



# **ESCUELA POLITÉCNICA NACIONAL**



## **FACULTAD DE INGENIERIA MECÁNICA**

### **TEMA:**

**“DISEÑO Y SIMULACIÓN DE UN ROBOT MÓVIL CON NAVEGACIÓN  
AUTÓNOMA PARA EL TRASLADO DE COCHES ENTRE DOS  
ESTACIONES DE TRABAJO EN LA EMPRESA AYMESA”**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
MAGISTER EN DISEÑO PRODUCCIÓN Y AUTOMATIZACIÓN INDUSTRIAL**

### **AUTOR:**

**Ing. Ricardo Maroto  
washington.maroto@epn.edu.ec**

### **DIRECTOR:**

**Ph.D. Álvaro Aguinaga  
alvaro.aguinaga@epn.edu.ec**

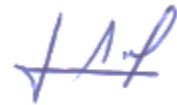
### **CODIRECTOR:**

**Ph.D. Édgar Cando  
edgar.cando@epn.edu.ec**

**Quito, 04 de enero de 2021**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por el ING. WASHINGTON RICARDO MAROTO SÁNCHEZ, bajo nuestra supervisión.



---

**Ph.D. Álvaro Aguinaga**  
**DIRECTOR DE PROYECTO**



---

**Ph.D. Édgar Cando**  
**CO-DIRECTOR DE PROYECTO**

## DECLARACIÓN

Yo, Washington Ricardo Maroto Sánchez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.



---

Ing. Washington Ricardo Maroto Sánchez

## DEDICATORIA

A Dios,

A mis padres Carlos y Carmen por ser el pilar de mi vida,

A mi esposa Elizabeth, a mis hijos Adrián y Paola, la familia que Dios me bendijo.

## **AGRADECIMIENTO**

A mi esposa Elizabeth por su apoyo incondicional, a mis hijos Adrián y Paola, que con sus sonrisas alegran mi corazón, los amo.

A mis padres, a mis hermanos, a mis ñaños que siempre me han apoyado para cumplir todas mis metas.

Al Ph.D. Álvaro Aguinaga, y Ph.D. Édgar Cando, por su dirección en el desarrollo del presente proyecto.

Al personal académico, administrativo y compañeros, que nos hemos formado en la Escuela Politécnica Nacional institución de prestigio referente de la educación universitaria a nivel nacional e internacional.

# ÍNDICE

INTRODUCCIÓN .....	1
Objetivo General.....	2
Objetivos Específicos .....	2
<b>1. MARCO TEÓRICO .....</b>	<b>3</b>
<b>1.1 Introducción a la robótica.....</b>	<b>3</b>
<b>1.1.1 Generalidades de la robótica.....</b>	<b>3</b>
<b>1.1.2 Clasificación de robots .....</b>	<b>5</b>
<b>1.1.3 Robots móviles .....</b>	<b>8</b>
<b>1.1.4 Percepción del entorno.....</b>	<b>11</b>
<b>1.2 Control Fuzzy .....</b>	<b>11</b>
<b>1.2.1 Lógica difusa .....</b>	<b>12</b>
<b>1.2.2 Funciones de membresía .....</b>	<b>13</b>
<b>1.2.3 Sistema de inferencia difusa .....</b>	<b>14</b>
<b>1.2.4 Defuzzificación .....</b>	<b>15</b>
<b>1.2.5 Estado del arte.....</b>	<b>16</b>
<b>2. METODOLOGÍA.....</b>	<b>18</b>
<b>2.1 Introducción .....</b>	<b>18</b>
<b>2.2 Diseño y modelación del control inteligente .....</b>	<b>18</b>
<b>2.2.1 Diseño sistema mecánico .....</b>	<b>18</b>
<b>2.2.2 Diseño del sistema electrónico.....</b>	<b>26</b>
<b>2.2.3 Diseño del sistema de percepción.....</b>	<b>30</b>
<b>2.2.4 Modelación matemática del control Fuzzy .....</b>	<b>34</b>
<b>2.3 Simulación computacional.....</b>	<b>36</b>
<b>2.3.1 Programación en Matlab del control inteligente .....</b>	<b>36</b>
<b>2.3.2 Desarrollo de la interfaz hombre máquina para control Fuzzy .....</b>	<b>52</b>
<b>2.3.3 Simulación computacional de la respuesta temporal del control difuso .....</b>	<b>53</b>
<b>2.4 Validación virtual .....</b>	<b>62</b>
<b>2.4.1 Programación del GUI de Matlab.....</b>	<b>62</b>
<b>2.4.2 Desarrollo de la interfaz hombre máquina para el robot móvil.....</b>	<b>64</b>
<b>2.4.3 Validación virtual, del comportamiento del robot móvil para el traslado de coches.....</b>	<b>66</b>

<b>3. RESULTADOS Y DISCUSIÓN .....</b>	<b>73</b>
<b>3.1 Pruebas y resultados .....</b>	<b>73</b>
<b>3.2 Análisis y discusión de resultados .....</b>	<b>83</b>
<b>4. CONCLUSIONES.....</b>	<b>92</b>
<b>4.1 Conclusiones .....</b>	<b>92</b>
<b>4.2 Recomendaciones .....</b>	<b>93</b>
<b>BIBLIOGRAFÍA .....</b>	<b>94</b>

## Índice de Figuras

Figura 1.1 Robot Unimate .....	5
Figura 1.2 Robot industrial o manipulador .....	6
Figura 1.3 Robot ASIMO .....	6
Figura 1.4 Dron .....	7
Figura 1.5 Robot Marino .....	8
Figura 1.6 Robot AIBO .....	8
Figura 1.7 Clasificación de robots móviles .....	9
Figura 1.8 Configuraciones cinemáticas RMR .....	10
Figura 1.9 Funciones de membresía .....	14
Figura 2.1 Fases del diseño .....	18
Figura 2.2 Robot KIVA      Figura 2.3 Robot MIR 200 .....	19
Figura 2.4 Sistema electrónico .....	26
Figura 2.5 Arduino Mega 2560 .....	27
Figura 2.6 Motor a pasos      Figura 2.7 Motor Nema 34 .....	28
Figura 2.8 Driver .....	29
Figura 2.9 XBee .....	30
Figura 2.10 Esquema de los sensores ultrasónicos .....	31
Figura 2.11 Sensor óptico .....	31
Figura 2.12 Cámara CCD .....	32
Figura 2.13 Tracción diferencial .....	34
Figura 2.14 Configuración de navegación autónoma .....	35
Figura 2.15 Pantalla del editor del sistema de inferencia difuso FIS .....	37
Figura 2.16 Add variable (Input, Output) .....	37
Figura 2.17 Membership Function (a) Edit y (b) Custom Membership Function .....	38
Figura 2.18 Rules Editor .....	39
Figura 2.19 Variables de entrada Control 1 (a) $\theta_1$ ; (b) $\theta_2$ y (c) EDO .....	44
Figura 2.20 Variables de salida Control 1 (a) RI y (b) RD .....	45
Figura 2.21 Variables de entrada Control 2 (a) EDO y (b) EDPD .....	46
Figura 2.22 Variables de salida Control 2 (a) RI y (b) RD .....	47
Figura 2.23 Inferencia Mamdani (a) Control 1 y (b) Control 2 .....	48
Figura 2.24 Regla de membresía 1 del Control 1 .....	49



Figura 2.25 Regla de membresía 47 del Control 1 .....	49
Figura 2.26 Regla de membresía 123 del Control 1 .....	50
Figura 2.27 Regla de membresía 7 del Control 2.....	51
Figura 2.28 Regla de membresía 15 del Control 2 .....	51
Figura 2.29 Pantalla de diseño GUI.....	52
Figura 2.30 Pantalla de diseño GUI.....	53
Figura 2.31 View Control 1 .....	54
Figura 2.32 View (a) Rules Viewer y (b) Surface Viewer .....	55
Figura 2.33 Rule Viewer Control 1 .....	56
Figura 2.34 Rule Viewer Control 2 .....	57
Figura 2.35 Superficie de control 1 RI (a) $\theta_1$ y $\theta_2$ (b) $\theta_1$ y EDO (c) $\theta_2$ y EDO .....	59
Figura 2.36 Superficie de control 1 RI (a) $\theta_1$ y $\theta_2$ (b) $\theta_1$ y EDO (c) $\theta_2$ y EDO .....	60
Figura 2.37 Superficie de control 2 (a) RI y (b) RD .....	61
Figura 2.38 GUI (a) Control 1 y (b) Control 2 .....	64
Figura 2.39 GUI (a) Control 1 y (b) Control 2 .....	65
Figura 2.40 Error fuera de rango.....	65
Figura 2.41 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha .....	68
Figura 2.42 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha .....	70
Figura 2.43 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha .....	72
Figura 3.1 Rules Viewer Control 1 .....	73
Figura 3.2 Guide Control 1 .....	74
Figura 3.3 Rules Viewer Control 2 .....	75
Figura 3.4 Guide Control 2 .....	75
Figura 3.5 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha.....	78
Figura 3.6 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha.....	80
Figura 3.7 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha.....	83
Figura 3.8 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha.....	86
Figura 3.9 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha.....	87
Figura 3.10 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha .....	88
Figura 3.11 Velocidades teta 1 .....	89
Figura 3.12 Velocidades teta 2 .....	90
Figura 3.13 Velocidades EDPD.....	91

## Índice de tablas

Tabla 1.1 Variables y valores lingüísticos.....	13
Tabla 2.1 Especificaciones.....	20
Tabla 2.2 Coeficientes de fricción .....	21
Tabla 2.3 Especificaciones Acero ASTM A36.....	25
Tabla 2.4 Variables de entrada y salida.....	40
Tabla 2.5 Valores lingüísticos .....	40
Tabla 2.6 Velocidades de ruedas.....	67
Tabla 2.7 Velocidades de ruedas.....	69
Tabla 2.8 Velocidades de ruedas.....	71
Tabla 3.1 Evaluación de datos de entradas Control 1 .....	84
Tabla 3.2 Evaluación de datos de entradas Control 2.....	84

## RESUMEN

La empresa Aymesa S.A. referente de la industria automotriz en el Ecuador desde hace 5 décadas, como parte de la mejora continua de sus procesos e innovación en el sistema de manejo de material que permita el incremento de la productividad, la reducción de tiempos de fabricación y costos, requiere el diseño y simulación de un robot móvil para el traslado de coches entre dos estaciones de trabajo con navegación autónoma que recorre desde una posición inicial hasta una posición deseada.

Para el estudio de este robot móvil con navegación autónoma, se establece el diseño de un robot con tracción diferencial y modelación de un control inteligente mediante el desarrollo de dos controles Fuzzy: ángulo de dirección del robot y la velocidad de avance. El software Matlab permite la simulación, programación y validación; a través de una interfaz gráfica en el ingreso de valores de entrada se obtiene valores de salida para cada control, permitiendo la visualización y el análisis del comportamiento de las velocidades de cada una de las ruedas.

**Palabras clave:** Robot móvil, Matlab, Control Fuzzy, Simulación

## **ABSTRACT**

The company Aymesa S.A. reference of the automotive industry in Ecuador for 5 decades, as part of the continuous improvement of its processes and innovation in the material handling system that allows the increase in productivity, the reduction of manufacturing times and costs, requires the design and simulation of a mobile robot for the transfer of cars between two work stations with autonomous navigation that travels from an initial position to a desired position.

For the study of this mobile robot with autonomous navigation, the design of a robot with differential traction and the modeling of an intelligent control is established through the development of two Fuzzy controls: robot direction angle and advance speed. The Matlab software allows simulation, programming and validation; through a graphical interface in the input of input values, output values are obtained for each control, allowing the visualization and analysis of the behavior of the speeds of each of the wheels.

**Keywords:** Mobile Robot, Matlab, Fuzzy Control, Simulation

## **INTRODUCCIÓN**

En la industria automotriz ecuatoriana, la empresa Aymesa S.A. es reconocida como la planta pionera en el ensamblaje de vehículos desde el año de 1972 (Aymesa, 2021), con el modelo Andino, la mejora continua de sus procesos, ha permitido a la empresa exportar y mantenerse en el mercado ecuatoriano, en la actualidad se dedica al ensamblaje de vehículos de las marcas Kía y Hyundai.

La planeación de un sistema de manejo de material se inicia con el análisis de los materiales a ser movidos, se selecciona de acuerdo a un listado de partes y se almacena en estanterías fijas o estanterías móviles a las que llamamos coches, los mismos que son trasladados por un operador mediante un transportador de remolque. En el proceso se debe garantizar que los materiales que primero ingresan deben ser los primeros en ser despachados.

Como política de mejora continua, la empresa ha estado innovando los procesos con calidad, eficiencia a fin de aumentar la productividad, reducir tiempos de fabricación y costos, se propone diseñar y simular un robot móvil para el traslado de coches entre dos estaciones de trabajo.

## **Objetivo General**

Diseñar y simular un robot móvil con navegación autónoma para el traslado de coches entre dos estaciones de trabajo en la empresa Aymesa.

## **Objetivos Específicos**

1. Diseñar y modelar el sistema mecatrónico y el control inteligente del robot móvil.
2. Modelar matemáticamente el control inteligente Fuzzy para el robot móvil.
3. Desarrollar la interfaz hombre máquina para control Fuzzy.
4. Simular el comportamiento del robot móvil en el traslado de coches.
5. Validar el modelo de control inteligente con el paquete GUI de Matlab.

# 1. MARCO TEÓRICO

## 1.1 Introducción a la robótica

En el proceso de desarrollo de la humanidad, la creatividad ha sido fundamental para generar nuevas ideas que faciliten las tareas de los individuos, el uso adecuado de herramientas, la mecanización, han permitido que actividades rutinarias o repetitivas se realicen en el menor tiempo y costo. La robótica está inmersa en todos los ámbitos por lo que ha generado el debate de cómo los robots influyen en la sociedad (R. Suárez, 2017), con la percepción de que el desarrollo en la industria o los servicios pueda crecer el desempleo lo que genera rechazo en la implementación, pero se debe considerar que el cambio permitiría otro tipo de empleo con la especialización de la mano de obra.

### 1.1.1 Generalidades de la robótica

#### Definición de Robótica

El escritor ruso Isaac Asimov en su relato de Runaround, definió a la Robótica como la ciencia que se encarga de estudiar y diseñar a los robots, que ha pasado desde la perspectiva de la ficción a la realidad en diferentes aplicaciones para el desarrollo de la sociedad, minimizando costos y tiempo.

Según (Vejarano Anzola, 2014), la Robótica es una ciencia que se dedica desde el diseño hasta la ejecución de proyectos, con el objeto de realizar tareas repetitivas y automatizadas, en la que se involucran algunas disciplinas tales como: mecánica, física, electrónica, control, inteligencia y visión artificial, etc.

#### Leyes de la Robótica

En el año de 1945 Isaac Asimov el impulsor de la palabra robot, enunció las tres leyes de la robótica (Cusco & Quishpe, 2018), que permiten ser base para estudios posteriores:

1. Un robot no puede perjudicar a un ser humano, ni con inacción permitir que un ser humano sufra daño.

2. Un robot debe obedecer las órdenes de un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia, mientras tal protección no entre en conflicto con la primera o segunda de estas leyes.

## **Definición de Robot**

Es un autómatas que está formado por la parte mecánica y un sistema electromecánico. La palabra robot procede de la palabra “robota” que en checo significa trabajo forzado, propuesto por Karel Capek en 1921 como parte de su novela R.U.R. (Rossum’s Universal Robot), le describe como una máquina que sustituye al humano para ejecutar tareas sin descanso.

Otra definición según (Reyes, 2011), hace referencia a la proporcionada por Robot Institute of América (RIA) de que “Un robot es un manipulador multifuncional reprogramable diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos programados para la ejecución de una variedad de tareas”.

El creador del primer robot industrial fue el norteamericano George Devol, que luego junto a Joseph Engelberger crearon en Connecticut la empresa Unimation, encargada de la fabricación de robots. En 1961 George Devol patentaba el primer robot industrial programable de la historia, conocido como Unimate representado en la Figura 1.1, en la empresa General Motors, era un robot que permitía realizar múltiples tareas y acceder a un gran espacio de trabajo con el espacio ocupado por el robot (Murray, 1994), estableciendo las bases de la robótica industrial moderna.

Los robots están diseñados para el trabajo difícil, repetitivo como: ensamblar componentes, tarjetas electrónicas, automóviles, manufactura, desarme de minas, atención al cliente, etc., actualmente están asumiendo los puestos de trabajo que requieren mucha precisión, por lo que no se cansan y sólo hacen las cosas que se les programa. Se tiene que reconocer que ningún equipo es totalmente estable o libre de errores, por lo que no podemos depender sólo del trabajo que realicen las máquinas, sino de adquirir el conocimiento adecuado para programarlos de acuerdo con nuestras necesidades.



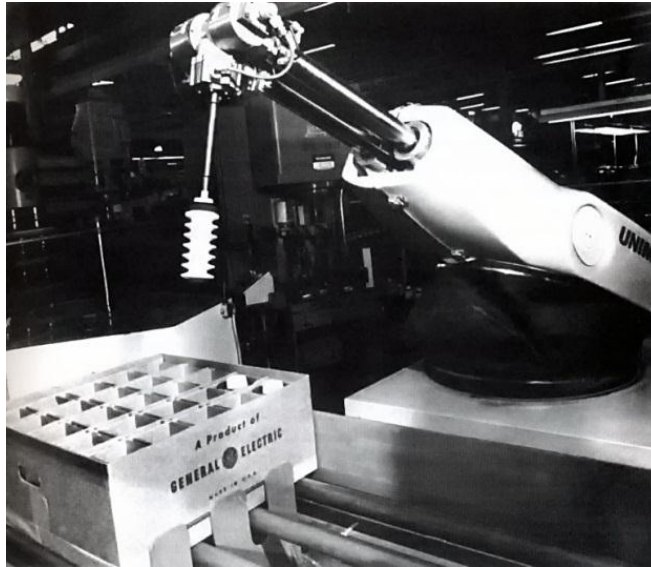


Figura 1.1 Robot Unimate

Fuente: (Trilnick, 1961)

El costo y accesibilidad en el mercado para adquirir los robots tiende a disminuir, por lo que es una oportunidad en las economías pequeñas de estar en igual de condiciones con las multinacionales, además son flexibles lo que permite reubicarse en menor tiempo.

### **1.1.2 Clasificación de robots**

El avance en el desarrollo de los robots ha logrado facilitar el avance significativo de la tecnología, con estructuras geométricas, mecánicas que definen su funcionalidad y aplicación (Reyes, 2011). Pueden clasificarse según lo siguiente:

- Robot industrial o manipulador
- Robot Humanoide
- Robots Móviles

#### **Robot industrial o manipulador**

Este tipo de robot es el más conocido ya que está presente en todas las actividades del sector industrial, con aplicaciones en la automatización o manipulación en los procesos, aumentando productividad, competitividad, eficiencia y rentabilidad para las empresas (Reyes, 2011). La Figura 1.2 muestra un manipulador industrial.



Figura 1.2 Robot industrial o manipulador

Fuente: (ABB, n.d.)

## Robots humanoides

También conocidos como androides, son máquinas antropomórficas capaces de imitar las funciones básicas del ser humano como: caminar, hablar, ver, recolectar, limpiar y trasladar objetos (Reyes, 2011), ejemplo de este tipo de robot es el ASIMO de Honda como se puede apreciar en la Figura 1.3.



Figura 1.3 Robot ASIMO

Fuente: (Honda - Asimo, n.d.)

## Robots móviles

Los robots móviles son dispositivos de transporte automático, sistema de locomoción, con cierta autonomía en sus traslados, que pueden ser terrestres, marinos o aéreos, las aplicaciones siempre se relacionan con tareas de riesgo o que sean nocivos para la salud.

- Aéreos
- Marinos
- Terrestres

**Aéreos.-** son vehículos aéreos no tripulados operadas a control remoto o pueden ser operados de forma autónoma utilizando planes de vuelo programados (Reyes M. et al., 2018) que pueden proporcionar imágenes aéreas de terreno y superficie, son útiles para la inspección en el ámbito militar o en el plano civil con aplicaciones en el ambiente, la agricultura (Barrientos et al., 2009), este tipo de dispositivos han evolucionado a lo que hoy conocemos como drones, ver Figura 1.4.



Figura 1.4 Dron

Fuente: (Unten, n.d.)

**Marinos.-** son utilizados en exploración o inspección para el control de aguas, control de salinidad, en el fondo marino aprovechando la autonomía del mismo, lo que ha permitido su uso en aguas profundas y no depender de los submarinos que son vehículos controlados por humanos. Éstos robots están equipados con sensores para navegación dentro del agua como: radar, visión telescópica, giroscopio, los mismos que le permiten sumergirse y elevarse (Reyes, 2011), representada en la Figura 1.5.



Figura 1.5 Robot Marino

Fuente: (*Robot-corales*, 2011)

**Terrestres.-** se trasladan mediante ruedas, patas u orugas, se utilizan en tareas de rastreo y traslado de objetos, evasión de obstáculos, limpieza del hogar (Reyes, 2011), ejemplo de este tipo puede ser el robot AIBO de Sony tipo mascota como se muestra en la Figura 1.6.



Figura 1.6 Robot AIBO

Fuente: (*Sony AIBO ERS-210*, n.d.)

### 1.1.3 Robots móviles

Los robots móviles son máquinas que no permanecen fijas, con una gran capacidad de desplazamiento, trasladándose de un punto a otro, planificando la trayectoria, la posición y el control de la velocidad, pudiendo evadir obstáculos, pero no tiene la precisión que dispone un robot industrial.

Dependiendo del medio donde trabajen los robots móviles, la Figura 1.7 muestra la clasificación de acuerdo con el sistema de locomoción:

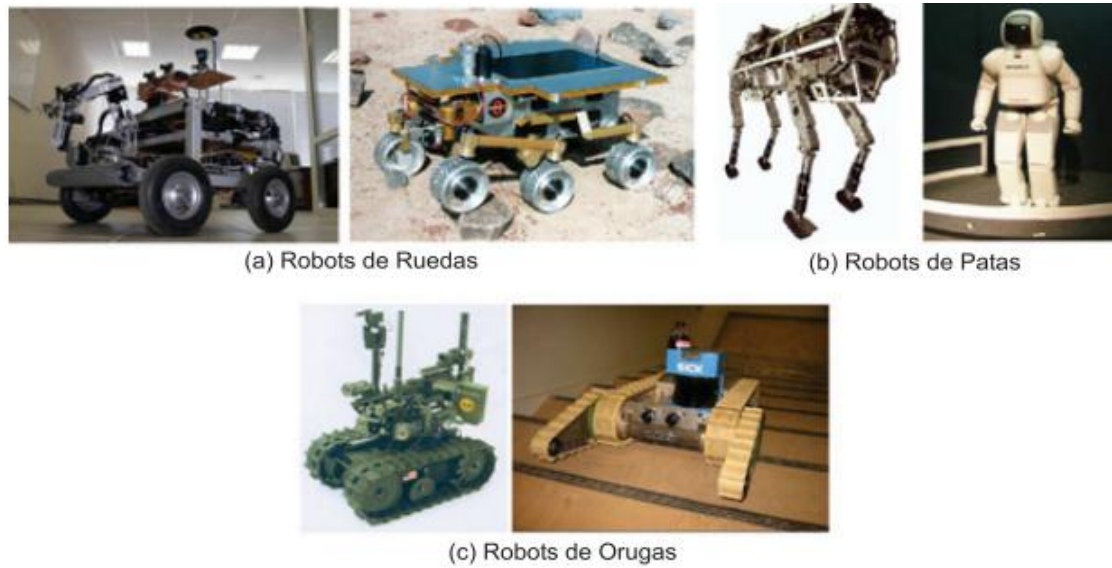


Figura 1.7 Clasificación de robots móviles

Fuente: (Víctor et al., n.d.)

## Orugas

Estos robots son los denominados todo terreno, están formadas por bandas laterales o cadenas ancladas en ruedas mejorando la tracción, realizan actividades de inspección o búsqueda en terrenos irregulares.

## Patas

Permite desplazamientos alcanzando como fin la estabilidad para no caerse, este tipo de locomoción toma partida en el movimiento para el traslado de los animales (Yandún, 2011). Presentándose como mejor opción respecto del uso de ruedas, pueden clasificarse de acuerdo con el número de patas de la siguiente manera:

- Robots de una sola pata
- Robots bípedos
- Robots cuadrúpedos
- Robots hexápodos

## Ruedas

Este tipo de robots fabricados con ruedas permite flexibilidad para transportar o trasladar cargas, que están mejor adaptados para circular en superficies planas, poco rugosas, pero la desventaja es la dificultad de traslado en terreno irregular de arena o grava.

### Tipos de ruedas

Los robots móviles según el tipo de movimiento que requieran realizar disponen de diferentes tipos de ruedas (Víctor et al., n.d.), que se indican a continuación:

- Rueda omnidireccional
- Rueda convencional
  1. Rueda fija
  2. Rueda orientable centrada
  3. Rueda orientable no centrada
- Rueda tipo castor
- Rueda de bolas

### Configuración cinemática de robots móviles con ruedas (RMR)

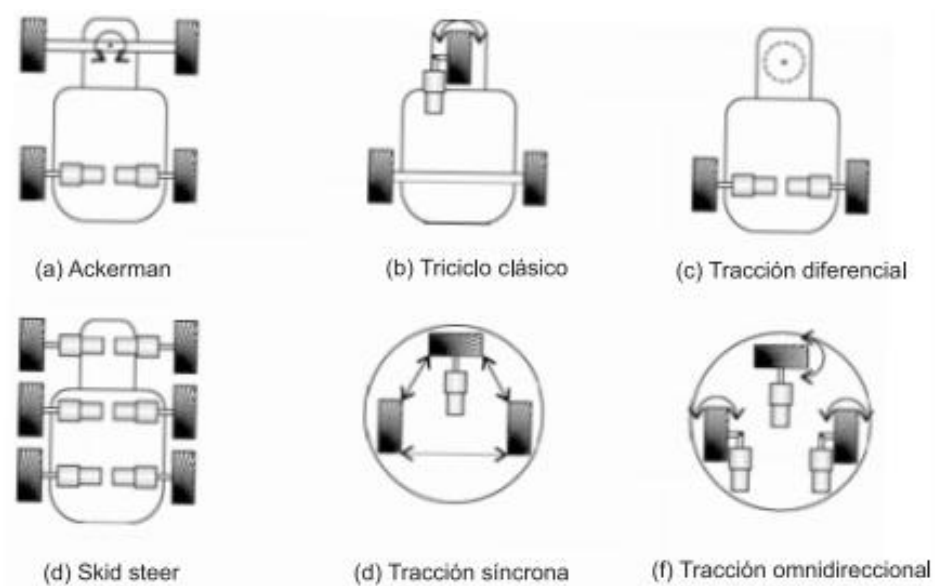


Figura 1.8 Configuraciones cinemáticas RMR

Fuente: (Calandín, 2006)

Existen diferentes configuraciones cinemáticas que dependen de la aplicación de las RMR, por lo que se tiene las siguientes configuraciones: Ackerman, triciclo clásico, tracción diferencial, skid steer, tracción síncrona y tracción omnidireccional (Calandín, 2006), ver Figura 1.8.

### **1.1.4 Percepción del entorno**

El robot determina su relación con el entorno mediante un sistema sensorial que interprete cualquier tipo de objeto u obstáculos y detecte posibles peligros para el funcionamiento del robot. La interacción de los sentidos a partir de la obtención de la información visual, táctil, etc., debe darse de manera rápida y precisa.

- Reconocimiento de imágenes
- Localización
- Mapas
- Mapeo y localización simultánea SLAM
- Planeación de trayectorias

Cada componente de la percepción del entorno en el desarrollo de un algoritmo difuso busca hacer el recorrido lo óptimo posible sin involucrar demasiados recursos en el proceso.

## **1.2 Control Fuzzy**

El control Fuzzy o control difuso es una herramienta de control de sistemas y procesos industriales que toma un valor de entrada, procesa por medio de algoritmos y genera una salida, basado en el comportamiento del ser humano o sentido común, en lugar de utilizar números o ecuaciones y que procesa reglas que genera uno o varios cambios en la salida.

En el control difuso el algoritmo de control utiliza la expresión si...entonces (if then), con algunas variables para obtener las soluciones pues se lo considera disperso, además se le considera lingüístico por el uso de variables lingüísticas debido a su facilidad de entendimiento con palabras como alto, bajo, rápido, lento, viejo, joven, etc. Las variables de los procesos se miden cuantitativamente.

## **Etapas del control Fuzzy**

El control difuso o control FUZZY, tiene las siguientes etapas para desarrollarse:

- Fuzzificación
- Funciones de membresía
- Sistema de inferencia difusa
- Defuzzificación

### **1.2.1 Lógica difusa**

La lógica difusa es un método de razonamiento aproximado, que permite representar matemáticamente la incertidumbre y facilita el modelado de la información. Se definen los conjuntos difusos que tienen elementos cuya pertenencia al conjunto no se puede establecer con certeza, pero tiene la capacidad de gestionar la imprecisión a diferencia de los conjuntos clásicos, permitiendo obtener un comportamiento coherente en función de las reglas que lo proveen.

La lógica difusa proporciona un medio para enfrentar situaciones del mundo real, situaciones complejas, más fáciles de caracterizar con palabras que por ecuaciones o modelos matemáticos. La lógica difusa permite el diseño de controladores que son robustos a la incertidumbre, los ruidos o perturbaciones (Parra et al., 2007).

En lógica difusa no hay una transición clara entre lo que es falso y lo que es verdadero. La lógica difusa permite tomar decisiones en función de las reglas poco precisas, cuya evaluación está sometida a cierta interpretación, fundamentado en conceptos percibidos de diferente manera por las personas desde el punto de vista de cada uno.

#### **Fuzzificación**

Se toman las estimaciones en tiempo real y se relegan valores difusos entre 0 y 1, se asocia un grado de pertenencia 0 (el valor lingüístico es totalmente falso), y 1 (el valor lingüístico es totalmente verdadero) es decir valores que tienen incertidumbre lingüística, este procedimiento se conoce como fuzzificación.



Se asocia a cada valor numérico un valor lingüístico y su grado de pertenencia transforma a un conjunto difuso. Los datos que ingresan al sistema difuso están expresados en lenguaje común, que se encarga de transformar la variable de entrada a conjuntos difusos con un grado de membresía (Suarez et al., 2018).

### Variable lingüística

Son variables cuyos valores son palabras o frases de un lenguaje natural, están conformados por los atributos o llamados también variables lingüísticos que puede adquirir una variable y cada atributo está representado por un conjunto difuso (Ruiz Alcántara, 2007), para cada variable se utiliza un nombre y un valor, representadas en la Tabla 1.1.

Variable lingüística	Valores lingüísticos
Velocidad	Muy alta, alta, moderada, baja, muy baja
Nivel de llenado	Muy lleno, lleno, más o menos lleno, poco nivel, muy bajo nivel
Estatura	Muy Alto, Alto, media, bajo, pequeño, enano

Tabla 1.1 Variables y valores lingüísticos

Fuente: Propia

### 1.2.2 Funciones de membresía

Las funciones de membresía convencionales ver figura 1.9, que permiten realizar un mapeo, están representadas en grados de membresía de 0 a 1 donde 1 representa pertenencia al conjunto y 0 ninguna pertenencia al conjunto, la forma de la función de membresía se elige de acuerdo al problema a ser resuelto y pueden ser representados por las siguientes funciones:

- Función singleton
- Función triangular
- Función trapezoidal

- Función gaussiana
- Función sigmoidea

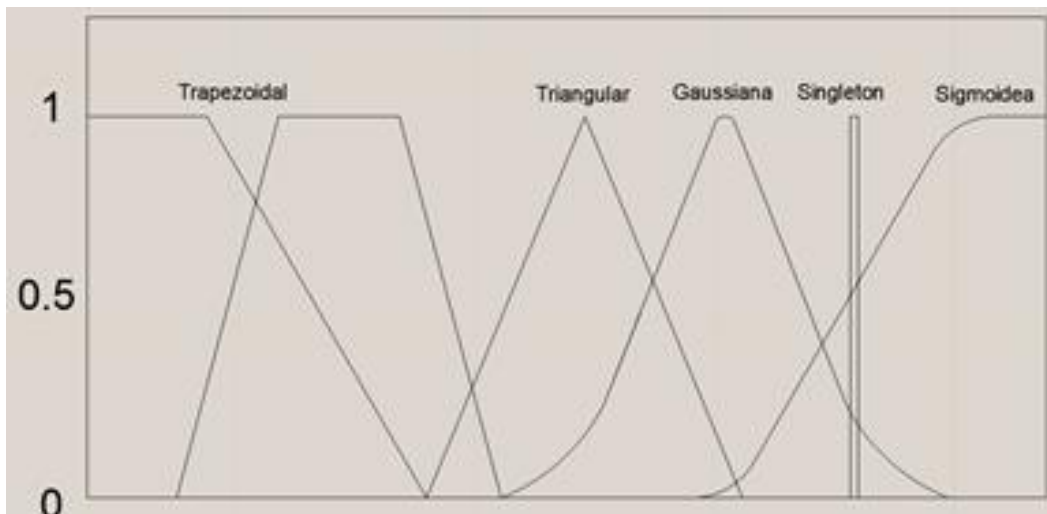


Figura 1.9 Funciones de membresía

### 1.2.3 Sistema de inferencia difusa

El grado de pertenencia de las variables lingüísticas se evalúa en la inferencia difusa, es la parte de control del programa encargada de procesar las reglas contenidas en la base del conocimiento, tomando en cuenta los datos nuevos y el estado del sistema, produciendo el cambio de estado para empezar un nuevo ciclo.

El mecanismo de inferencia es el que obtiene una salida difusa a partir de una entrada difusa, que incluye un proceso de control entrada de datos, conjunto de reglas de inferencia, defuzzificación en la salida y creación de un nuevo ciclo. Donde existe un conjunto difuso por cada variable a defuzzificar (Zaragoza Hernández et al., 2017).

Se define la regla de inferencia If-then, tipo Si... entonces, que tiene una condición 1: antecedente y una condición 2: consecuente:

- Proposición simple:

**SI** (condición precisa) **ENTONCES** (consecuente)

- Proposición compuesta:

**Si** (condición precisa 1) **conectivo** (AND, OUR, NOT) (condición precisa 2) **ENTONCES** (acción)

### 1.2.4 Defuzzificación

Las estimaciones difusas del fuzzificador y transforma en valores, por lo que el sistema comprende las órdenes y desarrolla las actividades preparadas. Se refiere al proceso inverso de la fuzzificación, este paso es más complejo por la variedad de métodos matemáticos y la presentación de las conclusiones en base a los conjuntos difusos. No necesariamente una entrada difusa tiene una salida difusa, sino un valor procesado (García, 2019). Las variables a defuzzificar están en función de los conjuntos difusos, es decir la salida del sistema debe ser un valor medible que interprete la variable lingüística.

#### Métodos de defuzzificación

Dentro de los métodos utilizados en la defuzzificación las más comunes son:

- Centro de gravedad o Centroide

Encuentra el centro del área del conjunto difuso que resulta de combinar todos los conjuntos difusos obtenidos o también calcula el promedio ponderado de la salida. Son empleados en modelado y control difuso pues permite un control preciso.

$$y_d = \frac{\int_S y \mu_Y(y) dy}{\int_S \mu_Y(y) dy}$$

- Media de los máximos (MOM)

Utilizan el valor máximo del conjunto resultante, calcula el promedio de las variables que tienen el mayor grado de membresía.

$$z = \sum_{j=1}^l \frac{z_j}{l}$$

### 1.2.5 Estado del arte

La robótica está siendo fundamental en el desarrollo de la industria en la actualidad, la aplicación de dicha tecnología es variada, puede desarrollarse en áreas de: soldadura, pintura, ensamblaje, etc. La lógica difusa facilita la interpretación de situaciones complejas, ya que no hay una transición clara entre falso y verdadero, pero permite tomar decisiones a través de la interpretación de funciones de membresías.

Según el estudio de (Colón et al., 2016) las decisiones del robot es modelado mediante una configuración de lógica difusa, menciona además que debe preferirse los conjuntos de entrada de forma trapezoidal así como el de tipo Singleton para definir conjuntos difusos de salida. El sistema difuso de inferencia utilizado es de tipo Mamdani. El uso de lógica difusa permite al robot mejorar el desempeño de caminar, ya que se aproxima a su objetivo de manera suave y gradual, disminuyendo las posibilidades de choque o derribo involuntario.

Otro estudio (Suarez et al., 2018), trata del desarrollo de un algoritmo de control híbrido basado en lógica difusa, y en el que el control difuso puede ser expresado de manera acertada como un control basado en la experiencia que interpreta el sentido común, en lugar de números o ecuaciones. El objetivo principal de estudio es el desarrollo e implementación de un controlador basado en lógica difusa que permita al robot el seguimiento de una trayectoria planeada.

El estudio de (Alba Mejía, 2019) considera a la lógica difusa como una rama de la inteligencia artificial así como una extensión de la lógica tradicional (Booleana), donde las variables de pertenencia utiliza patrones de razonamiento similares a los del pensamiento humano. Presenta la simulación del control difuso tipo PID además de tener la función de pertenencia tipo Singleton y en la simulación utiliza el software de MATLAB.

Según el estudio (Jesús et al., 2019) trata de la implementación de un sistema de control difuso tipo Mamdani y en la defuzzificación utiliza el método de Centro de Máximos, aunque el método del Centroide es el más utilizado ya que permite un control preciso, adicional al control difuso le permite implementar un control de razonamiento similar al de un ser humano en un sistema computacional, con la ventaja de que no sea necesario un sistema de comunicación inalámbrica o alámbrica hacia un computador.

La distancia del robot desde la posición inicial hasta la posición deseada, donde  $(X_r, Y_r)$  es el vector de posición inicial del robot y  $(X_t, Y_t)$  es el vector de posición deseada, se calcula de la siguiente manera:

$$P = \sqrt{(X_r - X_t)^2 + (Y_r - Y_t)^2}$$

En el estudio de (Alzate Gómez et al., 2007), presenta un control difuso implementado en una plataforma móvil para realizar el seguimiento de trayectorias, a través del movimiento de dos servomotores y la lectura de tres sensores de línea. En el desarrollo del control difuso se aplica el conector AND y se selecciona la norma t mínimo, definida como:

$$t[\mu_A(x), \mu_B(x)] = \mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

De acuerdo a lo anterior se usa el sistema de inferencia de Mamdani definida como:

$$\mu_{QMM}(x, y) = \min[\mu_{FP1}(x), \mu_{FP2}(y)]$$

Para encontrar las salidas del sistema, se utiliza el sistema de inferencia mínimo, definida de la siguiente manera:

$$\mu_{B'}(y) = \max \left[ \sup \min \left( \mu_{A'}(x), \mu_{A_1'}(x_1), \dots, \mu_{A_n'}(x_n), \mu_{B'}(y) \right) \right]$$

En resumen, mediante la lógica difusa el control de robots móviles está basado en comportamientos que pueden ser sintetizados en reglas si - entonces (if - then) de modo que el conocimiento queda expresado en términos lingüísticos fáciles de entender, además la lógica difusa permite el diseño de controladores sujetos a la incertidumbre, los ruidos y las perturbaciones.

El sistema de control tipo Mamdani que dispone de las fases de fuzzificación y defuzzificación, permiten convertir datos numéricos de los conjuntos difusos, a través del sistema de inferencia difusa y el conjunto de reglas de membresía. En el desarrollo del proyecto se utiliza el método del Centroide, a través del uso del software de simulación de MATLAB.

## 2. METODOLOGÍA

### 2.1 Introducción

La robótica móvil se ha desarrollado en los últimos años como una alternativa al uso de los manipuladores ya que éstos están anclados y su estructura es fija. Los robots móviles son versátiles ya que poseen un espacio de trabajo ilimitado, el desplazamiento de estos se los realiza por elementos de locomoción con la capacidad de adaptarse a todo tipo de terreno o ambiente. La autonomía es otra de las ventajas de trabajo ya que no requieren necesariamente de supervisión humana lo que les permite responder a situaciones cambiantes (Reyes, 2011, p. 509). La robótica móvil permite mediante un sistema de locomoción desplazarse dentro de un ambiente conocido o desconocido, por lo que es necesario conocer la posición de este.

### 2.2 Diseño y modelación del control inteligente

En este capítulo se determinan las especificaciones de diseño del robot, en base a los sistemas: mecánico, electrónico, de percepción y la modelación matemática del control Fuzzy.

#### 2.2.1 Diseño sistema mecánico

El diseño es un proceso lineal basado en identificar las fases del diseño; diseño conceptual, diseño preliminar y diseño detallado, mostrados en la Figura 2.1.

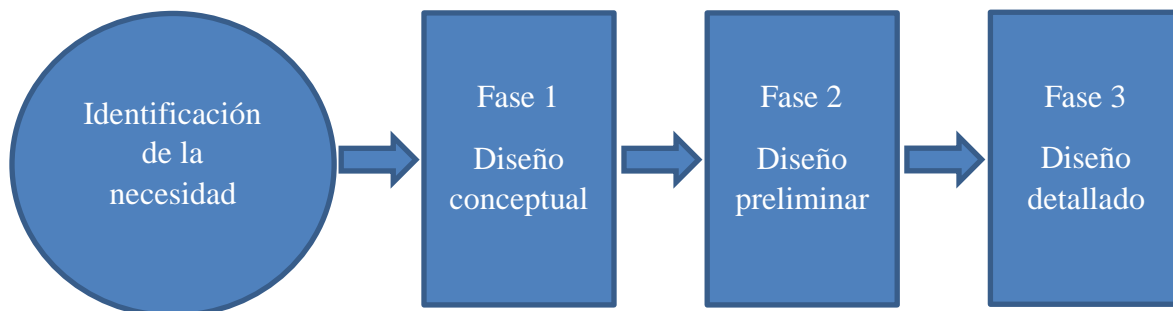


Figura 2.1 Fases del diseño

Fuente: (Bernal, 2005)

La necesidad identificada es el diseño y simulación de un robot móvil con navegación autónoma para el traslado de coches entre dos estaciones de trabajo en la empresa Aymesa, su función principal sería de remolque de 1 hasta 3 dispositivos.

### Especificaciones de diseño

En el diseño conceptual se buscan soluciones al problema basándose en la revisión de robots que cumplen similar función, están representadas en las Figuras 2.2 y 2.3.



Figura 2.2 Robot KIVA  
Fuente: (Amazon, n.d.)



Figura 2.3 Robot MIR 200  
Fuente: (Mobile Industrial Robots, n.d.)

Los robots KIVA pesan 145 kg pueden levantar pesos de 340kg, desplazamiento a una velocidad de 5.5 km/h conectados a WIFI, mientras que los robots MIR 200 pueden cargar hasta 200 kg, pero con remolque hasta 500 kg, con una velocidad de 4 km/h. En base al estudio de los dos modelos tienen locomoción por tracción diferencial se toma como referencia la segunda opción ya que permite el remolque de dispositivos una de las características de este estudio. Dicho diseño se adapta a las especificaciones mostradas en la Tabla 2.1.

El diseño preliminar del robot móvil que define la estructura, componentes y la selección de materiales inherentes para la fabricación de un robot móvil, ver ANEXO 1.

Diseño de un robot para el traslado de dispositivos	
Ítem	Especificaciones
1	La locomoción será de tracción diferencial de 6 ruedas 2 ruedas intermedias principales serán de tracción y dirección
2	El diseño será modular y que no contenga puntos de soldadura
3	La superficie de trabajo debe ser plana y rígida sin desniveles en el piso
4	La velocidad máxima será de 7,2 km/h $\approx$ 2 m/s
5	El diseño de la estructura y sistema de remolque será de Acero ASTM 36
6	Las dimensiones del robot: L= 1000 mm   Ancho= 500 mm   Altura= 500 mm
7	La masa total del robot no será superior a 100 kg

Tabla 2.1 Especificaciones

Fuente: Propia

## Parámetros de diseño

### Peso del robot

La carga que puede soportar la placa es la sumatoria de pesos que formarán parte del robot, se ha considerado según las especificaciones de la tabla 2.1 la masa total de diseño del robot de 100 kg, según esto se calcula el peso según la ecuación 1.1 por lo que se considera este valor de diseño.

$$\begin{aligned}
 m_{robot} &= 100 \text{ kg} \\
 W_{robot} &= m_{robot} * g \\
 W_{robot} &= (100\text{kg}) * \left(9,81 \frac{\text{m}}{\text{s}^2}\right) \\
 W_{robot} &= 981 \text{ N}
 \end{aligned}
 \tag{Ec. 1.1}$$

Donde;

m: Masa (kg)

g: Gravedad (m/s<sup>2</sup>)

W: Peso (N)



## Fuerza de rozamiento estática y cinética

Se debe determinar la fuerza que se requiere el robot móvil para trasladar a través de remolque de 1 a 3 coches con una masa máxima de 100 kg, en trayectoria recta con un coeficiente de fricción cinética de 0,8, de acuerdo a la Tabla 2.2, de los coeficientes de fricción (Alejandro Medina; Jesús Ovejero, 2010, p. 14) que corresponde al contacto de caucho sobre hormigón armado (seco).

<b>Materiales</b>	$\mu_e$	$\mu_c$
Acero sobre acero	0,7	0,6
Vidrio sobre vidrio	0,9	0,4
Teflón sobre teflón	0,04	0,04
Caucho sobre hormigón (seco)	1,0	0,8
Caucho sobre hormigón (mojado)	0,3	0,25
Esquí sobre nieve	0,1	0,03
Hielo sobre hielo	0,1	0,03
Articulaciones humanas	0,01	0,003

Tabla 2.2 Coeficientes de fricción

Fuente: (Alejandro Medina; Jesús Ovejero, 2010, p. 14)

La fuerza mínima requerida para que cada coche empiece a moverse está determinada por la fuerza de rozamiento estática, de acuerdo con la ecuación 1.2

$$F = F_{re} \quad \text{Ec. 1.2}$$

$$F_{re} = \mu_e \cdot N$$

Debemos tomar en cuenta que se ha considerado que se trasladará dentro de una superficie horizontal. Si la fuerza y el desplazamiento tienen la misma dirección y sentido, el ángulo es 0, por lo que hay que usar la ecuación 1.3

$$\sum F = m \cdot a \quad \text{Ec. 1.3}$$

$$\sum F = m \cdot 0$$

$$N - P = 0$$

$$N = P$$

$$N = m \cdot g$$

Reemplazamos ecuación 1.3 en ecuación 1.2 tendremos la ecuación 1.4, por tanto:

$$F_{re} = \mu_e \cdot N$$

$$F_{re} = \mu_e \cdot (m \cdot g)$$

$$F_{re} = 1 \cdot \left(100kg \cdot 9.81 \frac{m}{s^2}\right)$$

$$F_{re} = 981 N$$

Donde;

F: Fuerza (N)

N: Normal (N)

$F_{re}$ : Fuerza de rozamiento estático (N)

$\mu_e$ : Coeficiente de fricción estático (adimensional)

m: Masa del cuerpo (kg)

g: Gravedad ( $\frac{m}{s^2}$ )

La fuerza de rozamiento cinética viene expresada en la ecuación 1.5

$$F = F_{rc} \quad \text{Ec. 1.5}$$

$$F_{rc} = \mu_c \cdot N$$

$$F_{rc} = \mu_c \cdot (m \cdot g)$$

$$F_{rc} = 0.8 \cdot \left(100kg \cdot 9.81 \frac{m}{s^2}\right)$$

$$F_{rc} = 784,8 N$$

Donde;

F: Fuerza (N)

N: Normal (N)

$F_{rc}$ : Fuerza de rozamiento cinético (N)

$\mu_c$ : Coeficiente de fricción cinético (adimensional)

m: Masa del cuerpo (kg)

g: Gravedad ( $\frac{m}{s^2}$ )

## Torque

Para objeto de cálculo consideramos el valor máximo de la fuerza de rozamiento cinético de 784,8 N y el radio de la rueda de 0,15m, ver ecuación 1.6

$$T = F_{rc} * r_{rueda} \quad \text{Ec. 1.6}$$

$$T = 784,8N * 0,15m$$

$$T = 117,72Nm$$

Donde el torque de cada rueda viene dado por ecuación 1.7

$$T_{rueda} = \frac{T}{2} \quad \text{Ec. 1.7}$$

$$T_{rueda} = \frac{117,72Nm}{2}$$

$$T_{rueda} = 58,86Nm$$

Donde;

$F_{rc}$ : Fuerza de rozamiento cinético (N)

$r_{rueda}$ : Radio de la rueda (m)

T: Torque (Nm)

## Potencia requerida de diseño

$$P = T_{rueda} * \omega_{diseño} \quad \text{Ec. 1.8}$$

Debemos calcular la velocidad angular de la rueda ( $\omega_{diseño}$ ) ver ecuación 1.9

$$\omega_{diseño} = \frac{V_{diseño}}{r_{rueda}} \quad \text{Ec. 1.9}$$

$$\omega_{diseño} = \frac{2 \text{ m/s}}{0,15\text{m}}$$
$$\omega_{diseño} = 13,3 \text{ rad/s}$$

Reemplazando 1.9 en 1.8 tenemos lo siguiente:

$$P = T_{rueda} * \omega_{diseño}$$
$$P = 58,86\text{Nm} * 13,3 \text{ rad/s}$$
$$P = 782,8 \text{ W}$$
$$P = 1,05 \text{ HP}$$

Donde;

$T_{rueda}$ : Torque de la rueda (Nm)

$\omega_{diseño}$ : Velocidad angular diseño (rad/s)

$V_{diseño}$ : Velocidad de diseño (m/s)

$r_{rueda}$ : Radio de la rueda (m)

$P$ : Potencia (HP)

## Diseño de la estructura

La estructura del robot es el elemento principal en donde se van a ensamblar todas las partes y piezas requeridas para proporcionar un sistema de locomoción estable, que incluye la motricidad de las ruedas, el sistema electrónico y el sistema de percepción. Se utilizaría plancha de acero ASTM A36 ver Tabla 2.3, entre las características que podemos mencionar es un acero de bajo contenido de carbono inferior al 0,3 %, que facilita el conformado, el mecanizado y la soldadura, ver ANEXO 2. Para mitigar la corrosión de la estructura se aplicaría una capa de pintura anticorrosiva.

Especificaciones Acero ASTM A36	
Densidad ( $kg/m^3$ )	7850
Límite de fluencia (MPa)	250
Límite de rotura (MPa)	400
Módulo de elasticidad (GPa)	200
Elongación	20%

Tabla 2.3 Especificaciones Acero ASTM A36

Fuente: (*ASTM A36 steel*, n.d.)

### **Análisis estructural de los elementos mecánicos del robot**

Se muestra el análisis por elementos finitos en Solidworks de los elementos críticos del robot como son: la estructura, el elemento de arrastre macho que es fijo y el elemento de arrastre hembra que es portátil, el material para los elementos críticos es acero ASTM36.

La simulación de los esfuerzos está representada en 4 figuras a) Restricciones y fuerzas, b) Tensiones, c) Desplazamientos y d) Factor de seguridad, ver ANEXO 3. El software de Solidworks se aplica los métodos de análisis de elementos finitos, para individualizar los componentes del diseño en elementos sólidos, el mismo que emplea el método de elementos finitos para el cálculo de desplazamientos, tensiones y el factor de seguridad.

El factor de seguridad o coeficiente de seguridad conocido como factor de diseño  $n$ , es una medida de seguridad relativa de un elemento mecánico bajo la acción de una carga. El factor de seguridad con valores inferiores a 1 nos indica que el material ha fallado, con valores iguales a 1 nos indica que el material va a fallar, y con valores superiores a 1 el material no va a fallar por lo que se considera material seguro. En edificación e ingeniería se usan factores de seguridad entre 2 y 5, en ingeniería aeronáutica se utilizan factores de seguridad entre 1,5 y 2 (Ruiz, 2015).

## 2.2.2 Diseño del sistema electrónico

En el diseño del sistema electrónico constituye el circuito eléctrico y electrónico requerido para el funcionamiento del robot, pueden considerarse los sistemas integrados para aplicaciones analógicas o digitales. Los tipos de circuitos utilizados en el funcionamiento de los robots de acuerdo con el diagrama de bloques ver Figura 2.4, son los siguientes:

### Diagrama de bloques del sistema electrónico

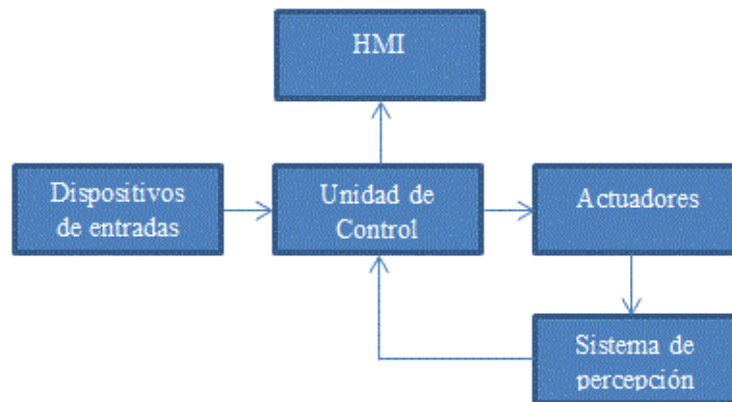


Figura 2.4 Sistema electrónico

Fuente: Propia

### Unidad de control

Se refiere a las tarjetas electrónicas que se encargan del procesamiento de información y de la monitorización de los sistemas físicos y del estado interno del robot, están compuestos por entradas a través de los sensores y las salidas a través de los actuadores en este caso mediante motores paso a paso.

- **Arduino Mega**

Arduino es una plataforma electrónica abierta para la creación y construcción de prototipos basados en software y hardware libre. El Arduino mega (*Arduino Official Store*, n.d.), es una tarjeta electrónica digital que transforman datos digitales en señales analógicas y viceversa, además de procesar los datos de los sensores, basado en el microcontrolador ATmega2560 ver Figura 2.5, tiene 54 pines de entrada / salida, 16

entradas analógicas, 4 UART (puertos seriales), un oscilador de cristal 16 MHz, conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio, ver ANEXO 4.



Figura 2.5 Arduino Mega 2560

Fuente: (Arduino Official Store, n.d.)

## Actuadores

Los actuadores se encargan del movimiento o desplazamiento, pueden ser hidráulicos, eléctricos o neumáticos, el proyecto se centrará en los actuadores eléctricos accionados por energía eléctrica a través de una batería.

- **Motor paso a paso**

El motor paso a paso según (Mecafenix, 2017) es una máquina electromecánica capaz de transformar energía eléctrica en energía mecánica rotacional a través de pulsos, es decir que por cada pulso el rotor del motor gira un ángulo determinado ideal para mecanismos donde requieren movimientos precisos, ver Figura 2.6. Las inercias del arranque y de las paradas impiden que el robot alcance la velocidad nominal de manera instantánea, por lo tanto, la frecuencia de los pulsos debe aumentarse progresivamente.

- **Motor CNC Nema 34**

Como parte de las especificaciones el sistema de locomoción que será de tracción diferencial ver Figura 2.7, de 6 ruedas 2 ruedas intermedias principales serán de tracción y dirección, por lo que es necesario utilizar dos motores paso a paso NEMA 34 12Nm 6A

(ANEXO 5) y debido a la variación de pulsos en las ruedas varía la dirección del robot y así evitar choque con obstáculos.



Figura 2.6 Motor a pasos

Fuente: (Compañía Levantina de Reductores, 2020)

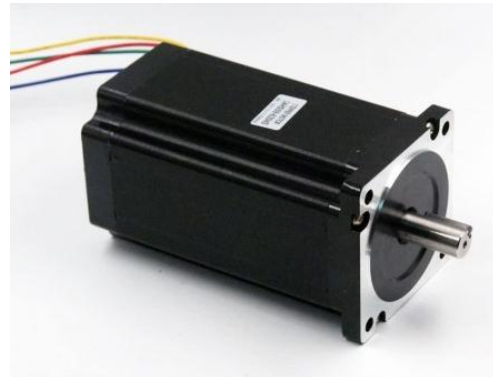


Figura 2.7 Motor Nema 34

Fuente: (Amazon, 2020)

Por cada revolución de la rueda para este tipo de motor viene determinado por 200 pulsos es decir tenemos  $1,8^\circ$  por cada pulso, a su vez cada rueda se controla de manera independiente y para que el robot logre el movimiento en línea recta requiere que la velocidad de los dos motores sea igual.

- **Driver**

Es un controlador que envía órdenes a cada motor para aumentar los pulsos y por ende cambiar la dirección sea éstos para: girar, virar en una esquina o evadir obstáculos, otra de las funciones es mantener la velocidad con la misma cantidad de pulsos en ambos motores para que el robot móvil se mantenga en línea recta. El driver MA860A es compatible con Arduino mega, ver Figura 2.8, con regulación de potencia desde 2 A hasta 7,2 A.

**Fuente de alimentación:** es la energía que va a consumir todo el sistema eléctrico y electrónico para la operación de los robots móviles, se utilizará una batería de 12 V pero es recomendable utilizar siempre un emulador de batería pues resulta seguro, evitando cualquier fuente de peligro por la repetitividad de pruebas del robot (Zollo, 2015), la batería provee energía para el funcionamiento de un robot en base al peso del robot, sistema de locomoción, actividades que vaya a ejecutar, tiempo de operación, almacenamiento de energía y sistema de comunicación.





Figura 2.8 Driver

Fuente: (Stepperonline, 2020)

## Sistema de control

Es el sistema que se encarga de controlar los motores, manejo de información del sistema de percepción, algoritmos de navegación y algoritmos de procesamiento de información, las principales herramientas utilizadas en la robótica móvil son: lógica difusa, redes neuronales, inteligencia artificial, etc. (Bermúdez, 2002).

A partir de la información recibida del sistema de percepción, el sistema de control se encarga de la decisión de los actuadores, el sistema de control tiene dos niveles:

- Control de bajo nivel: actúa sobre el sistema de tracción y dirección
- Control de alto nivel: Evita obstáculos

## Control inalámbrico

La comunicación sería mediante un módulo XBee, que implementa un enlace de comunicaciones inalámbricas punto a punto a una frecuencia de 2,4 GHz y que puede transmitir señal hasta 30 metros en espacios cerrados y 100 metros espacios abiertos, el sistema se controla a si mismo por la transferencia de caracteres utilizando un puerto USB, ver Figura 2.9.

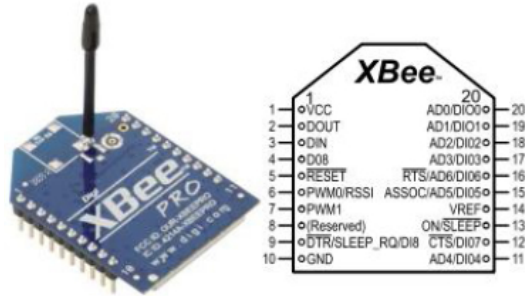


Figura 2.9 XBee

Fuente: (Aprendiendo Arduino, 2020)

### 2.2.3 Diseño del sistema de percepción

La robótica móvil se caracteriza por su nivel de autonomía que se basa en el sistema de navegación, por lo que se debe disponer de sensores adecuados que permitan captar la información del entorno, constituyéndose en el sistema de percepción del robot. El sistema de percepción dentro de un robot móvil tiene los siguientes objetivos:

- Permitir una navegación segura, que evite colisiones.
- Trasladarse de una posición inicial a una posición deseada evadiendo obstáculos.
- Estimar la posición del robot.

#### Sensores

Los sensores son elementos encargados de transformar o convertir una medida física en medidas eléctricas.

#### Sensores de presencia y proximidad

El propósito de este tipo de sensores es determinar la ubicación de un objeto en un intervalo de distancia. Se requiere la información de dos variables: proximidad y distancia. Los principales tipos de sensores de proximidad son los siguientes:

#### Sensores ultrasónicos

Este tipo de sensor está constituido por un emisor y un receptor de ondas de ultrasonido cuya frecuencia de 20KHz supera la capacidad auditiva humana, es un sensor de

proximidad sin contacto físico que puede detectar objetos transparentes pero que no pueden aplicarse en ambientes ruidosos.

- **Sensor ultrasónico HC-SR04**

El sensor ultrasónico o de sensor de proximidad HC-SR04 (ver ANEXO 6), es compatible con la placa Arduino pues permite la medición de objetos en un rango de 0,02 a 4,5m y sólo requiere de un terminal E/S. Se requiere el uso de 3 sensores para abarcar el ancho del robot según el esquema de acuerdo con la Figura 2.10.

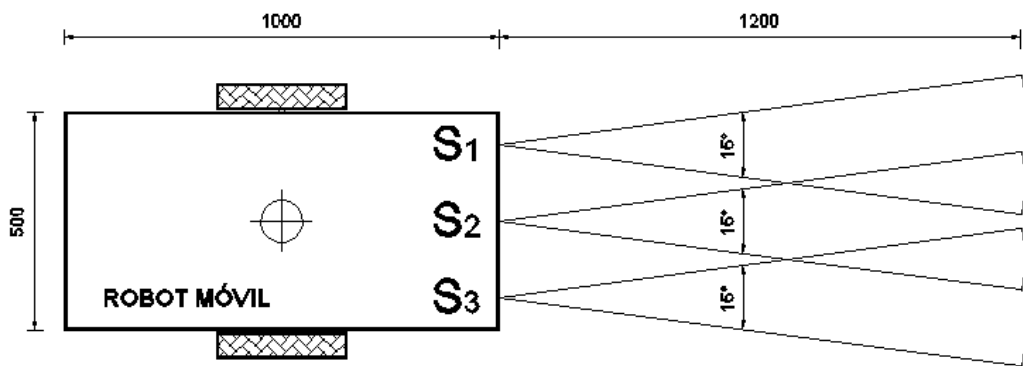


Figura 2.10 Esquema de los sensores ultrasónicos

Fuente: Propia

## Sensores ópticos

Este tipo de sensor es similar al sensor ultrasónico está constituido por un emisor y un receptor, pero de un diodo emisor de luz, ver Figura 2.11.

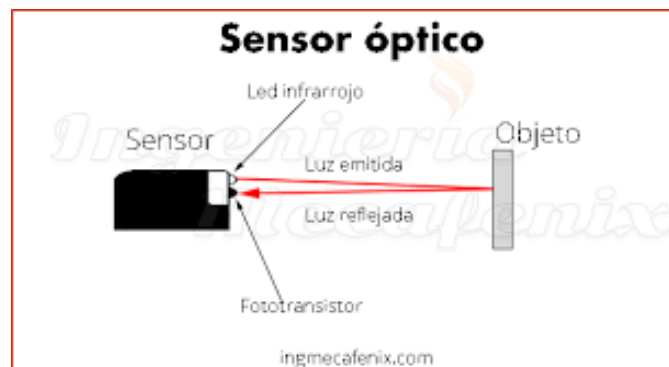


Figura 2.11 Sensor óptico

Fuente: (Mecafenix, 2017)

## Sensores de visión

En este tipo de sensores la información del entorno es captada mediante una cámara digital como elemento de visión artificial, un ejemplo de estas puede ser una cámara CCD de carácter pasivo que no emite energía del entorno en la captación de las imágenes, ver Figura 2.12.

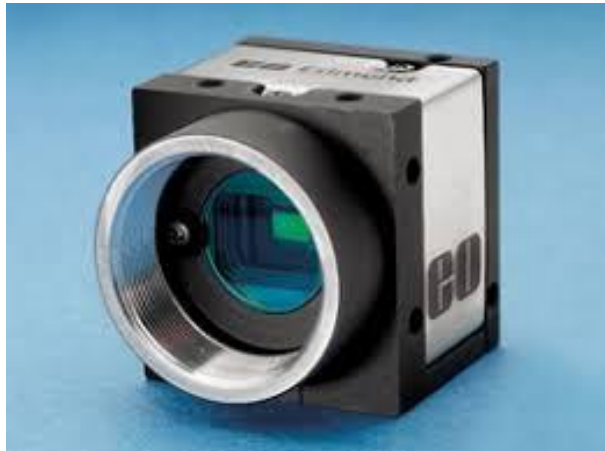


Figura 2.12 Cámara CCD

Fuente: (*Cámara CCD*, n.d.)

## Sistema de navegación

Un robot percibe el entorno y transmite la información de sus sensores de acuerdo a las variables físicas que tiene que medir. La información que nos proporciona los sensores es una función importante para cualquier sistema de navegación autónoma, por lo que las características principales son:

- Sistemas de guías
- Odometría
- Faros
- Mapeo
- Planeación de trayectorias

## **Sistemas de guías**

Son sistemas que cuentan con trayectorias definidas pueden ser cables guías colocados en el piso, rayas pintadas, columnas magnéticas o rayos láser a través de sensores láser. Estos sistemas están limitados a un número fijo de caminos, la desventaja de este tipo de navegación son los obstáculos en la guía por lo que pueden chocar, o para eliminar esta restricción sería retirar los obstáculos.

## **Odometría**

Es la posición y orientación del robot, sin embargo, este tipo de control falla si varía las condiciones de la rueda sean por desgaste en el arranque o deformidades en el piso. Se obtiene la posición del robot por medio de decodificadores incrementales ligados a las ruedas robot y para corregir los errores por las condiciones de las ruedas es necesario implementar sensores externos como: sensores infrarrojos, láser, cámaras de video.

## **Faros**

Son utilizados para corregir los errores del control por Odometría, calculando su posición y corrigiendo los errores de su trayectoria, mediante sensores infrarrojos.

## **Mapeo**

Para lograr mayor autonomía es indispensable un mapa y de algoritmos que permitan manipular datos, de esta manera conocer la posición del robot dentro del ambiente de trabajo, la visión artificial a través de sensores de ultrasonido o rayos láser puede compararse con el mapa y determinar la posición del robot.

## **Planeación de trayectorias**

En la planeación de trayectorias es el conjunto de técnicas que permiten llevar a un robot desde una posición inicial a una posición final, y que pueda navegar libremente en la que pueden existir o no la presencia de obstáculo, encontrando la trayectoria adecuada que permita movilizarse entre dos estaciones de trabajo.

## 2.2.4 Modelación matemática del control Fuzzy

Sea  $X$  es el conjunto universal y  $x$  los elementos de  $X$ , entonces un conjunto difuso  $A$  en  $X$  se define como una serie de pares ordenados  $A = \{(x, \mu_A(x)) \text{ donde } x \in X\}$ , la función de membresía está representada por  $\mu_A(x)$  para el conjunto difuso  $A$ , donde cada elemento de  $X$  le corresponde un grado de membresía entre 0 y 1.

El conjunto difuso se encuentra asociado a una variable lingüística caracterizada por  $(X, T(x), x)$  donde  $x$  es la variable,  $T(x)$  son los valores lingüísticos de las variables de entrada y salida representadas por nombres.

El sistema de tracción diferencial permite al robot representar en un plano  $X, Y$  la posición inicial del punto  $(X_{pi}, Y_{pi})$ , por lo que las variables de navegación autónoma son: el ángulo de dirección  $\theta$  y la velocidad del robot, ver Figura 2.13.

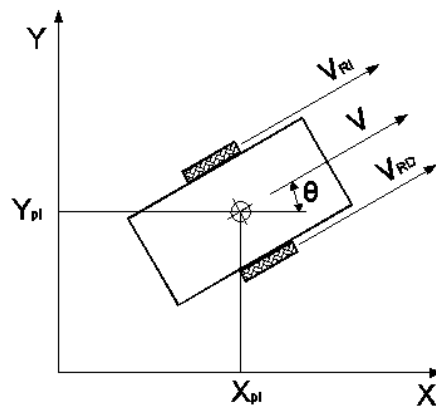


Figura 2.13 Tracción diferencial

Fuente: (Venegas, 2006)

Dónde;

$\theta$  : Ángulo de dirección ( $^{\circ}$ )

$V$ : Velocidad (m/s)

$V_{RI}$ : Velocidad rueda izquierda (m/s)

$V_{RD}$ : Velocidad rueda derecha (m/s)

$X_{pi}$ : Posición inicial eje X

$Y_{pi}$ : Posición inicial eje Y

X, Y: Ejes

Adicional se considera tres características de navegación en función de la velocidad:

1. Si la velocidad de la rueda izquierda es mayor que la velocidad de la rueda derecha el robot se dirige hacia la derecha,
2. Si la velocidad de la rueda derecha es mayor que la velocidad de la rueda izquierda el robot se dirige hacia izquierda,
3. Si las velocidades de ambas ruedas son iguales el robot se dirige en línea recta.

Para crear el control difuso se define lo que se pretende lograr en este caso sería: simular un robot con navegación autónoma para el traslado de coches entre dos estaciones de trabajo, que deberá recorrer desde una posición inicial hasta una posición deseada, evadiendo cualquier obstáculo que se presente en el trayecto, por lo que se propone el desarrollo de dos controles difusos: ángulo de dirección del robot y la velocidad. En la Figura 2.14, se muestra la configuración del robot para aplicar la navegación autónoma.

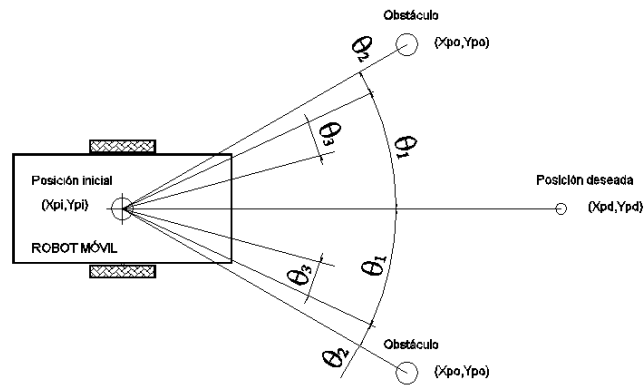


Figura 2.14 Configuración de navegación autónoma

Fuente: Propia

Dónde;

$\theta_1$  : Ángulo error entre la posición del robot y la posición del obstáculo ( $^\circ$ )

$\theta_2$  : Ángulo error entre la posición del robot y la posición deseada ( $^\circ$ )

$\theta_3$  : Ángulo de dirección del robot ( $^\circ$ )

$(X_{pi}, Y_{pi})$ : Coordenadas del robot en la posición inicial

$(X_{po}, Y_{po})$ : Coordenadas del obstáculo respecto de la posición inicial

$(X_{pd}, Y_{pd})$ : Coordenadas de la posición deseada respecto de la posición inicial

## 2.3 Simulación computacional

La simulación de un robot requiere de un software que permita la creación de modelos y entornos virtuales para realizar el análisis similar a un robot real, mediante la aplicación de algoritmos que simule tareas o comportamientos en la navegación autónoma.

### 2.3.1 Programación en Matlab del control inteligente

#### Matlab

Matlab es una herramienta informática que surgió para resolver cálculos matemáticos, especialmente operaciones con matrices. Constituye un entorno de trabajo sencillo, para el desarrollo de código, análisis de datos, computación numérica y crear gráficos para representar los datos.

#### Operaciones

Las operaciones que se pueden realizar en conjuntos difusos son similares a las que se realizan en la lógica tradicional. Las operaciones más utilizadas son unión, intersección y complemento, en el presente estudio vamos a utilizar la operación unión representado en el Matlab por el conector AND.

- **Unión**

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

- **Intersección**

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$$

- **Complemento**

$$\mu_A(x) = 1 - \mu_A(x)$$

$$\mu_B(x) = 1 - \mu_B(x)$$

La programación en Matlab del control inteligente se basa en la utilización del Fuzzy Logic Toolbox, que permite la aplicación de dos sistemas de inferencia: tipo Mamdani y tipo Sugeno, para el desarrollo de este proyecto utilizaremos el tipo Mamdani que contiene tres pantallas:



- Editor del sistema de inferencia difuso (FIS)
- Editor de las funciones de membresía (MF)
- Editor de las reglas de membresía (Rules)

## Editor del sistema de inferencia difuso (FIS)

El sistema de inferencia que se trabajará en este proyecto es de tipo Mamdani, abrimos Matlab y tecleamos la palabra fuzzy en la ventana y damos enter, nos mostrará la pantalla del editor de inferencia difuso FIS con Mamdani por default, ver Figura 2.15.

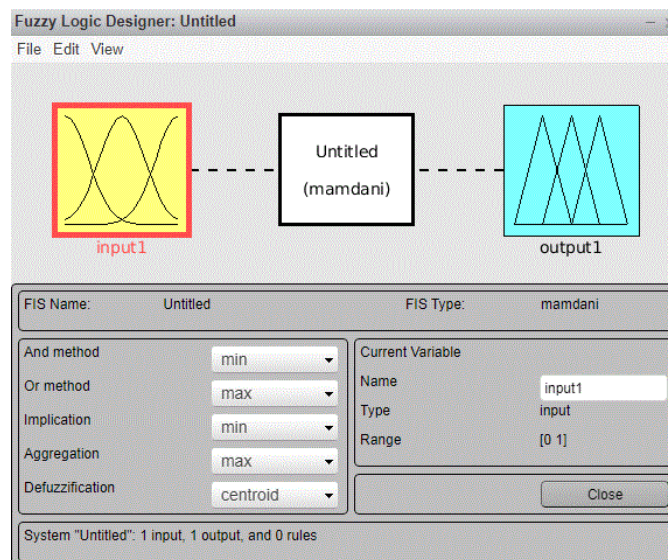


Figura 2.15 Pantalla del editor del sistema de inferencia difuso FIS

Fuente: Propia

En el menú Edit se dispone de las opciones añadir variables de entrada (Input) y de salida (Output), ver Figura 2.16 también están las funciones de membresía (Membership Function Editor) ver Figura 2.17 y las reglas de membresía (Rules) ver Figura 2.18.

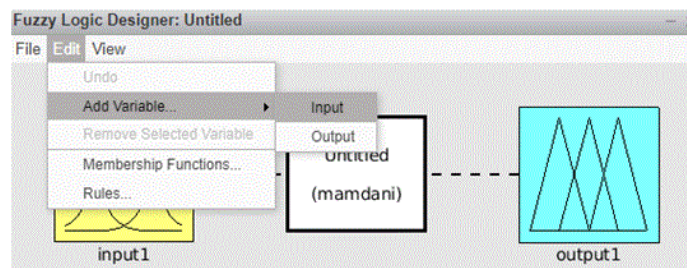
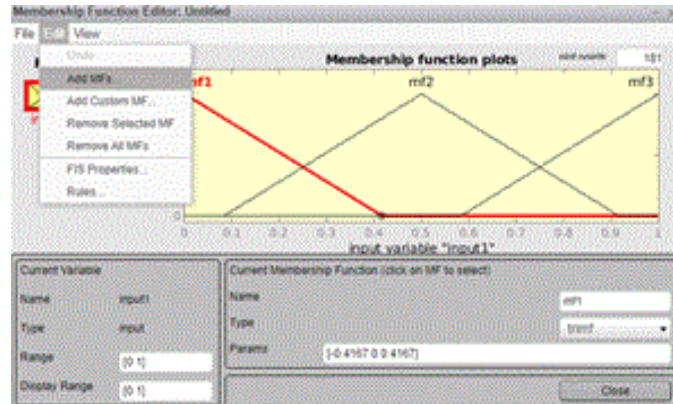


Figura 2.16 Add variable (Input, Output)

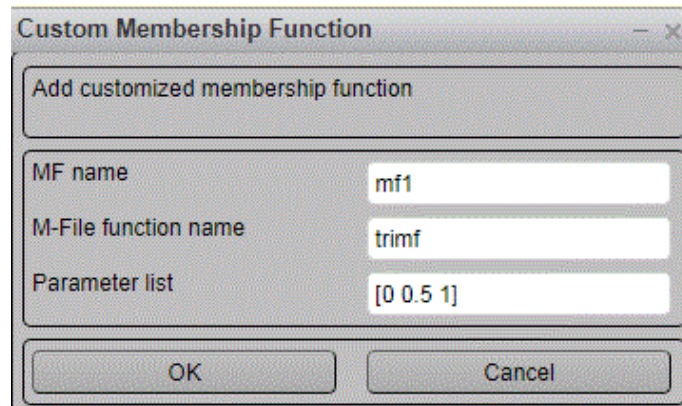
Fuente: Propia

## Editor de las funciones de membresía (MFs)

En el menú Edit de las MFs están los submenús de añadir funciones de membresía (MFs), añadir editor personalizado (Add Custom MFs), eliminar selección de membresía (Remove Selected MFs), eliminar todas las funciones de membresía (Remove All MFs), se puede colocar el rango de valores de cada función de membresía, ver Figura 2.17.



(a)



(b)

Figura 2.17 Membership Function (a) Edit y (b) Custom Membership Function

Fuente: Propia

## Editor de las reglas de membresía (Rules)

En el menú Edit de las reglas de membresía, se visualizan las entradas y salidas, pueden tener las operaciones(connection) AND y OR, adicional de los submenús de eliminar, añadir o cambiar reglas de membresía, ver Figura 2.18.

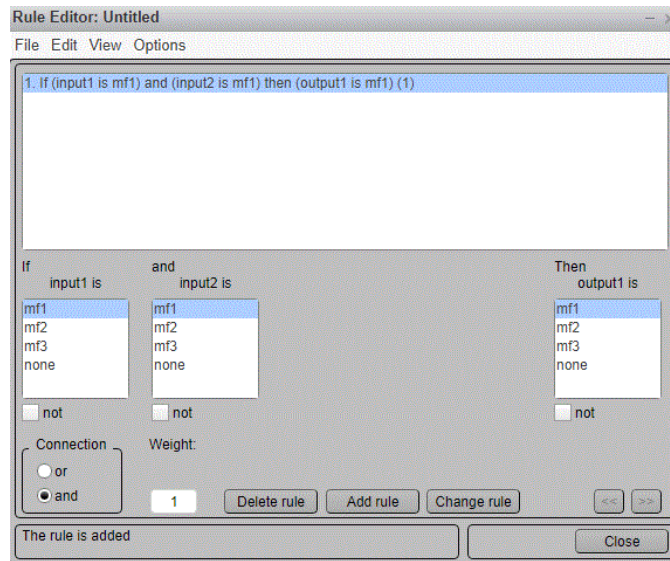


Figura 2.18 Rules Editor

Fuente: Propia

## Fuzzificación

En el proceso de fuzzificación se asignan variables lingüísticas a las variables de entrada y salida del control difuso, la información de las variables de entrada está expresadas en la lectura de los sensores de proximidad o ultrasónicos, se ha considerado realizar 2 controles: el Control 1 con 3 variables de entrada y 2 de salida (ángulo de dirección o giro del robot); y el Control 2 con 2 variables de entrada y 2 de salida (velocidad del robot)

En el Control 1 les corresponde: 3 entradas, el ángulo de obstáculo ( $\theta_1$ ), ángulo de error ( $\theta_2$ ) y el error de distancia del obstáculo (EDO), con 2 salidas de ángulo de dirección o giro ( $\theta_3$ ) de las ruedas izquierda (RI) y derecha (RD); mientras que para el Control 2 corresponde: 2 entradas error de distancia del obstáculo (EDO) y error de distancia de la posición deseada (EDPD), con 2 salidas de velocidad por intermedio de las ruedas izquierda (RI) y derecha (RD), ver Tabla 2.4.

Dónde:

$\theta_1$ : Ángulo error entre la posición del robot y la posición del obstáculo

$\theta_2$ : Ángulo error entre la posición del robot y la posición deseada

$\theta_3$ : Ángulo de dirección del robot

EDO: Error de distancia del obstáculo

EDPD: Error de distancia de la posición deseada

ENTRADAS				SALIDAS			
Ángulo del obstáculo	Ángulo error	Error de distancia del obstáculo	Error de distancia de la posición deseada	Ángulo de dirección		Velocidad	
$\theta_1$	$\theta_2$	EDO	EDPD	$\theta_3$			
				RI	RD	RI	RD

Tabla 2.4 Variables de entrada y salida

Fuente: Propia

Definidas las variables de entrada y salida, se definen los conjuntos y los valores lingüísticos, que se emplearán dentro de cada variable lingüística, ver Tabla 2.5.

		CONTROL 1				CONTROL 2		
<b>ENTRADAS</b>	Ángulo del obstáculo	Ángulo error	Valores Lingüísticos	Error de distancia del obstáculo	Error de distancia del obstáculo	Error de distancia de la posición deseada	Valores Lingüísticos	
	$\theta_1$	$\theta_2$		EDO	EDO	EDPD		
	LI	LI	LEJOS IZQUIERDA	C	C	C	CERO	
	CI	CI	CERCA IZQUIERDA	MCE	MCE	MCE	MUY CERCA	
	C	C	CERO	CE	CE	CE	CERCA	
	CD	CD	CERCA DERECHA	LE	LE	LE	LEJOS	
	LD	LD	LEJOS DERECHA	MLE	MLE	MLE	MUY LEJOS	
<b>SALIDAS</b>	Ángulo de dirección		Valores Lingüísticos	Velocidad		Valores Lingüísticos		
	$\theta_3$			RI	RD			
	RI	RD		RI	RD			
	DE	DE	DETENER	DE	DE	DETENER		
	MLE	MLE	MUY LENTO	MLE	MLE	MUY LENTO		
	LE	LE	LENTO	LE	LE	LENTO		
	RA	RA	RÁPIDO	RA	RA	RÁPIDO		
	MRA	MRA	MUY RÁPIDO	MRA	MRA	MUY RÁPIDO		

Tabla 2.5 Valores lingüísticos

Fuente: Propia

## Valores lingüísticos de entrada del Control 1

La expresión “**lejos izquierda**” puede representar a las dos variables de entrada (teta1, teta2), de que la posición del obstáculo está lejos de la posición del robot, pero que está a la izquierda de la posición deseada.

La expresión “**cerca izquierda**” puede representar para las dos variables de entrada de que está cerca la posición del obstáculo y la posición del robot, pero no están alineadas para justificarle como cero.

La expresión “**cero**” puede representar que el ángulo de las dos variables de entrada es mínimo o que están alineadas con la posición del robot, por lo que sería necesario que el robot se dirija a la izquierda o a la derecha del obstáculo para que lo evada.

La expresión “**cerca derecha**” puede representar para las dos variables de entrada que está cerca la posición del obstáculo y la posición del robot, pero que está a la derecha de la posición deseada.

La expresión “**lejos derecha**” puede representar para las dos variables de entrada que el movimiento del robot está hacia la izquierda respecto de la posición deseada.

Los valores lingüísticos de la tercera entrada del control 1, correspondiente al error de distancia entre la posición del robot y la posición deseada, están expresados en las entradas del control 2.

## Valores lingüísticos de entrada del Control 2

La expresión “**cero**” puede representar que la distancia entre el robot y la posición deseada sea el mínimo o que esté dentro de la distancia que le permitirá detenerse.

La expresión “**muy cerca**” puede representar que la distancia entre la posición del robot y la posición deseada sea mayor a cero pero que ya estaría acercándose para detenerse.

La expresión “**cerca**” puede representar que la distancia entre la posición del robot y la posición deseada sea mayor a muy cerca por lo que el robot continúa moviéndose, hasta detenerse.

La expresión “**lejos**” puede representar a la distancia entre la posición del robot y la posición deseada está alejado, por lo que se mantendría moviéndose.

La expresión “**muy lejos**” puede representar que la distancia entre la posición del robot y la posición deseada está en el perímetro de la lectura de los sensores de proximidad o aún más lejos.

### **Valores lingüísticos de salida Controles 1 y 2**

Las expresiones de los 2 controles son muy similares ya que están gobernados por los motores de paso a paso de cada rueda, siendo estas: rueda izquierda (RI) y rueda derecha (RD), se denotan de la siguiente manera:

La expresión “**detener**” puede representar a la velocidad angular cuyo valor sea cero, es decir que la rueda no gira.

La expresión “**muy lento**” puede representar a la velocidad angular cuyo valor sea mínimo pues podría deberse a que está cerca de detenerse o que estaría evadiendo algún obstáculo.

La expresión “**lento**” puede representar a la velocidad angular cuyo valor sea medio, pero que puede girar sin dificultad dependiendo de la distancia entre la posición del robot y la posición deseada.

La expresión “**rápido**” puede representar a la velocidad angular cuyo valor sea medio alto respecto del valor muy lento o detenerse.

La expresión “**muy rápido**” puede representar a la velocidad angular cuyo valor sea alto, y permita generar mayores rpm para alcanzar la posición deseada.

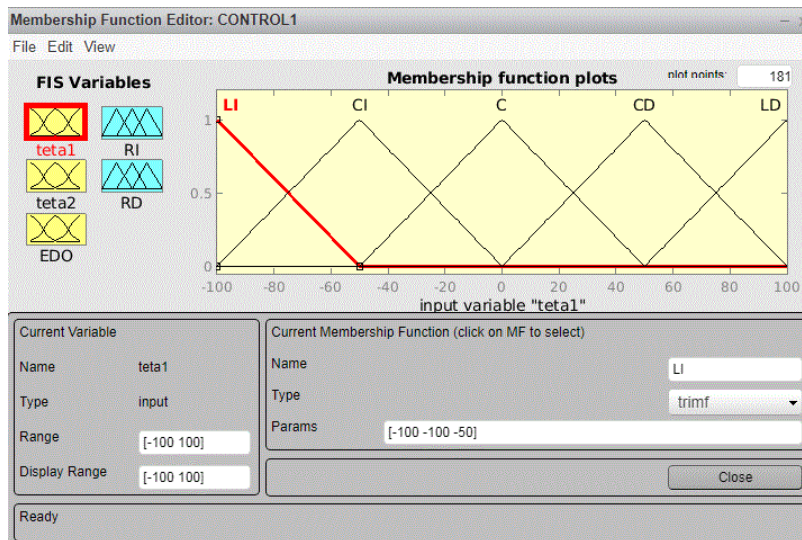


## Funciones de membresía

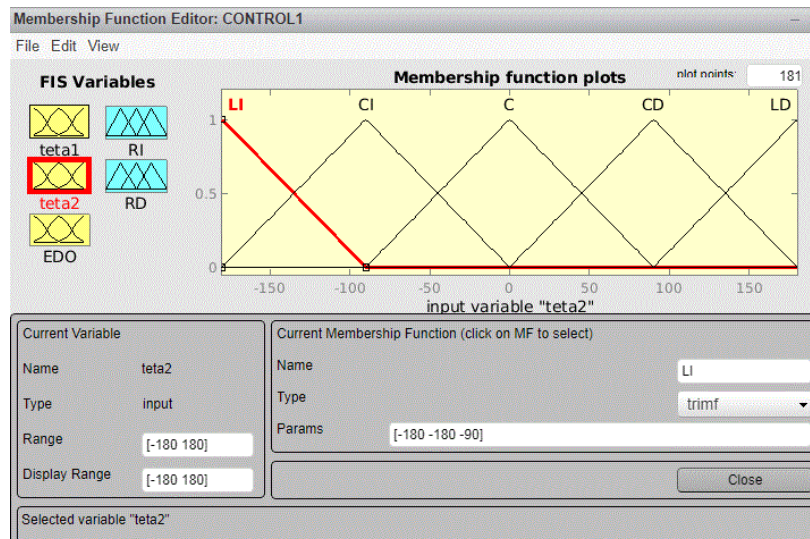
### Control 1 Variables de entrada

Para el desarrollo del control 1 se considera 3 entradas y 2 salidas: El rango de cada variable es el valor numérico entre el mínimo y el máximo, de la siguiente manera:

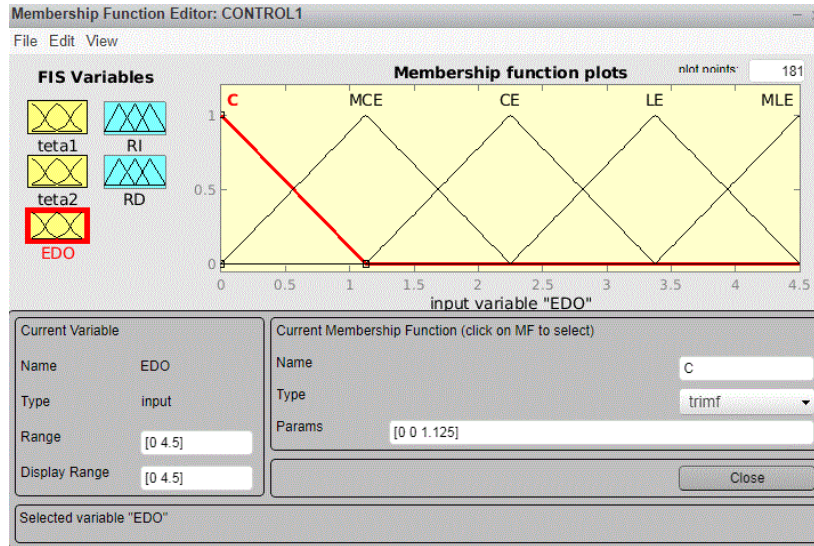
1.  $\theta_1$  **Rango:**  $-100^\circ \leq \theta < +100^\circ$ , ver Figura 2.19 (a).
2.  $\theta_2$  **Rango:**  $-180^\circ \leq \theta < +180^\circ$ , ver Figura 2.19 (b).
3. EDO **Rango:**  $0 \leq \theta < +4,5 (m)$ , ver Figura 2.19 (c).



(a)



(b)



(c)

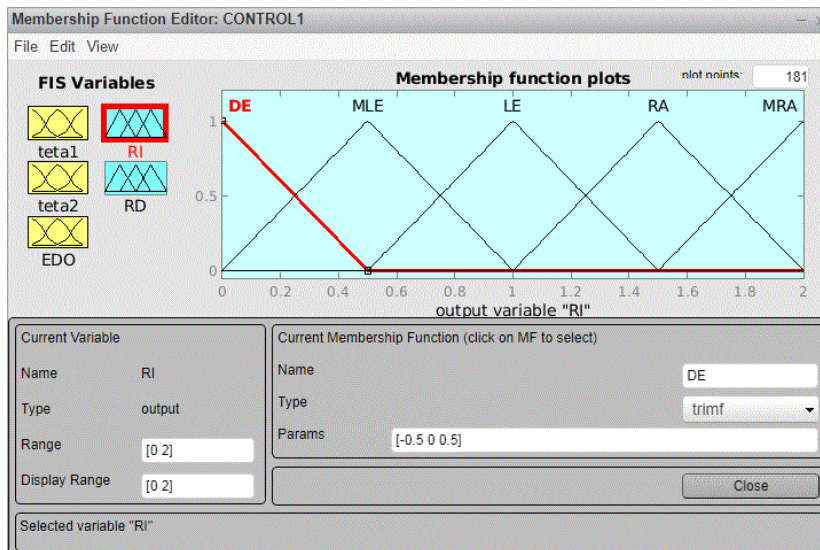
Figura 2.19 Variables de entrada Control 1 (a)  $\theta_1$ ; (b)  $\theta_2$  y (c) EDO

Fuente: Propia

## Control 1 Variables de salida

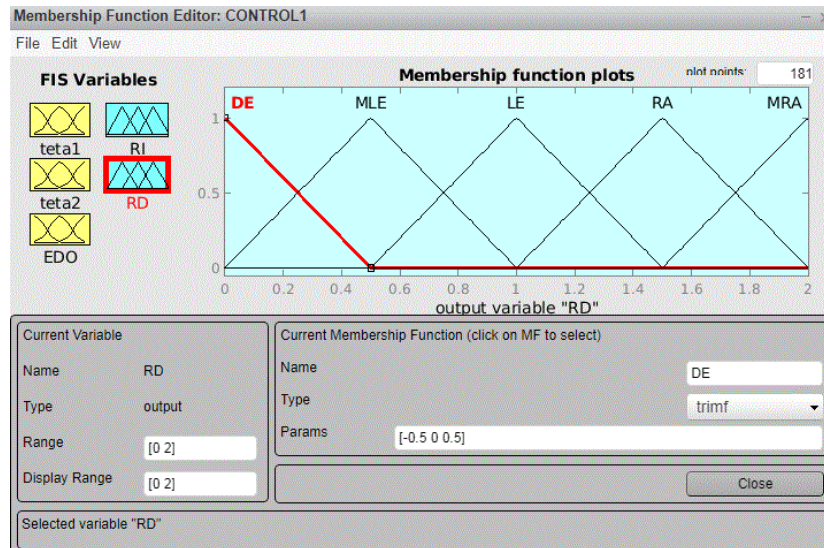
Angulo de dirección (teta 3)      **Rango:**  $0 \leq \theta < +2,0$  (m/s)

1. Rueda izquierda (RI), ver Figura 2.20 (a).
2. Rueda derecha (RD), ver Figura 2.20 (b).



(a)





(b)

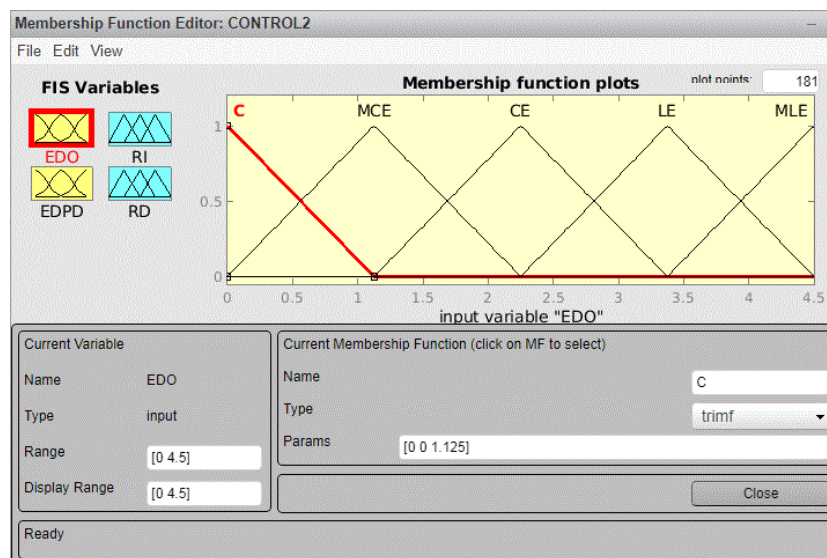
Figura 2.20 Variables de salida Control 1 (a) RI y (b) RD

Fuente: Propia

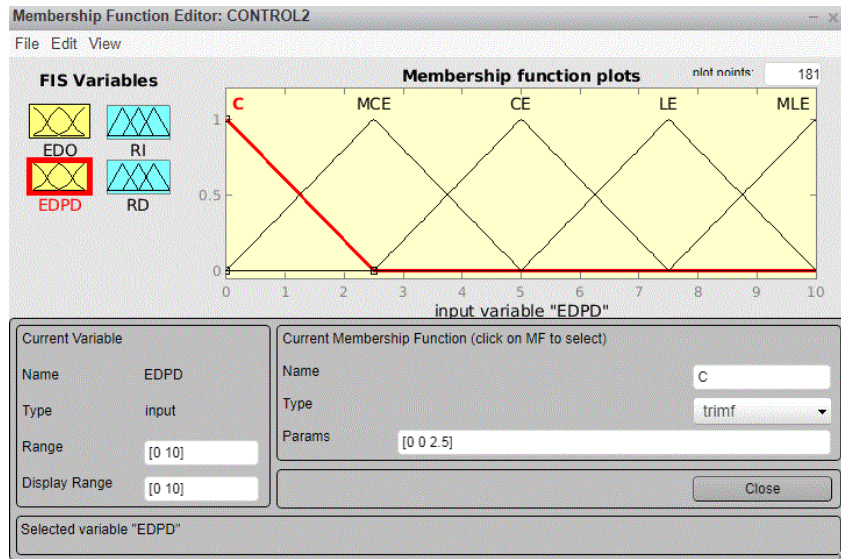
## Control 2 Variables de entrada

Mientras que para el desarrollo del control 2 con 2 entradas y 2 salidas: El rango de cada variable es el valor numérico entre el mínimo y el máximo, de la siguiente manera:

1. EDO **Rango:**  $0 \leq \theta < +4,5$  (m), ver Figura 2.21 (a).
2. EDPD **Rango:**  $0 \leq \theta < +10$  (m), ver Figura 2.21 (b).



(a)



(b)

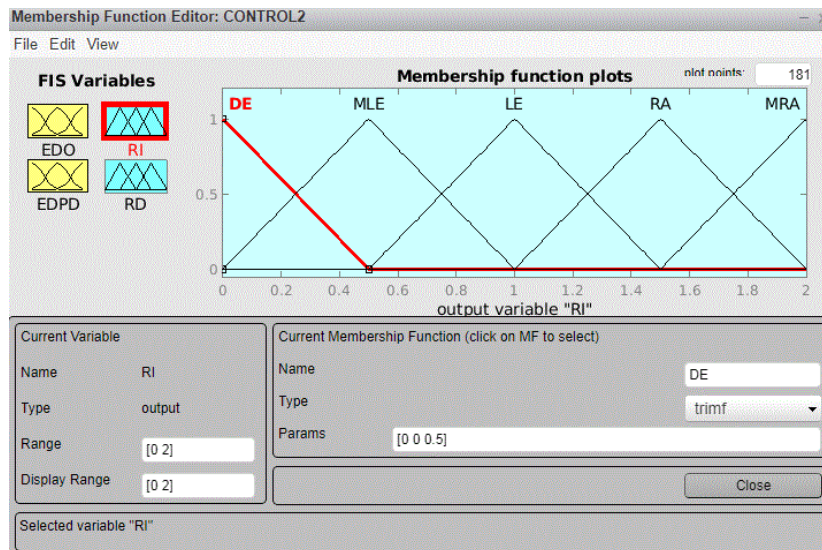
Figura 2.21 Variables de entrada Control 2 (a) EDO y (b) EDPD

Fuente: Propia

## Control 2 Variables de salida

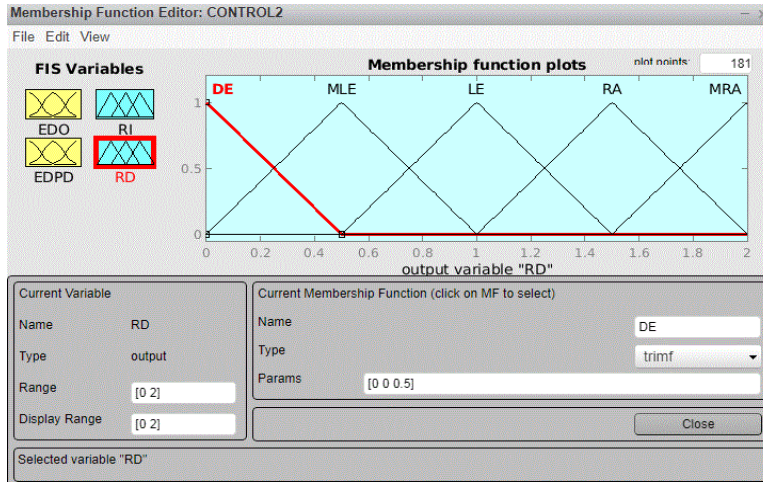
*Velocidad*      **Rango:**  $0 \leq \theta < +2,0$  (m/s)

1. Rueda izquierda (RI), ver Figura 2.22 (a).
2. Rueda derecha (RD), ver Figura 2.22 (b).



(a)





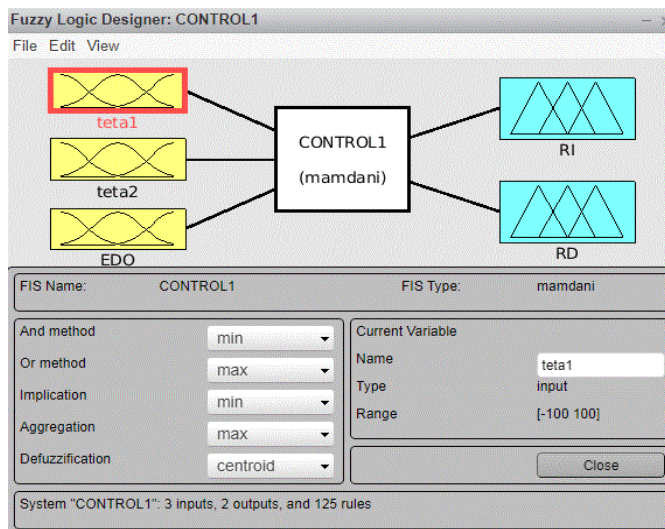
(b)

Figura 2.22 Variables de salida Control 2 (a) RI y (b) RD

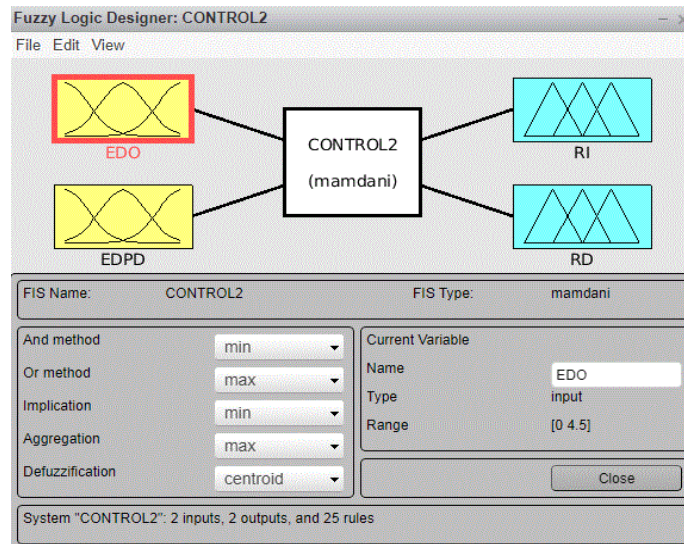
Fuente: Propia

## Reglas de membresía

Definidas las funciones de membresía de las variables de entrada y salida, se procede a la elaboración de las reglas de membresía del controlador, en base a las reglas heurísticas IF (antecedente) y THEN (consecuente). En esta etapa el controlador toma las decisiones o acciones del sistema, aplicamos la operación AND. Las decisiones del controlador están de acuerdo a las decisiones de las velocidades angulares de las ruedas izquierda y derecha, para determinar el ángulo de error de la dirección o giro y la velocidad lineal del robot, de cada control, ver Figura 2.23.



(a)



(b)

Figura 2.23 Inferencia Mamdani (a) Control 1 y (b) Control 2

Fuente: Propia

### Reglas de membresía: Control 1

El número de reglas de membresía generadas del control 1 son 125, que resultan de combinar 3 variables de entrada y 2 variables de salida, con 5 valores lingüísticos cada uno, para controlar el ángulo de la dirección del robot entre la posición del robot y la posición deseada, de las cuales podríamos mencionar las siguientes:

**SI** teta 1 es LI, teta 2 es LI y EDO es C, **ENTONCES** RI es DE y RD es DE

Esta regla de membresía cuantifica que, si el ángulo teta 1 está lejos izquierda, el ángulo teta 2 lejos izquierda, y EDO es cero, los ángulos de la posición del robot respecto de la posición del obstáculo la posición del robot está lejos de la posición deseada pero cerca de la posición del obstáculo, nos indica que la velocidad angular de ambas ruedas deben ser cero es decir el robot debe detenerse para evitar chocar con el obstáculo, ver Figura 2.24.

**SI** teta 1 es CI, teta 2 es LD y EDO es MCE, **ENTONCES** RI es MLE y RD es RA

Esta regla de membresía cuantifica que si el ángulo teta 1 es cerca izquierda, el ángulo teta 2 es lejos derecha, y EDO es muy cerca, la posición del robot está cerca de la posición del obstáculo por lo que se debe evitar chocarse, la posición del robot debe girar

hacia la izquierda aumentando la velocidad angular en la rueda derecha y baja velocidad angular en la rueda izquierda, luego de haber evitado el obstáculo debe procederse al aumento en la velocidad angular rueda izquierda y disminución en la velocidad angular rueda derecha a fin de alinearse con la posición deseada, ver Figura 2.25.

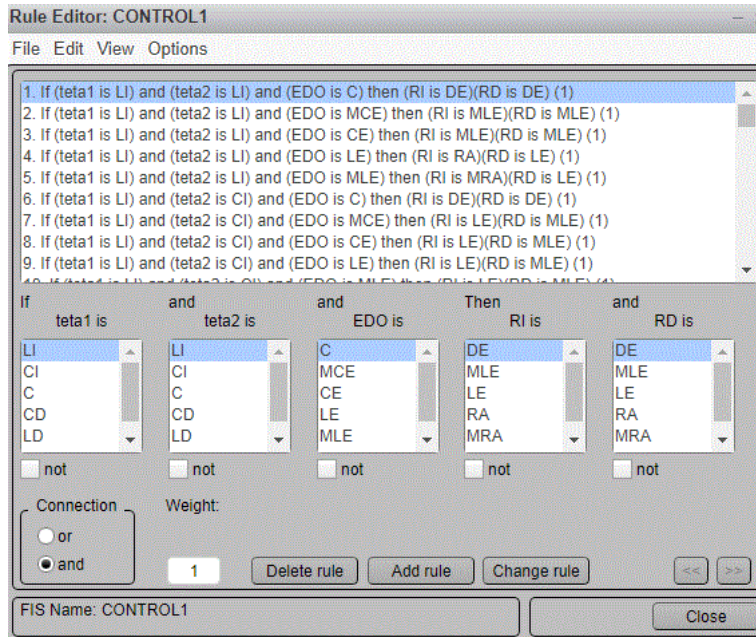


Figura 2.24 Regla de membresía 1 del Control 1

Fuente: Propia

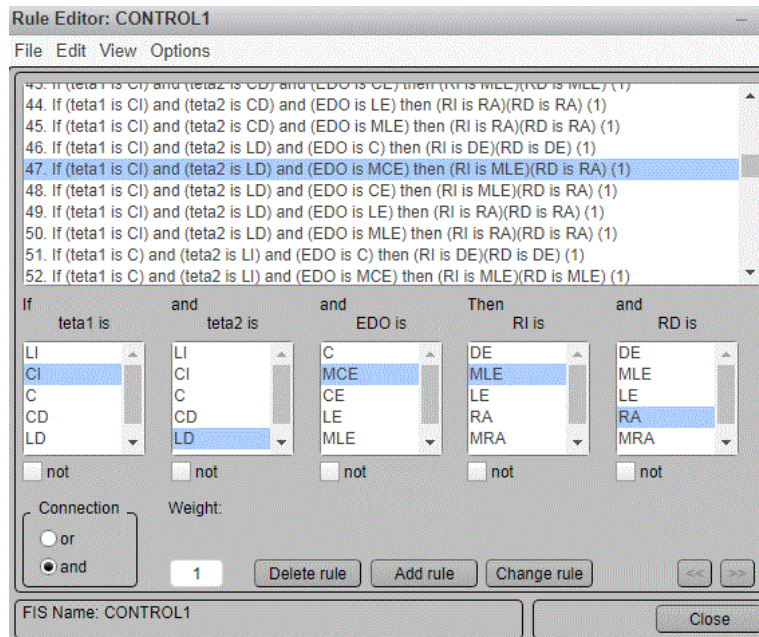


Figura 2.25 Regla de membresía 47 del Control 1

Fuente: Propia



**SI** teta 1 es LD, teta 2 es LD y EDO es CE, **ENTONCES** RI es MLE y RD es MLE

Esta regla de membresía cuantifica que, si el ángulo teta 1 es lejos derecha, el ángulo teta 2 es lejos derecha, y EDO es cerca, entonces la velocidad angular en ambas ruedas es muy lenta para evitar chocar con el obstáculo y luego alinearse con la posición deseada, ver Figura 2.26.

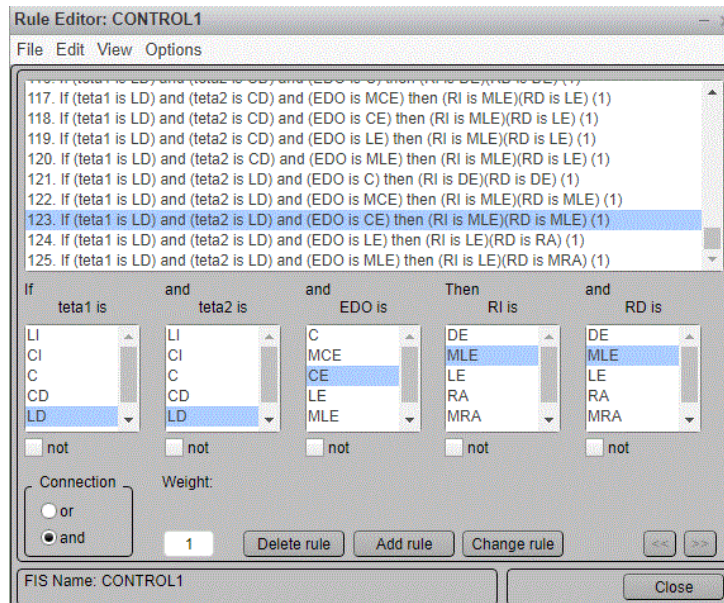


Figura 2.26 Regla de membresía 123 del Control 1

Fuente: Propia

## Reglas de membresía: Control 2

Las reglas de membresía generadas del control 2 son 25, que resultan de combinar 2 variables de entrada y 2 variables de salida, con 5 valores lingüísticos cada uno, para controlar la velocidad lineal de las ruedas entre la posición del robot y la posición deseada, de las cuales podremos mencionar las siguientes:

**SI** EDO es MCE y EDPD es MCE, **ENTONCES** RI es MLE y RD es MLE

Esta regla de membresía cuantifica que si la distancia entre la posición del robot y la posición del obstáculo está muy cerca, y entre la posición del robot con la posición deseada es muy cerca, la velocidad angular para ambas ruedas debe ser muy lenta ya que se acerca a la posición deseada por lo que luego tendrán que detenerse, ver Figura 2.27.

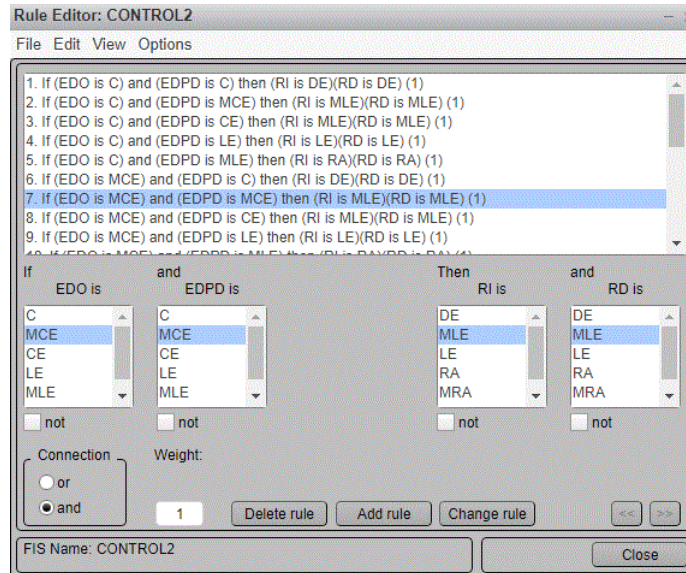


Figura 2.27 Regla de membresía 7 del Control 2

Fuente: Propia

**SI** EDO es CE y EDPD es MLE, **ENTONCES** RI es RA y RD es RA

Esta regla de membresía cuantifica que, si la distancia entre la posición del robot y la posición del obstáculo está cerca, la posición del robot con la posición deseada está muy lejos, indica que debe aumentarse la velocidad angular para ambas ruedas a fin de acercarse rápido a la posición deseada, ver Figura 2.28.

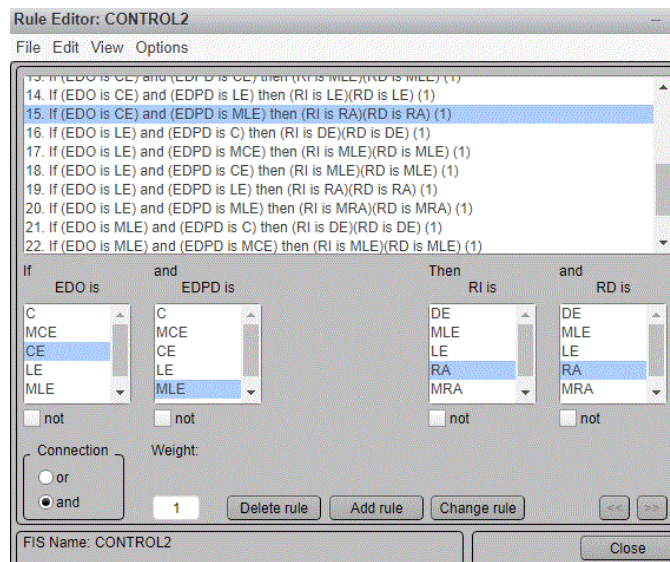


Figura 2.28 Regla de membresía 15 del Control 2

Fuente: Propia

## 2.3.2 Desarrollo de la interfaz hombre máquina para control Fuzzy

La interfaz hombre máquina conocida como HMI es un panel de control que conecta a un usuario con una máquina, es el entorno que permite comunicarse con los sensores de entrada y salida. En el desarrollo de este proyecto se ingresa en la ventana de comandos de Matlab la palabra `guide` y ejecutamos, nos aparece la ventana para seleccionar plantillas, escogemos una ventana de desarrollo Blank GUI (Default) ver Figura 2.29.

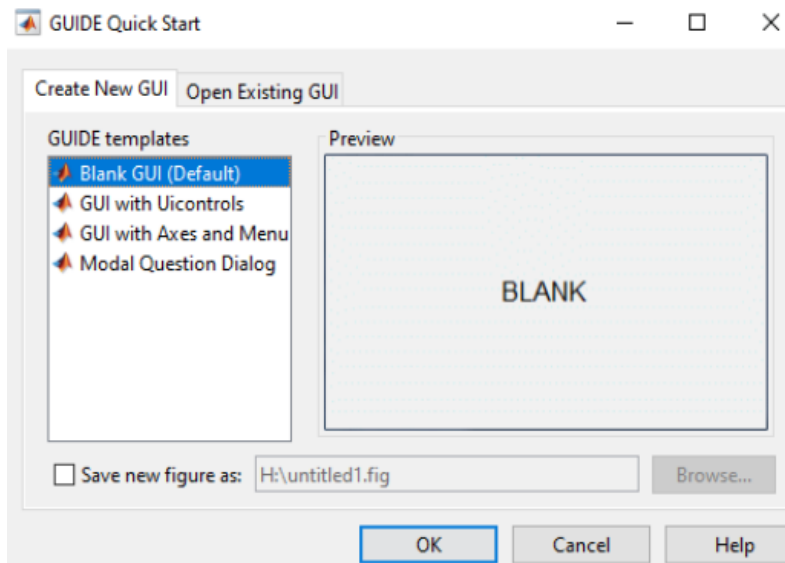


Figura 2.29 Pantalla de diseño GUI

Fuente: Propia

Como se muestra en la Figura 2.30, hay varios botones de los elementos del área de diseño del GUI, que se arrastran y se sueltan, entre los que podremos definir los siguientes:

- Check box: Indica el estado de una opción
- Edit Text: Edición de texto
- Pop-up Menu: Lista de opciones
- Push Button: Acción al pulsar
- Radio Button: Selecciona una opción
- Sliders: Representa un rango de valores



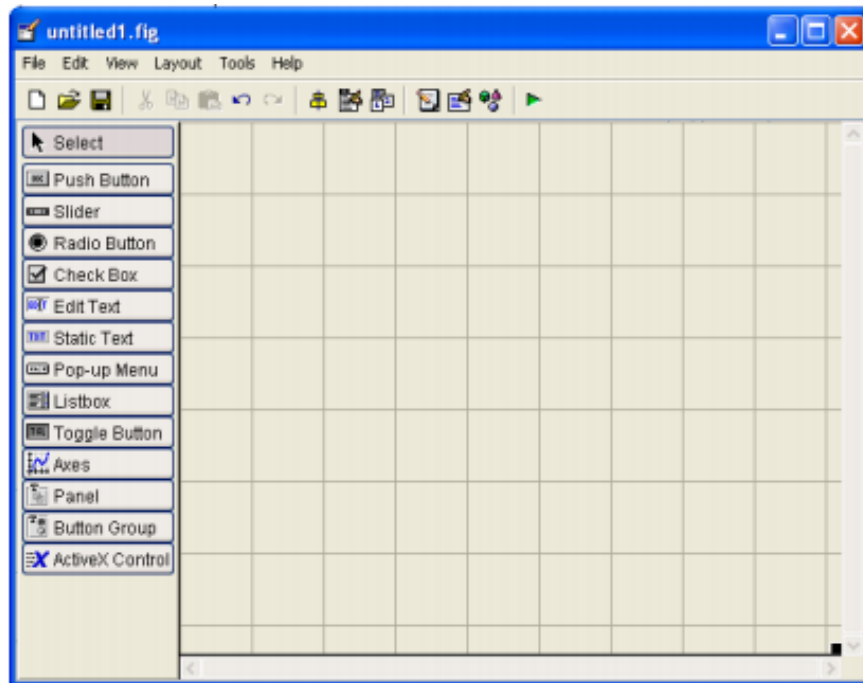


Figura 2.30 Pantalla de diseño GUI

Fuente: Propia

Al pulsar en el elemento seleccionado procedemos a personalizar en base al Property Inspector, de igual manera al hacer clic en la pantalla de diseño se abre View Callbacks, que al ejecutarse se abre el archivo .m asociado a nuestro sistema. La operación automática de guardado del proyecto genera dos ficheros tipo .m que contiene el código que controla el GUI y tipo .fig que contiene los elementos gráficos.

### **2.3.3 Simulación computacional de la respuesta temporal del control difuso**

Para observar cómo trabaja el sistema, tecleamos dentro de la función Edit de las reglas de membresía la función View, en la que se desplegará las reglas (Rules) y superficies (Surface) de los controladores, ver Figura 2.31.

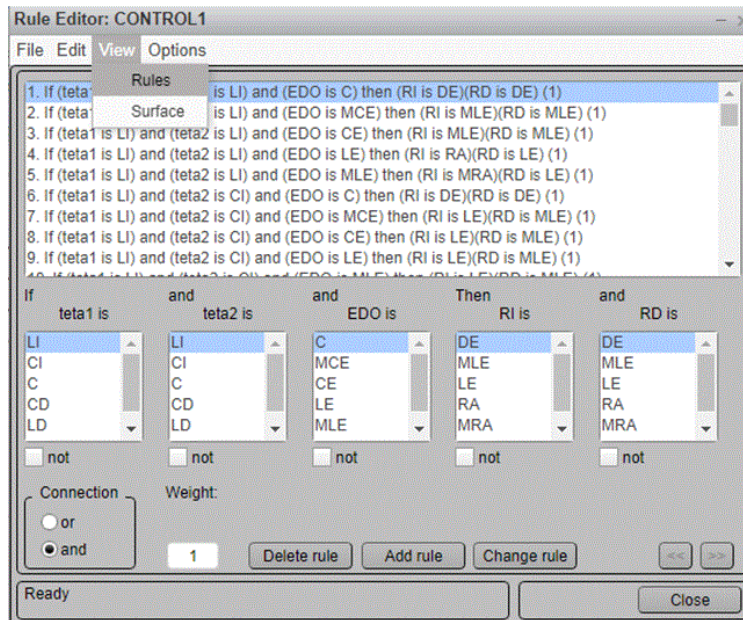


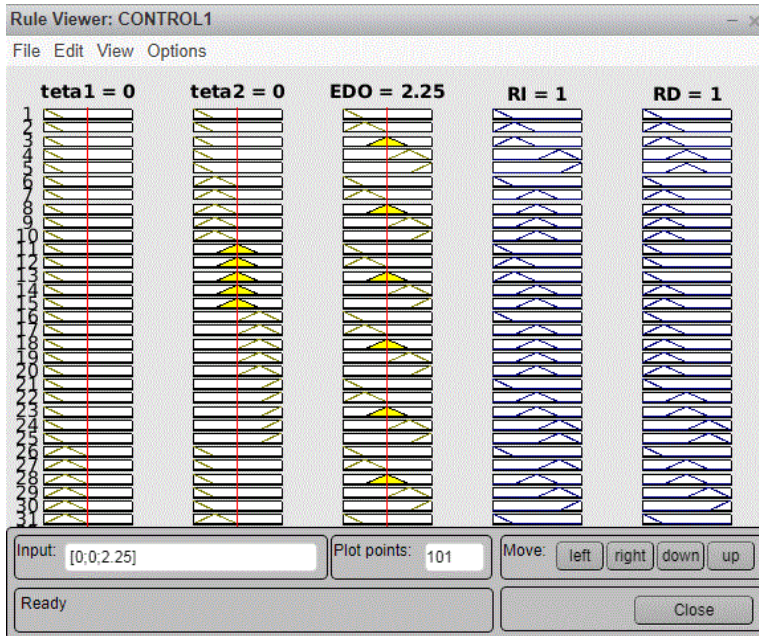
Figura 2.31 View Control 1

Fuente: Propia

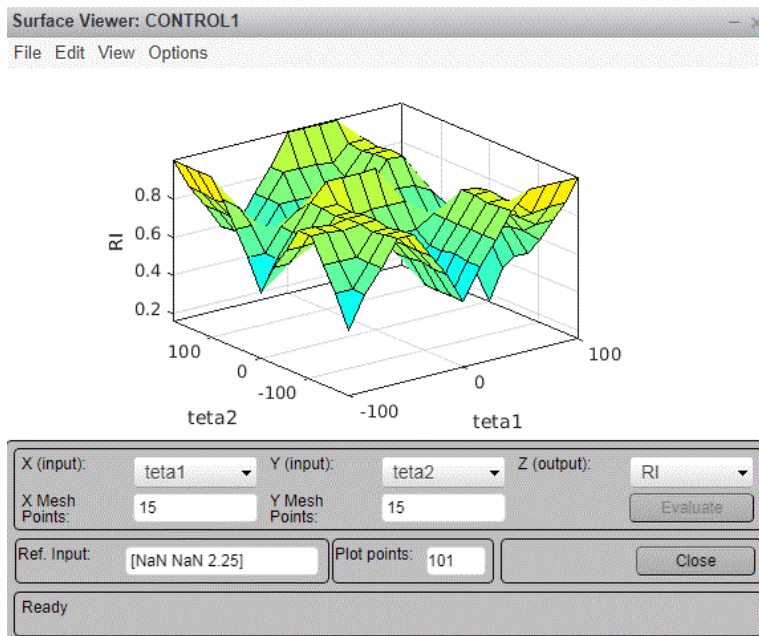
Al ingresar en View y la opción Rules se nos desplegará una pantalla donde para el caso del control 1 las 3 primeras columnas de la izquierda representan las entradas (Input) y las 2 columnas de la derecha representan las salidas (Output) del sistema; en el control 2 las 2 primeras columnas de la izquierda representan las entradas (Input) y las 2 columnas de la derecha representan las salidas (Output) del sistema, ver Figura 2.32 (a).

Las variables de entrada coloreadas de amarillo que intervienen en el proceso de inferencia y las variables de salida coloreadas de azul nos indican que reglas de membresía se han activado y en qué medida, procedemos a evaluar el sistema de cada uno de los controles.

Según la figura 2.32 (b), al ingresar en View y la opción Surface se nos despliega una pantalla con una gráfica representada en tres dimensiones con la visualización de 2 variables de entrada y 1 variable de salida.



(a)



(b)

Figura 2.32 View (a) Rules Viewer y (b) Surface Viewer

Fuente: Propia

## Reglas de control (Rules Viewer)

### Control 1

En la parte superior de la pantalla de las reglas de membresía se observan los valores a fuzzificar, ver Figura 2.33 los valores de las variables de entradas del sistema son: el ángulo teta1 cuyo valor es -83.6, el ángulo teta2 cuyo valor es 48.4 y el EDO es 3.06, la inferencia del sistema indica que la velocidad en la rueda izquierda RI el valor es 1.02 y la velocidad en la rueda derecha RD el valor es 0.891.

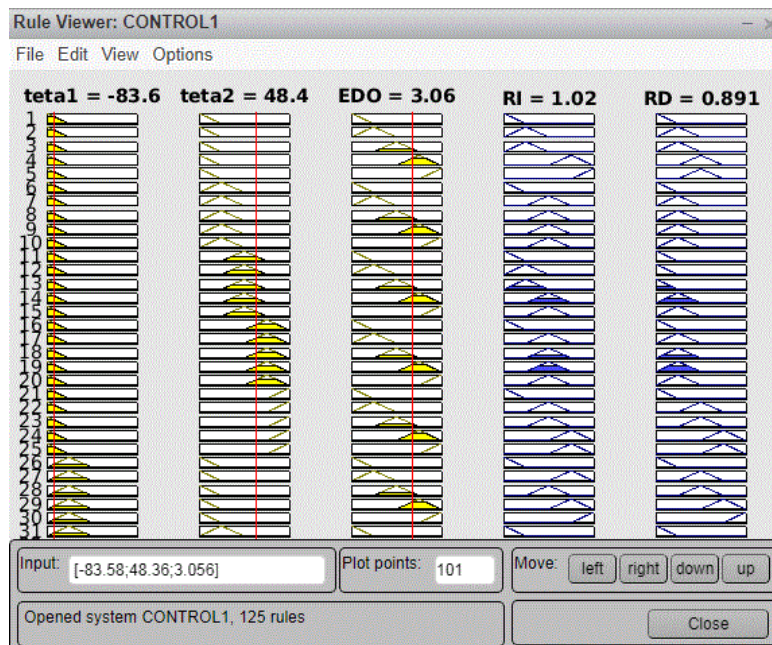


Figura 2.33 Rule Viewer Control 1

Fuente: Propia

### Control 2

En la parte superior de la pantalla de las reglas de membresía se observan los valores a fuzzificar, según la Figura 2.34 los valores de las variables de entradas del sistema son: EDO= 2.73 y el EDPD= 5, la inferencia del sistema indica que la velocidad en ambas ruedas es de 0.5.



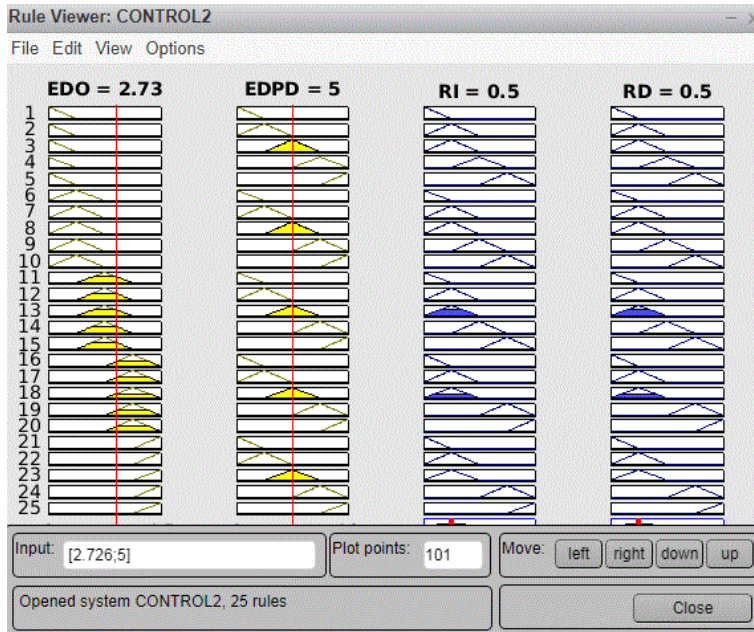


Figura 2.34 Rule Viewer Control 2

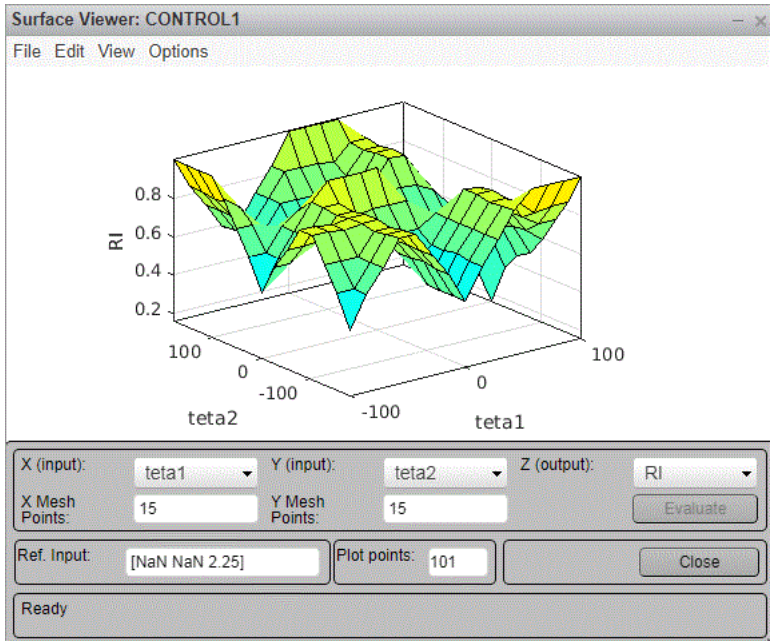
Fuente: Propia

## Superficies de control (Surface Viewer)

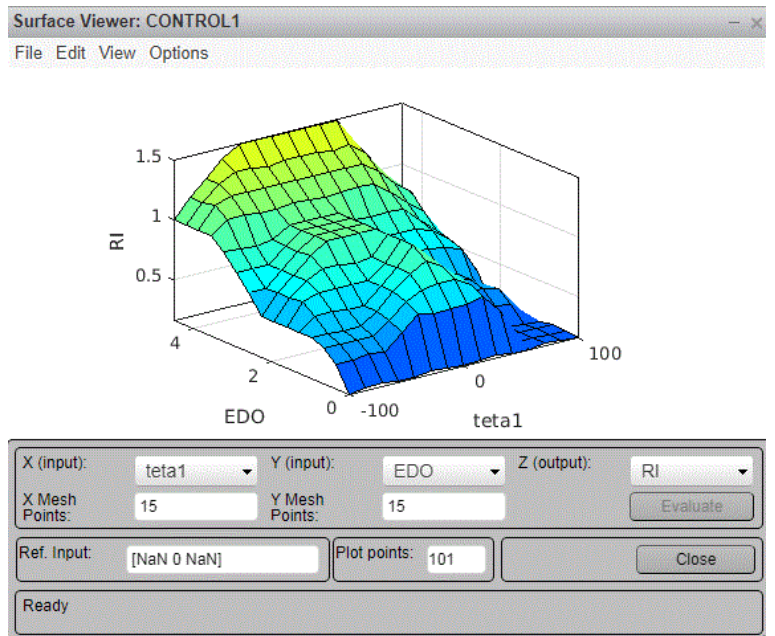
### Control 1

Las superficies de control 1 se desplegará 6 gráficas que resultan de combinar 2 variables de las 3 variables de entrada, y de 2 variables de salida, que genera 3 gráficas de salida de la rueda izquierda RI (a) Input ( $\theta_1$  y  $\theta_2$ ) (b) Input ( $\theta_1$  y EDO) (c) Input ( $\theta_2$  y EDO) ver figura N 2. 35 y 3 gráficas de salida similares de la rueda derecha RD, ver Figura 2. 36.

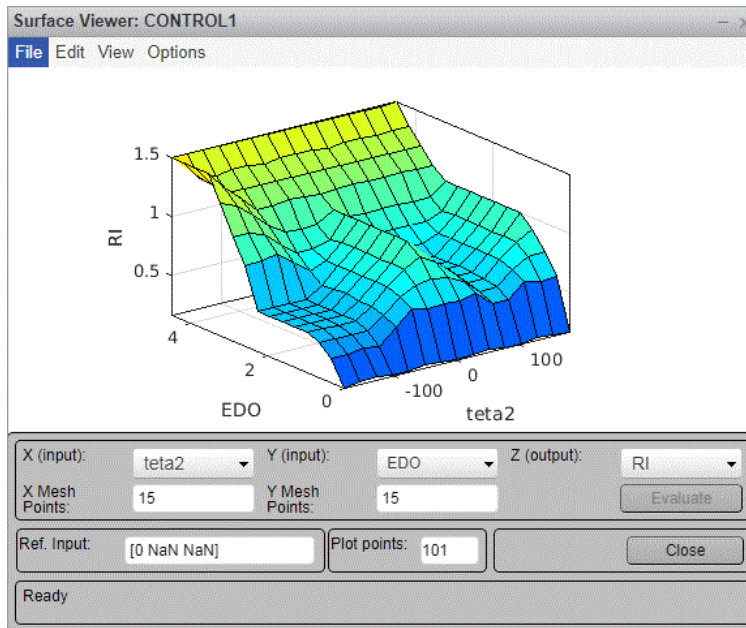
Las gráficas de la Figura 2.35, que generaron 3 gráficas en las cuales, la gráfica (a)  $\theta_1$  (-100°, +100°);  $\theta_2$  (-180°, +180°), una altura RI (0,1); la gráfica (b)  $\theta_1$  (-100°, +100°); EDO (0,4.5), una altura RI(0,1.5) y la gráfica (c)  $\theta_2$  (-180°, +180°); EDO (0,4.5), una altura RI (0,1.5). En el sistema prevalece el giro de dirección del robot hacia la derecha.



(a)



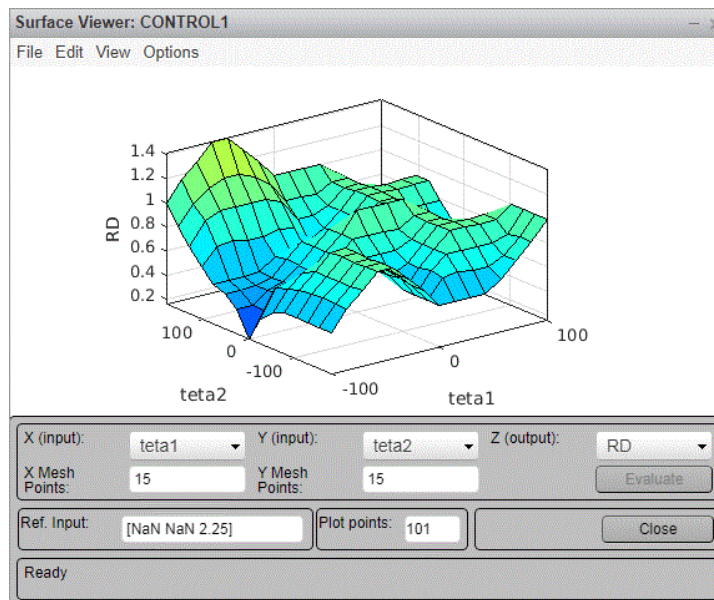
(b)



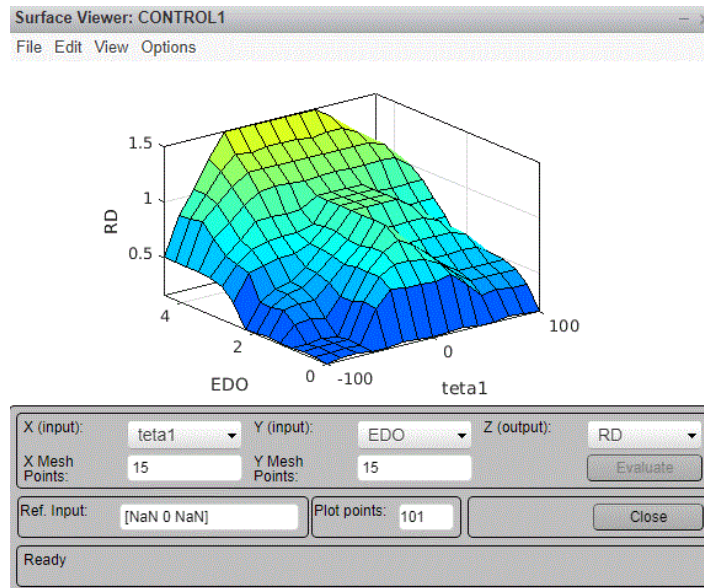
(c)

Figura 2.35 Superficie de control 1 RI (a)  $\theta_1$  y  $\theta_2$  (b)  $\theta_1$  y EDO (c)  $\theta_2$  y EDO

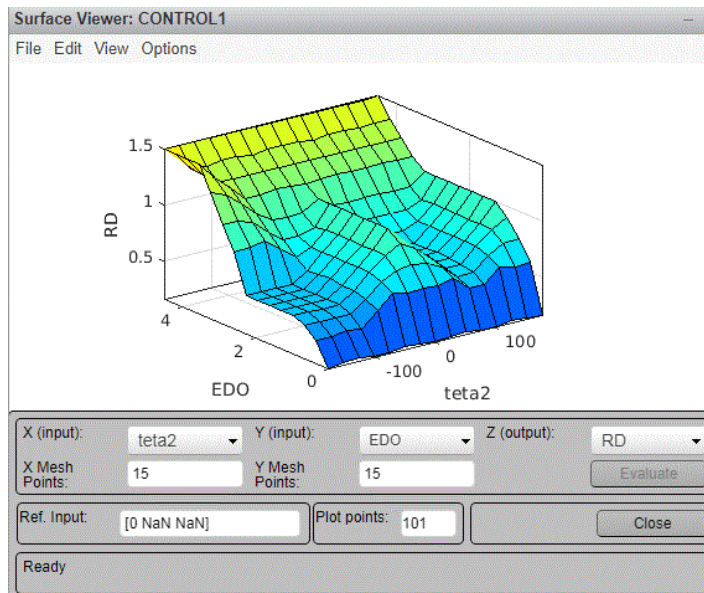
Fuente: Propia



(a)



(b)



(c)

Figura 2.36 Superficie de control 1 RI (a)  $\theta_1$  y  $\theta_2$  (b)  $\theta_1$  y EDO (c)  $\theta_2$  y EDO

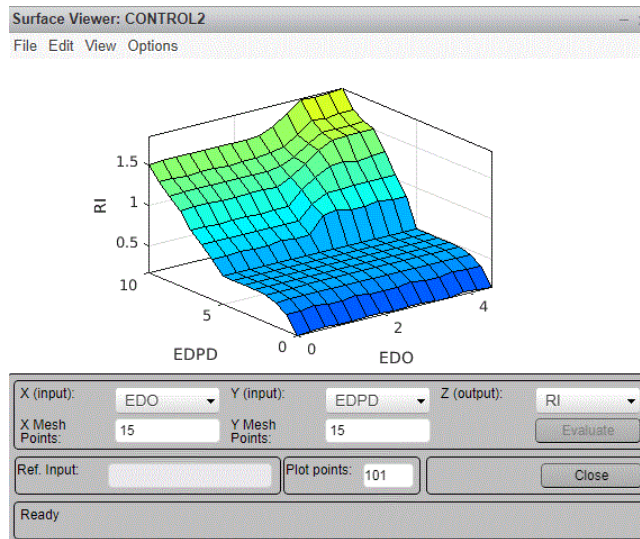
Fuente: Propia

Según las gráficas de la Figura 2.36, la gráfica (a)  $\theta_1(-100^\circ,+100^\circ)$ ;  $\theta_2 (-180^\circ,+180^\circ)$ , una altura RD (0,1.4); en las gráficas (b) y (c) son similares a las expresadas en la Figura 2.35. El sistema permite al control evitar obstáculos, prevaleciendo el giro de dirección del robot hacia la izquierda.

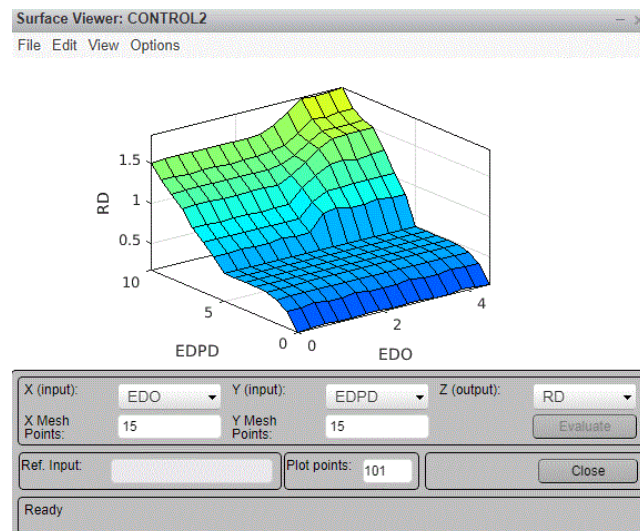


## Control 2

Al ingresar a la pantalla de las superficies de control (Surface Viewer) del control 2 se desplegará 2 gráficas que resultan de combinar 2 variables de entrada y 2 variables de salida, es decir nos genera 1 gráfica para la rueda izquierda RI (a), y 1 gráfica para la rueda derecha RD (b), ver Figura 2. 37 con las mismas entradas Input (EDO y EDPD). Las superficies de control 2 de ambas ruedas, generaron 2 gráficas con las mismas variables de entrada EDO (0,4.5); EDPD (0,10) y una altura para ambas ruedas de (0,2).



(a)



(b)

Figura 2.37 Superficie de control 2 (a) RI y (b) RD

Fuente: Propia

## 2.4 Validación virtual

En esta sección se mostrará la implementación de la interfaz hombre máquina de este proyecto, a través de la programación en Matlab con el archivo .m y la representación gráfica con el archivo .fig, y su validación a través de los datos ingresados en el control fuzzy.

### 2.4.1 Programación del GUI de Matlab

El programa de MATLAB tiene entre sus prestaciones básicas: manipulación de matrices, representación de datos y funciones, implementación de algoritmos, creación de interfaces de usuario GUI. Permite crear interfaces mediante el editor de diseño de GUIDE, en el que se diseña gráficamente y genera de manera automática el código de MATLAB, ver ANEXO 7 con el código Tesis.m del Control 1 y ANEXO 8 con el código Tesis2.m del Control 2.

Previo a la programación del GUI es necesario que las variables de entrada de cada uno de los controles corresponden al ingreso en forma de vectores, y que tipo de presentación de datos de salida se necesitan en este caso serán tipo gráfico. Los valores de las propiedades de los elementos (color, valor, posición, string, static text, etc.) y los valores de las variables de entrada se almacenan en una estructura dentro del programa .m, mediante un identificador handles, que es el identificador de datos del proyecto, el mismo que es salvado con la instrucción guidata, como se muestra de la siguiente manera:

```
handles(hObject)= hObject;  
guidata ( hObject, handles);
```

guidata, es la sentencia para salvar los datos del proyecto, además es la función que guarda las variables y propiedades de los elementos de la estructura de datos del proyecto. Esta sentencia nos garantiza que cualquier cambio de propiedades quede almacenado.

La primera línea crea la variable de entrada en la estructura de datos del proyecto mediante handles y la segunda graba el valor mediante guidata:

```
handles.teta1=teta1;  
guidata(hObject, handles);
```

Para seleccionar las propiedades de cada elemento seleccionamos la opción Properties Inspector abriéndose un submenú con todas las propiedades que podamos editar como el color, valor, posición, string, static text, fuente, etc.

Uno de los elementos es el Edit Callback, que nos posiciona en la sección del programa que corresponde a la subrutina que se ejecutará cuando se realice una acción sobre algún elemento que estemos editando:

```
function Tesis_OpeningFcn(hObject, eventdata, handles, varargin)  
function varargout = Tesis_OutputFcn(hObject, eventdata, handles)  
function teta1_Callback(hObject, eventdata, handles)
```

En el desarrollo de este proyecto debemos utilizar las funciones FIS de ambos controles con el ingreso de las subrutinas:

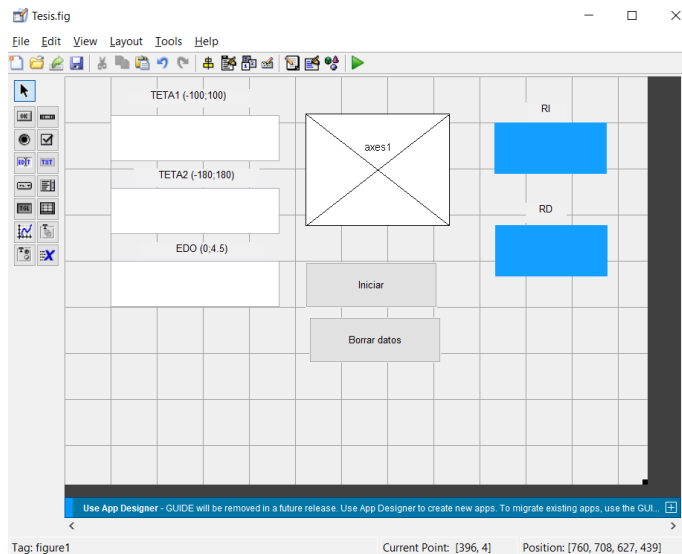
```
fis = readfis('C:\Users\Kevin\Documents\MATLAB\CONTROL1.fis');  
fis = readfis('C:\Users\Kevin\Documents\MATLAB\CONTROL2.fis');
```

Se debe limitar el ingreso de valores en las variables de entradas de la interfaz de usuario, mediante la sentencia if – end, si ingresamos valores dentro de los rangos establecidos se generan valores de salida de acuerdo con el control difuso propuesto, en el caso de que se digite valores fuera del rango sugerido en la pantalla se genera un error, por lo que genera una pantalla de error.

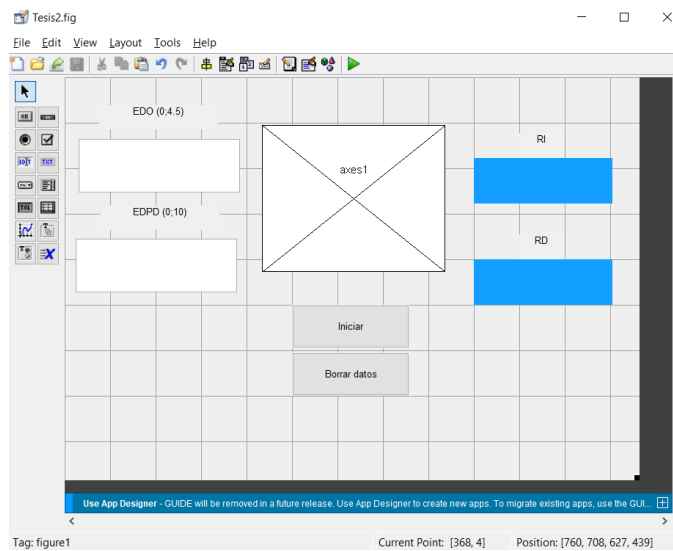
```
if teta1<-100 || teta1>100 || isnan(teta1)  
    errordlg('Teta1 fuera de rango','error');  
    return;  
end
```

## 2.4.2 Desarrollo de la interfaz hombre máquina para el robot móvil

Es la representación gráfica de los controladores, mediante el archivo .fig, es decir es la apariencia final de nuestra interfaz gráfica GUI, abrimos guide en GUI y seleccionamos BLACK GUIDE (DEFAULT), ingresamos pushbuttons, static text, axes, pop-up menú, cualquier cambio realizamos a través de Properties Inspector o con doble click en uno de los elementos, ver Figuras 2.38 (a) Control 1 y (b) Control 2.



(a)



(b)

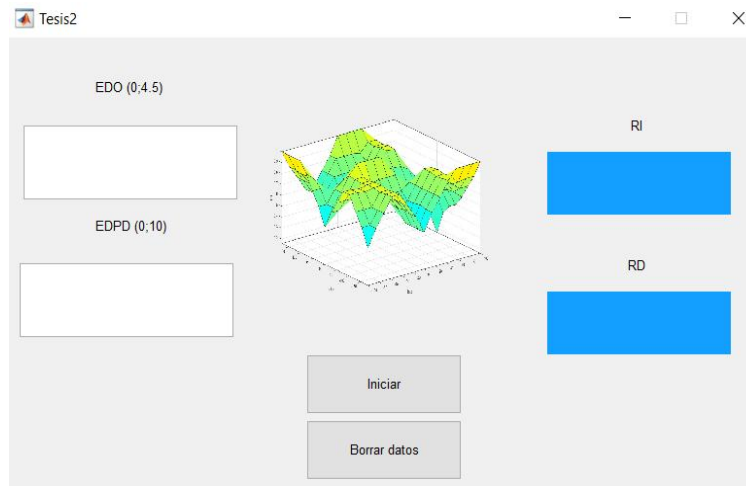
Figura 2.38 GUI (a) Control 1 y (b) Control 2

Fuente: Propia

Cuando está completo el proyecto ejecutamos en el ícono Run Figure, nos generará los gráficos GUI de cada control con las variables de entrada y de salida previo al ingreso de valores, ver Figura 2.39.



(a)



(b)

Figura 2.39 GUI (a) Control 1 y (b) Control 2

Fuente: Propia

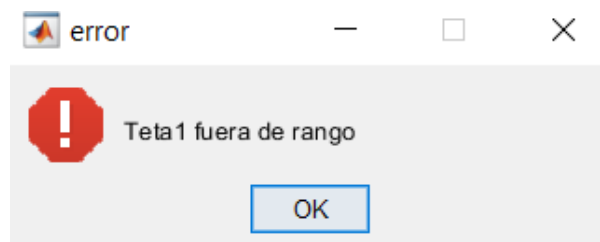


Figura 2.40 Error fuera de rango

Fuente: Propia

Se debe limitar el ingreso de valores en los pushbuttons de entrada mediante la sentencia de if – end en el código de MATLAB, si se ingresa valores fuera del rango permitido debe generar una alerta gráfica de error, ver Figura 2.40, por lo que se debe borrar los datos e ingresar nuevamente de acuerdo con cada rango establecido.

### 2.4.3 Validación virtual, del comportamiento del robot móvil para el traslado de coches

Se realiza una validación entre varios escenarios que pueden relacionarse entre sí, en el control 1: ángulo de posición del robot teta1 respecto de la posición del obstáculo y ángulo de posición del robot teta 2 respecto de la posición deseada, ambas mediante la variación de velocidad de las ruedas; en el control 2 el movimiento del robot a través de la velocidad de ambas ruedas; y el análisis de los valores de las velocidades de cada rueda:

#### Control 1

##### Ángulo teta 1

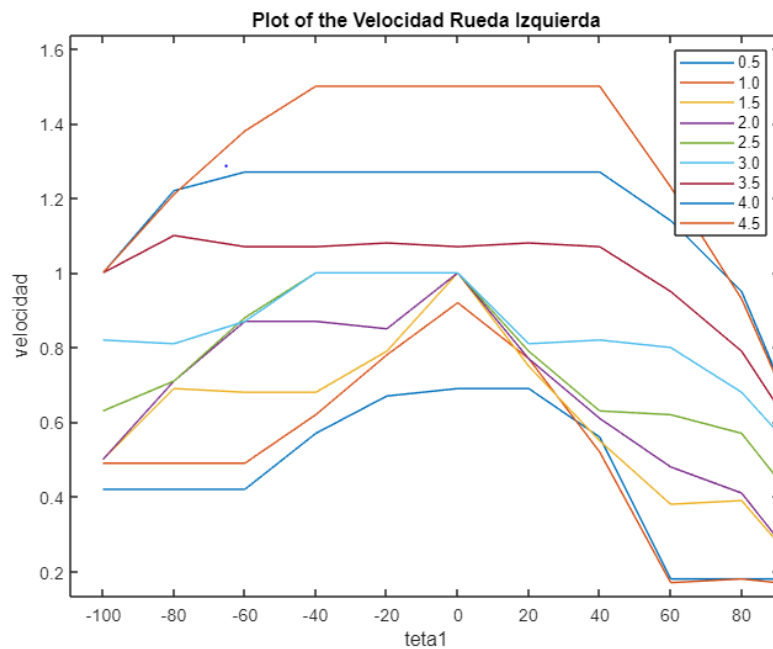
Se ingresan en el guide del control 1 Tesis.m, valores de entrada teta 1 con un rango de  $[-100^{\circ}, 100^{\circ}]$  e intervalos de  $20^{\circ}$  y valores de EDO ascendente de 0.5 m, colocamos iniciar y nos mostrará dos valores de salida de las ruedas, luego procedemos a tabular, según la Tabla 2.6, luego separamos las filas de valores de velocidad tanto de la rueda izquierda como de la rueda derecha para proceder a elaborar el programa que nos permitirá graficar en Matlab.

Con los datos de la Tabla 2.6, creamos un programa ver ANEXO 9 que permite generar una gráfica, a través de la función plot, y cuyos valores de salida corresponden a las velocidades de las ruedas en función del ángulo teta 1 y el EDO; para añadir más gráficas en la misma ventana, se deben anteceder en el programa: **hold on** al iniciar y **hold off** al finalizar. Para representar el error de distancia del obstáculo ingresamos legend con un rango (0.5 - 4.5), de manera ascendente e intervalos de 0.5 ver Figura 2.41.

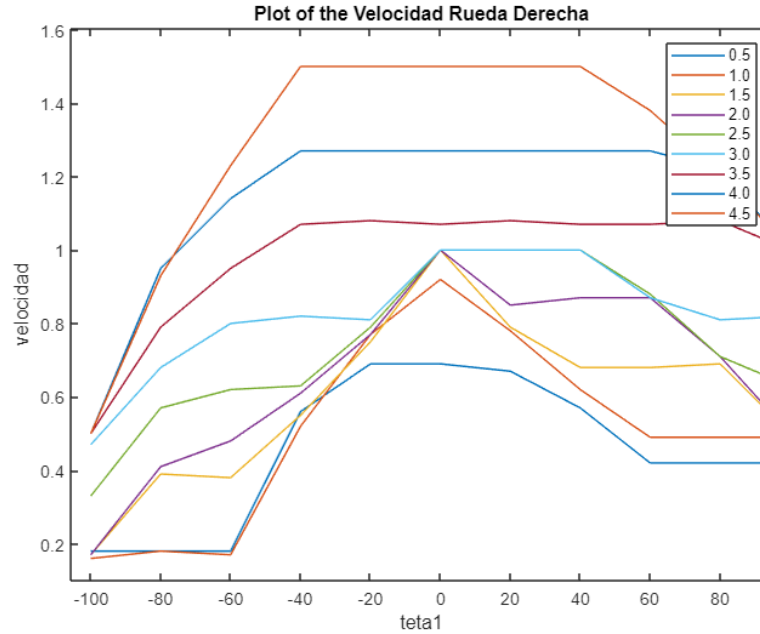
RUEDAS	EDO	teta 1(°)										
		-100	-80	-60	-40	-20	0	20	40	60	80	100
RI	0,5	0,42	0,42	0,42	0,57	0,67	0,69	0,69	0,56	0,18	0,18	0,18
RD	0,5	0,18	0,18	0,18	0,56	0,69	0,69	0,67	0,57	0,42	0,42	0,42
RI	1,0	0,49	0,49	0,49	0,62	0,78	0,92	0,77	0,52	0,17	0,18	0,16
RD	1,0	0,16	0,18	0,17	0,52	0,77	0,92	0,78	0,62	0,49	0,49	0,49
RI	1,5	0,50	0,69	0,68	0,68	0,79	1,00	0,75	0,55	0,38	0,39	0,17
RD	1,5	0,17	0,39	0,38	0,55	0,75	1,00	0,79	0,68	0,68	0,69	0,50
RI	2,0	0,50	0,71	0,87	0,87	0,85	1,00	0,77	0,61	0,48	0,41	0,17
RD	2,0	0,17	0,41	0,48	0,61	0,77	1,00	0,85	0,87	0,87	0,71	0,50
RI	2,5	0,63	0,71	0,88	1,00	1,00	1,00	0,79	0,63	0,62	0,57	0,33
RD	2,5	0,33	0,57	0,62	0,63	0,79	1,00	1,00	1,00	0,88	0,71	0,63
RI	3,0	0,82	0,81	0,87	1,00	1,00	1,00	0,81	0,82	0,80	0,68	0,47
RD	3,0	0,47	0,68	0,80	0,82	0,81	1,00	1,00	1,00	0,87	0,81	0,82
RI	3,5	1,00	1,10	1,07	1,07	1,08	1,07	1,08	1,07	0,95	0,79	0,50
RD	3,5	0,50	0,79	0,95	1,07	1,08	1,07	1,08	1,07	1,07	1,08	1,00
RI	4,0	1,00	1,22	1,27	1,27	1,27	1,27	1,27	1,27	1,14	0,95	0,50
RD	4,0	0,50	0,95	1,14	1,27	1,27	1,27	1,27	1,27	1,27	1,22	1,00
RI	4,5	1,00	1,21	1,38	1,50	1,50	1,50	1,50	1,50	1,23	0,93	0,50
RD	4,5	0,50	0,93	1,23	1,50	1,50	1,50	1,50	1,50	1,38	1,21	1,00

Tabla 2.6 Velocidades de ruedas

Fuente: Propia



(a)



(b)

Figura 2.41 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha

Fuente: Propia

## Ángulo teta 2

Ingresamos en el guide de Tesis.m, los valores del ángulo teta 2 en el rango  $[-180^{\circ}, 180^{\circ}]$  e intervalos de  $45^{\circ}$ , y valores de EDO ascendente de 0.5 m, se coloca en la pantalla iniciar y nos mostrará dos valores de salida de las ruedas, se procede a tabular los valores resultantes en la Tabla 2.7, separamos las filas de valores de velocidad en ambas ruedas para elaborar el programa correspondiente que nos permita graficar en Matlab.

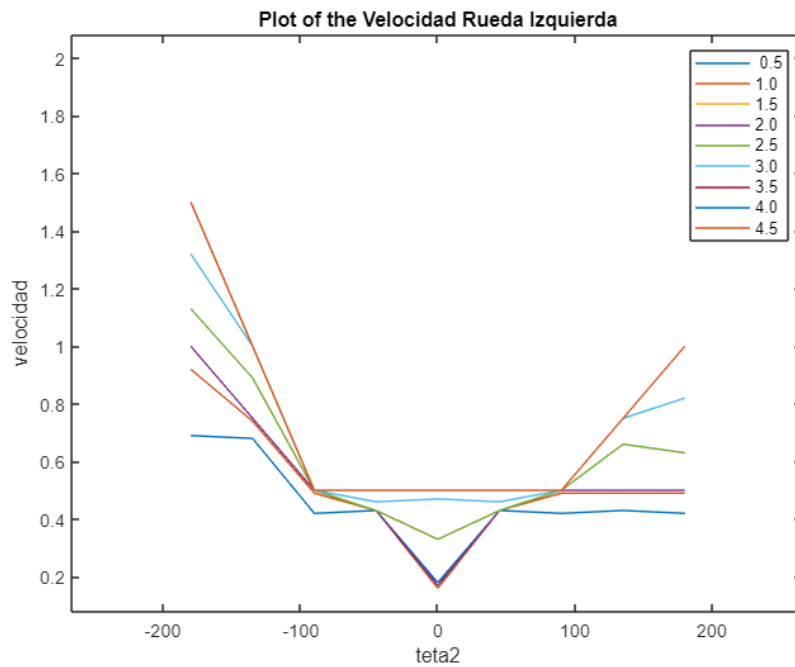
Según los datos de la Tabla 2.7, al igual que en el control 1 se procede a crear una gráfica con el ingreso del programa control 1 teta 2 ver ANEXO 10, usando la función plot, para representar las velocidades de las ruedas en función del ángulo teta 2 y el error de distancia del obstáculo, ver Figura 2.42.



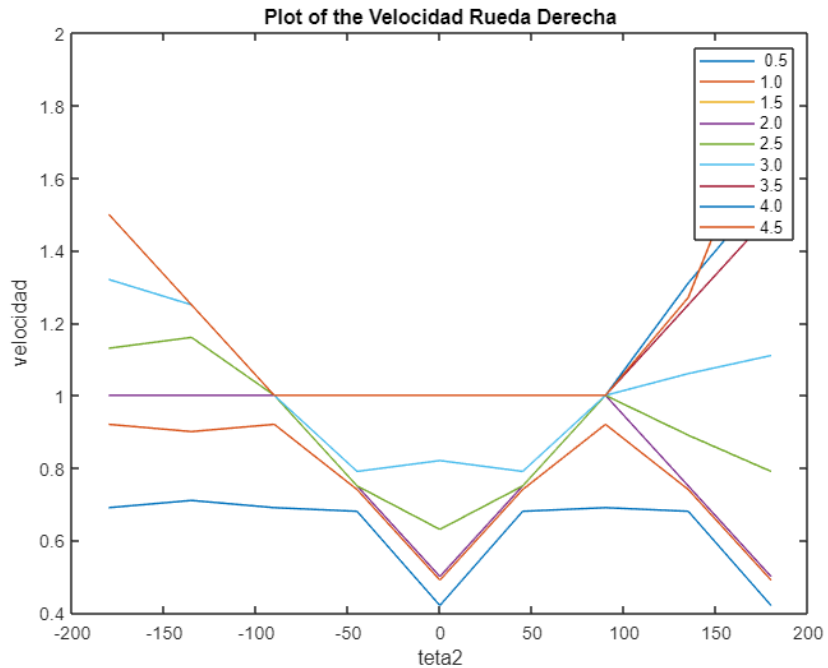
RUEDAS	EDO	teta 2									
		-180	-135	-90	-45	0	45	90	135	180	
RI	0,5	0,69	0,68	0,42	0,43	0,18	0,43	0,42	0,43	0,42	
RD	0,5	0,69	0,71	0,69	0,68	0,42	0,68	0,69	0,68	0,42	
RI	1,0	0,92	0,74	0,49	0,43	0,16	0,43	0,49	0,49	0,49	
RD	1,0	0,92	0,90	0,92	0,74	0,49	0,74	0,92	0,74	0,49	
RI	1,5	1,00	0,75	0,50	0,43	0,17	0,43	0,50	0,50	0,50	
RD	1,5	1,00	1,00	1,00	0,75	0,50	0,75	1,00	0,75	0,50	
RI	2,0	1,00	0,75	0,50	0,43	0,17	0,43	0,50	0,50	0,50	
RD	2,0	1,00	1,00	1,00	0,75	0,50	0,75	1,00	0,75	0,50	
RI	2,5	1,13	0,89	0,50	0,43	0,33	0,43	0,50	0,66	0,63	
RD	2,5	1,13	1,16	1,00	0,75	0,63	0,75	1,00	0,89	0,79	
RI	3,0	1,32	1,00	0,50	0,46	0,47	0,46	0,50	0,75	0,82	
RD	3,0	1,32	1,25	1,00	0,79	0,82	0,79	1,00	1,06	1,11	
RI	3,5	1,50	1,00	0,50	0,50	0,50	0,50	0,50	0,75	1,00	
RD	3,5	1,50	1,25	1,00	1,00	1,00	1,00	1,00	1,25	1,50	
RI	4,0	1,50	1,00	0,50	0,50	0,50	0,50	0,50	0,75	1,00	
RD	4,0	1,50	1,25	1,00	1,00	1,00	1,00	1,00	1,31	1,58	
RI	4,5	1,50	1,00	0,50	0,50	0,50	0,50	0,50	0,75	1,00	
RD	4,5	1,50	1,25	1,00	1,00	1,00	1,00	1,00	1,27	1,84	

Tabla 2.7 Velocidades de ruedas

Fuente: Propia



(a)



(b)

Figura 2.42 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha

Fuente: Propia

## Control 2

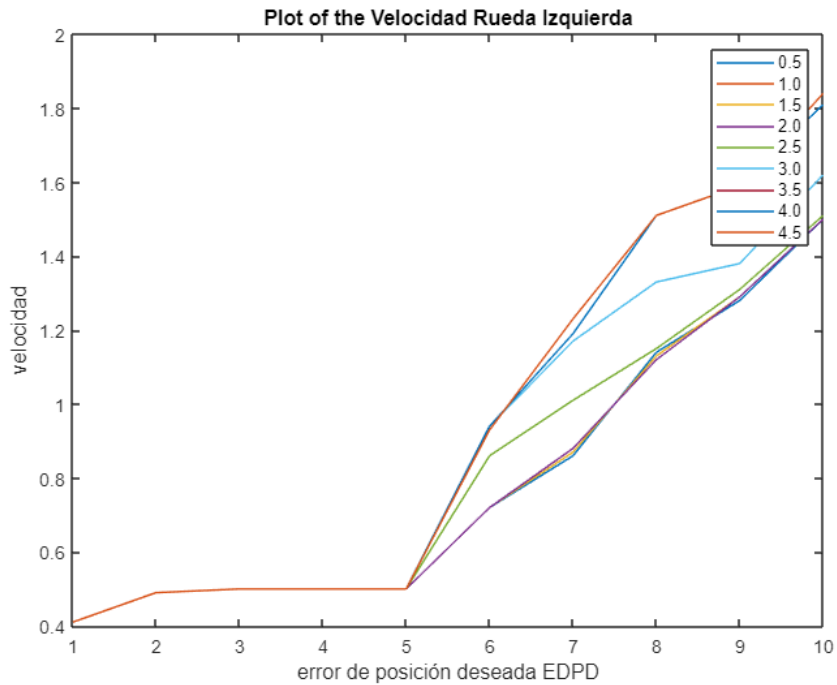
Se ingresaron valores en distancias de la posición deseada [1,10] a intervalos de 1, en el guide del control 2 Tesis2.m, tabulamos en Excel, según la Tabla 2.8, separamos las filas de valores de velocidad tanto de la rueda izquierda como de la rueda derecha para proceder a graficar.

En la Tabla 2.8, se puede desarrollar el programa control 2 EDPD ver ANEXO 11 que permite crear una gráfica, usando la función plot, que representan las velocidades de las ruedas en función del error de distancia deseada EDPD y el error de distancia del obstáculo; ver Figura 2.43.

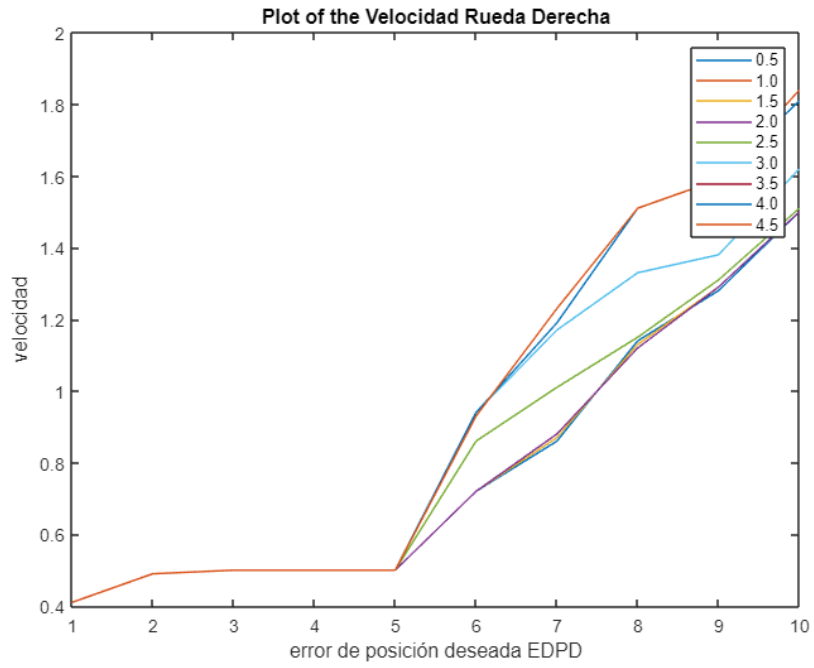
RUEDAS	EDO	error de posición deseada EDPD(m)									
		1	2	3	4	5	6	7	8	9	10
RI	0,5	0,41	0,49	0,5	0,5	0,5	0,72	0,86	1,14	1,28	1,5
RD	0,5	0,41	0,49	0,5	0,5	0,5	0,72	0,86	1,14	1,28	1,5
RI	1,0	0,41	0,49	0,5	0,5	0,5	0,72	0,88	1,12	1,29	1,5
RD	1,0	0,41	0,49	0,5	0,5	0,5	0,72	0,88	1,12	1,29	1,5
RI	1,5	0,41	0,49	0,5	0,5	0,5	0,72	0,87	1,13	1,29	1,5
RD	1,5	0,41	0,49	0,5	0,5	0,5	0,72	0,87	1,13	1,29	1,5
RI	2,0	0,41	0,49	0,5	0,5	0,5	0,72	0,88	1,12	1,29	1,5
RD	2,0	0,41	0,49	0,5	0,5	0,5	0,72	0,88	1,12	1,29	1,5
RI	2,5	0,41	0,49	0,5	0,5	0,5	0,86	1,01	1,15	1,31	1,51
RD	2,5	0,41	0,49	0,5	0,5	0,5	0,86	1,01	1,15	1,31	1,51
RI	3,0	0,41	0,49	0,5	0,5	0,5	0,94	1,17	1,33	1,38	1,62
RD	3,0	0,41	0,49	0,5	0,5	0,5	0,94	1,17	1,33	1,38	1,62
RI	3,5	0,41	0,49	0,5	0,5	0,5	0,93	1,23	1,51	1,59	1,84
RD	3,5	0,41	0,49	0,5	0,5	0,5	0,93	1,23	1,51	1,59	1,84
RI	4,0	0,41	0,49	0,5	0,5	0,5	0,94	1,19	1,51	1,59	1,81
RD	4,0	0,41	0,49	0,5	0,5	0,5	0,94	1,19	1,51	1,59	1,81
RI	4,5	0,41	0,49	0,5	0,5	0,5	0,93	1,23	1,51	1,59	1,84
RD	4,5	0,41	0,49	0,5	0,5	0,5	0,93	1,23	1,51	1,59	1,84

Tabla 2.8 Velocidades de ruedas

Fuente: Propia



(a)



(b)

Figura 2.43 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha

Fuente: Propia

### 3. RESULTADOS Y DISCUSIÓN

En este capítulo se detalla el funcionamiento del proyecto control difuso, las simulaciones realizadas y los resultados obtenidos

#### 3.1 Pruebas y resultados

##### Desarrollo de la interfaz hombre máquina

Ingresamos la palabra fuzzy y ejecutamos en la ventana de comandos de MATLAB nos desplegará la pantalla de fuzzy, abrimos Rules Viewer; ingresamos la palabra guide y ejecutamos nos mostrará la pantalla del HMI, seleccionamos el archivo guardado Tesis.fig correspondiente a la interfaz del Control, seleccionamos valores en el fuzzy luego valores en el guide, comparamos.

##### Control 1

En la ventana de comandos de Matlab se ingresa la palabra fuzzy, ejecutamos y abrimos las reglas de membresía (Rules Viewer), seleccionamos el ángulo  $teta1 = -83.58^\circ$ , el ángulo  $teta2 = 48.36^\circ$  y el EDO= 3.056, la inferencia del sistema indica que la velocidad en la rueda izquierda  $RI = 1.02$ , velocidad en la rueda derecha  $RD = 0.891$ , ver Figura 3.1.

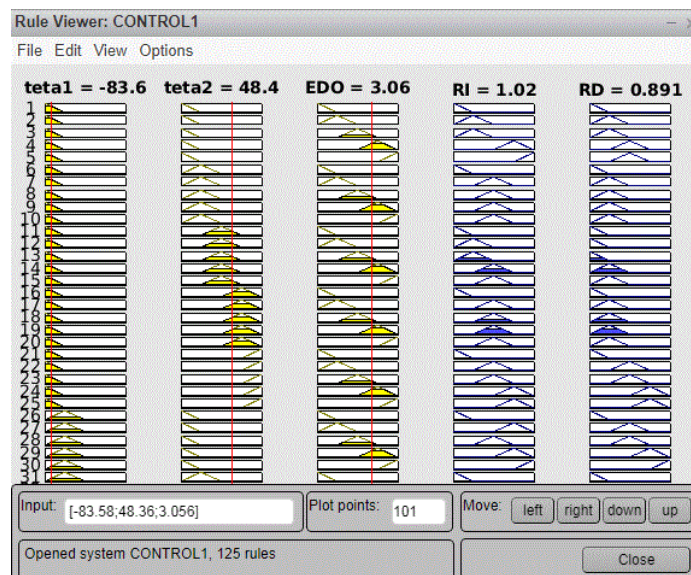


Figura 3.1 Rules Viewer Control 1

Fuente: Propia

Luego ingresamos la palabra guide y ejecutamos nos sale la pantalla de la interfaz procedemos a ingresar los valores de entrada antes mencionados presionamos el botón iniciar nos saldrá los valores de salida RI= 1.01913 y RD= 0.890817, ver Figura 3.2.

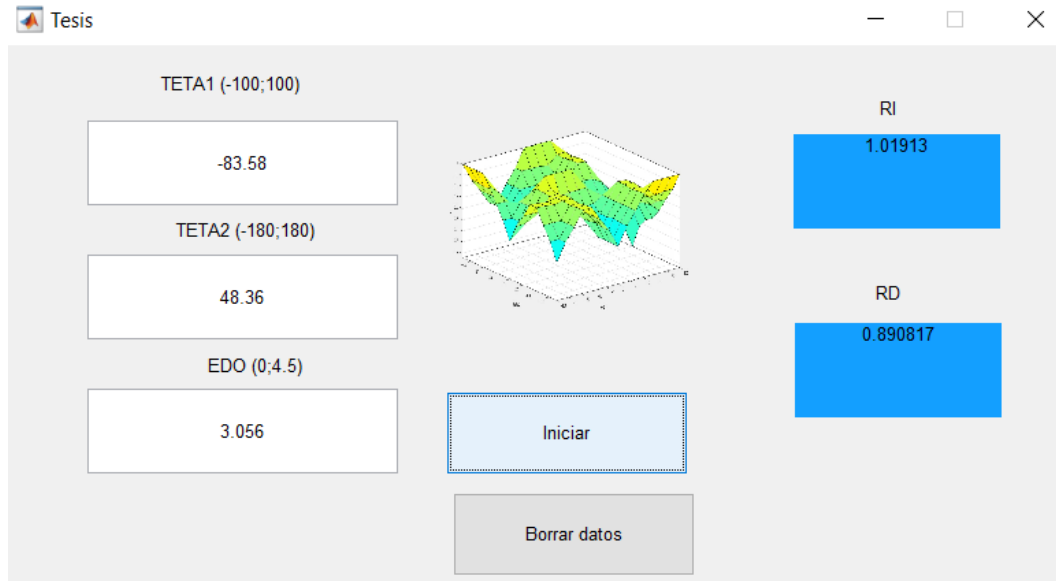


Figura 3.2 Guide Control 1

Fuente: Propia

## Control 2

Al igual que en el Control 1 en la ventana de comandos de Matlab se ingresa la palabra fuzzy, ejecutamos y abrimos las reglas de membresía (Rules Viewer), ingresamos los valores EDO= 2.73 y el EDPD= 5, la inferencia del sistema indica que la velocidad en ambas ruedas RI, RD son valores de 0.5, ver Figura 3.3.

Al ingresar palabra guide y ejecutamos, ingresamos los valores de entrada de EDO= 2.726 y EDPD= 5, presionando el botón iniciar en la pantalla de la interfaz gráfica nos saldrá los valores de salida RI y RD = 0.5, ver Figura 3.4.

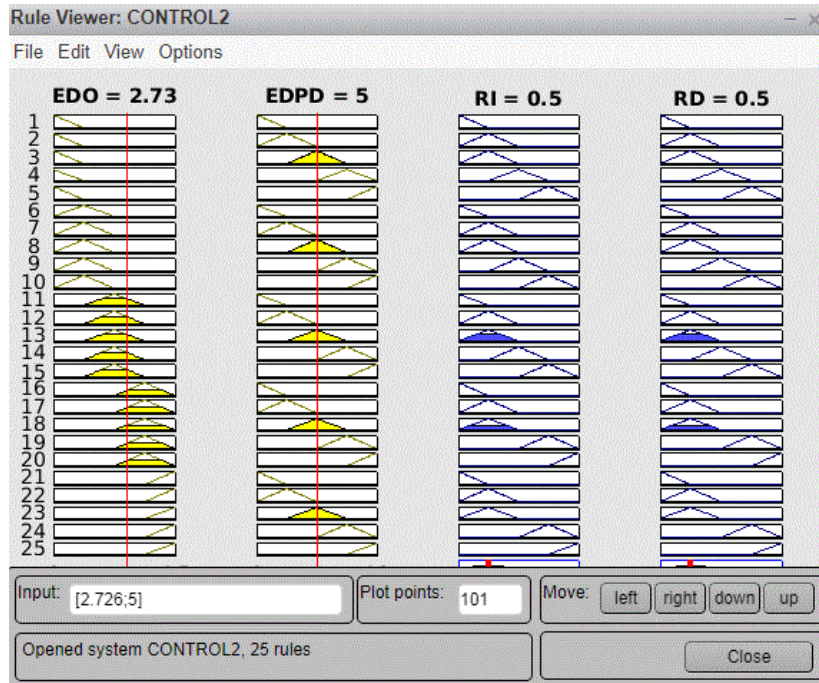


Figura 3.3 Rules Viewer Control 2

Fuente: Propia

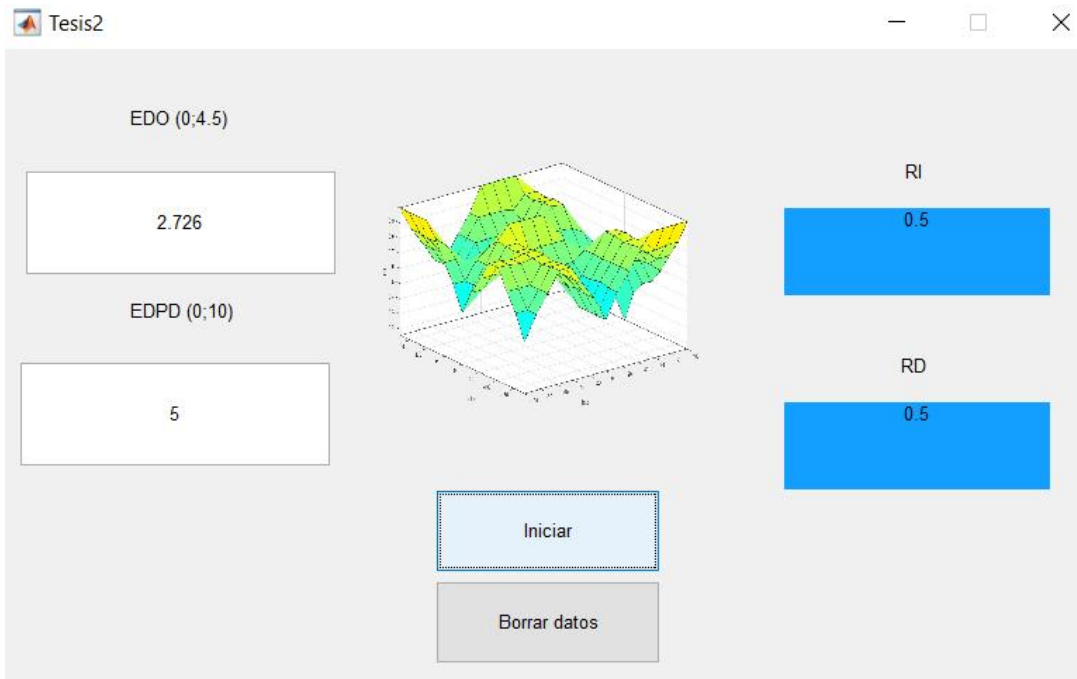


Figura 3.4 Guide Control 2

Fuente: Propia

## Validación virtual, del comportamiento del robot móvil para el traslado de coches

Se procede a simplificar la cantidad de gráficas dentro de cada programa a fin de que sea fácil su interpretación, se considera 5 grupos de datos y las correspondientes sentencias, entre los que podemos mencionar:

### Control 1

#### Ángulo teta 1

En el desarrollo del programa se utiliza la función plot, en el eje x se representa los ángulos teta1, en el eje y las velocidades de ambas ruedas, ver Figura 3. 5.

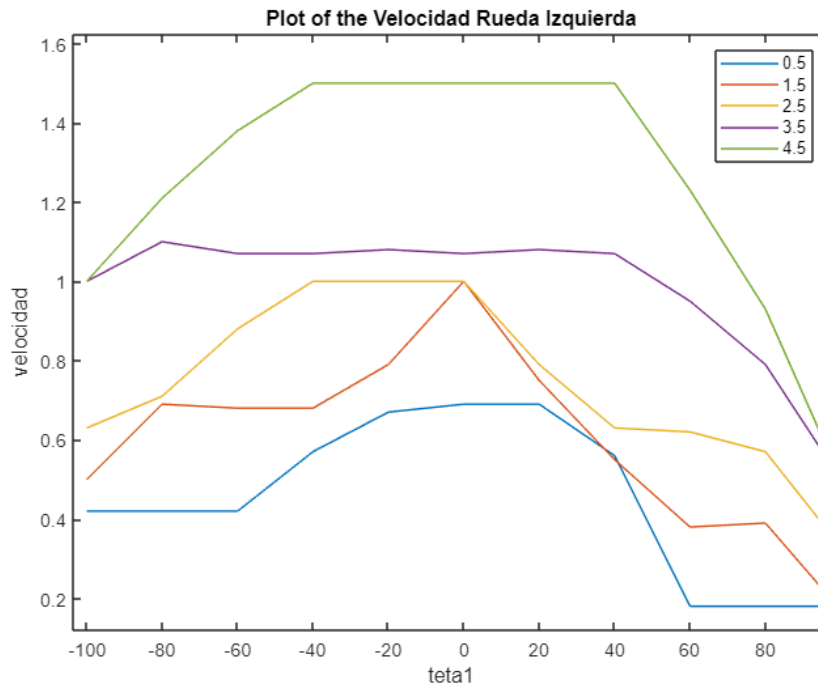
Programa Velocidad Rueda Izquierda

```
x=[-100 -80 -60 -40 -20 0 20 40 60 80 100]
o=[0.42 0.42 0.42 0.57 0.67 0.69 0.69 0.56 0.18 0.18 0.18]
plot(x,o)
xlabel('teta1')
ylabel('velocidad')
title('Plot of the Velocidad Rueda Izquierda')
hold on
q=[0.50 0.69 0.68 0.68 0.79 1.00 0.75 0.55 0.38 0.39 0.17]
s=[0.63 0.71 0.88 1.00 1.00 1.00 0.79 0.63 0.62 0.57 0.33]
u=[1.00 1.10 1.07 1.07 1.08 1.07 1.08 1.07 0.95 0.79 0.50]
w=[1.00 1.21 1.38 1.50 1.50 1.50 1.50 1.50 1.23 0.93 0.50]
plot(x,q)
plot(x,s)
plot(x,u)
plot(x,w)
hold off
legend('0.5','1.5','2.5','3.5','4.5')
```

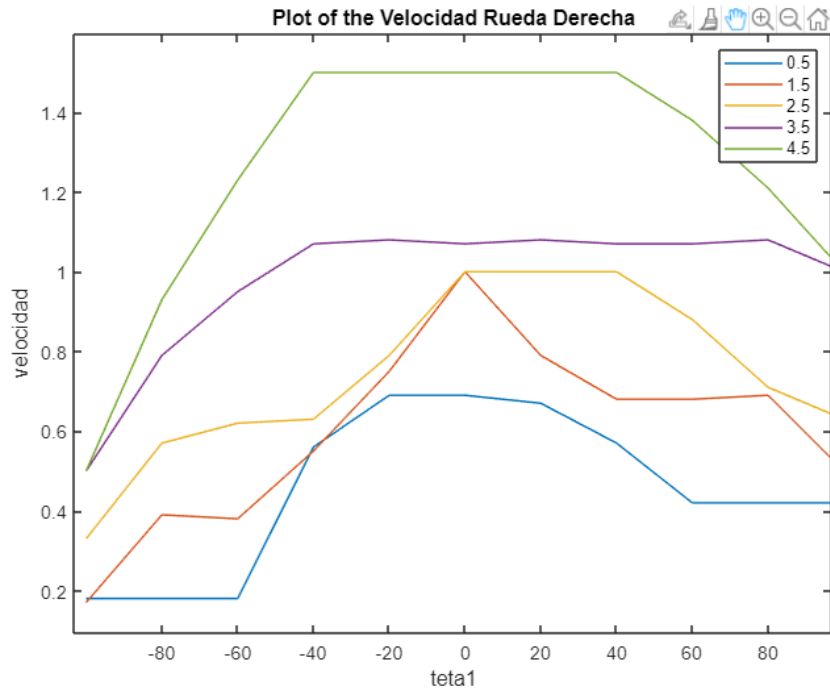


## Programa Velocidad Rueda Derecha

```
x=[-100 -80 -60 -40 -20 0 20 40 60 80 100]
o=[0.18 0.18 0.18 0.56 0.69 0.69 0.67 0.57 0.42 0.42 0.42]
plot(x,o)
xlabel('teta1')
ylabel('velocidad')
title('Plot of the Velocidad Rueda Derecha')
hold on
q=[0.17 0.39 0.38 0.55 0.75 1.00 0.79 0.68 0.68 0.69 0.50]
s=[0.33 0.57 0.62 0.63 0.79 1.00 1.00 1.00 0.88 0.71 0.63]
u=[0.50 0.79 0.95 1.07 1.08 1.07 1.08 1.07 1.07 1.08 1.00]
w=[0.50 0.93 1.23 1.50 1.50 1.50 1.50 1.50 1.38 1.21 1.00]
plot(x,q)
plot(x,s)
plot(x,u)
plot(x,w)
hold off
legend('0.5','1.5','2.5','3.5','4.5')
```



(a)



(b)

Figura 3.5 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha

Fuente: Propia

## Ángulo teta 2

En el desarrollo del programa se utiliza la función plot, en el eje x se representa los ángulos teta2, en el eje y las velocidades de ambas ruedas, ver Figura 3. 6.

Programa Velocidad Rueda Izquierda

```
x=[-180 -135 -90 -45 0 45 90 135 180]
o=[0.69 0.68 0.42 0.43 0.18 0.43 0.42 0.43 0.42]
plot(x,o)
xlabel('teta2')
ylabel('velocidad')
title('Plot of the Velocidad Rueda Izquierda')
hold on
q=[1.00 0.75 0.50 0.43 0.17 0.43 0.50 0.50 0.50]
s=[1.13 0.89 0.50 0.43 0.33 0.43 0.50 0.66 0.63]
u=[1.50 1.00 0.50 0.50 0.50 0.50 0.50 0.75 1.00]
```

```

w=[1.50 1.00 0.50 0.50 0.50 0.50 0.50 0.75 1.00]
plot(x,q)
plot(x,s)
plot(x,u)
plot(x,w)
hold off
legend(' 0.5','1.5','2.5','3.5','4.5')

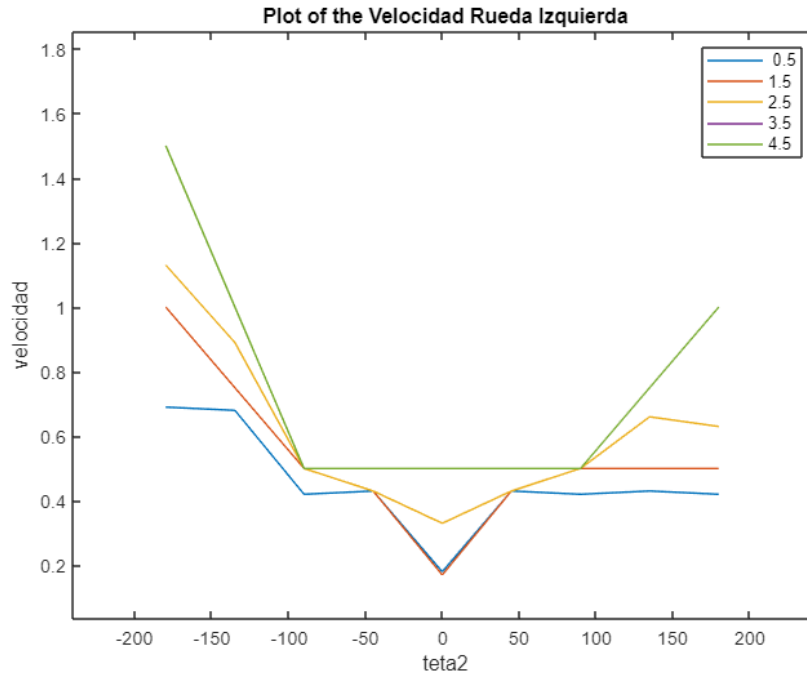
```

### Programa Velocidad Rueda Derecha

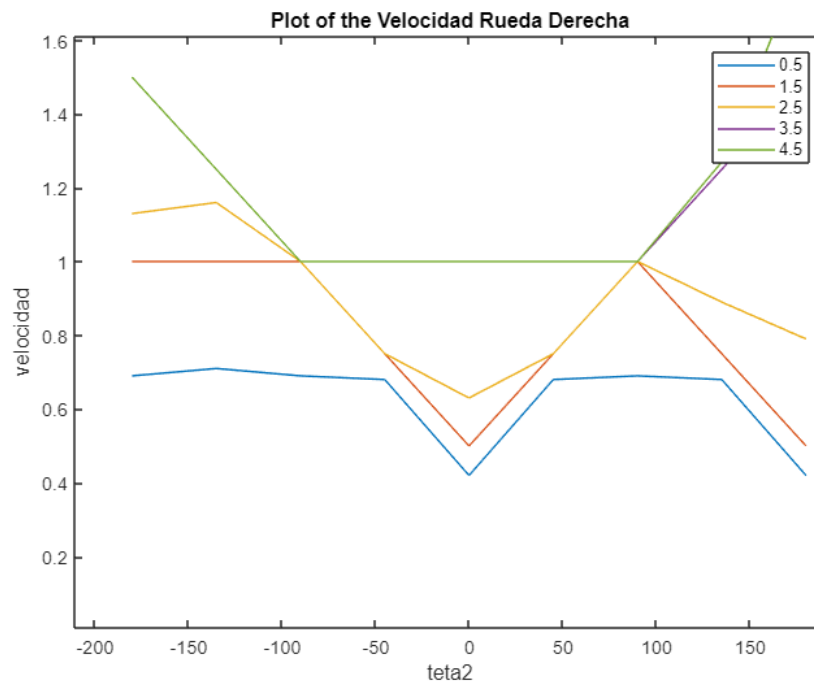
```

x=[-180 -135 -90 -45 0 45 90 135 180]
o=[0.69 0.71 0.69 0.68 0.42 0.68 0.69 0.68 0.42]
plot(x,o)
xlabel('teta2')
ylabel('velocidad')
title('Plot of the Velocidad Rueda Derecha')
hold on
q=[1.00 1.00 1.00 0.75 0.50 0.75 1.00 0.75 0.50]
s=[1.13 1.16 1.00 0.75 0.63 0.75 1.00 0.89 0.79]
u=[1.50 1.25 1.00 1.00 1.00 1.00 1.00 1.25 1.50]
w=[1.50 1.25 1.00 1.00 1.00 1.00 1.00 1.27 1.84]
plot(x,q)
plot(x,s)
plot(x,u)
plot(x,w)
hold off
legend(' 0.5','1.5','2.5','3.5','4.5')

```



(a)



(b)

Figura 3.6 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha  
Fuente: Propia

## Control 2

### Error de distancia EDPD

En el desarrollo del programa se utiliza la función plot, en el eje x se representa el error de distancia deseada EDPD, en el eje y las velocidades de ambas ruedas, ver Figura 3.7.

Programa Velocidad Rueda Izquierda

```
x=[1 2 3 4 5 6 7 8 9 10]
o=[0.41 0.49 0.50 0.50 0.50 0.72 0.86 1.14 1.28 1.50]
plot(x,o)
xlabel('error de posición deseada EDPD')
ylabel('velocidad')
title('Plot of the Velocidad Rueda Izquierda')
hold on
q=[0.41 0.49 0.50 0.50 0.50 0.72 0.87 1.13 1.29 1.50]
s=[0.41 0.49 0.50 0.50 0.50 0.86 1.01 1.15 1.31 1.51]
u=[0.41 0.49 0.50 0.50 0.50 0.93 1.23 1.51 1.59 1.84]
w=[0.41 0.49 0.50 0.50 0.50 0.93 1.23 1.51 1.59 1.84]
plot(x,q)
plot(x,s)
plot(x,u)
plot(x,w)
hold off
legend(' 0.5','1.5','2.5','3.5','4.5')
```

Programa Velocidad Rueda Derecha

```
x=[1 2 3 4 5 6 7 8 9 10]
o=[0.41 0.49 0.50 0.50 0.50 0.72 0.86 1.14 1.28 1.50]
plot(x,o)
xlabel('error de posición deseada EDPD')
ylabel('velocidad')
title('Plot of the Velocidad Rueda Derecha')
```

**hold on**

q=[0.41 0.49 0.50 0.50 0.50 0.72 0.87 1.13 1.29 1.50]

s=[0.41 0.49 0.50 0.50 0.50 0.86 1.01 1.15 1.31 1.51]

u=[0.41 0.49 0.50 0.50 0.50 0.93 1.23 1.51 1.59 1.84]

w=[0.41 0.49 0.50 0.50 0.50 0.93 1.23 1.51 1.59 1.84]

plot(x,p)

plot(x,q)

plot(x,r)

plot(x,s)

plot(x,t)

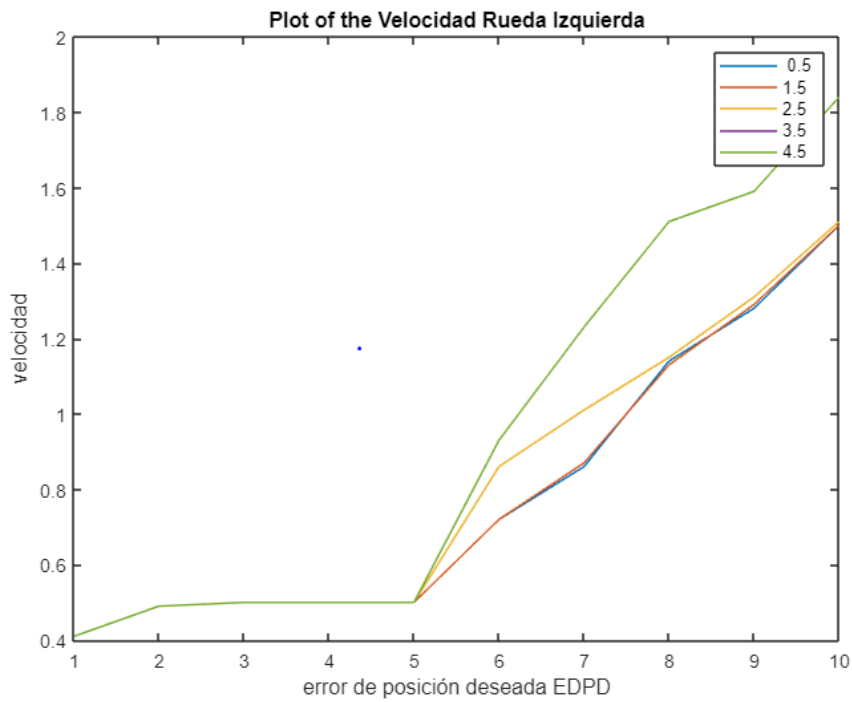
plot(x,u)

plot(x,v)

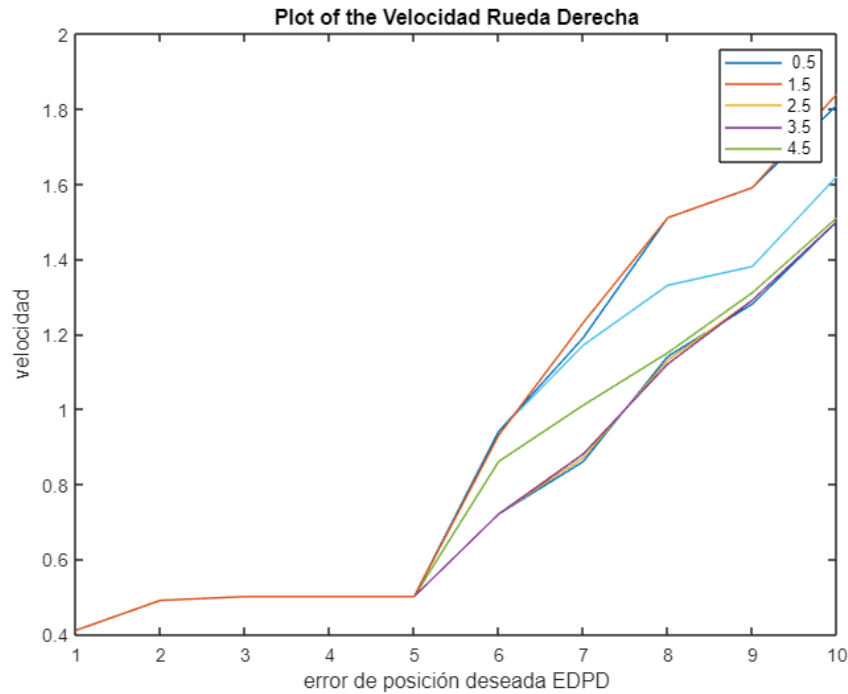
plot(x,w)

**hold off**

legend(' 0.5','1.5','2.5','3.5','4.5')



(a)



(b)

Figura 3.7 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha

Fuente: Propia

## 3.2 Análisis y discusión de resultados

### Desarrollo de la interfaz hombre máquina

En el análisis de los datos de las variables de salida RI, RD en el Control 1, se elabora una tabla que contiene los valores de entrada de tres variables teta 1, teta 2 y EDO, con sus variables de salida en cada gráfica generada en fuzzy y guide, calculamos el porcentaje de error entre los dos valores que resultan de restar el valor obtenido en las reglas de membresía del control fuzzy con el archivo CONTROL1.fis y los valores de la guide de la interfaz en el archivo Tesis.fig, según la tabla 3.1, se analizan 6 grupos de datos.

Mientras que en el análisis de los datos de las variables de salida RI, RD en el Control 2, se procede a elaborar una tabla que contenga los valores de entrada de dos variables EDO y EDPD, con las variables de salida, de las gráficas en fuzzy y guide, calculamos el porcentaje de error entre los dos valores, al restar el valor obtenido en las reglas de

membresía del control fuzzy con el archivo CONTROL2.fis y los valores de la guide de la interfaz en el archivo Tesis2.fig, según la Tabla 3.2, se analizan 4 grupos de datos.

CONTROL 1	teta 1	teta 2	EDO	RI			RD		
				Fuzzy	guide	%	fuzzy	guide	%
1	-83,6	48,4	3,1	1,020	1,019	0,087	0,891	0,891	0,018
2	-81,7	-142,7	0,2	0,742	0,742	0,044	0,527	0,527	0,030
3	0,0	0,0	4,5	1,500	1,500	0,000	1,500	1,500	0,000
4	33,0	65,0	0,0	0,176	0,176	0,039	0,176	0,176	0,039
5	33,0	65,0	4,5	1,620	1,615	0,473	1,620	1,615	0,473
6	-50,0	-50,0	3,2	1,270	1,272	0,228	1,160	1,163	0,322

Tabla 3.1 Evaluación de datos de entradas Control 1

Fuente: Propia

CONTROL 2	EDO	EDPD	RI			RD		
			fuzzy	Guide	%	fuzzy	guide	%
1	0,0	0,0	0,160	0,160	0,000	0,160	0,160	0,000
2	0,5	5,0	0,500	0,500	0,000	0,500	0,500	0,000
3	1,5	0,9	0,392	0,392	0,028	0,392	0,392	0,028
4	3,5	8,0	1,510	1,512	0,154	1,510	1,512	0,154

Tabla 3.2 Evaluación de datos de entradas Control 2

Fuente: Propia



Los porcentajes de variación entre los valores de salida de las gráficas de fuzzy y guide, son menos de 1%, por lo que se consideran que se han desarrollado de acuerdo con los requerimientos del control difuso.

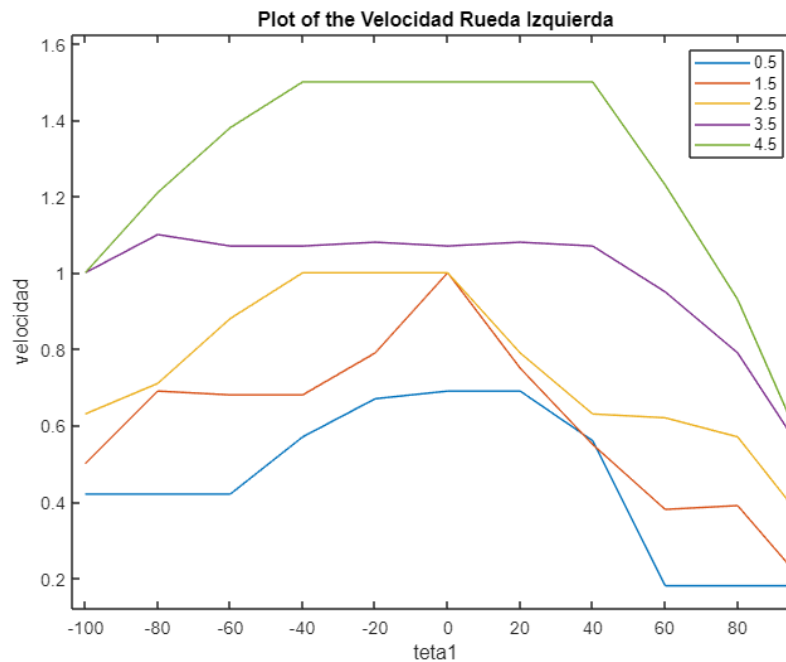
## Validación virtual, del comportamiento del robot móvil para el traslado de coches

Se procede a evaluar las gráficas resultantes de los dos controladores en función de las velocidades de ambas ruedas.

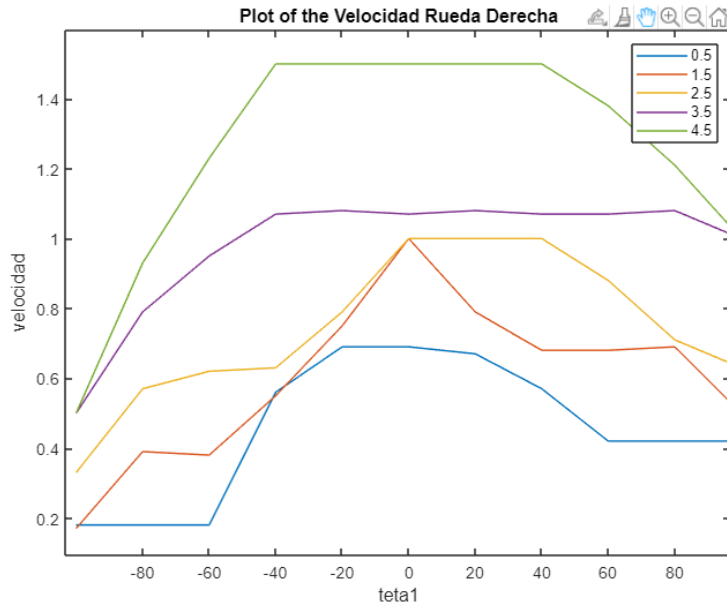
### Control 1

#### Ángulo teta 1

En la Figura 3.8, nos muestra que a mayor distancia se encuentra el robot de la posición del obstáculo aumenta la velocidad de traslado del robot, las gráficas de ambas ruedas son similares, la velocidad mínima cuando teta 1 sean  $-20^\circ$  o  $20^\circ$  es de 0.6 m/s, no se debe considerar cuando teta 1 sea 0 ya que mostraría que el obstáculo está alineado con el eje del robot y la velocidad debe ser mínima para evadir el obstáculo o para detenerse.



(a)



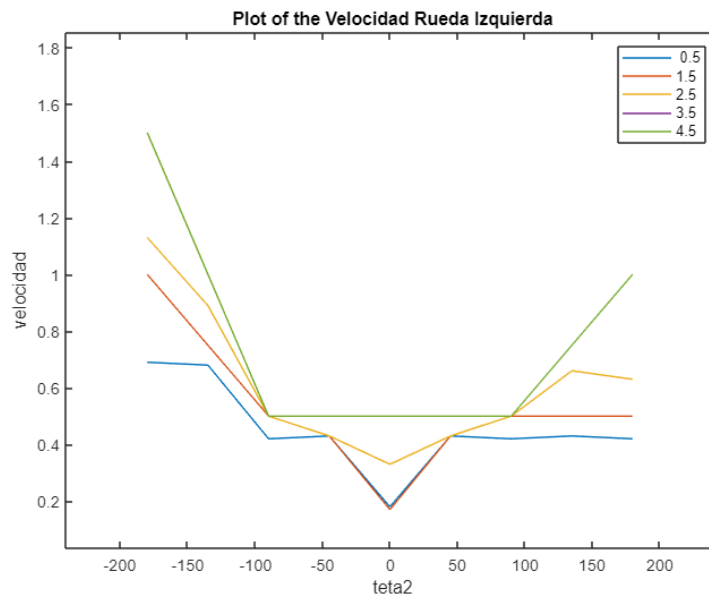
(b)

Figura 3.8 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha

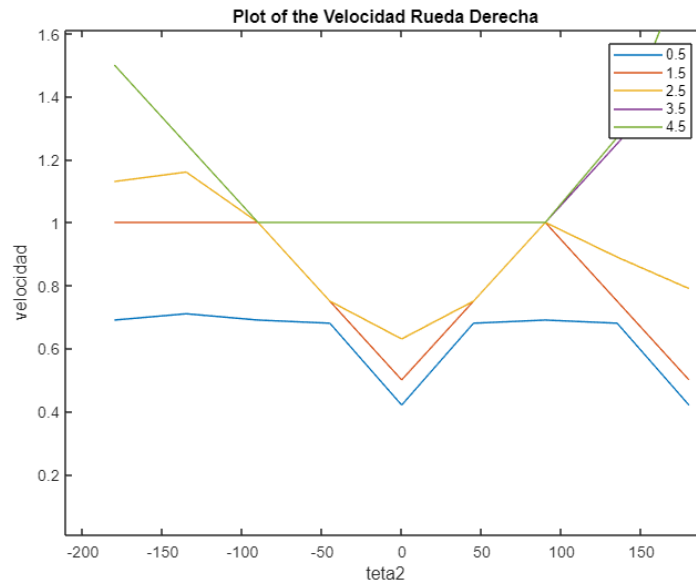
Fuente: Propia

## Ángulo teta 2

En la Figura 3.9, nos indica que a mayor distancia se encuentra el robot de la posición del obstáculo aumenta la velocidad de traslado del robot. Pero cuando se muestra en la gráfica el valor de teta 2 de 0, la velocidad en la rueda izquierda es 0.2 mientras que en la rueda derecha la velocidad mínima es 0.4 m/s.



(a)



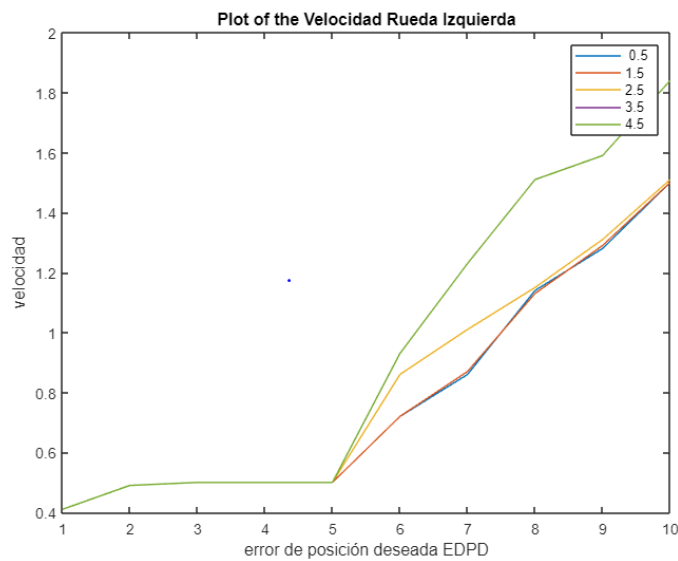
(b)

Figura 3.9 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha  
Fuente: Propia

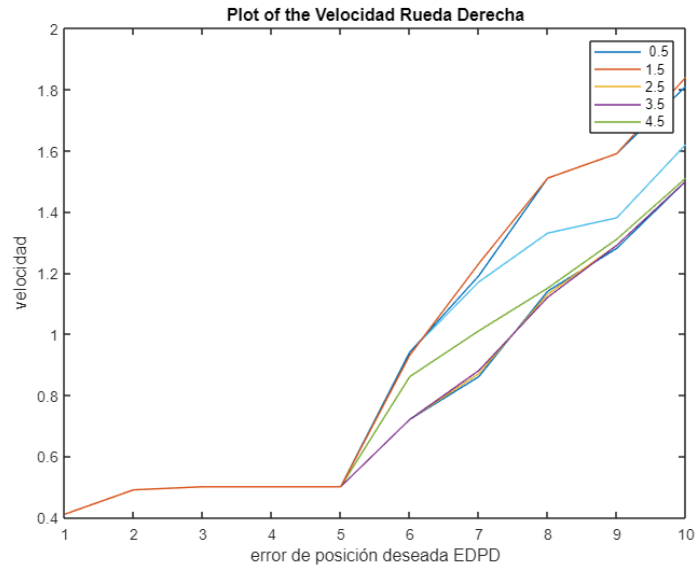
## Control 2

### Error de distancia EDPD

En la Figura 3.10, nos indica que entre distancias de 0 - 5 m, la velocidad tiende a ser constante cercano a 0.5 m/s, pero al aumento de distancia en forma progresiva también aumenta la velocidad del robot.



(a)



(b)

Figura 3.10 Velocidades (a) Rueda Izquierda y (b) Rueda Derecha

Fuente: Propia

### Diferencia de velocidades

Se procede a promediar los datos del eje y, por lo que nos genera una sola gráfica que representa a ambas velocidades de las ruedas.

### Control 1

Programa ángulo teta 1

```

x=[-100 -80 -60 -40 -20 0 20 40 60 80 100]
ri=[0.71 0.82 0.88 0.95 0.99 1.05 0.94 0.84 0.66 0.56 0.33]
plot(x,ri)
xlabel('teta1')
ylabel('velocidad')
title('Plot of the Velocidad de Ruedas')
hold on
rd=[0.33 0.56 0.66 0.84 0.94 1.05 0.99 0.95 0.88 0.82 0.71]
plot(x,rd,'r--')
hold off
legend('rueda izquierda','rueda derecha')

```

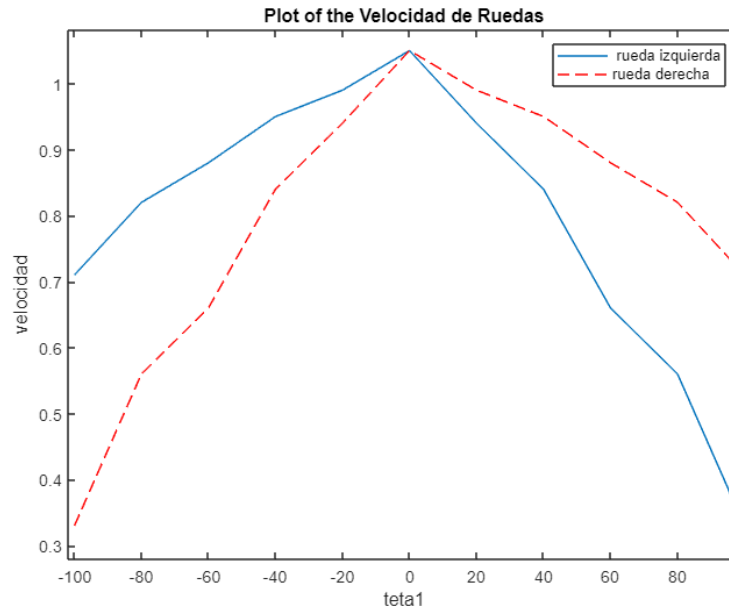


Figura 3.11 Velocidades teta 1

Fuente: Propia

La Figura 3.11 nos indica que hay diferencia de velocidades entre la rueda izquierda y la rueda derecha, el valor inicial cuando teta 1 es -100, el valor de velocidad de la rueda izquierda=0.7 mientras que para la rueda derecha=0.3.

Programa ángulo teta 2

```

x=[-180 -135 -90 -45 0 45 90 135 180]
ri=[1.17 0.87 0.49 0.46 0.33 0.46 0.49 0.62 0.71]
plot(x,ri)
xlabel('teta2')
ylabel('velocidad')
title('Plot of the Velocidad de Ruedas')
hold on
rd=[1.17 1.09 0.96 0.83 0.71 0.83 0.96 0.97 0.97]
plot(x,rd,'r--')
hold off
legend('rueda izquierda','rueda derecha')

```

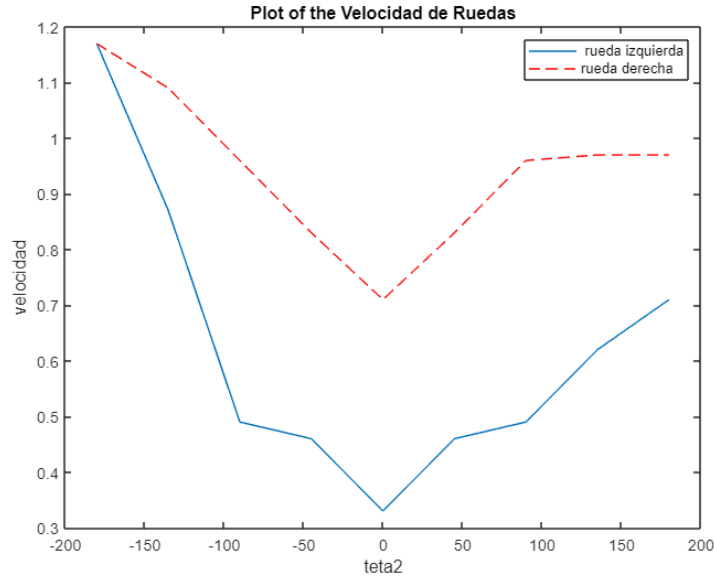


Figura 3.12 Velocidades teta 2

Fuente: Propia

La Figura 3.12 nos indica que hay diferencia de velocidades entre la rueda izquierda y la rueda derecha, el valor inicial cuando teta 1 es  $-180^\circ$ , consideramos que el ángulo de posición del obstáculo es de  $-100^\circ$ , es decir está lejos de la posición del robot respecto de la posición deseada, el valor de velocidad de ambas ruedas es = 1.17 m/s.

## Control 2

### Error de distancia de la posición deseada

```
x=[1 2 3 4 5 6 7 8 9 10]
ri=[0.41 0.49 0.50 0.50 0.50 0.83 1.04 1.28 1.40 1.62]
plot(x,ri)
xlabel('error de posición deseada EDPD')
ylabel('velocidad')
title('Plot of the Velocidad de Ruedas')
hold on
rd=[0.41 0.49 0.50 0.50 0.50 0.83 1.04 1.28 1.40 1.62]
plot(x,rd,'r--')
hold off
legend('rueda izquierda','rueda derecha')
```

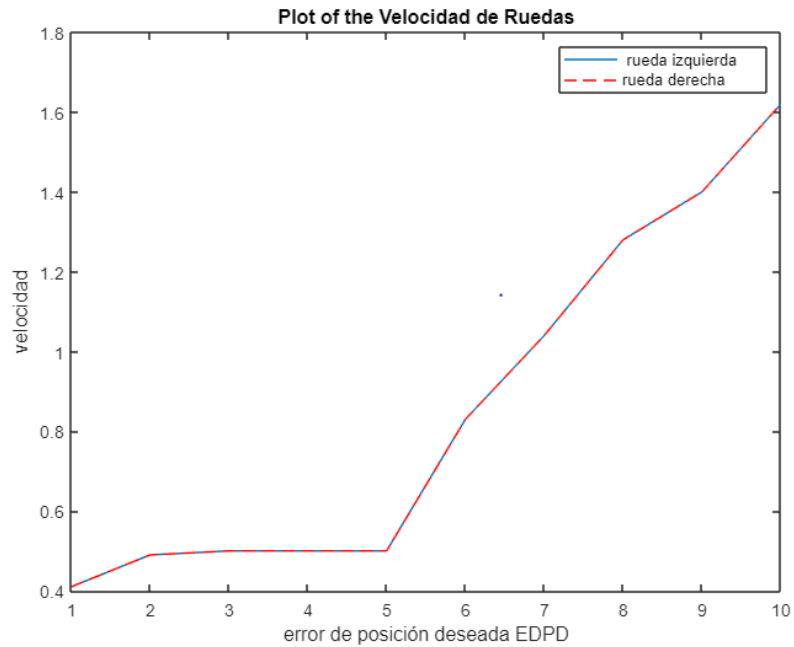


Figura 3.13 Velocidades EDPD

Fuente: Propia

La Figura 3.13 del control 2 nos indica que no hay diferencia de velocidades entre la rueda izquierda y la rueda derecha pues la gráfica de la rueda derecha se superpone sobre la gráfica de la rueda izquierda, por lo que se complementa con el control 1 para que navegue de forma autónoma a través de un control difuso.

## 4. CONCLUSIONES

### 4.1 Conclusiones

Se diseñó el sistema mecánico, sistema electrónico y el sistema de percepción que nos permitirá tomar como base para la construcción de un prototipo en un entorno real.

Se modeló el control inteligente del robot móvil mediante el diseño de dos controladores de control difuso, uno para controlar la dirección y otro para la velocidad de avance, por medio del control de las velocidades de las ruedas izquierda y derecha.

El número de reglas de membresía del control 1 son 125, en el control del ángulo de dirección del robot; y del control 2 son 25 reglas en el control de velocidad de avance, ambos controles con 5 valores lingüísticos cada uno.

Se desarrolló la interfaz hombre máquina en GUI de Matlab de manera sencilla, con el ingreso de las variables de entrada se realizó el análisis de datos de las variables de salida de las velocidades de las ruedas izquierda y derecha para cada uno de los controladores.

En el desarrollo de la interfaz se generaron los archivos en guide Tesis.m, Tesis.fig, Tesis2.m y Tesis2.fig con el ingreso en el programa de los archivos de las reglas de membresía CONTROL1.fis y CONTROL2.fis.

Se validó el modelo con el cálculo y análisis de las velocidades lineales de las ruedas izquierda y derecha expresadas en m/s tanto en fuzzy como en guide de Matlab, con el ingreso de tres variables de entrada para el control 1: ángulo teta 1, ángulo teta 2, error de distancia del obstáculo (EDO) y dos variables de entrada para el control 2: error de distancia del obstáculo (EDO), error de distancia de la posición deseada EDPD.

Se han cumplido con los objetivos de la tesis, diseño del sistema mecánico, electrónico y de percepción, modelación del control fuzzy, desarrollo de la interfaz hombre máquina en el paquete GUI de Matlab, además se validó los dos controladores propuestos.



## **4.2 Recomendaciones**

Si se requiere aumentar la precisión del sistema de percepción, se recomendaría usar sensores con otro principio de funcionamiento para complementar a los sensores de proximidad, se podría incorporar sensores láser tipo Lidar.

Ampliar el conocimiento para implementar el uso de la comunicación inalámbrica usando WIFI a través de Smartphone, ya que el uso de la comunicación a través del internet permite encaminarse al desarrollo de aplicaciones en tiempo real.

Se recomienda desarrollar un sistema de adquisición de datos, que permita comprobar la implementación de los controladores propuestos mediante la mayor cantidad de pruebas en la interfaz gráfica, por medio de validación en un entorno de realidad virtual y/o realidad aumentada.

## BIBLIOGRAFÍA

- ABB. (n.d.). *IRB 4400*. <https://new.abb.com/products/robotics/es/robots-industriales/irb-4400>
- Alba Mejía, E. J. (2019). *Modelado y sintonización de un controlador PID de un robot tipo puma de tres grados de libertad utilizando lógica difusa*. 86.
- Alejandro Medina; Jesús Ovejero. (2010). *Leyes de Newton y sus aplicaciones*. Universidad de Salamanca, 24.
- Alzate Gómez, A., López, A., & Restrepo Patiño, C. (2007). Control difuso de una plataforma móvil para el seguimiento de trayectorias. *Scientia et Technica*, 3(35), 169–174. <https://doi.org/10.22517/23447214.5391>
- Amazon. (n.d.). *Robot KIVA*.
- Amazon. (2020). *Motor Nema 34 12Nm*.
- Aprendiendo Arduino. (2020). *ZigBee / XBee*.
- Arduino Official Store. (n.d.). Retrieved October 16, 2020, from <https://store.arduino.cc/usa/mega-2560-r3>
- ASTM A36 steel. (n.d.). Retrieved October 16, 2020, from <http://www.gneesteel.com/products/carbon-steel/a36-steel.html>
- Aymesa. (2021). *Automóviles de Ecuador* (pp. 3–6).
- Barrientos, A., Del Cerro, J., Gutiérrez, P., San Martín, R., Martínez, A., & Rossi, C. (2009). Vehículos aéreos no tripulados para uso civil. Tecnología y aplicaciones. *Grupo de Robótica y Cibernética, Universidad Politécnica de Madrid*, 1–29. <http://webdiis.unizar.es/~neira/docs/ABarrientos-CEDI2007.pdf>
- Bermúdez, G. (2002). Robots móviles. *TECNURA 10*. [http://platea.pntic.mec.es/vgonzale/cyr\\_0708/archivos/\\_15/Tema\\_5.5.htm](http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_5.5.htm)
- Bernal, J. C. (2005). *Diseño conceptual de productos asistido por ordenador: Un estudio analítico sobre aplicaciones y definición de la estructura básica de un nuevo programa*. <https://upcommons.upc.edu/handle/2117/94147>

- Calandín, L. I. G. (2006). *Modelado Cinemático y Control de Robots Móviles con Ruedas*. 306.  
<http://riunet.upv.es/bitstream/handle/10251/1840/tesisUPV2519.pdf>
- Cámara CCD. (n.d.).
- Colón, F. F., Espino, E. R., Humberto, J., & Azuela, S. (2016). Navegación de un robot humanoide mediante redes de Petri y lógica difusa. *Rcs.Cic.Ipn.Mx*, 135, 207–220.
- Compañía Levantina de Reductores. (2020). *Motor paso a paso*.
- Cusco, J., & Quishpe, J. (2018). Ética aplicada a la robótica. *Caribeña de Ciencias Sociales*, marzo. <https://www.eumed.net/rev/caribe/2018/03/etica-robotica.html>
- García, M. (2019). *Lógica difusa, como medio para la interpretación de incertidumbre en las organizaciones*. <https://www.gestiopolis.com/logica-difusa-interpretacion-incertidumbre-en-las-organizaciones/>
- Honda - Asimo. (n.d.). <https://mundocontact.com/honda-dice-adios-a-su-robot-asimo/>
- Jesús, D., Cristino, S., Lozano, H. M., Espino, E. R., & Ponce Ponce, V. H. (2019). Sistema de navegación y evasión de obstáculos aplicando un sistema de control difuso en una placa Arduino UNO. *Rcs.Cic.Ipn.Mx*, 148(10), 291–303.
- Mecafenix, I. (2017). *Motor paso a paso*.  
<https://www.ingmecafenix.com/electricidad-industrial/motor-paso-a-paso/>
- Mobile Industrial Robots. (n.d.). *Robot MIR 200*.
- Murray, R. M. (1994). *A Mathematical Introduction to Robotic Manipulation*.
- Nicelectric. (2020). *Nema 34 Motor paso a paso*.
- Parra, H., Hernando, L. R. G., & Bueno, M. (2007). Navegación de robots móviles mediante comportamientos utilizando lógica difusa. *Scientia et Technica Año XIII*, 34. <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/5541>
- R. Suárez. (2017). *¿Son los robots fácilmente aceptados en la sociedad?*

- Reyes, F. (2011). *Robótica-Control de Robots Manipuladores*.
- Reyes M., A., Sánchez L., A., Guevara S., F., & Toriz P., A. (2018). Planificación de movimientos para robots aéreos no tripulados. *Research in Computing Science*, 147(7), 99–113. <https://doi.org/10.13053/racs-147-7-9>
- Robot-corales*. (2011). <https://blogthinkbig.com/robot-corales>
- Robotics, A. (2005). *Sensor Ultrasonido SRF08*.  
<http://www.acroname.com/robotics/parts/R145-SRF08.html>
- Ruiz Alcántara, A. H. (2007). *Control Difuso vs Control PID*.
- Ruiz, M. C. (2015). *Resistencia de Materiales*.
- Sony AIBO ERS-210*. (n.d.). <https://www.amazon.com/-/es/ERS210-Sony-AIBO-ERS-210/dp/B00006RZ1T>
- Stepperonline. (2020). *Driver MA860H*.
- Suarez, F., Díaz, E., Velásquez, S., & Velásquez, J. (2018). *Control Híbrido Pid-Difuso En Robot Seguidor De Hybrid Pid-Diffused Control in Robot*. 22, 4–15.  
<http://uctunexpo.autanabooks.com/index.php/uct/article/view/130>
- Trilnick, C. (1961). *Unimate*. <https://proyectoidis.org/4000-pound-unimate/>
- Unten, R. (n.d.). *Dron*. <https://www.tni.org/my/node/24488>
- Vejarano Anzola, Á. A. (2014). Cine: Yo robot. *Revista Policía y Seguridad Pública*, 421–425. <https://doi.org/10.5377/rpsp.v4i2.1775>
- Venegas, C. V. (2006). *Sistemas de locomoción*. Universidad de Sevilla.  
[http://www.esi2.us.es/~vivas/ayr2iaei/LOC\\_MOV.pdf](http://www.esi2.us.es/~vivas/ayr2iaei/LOC_MOV.pdf)
- Víctor, R., Barrientos, S., José, R., García, S., Ramón, S., & Ortigoza, ♣. (n.d.). 2 polibits 2007 Robots Móviles: Evolución y Estado del Arte. In *redalyc.org*. Retrieved February 14, 2020, from <https://www.redalyc.org/pdf/4026/402640448003.pdf>
- Yandún, A. (2011). *Planeación y seguimiento de trayectorias para un robot móvil*. <http://bibdigital.epn.edu.ec/handle/15000/3772>
- Zaragoza Hernández, J., Mendoza, A. G., Jalomo, J., & Remai, C. (2017). Control

para un sistema de aire acondicionado, empleando un desarrollo electrónico embebido con aplicación de técnicas de control difuso PD+I y control PID clásico. *Remai.Ipn.Mx*, 53(9), 1689–1699.

Zollo, B. (2015). *Ventajas de simular baterías con fuentes de alimentación*. 72–73.