquitecturas.

Con formato: Revisar la ortografía y la gramática

Con formato: Español (Chile)

**Con formato:** Revisar la ortografía y la gramática

Tabla 5-1 Resultados de las simulaciones.

	Tbe	Be	SM1	SM2	d	v Prom
NEURO DIFUSO	908,88	0,72	3.761,20	2.151,24	196,81	0,54
DIFUSO	12.976,58	4,49	3.777,29	1.973,65	404,02	0,79
NEURONAL	12.727,73	4,63	3.870,36	2.326,51	181,91	0,37
REACTIVO	7.279,70	2,50	3.278,47	1.488,42	180,68	0,34

**Con formato:** Español (Chile)

El modelo neuro-difuso, muestra curvas más suaves tanto en Tbe como en Be, lo cual implica un ahorro energético y tiempo. Junto con esto, recorre el circuito con la segunda velocidad más alta, solo superado por el controlador difuso, el cual resulto deficiente en las métricas de suavidad y seguridad. En cuando a las métricas de seguridad (SM1 y SM2) el controlador Neuro – Difuso obtuvo buenos resultados junto con el controlador Neuronal, lo cual se traduce en un comportamiento más seguro.

En el caso de la medición de la velocidad promedio, la arquitectura Neuro – difusa, tuvo una velocidad promedio de 0.54 m/s, lo cual es inferior a la alcanzada por la arquitectura difusa. Mientras que en la distancia recorrida, el controlador difuso recorrió una mayor distancian, siendo esta mayor al doble en comparación a las otras arquitecturas.

En resumen la arquitectura neuro – difusa presento un rendimiento superior en la mayoría de las métricas utilizadas. Destacando sobre todo el TB<sub>E</sub> – suavidad de la curva en toda la trayectoria – el cual fue el valor mínimo de todas las arquitecturas comparadas. Esta métrica está directamente correlacionada con una mayor eficiencia energética. Junto con esto, la distancia mínima promedio obtenida por esta arquitectura, fue superior en

comparación con otras, lo cual implica esta arquitectura presenta una efectiva evasión temprana obstáculos presentes y posibles emergentes.

En el punto 7.3 Anexo C, se encuentra el análisis estadístico completo realizado sobre las simulaciones.

## **6 CONCLUSIONES Y TRABAJO FUTURO**

### **6.1 Conclusiones finales.**

De los experimentos realizados, y tras analizar los resultados, se puede llegar a las siguientes conclusiones, el controlador neuro-difuso:

1. En comparación con los otros controladores, presenta una mayor estabilidad, al mismo tiempo una menor probabilidad de colisión.
2. Su trayectoria describe curvas de mayor suavidad, lo cual implica un mejor consumo energético.
3. Posee el comportamiento esperado en esta tesis, evade obstáculos y también evita puertas y pasillos cuando estos se presentan por alguno de sus flancos realizando acciones preventivas tempranas.
4. Es controlador es capaz de salir de zonas que presentan configuración de trampa, evitando permanecer en ellas en un bucle infinito.
5. Esta limitados por las características físicas de la distribución del anillo ultrasónico.

## 6.2 Trabajo Futuro.

Al concluir con este trabajo de investigación, es deducible que, el comportamiento de robot móvil puede ser mejorado considerablemente, si se reemplaza el anillo ultrasónico, por un sensor de barrido laser. De esta forma, se proveería al modelo de una malla de alta resolución de sensado y también mayor alcance. Por el cual el modelo sería más completo, en cuanto a conocimiento del entorno.

# 7 ANEXOS

## 7.1 Anexo A: Controlador Difuso.

### 7.1.1 Funciones de pertenencia.

En este apartado se presentan las funciones de pertenencia para las variables de entrada y de salida, tanto de forma gráfica como en tabla con sus respectivos parámetros. La Ecuación 7-1 muestra la variable de entrada mientras que la Figura 7-2 muestra la variable de salida, en la Tabla 7-1 se muestran los parámetros utilizados en las funciones de pertenencia tanto para las entradas y salidas.

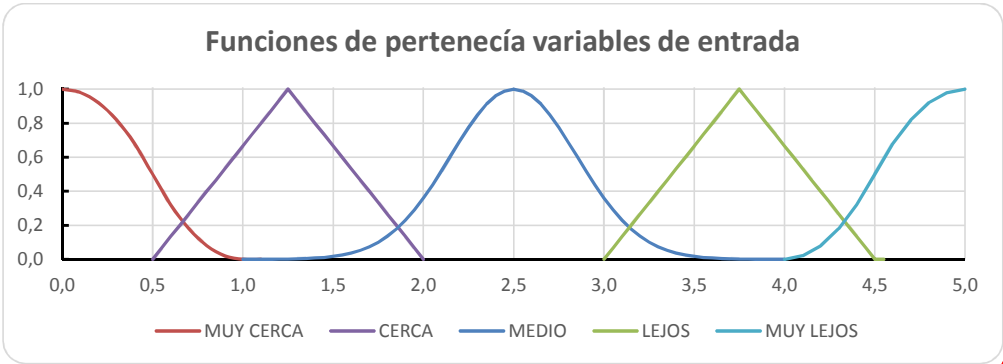


Figura 7-1 Funciones de pertenencia variables de entrada.

Con formato: Revisar la ortografía y la gramática

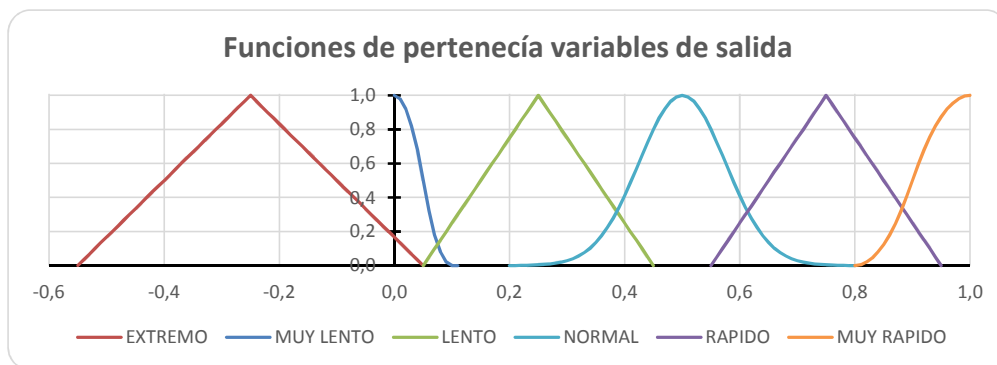


Figura 7-2 Funciones de pertenencia variables de salida.

Tabla 7-1 funciones y parámetros funciones de pertenencia

Entrada			Salida		
Variable	Función	Parámetros	Variable	Función	Parámetros
Muy cerca	Z	a: 0 b: 1	Extremo	Tri	a: -0.55 b: -0.25 c: 0.05
Cerca	Tri	a: 0.5 b: 1.25 c: 2	Muy lento	Z	a: 0 b: 0.1
Medio	Gauss	c: 2.5 $\beta$ : 0.35	Lento	Tri	a: 0.05 b: 0.25 c: 0.45
Lejos	Tri	a: 3 b: 3.75 c: 4.5	Normal	Gauss	c: 0.5 $\beta$ : 0.075
Muy lejos	S	a: 4 b: 5	Rápido	Tri	a: 0.55 b: 0.75 c: 0.95
			Muy rápido	S	a: 0.8 b: 1

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

Con formato: Revisar la ortografía y la gramática

### 7.1.2 Reglas difusas

En la Tabla 7-2 se presenta el listado completo de las reglas difusas del último modelo implementado en el Robot.

Tabla 7-2 Reglas difusas

	ENTRADA			SALIDA	
	FRENTE	IZQUIERDA	DERECHA	IZQUIERDA	DERECHA
1	MUY CERCA	MUY CERCA	MUY CERCA	EXTREMO	EXTREMO
2	MUY CERCA	MUY CERCA	CERCA	LENTO	EXTREMO
3	MUY CERCA	MUY CERCA	MEDIO	LENTO	EXTREMO
4	MUY CERCA	MUY CERCA	LEJOS	LENTO	EXTREMO
5	MUY CERCA	MUY CERCA	MUY LEJOS	LENTO	EXTREMO
6	MUY CERCA	CERCA	MUY CERCA	EXTREMO	LENTO
7	MUY CERCA	CERCA	CERCA	LENTO	EXTREMO
8	MUY CERCA	CERCA	MEDIO	LENTO	EXTREMO
9	MUY CERCA	CERCA	LEJOS	LENTO	EXTREMO
10	MUY CERCA	CERCA	MUY LEJOS	LENTO	EXTREMO
11	MUY CERCA	MEDIO	MUY CERCA	EXTREMO	LENTO
12	MUY CERCA	MEDIO	CERCA	EXTREMO	LENTO
13	MUY CERCA	MEDIO	MEDIO	LENTO	EXTREMO
14	MUY CERCA	MEDIO	LEJOS	LENTO	EXTREMO
15	MUY CERCA	MEDIO	MUY LEJOS	LENTO	EXTREMO
16	MUY CERCA	LEJOS	MUY CERCA	EXTREMO	LENTO
17	MUY CERCA	LEJOS	CERCA	EXTREMO	LENTO
18	MUY CERCA	LEJOS	MEDIO	EXTREMO	LENTO
19	MUY CERCA	LEJOS	LEJOS	LENTO	EXTREMO
20	MUY CERCA	LEJOS	MUY LEJOS	LENTO	EXTREMO
21	MUY CERCA	MUY LEJOS	MUY CERCA	EXTREMO	LENTO
22	MUY CERCA	MUY LEJOS	CERCA	EXTREMO	LENTO
23	MUY CERCA	MUY LEJOS	MEDIO	EXTREMO	LENTO
24	MUY CERCA	MUY LEJOS	LEJOS	EXTREMO	LENTO
25	MUY CERCA	MUY LEJOS	MUY LEJOS	LENTO	EXTREMO
26	CERCA	MUY CERCA	MUY CERCA	LENTO	MUY LENTO
27	CERCA	MUY CERCA	CERCA	LENTO	MUY LENTO
28	CERCA	MUY CERCA	MEDIO	LENTO	MUY LENTO
29	CERCA	MUY CERCA	LEJOS	LENTO	MUY LENTO
30	CERCA	MUY CERCA	MUY LEJOS	LENTO	MUY LENTO
31	CERCA	CERCA	MUY CERCA	MUY LENTO	LENTO

32	CERCA	CERCA	CERCA	LENTO	MUY LENTO
33	CERCA	CERCA	MEDIO	LENTO	MUY LENTO
34	CERCA	CERCA	LEJOS	LENTO	MUY LENTO
35	CERCA	CERCA	MUY LEJOS	LENTO	MUY LENTO
36	CERCA	MEDIO	MUY CERCA	MUY LENTO	LENTO
37	CERCA	MEDIO	CERCA	MUY LENTO	LENTO
38	CERCA	MEDIO	MEDIO	NORMAL	LENTO
39	CERCA	MEDIO	LEJOS	NORMAL	LENTO
40	CERCA	MEDIO	MUY LEJOS	NORMAL	LENTO
41	CERCA	LEJOS	MUY CERCA	MUY LENTO	LENTO
42	CERCA	LEJOS	CERCA	MUY LENTO	LENTO
43	CERCA	LEJOS	MEDIO	LENTO	NORMAL
44	CERCA	LEJOS	LEJOS	RAPIDO	NORMAL
45	CERCA	LEJOS	MUY LEJOS	RAPIDO	NORMAL
46	CERCA	MUY LEJOS	MUY CERCA	MUY LENTO	LENTO
47	CERCA	MUY LEJOS	CERCA	MUY LENTO	LENTO
48	CERCA	MUY LEJOS	MEDIO	LENTO	NORMAL
49	CERCA	MUY LEJOS	LEJOS	NORMAL	RAPIDO
50	CERCA	MUY LEJOS	MUY LEJOS	RAPIDO	NORMAL
51	MEDIO	MUY CERCA	MUY CERCA	MUY LENTO	MUY LENTO
52	MEDIO	MUY CERCA	CERCA	MUY LENTO	MUY LENTO
53	MEDIO	MUY CERCA	MEDIO	LENTO	LENTO
54	MEDIO	MUY CERCA	LEJOS	LENTO	LENTO
55	MEDIO	MUY CERCA	MUY LEJOS	LENTO	LENTO
56	MEDIO	CERCA	MUY CERCA	MUY LENTO	MUY LENTO
57	MEDIO	CERCA	CERCA	LENTO	LENTO
58	MEDIO	CERCA	MEDIO	LENTO	LENTO
59	MEDIO	CERCA	LEJOS	LENTO	LENTO
60	MEDIO	CERCA	MUY LEJOS	LENTO	LENTO
61	MEDIO	MEDIO	MUY CERCA	LENTO	LENTO
62	MEDIO	MEDIO	CERCA	LENTO	LENTO
63	MEDIO	MEDIO	MEDIO	NORMAL	NORMAL
64	MEDIO	MEDIO	LEJOS	NORMAL	NORMAL
65	MEDIO	MEDIO	MUY LEJOS	NORMAL	NORMAL
66	MEDIO	LEJOS	MUY CERCA	LENTO	LENTO
67	MEDIO	LEJOS	CERCA	LENTO	LENTO
68	MEDIO	LEJOS	MEDIO	NORMAL	NORMAL
69	MEDIO	LEJOS	LEJOS	RAPIDO	RAPIDO
70	MEDIO	LEJOS	MUY LEJOS	RAPIDO	RAPIDO
71	MEDIO	MUY LEJOS	MUY CERCA	LENTO	LENTO
72	MEDIO	MUY LEJOS	CERCA	LENTO	LENTO
73	MEDIO	MUY LEJOS	MEDIO	NORMAL	NORMAL
74	MEDIO	MUY LEJOS	LEJOS	RAPIDO	RAPIDO

75	MEDIO	MUY LEJOS	MUY LEJOS	RAPIDO	RAPIDO
76	LEJOS	MUY CERCA	MUY CERCA	MUY LENTO	MUY LENTO
77	LEJOS	MUY CERCA	CERCA	MUY LENTO	MUY LENTO
78	LEJOS	MUY CERCA	MEDIO	LENTO	LENTO
79	LEJOS	MUY CERCA	LEJOS	LENTO	LENTO
80	LEJOS	MUY CERCA	MUY LEJOS	LENTO	LENTO
81	LEJOS	CERCA	MUY CERCA	MUY LENTO	MUY LENTO
82	LEJOS	CERCA	CERCA	LENTO	LENTO
83	LEJOS	CERCA	MEDIO	NORMAL	NORMAL
84	LEJOS	CERCA	LEJOS	NORMAL	NORMAL
85	LEJOS	CERCA	MUY LEJOS	NORMAL	NORMAL
86	LEJOS	MEDIO	MUY CERCA	LENTO	LENTO
87	LEJOS	MEDIO	CERCA	NORMAL	NORMAL
88	LEJOS	MEDIO	MEDIO	RAPIDO	RAPIDO
89	LEJOS	MEDIO	LEJOS	RAPIDO	RAPIDO
90	LEJOS	MEDIO	MUY LEJOS	RAPIDO	RAPIDO
91	LEJOS	LEJOS	MUY CERCA	LENTO	LENTO
92	LEJOS	LEJOS	CERCA	NORMAL	NORMAL
93	LEJOS	LEJOS	MEDIO	RAPIDO	RAPIDO
94	LEJOS	LEJOS	LEJOS	MUY RAPIDO	MUY RAPIDO
95	LEJOS	LEJOS	MUY LEJOS	MUY RAPIDO	MUY RAPIDO
96	LEJOS	MUY LEJOS	MUY CERCA	LENTO	LENTO
97	LEJOS	MUY LEJOS	CERCA	NORMAL	NORMAL
98	LEJOS	MUY LEJOS	MEDIO	RAPIDO	RAPIDO
99	LEJOS	MUY LEJOS	LEJOS	MUY RAPIDO	MUY RAPIDO
100	LEJOS	MUY LEJOS	MUY LEJOS	MUY RAPIDO	MUY RAPIDO
101	MUY LEJOS	MUY CERCA	MUY CERCA	MUY LENTO	MUY LENTO
102	MUY LEJOS	MUY CERCA	CERCA	MUY LENTO	MUY LENTO
103	MUY LEJOS	MUY CERCA	MEDIO	LENTO	LENTO
104	MUY LEJOS	MUY CERCA	LEJOS	LENTO	LENTO
105	MUY LEJOS	MUY CERCA	MUY LEJOS	LENTO	LENTO
106	MUY LEJOS	CERCA	MUY CERCA	MUY LENTO	MUY LENTO
107	MUY LEJOS	CERCA	CERCA	LENTO	LENTO
108	MUY LEJOS	CERCA	MEDIO	NORMAL	NORMAL
109	MUY LEJOS	CERCA	LEJOS	NORMAL	NORMAL
110	MUY LEJOS	CERCA	MUY LEJOS	NORMAL	NORMAL
111	MUY LEJOS	MEDIO	MUY CERCA	LENTO	LENTO
112	MUY LEJOS	MEDIO	CERCA	NORMAL	NORMAL
113	MUY LEJOS	MEDIO	MEDIO	RAPIDO	RAPIDO
114	MUY LEJOS	MEDIO	LEJOS	RAPIDO	RAPIDO
115	MUY LEJOS	MEDIO	MUY LEJOS	RAPIDO	RAPIDO
116	MUY LEJOS	LEJOS	MUY CERCA	LENTO	LENTO

117	MUY LEJOS	LEJOS	CERCA	NORMAL	NORMAL
118	MUY LEJOS	LEJOS	MEDIO	RAPIDO	RAPIDO
119	MUY LEJOS	LEJOS	LEJOS	MUY RAPIDO	MUY RAPIDO
120	MUY LEJOS	LEJOS	MUY LEJOS	MUY RAPIDO	MUY RAPIDO
121	MUY LEJOS	MUY LEJOS	MUY CERCA	LENTO	LENTO
122	MUY LEJOS	MUY LEJOS	CERCA	NORMAL	NORMAL
123	MUY LEJOS	MUY LEJOS	MEDIO	RAPIDO	RAPIDO
124	MUY LEJOS	MUY LEJOS	LEJOS	MUY RAPIDO	MUY RAPIDO
125	MUY LEJOS	MUY LEJOS	MUY LEJOS	MUY RAPIDO	MUY RAPIDO

## 7.2 Anexo B: Red neuronal.

En este apartado se presenta el modelo de la red neuronal, específicamente los pesos sinápticos y también la función de activación utilizada para las neuronas.

La función de activación utilizada en el modelo neuronal fue la función Logística. En la Ecuación 7-1 se presenta su ecuación.

$$\frac{1}{1 + e^{-x}}$$

Ecuación 7-1

En la Tabla 7-3,

Tabla 7-4 y Tabla 7-5 se presentan los pesos sinápticos utilizados en la implementación de la red neuronal.

Tabla 7-3 Pesos sinápticos IN - HIDE 1

Z	TO	HIDE 1.1	HIDE 1.2	HIDE 1.3	HIDE 1.4	HIDE 1.5	HIDE 1.6	HIDE 1.7	HIDE 1.8	HIDE 1.9
	BIAS	-1,75	-2,92	0,57	1,62	-2,35	3,47	0,79	-3,30	-2,59
IN 1	0,19	-1,36	-1,71	0,81	7,86	-2,55	-1,06	44,49	-6,33	
IN 2	0,00	0,21	3,27	-26,97	-6,33	-0,10	-37,24	1,46	0,09	
IN 3	-8,17	2,55	-0,94	26,14	-0,94	0,08	37,46	-45,47	2,20	

Tabla 7-4 Pesos sinápticos HIDE 1 - HIDE 2

Con formato: Revisar la ortografía y la gramática

HIDE 1	TO	HIDE 2.1	HIDE 2.2	HIDE 2.3	HIDE 2.4	HIDE 2.5	HIDE 2.6
	BIAS	-7,32	6,98	-9,73	-5,57	4,96	-3,93
HIDE 1.1	-5,67	-0,41	8,99	14,49	-0,78	5,04	
HIDE 1.2	-0,45	1,49	-4,11	-14,14	-0,98	-18,23	
HIDE 1.3	2,98	-28,96	-7,57	3,78	-10,07	0,34	
HIDE 1.4	0,29	-12,11	5,25	22,10	-1,10	0,29	
HIDE 1.5	-8,34	69,15	-1,49	-4,77	4,64	0,23	
HIDE 1.6	4,97	-0,74	5,04	-12,75	2,72	1,48	
HIDE 1.7	-0,61	16,30	1,59	-6,30	2,39	-0,85	
HIDE 1.8	-2,22	45,64	-0,10	-0,29	6,94	0,77	
HIDE 1.9	4,10	-14,17	0,75	-8,88	-1,53	19,51	

Tabla 7-5 Pesos sinápticos HIDE 2 - OUT

HIDE 2	TO	OUT
BIAS		3,30
HIDE 2.1		-7,57
HIDE 2.2		-0,81
HIDE 2.3		-33,67
HIDE 2.4		1,76
HIDE 2.5		-2,66
HIDE 2.6		-21,05

### 7.3 Anexo C: Análisis estadístico.

Para la validación estadística de los resultados, primero se realizó un prueba de normalidad. Una vez que se pudo suponer que los datos distribuyen en forma normal se realizó un análisis de varianzas – ANOVA – de un solo factor a cada métrica.

Con formato: Español (Chile)

La prueba de ANOVA se realizó con 15 muestras para cada una de las 4 arquitecturas. Y se trabajó bajo un nivel de confianza de 95%.

Los resultados del análisis ANOVA son:

### 7.3.1 Análisis ANOVA para la métrica $TB_E$

#### RESUMEN

Grupos	Cuenta	Suma	Promedio	Varianza
NEURO DIFUSO	15	13.633,2	908,9	249.993
DIFUSO	15	194.648,8	12.976,6	47.899.573
NEURONAL	15	190.915,9	12.727,7	313.705.841
REACTIVO	15	109.195,4	7.279,7	85.294.503

#### ANÁLISIS DE VARIANZA

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Entre grupos	1,46E+09	3,00	485.124.280	4,34	0,01	2,77
Dentro de los grupos	6,26E+09	56,00	111.787.478			
Total	7,72E+09	59,00				

Con formato: Español (Chile)

El análisis ANOVA para la métrica  $TB_E$ , muestra que la hipótesis nula es rechazada para todo valor de alfa. Por lo cual las medias poblacionales de las distintas arquitecturas no son similares.

Para poder saber cuál de estas medias son similares se aplicó el método de Tukey, con un  $w$  calculado de 10.209,0. En consecuencia, la evidencia

estadística muestras que el valor medio del  $TB_E$  para un modelo Neuro – Difuso, será diferente a las medias de las otras arquitectura, como se puede ver en la Tabla 7-6.

Tabla 7-6 TBe método de Tukey.

NEURO DIFUSO	REACTIVO	NEURONAL	DIFUSO
908.9	7,279.7	12,727.7	12,976.6

Con formato: Español (Chile)

Con formato: Español (Chile)

Con formato: Revisar la ortografía y la gramática

Con formato: Español (Chile)

Con formato: Español (Chile)

Con formato: Español (Chile)

### 7.3.2 Análisis ANOVA para la métrica BE

#### RESUMEN

Grupos	Cuenta	Suma	Promedio	Varianza
NEURO DIFUSO	15	10,85	0,72	0,16
DIFUSO	15	67,42	4,49	5,41
NEURONAL	15	69,40	4,63	41,45
REACTIVO	15	37,50	2,50	9,52

#### ANÁLISIS DE VARIANZA

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Entre grupos	154,26	3	51,42	3,64	0,02	2,77
Dentro de los grupos	791,56	56	14,14			
Total	945,82	59				

Con formato: Español (Chile)

El análisis ANOVA para la métrica Be, muestras que la hipótesis nula es rechazada para todo valor de alfa. Por lo cual las medias poblacionales de las distintas arquitecturas no son similares.

Con formato: Español (Chile)

Para poder saber cuál de estas medias son similares se aplicó el método de Tukey, con un w calculado de 3,6. En consecuencia, la evidencia estadística muestras que el valor medio del Be para un modelo Neuro – difuso, será diferente a las medias de la arquitectura Difusa y Neuronal. Sin embargo, y debido a la gran variabilidad del modelo reactivo, no hay suficiente evidencia estadística para afirmar que las medias poblaciones de la arquitectura Neuro – Difusa y Reactiva sean diferentes, como se puede ver en la Tabla 7-7.

Con formato: Español (Chile)

Con formato: Español (Chile)

Tabla 7-7 Be método de Tukey.

NEURO DIFUSO	REACTIVO	DIFUSO	NEURONAL
0.7	2.5	4.5	4.6

Con formato: Español (Chile)

Con formato: Español (Chile)

Con formato: Español (Chile)

Con formato: Español (Chile)

### 7.3.3 Análisis ANOVA para la métrica velocidad Promedio.

RESUMEN				
Grupos	Cuenta	Suma	Promedio	Varianza
NEURO DIFUSO	15	8,07	0,54	0,0040
DIFUSO	15	11,91	0,79	0,0011
NEURONAL	15	5,51	0,37	0,0001
REACTIVO	15	5,07	0,34	0,0002

## ANÁLISIS DE VARIANZA

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Entre grupos	1,97	3	0,66	477,81	7,56E-40	2,77
Dentro de los grupos	0,08	56	0,00			
Total	2,05	59				

Con formato: Español (Chile)

El análisis ANOVA para la métrica Velocidad Promedio, muestra que la hipótesis nula es rechazada para todo valor de alfa. Por lo cual las medias poblacionales de las distintas arquitecturas no son similares.

Para poder saber cuál de estas medias son similares se aplicó el método de Tukey, con un  $w$  calculado 0,036. En consecuencia, la evidencia estadística muestra que el valor medio de la velocidad promedio es distinto entre todos los tipos de arquitectura, como se muestra en la Tabla 7-8.

Con formato: Español (Chile)

Con formato: Español (Chile)

Tabla 7-8 Velocidad promedio método de Tukey.

REACTIVO	NEURONAL	NEURO DIFUSO	DIFUSO
0.34	0.37	0.54	0.79

Con formato: Español (Chile)

### 7.3.4 Análisis ANOVA para la métrica SM1

#### RESUMEN

Grupos	Cuenta	Suma	Promedio	Varianza
NEURO DIFUSO	15	56.418	3.761	35.623,95
DIFUSO	15	56.659	3.777	12.099,01
NEURONAL	15	58.055	3.870	4.421,66
REACTIVO	15	49.177	3.278	44.680,93

ANÁLISIS DE VARIANZA

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Entre grupos	3.198.851	3	1.066.284	44,05	9,36E-15	2,77
Dentro de los grupos	1.355.558	56	24.206			
Total	4.554.409	59				

Con formato: Español (Chile)

El análisis ANOVA para la métrica SM1, muestran que la hipótesis nula es rechazada para todo valor de alfa. Por lo cual las medias poblacionales de las distintas arquitecturas no son similares.

Para poder saber cuál de estas medias son similares se aplicó el método de Tukey, con un w calculado 150,242. En consecuencia, la evidencia estadística muestra que el valor medio del TB<sub>E</sub> para un modelo reactivo, será diferente a las medias de las otras arquitecturas, como se puede ver en la Tabla 7-9.

Con formato: Español (Chile)

Con formato: Español (Chile)

Tabla 7-9 SM1 método de Tukey.

REACTIVO	NEURO DIFUSO	NEURONAL	DIFUSO
3.278,47	3.761,20	3.870,36	3.777,29

Con formato: Español (Chile)

### 7.3.5 Análisis ANOVA para la métrica SM2

Con formato: Español (Chile)

#### RESUMEN

Grupos	Cuenta	Suma	Promedio	Varianza
NEURO DIFUSO	15	32.268,64	2.151,24	33.475,26
DIFUSO	15	29.604,81	1.973,65	19.524,78
NEURONAL	15	34.897,60	2.326,51	3.832,82
REACTIVO	15	22.326,27	1.488,42	46.266,26

#### ANÁLISIS DE VARIANZA

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Entre grupos	5.864.792	3	1.954.930,8	75,8	1,03E-19	2,77
Dentro de los grupos	1.443.388	56	25.774,8			
Total	7.308.180	59				

Con formato: Español (Chile)

El análisis ANOVA para la métrica SM2, muestra que la hipótesis nula es rechazada para todo valor de alfa. Por lo cual las medias poblacionales de las distintas arquitecturas no son similares.

Para poder saber cuál de estas medias son similares se aplicó el método de Tukey, con un  $w$  calculado 155,033. En consecuencia, la evidencia estadística muestra que el valor medio de la SM1 es distinto entre todos los tipos de arquitectura, como se muestra en la Tabla 7-10.

Con formato: Español (Chile)

Con formato: Español (Chile)

Tabla 7-10 SM2 método de Tukey.

REACTIVO	DIFUSO	NEURO DIFUSO	NEURONAL
1,488.42	1,973.65	2,151.24	2,326.51

Con formato: Español (Chile)

## 8 TRABAJOS CITADOS

Con formato: Español (Chile)

Alhaj Ali, S., Ghaffari, M., Liao, X., & Hall, E. (Diciembre de 2006). Mobile Robotics, Moving Intelligence. (J. Buchli, Ed.) *i-techonline*, 5, 22.

Auat Cheeín, F. A., di Sciascio, F., & Carelli, R. (2008). Planificación de Caminos y Navegación de un Robot Móvil en Entornos Gaussianos mientras realiza tareas de SLAM.

Barragán Piña, A. (2009). *Síntesis de sistemas de control Borroso Estables por Diseño*. Huelva , España.

Baturone, I., & Gersnoviez, A. (2010). DISEÑO DE SISTEMAS NEURO-DIFUSOS PARA CONTROL DE ROBOTS MEDIANTE XFUZZY 3.

Bertona, L. (2005). *ENTRENAMIENTO DE REDES NEURONALES BASADO EN ALGORITMOS EVOLUTIVOS*. Buenos Aires, Argentina.

Cordoba G., M. (s.f.). *EL EFIGENIA EJ-1. Una Aeronave Robot de despegue y aterrizaje vertical (S/VTOL) autocontrolada mediante Inteligencia Artificial*.

Flórez, R., & Fernández, J. (2008). *Las Redes Neuronales Artificiales*. NETBIBLO.

Gersnoviez, A., & Baturone, I. (2008). CONTROLADOR PARA NAVEGACIÓN REACTIVA EVITANDO OBSTÁCULOS MEDIANTE TÉCNICAS NEURO-FUZZY.

Gómez Skarmeta, A., Martínez Barberá, H., & García López, P. (2001). UNA ARQUITECTURA DE AGENTES DIFUSOS PARA ROBOTS AUTÓNOMOS MÓVILES.

Gonzales Acevedo, H., & Mella Castañeda, C. (2007). *ESTUDIO COMPARATIVO DE TRES TÉCNICAS DE NAVEGACIÓN PARA ROBOTS MÓVILES*. Bucaramanga, Colombia.

Gutiérrez Zea, L., Jurko Vásquez, L., & Zuluaga, C. (2000). *DISEÑO Y CONSTRUCCIÓN DE UN VEHÍCULO SUBACUÁTICO CONTROLADO EN FORMA DUAL*. Medellín, Colombia.

J. Russell, S., & Norvig, P. (2004). *INTELIGENCIA ARTIFICIAL UN ENFOQUE MODERNO* (Segunda edición ed.). (D. Fayerman Aragón, Ed.) Madrid, España: PEARSON EDUCACION.

Medina Varela, P., Restrepo Correa, H., & Cruz Trejos, E. (2009). *OPERACIÓN DE SISTEMAS AGV (VEHÍCULOS GUIADOS AUTOMATIZADOS) CON FLUJO ESTÁTICO*. Pereira, Colombia.

Mobile Robots. (2012). *mobilerobots*. Obtenido de mobilerobots: <http://www.mobilerobots.com/Software/MobileSim.aspx>

Mora, M., & Tornero, J. (2007). Planificación de movimientos mediante la propagación de campos vectoriales artificiales. *8° Congreso Iberoamericano de Ingeniería Mecánica*.

Mosso Solano, J., & Vivas Alban, O. (2000). *SISTEMA SOFTWARE NEURODIFUSO PARA EL CONTROL DE UN ROBOT AUTONOMO*. Colombia.

Muñoz, N. D., Valencia, J. A., & Londoño, N. (Marzo de 2010). Quantitative Performance Metrics for Mobile Robots Navigation. (A. Barrera, Ed.) *Mobile Robots Navigation*, 17. doi:10.5772/8988

Ogata, K. (1998). *Ingeniería de control moderna* (Tercera Edición ed.). Prentice Hall.

Pedroza Reyes, A., Sanches Lopez, C., & Rodriguez Marmolejo, H. (2007). *Control de un Vehiculo Guiado Automaticamente (AGV)*. Aguascalientes, Mexico.

Pérez de Diego, D. (s.f.). 4º Ingeniería de Telecomunicación.

Pino, R., Gómez, A., & de Abajo, N. (2001). *Introducción a la inteligencia artificial : sistemas expertos, redes neuronales artificiales y computación evolutiva*. Oviedo.

Rios G., L., Bueno L., M., & Sanchez A., S. (2008). *GENERACIÓN DE TRAYECTORIAS PARA UN ROBOT MÓVIL EMPLEANDO REDES NEURONALES*. Pereira, Colombia.

Rodríguez, H. (1998). Planificación de Movimiento Mediante Campo de Potencial con Restricciones Dinámicas para Robots Móviles. *Revista de I + D Tecnológico*, 43-48.

- Ruiz de Garibay Pascual, J. (2007). Robótica: Estado del arte.
- Ruiz Del Solar, J., & Salazar, R. (s.f.). Introducción a la robótica.
- Tsankova, D. (Marzo de 2010). Neural Networks Based Navigation and Control of a Mobile Robot in a Partially Known Environment. *Mobile Robots Navigation, 10(10)*. Sofia, Bulgaria: InTechWeb.org.
- Uhrhan, C., Jaramillo, A., Plata, S., Buss, A., & Mojica, F. (2000). ESCARABOT: Vehículo Autoguiado para Transporte de Carga en Entornos Industriales.
- Vallejo Rodríguez, E. (2004). *APRENDIZAJE EVOLUTIVO DE REGLAS FUZZY EN UN SISTEMA CLASIFICADOR MODIFICADO PARA CONTROL DE AGENTES MÓVILES*. Valencia, España.
- Whitbrook, A. (2010). *Programming Mobile Robots with Aria and Palyer*. Springer.
- Zamora, M., Tomás-Balibrea, L., Martínez, H., & Skarmeta, A. (s.f.). *Navegación Planificada de un Robot Móvil en Entornos Interiores Desconocidos*. Murcia, España.