

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO-CAMPUS SUR

CARRERA DE INGENIERÍA ELECTRÓNICA

MENCIÓN SISTEMAS INDUSTRIALES

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN CONTROL
DE VELOCIDAD DIFUSO APLICADO AL ROBOT MÓVIL
ROBOTINO.**

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRÓNICO**

**ABRIL CARRILLO ERIKA PAOLA
PACHECO PINEDA ANDRÉS GEOVANNY**

DIRECTOR ING. RODRIGO TUFIÑO

Quito, Junio 2012.

DECLARACIÓN

Nosotros, Abril Carrillo Erika Paola y Pacheco Pineda Andrés Geovanny, declaramos bajo juramento que el presente trabajo es de nuestra autoría, no ha sido previamente presentado para algún tipo de calificación profesional y que hemos consultado las referencias bibliográficas que se incluyen en el documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual del presente trabajo a la Universidad Politécnica Salesiana, por su reglamento institucional vigente.

Abril Carrillo Erika P.

Pacheco Pineda Andrés G.

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Abril Carrillo Erika Paola y Pacheco Pineda Andrés Geovanny bajo mi dirección.

Ing. Rodrigo Tufiño.

Director de tesis.

AGRADECIMIENTOS

Un agradecimiento especial para el Ing. Rodrigo Tufiño, director del presente trabajo, por sus aportes, comentarios, críticas y sugerencias dadas durante el desarrollo de esta investigación.

De la misma manera extendemos nuestros agradecimientos al Físico Bayardo Campuzano por hacernos conocer su punto de vista con respecto a nuestro proyecto de titulación.

Al Ing. Germán Arévalo por permitirnos usar de los materiales de la carrera de ingeniería electrónica, así como también a los encargados de los laboratorios por su disponibilidad y paciencia.

Todos ustedes han hecho posible el desarrollo y la exitosa culminación de este proyecto de titulación.

AGRADECIMIENTOS

Agradezco mi madre por los ánimos que me dió durante el desarrollo de este proyecto, de la misma manera a mi compañero de tesis por su paciencia y dedicación que hicieron que no decline mi anhelo de culminar lo antes posible cumpliendo con el cronograma planteado.

Un agradecimiento especial a mi compañero de vida, que en los instantes más difíciles estuvo a mi lado dándome las fuerzas necesarias, iluminando mi camino y con los consejos apropiados en todo momento, en todo lugar.

A todos en general que con sus comentarios e intenciones hicieron de mi alguien fuerte con capacidad de cumplir lo que en un comienzo me propuse.

Erika Abril C.

AGRADECIMIENTOS

En primer lugar dar gracias a dios por iluminarme cada día y por las fuerzas dadas para culminar mis estudios.

Agradezco el amor, cariño y apoyo de mi madre Clemencia y de mi hermano Danilo que siempre han estado conmigo en las buenas y en las malas, apoyándome para culmine mi carrera con la realización de este proyecto de tesis.

A mi padre Luis y a mi tío Gustavo por su cariño incondicional, sus consejos y su apoyo en todo momento.

Andrés Pacheco P.

CONTENIDO

	Pág.
CAPÍTULO 1.....	1
INTRODUCCIÓN	1
1.1 HIPOTESIS.....	1
1.2 OBJETIVOS	1
1.2.1 Objetivo general.....	1
1.2.1 Objetivos Específicos	2
CAPÍTULO 2.....	3
2.1 LÓGICA DIFUSA	3
2.2.1 Introducción a la Lógica difusa	3
2.1.2 Historia.....	3
2.1.3 Fundamentos	5
2.1.3.1 Teoría de conjuntos difusos.....	5
2.1.3.3 Conjuntos clásicos	5
2.1.3.5 Función de pertenencia.....	6
2.1.3.6 Tipos de funciones de pertenencia	7
2.1.4 Operaciones con conjuntos difusos.....	9
2.1.5 Sistemas Difusos.....	11
2.1.5.1 Fusificación	13
2.1.5.2 Inferencia.....	14
2.1.5.3 Base del conocimiento.....	15
2.1.6 Aplicaciones actuales.....	20
2.2 ROBÓTICA	22
2.2.1 Introducción a la Robótica.....	22
2.2.2 Historia.....	23

2.2.2.1	Antecedentes Históricos	23
2.2.3	Definición y Clasificación del robot.....	27
2.2.3.1	Definición de Robot Industrial	27
2.2.3.2	Otros tipos de Robots	27
2.2.4	Robot Móvil.....	28
2.2.5	Definición	28
2.2.5.1	Generalidades.....	28
2.2.5.2	Robots Móviles Autónomos.....	28
2.2.5.3	Sistemas de locomoción para robots móviles	29
2.2.5.4	Percepción	32
2.2.5.5	Aplicaciones Actuales	32
2.3	SENSORES.....	35
2.3.1	Sensor de proximidad inductivo analógico.....	35
2.3.2	Sensores de medición de distancia por infrarrojo.....	37
2.3.3	Detectores de posición ópticos	38
2.3.3.1	Detectores ópticos con fibra óptica	39
	CAPITULO 3.....	40
3.1	ANÁLISIS	40
3.1.1	ROBOTINO.....	40
3.1.1.1	Diseño y Función.....	41
3.1.1.2	Chasis y puente de mando.....	41
3.1.2	Sensores de Robotino.....	45
3.1.2.1	Sensor inductivo de proximidad	45
3.1.2.2	Detectores ópticos con fibra óptica	47
3.1.2.3	Sensor anticolidión.....	49
3.1.2.4	Sensores de medición de distancia por infrarrojos.....	50
3.1.3	Prueba de sensores	51
3.1.3.1	Sensores Ópticos	51
3.1.3.2	Sensor Inductivo.....	51

3.2	SIMULACIÓN Y AJUSTES	51
3.2.1	Pruebas preliminares de velocidad y ángulo de giro	52
3.3	MODELO NUMÉRICO	55
3.3.1	Descripción del procedimiento	55
3.3.2	Método utilizado: Regresión lineal múltiple	56
3.3.3	Tablas obtenidas, resolución y ecuación final	57
3.3.4	Diseño del Fis	59
3.3.5	Diseño del FIS con Matlab	73
3.3.6	Simulink.....	75
	CAPITULO 4.....	80
4.1	DESARROLLO	80
4.1.1	Labview 2010	80
4.1.2	Herramientas de software utilizadas	80
4.1.2.1	Robotino Labview Driver	80
4.1.2.2	Fuzzy Logic Toolkit.....	83
4.1.2.3	Database Toolkit.....	85
4.2	IMPLEMENTACIÓN DEL PROGRAMA	86
4.2.1	Conexión entre Robotino y Labview	86
4.2.2	Implementación del controlador difuso	88
4.2.3	Pantallas	93
4.2.3.1	Pantalla presentación.....	93
4.2.3.2	Pantalla Principal.....	95
4.2.3.3	Pantalla de Históricos.....	97
4.2.3.4	Pantalla Cámara.....	98
4.2.3.5	Pantalla Estado del robot	98
4.2.3.6	Pantalla Controlador	99
4.3	ANÁLISIS Y RESULTADOS.....	101

4.3.1	Recorridos seguidos mediante un controlador convencional	101
4.3.2	Recorrido seguido usando el controlador difuso	102
4.3.3	Valores de velocidad y ángulo de giro obtenidos en el diseño del controlador en Matlab y el controlador implementado en Labview	103
4.3.4	Valores de velocidad y ángulo de giro obtenidos en la simulación.....	105
4.3.5	Comparación entre el recorrido utilizando controlador FIS Vs recorrido utilizando controlador convencional	106
4.3.5.1	Tiempos de recorrido seguido usando el controlador difuso.....	107
4.3.5.2	Tiempos de recorrido seguido usando un control convencional	108
4.3.5.3	Cálculos Estadísticos.....	110
CAPITULO 5.....		118
5.1	CONCLUSIONES	118
5.2	RECOMENDACIONES	119
BIBLIOGRAFÍA.....		121
ANEXO 1.....		126
	Procesamiento digital de imágenes	126
ANEXO 2.....		140
	Diseño en Fuzzy System Designer de Labview	140
ANEXO 3.....		146
	Conexión Robotino con Matlab	146

ABREVIATURAS

°C: Grados centígrados, unidad de medida de la temperatura.

E/S: Entradas y/o salidas.

FAM: Memorias asociativas difusas.

FIS: Fuzzy Inference System, sus siglas en español sistema de inferencia difusa.

Hz: Hertz, unidad de medida de frecuencia.

IFR: International Federation of Robotics.

ISO: Organización Internacional de Estándares.

JPEG: Formato de archivo de imágenes, Joint Photographic Experts Group.

Labview: Laboratory Virtual Instrument Engineering Workbench.

Led: Light emitting diode.

Matlab: MATrix LABoratory.

Mimo: Múltiples Entradas, múltiples salidas.

Mm: Milímetros, unidad de medida de distancia.

Mm/s: Milímetros/segundos, unidad de medida velocidad.

RGB: Rojo, Verde y Azul

RIA: Asociación de Industrias Robóticas.

USB: Universal serial bus.

V: Voltios, unidad de medida de voltaje.

VGA: Video Graphics Array.

Vi: Instrumento Virtual

ÍNDICE DE FIGURAS

	Pág.
Figura 2.1 Lógica clásica vs. Lógica difusa.....	5
Figura 2.2 Valores lingüísticos de la variable difusa "Velocidad".	6
Figura 2.3 Representación de una función de pertenencia.....	6
Figura 2.4 Función triangular.....	7
Figura 2.5 Función trapezoidal.	7
Figura 2.6 Función Gaussiana.....	8
Figura 2.7 Función Sigmoidal.....	8
Figura 2.8 Conjuntos difusos A y B.....	9
Figura 2.9 A Unión B.....	10
Figura 2.10 A Intersección B.	10
Figura 2.11 Complemento de B.	10
Figura 2.12 Estructura de un Sistema difuso.	11
Figura 2.13 Representación gráfica para el ejemplo.....	11
Figura 2.14 Conjuntos difusos para la entrada Velocidad.	12
Figura 2.15 Conjuntos difusos para la entrada Distancia.....	12
Figura 2.16 Conjuntos difusos para la salida % Freno.....	12
Figura 2.17 Regla 9.	16
Figura 2.18 Regla 10.	17
Figura 2.19 Regla 14.	17
Figura 2.20 Regla 15.	17
Figura 2.21 Suma de las cuatro reglas.	17
Figura 2.22 Representación gráfica del método del centroide.	18
Figura 2.23 Representación de la defusificación.....	19
Figura 2.24 Representación gráfica de X2.....	19
Figura 2.25 Lavadora controlada mediante lógica difusa.....	21
Figura 2.26 Sistema de ventilación implementado con lógica difusa.....	21
Figura 2.27 Pájaros de Herón.....	24
Figura 2.28 Gallo de Estrasburgo.	25
Figura 2.29 León Mecánico.	25
Figura 2.30 Pato de Vaucanson.....	26

Figura 2.31 Hiladora Giratoria.....	26
Figura 2.32 Robots Bípedos. Figura 2.33 BigDog Robot Cuadrúpedo.	30
Figura 2.34 MorpHex Robot Hexápodo.	30
Figura 2.35 Robotino Festo.....	31
Figura 2.36 Negotiator Robot.	31
Figura 2.37 Windoro, el robot limpiacristales.	33
Figura 2.38 Robot RIBA 2.....	34
Figura 2.39 ROOMBA.....	34
Figura 2.40 Robot que aprende, toma sus propias decisiones y consulta sus dudas con otros robots.	35
Figura 2.41 (a) Sensor inductivo, (b) Componentes sensor inductivo.....	36
Figura 2.42 Aplicaciones de los sensores inductivos.....	37
Figura 2.43 Principio de funcionamiento de un sensor de medición de distancia por infrarrojo.	38
Figura 2.44 Principio de funcionamiento de un detector de posición óptico.....	38
Figura 3.1 Robotino FESTO	40
Figura 3.2 Chasis Robotino.....	41
Figura 3.3 Componentes mecánicos de Robotino.....	42
Figura 3.4 Unidad de accionamiento Robotino.....	43
Figura 3.5 Unidad de Control Robotino.....	44
Figura 3.6 Interface entradas/salidas.....	45
Figura 3.7 Sensor inductivo.	46
Figura 3.8 Conexión del sensor de proximidad Inductivo en la interface de E/S.....	46
Figura 3.9 Sensor óptico.	47
Figura 3.10 Conexión del sensor óptico con fibra óptica en la interface de E/S.	48
Figura 3.11 Montaje de sensores ópticos.	49
Figura 3.12 (1) Sensores de distancia, (2) Sensor anticolidión.	50
Figura 3.13 Camino para curvas preliminares.	52
Figura 3.14 Curva abierta.....	53
Figura 3.15 Línea recta	53
Figura 3.16 Curva Tipo “s” primer tramo.....	54
Figura 3.17 Curva Tipo “S” segundo tramo	54
Figura 3.18 Curva en “U”	55
Figura 3.19 Bloques modelo matemático en Simulink.....	75

Figura 3.20 Posición actual del móvil.....	78
Figura 3.21 Salida velocidad en Simulink.	78
Figura 3.22 Salida ángulo de giro en Simulink.....	79
Figura 4.1 Librería de Robotino para Labview	81
Figura 4.2 Librería de Fuzzy Logic para Labview.....	83
Figura 4.3 Librería de Database para Labview.	85
Figura 4.4 Comunicación entre Labview y Robotino (a) Panel frontal, (b) Diagrama de bloques.	87
Figura 4.5 Conexión de los Sub-VIs AnalogInput.vi y DigitalInput.vi.....	87
Figura 4.6 Conexión de los Sub-VIs OmniDrive.vi y Motor.vi	88
Figura 4.7 Variables lingüísticas y conjuntos difusos para entradas y salidas.....	90
Figura 4.8 Reglas difusas.	90
Figura 4.9 Diagnóstico del sistema.	91
Figura 4.10 Diagrama de flujo de la implementación del programa.....	92
Figura 4.11 Secuencia de despliegue de las pantallas.....	93
Figura 4.12 Pantalla Presentación.	94
Figura 4.13 Secuencia de operación de la pantalla presentación.	94
Figura 4.14 Pantalla principal.	95
Figura 4.15 Secuencia de operación para la pantalla principal.....	96
Figura 4.16 Pantalla de históricos.	97
Figura 4.17 Secuencia de operación para la pantalla de históricos.....	97
Figura 4.18 Pantalla cámara.	98
Figura 4.19 Secuencia de operación para la pantalla Cámara.....	98
Figura 4.20 Pantalla estado del robot.....	99
Figura 4.21 Secuencia de operación para la pantalla Estado del robot.....	99
Figura 4.22 Pantalla Controlador.	100
Figura 4.23 Secuencia de operación para la pantalla Controlador.....	101
Figura 4.24 Pista diseñada para pruebas de recorrido.....	107

INDICE DE TABLAS

	Pág.
Tabla 2.1 Componentes de una memoria asociativa difusa.....	15
Tabla 2.2 Memoria asociativa difusa para el ejemplo dado.....	16
Tabla 3.1 Descripción de componentes mecánicos de Robotino.....	42
Tabla 3.2 Conexión para el sensor de proximidad inductivo.....	46
Tabla 3.3 Datos técnicos relevantes del sensor de proximidad inductivo.....	47
Tabla 3.4 Conexión para el sensor óptico.....	48
Tabla 3.5 Datos técnicos relevantes del detector óptico con fibra óptica.....	49
Tabla 3.6 Estado de los sensores ante diferentes colores.....	51
Tabla 3.7 Sensor inductivo ante la cinta de aluminio a 2 mm del suelo.....	51
Tabla 3.8 Pruebas preliminares de velocidad y ángulo de giro.....	52
Tabla 3.9 Datos tomados del funcionamiento del sistema.....	58
Tabla 3.10 Sumatorias de las variables del sistema.....	58
Tabla 3.11 Cálculos realizados con las sumatorias obtenidas.....	59
Tabla 3.12 Diseño del Fis con Matlab.....	74
Tabla 3.13 Bloques a configurar en Simulink.....	77
Tabla 4.1 Descripción componentes de Robotino Labview driver.....	83
Tabla 4.2 Descripción componentes de Fuzzy Logic Toolkit.....	84
Tabla 4.3 Descripción componentes de Database Toolkit.....	86
Tabla 4.4 Fidelidad de seguimiento del móvil a distintos valores de velocidad y ángulo de giro.....	102
Tabla 4.5 Recorrido seguido usando el controlador difuso.....	103
Tabla 4.6 Valores tomados del diseño del controlador en Matlab.....	104
Tabla 4.7 Valores tomados del diseño del controlador en Labview.....	105
Tabla 4.10 Datos obtenidos de la simulación.....	106
Tabla 4.13 Tiempo de recorrido con control difuso.....	107
Tabla 4.14 Tiempo de recorrido con velocidad de 280 mm/s y ángulo de $\pm 20^\circ$ usando controlador convencional.....	108
Tabla 4.15 Tiempo de recorrido con velocidad de 350 mm/s y ángulo de $\pm 20^\circ$ usando controlador convencional.....	109
Tabla 4.16 Eficiencia del controlador FIS en comparación a un controlador convencional.....	110

Tabla 4.17 Muestras tomadas con el controlador implementado en Labview.....	111
Tabla 4.18 Muestras tomadas con un control convencional de 280 mm/s de velocidad y un ángulo de $\pm 20^\circ$	112
Tabla 4.19 Muestras tomadas con un control convencional de 350 mm/s de velocidad y un ángulo de $\pm 20^\circ$	114
Tabla 4.20 Resultados de la desviación estándar.....	116
Tabla 4.21 Comparación Control difuso vs. Control convencional.....	117

RESUMEN

La Robótica ha venido transformando la vida del ser humano desde sus inicios no solo en el ámbito de la industria sino también se encuentra presente en las diversas tareas cotidianas, facilitando la vida al ser humano y volviendo más eficientes las tareas que antes se las realizaba sin ayuda de robots.

Los avances de la robótica son dignos de seguimiento, para poder no solo aprender que es lo que está pasando en la actualidad sino que también se pueda aportar con dichos avances, esta es una de las motivaciones principales que lleva a la realización del presente proyecto de tesis, haciendo uso del robot móvil Robotino de FESTO.

En nuestro país, son muy conocidos los concursos que se realizan de robots seguidores de línea, en los cuales, se evalúa la capacidad que tiene el estudiante de dotar al robot de eficiencia en el objetivo que se plantee, ya sea éste; fidelidad del seguimiento de línea en el transcurso de la competencia o velocidad con la cual pueda culminar el recorrido.

Este proyecto de titulación, incorpora al robot móvil un sistema de control por lógica difusa, el mismo que será implementado al robot mediante el software Labview, así mismo se demuestra mediante un modelo numérico la eficacia del nuevo sistema de control implementado.

En este proyecto de titulación están aplicados todos los conocimientos adquiridos en lo referente a la lógica difusa, la robótica, el manejo del software Labview, y el uso de la herramienta matemática Matlab.

PRESENTACIÓN

El presente proyecto de titulación está orientado para las personas que se encuentren interesadas en uso y control del robot móvil Robotino de Festo, específicamente un control basado en lógica difusa, utilizando el lenguaje de programación Labview , con el fin de que este recorra el trayecto planteado en el menor tiempo posible resultando fácil de aplicar y fácil de entender.

En el Capítulo1 se describe de manera general la introducción, los objetivos y el planteamiento del problema del presente proyecto de titulación.

En el capítulo 2 se presenta una revisión sobre lógica difusa, la robótica y una breve explicación de sensores. La parte de lógica difusa cuenta con una explicación completa y detallada además de un ejemplo para una mejor comprensión. En robótica se revisan un poco de historia, clasificación, aplicaciones actuales y la robótica móvil y por último en la parte de sensores una breve revisión de los sensores que forman parte del conjunto de Robotino.

El capítulo 3 cuenta con el análisis de Robotino y sus componentes, pruebas preliminares de los sensores utilizados y el diseño del controlador difuso en Matlab y su respectiva simulación en Simulink.

En el capítulo 4 está detallada la implementación del controlador en Labview, la descripción de las pantallas que constan en el programa así como todas las pruebas realizadas del sistema con sus resultados.

Y finalmente el Capítulo 5 consta de las respectivas conclusiones y recomendaciones dadas en el presente trabajo.

CAPÍTULO 1

INTRODUCCIÓN

Esta investigación es necesaria para todo aquel que esté interesado en trabajar con un robot móvil sea estudiante o docente; porque gracias a ésta se podrá determinar en un futuro, si al implementar un control difuso de velocidad se puede o no tener alguna ventaja frente a un robot móvil que no lo esté usando.

Asimismo y complementariamente sería conveniente su lectura y aplicación, para estudiantes vinculados a concursos de robótica móvil; porque de ser nuestra investigación satisfactoria, dichos estudiantes podrán aplicar el control difuso de velocidad y obtener mejores resultados en la competencia.

El motivo que nos lleva a realizar este trabajo es el de dotar de nuevas herramientas de trabajo a un robot móvil mediante un control difuso de velocidad, mismo que va a ser llevado a cabo con un software que no es propio de dicho robot y de esta forma probar si el robot puede soportar tanto a nivel de su hardware y software el uso de un medio distinto para su programación y control.

1.1 HIPOTESIS

¿La implementación de un controlador difuso de velocidad, en un robot móvil seguidor de línea, garantiza una ventaja con respecto a un robot móvil sin controlador?

1.2 OBJETIVOS

1.2.1 Objetivo general

Analizar, Diseñar e Implementar un controlador difuso de velocidad en el robot móvil Robotino de Festo.

1.2.1 Objetivos Específicos

- Investigar sobre el estado del arte de la robótica móvil y los controladores difusos.
- Revisar la documentación técnica del hardware y software a utilizar.
- Realizar un análisis para parametrizar las señales de entrada del controlador difuso.
- Diseñar y probar a través de un modelo numérico el funcionamiento de un controlador difuso de velocidad.
- Construir una interface gráfica fácil e intuitiva acorde al sistema.
- Realizar las pruebas respectivas con el Robot Móvil.

CAPÍTULO 2

2.1 LÓGICA DIFUSA

2.2.1 Introducción a la Lógica difusa

La lógica difusa es una lógica alternativa a lo que es la lógica clásica, ya que esta permite trabajar con información la cual no es exacta, lo cual no ocurre con la lógica clásica ya que esta trabaja con información definida y precisa.

Esta lógica fue implementada con el fin de emular en cierta forma el comportamiento humano ya que el razonamiento del ser humano se relaciona con este tipo de información imprecisa, un ejemplo de este tipo de razonamiento en el ser humano es cuando se trata de describir la estatura de una persona, se acostumbra a decir alto, mediano o pequeño pero estas descripciones no están sujetas a un valor exacto, sino que más bien se trata de asociarlas a un rango de valores.

2.1.2 Historia

La lógica difusa es algo reciente con muy poco tiempo de trabajo, pero su origen se data de los tiempos de los filósofos Aristóteles y Platón, las Leyes del Pensamiento de Aristóteles permitieron el desarrollo de la lógica y de las matemáticas. “Tanto Aristóteles como Platón son los primeros en considerar que las cosas no tienen por qué ser de un cierto tipo o dejar de serlo, sino que hay un rango entre sus extremos, considerando que existen diferentes grados de verdad y falsedad”¹.

En el siglo XVIII aparece el filósofo David Hume quien habla sobre la lógica del sentido común, concluyendo en que el razonamiento se lo adquiere de las vivencias a lo largo de nuestras vidas.

Entre los años de 1917 a 1920 Jan Lukasiewicz, desarrolló la primera lógica de vaguedades. Para él los conjuntos tienen un posible grado de pertenencia con valores que oscilan entre 0 y 1, y en este intervalo existen un número infinito de valores.

¹ Historia de la lógica borrosa, disponible en el ítem contenido de [Web_01].

Para el año de 1965 se hizo por primera vez la presentación de un trabajo sobre conjuntos difusos, fue Lofti Asier Zadeh quien publicó su trabajo denominado "Fuzzy Sets" (Conjuntos Difusos), de esta forma se lo llamo a Zadeh el padre del termino borroso.

Si bien en un principio lo dicho por Zadeh no fue considerado por la comunidad científica, luego de varios años fue consiguiendo adeptos con lo que sus teorías se fueron ampliando y sobre todo asentando, es así como varios años después otros científicos comenzaron a aplicar la teoría de la lógica difusa en diversos proyectos.

En 1974 Assilian y Mamdani en el Reino Unido desarrollaron el primer controlador difuso diseñado para la máquina de vapor, creando así un hito en el desarrollo de la lógica difusa. Pero la primera implantación real de un controlador de este tipo no fue realizada hasta 1980 por F.L. Smidth & Co. en una planta cementera en Dinamarca.

En 1987 Hitachi usa un controlador fuzzy para el control del tren de Sendai, el cual usa uno de los sistemas más novedosos creados por el hombre. Desde entonces, el controlador ha realizado su trabajo correctamente con la consiguiente satisfacción por parte de los usuarios de dicho tren.

En 1993, Fuji aplica la Lógica Borrosa para el control de inyección química en plantas depuradoras de agua por primera vez en Japón. Ha sido precisamente aquí, en donde más apogeo ha tenido la Lógica Borrosa, creándose estrechas colaboraciones entre el gobierno, las universidades y las industrias, estableciendo proyectos llevados a cabo por el Ministerio de Industria y Comercio (MITI) y la Agencia de Ciencia y Tecnología (STA) en consorcio con el Laboratory for International Fuzzy Engineering Research (LIFE).

Otro factor decisivo para continuar con la investigación de este campo es el interés en las redes neuronales y su relación con los sistemas fuzzy. Se buscan fusiones entre las dos técnicas obteniéndose como resultado los sistemas neuro-fuzzy, que usan métodos de aprendizaje basados en redes neuronales para identificar además de la lógica difusa para optimizar sus parámetros.

2.1.3 Fundamentos

2.1.3.1 Teoría de conjuntos difusos

La lógica difusa nos permite trabajar con información aproximada o la cual tiene un alto grado de imprecisión, mientras que la lógica clásica trabaja con información precisa, decimos que los conjuntos difusos parten de la teoría clásica de conjuntos dicho esto surge la necesidad de definir los conjuntos difusos como los conjuntos clásicos.

2.1.3.2 Conjuntos difusos

Un Conjunto Difuso es aquel que puede contener elementos con grados parciales de pertenencia, partiendo de la teoría clásica de conjuntos, definiendo al grado de pertenencia como un número real entre cero y uno.

2.1.3.3 Conjuntos clásicos

Los conjuntos clásicos no son más que una colección de elementos, que surgen de la necesidad del ser humano de clasificar objetos y conceptos.

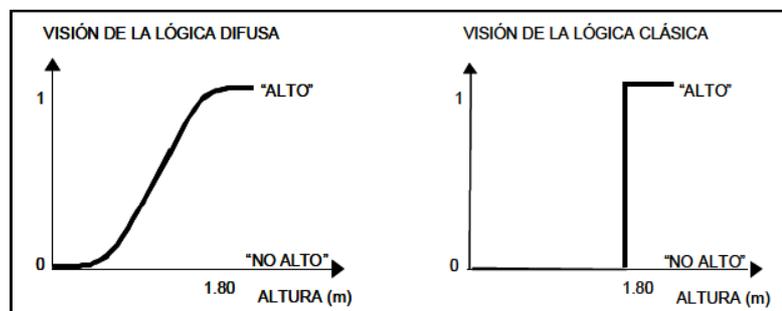


Figura 2.1 Lógica clásica vs. Lógica difusa.

Fuente: [Web_02].

2.1.3.4 Variables lingüística

Una variable lingüística como su nombre lo sugiere, es una variable cuyos valores son palabras o sentencias en un lenguaje natural. Por ejemplo, la velocidad de un coche, "Velocidad" es una variable lingüística si sus valores son "muy lenta", "lenta", "normal", "rápido" y "muy rápido".

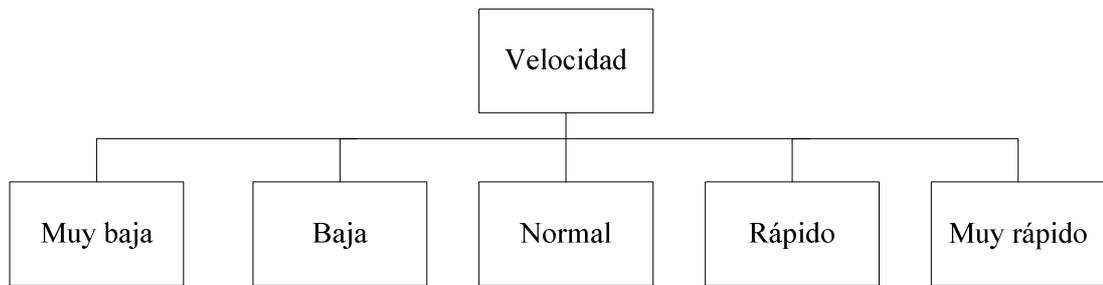


Figura 2.2 Valores lingüísticos de la variable difusa "Velocidad".

Fuente: [Los autores].

2.1.3.5 Función de pertenencia

“La función de pertenencia en un conjunto nos indica el grado en que cada elemento de un universo dado, pertenece a dicho conjunto”².

Es decir, la función de pertenencia de un conjunto A sobre un universo X será de la forma: $\mu_A: X \rightarrow [0,1]$, donde $\mu_A(x) = r$, si r es el grado en que x pertenece a A.

Las funciones de pertenencia son una forma de representar gráficamente un conjunto borroso sobre un universo.

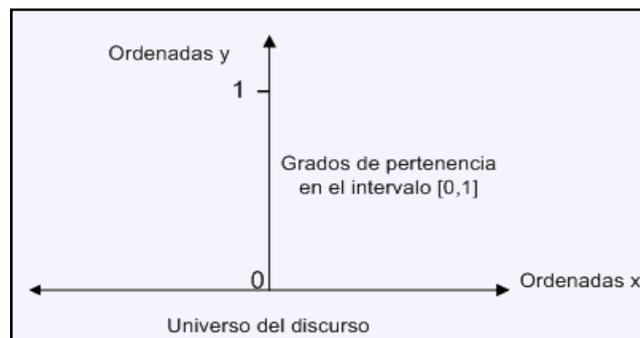


Figura 2.3 Representación de una función de pertenencia.

Fuente: [Web_01].

La forma de la función característica utilizada depende del criterio aplicado para la resolución del problema, el universo y del punto de vista del experto o usuario.

La condición que debe cumplir una función característica es que tiene que tomar valores entre 0 y 1.

²Funciones de pertenencia, más información en el ítem contenido de [Web_01].

Las funciones de pertenencia más utilizadas por su sencillez en los cálculos matemáticos son particularmente las triangulares y trapezoidales.

2.1.3.6 Tipos de funciones de pertenencia

- **Función triangular**

Se encuentra definida mediante el límite inferior a , el superior b y el valor modal m , tal que $a < m < b$. La función no tiene porqué ser simétrica.

$$U_A(x) \begin{cases} 0 & \text{si } x < a \\ \frac{x-a}{m-a} & \text{si } a \leq x < m \\ \frac{b-x}{b-m} & \text{si } m \leq x < b \\ 0 & \text{si } x \geq b \end{cases}$$

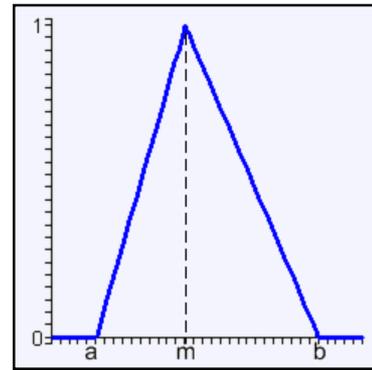


Figura 2.4 Función triangular.

Fuente: [Web_01].

- **Función Trapezoidal**

Definida por sus límites inferior a , superior d , y los límites de soporte inferior b y superior c , tal que $a < b < c < d$.

$$U_A(x) \begin{cases} 0 & \text{si } (x < a) \text{ ó } (x > d) \\ \frac{x-a}{b-a} & \text{si } a \leq x < b \\ 1 & \text{si } b \leq x < c \\ \frac{b-x}{b-m} & \text{si } c \leq x < d \end{cases}$$

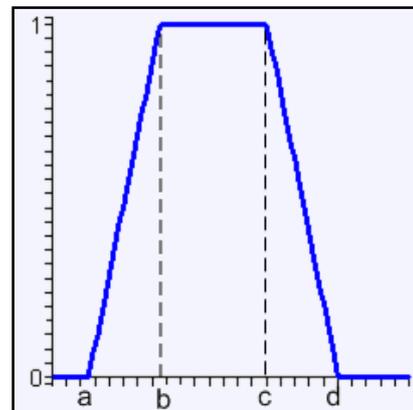


Figura 2.5 Función trapezoidal.

Fuente: [Web_01]

- **Función Gaussiana**

Definida por su valor medio m y el parámetro $k < 0$. Esta función es la típica campana de Gauss y cuanto mayor es el valor de k , más estrecha es dicha campana.

$$U_A(x) = e^{-k(x-m)^2}$$

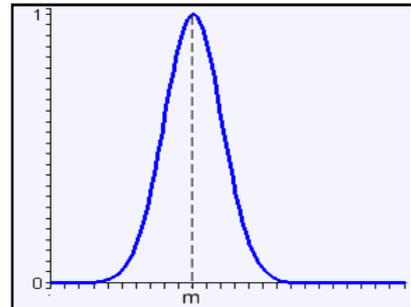


Figura 2.6 Función Gaussiana.

Fuente: [Web_01].

- **Función Sigmoidal**

Definida por sus límites inferior a , superior b y el valor m ó punto de inflexión, tales que $a < m < b$. Mientras mayor sea la distancia $a-b$ más lento es el crecimiento, el caso mas usual de esta función es cuando $m = (a+b)/2$ como se muestra en la gráfica.

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

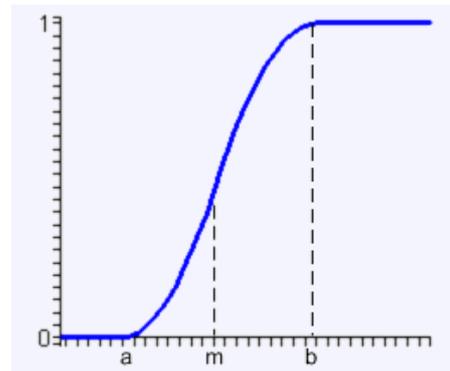


Figura 2.7 Función Sigmoidal.

Fuente: [Web_01].

Se mencionan las principales funciones de pertenencia, pero además de estas existen varias más como la función unitaria [Del Brío], función Pseudo-Exponencial, función gamma [Web_01].

2.1.4 Operaciones con conjuntos difusos

Los Conjuntos difusos operan entre sí del mismo modo que los conjuntos clásicos. Las operaciones para los conjuntos borrosos se definen de igual forma que se realizan las operaciones en los conjuntos clásicos, por lo que se definen las operaciones unión, intersección y complemento.

- Unión de conjuntos: $U_{A \cup B}(x) = \text{Max} [U_A(x), U_B(x)]$
(Ver figura 2.10)
- Intersección de conjuntos: $U_{A \cap B}(x) = \text{Min} [U_A(x), U_B(x)]$
(Ver figura 2.11)
- Complemento de un conjunto: $U_{cA}(x) = 1 - U_A(x)$
(Ver figura 2.12)

La teoría clásica de conjuntos puede considerarse como un caso particular de la borrosa.

Ejemplo:

Para los conjuntos difusos A y B; realizar las siguientes operaciones:

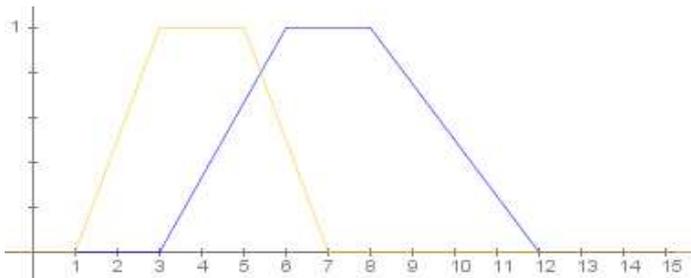


Figura 2.8 Conjuntos difusos A y B.

Fuente: [Los autores].

a) $A \cup B$

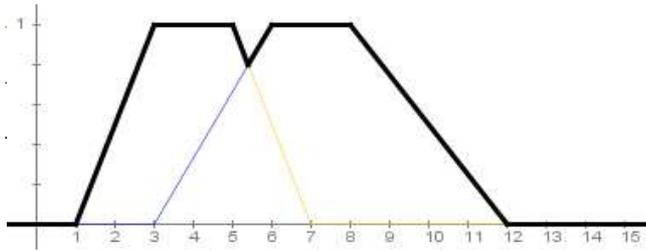


Figura 2.9 A Unión B.

Fuente: [Los autores].

b) $A \cap B$

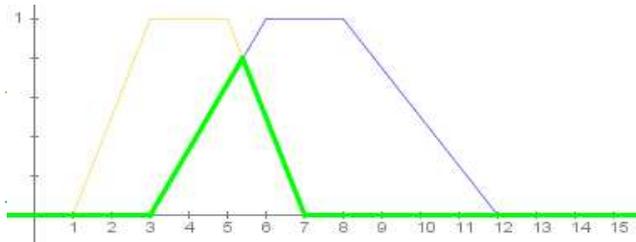


Figura 2.10 A Intersección B.

Fuente: [Los autores].

c) B complemento

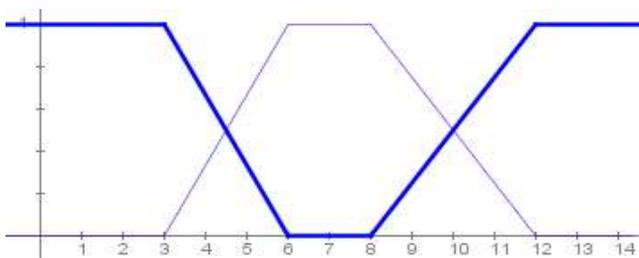


Figura 2.11 Complemento de B.

Fuente: [Los autores].

2.1.5 Sistemas Difusos

A un sistema difuso también se lo conoce como sistema de inferencia difusa (FIS), es una forma de transformar un espacio de entradas en un espacio de salidas utilizando lógica difusa. La estructura de un sistema difuso es la siguiente.

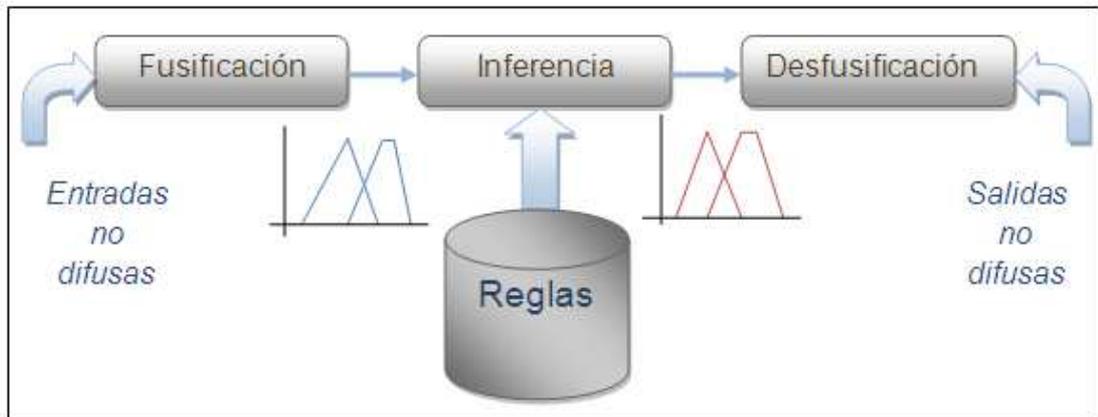


Figura 2.12 Estructura de un Sistema difuso.

Fuente: [Los autores].

Para realizar una explicación más clara de cómo funciona cada parte de un sistema se implementa un ejemplo, mismo que va a ser desarrollado a la par de cada parte que interviene en el sistema difuso.

Ejemplo Sistema difuso: Se tiene dos entradas para el sistema la velocidad (Km/h) y la distancia (m) y una salida freno (%). Implementar un sistema difuso para:

- Velocidad = 83 km/h
- Distancia = 7 m.
- % Freno =?

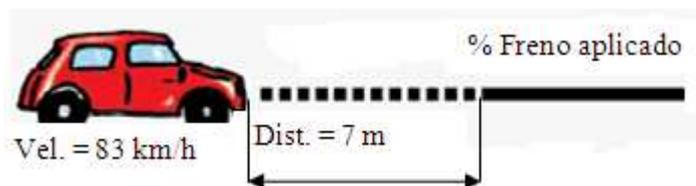


Figura 2.13 Representación gráfica para el ejemplo.

Fuente: [Los autores].

Entrada 1: Velocidad.

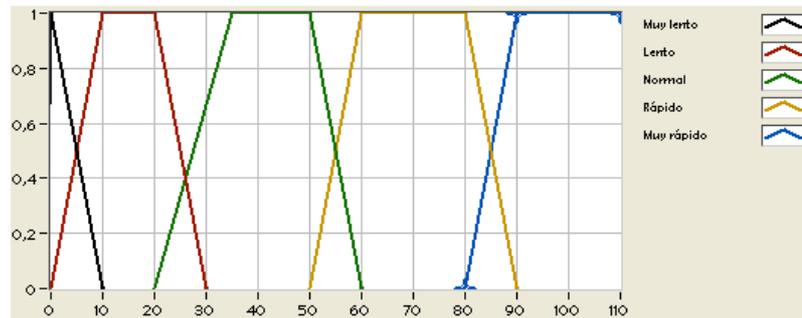


Figura 2.14 Conjuntos difusos para la entrada Velocidad.

Fuente: [Labview 2010].

Entrada 2: Distancia.

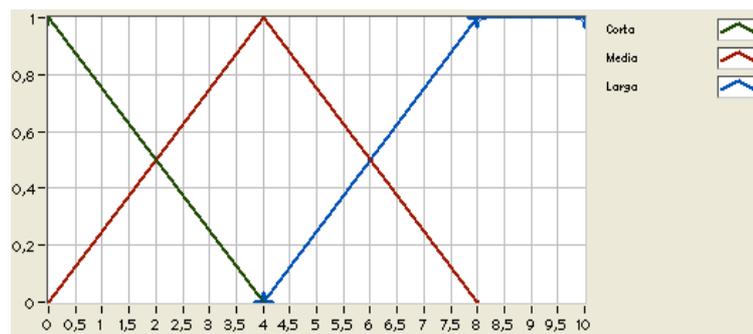


Figura 2.15 Conjuntos difusos para la entrada Distancia.

Fuente: [Labview 2010].

Salida: % Freno.

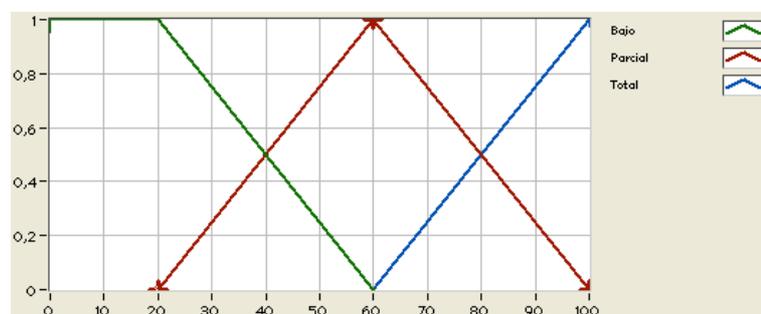


Figura 2.16 Conjuntos difusos para la salida % Freno.

Fuente: [Labview 2010].

2.1.5.1 Fusificación

Establece una relación entre los puntos de entrada no difusos y sus correspondientes conjuntos difusos, en otras palabras convierten valores reales en valores difusos.

En la fusificación se asigna un determinado grado de pertenencia a los puntos de entrada con relación a los conjuntos difusos que se han determinado para el sistema.

Continuación Ejemplo Sistema difuso

- Fusificación de la entrada velocidad.

Definimos en cual o cuales de las funciones de pertenencia se encuentra el valor de 83 km/h.

$$U_{rápido}(V): \quad \frac{V}{10} - 5 \quad 50 < V \leq 60$$

$$160 < V \leq 80$$

$$9 - \frac{V}{10} \quad 80 < V \leq 90$$

0 Otros casos.

$$U_{muyrápido}(V): \quad \frac{V}{10} - 8 \quad 80 < V \leq 90$$

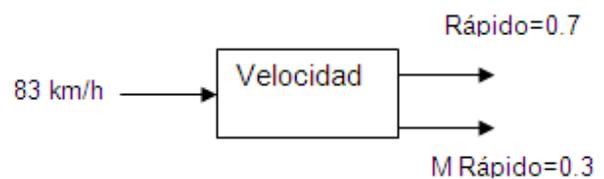
$$190 < V \leq 110$$

0 Otros casos.

Fusificación para: Velocidad = 83 km/h.

$$U_{rápido}(83): \quad 9 - \frac{83}{10} = 0.7$$

$$U_{muyrápido}(83): \quad \frac{83}{10} - 8 = 0.3$$



- Fusificación de la entrada distancia.

Definimos en cual o cuales de las funciones de pertenencia se encuentra el valor de 7 m.

$$U_{Medio}(D): \quad \frac{D}{4} \quad 0 < D \leq 4$$

$$2 - \frac{D}{4} \quad 4 < D \leq 8$$

$$0 \quad \text{Otros casos.}$$

$$U_{Largo}(D): \quad \frac{D}{4} - 1 \quad 4 < D \leq 8$$

$$18 < V \leq 10$$

Fusificación para: Distancia = 7 m.

$$U_{Medio}(D): \quad 2 - \frac{7}{4} = 0.25$$

$$U_{Largo}(D): \quad \frac{7}{4} - 1 = 0.75$$



2.1.5.2 Inferencia

Se llaman dispositivos de inferencia borrosa a los sistemas que interpretan las reglas de tipo IF – THEN de una base de reglas, con el fin de obtener los valores de salida.

Este tipo de reglas se definen como:

If< antecedente >Then< consecuente >

Estas reglas difusas combinan uno o más conjuntos difusos de entrada, llamados antecedentes o premisas y les asocian a un conjunto difuso de salida llamado consecuente o consecuencia. Los antecedentes se asocian mediante operadores lógicos tales como: Y & Ó.

Ejemplo: “Si error es positivo_pequeño Y Derivada_error es negativa_pequeña entonces acción es positiva_pequeña”.

Dónde:

Error y derivada_error → Entradas.
 Y → Conector lógico.
 Acción → Salida.

2.1.5.3 Base del conocimiento

Todas las reglas difusas se agrupan y forman la base del conocimiento, ya que éstas expresan el conocimiento del problema a tratar en base a la relación que existe entre los antecedentes y los consecuentes.

2.1.5.4 F.A.M (Memorias asociativas difusas)

Esta matriz nos permite crear las reglas que asocian los antecedentes con los consecuentes, tiene la siguiente estructura:

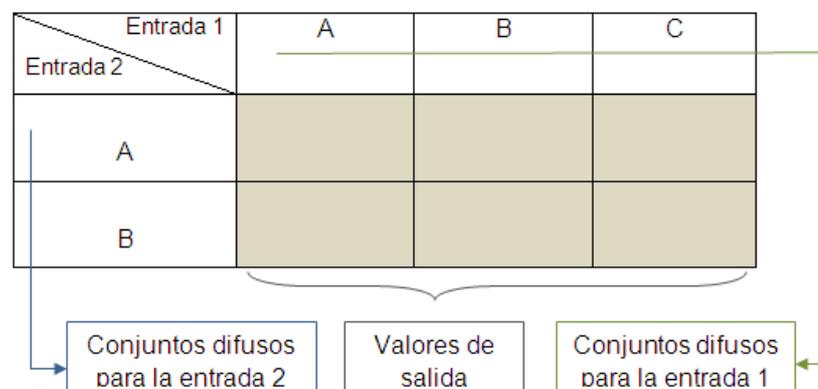


Tabla 2.1 Componentes de una memoria asociativa difusa.

Fuente: [Los autores].

Continuación ejemplo sistema difuso

- Inferencia:

Vel. \ Dist.	Muy lento	Lento	Normal	Rápido	Muy rápido
Corta	Bajo	Bajo	Parcial	Total	Total
Media	Bajo	Bajo	Parcial	Parcial (9)	Total (10)
Larga	Bajo	Bajo	Bajo	Parcial (14)	Parcial (15)

Tabla 2.2 Memoria asociativa difusa para el ejemplo dado.

Fuente: [Los autores].

Regla 9: IF Velocidad es Rápido AND distancia es Media THEN Freno es parcial.

Regla 10: IF Velocidad es Muy rápido AND distancia es Media THEN Freno es total.

Regla 14: IF Velocidad es Rápido AND distancia es Larga THEN Freno es parcial.

Regla 15: IF Velocidad es Muy rápido AND distancia es Larga THEN Freno es parcial.

- Inferencia aplicando Intersección (Mínimo).

Regla 9: If Min [V (rápido), D (media)] Then F (parcial)

$$\text{Min } [0.7, 0.25] = 0.25$$

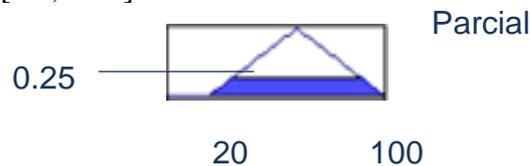


Figura 2.17 Regla 9.

Fuente: [Los autores].

Regla 10: If Min [V (Muy rápido), D (media)] Then F (total).

$$\text{Min } [0.3, 0.25] = 0.25$$

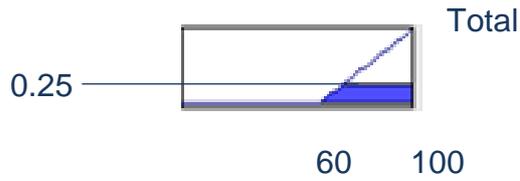


Figura 2.18 Regla 10.

Fuente: [Los autores].

Regla 14: If Min [V (rápido), D (larga)] Then F (parcial).

$$\text{Min} [0.7, 0.75] = 0.7$$

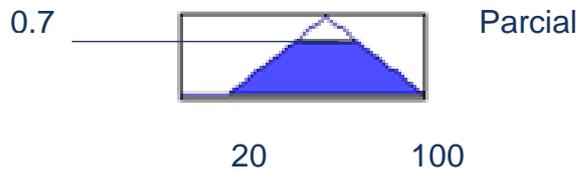


Figura 2.19 Regla 14.

Fuente: [Los autores].

Regla 15: If Min [V (Muy rápido), D (larga)] Then F (total).

$$\text{Min} [0.3, 0.75] = 0.3$$

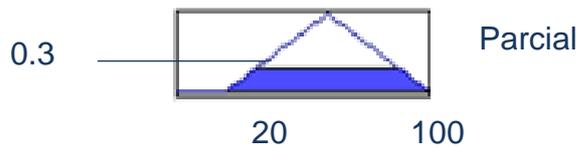


Figura 2.20 Regla 15.

Fuente: [Los autores].

Sumatoria de las reglas para la velocidad de 83 km/h y la distancia de 7 m.

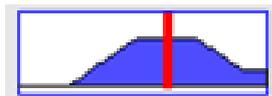


Figura 2.21 Suma de las cuatro reglas.

Fuente: [Los autores].

2.1.5.5 Desfusificación

Ante cualquier problema la solución a obtener debe de ser un número y no un conjunto difuso por lo tanto se debe de realizar una transformación, es decir transformar al conjunto difuso obtenido en la inferencia.

Uno de los métodos más utilizados es el método del centroide, el cual se encarga de calcular el centro del área definido por el conjunto difuso obtenido en la inferencia como se mencionó antes.

“El método más popular de desfusificación es el cálculo del centro promedio, el cual retorna el centro del área bajo la curva”³.

De acuerdo a este método el valor de la salida del conjunto difuso se lo obtiene mediante la siguiente ecuación:

$$g = \frac{\sum_{i=1}^n X_i * U(X_i)}{\sum_{i=1}^n U(X_i)}$$

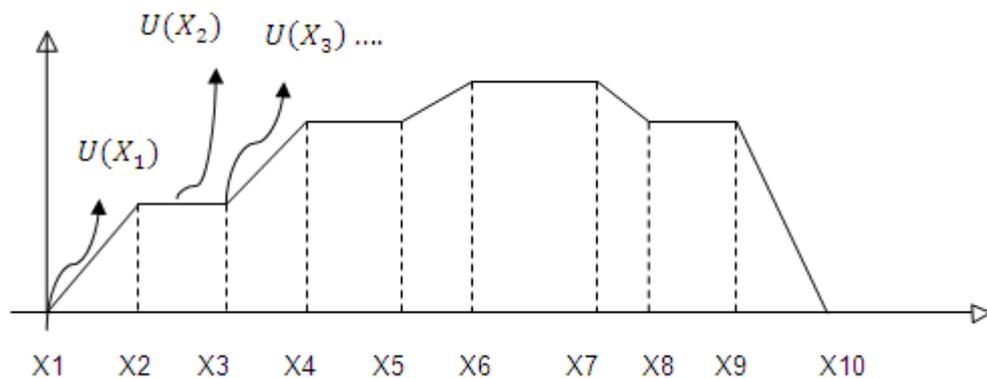


Figura 2.22 Representación gráfica del método del centroide.

Fuente: [Los autores].

³ Sistema difuso, disponible en <http://medicpinos0.tripod.com/logicafussy/id19.html>.

Continuación ejemplo sistema difuso

- Defusificación:

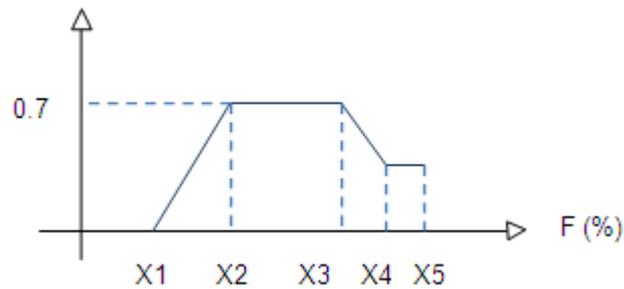


Figura 2.23 Representación de la defusificación.

Fuente: [Los autores].

Para obtener X2, X3 y X4 se aplica geometría analítica.

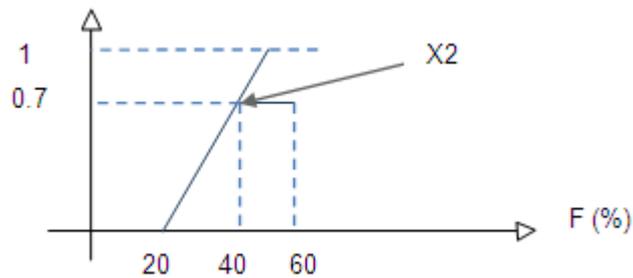


Figura 2.24 Representación gráfica de X2.

Fuente: [Los autores].

Para X2:

$$Y = \frac{y_2 - y_1}{x_2 - x_1} * (x - x_1)$$

$$Y = \frac{1 - 0}{60 - 20} * (x - 20)$$

$$0.7 = \frac{1}{40} * (x - 20)$$

$$28 = x - 20$$

$$x = 48$$

$$x_2 = 48$$

$$x_3 = 72$$

$$x_4 = 95$$

Nota: Se tiene que seguir un procedimiento similar para obtener X3 y X4.

$$X_1 = 0 \quad U_{(X1)} = 0$$

$$X_1 = 48 \quad U_{(X1)} = 0.7$$

$$X_1 = 72 \quad U_{(X1)} = 0.7$$

$$X_1 = 95 \quad U_{(X1)} = 0.25$$

$$X_1 = 100 \quad U_{(X1)} = 0.25$$

$$g = \frac{(0 * 0) + (48 * 0.7) + (72 * 0.7) + (95 * 0.25) + (100 * 0.25)}{0 + 0.7 + 0.7 + 0.25 + 0.25}$$

$$g = \frac{132.75}{1.90}$$

$$g = 69.86$$

$$g = 69.86 \% \text{ de freno}$$

2.1.6 Aplicaciones actuales

Dentro de las aplicaciones donde la lógica difusa ha intervenido en los últimos años están el manejo de distintos tipos de instrumentos, máquinas y en diversos ámbitos de la vida cotidiana. Algunos casos por ejemplo son los estabilizadores de imágenes en grabadoras de vídeo, controladores de ascensores e ingeniería de terremotos.

Si a cada una de las aplicaciones de la lógica difusa se las agrupa en tres grandes grupos tendríamos lo siguiente:

- Productos creados para el consumidor: Lavadoras difusas, hornos microondas, sistemas térmicos, traductores lingüísticos, cámaras de vídeo, televisores, estabilizadores de imágenes digitales y sistemas de foco automático en cámaras fotográficas.



Figura 2.25 Lavadora controlada mediante lógica difusa.

Fuente: [Web_11].

- Sistemas: Elevadores, trenes, automóviles, controles de tráfico, sistemas de control de acondicionadores de aire que evitan las oscilaciones de temperatura y sistemas de reconocimiento de escritura.
- En las actividades domésticas cotidianas se inventó un sistema de ventilación. Utiliza control difuso para conmutar un ventilador según los conocimientos de cantidad de polvo, olores, temperatura y humedad ambiente.



Figura 2.26 Sistema de ventilación implementado con lógica difusa.

Fuente: [Web_13].

- Software: Diagnóstico médico, seguridad, comprensión de datos, tecnología informática y bases de datos difusas para almacenar y consultar información imprecisa
- La Lógica difusa se incluyó en ascensores para reducir el tiempo de espera.
- “A partir de 1990 se la comienza a implementar en los controles de inyección electrónica de carburante y en los sistemas de control de guiado automático de coches, haciendo los controles complejos más eficientes y fáciles de utilizar”⁴.

2.2 ROBÓTICA

2.2.1 Introducción a la Robótica

Robot. Esta palabra viene del checo “robota” (siervo, trabajo forzado.) En la obra “R.U.R.” Rossum’s Universal Robots, el escritor Karel Capek en 1921 introdujo la palabra robot y la popularizó, debido al contenido de la misma, en el cual se deshumanizaba a las personas debido al maquinismo.⁷

Como definición del término robot se encuentra una amplia gama de conceptos, entre los cuales no se ha podido universalizar uno solo. Cada quien tiene su propio concepto de robot, para algunos será una máquina que imita personas en apariencia, comportamiento o acciones, para otros quizás solo es una máquina que opera automáticamente. Un robot es difícil de definir pero esto no es un problema teniendo en cuenta la idea de Joseph Engelberg.

Quizás, Joseph Engelberg (padre de la robótica industrial) lo resumió inmejorablemente cuando dijo: "Puede que no sea capaz de definirlo, pero sé cuándo veo uno"⁵.

Los primeros robots se remontan a la mitología griega y a obras de ficción. Eran en gran parte ideas, para algunos imposibles, para otros dignas de investigación. Gracias

⁴ Aplicaciones de lógica difusa, disponible en <http://www.dei.uc.edu.py/tai2000/logica/4.htm>.

⁵ Valiosa información sobre la robótica, disponible en, [Web_06].

aquellas personas que creyeron que no era del todo imposible la creación de robots, hoy en día tenemos una gama interesante de robots al servicio de la humanidad.

Si se toma en cuenta las clases de robots que existen actualmente en nuestro entorno, y la funcionalidad de los mismos, es muy fácil pensar que llegaran a formaran parte indispensable para nosotros los seres humanos, y que nuestras funciones productivas se cumplirán únicamente con ayuda de ellos.

Hoy en día ni los animales se salvan de la robótica. Partiendo de esto, es necesario nombrar el hecho de que ya existen robots que son capaces de actuar con los animales.

“Creemos que podremos construir máquinas o robots que puedan comunicarse con los animales y tomar decisiones que influyan en su conducta”⁶.

La Robótica ha llegado a dar un cambio realmente inesperado a la sociedad. Por ello es fundamental tener en cuenta que es una ciencia que no dejará de crecer, los avances que se están dando son inmensos, no solo con la robótica industrial sino también con la robótica móvil.

2.2.2 Historia

Existe un inmenso grupo de personas que están ligadas a la expansión de la robótica y su mejoramiento, desde los grandes pensadores de la historia, pasando por aficionados a la robótica y la tecnología, hasta llegar a las grandes compañías creadoras de robots. La robótica en sí, ha estado presente desde siempre, nació como una idea de aligerar la vida al ser humano y hoy en día está hasta para la diversión.

2.2.2.1 Antecedentes Históricos

A lo largo de toda la historia, el hombre se ha sentido fascinado por máquinas y dispositivos capaces de imitar las funciones y los movimientos de los seres vivos.

⁶Artículo “Nacen las roboto-rachas”, Dr. José Halloy, Universidad Libre de Bruselas, [Web_04].

Los griegos tenían una palabra específica para denominar a estas máquinas: *automatos*. De esta palabra deriva la actual *automata*: máquina que imita la figura y movimientos de un ser animado.

Los mecanismos animados de Herón de Alejandría (85 d.C.) se movían a través de dispositivos hidráulicos, poleas y palancas y tenían fines eminentemente lúdicos.⁷

En el año 62 Herón de Alejandría describe múltiples aparatos en su libro "Automata". Entre ellos aves que vuelan, gorjean y beben. Todos ellos fueron diseñados como juguetes, sin mayor interés por encontrarles aplicación. Sin embargo, describe algunos como un molino de viento para accionar un órgano o un precursor de la turbina de vapor.⁸

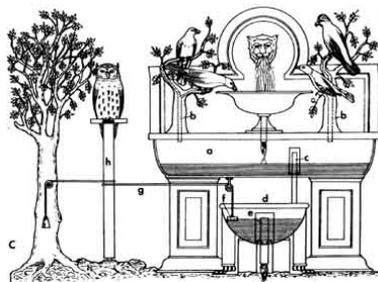


Figura 2.27 Pájaros de Herón.

Fuente: [Web_12].

Algunos ejemplos de los avances que se dieron respecto a los autómatas en la historia son: El Hombre de hierro de Alberto Magno (1204-1282) o la cabeza parlante de Roger Bacon (1214-1294), El Gallo de Estrasburgo (1352). Este autómata es el más antiguo que se conserva en la actualidad, formaba parte del reloj de la torre y la catedral de Estrasburgo y al dar la hora movía las alas y el pico.⁷

⁷Más información en: [Barrientos].

⁸ Más información en:[Web_12]

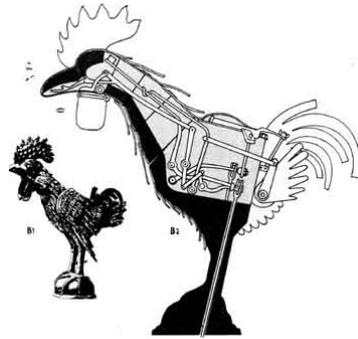


Figura 2.28 Gallo de Estrasburgo.

Fuente: [Web_12].

Durante el siglo XV y XVI, se dieron otros avances de los cuales los más relevantes son. El León mecánico construido por Leonardo Da Vinci (1452-1519) para el rey Luis XII de Francia, que se abría el pecho con su garra y mostraba el escudo de armas del rey.



Figura 2.29 León Mecánico.

Fuente: [Web_13].

En España es conocido el Hombre de palo, construido por Juanelo Turriano en siglo XVI para el emperador Carlos V. Este autómata con forma de monje, andaba y movía la cabeza, ojos, boca y brazos. Durante los siglos XVII y XVIII se crearon ingenios mecánicos que representaban figuras humanas, animales o pueblos enteros.

Jacques Vaucanson (1709-1782), autor del primer telar mecánico, construyó varios muñecos animados, entre los que se destacan un flautista capaz de tocar varias melodías y un pato (1738) capaz de graznar, beber, comer, digerir y evacuar la comida.

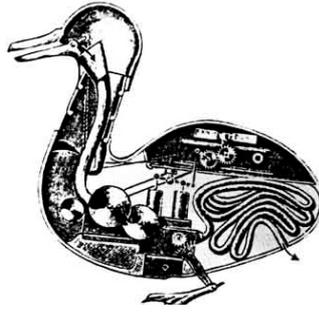


Figura 2.30 Pato de Vaucanson.

Fuente: [Web_12].

El relojero suizo Pierre Jaquet Droz (1721-1790) y sus hijos Jaquet y Henri-Louis construyeron muñecos capaces de escribir (1770), dibujar (1772) y tocar diversas melodías en un órgano (1773). Estos aún se conservan en el museo de Arte e Historia de Neuchastel, Suiza.

La hiladora giratoria de Hargreaves (1770), la hiladora mecánica de Crompton (1779), el telar mecánico de Cartwright (1785) y el telar de Jacquard (1801), fueron las, invenciones mecánicas automáticas para la producción, que dieron paso a la automatización industrial.

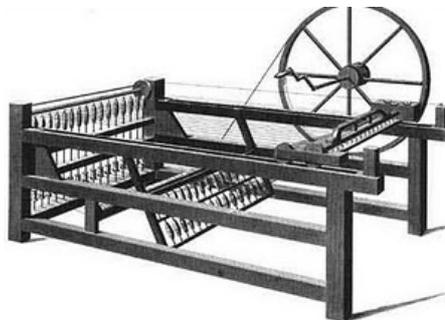


Figura 2.31 Hiladora Giratoria.

Fuente: [Web_12].

2.2.3 Definición y Clasificación del robot.

2.2.3.1 Definición de Robot Industrial

Robot industrial es un manipulador multifuncional reprogramable capaz de mover materias, piezas herramientas o dispositivos especiales según trayectorias variables y programadas para realizar tareas diversas. Concepto según la RIA⁹.

La ISO¹⁰ define al robot móvil como un manipulador de tres o más ejes, con control automático reprogramable, multifuncional móvil o no, destinado a ser utilizado en aplicaciones de automatización industrial.

2.2.3.2 Otros tipos de Robots

Robot de Servicio (IFR¹¹): Un robot que opera de manera semi o totalmente autónoma para realizar servicios útiles a los humanos y equipos, excluido operaciones de manufactura.

Robot Doméstico: Aquel destinado a ser usado por humanos sin formación técnica específica, al objeto de servirle como ayudante o colaborador en sus quehaceres o actividades diarias.

Robot Móvil (ISO): Robot que contiene todo lo necesario para su pilotaje y movimiento (control potencia y sistemas de navegación).

Robot Tele operado (ISO): Un robot que puede ser operado por un operador humano, extendiendo las capacidades sensoriales y motoras de este a localizaciones remotas.

⁹RIA: Asociación de Industrias Robóticas.

¹⁰Organización Internacional de Estándares.

¹¹IFR International Federation of Robotics.

2.2.4 Robot Móvil

2.2.5 Definición

Un robot móvil es una máquina que usa técnicas de navegación automáticas. Los mecanismos que los componen cuentan con partes mecánicas y eléctricas.

2.2.5.1 Generalidades

En los años sesenta se dio a conocer los vehículos filo guiados o robots móviles autónomos que eran capaces de desplazarse siguiendo un cable enterrado o unas marcas señaladas en el piso usando sensores ópticos. Estos eran aplicados principalmente en la industria.

En los años 70 se intentó incrementar la autonomía en los robots móviles para obtener mayor control. Sin embargo, la tecnología de la época imposibilitó desarrollar dicho control ya que muchas de las aplicaciones se orientaban a sistemas de visión y capacidad computacional, sumada a esto estaba el requerimiento de sensores con mayores capacidades que los que había disponibles en esa época.

El avance que tuvo la computación, para los años 80's, hizo que los robots móviles autónomos pudieran ser programados de mejor manera, agregándoles características importantes como detectar obstáculos, tomar decisiones, etc. Todo esto debido a que se logra implementar técnicas de inteligencia artificial en los robots móviles para incrementar la autonomía.

2.2.5.2 Robots Móviles Autónomos

Son todos aquellos que perciben su entorno por medio de sus sensores, y actúan sobre el mismo sin o con poca intervención humana, tales como: Seguidor de líneas en pistas con diferentes desniveles, buscador de luminosidad, etc.

Se considera que un robot móvil completamente autónomo es capaz de recibir información de su entorno, desplazarse desde un estado inicial a un estado final,

evitar obstáculos sin la intervención humana, y también pueden tener la capacidad de tomar decisiones y ajustarse a nuevas estrategias para cumplir los objetivos en beneficio del usuario.

Uno de los principales problemas que se tiene en los robots móviles se encuentra en su sistema de navegación, en la actualidad se están utilizando algoritmos computacionales y sistemas de posicionamiento global (GPS) para su posicionamiento y orientación, basados en técnicas de Inteligencia Artificial, reduciendo los errores acumulados por técnicas de odometría y sensores tales como: Acelerómetros, giróscopos, etc.¹².

2.2.5.3 Sistemas de locomoción para robots móviles

Uno de los sistemas más importantes para un robot móvil es el sistema de locomoción Este sistema es el complemento del sistema de manipulación con el que cuenta el robot.¹³

Para la manipulación de objetos se depende únicamente de la fuerza del brazo robótico, pero para la locomoción, el entorno en el que se encuentra el robot es el que, permite o no, la movilidad.

El entorno puede ser rocoso, húmedo, seco, etc. por esta razón y para que el robot adquiriera cierto grado de fuerza tanto para movilizarse como para manipular objetos se han generado mecanismos que deben adaptarse a los robots si se desea que estos puedan interactuar con facilidad en el medio. Los mecanismos creados dependen de ciertas características:

- Estabilidad
 - Número y geometría de los puntos de contacto.
 - Centro de gravedad.
 - Estabilidad dinámica y estática.
 - Inclinación del terreno.

¹²Más información en: [Ollero]

¹³ Capítulo 2 [Siegwart].

- Características de contacto
 - Punto de contacto o camino.
 - Angulo de contacto
 - Fricción.

- Tipo de entorno
 - Estructura
 - Medio (agua, aire, suelo rocoso o liso, etc.)

Así, teniendo en cuenta las características anteriormente planteadas y el entorno en el que van a estar inmersos los robots móviles, se han desarrollado algunos tipos de sistemas de locomoción, estos son:

Locomoción por patas: La locomoción por patas se caracteriza por tener varios puntos de contacto entre el robot y el suelo. Entre estos robots se han desarrollado bípedos (semejantes al ser humano), cuadrúpedos y hexápodos. Son complejos de controlar dependiendo del número de patas que posea el robot.

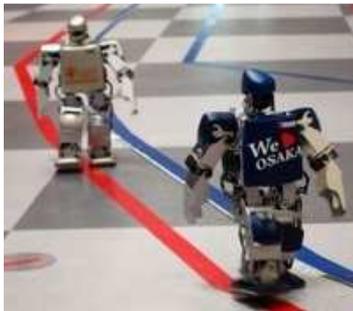


Figura 2.32 Robots Bípedos.

Fuente: [Web_14].



Figura 2.33 BigDog Robot Cuadrúpedo.

Fuente: [Web_15].



Figura 2.34 MorpHex Robot Hexápodo.

Fuente: [Web_16].

Locomoción por ruedas: La locomoción por ruedas es la más popular entre los robots móviles, tomando en cuenta que es una configuración fácil de construir, consume una menor cantidad de energía y alcanza velocidades altas según el tipo de ruedas colocadas.



Figura 2.35 Robotino Festo.

Fuente: [Manual Robotino].

Entre las diferentes configuraciones para las ruedas de robots móviles se encuentran: configuración Ackerman, triciclo, diferencial, síncrona, omnidireccional.¹⁴

Locomoción por oruga o deslizamiento: Las orugas se construyen uniendo, mediante una cadena que rodea las llantas, las ruedas delanteras y traseras, cuyo fin es aumentar la superficie de contacto con el suelo y conseguir una mayor tracción. Las orugas permiten rebasar mayores obstáculos que solamente usando ruedas e incluso subir escaleras. Como desventaja, presenta la gran cantidad de energía que necesita el robot en los giros. El giro se realiza rotando las cadenas en sentidos opuestos, de modo que las fuerzas inversas hacen girar el robot¹⁵.



Figura 2.36 Negotiator Robot¹⁶.

Fuente: [Web_18].

¹⁴ Mayor información en [Tesis_01] Pág. 08

¹⁵ Artículo: Locomoción en robots, disponible en: [Web_10].

¹⁶ Mayor información [Web_20]

2.2.5.4 Percepción

Debido a la capacidad de los robots móviles para trasladarse, en el entorno en que se encuentren, y al propósito para el cual hayan sido programados, harán uso de diferentes sensores¹⁷.

El robot móvil dependerá de los sensores con los que cuente para reconocer el medio por el que va a transitar, su posición en relación al entorno, velocidad y orientación. Los sensores serán los encargados de adquirir la información necesaria para el correcto funcionamiento del robot, transformando la energía de una forma a otra, en el mayor de los casos transforma las señales analógicas a digitales.¹⁸

2.2.5.5 Aplicaciones Actuales

En la actualidad, los robots comerciales e industriales son ampliamente utilizados, para realizar tareas de forma más exacta y precisa que los humanos además de ser asignados a cumplir tareas en lugares muy peligrosos, sucios o tediosos para los humanos; como en plantas de manufactura, montaje y embalaje, en transporte, en exploraciones en la Tierra y en el espacio, cirugía, armamento, investigación en laboratorios y en la producción en masa de bienes industriales o de consumo. Otras aplicaciones incluyen la limpieza de residuos tóxicos, minería, búsqueda y rescate de personas y localización de minas terrestres.

En Japón se está analizando la posibilidad de el cuidado del hogar para la población de edad avanzada pueda ser desempeñado por robots.

Los robots parecen estar abaratándose y reduciendo su tamaño, una tendencia relacionada con la miniaturización de los componentes electrónicos que se utilizan para controlarlos.

¹⁷ Dispositivos que usan procesamiento electrónico para la interpretación de información y evaluación de la misma a través de ordenadores o micro controladores.

¹⁸ Mayor información. [Tesis_01] Subcapítulo 2.2 Percepción, Pág. 13

Algunos de los robots disponibles actualmente

Windoro, el robot limpiacristales

Este robot de la compañía coreana PIRO consta de dos partes que se unen a través del cristal mediante un sistema de imanes de neodimio. Un juego de paños giratorios a alta velocidad y detergente limpiacristales hacen el resto.



Figura 2.37 Windoro, el robot limpiacristales.

Fuente: [Web_08].

RIBA 2, robot levanta pacientes

Investigadores en Japón presentaron un robot que puede levantar del suelo a un paciente que pese hasta 80 kilogramos y colocarlo en una silla de ruedas, una innovación que afirma liberará a los trabajadores de centros de salud de una tarea agotadora.

RIBA 2, robot suave al tacto, se mueve sobre ruedas y responde a comandos de voz. El equipo creador de RIBA 2 está conformado por expertos de RIKEN, un instituto de investigación de ciencias naturales de Japón, y de Tokai Rubber Industries.¹⁹

¹⁹Más información en: [Web_17].



Figura 2.38 Robot RIBA 2.

Fuente: [Web_17].

Roomba 520, robot aspiradora

Robot Roomba es un robot de aspirado inteligente y eficiente, cuenta con un avanzado software de inteligencia que usa docenas de sensores que le permiten monitorear su entorno y ajustar su comportamiento hasta 67 veces por segundo, garantizando así una limpieza efectiva, inteligente y segura.²⁰

Disponible en Ecuador.



Figura 2.39 ROOMBA.

Fuente: [Web_18].

Robot que aprende, toma sus propias decisiones y consulta sus dudas con otros robots

Científicos del Instituto Tecnológico de Tokio acaban de presentar un robot capaz de aprender de su entorno, deducir el curso de acción en función de cada nuevo

²⁰ Más información en: [Web_18].

aprendizaje y, preguntar por internet a otros robots cómo hacer algo que no sabe hacer. Esto es posible gracias a un nuevo algoritmo de programación llamado SOINN. Con este algoritmo el robot aprende de situaciones que ya ha experimentado, incluso permite a un robot aprender de las experiencias de otro. Una conexión a internet basta para que un robot que nunca ha preparado té pueda aprender a hacerlo ‘hablando’ con otros robots.



Figura 2.40 Robot que aprende, toma sus propias decisiones y consulta sus dudas con otros robots.

Fuente: [Web_08].

2.3 SENSORES

2.3.1 Sensor de proximidad inductivo analógico

Los sensores inductivos son una clase especial de sensores los cuales incorporan una bobina electromagnética misma que es usada para detectar materiales metálicos ferrosos. Son de gran utilización en la industria, tanto para aplicaciones de posicionamiento como para detectar la presencia o ausencia de objetos metálicos en un determinado contexto: detección de paso, de atasco, de codificación y de conteo, este tipo de sensor ignora cualquier objeto no metálico.

Los valores de salida que genera un sensor inductivo varían de acuerdo a la distancia del objeto y al tipo de material.

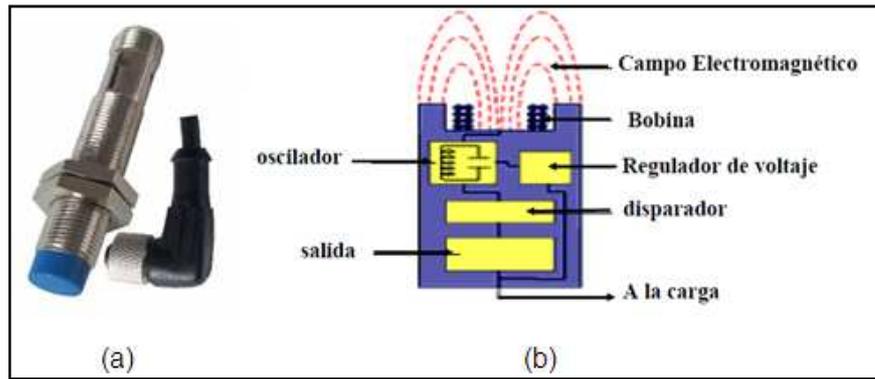


Figura 2.41 (a) Sensor inductivo, (b) Componentes sensor inductivo.

Fuente: [Web_20].

En el mercado se puede encontrar sensores inductivos con blindaje y sin blindaje, los sensores con blindaje constan de un anillo metálico en su núcleo para restringir la radiación lateral de campo mientras que los sensores sin blindaje poseen una mayor distancia de sensado en comparación con los blindados.

2.3.1.1 Aplicaciones de los sensores Inductivos

Entre las aplicaciones más comunes para los sensores inductivos se tiene las siguientes:

- Detección de posición totalmente abierta o totalmente cerrada de una válvula.
- Detectar la presencia de latas y tapas metálicas.
- Control filo guiado.
- Detección de rupturas de brocas o puntas de fresadora.

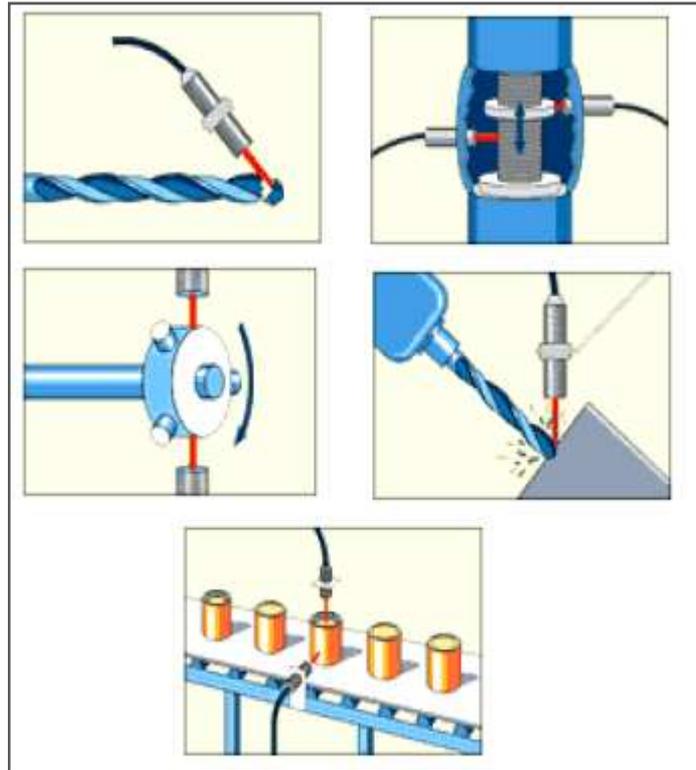


Figura 2.42 Aplicaciones de los sensores inductivos.

Fuente: [Web_20].

2.3.2 Sensores de medición de distancia por infrarrojo

Los detectores de rayos infrarrojos constan de un emisor que proyecta un haz de luz infrarroja, de un receptor y de la electrónica de procesamiento.

Considerando que el emisor y el receptor están montados a una pequeña distancia entre sí, se forma un triángulo formado por los rayos emitidos y por aquellos que, por reflexión, son captados por el receptor.

El receptor es un detector sensible a las posiciones, que es capaz de diferenciar entre los diversos puntos de incidencia de la luz. Procesando las señales, esos diversos puntos de incidencia se convierten en un valor analógico de tensión.

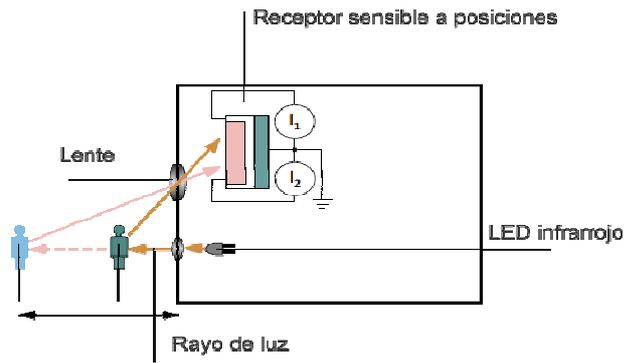


Figura 2.43 Principio de funcionamiento de un sensor de medición de distancia por infrarrojo.

Fuente: [Los autores].

2.3.3 Detectores de posición ópticos

Los detectores ópticos utilizan medios ópticos y electrónicos para detectar objetos, los mismos que utilizan luz roja o infrarroja. Los diodos luminosos semiconductores (LED) son fuentes plenamente fiables de luz roja o infrarroja. Son pequeños y robustos, de gran duración y permiten una sencilla modulación. Sus elementos receptores pueden ser fotodiodos o transistores fotosensibles, una de las principales ventajas de este tipo de sensores es que son menos sensibles a interferencias.

Además la luz roja tiene la ventaja que puede ser detectada a simple vista, lo que es importante al efectuar el ajuste de los ejes ópticos de los detectores.

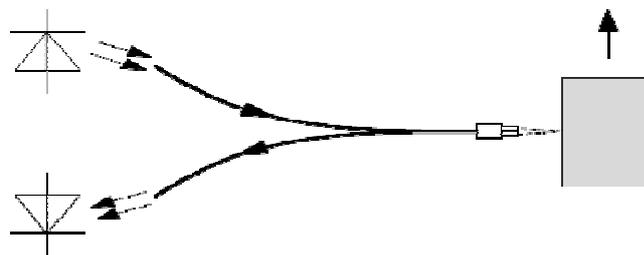


Figura 2.44 Principio de funcionamiento de un detector de posición óptico.

Fuente: [Los autores].

Los detectores ópticos tienen dos elementos principales: el emisor y el receptor. Dependiendo de su construcción y aplicación, se utilizan adicionalmente reflectores y fibras ópticas.

El emisor y el receptor pueden estar incluidos en un mismo cuerpo como los detectores de reflexión directa o detector de retro reflexión y en dos cuerpos diferentes como las barreras de luz unidireccional.

2.3.3.1 Detectores ópticos con fibra óptica

Los detectores ópticos con fibra óptica se utilizan si los detectores convencionales ocupan un espacio demasiado grande.

Las fibras ópticas de polímero son óptimas para grandes longitudes, debido a su bajo nivel de amortiguación de luz.

Dentro de sus principales ventajas tenemos:

- Fácil detección de objetos en lugares de difícil acceso.
- Pueden ser montados aparte del cuerpo del detector.
- Detección precisa de objetos pequeños.
- Posibilidad de montaje móvil.

En ambos tipos de detectores ópticos se modula la señal óptica (ciclos de emisión) para suprimir las interferencias ocasionadas por fuentes de luz ajenas. El receptor está coordinado con esos ciclos. Especialmente tratándose de luz infrarroja, pueden utilizarse adicionalmente filtros de luz diurna, con el fin de conseguir que los detectores sean menos sensibles a la luz del medio ambiente.

En el caso de detectores de reflexión directa, las funciones de conmutación son las siguientes:

- Conmutación a claro.- Se cierra la salida si un objeto atraviesa el rayo de luz, salida normalmente abierta, NO = Normally Open.
- Conmutación a oscuro.- Se abre la salida si un objeto atraviesa el rayo de luz, salida normalmente cerrada, NC = Normally Closed.

CAPITULO 3

3.1 ANÁLISIS

3.1.1 ROBOTINO

Robotino es un robot móvil de alta calidad, con accionamiento omnidireccional²¹ desarrollado y fabricado por la empresa FESTO.

Este robot móvil es ampliamente utilizado con finalidades didácticas, algunas de ellas, como el reconocimiento y la familiarización con sus componentes, actuadores eléctricos y sensores, la investigación sobre las diferentes tecnologías de regulación y algunas técnicas de procesamiento de imágenes y programación.

Además de esto el robot cuenta con su propio software de programación y con su simulador de entorno virtual en 3D. Este simulador es de gran utilidad, ya que le permite al estudiante hacer pruebas de funcionamiento sin utilizar al robot de forma física.



Figura 3.1 Robotino FESTO

Fuente: [Manual Robotino].

Tiene tres unidades de accionamiento que permiten realizar movimientos en todas las direcciones adelante, atrás y lateralmente. Además el robot puede girar sobre un punto.

²¹Que radian o captan en todas direcciones, 360°.

El robot cuenta con una webcam y varios tipos de sensores analógicos y digitales. El robot móvil Robotino puede programarse con el software Robotino View utilizando un ordenador, la comunicación entre ambos es del tipo LAN inalámbrico.

3.1.1.1 Diseño y Función

3.1.1.2 Chasis y puente de mando

Las baterías recargables, las unidades de accionamiento y la cámara se hallan montadas en el chasis, el mismo que está fabricado con una plataforma de acero inoxidable, en el que también se hallan situados los sensores de medición de distancia y el sensor anticolidión.

El diseño del chasis está pensado de tal manera que los componentes que lo conforman se ajusten al espacio que les ha sido asignado. Para el acoplamiento de sensores o actuadores extras, el chasis cuenta con un espacio adicional para ofrecer la opción de adjuntarlos.²²

El chasis ha sido diseñado especulando una manipulación frecuente y un uso didáctico del robot móvil, por esta razón y para proteger los sensores y el puente de mando, cuenta con dos asas, del mismo material que la plataforma, en los extremos para proveer de un agarre seguro y facilitar el traslado del robot móvil de un lugar a otro sin causar inconvenientes.

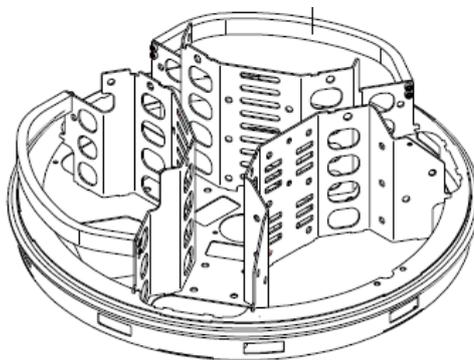


Figura 3.2 Chasis Robotino

Fuente: [Manual Robotino].

²²Chasis y puente de mando [Manual Robotino].

El robot móvil Robotino de Festo, trabaja con baterías recargables a 12V de corriente continua, con capacidad de funcionamiento constante que oscila entre las 2 horas antes de necesitar una recarga.

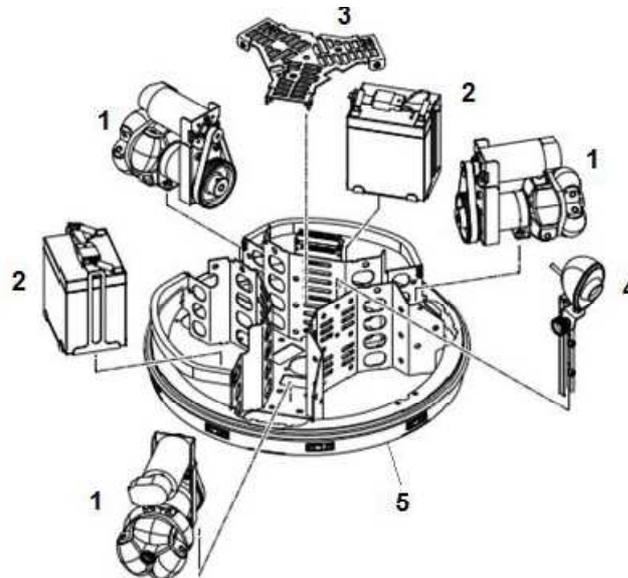


Figura 3.3 Componentes mecánicos de Robotino

Fuente: [Manual Robotino].

1	Motores DC
2	Baterías
3	Placa de fondo con listón protector
4	Plataforma de trabajo con webcam
5	Chasis

Tabla 3.1 Descripción de componentes mecánicos de Robotino.

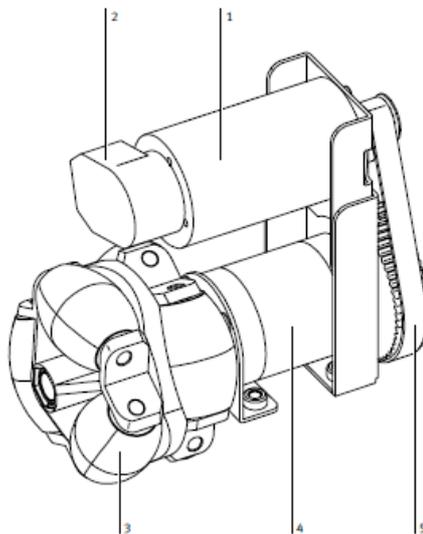
Fuente: [Manual Robotino].

3.1.1.3 Módulo de la unidad de accionamiento

El robot móvil Robotino es activado por 3 unidades de accionamiento omnidireccionales independientes. Las mismas que se hallan montadas formando un ángulo de 120° entre sí.

Cada una de las 3 unidades de accionamiento consta de los siguientes componentes:

- Motor DC
- Reductor con una relación de reducción de 16:1
- Rodillos omnidireccionales
- Correa dentada
- Encoder²³ incremental



Motor (1), Encoder incremental (2), Rodillos omnidireccionales (3), Reductor (4),
Correa dentada (5)

Figura 3.4 Unidad de accionamiento Robotino

Fuente: [Manual Robotino].

3.1.1.4 Módulo de cámara

El robot móvil Robotino[®] está equipado con una cámara web. Puede ajustarse su altura y su inclinación. La cámara permite visualizar imágenes en directo con la ayuda del software Robotino View, el mismo que ofrece diversas opciones de procesamiento de imágenes, con lo que se pueden evaluar las imágenes. Mediante un segmentador localiza superficies del mismo color en una imagen establecida y puede determinar la posición y tamaño de cualquier segmento. Pueden detectarse líneas en

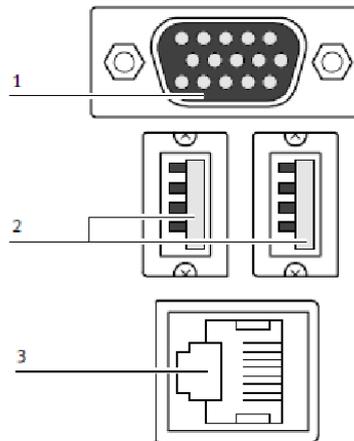
²³Encoder Incremental disponible en facultad.bayamon.inter.edu/arincon/encoderIncrementales.pdf

imágenes de vídeo. Los resultados pueden utilizarse para señalar objetos con precisión, así como para el seguimiento de recorridos y de objetos.²⁴

3.1.1.5 Unidad de Control

La unidad de control del robot móvil Robotino, está equipada con interfaces de entrada y salida de datos, estas son:

- **Ethernet;** Desde una computadora, el usuario puede acceder al sistema del robot móvil.
- **USB;** mediante esta interfaz se puede conectar cámara web, teclado y mouse, para el ingreso o manipulación de datos.
- **VGA;** Para conectar una pantalla y poder acceder al sistema operativo y a la librería C++ sin un PC, en caso de ser imposible o de no desear conectarse vía WLAN.



Zócalo conector VGA (1), Puertos USB (2), Interface Ethernet (3)

Figura 3.5 Unidad de Control Robotino

Fuente: [Manual Robotino].

²⁴ Unidad de control [Manual Robotino].

3.1.1.6 Módulo de Entradas y Salidas E/S

La interface E/S permite conectar sensores y actuadores adicionales. Estos se conectan por medio de un conector incluido.

- 8 entradas analógicas de 0 a 10 V (AIN1 hasta AIN8)
- 8 entradas digitales (DI1 hasta DI8)
- 8 salidas digitales (DO1 hasta DO8)
- 2 relés para actuadores adicionales (RELO y REL1). Los contactos de los relés pueden utilizarse como contacto normalmente abierto o contacto normalmente cerrado.

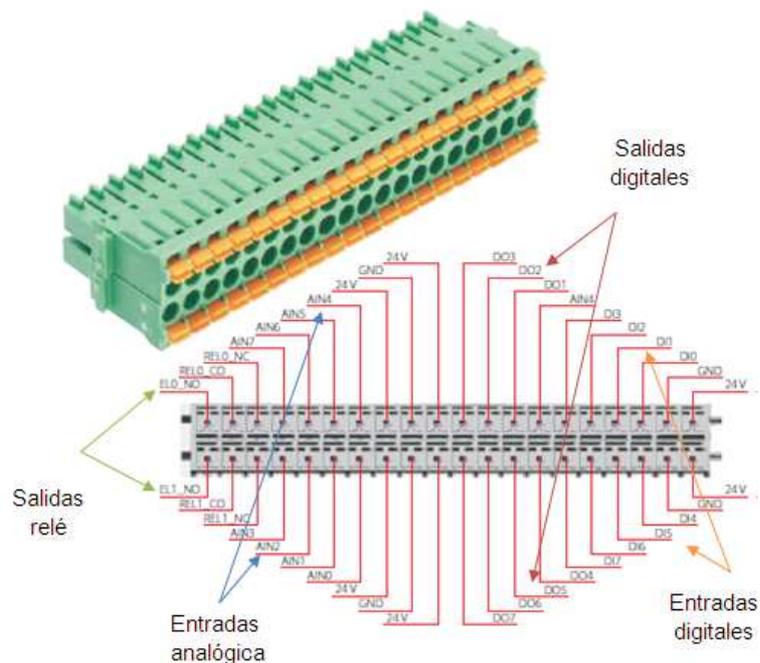


Figura 3.6 Interface entradas/salidas

Fuente: [Manual Robotino].

3.1.2 Sensores de Robotino

3.1.2.1 Sensor inductivo de proximidad

Es un componente adicional de Robotino que sirve para detectar objetos metálicos en el suelo y se utiliza para el control filo guiado. Este sensor cuenta con blindaje

metálico para detectar franjas metálicas en el piso, la señal que este envía varía de acuerdo a la altura a la que se encuentra del piso y del material a utilizarse.



Figura 3.7 Sensor inductivo.

Fuente: [Véase Nota²⁵].

Cable	Denominación
Café	Alimentación 24v
Azul	Tierra Gnd
Negro	Entrada AIN1
Blanco	--

Tabla 3.2 Conexión para el sensor de proximidad inductivo.

Fuente: [Manual Robotino].

En la siguiente figura se muestra la conexión de sus cables de funcionamiento dentro de la interface de E/S.

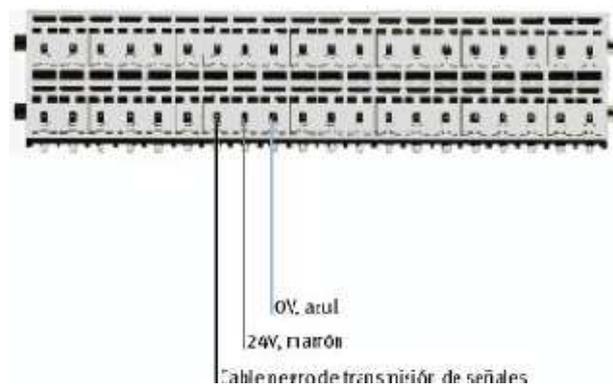


Figura 3.8 Conexión del sensor de proximidad Inductivo en la interface de E/S.

Fuente: [Manual Robotino].

²⁵<http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robotsrobotino/analogue-inductive-sensor.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC44NTguNjUxNQ>

Datos técnicos	
Tensión de funcionamiento	15-30 V DC
Tensión de salida	0-10 V
Diámetro	M12
Margen de detección	0 a 6 mm
Frecuencia de conmutación	1000 Hz
Temperatura ambiente	-25 a 70 °C
Material del cuerpo	Latón cromado

Tabla 3.3 Datos técnicos relevantes del sensor de proximidad inductivo.

Fuente: [Véase Nota²⁶].

3.1.2.2 Detectores ópticos con fibra óptica

Estos pueden ser implementados para un seguidor de línea, los cables flexibles de fibra óptica se conectan a una unidad óptica que funciona con luz roja y detectan la luz reflejada.

Estos sensores inciden luz difusa, la detección varía de acuerdo al color del material y de acuerdo al tipo de superficie en la que se trabaje, estos detectores no pueden detectar diferencias graduales en la luz reflejada, es decir este detector lo que hace es enviar una señal binaria (0 o 1) dependiendo de la cantidad de luz reflejada.



Figura 3.9 Sensor óptico.

Fuente: [Véase Nota²⁷].

²⁶<http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robotsrobotino/analogue-inductive-sensor.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC44NTguNjUxNQ>

²⁷<http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robotsrobotino/opto-sensor.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC44NTguNjUxNA>

Además poseen un potenciómetro el mismo que sirve para regular el sensor es decir incrementar o reducir la distancia máxima de reflexión, cuentan con un tiempo de respuesta de 0.5 ms ante las variaciones de luz.

Cable	Denominación
Café-1	Alimentación 24v
Azul-1	Tierra Gnd
Blanco-1	Entrada DI1
Negro-1	Entrada DI2
Café-2	Alimentación 24v
Azul-2	Tierra Gnd
Blanco-2	Entrada DI3
Negro-2	Entrada DI4

Tabla 3.4 Conexión para el sensor óptico.

Fuente: [Manual Robotino].

En la siguiente figura se muestra la conexión de los cables para cada uno de los detectores dentro de la interface de E/S.

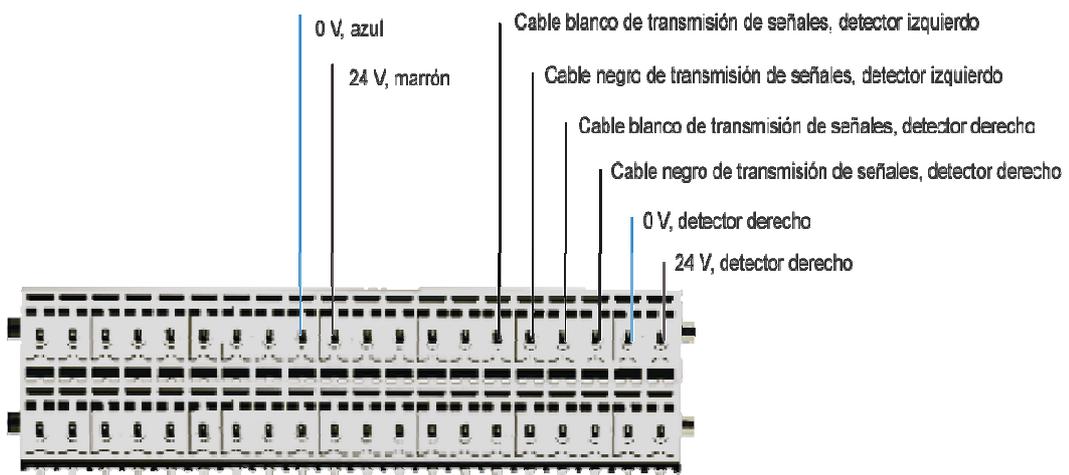


Figura 3.10 Conexión del sensor óptico con fibra óptica en la interface de E/S.

Fuente: [Manual Robotino].

Datos técnicos	
Tensión de funcionamiento	15-30 V DC
Margen de medición	120 mm
Señal de salida	Digital
Conexión	Cable de fibra óptica

Tabla 3.5 Datos técnicos relevantes del detector óptico con fibra óptica.

Fuente: [Véase Nota²⁸].

Montaje de los sensores ópticos con fibra óptica

Escuadra de montaje de fibra óptica (1)

Ranuras de montaje (2)

Cables de fibra óptica (3)

Unidad de fibra óptica (4)

Agujeros de montaje (5)

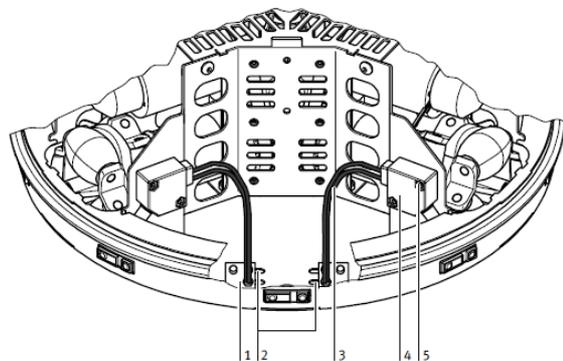


Figura 3.11 Montaje de sensores ópticos.

Fuente: [Manual Robotino].

3.1.2.3 Sensor anticolidión

“El sensor anticolidión está formado por una banda de detección fijada alrededor de un aro que circunda el chasis. Una cámara de conmutación se halla situada dentro de un perfil de plástico. Dos superficies conductoras se hallan dispuestas dentro de la cámara, manteniendo una determinada distancia entre sí.

Estas superficies entran en contacto cuando se aplica una mínima presión a la banda. Con ello, una señal perfectamente reconocible es transmitida a la unidad de control.

²⁸<http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robotsrobotino/opto-sensor.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC44NTguNjUxNA>

Las colisiones con objetos en cualquier punto del cuerpo se detectan y, por ejemplo, se provoca la detención del Robotino.²⁹

3.1.2.4 Sensores de medición de distancia por infrarrojos

Robotino se encuentra equipado con nueve sensores de medición de distancia por infrarrojos, los cuales están montados en el chasis formando un ángulo de 40° entre sí.

Cada uno de estos sensores puede ser interrogado individualmente por medio de la placa de circuito de E/S. Con ello pueden evitarse obstáculos, pueden mantenerse distancias y adoptar protecciones frente a un determinado objetivo.

Los sensores son capaces de medir distancias con precisión o relativas a objetos, con valores entre 4 y 30 cm. La conexión del sensor es especialmente sencilla e incluye tan sólo una señal de salida analógica y la alimentación. Este sensor emite una señal analógica.



Figura 3.12 (1) Sensores de distancia, (2) Sensor anticollision.

Fuente: [Manual Robotino].

²⁹Sensor anticollision, [Manual Robotino].

3.1.3 Prueba de sensores

3.1.3.1 Sensores Ópticos

Color cinta	Sensor Izquierdo	Sensor Derecho
Blanco	Falso	Falso
Negro	True	True
Azul	True / False	True / False

Tabla 3.6 Estado de los sensores ante diferentes colores.

Fuente: [Los autores].

3.1.3.2 Sensor Inductivo

Material: Cinta de aluminio	Voltaje (voltios)
Centro de la cinta.	2,251
Extremos de la cinta.	2,251-9,96
Fuera de la cinta.	9,96

Tabla 3.7 Sensor inductivo ante la cinta de aluminio a 2 mm del suelo.

Fuente: [Los autores].

3.2 SIMULACIÓN Y AJUSTES

Se elaboró la pista a seguir para las pruebas preliminares utilizando cinta de aluminio cubierta con cinta negra y de esta forma hacer uso de los sensores inductivo y óptico.

Para la toma de mediciones de velocidad, la pista fue dividida en tipos de curva, aplicando a cada tipo una velocidad y ángulo de giro diferente.



Figura 3.13 Camino para curvas preliminares.

Fuente: [Los autores].

3.2.1 Pruebas preliminares de velocidad y ángulo de giro

Para cada curvatura se realizó pruebas con diferentes valores de velocidad y ángulo de giro, para verificar si el móvil cuenta con la capacidad de seguir la línea, dando como resultado la siguiente tabla.

CURVA ABIERTA		LINEA RECTA		CURVA TIPO "S"		CURVA "U"	
Velocidad (mm/s)	Ángulo (grados)	Velocidad (mm/s)	Ángulo (grados)	Velocidad (mm/s)	Ángulo (grados)	Velocidad (mm/s)	Ángulo(grados)
260	10	500	< 5	400	20	280	40
280	10	600	< 5	350	30	260	60
300	10	700	< 5	300	40	250	60
400	10	800	< 5	250	20	400	40
350	30	—	—	—	—	300	40
350	20	—	—	—	—	240-250	50-60
300	20	—	—	—	—	—	—

Tabla 3.8 Pruebas preliminares de velocidad y ángulo de giro.

Fuente: [Los autores].

Con las velocidades y ángulos marcados, el robot se comporta bien, y logra seguir la curva.

En las figuras 3.14 a la 3.18 se muestra cada una de las curvas con las que se realizó la simulación.



Figura 3.14 Curva abierta

Fuente: [Los autores].



Figura 3.15 Línea recta

Fuente: [Los autores].



Figura 3.16 Curva Tipo “s” primer tramo

Fuente: [Los autores].



Figura 3.17 Curva Tipo “S” segundo tramo

Fuente: [Los autores].



Figura 3.18 Curva en “U”

Fuente: [Los autores].

En curvas cerradas, el ángulo debe ser alto y la velocidad baja, en una línea recta el ángulo de giro no debe ser cero, ya que al poner un ángulo nulo pierde la línea, funciona cualquier velocidad con un ángulo de 5° .

El principal problema es la luz. Se debe incorporar una luz que provea de luminosidad constante durante el recorrido de la pista.

3.3 MODELO NUMÉRICO

3.3.1 Descripción del procedimiento

El sistema de estudio se comporta de forma dinámica de tal forma que para obtener el modelo numérico que describa su comportamiento se desarrolla un modelo experimental ya que se parte de una base de datos experimentales tomados del sistema en funcionamiento.

Para el desarrollo del modelo experimental se utiliza la regresión lineal múltiple ya que el sistema sujeto a estudio es una función que depende de otras dos variables, tanto de la velocidad como el ángulo de giro y la función a obtener es la distancia en mm que se encuentra alejado el robot del centro de la línea. Dicha distancia a la vez

se encuentra en función del voltaje que emite el sensor inductivo, cuando el robot se encuentra en el centro de la línea el sensor emite un voltaje de 2,251 voltios el cual se va elevando conforme el robot se va alejando del centro hasta llegar a los 9,960 voltios.

La distancia del centro de la línea hasta el extremo es de 15 mm, para obtener esta distancia en función de voltaje se realiza un escalamiento el cual está conformado por dos bloques:

- El primer bloque se divide el voltaje del sensor inductivo por el voltaje mínimo que emite el mismo.

$$\text{Bloque 1} = \text{Voltaje emitido x el sensor} - \text{Voltaje mínimo}(2.251)$$

- En el segundo bloque el resultado de esta división se lo multiplica por la división entre la distancia máxima (15mm) y el voltaje máximo que emite el sensor inductivo.

$$\text{Bloque 2} = \text{Bloque1} \left(\frac{\text{Dist. máxima (15 mm)}}{\text{Voltaje máximo}(7.71)} \right)$$

De esta forma se logra obtener la distancia en función del voltaje.

Para obtener los datos experimentales para el desarrollo del modelo es necesario generar una base de datos la cual conste del voltaje que emite el sensor, la velocidad y ángulo de giro del robot cuando este se encuentre trabajando. Una vez realizado el escalamiento para obtener la distancia se procede a trabajar con el método de regresión lineal múltiple.

3.3.2 Método utilizado: Regresión lineal múltiple

“La regresión lineal múltiple es una extensión de la regresión lineal en la que y es una función lineal de dos o más variables independientes”³⁰, como el siguiente ejemplo donde se muestra una función lineal de x_1 y x_2 :

$$y = a_0 + a_1x_1 + a_2x_2$$

³⁰ [Chapra], Regresión lineal múltiple.

Esta ecuación es muy útil cuando se realiza ajuste de datos experimentales, donde la variable sujeta a estudio (variable dependiente) es una función de otras dos variables, y para este caso bidimensional, la línea de regresión se convierte en un plano.

Para la determinación de los mejores valores para los coeficientes se determinan al realizar la suma de los cuadrados de los residuos,

$$S = \sum_{i=1}^n (y_i - a_0 - a_1x_{1i} - a_2x_{2i})^2$$

Y derivando con respecto cada uno de los coeficientes desconocidos,

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1x_{1i} - a_2x_{2i})$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_{1i}(y_i - a_0 - a_1x_{1i} - a_2x_{2i})$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_{2i}(y_i - a_0 - a_1x_{1i} - a_2x_{2i})$$

Los coeficientes que dan la suma mínima de los cuadrados de los residuos se obtienen al igualar a cero las derivadas parciales y expresando el resultado en forma matricial.

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_{1i} \\ \sum y_i x_{2i} \end{bmatrix}$$

Luego de formar la matriz se procede a resolver un sistema de ecuaciones de 3*3.

3.3.3 Tablas obtenidas, resolución y ecuación final

Para la resolución del sistema lo primero a obtener son las sumatorias a través de una toma 20 datos de cada una de las variables. A continuación se detalla la tabla de datos del funcionamiento del robot con los cuales se realizan los cálculos.

Giro (grados)	Velocidad (mm/s)	Distancia del centro (mm)
-27,551	228,122	13.41
-25,759	293,329	11.81
-25,035	351,087	11.51
-4,365	579,743	1.85
-21,568	423,190	9.68
-19,256	441,104	8.62
-24,585	351,087	10.90
-20,581	423,190	9.68
-18,698	441,104	8.62
-13,256	520,173	6.03
-8,857	578,859	3.90
-6,44	581,497	2.92
-12,854	543,296	5.88
-18,698	441,104	8.47
-28,65	282,277	14.04
-30,987	226,595	15.00
-6,651	581,223	2.99
-8,857	578,859	5.35
-6,44	581,497	2.92
-28,65	282,277	14.04

Tabla 3.9 Datos tomados del funcionamiento del sistema.

Fuente: [Los autores].

\sum Giro	\sum Velocidad	\sum Dist. Centro
-357.748729	8729.613	167.61

Tabla 3.10 Sumatorias de las variables del sistema.

Fuente: [Los autores]

Con las sumatorias de los datos se realiza los cálculos requeridos para desarrollar la matriz a resolver.

$x_1(\sum \text{Giro})$	$x_2(\sum \text{Vel})$	$y(\sum \text{Dist. Centro})$	$x_1 * x_2$
-357.74	8729.61	167.61	-3122917.79
$x_1 * y$	$x_2 * y$	x_1^2	x_2^2
-59959.03	1463126.69	127976.76	76206143.13

Tabla 3.11 Cálculos realizados con las sumatorias obtenidas.

Fuente: [Los autores]

Y con estos datos se forma la matriz, para obtener el valor de las constantes a_0, a_1, a_2 .

$$\begin{bmatrix} 20 & -357.74 & 8729.61 \\ -357.74 & 127976.76 & -3122917.79 \\ 8729.61 & -3122917.79 & 76206143.13 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 167.61 \\ -59959.03 \\ 1463126.69 \end{bmatrix}$$

Dando como resultado los siguientes valores:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1.8153 * 10^{-4} \\ -0.4683 \\ 7.0966 * 10^{-6} \end{bmatrix}$$

Las constantes a_0, a_1, a_2 se las reemplaza en la ecuación:

$$D = a_0 + \theta a_1 + V a_2$$

Resultando:

$$D = 1.8153 * 10^{-4} - 0.4683\theta + 7.0966 * 10^{-6}V$$

Ecuación que describe el comportamiento del sistema.

3.3.4 Diseño del Fis

Para tener una idea mas clara de lo que será el diseño de un sistema difuso con ayuda de una herramienta computacional, como Matlab, se realiza previamente el diseño del sistema de forma analítica.

Pasos típicos en el diseño de un sistema basado en lógica difusa

- **Identificar las variables del dominio**
 - Variables “de estado”: describen las entradas del sistema

Inductivo
OpticoIzq
OpticoDer

- Variables “de control”: describen las salidas del sistema
Velocidad
Giro

- **Identificar los términos lingüísticos vinculados a cada variable y su respectiva definición.**

- Definir las “partes difusas” de cada variable (etiquetas lingüísticas).
- Definir las funciones grado de verdad asociadas a cada etiqueta lingüística.

Se pretenderá que las particiones definidas “cubran ” la totalidad del espacio de valores de las distintas variables.

- Es favorable que exista solapamiento.

- **Construir las reglas difusas que correspondan. (crear una representación simbólica)**

- Se crea una FAM (Fuzzy Association Matrix) para cada variable “de control”, vinculando las posibles combinaciones de las variables “de estado” (antecedente) con una de sus etiquetas lingüísticas (consecuente).

- **Seleccionar la implementación y los parámetros de los operadores difusos a utilizar. (parametrizar el “funcionamiento numérico”)**

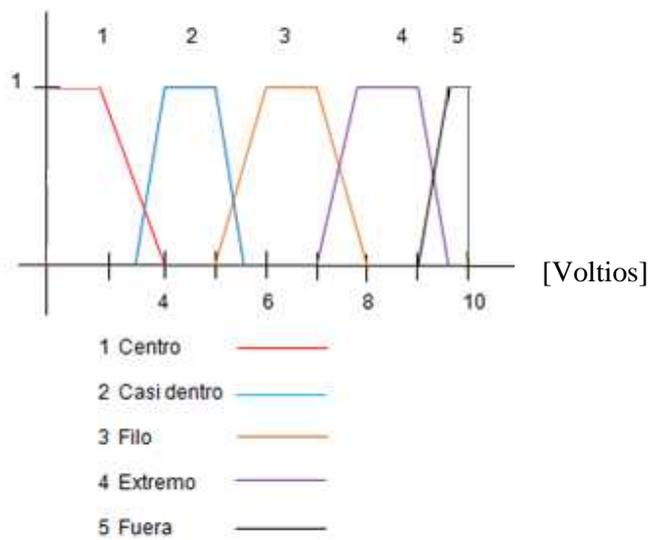
- a) Implementación conectivas difusas (\wedge , \vee , \neg)
- b) Implementación implicación difusa (\rightarrow): Mamdani

- c) Activación de reglas: normalmente se considerará que una regla difusa se activa si el grado de verdad de su antecedente es 0.
- d) Operadores: fuzzyficación y desfuzzyficación.

Ejemplo: Control difuso de velocidad aplicado al robot móvil Robotino.

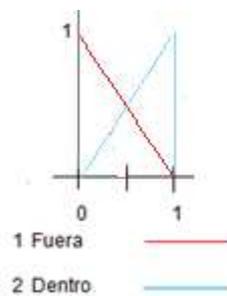
Variables “de estado”:

- Entrada1: Sensor Inductivo



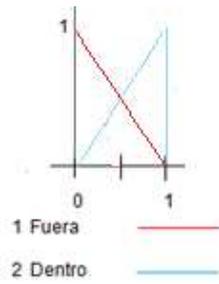
Para la variable Sensor Inductivo, se asumen 5 etiquetas lingüísticas: centro (C), casi dentro (CD), filo (F), extremo (E), fuera (FU)

- Entrada 2: Sensor Optico Izquierdo



Para la variable Optico Izquierdo, se asumen 2 etiquetas lingüísticas: Fuera (FU), y dentro (D).

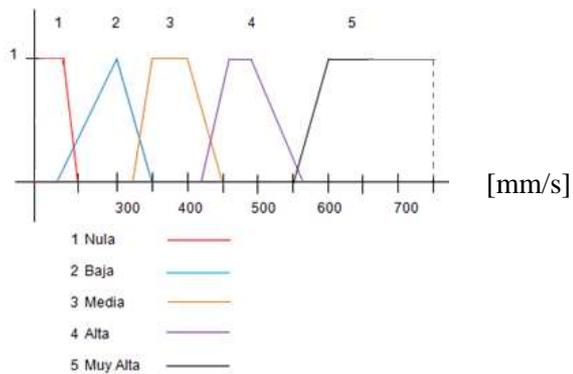
- Entrada 3: Sensor Optico Derecho



Para la variable Optico Derecho, se asumen 2 etiquetas lingüísticas: Fuera (FU), y dentro (D).

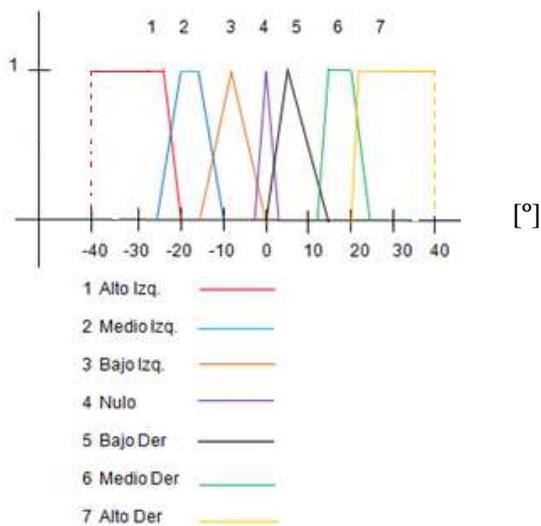
VARIABLES “de control”:

- Salida1: Velocidad [mm/s]



Para la variable Velocidad, se asumen 5 etiquetas lingüísticas: nula (N), baja (B), media (M), alta (A) y muy alta (MA).

- Salida2: Giro



Para la variable Giro, se asumen 7 etiquetas lingüísticas: Alto Izq (AI), Medio Izq (MI), Bajo Izq (BI), Nulo (N), Bajo Der (BD), Medio Der (MD), Alto Der (AD).

Reglas difusas:

Las reglas que esta obedeciendo el robot móvil, en este control difuso son las siguientes:

1. IF "Inductivo" IS "Centro"
THEN "Velocidad" IS "Muy Alta" ALSO "Giro" IS "Nulo".
2. IF "Inductivo" IS "Fuera"
THEN "Velocidad" IS "Nula".
3. IF "Inductivo" IS "Centro" AND "Optico_Izq" IS "Fuera"
THEN "Velocidad" IS "MuyAlta" ALSO "Giro" IS "Nulo".
4. IF "Inductivo" IS "Centro" AND "Optico_Der" IS "Fuera"
THEN "Velocidad" IS "MuyAlta" ALSO "Giro" IS "Nulo".
5. IF "Inductivo" IS "CasiDentro" AND "Optico_Der" IS "Dentro" THEN "Velocidad" IS "Alta" ALSO "Giro" IS "Medio_Izq".
6. IF "Inductivo" IS "CasiDentro" AND "Optico_Izq" IS "Dentro" THEN "Velocidad" IS "Alta" ALSO "Giro" IS "Medio_Der".
7. IF "Inductivo" IS "Filo" AND "Optico_Izq" IS "Dentro" THEN "Velocidad" IS "Media" ALSO "Giro" IS "Medio_Der".
8. IF "Inductivo" IS "Filo" AND "Optico_Der" IS "Dentro" THEN "Velocidad" IS "Media" ALSO "Giro" IS "Medio_Izq".
9. IF "Inductivo" IS "Extremo" AND "Optico_Der" is "Dentro" THEN "Velocidad" IS "Baja" ALSO "Giro" IS "Alto_Izq".
10. IF "Inductivo" IS "Extremo" AND "Optico_Izq" IS "Dentro" THEN "Velocidad" IS "Baja" ALSO "Giro" IS "Alto_Der".
11. IF "Inductivo" IS "Fuera" AND "Optico_Izq" IS "Dentro" THEN "Velocidad" IS "Nula" ALSO "Giro" IS "Alto_Der".
12. IF "IF "Inductivo" IS "Fuera" AND "Optico_Der" IS "Dentro" THEN "Velocidad" IS "Nula" ALSO "Giro" IS "Alto_Izq".
13. IF "Inductivo" IS "CasiDentro" THEN "Velocidad" IS "Alta".

14. IF "Inductivo" IS "Filo" THEN "Velocidad" IS "Media"

Se consideran las siguientes FAM para las variables "de control"

Velocidad: Si Sensor Inductivo y Op Izq

Op Izq \ Inductivo	D	FU
C	x	MA
CD	x	A
F	M	M
E	B	B
FU	N	x

Velocidad: Si Sensor Inductivo y Op Der

Op Der \ Inductivo	D	FU
C	x	MA
CD	x	A
F	M	M
E	B	B
FU	N	x

Giro:

Op Izq \ Inductivo	D	FU
C	x	N
CD	x	BD
F	BD	MD
E	MD	MD
FU	AD	x

Op Der \ Inductivo	D	FU
C	x	N
CD	x	BI
F	BI	MI
E	MI	MI
FU	AI	x

Nota: “x” representa una condicion no válida.

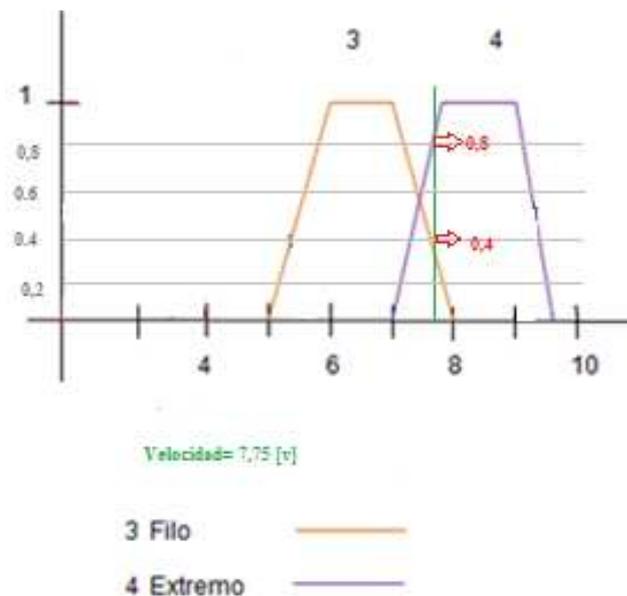
Ejemplo de funcionamiento:

Se suponen las siguientes entradas en los sensores del sistema:

- Sensor inductivo: 7.75 [v]
- Sensor óptico derecho: 1 (entrada digital)
- Sensor óptico izquierdo: 1 (entrada digital)

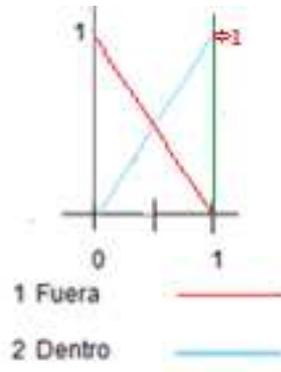
Fuzzyficación:

Variable Inductivo = 7,75 [v]



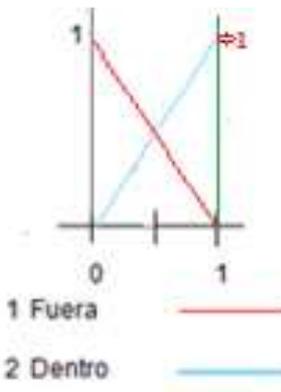
El valor elegido, “Inductivo=7,75 [v]”, corresponde con un grado de verdad 0,4 para el valor difuso Filo (F) y con un grado de verdad 0,8 para el valor difuso Extremo (E).

Variable Optico Der.= 1



El valor “Op Der =1” corresponde a un grado de verdad de 1 para el valor difuso Dentro (D).

Variable Optico Izq.=1



El valor “Op Izq =1” corresponde a un grado de verdad de 1 para el valor difuso Dentro (D).

Reglas Activadas:

Velocidad: Si Sensor Inductivo y Op Izq

Op Izq \ Inductivo	D	FU
C	x	MA
CD	x	A
F	M	M
E	B	B
FU	N	x

Velocidad: Si Sensor Inductivo y Op Der

Op Der \ Inductivo	D	FU
C	x	MA
CD	x	A
F	M	M
E	B	B
FU	N	x

Giro:

Op Izq \ Inductivo	D	FU
C	x	N
CD	x	BD
F	BD	MD
E	MD	MD
FU	AD	x

Op Der \ Inductivo	D	FU
C	x	N
CD	x	BI
F	BI	MI
E	MI	MI
FU	AI	x

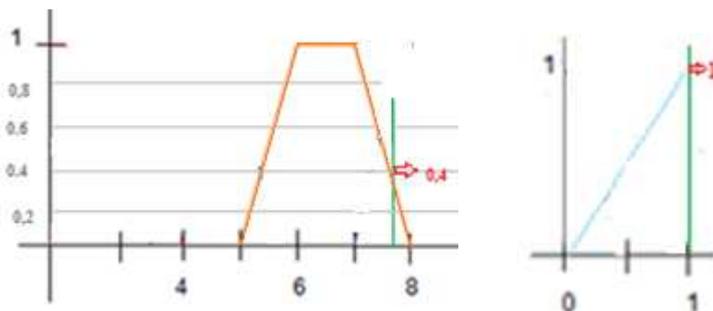
Nota: "x" representa una condicion no válida.

1. IF "Inductivo" IS "Filo" AND "Optico_Izq" IS "Dentro"
THEN "Velocidad" IS "Media" ALSO "Giro" IS "Medio_Der".
2. IF "Inductivo" IS "Filo" AND "Optico_Der" IS "Dentro"
THEN "Velocidad" IS "Media" ALSO "Giro" IS "Medio_Izq".
3. IF "Inductivo" IS "Extremo" AND "Optico_Der" is "Dentro"
THEN "Velocidad" IS "Baja" ALSO "Giro" IS "Alto_Izq".
4. IF "Inductivo" IS "Extremo" AND "Optico_Izq" IS "Dentro"
THEN "Velocidad" IS "Baja" ALSO "Giro" IS "Alto_Der".

Inferencia al Mínimo (AND)

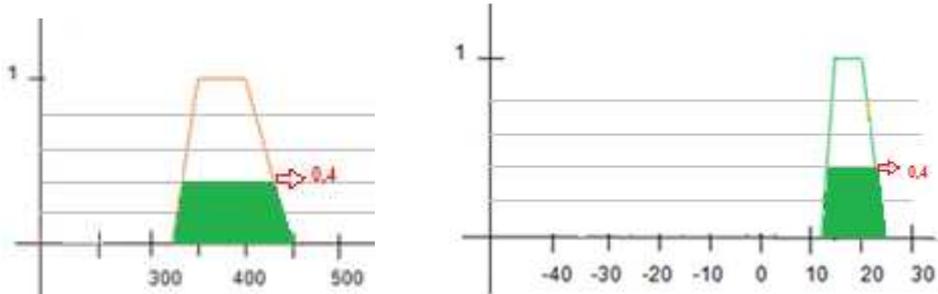
IF "INDUCTIVO" IS "Filo" AND "Optico_Izq" IS "Dentro" THEN
"Velocidad" IS "Media" ALSO "Giro" IS "Medio_Der".

Inductivo es "F" AND OpticoIzq es "D" \longrightarrow $\min\{0,4; 1\} = 0,4$



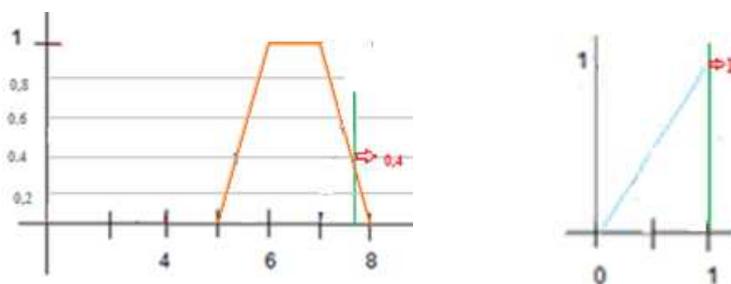
THEN

Velocidad es "M" ALSO Giro es "MD"



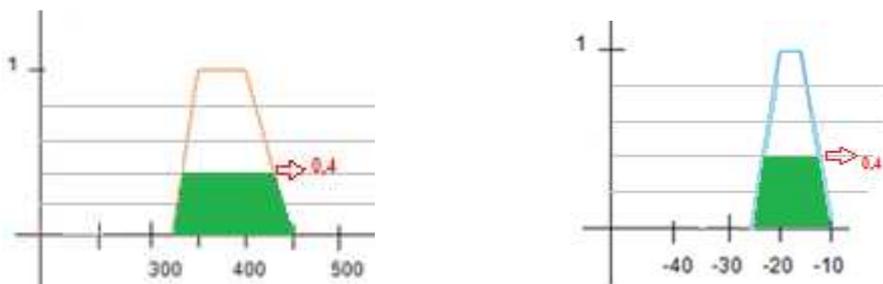
IF "INDUCTIVO" IS "Filos" AND "Optico_Der" IS "Dentro" THEN "Velocidad" IS "Media" ALSO "Giro" IS "Medio_Izq".

Inductivo es "F" AND OpticoDer es "D" \longrightarrow $\min\{0,4; 1\} = 0,4$



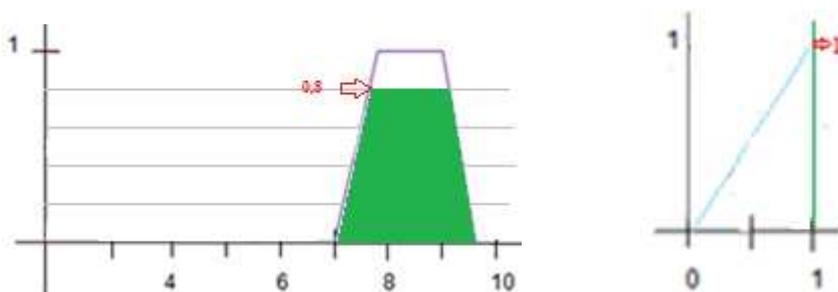
THEN

Velocidad es "M" ALSO Giro es "MI"



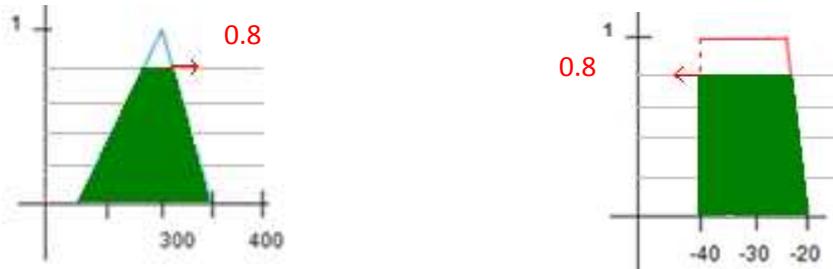
IF "Inductivo" IS "Extremo" AND "Optico_Der" is "Dentro" THEN "Velocidad" IS "Baja" ALSO "Giro" IS "Alto_Izq".

Inductivo es "E" AND OpticoDer es "D" $\min\{0,8; 1\} = 0,8$



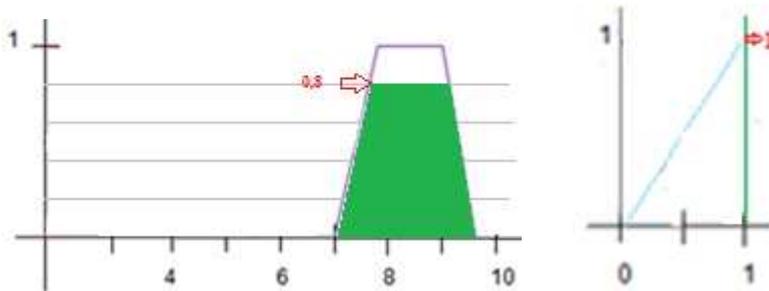
THEN

Velocidad es "B" ALSO Giro es "AI"



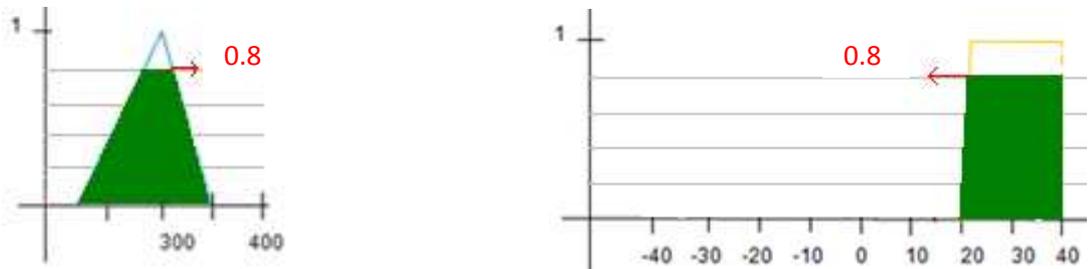
IF "Inductivo" IS "Extremo" AND "Optico_Izq" IS "Dentro" THEN "Velocidad" IS "Baja" ALSO "Giro" IS "Alto_Der".

Inductivo es "E" AND OpticoIzq es "D" $\longrightarrow \min\{0,8; 1\} = 0,8$



THEN

Velocidad es "B" ALSO Giro es "AD".



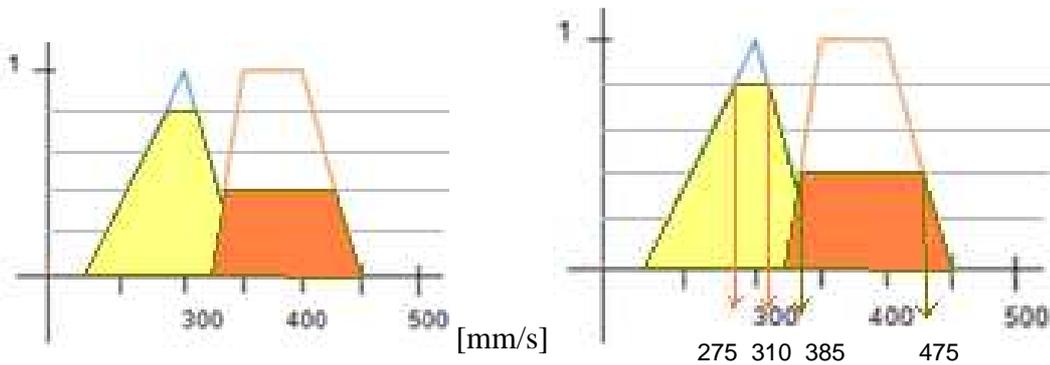
Defuzzyficación

Después de realizar la inferencia al mínimo con cada una de las reglas activadas, se tendrán unas nuevas gráficas.

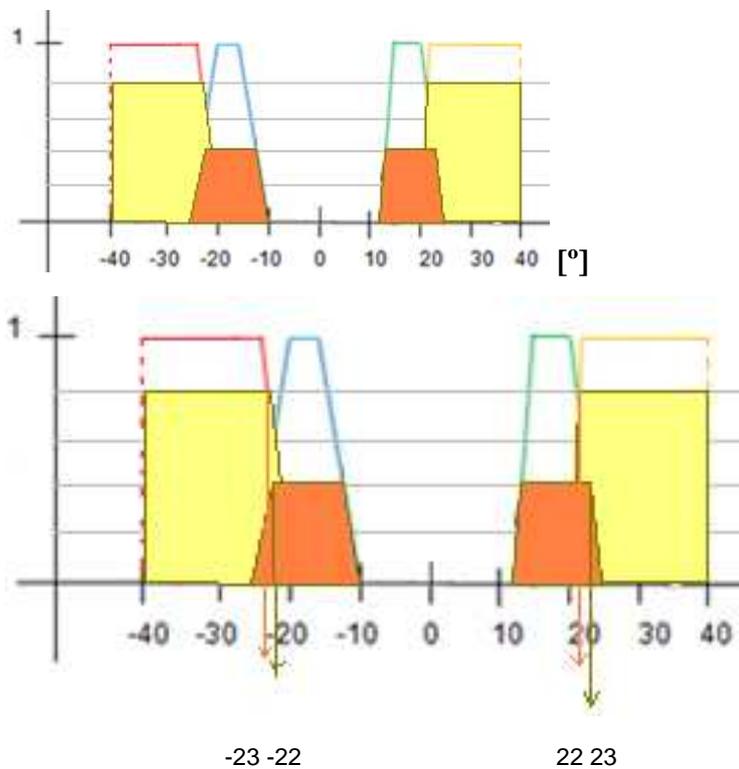
Estas gráficas deberán sumarse para formar una sola gráfica para posteriormente realizar la desfuzzyficación, aplicando la respectiva fórmula.

$$g = \frac{\sum_{i=1}^n Xi \times \mu(Xi)}{\sum_{i=1}^n \mu(Xi)}$$

Salida1: Velocidad [mm/s]



Salida2: Giro [°]



$$g = \frac{\sum_{i=1}^n Xi \times \mu(Xi)}{\sum_{i=1}^n \mu(Xi)}$$

$$g = \frac{(275 \times 0,8) + (310 \times 0,8) + (385 \times 0,8) + (475 \times 0,8) + (-22 \times 0,4) + \dots + (23 \times 0,4)}{0,8 + 0,8 + 0,8 + 0,8 + .04 + 0,4 + 0,4 + 0,4}$$

$$g = \frac{1156}{5.12}$$

$$g = 240.83$$

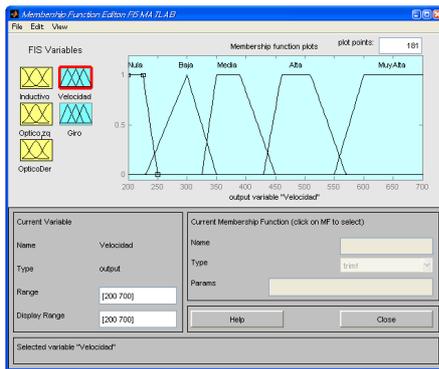
Variable de control

Salida1: Velocidad a 240.83 [mm/s]

Salida2: Angulo será $\pm 22^\circ, 22^\circ$ según la entrada que detecten los sensores digitales ópticos (sensor izquierdo / sensor derecho).

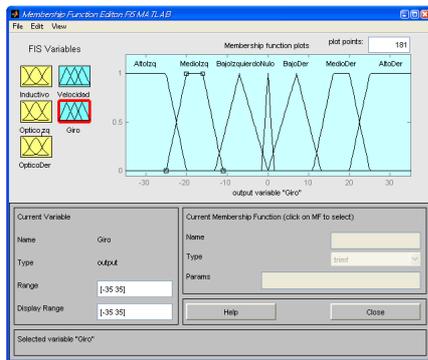
3.3.5 Diseño del FIS con Matlab

	<p>Utilizando el FIS Editor de la librería de Fuzzy Logic de Matlab, se procede a diseñar el sistema difuso, asignando las respectivas entradas y salidas, así como también las reglas que deberá seguir dicho sistema.</p>
	<p>En el Membership Function Editor se irán agregando las diferentes funciones de membresía, tantas como se requieran.</p> <p>Entrada Inductivo: Consta de cinco conjuntos difusos: centro, casi dentro, filo, extremo y fuera.</p>
	<p>Se agregarán una a una las entradas y los conjuntos difusos de las mismas, teniendo en cuenta los rangos de cada una y el tipo de función a dibujar.</p> <p>Entrada Optico Izq: dos conjuntos difusos; Fuera y dentro</p>
	<p>Entrada Optico Der: Dos conjuntos difusos Fuera y dentro.</p>

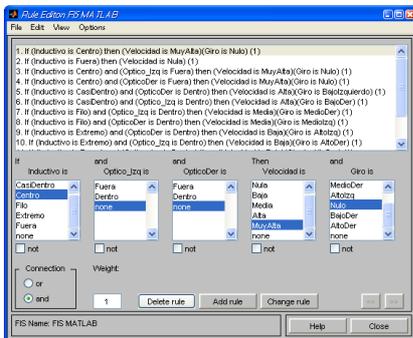


Asi mismo en el Membership Function Editor se agregaran las salidas del sistema.

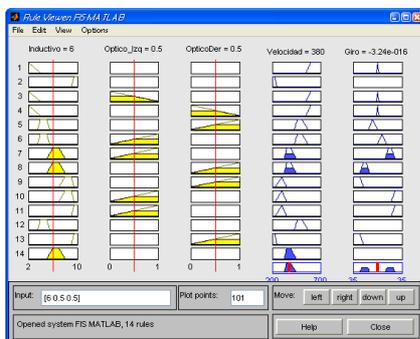
Salida Velocidad: Consta de cinco conjuntos difusos, Nula, baja, media, alta y muy alta. Estos conjuntos determinan el rango de velocidad asignada.



Salida Giro: Consta de siete conjuntos difusos, cada uno de ellos determinara el grado de giro que debe efectuarse según la señal que proviene de los sensores opticos.



Las reglas seran asignadas según las normas que se quieran aplicar en el transcurso del sistema.



Se podra visualizar el comportamiento del sistema al ir asignando valores a las entradas. Solo sera una simulacion del sistema.

Tabla 3.12 Diseño del Fis con Matlab.

Fuente: [Los autores].

3.3.6 Simulink

En la figura 3.27 se muestra el modelo que describe el comportamiento del sistema, cuya función que depende de la velocidad y del ángulo de giro está definida³¹ e ingresa a Simulink a través del bloque Fcn, para que el Fis diseñado previamente funcione es necesario exportarlo hacia el workspace de Simulink desde el Fuzzy Toolbox como una variable con el nombre del archivo .fs.

El robot móvil Robotino se conecta a Matlab vía wireless, con lo cual las entradas para el control difuso en la simulación son los valores que emiten los sensores ópticos e inductivo del robot.

Además de los bloques para conexión con Robotino el modelo cuenta con indicadores para visualizar el valor de las salidas del controlador y tres bloques de tipo to workspace, que guardan en una variable los datos de la simulación para la salida deseada como un array en el orden en que se van muestreando y con lo cual se puede obtener los gráficos de estos valores desde Matlab.

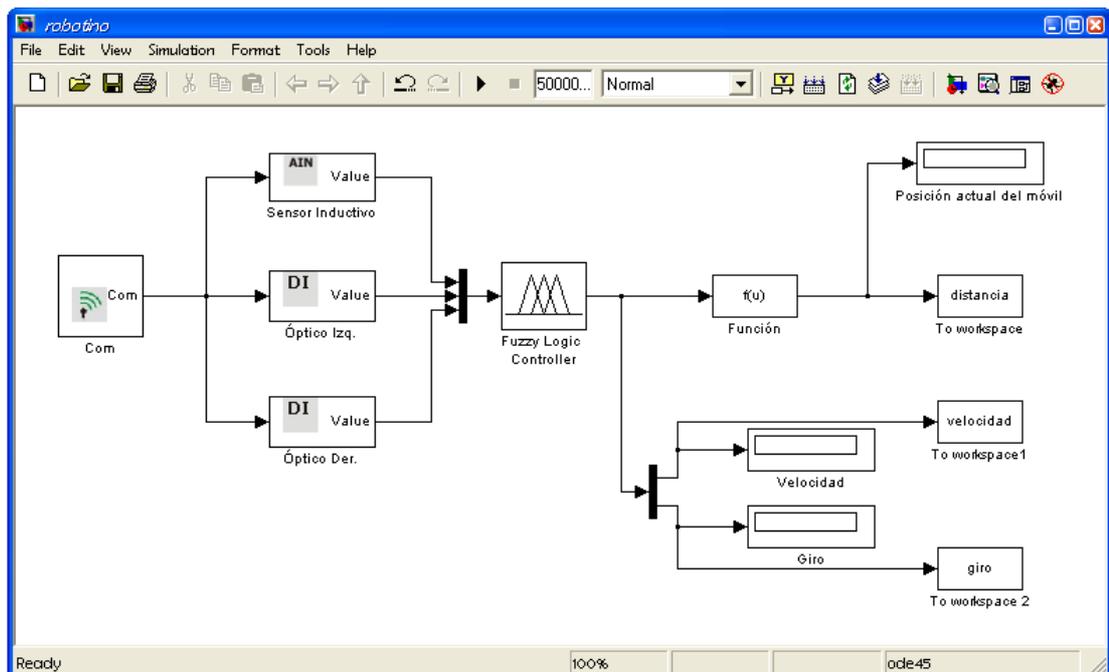
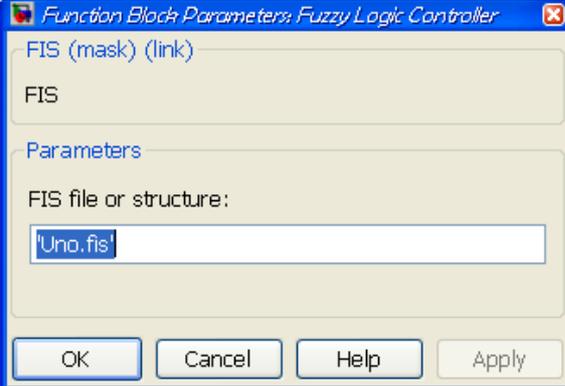
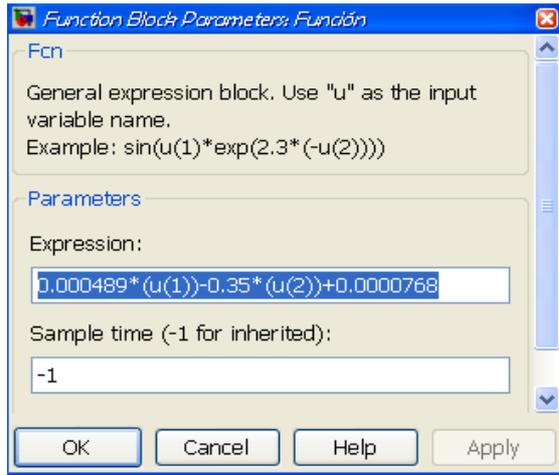
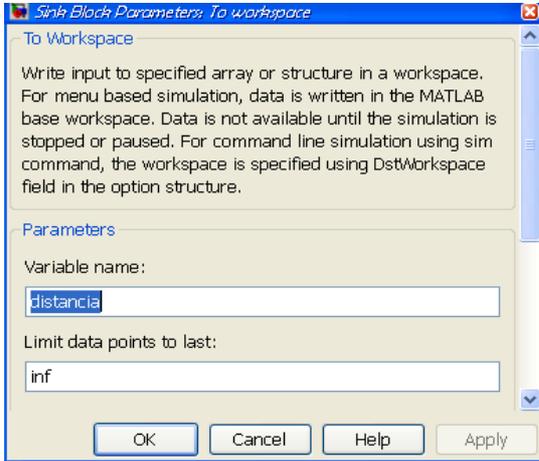


Figura 3.19 Bloques modelo matemático en Simulink.

Fuente: [Matlab 2010].

³¹Función que describe el comportamiento del sistema, Capítulo 3 página 59.

Para la simulación se escoge un tiempo de 1000 ms, ingresan constantes e indicadores, además se configuran los siguientes bloques:

<p style="text-align: center;">Fuzzy Logic Controller</p> 	<p>En este bloque se ingresa el nombre del archivo .Fis, que contiene el diseño del controlador que debe estar guardado en la misma carpeta donde se encuentra almacenado el modelo numérico.</p>
<p style="text-align: center;">Function block</p> 	<p>En el bloque Function block se ingresa la ecuación que describe el comportamiento del modelo, misma que depende de u (1) ángulo de giro y u (2) velocidad.</p>
<p style="text-align: center;">To Workspace</p> 	<p>En el bloque to workspace se almacena la salida deseada asignándole un nombre a la variable y se la guarda como un array.</p>

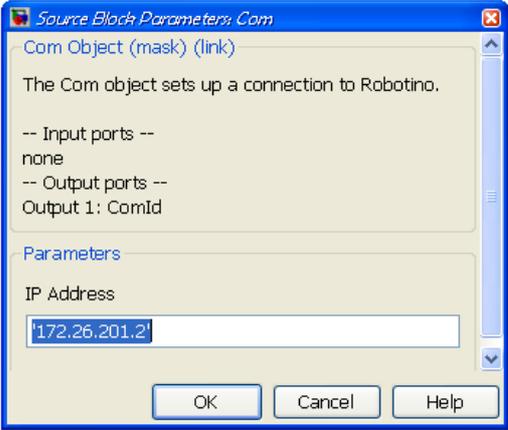
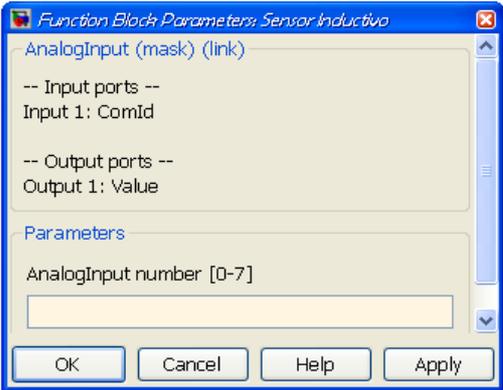
<p style="text-align: center;">Com Id</p> 	<p>En el bloque Com se configura la dirección IP para establecer la conexión entre el robot móvil Robotino y Simulink.</p>
<p style="text-align: center;">Digital Input</p> 	<p>En el bloque Digital input se configura el número de la entrada digital a la cual se encuentra conectado cada sensor óptico en la bornera del robot móvil Robotino.</p>
<p style="text-align: center;">Analog Input</p> 	<p>En el bloque Analog input se configura el número de la entrada analógica en la que se encuentra conectado el sensor inductivo en la bornera del robot móvil Robotino.</p>

Tabla 3.13 Bloques a configurar en Simulink.

Fuente: [Matlab 2010].

Desde Matlab se procede a graficar cada una de las variables que se encuentran almacenadas en los bloques de tipo to workspace, al ser un sistema dinámico se observa como los valores constantemente están cambiando para mantener la posición actual del móvil en 0.

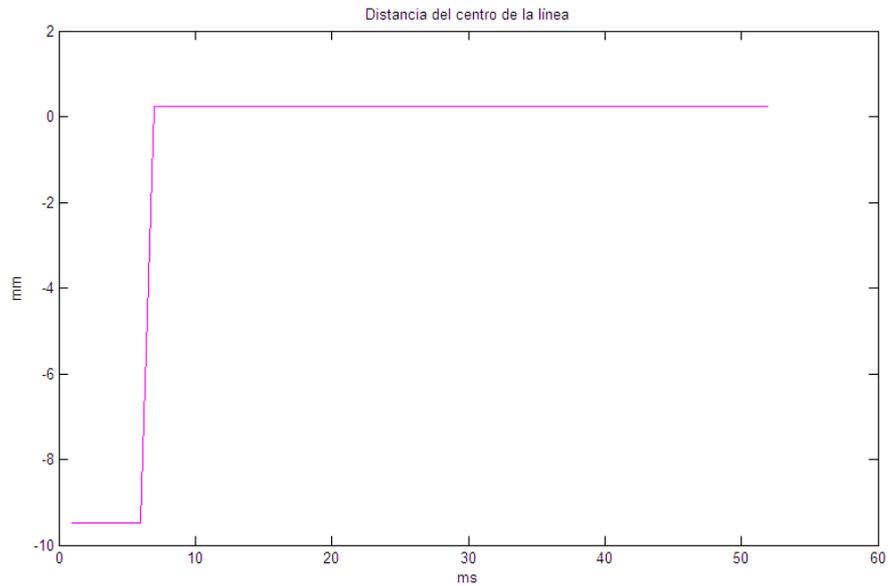


Figura 3.20 Posición actual del móvil.

Fuente: [Matlab 2010].

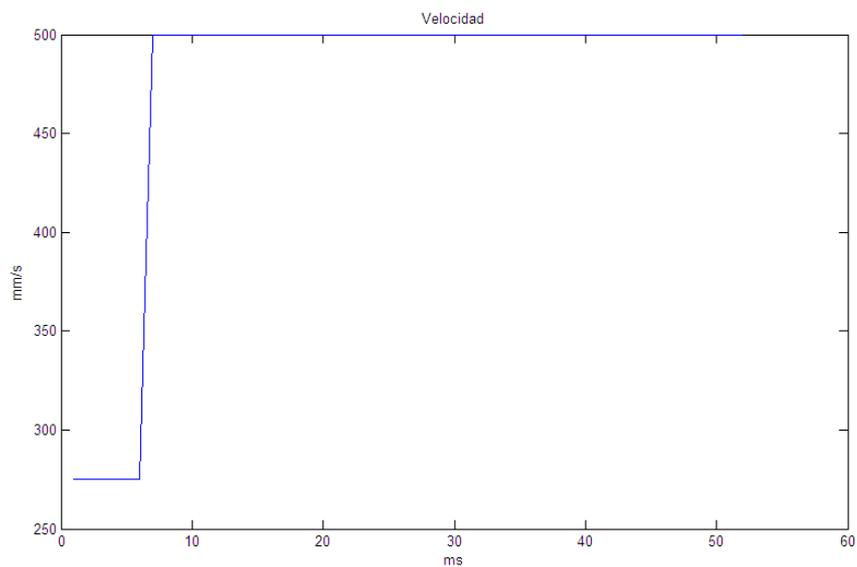


Figura 3.21 Salida velocidad en Simulink.

Fuente: [Matlab 2010].

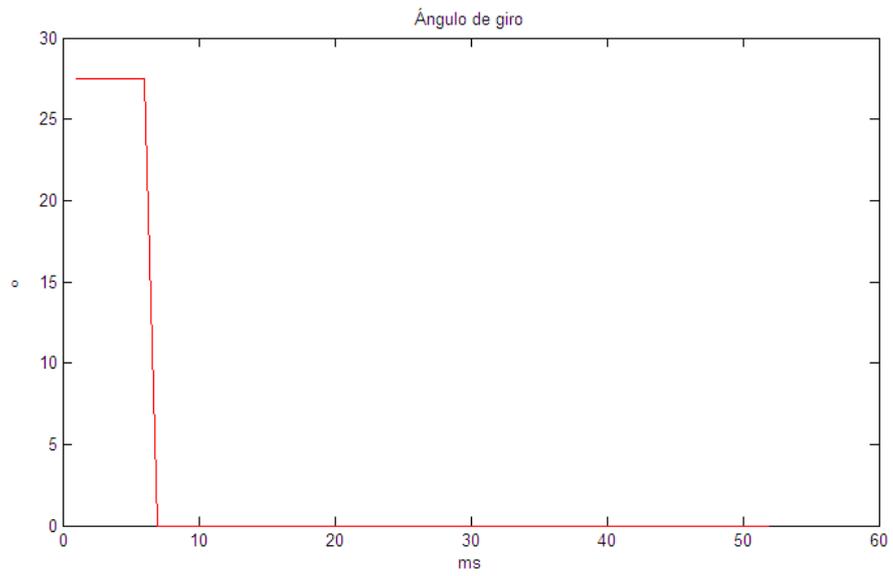


Figura 3.22 Salida ángulo de giro en Simulink.

Fuente: [Matlab 2010].

CAPITULO 4

4.1 DESARROLLO

4.1.1 Labview 2010

Para el desarrollo del presente proyecto se ha pensado en el software Labview, un software, orientado a la programación gráfica, desarrollado por National Instruments. La plataforma Labview es escalable a través de múltiples objetivos y sistemas operativos, desde su introducción en 1986 se ha vuelto un líder en la industria.

Un programa que ha sido desarrollado en Labview es denominado instrumento virtual o (VI) por ser tan parecido con los instrumentos disponibles físicamente.

Labview es considerado un sistema revolucionario de programación gráfica para ser utilizado en aplicaciones que estén involucradas con adquisición, control, análisis y presentación de datos.

Entre algunas de las ventajas que ofrece este software se puede mencionar que brinda una integración con miles de dispositivos de hardware y propone muchas librerías integradas para análisis y visualización de datos, además al contar con programación gráfica, hace de la programación una tarea menos tediosa y más amigable con personas menos involucradas en la programación por líneas de texto.

4.1.2 Herramientas de software utilizadas

4.1.2.1 Robotino Labview Driver

Robotino y sus entonos de trabajo y programación son pertenecientes a Festo, por lo tanto al no pertenecer a National Instruments es necesario para su programación en Labview el uso de una librería propia exclusiva de Robotino que permite programar sin problemas.

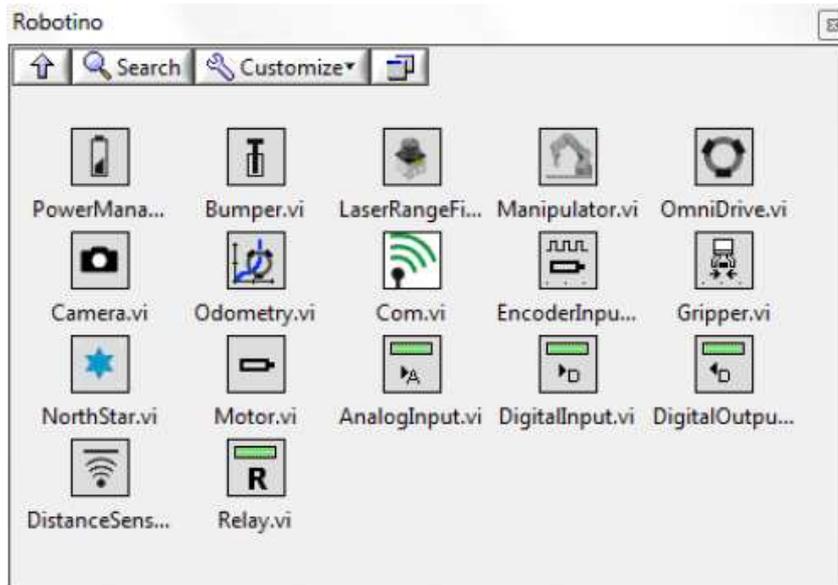


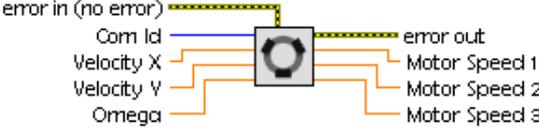
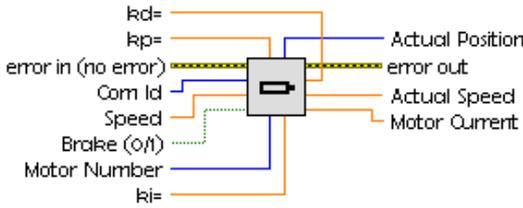
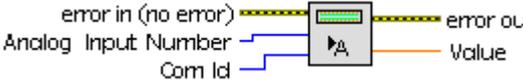
Figura 4.1 Librería de Robotino para Labview

Fuente: [Labview 2010].

En esta librería se encuentran los distintos VI's, los cuales son necesarios para la programación de los componentes de Robotino.

A continuación se detallaran los VI's que serán empleados en la programación.

<p style="text-align: center;">Com.vi</p>	<p>Antes de comenzar con la programación es necesario puntualizar que el establecer la comunicación entre Robotino y el PC es de suma importancia ya que sin la comunicación no es posible trabajar con el robot. Para ello se hace uso del VI Com.vi, el mismo que pide ingresar la dirección IP del robot y dar un valor de verdadero a Connect para habilitar la comunicación. A la salida se tiene el Com Id que será un valor igual o mayor que cero. De no establecerse la conexión el valor por default es -1.</p>
--	---

<p style="text-align: center;">OmniDrive.vi</p> 	<p>Este VI sirve para que el robot gire en su propio eje, con un ángulo y sentido asignado por el programador del robot. Dicho VI permite ingresar una velocidad en “x” y una en “y”, estas velocidades están en milímetros por segundo. Omega velocidad angular, dada en grados por segundo es otra entrada para este VI la misma que se debe establecer para que el robot recorra la trayectoria con cierto grado de giro. A la salida de este VI se tiene las velocidades de los motores, las mismas que están en RPM y serán asignadas.</p>
<p style="text-align: center;">Motor.vi</p> 	<p>Este VI dispone en la entrada Com Id necesario para la comunicación, la velocidad Speed en la cual ingresara la velocidad asignada en OmniDrive.vi. La entrada Motor Number debe asignársele el número del motor (1,2 ó 3). A las salidas de este VI se tiene la velocidad actual, la corriente del motor y la posición actual. El VI Motor.vi cuenta con un PID integrado.</p>
<p style="text-align: center;">Analog Input.vi</p> 	<p>El VI de AnalogInput.vi es el que se encarga de la programación de los sensores análogos. Tiene una entrada Analog Input Number la cual debe asignarse según la entrada analógica en la que se encuentre el sensor. La entrada Com Id necesario para la comunicación. La salida Value, un indicador para poder ver los valores que están siendo receptados por el sensor análogo, en este caso el sensor inductivo.</p>

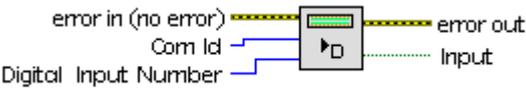
<p style="text-align: center;">Digital Input.vi</p> 	<p>El VI de DigitalInput.vi es el que se encarga de la programación de los sensores digitales. Este vi tiene una entrada Digital Input Number la cual debe asignarse según la entrada digital en la que se encuentre el sensor. Com Id, esta entrada del VI es necesaria para la comunicación. La salida del VI, Input, es un indicador para poder ver los valores que ingresan por este sensor.</p>
<p style="text-align: center;">Sumidero de error</p> 	<p>La salida a la cual irán conectadas todas las salidas de error de los VI's utilizados para indicar si existe algun error en la comunicación.</p>

Tabla 4.1 Descripción componentes de Robotino Labview driver.

Fuente: [Labview 2010].

4.1.2.2 Fuzzy Logic Toolkit

La librería de Lógica Difusa de Labview cuenta con VI's listos para su aplicación en la programación de controladores difusos.

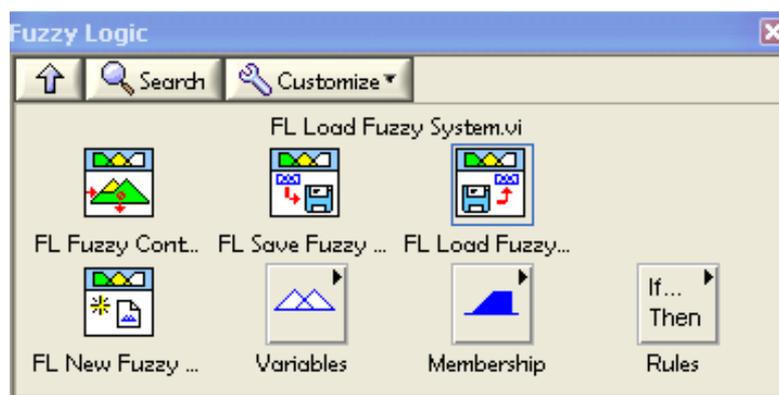


Figura 4.2 Librería de Fuzzy Logic para Labview.

Fuente: [Labview 2010].

A continuación se detalla la descripción de los VI's utilizados.

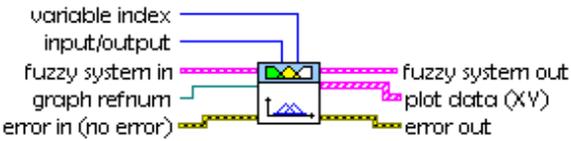
<p style="text-align: center;">FL Load Fuzzy System VI</p> 	<p>Para hacer uso de este VI, se necesita previamente el archivo .fs en el cual se encuentra el diseño del controlador.</p> <p>File Path en esta entrada se especifica la dirección en la cual está almacenado el archivo .fs este VI retorna un error de salida. Fuzzy System Out retorna la información completa del sistema difuso. Se debe conectar la salida de fuzzy System Out de este VI con la entrada fuzzy System in del siguiente VI a utilizar.</p>
<p style="text-align: center;">FL Fuzzy Controller (MIMO)</p> 	<p>Multiple-input multiple-output Sirve para implementar un controlador difuso de múltiples entradas y múltiples salidas. Cuenta con entradas las cuales son; Fuzzy System in, input values. Fuzzy System in especifica la información del control fuzzy, esta irá conectada a la salida fuzzy System Out de otro VI, de esta manera se formara el sistema multiple entrada multiple salida.</p>
<p style="text-align: center;">FL Plot Variable. vi</p> 	<p>FL Plot Variable.vi permite la visualización de las variables del control difuso. Donde se puede apreciar sus características principales.</p>

Tabla 4.2 Descripción componentes de Fuzzy Logic Toolkit.

Fuente: [Labview 2010].

4.1.2.3 Database Toolkit

La librería para el manejo de bases de datos permite trabajar con cualquier proveedor que cumpla con Microsoft ActiveX Data Object (DO9) estándar, trabaja con cualquier base de datos que cumpla con OLE DB, además permite usar instrucciones SQL con todos los sistemas de bases de datos soportados incluyendo sistemas que no sean SQL, en este caso permitirá crear una base de datos para visualizarla en Microsoft Access.

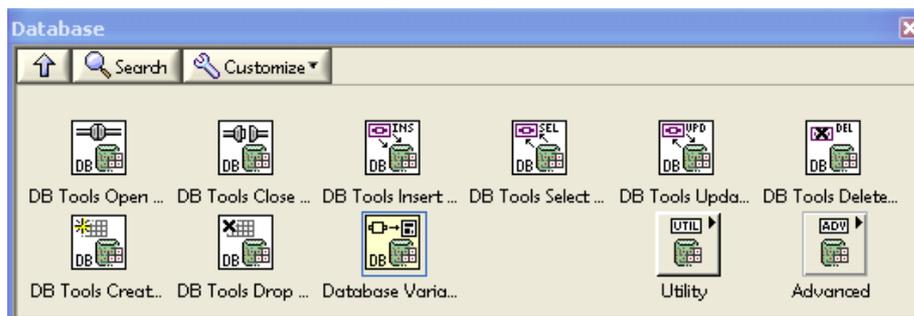


Figura 4.3 Librería de Database para Labview.

Fuente: [Labview 2010].

A continuación se detalla la descripción de los VI's utilizados.

<p style="text-align: center;">DB Tools Open Connection VI</p>	<p>Abre una conexión de base de datos utilizando la ruta de la información de conexión y devuelve una referencia de conexión. La ruta de la información de conexión es la ruta del archivo que almacena la información de conexión.</p>
<p style="text-align: center;">DB Tools Insert Data VI</p>	<p>Inserta una nueva fila en la base de datos identificado por la referencia de conexión. A través de la entrada columns se inserta las columnas y su nombre a mostrar en la base de datos y a través de la entrada table se ingresa el nombre del archivo.</p>

<p>DB Tools Close Connection VI</p> 	<p>Cierra una conexión de base de datos mediante la destrucción de su referencia de conexión a la que se encuentre asociada.</p>
<p>Connection Information</p> 	<p>Este es el directorio absoluto, que se encarga de guardar la información de conexión.</p> <p>Crea el enlace de datos entre Microsoft y labview, mismo que tiene una extensión .udl.</p>

Tabla 4.3 Descripción componentes de Database Toolkit.

Fuente: [Labview 2010].

4.2 IMPLEMENTACIÓN DEL PROGRAMA

4.2.1 Conexión entre Robotino y Labview

Para el control del robot móvil Robotino mediante Labview se hace uso de la librería propia de Robotino, en el diagrama de bloques para su control es necesario utilizar Sub-VIs tales como Com.vi, AnalogInput.vi, DigitalInput.vi, OmniDrive.vi y Motor.vi, mismos que se detallan en el literal 4.1.2.1.

Para que no exista ningún tipo de error de comunicación entre Robotino y Labview es necesario tanto Com.vi así como que el error atraviese todos los Vis usados para el control de Robotino, los errores del anterior Sub-Vi son llevados al Vi posterior hasta llegar a un sumidero de error.

El Sub-Vi lleva el nombre del host y se añade dos entradas de tipo booleano conectar y desconectar, una vez ingresado el nombre del host su utiliza conectar para establecer la conexión con Robotino y desconectar para terminar la conexión.

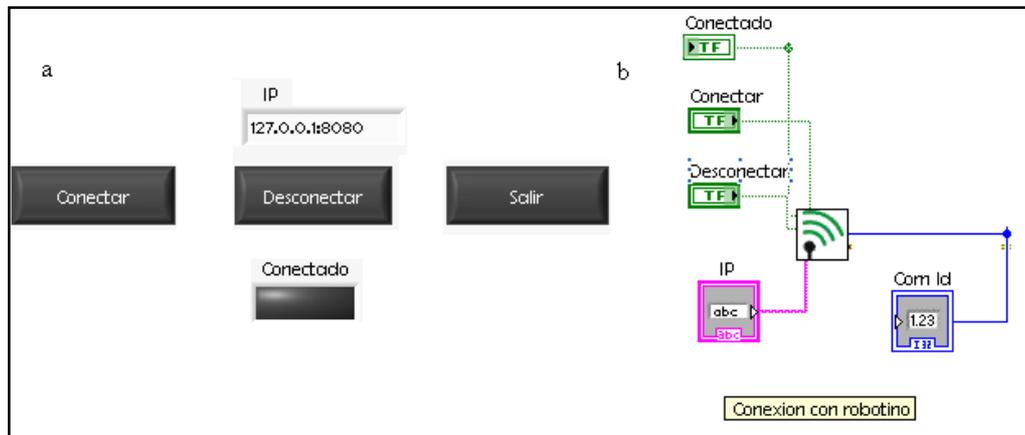


Figura 4.4 Comunicación entre Labview y Robotino (a) Panel frontal, (b) Diagrama de bloques.

Fuente: [Labview 2010].

Una vez establecida la conexión se devuelve un Com Id este identificador es un número mismo que es utilizado por los otros Vis para comunicarse con Robotino.

El sensor Inductivo está ubicado en la parte delantera de Robotino, el Sub-VI AnalogInput.vi debe de estar configurado para recibir la lectura proveniente de la entrada 0 que es la entrada a la que está conectado el sensor inductivo en la interface de entradas / salidas del robot, de igual manera con los sensores ópticos los Sub-VIs DigitalInput.vi se encuentran conectados en las entradas 0 para el sensor óptico izquierdo y 2 para el sensor óptico derecho dentro de la interface de entradas / salidas.

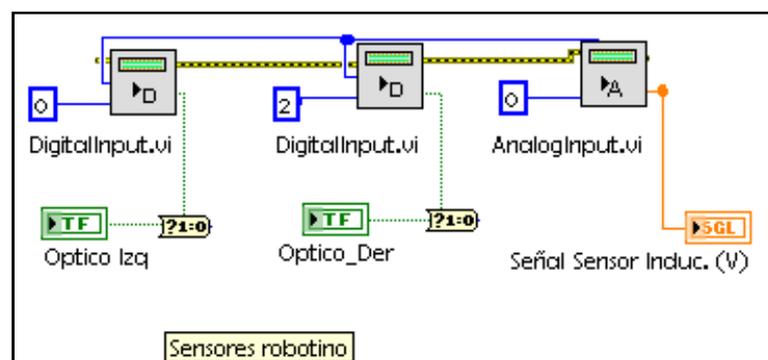


Figura 4.5 Conexión de los Sub-VIs AnalogInput.vi y DigitalInput.vi

Fuente: [Labview 2010].

El sub-Vi Omnidrive.vi recibe los parámetros de la entrada Velocidad tanto en X como en Y, para que el robot se mueva siempre hacia adelante se ingresa la velocidad en X, además recibe el valor de omega que es el ángulo que requiere el robot para girar en la dirección indicada, estas entradas provienen del controlador difuso.

Los Sub-Vis de cada motor se deben de identificar con el número de motor (0 motor uno, 1 motor dos y 2 para el motor tres) y reciben como parámetros de entrada las salidas del Sub-VI Omnidrive.vi.

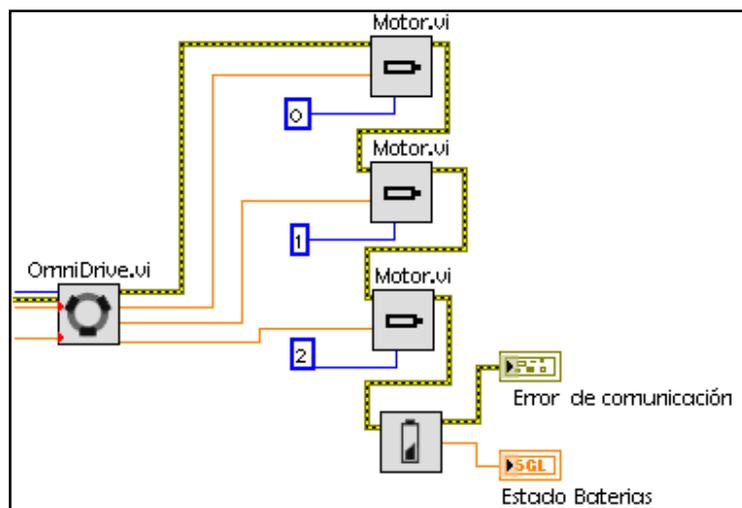


Figura 4.6 Conexión de los Sub-VIs Omnidrive.vi y Motor.vi

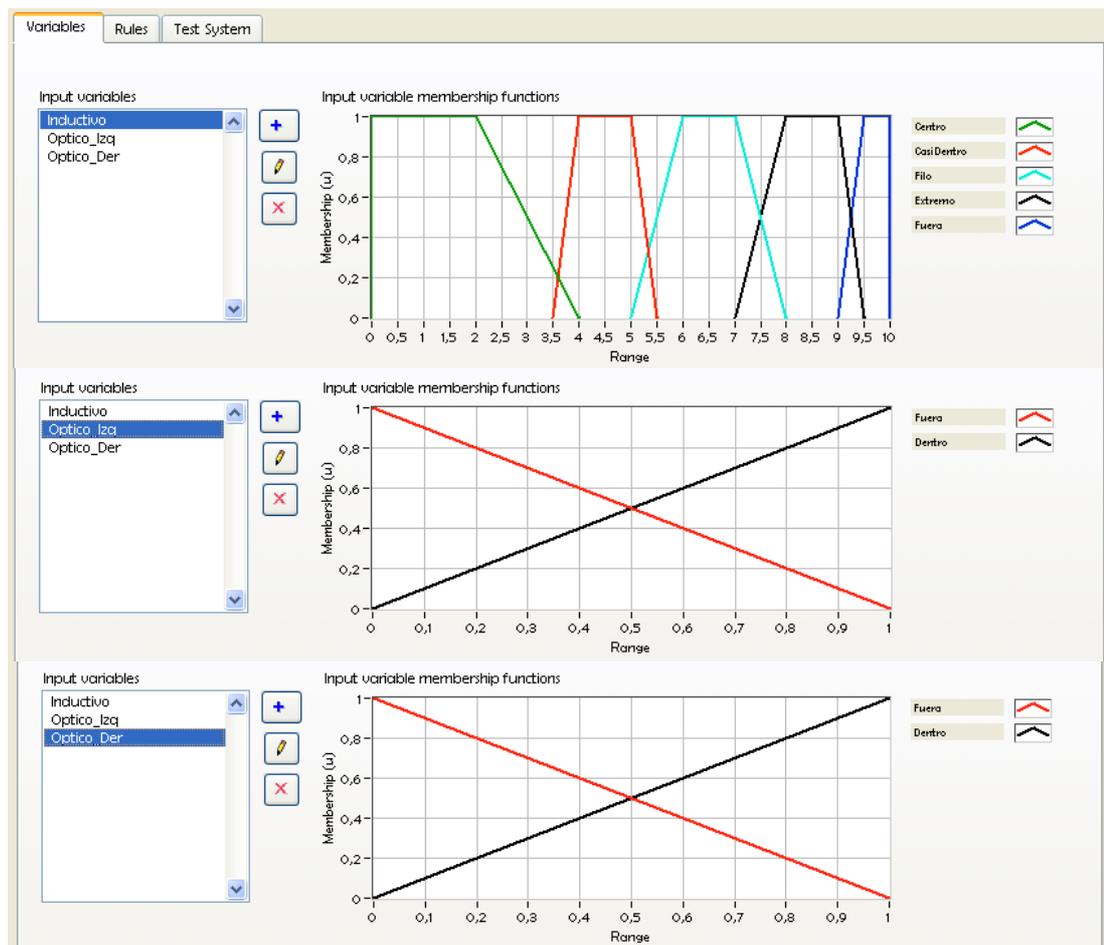
Fuente: [Labview 2010].

4.2.2 Implementación del controlador difuso

Con motivo de proporcionar un control de velocidad al robot móvil Robotino el diseño del mismo consiste de la interfaz que se desarrolla mediante la plataforma Labview, así como el control difuso con morfología MIMO (Múltiples entradas múltiples salidas), el cual contempla tres entradas las cuales se reciben del sensor inductivo y dos sensores ópticos con sus dos salidas de control tanto la velocidad como el ángulo de giro.

El desarrollo del diseño se realizará en su primera etapa obteniendo los valores que entregan los sensores mediante pruebas preliminares y de esta forma agregar estos valores a los conjuntos difusos de entrada (variables lingüísticas³²).

Para el diseño del controlador difuso se utiliza el “Fuzzy System Designer” propio de Labview para crear las funciones de membrecía de cada una de las variables lingüísticas tanto de entrada como de salida, las cuales se pueden observar en la siguiente figura.



³²Capítulo II Variable lingüística, página 5

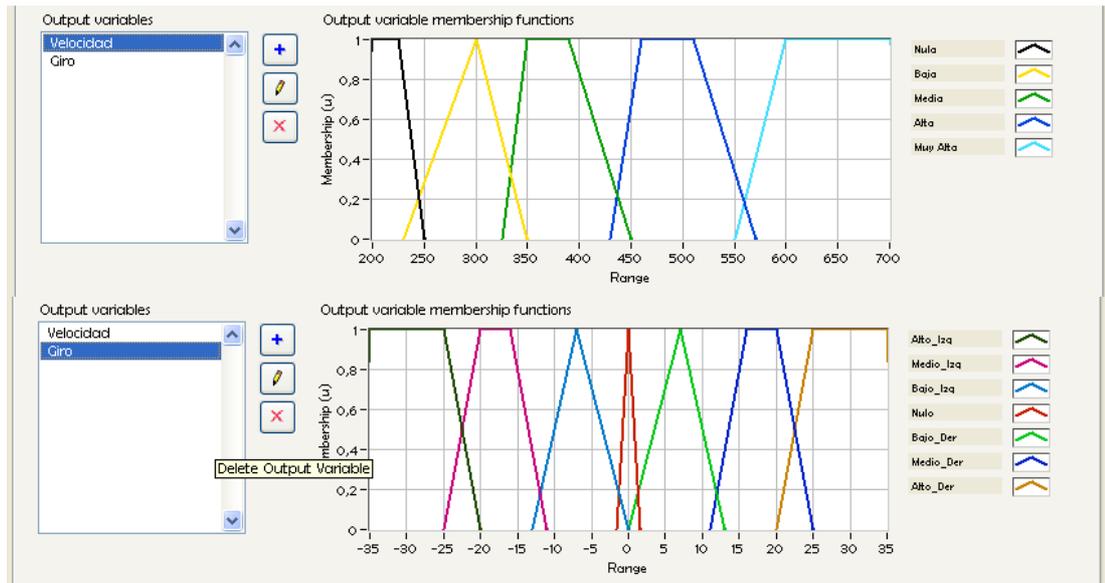


Figura 4.7 Variables lingüísticas y conjuntos difusos para entradas y salidas.

Fuente: [Labview 2010].

Una vez creadas los conjuntos difusos se genera la base de reglas difusas³³, la cual es editada de acuerdo al comportamiento que presenta el robot para de esta forma obtener una respuesta del controlador adecuada según las condiciones de trabajo estimadas para la aplicación.

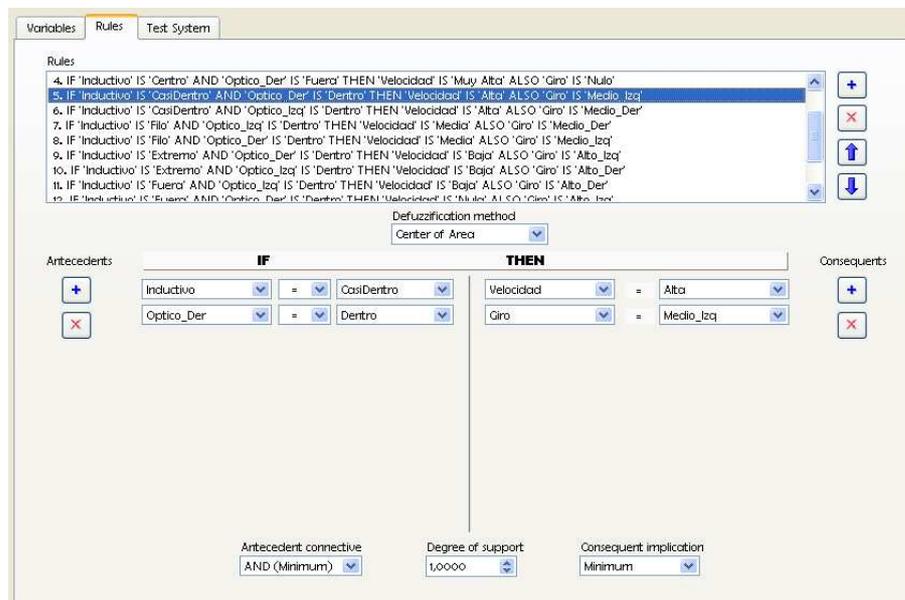


Figura 4.8 Reglas difusas.

Fuente: [Labview 2010].

³³ Capítulo II Base del conocimiento, página 15

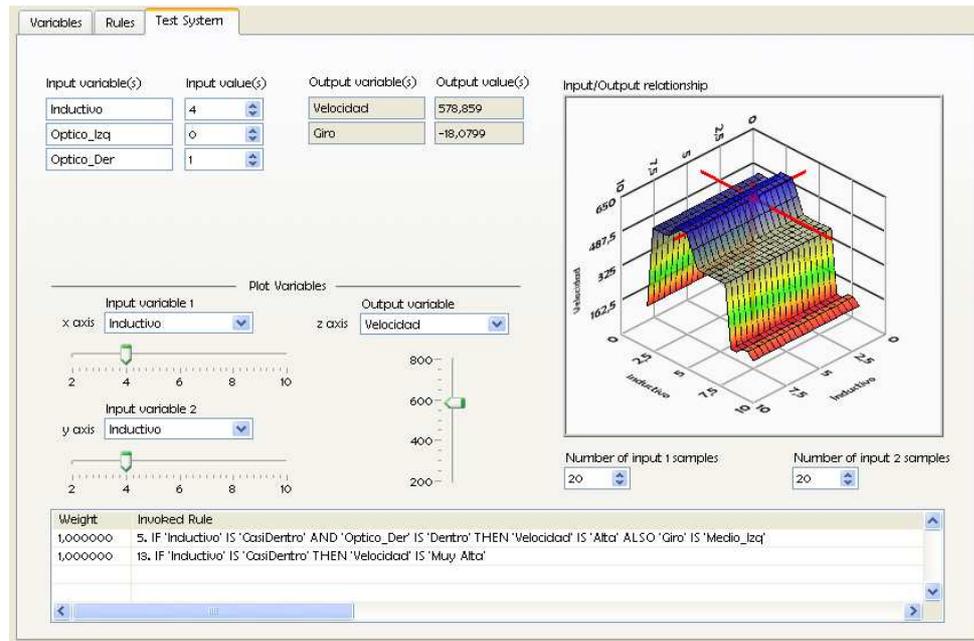


Figura 4.9 Diagnóstico del sistema.

Fuente: [Labview 2010].

A continuación se detalla a través de un diagrama de flujo la implementación del programa.

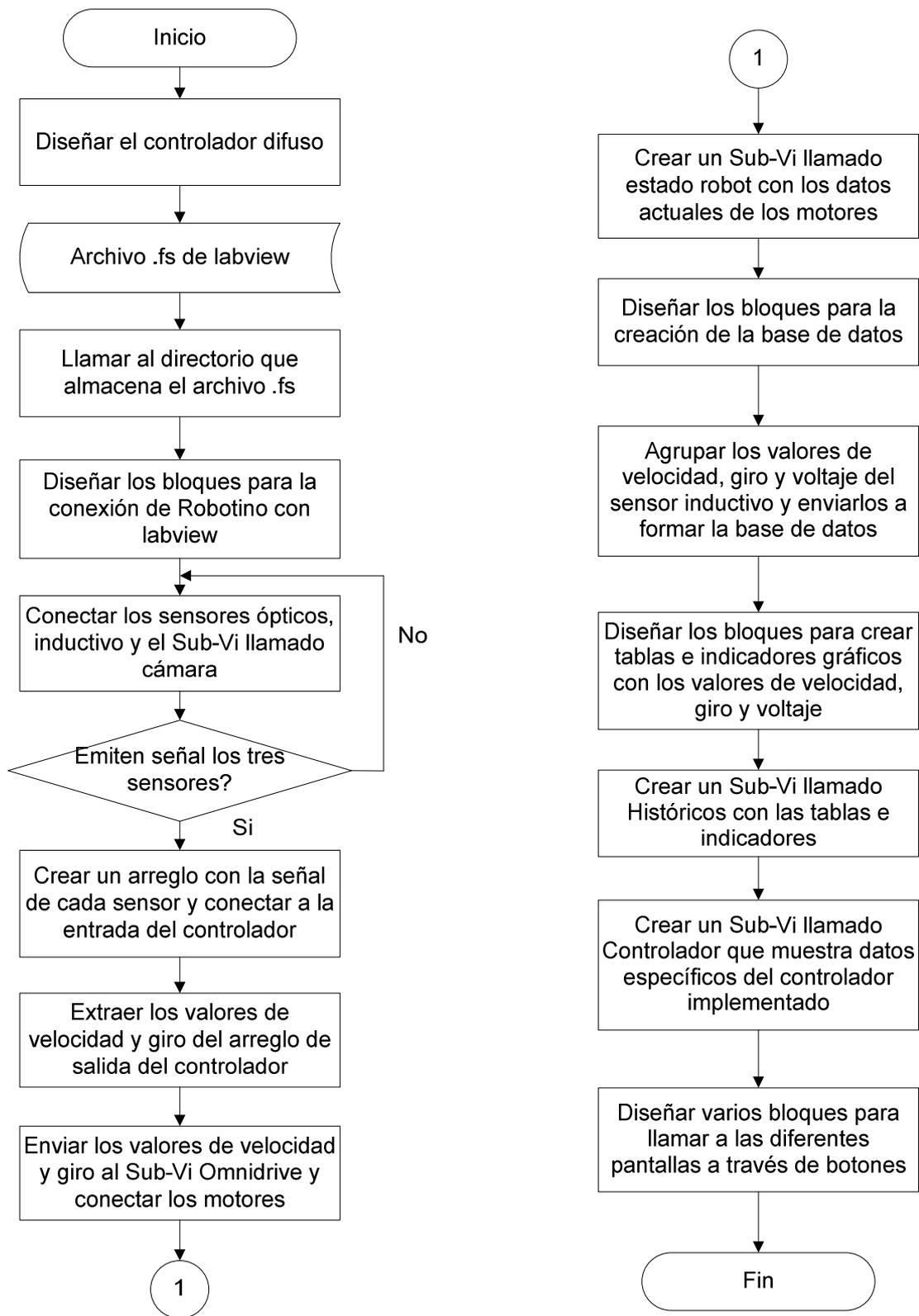


Figura 4.10 Diagrama de flujo de la implementación del programa.

Fuente: [Los autores].

4.2.3 Pantallas

La aplicación se desarrolla en varias pantallas, en cada una de estas pantallas se ejecuta una parte de la programación realizada en el control del sistema.

La secuencia para el despliegue de las pantallas se presenta a continuación:

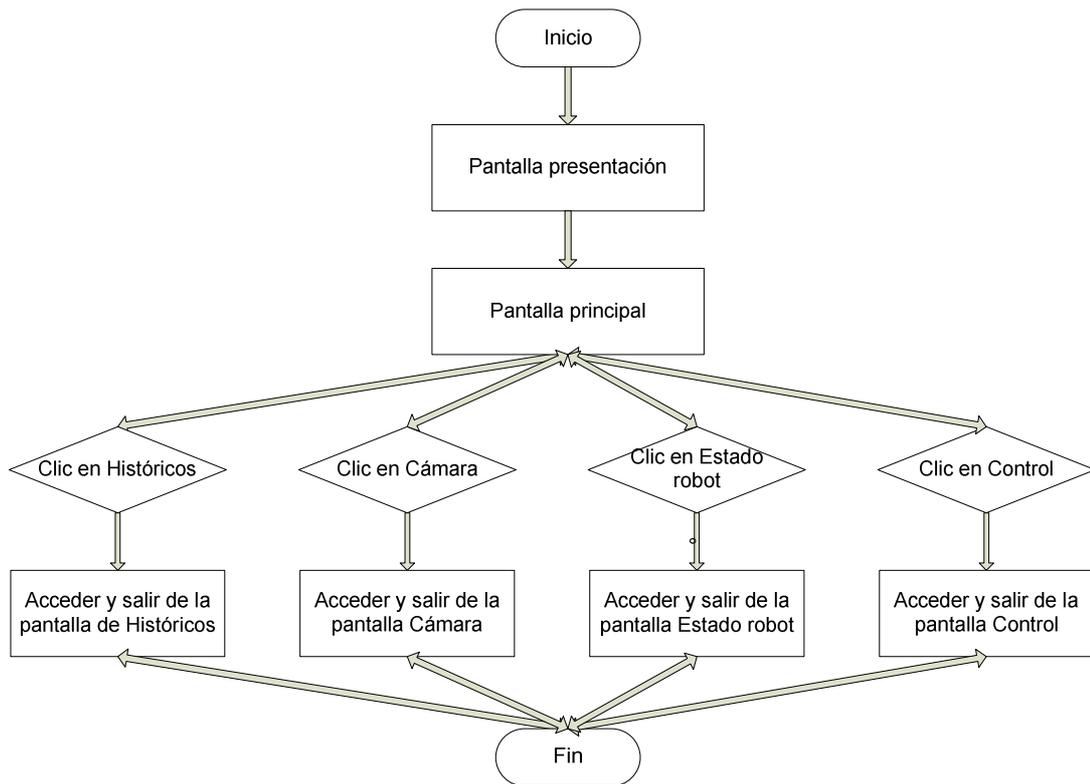


Figura 4.11 Secuencia de despliegue de las pantallas.

Fuente: [Los autores].

4.2.3.1 Pantalla presentación

La pantalla presentación es la primera en aparecer ya que esta es la introducción al programa en la que se muestra el título del proyecto, los autores así como el director del mismo, esta pantalla corre una sola vez dentro del programa ya que al dar clic sobre el botón ingresar esta se cierra dando paso a la pantalla principal del programa.



Figura 4.12 Pantalla Presentación.

Fuente: [Los autores].

El diagrama de flujo correspondiente para la pantalla Presentación se muestra a continuación:

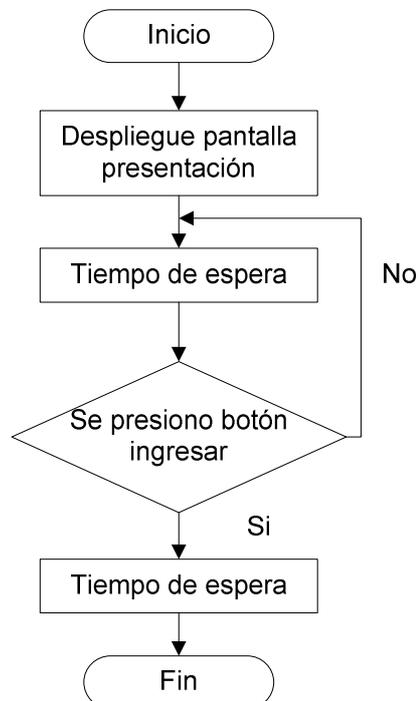


Figura 4.13 Secuencia de operación de la pantalla presentación.

Fuente: [Los autores].

4.2.3.2 Pantalla Principal

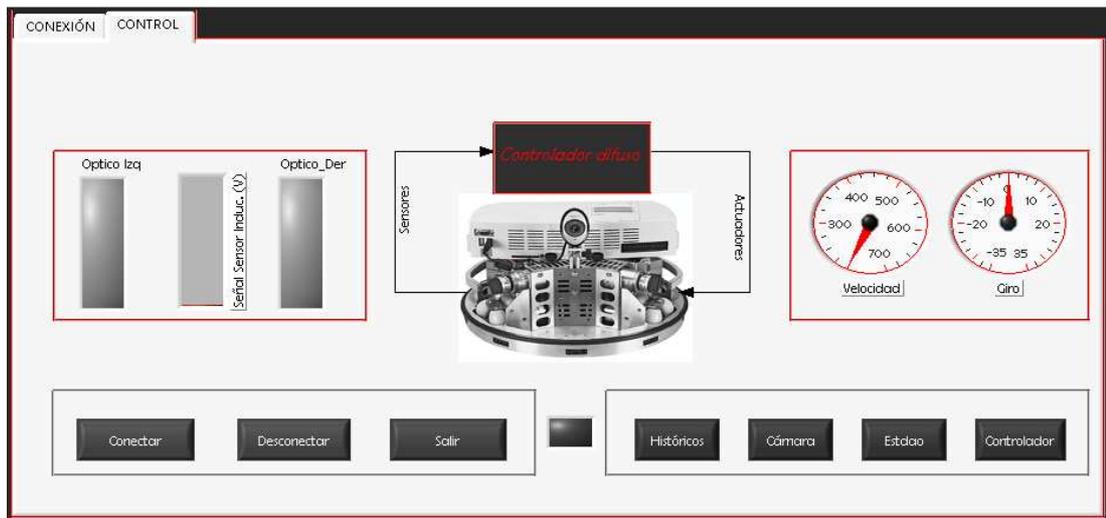
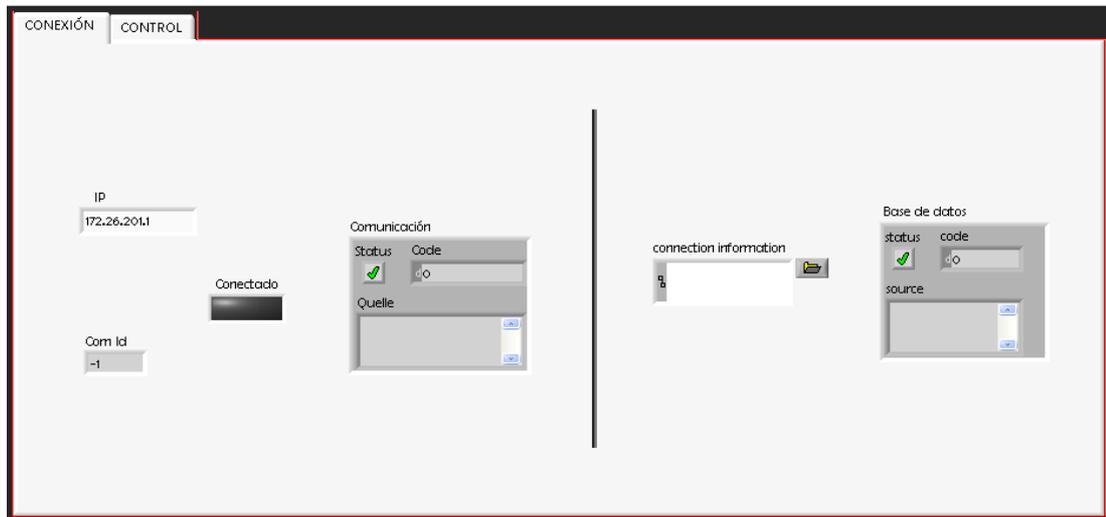


Figura 4.14 Pantalla principal.

Fuente: [Los autores].

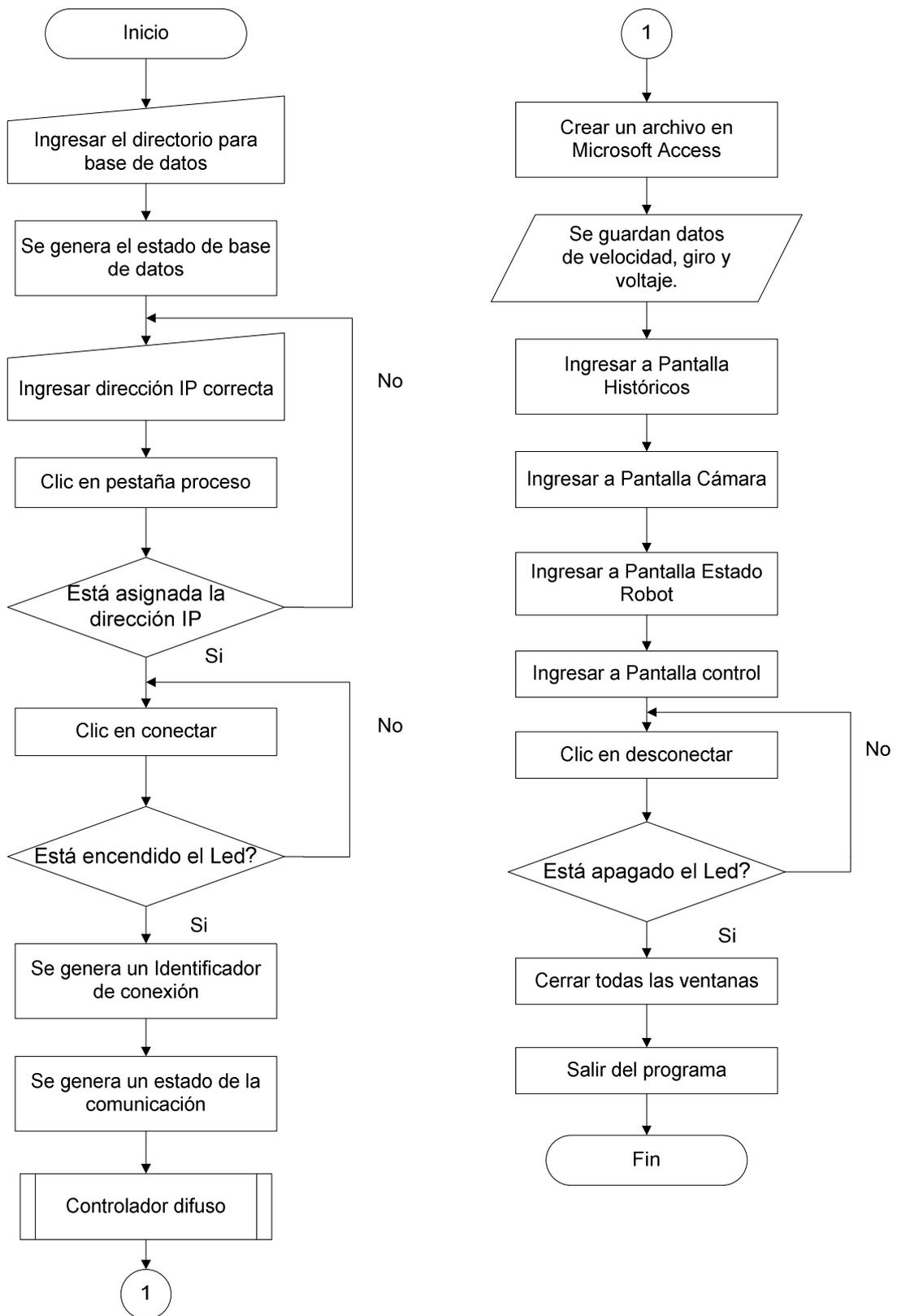


Figura 4.15 Secuencia de operación para la pantalla principal.

Fuente: [Los autores].

4.2.3.3 Pantalla de Históricos

En la pantalla de históricos se accede a los datos generados por el controlador difuso así como los datos que genera el sensor inductivo además de visualizar el comportamiento de estos.

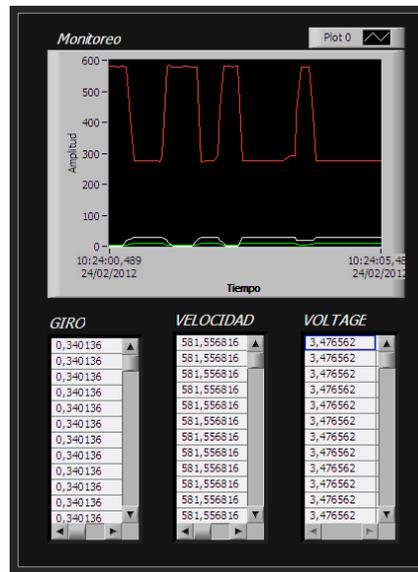


Figura 4.16 Pantalla de históricos.

Fuente: [Los autores].

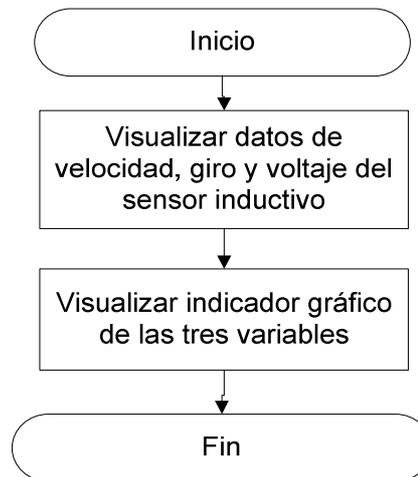


Figura 4.17 Secuencia de operación para la pantalla de históricos.

Fuente: [Los autores].

4.2.3.4 Pantalla Cámara

Dentro de esta pantalla se ingresa a visualizar la imagen que emite la cámara web conectada al robot móvil Robotino.



Figura 4.18 Pantalla cámara.

Fuente: [Los autores].

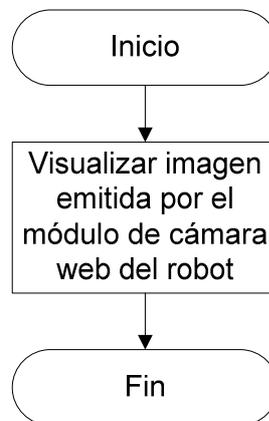


Figura 4.19 Secuencia de operación para la pantalla Cámara.

Fuente: [Los autores].

4.2.3.5 Pantalla Estado del robot

Esta pantalla permite visualizar las distintas velocidades que presentan cada uno de los tres motores, mismas que van cambiando en tiempo real conforme lo indique el

controlador difuso, además de esto se puede visualizar la carga presente en las baterías.

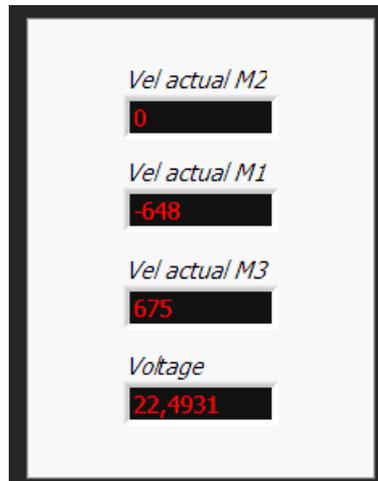


Figura 4.20 Pantalla estado del robot.

Fuente: [Los autores].

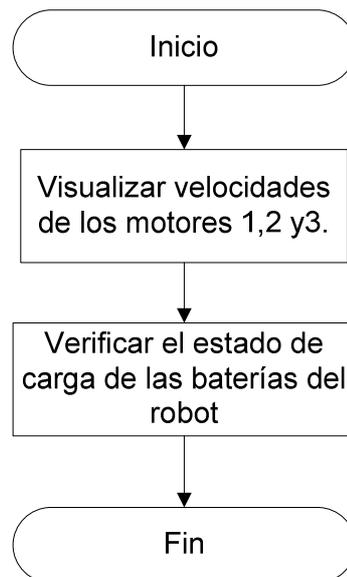


Figura 4.21 Secuencia de operación para la pantalla Estado del robot.

Fuente: [Los autores].

4.2.3.6 Pantalla Controlador

Esta pantalla consta de tres pestañas en las cuales se puede apreciar las variables de entrada y salida del controlador así como sus respectivos conjuntos difusos, además se puede visualizar las reglas del controlador.

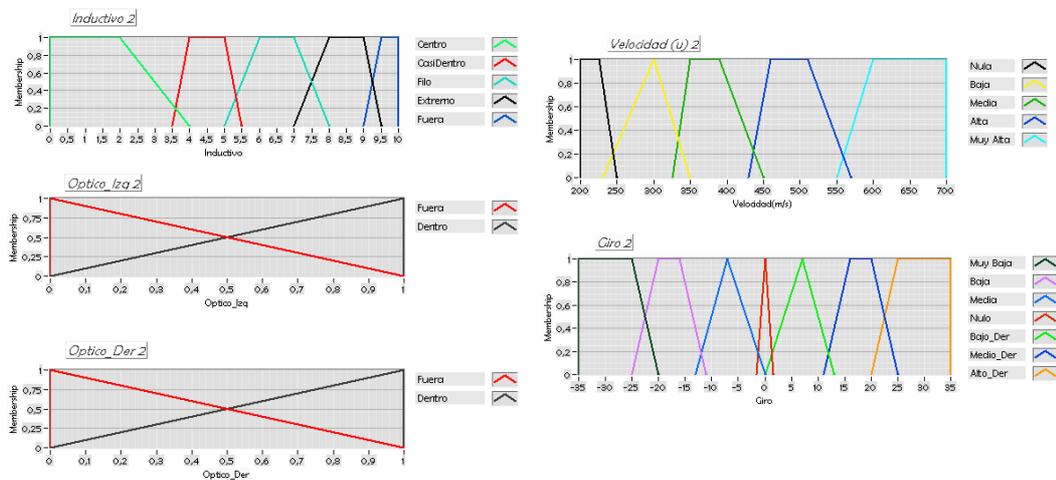
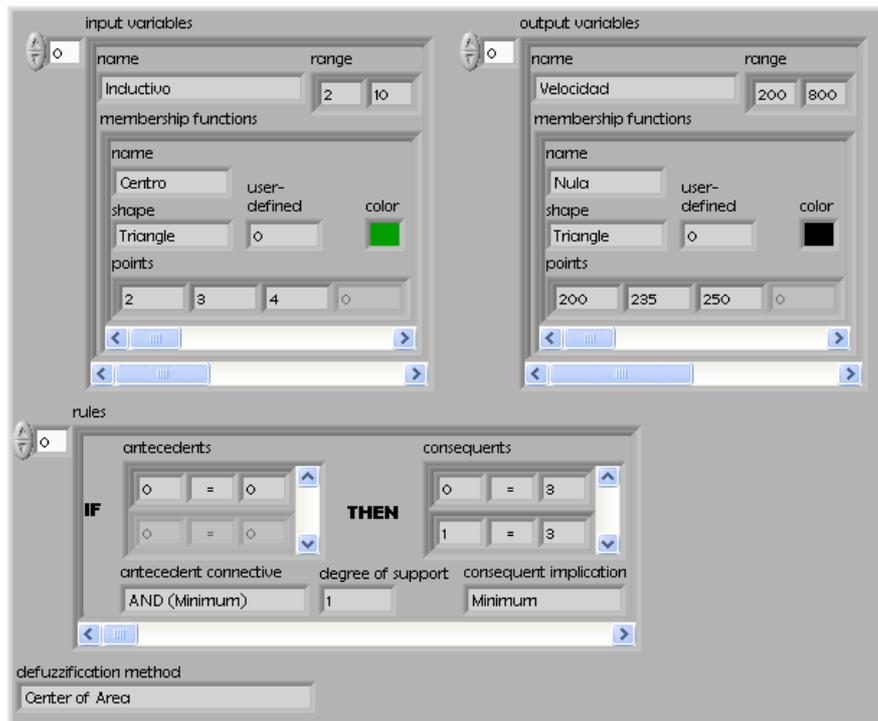


Figura 4.22 Pantalla Controlador.

Fuente: [Los autores].

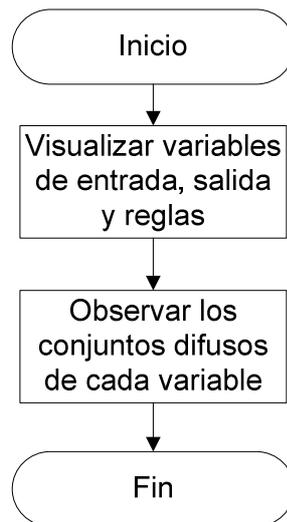


Figura 4.23 Secuencia de operación para la pantalla Controlador.

Fuente: [Los autores].

4.3 ANÁLISIS Y RESULTADOS

4.3.1 Recorridos seguidos mediante un controlador convencional

Los recorridos seguidos por el robot móvil Robotino fueron línea recta, curva cerrada, curva abierta y curva parcialmente abierta, se tiene como dato importante que el robot no avanza con velocidades menores o iguales a 250 mm/s, por esta razón se parte desde 260 mm/s como velocidad de inicio, además de esto cuando el robot sobrepasa la velocidad de 350 mm/s el seguimiento de la línea del mismo es defectuoso perdiendo pista en muchas de las curvas que se presentan.

Las velocidades y ángulos para cada una de las pruebas dentro del seguimiento sin control difuso son colocados manualmente en la programación.

Ángulo	Velocidad	Funciona	No funciona	Observaciones
10	260	X		En línea recta
20	260	X		Curva abierta
30	260	X		En curva abierta. En línea recta presenta un cascabeleo notable.
40	260	X		En curva abierta. No sigue línea recta.
10	300		X	Pierde pista
20	300		X	Pierde pista
30	300		X	Pierde pista
40	300	X		Curva abierta, Cascabelea en línea recta.
10	350		X	Solo gira no sigue línea.
20	350	X		Línea recta con cascabeleo.
30	350	X		Línea recta con cascabeleo
40	350		X	No sigue línea.
50	350	X		Solo curva abierta
50	400	X		Curva abierta
60	500	X		Curva abierta

Tabla 4.4 Fidelidad de seguimiento del móvil a distintos valores de velocidad y ángulo de giro.

Fuente: [Los autores].

4.3.2 Recorrido seguido usando el controlador difuso

En el recorrido que realiza el robot con el controlador difuso se puede observar como los valores para la velocidad y para el giro van cambiando de forma dinámica, además cuenta con un acondicionamiento de la señal del sensor inductivo para visualizar la distancia a la que el robot se encuentra alejado del centro de la línea.

Ángulo	Velocidad	Voltaje	Distancia del centro	Observaciones
-19,256	441,104	6.680	13.41	Curva semi abierta sin cascabeleo.
-24,585	351,087	7.852	11.81	Curva cerrada sin cascabeleo.
-20,581	423,190	7.227	11.51	Curva semi abierta sin cascabeleo.
-18,698	441,104	6.600	1.85	Curva semi abierta sin cascabeleo.
-13,256	520,173	5.352	9.68	Curva abierta sin cascabeleo.
-8,857	578,859	4.258	8.62	Línea recta sin cascabeleo.
-6,44	581,497	3.750	10.90	Línea recta sin cascabeleo.
-12,854	543,296	5.273	9.68	Curva abierta sin cascabeleo.
-18,698	441,104	6.602	8.62	Curva semi abierta sin cascabeleo.
-28,65	282,277	9.466	6.03	Curva cerrada sin cascabeleo.
-30,987	226,595	9.961	3.90	Curva cerrada sin cascabeleo.
-6,651	581,223	3.789	2.92	Línea recta sin cascabeleo.
-8,857	578,859	5.00	5.88	Línea recta sin cascabeleo.
-6,44	581,497	3.750	8.47	Línea recta sin cascabeleo.
-28,65	282,277	9.466	14.04	Curva cerrada sin cascabeleo.

Tabla 4.5 Recorrido seguido usando el controlador difuso.

Fuente: [Los autores].

Para obtener una ecuación que describa la relación entre el voltaje del sensor inductivo y la distancia a la que el móvil se encuentra de la línea se utilizó el método de ajuste de mínimos cuadrados con el conjunto de datos que se presentan en la tabla 4.5 dando como resultado la siguiente ecuación:

$$d = -4.42 + 1.96 * v$$

Esta ecuación describe la distancia a la que el móvil se encuentra del centro de la línea en función del voltaje del sensor inductivo.

4.3.3 Valores de velocidad y ángulo de giro obtenidos en el diseño del controlador en Matlab y el controlador implementado en Labview

Para obtener los valores de salida del controlador tanto del diseño en Matlab como del controlador implementado en Labview se introducen varios datos en las entradas

del controlador para obtener varias combinaciones de las mismas y de esta forma obtener distintas salidas de velocidad y ángulo de giro.

Matlab				
Entradas			Salidas	
Inductivo	Opt_Izq	Opt_Der	<i>Velocidad</i>	<i>Giro</i>
2,025	0	0	584	0
3	0	0	579	0
5	0	0	494	0
7,6	0	0	384	0
2,89	1	0	579	0
3,75	1	0	559	5,86
4,96	1	0	494	6,67
6,23	1	0	380	18
8,75	1	0	293	27,5
2,5	0	1	581	0
3,25	0	1	580	0
5,65	0	1	382	-18
7,8	0	1	319	-25,52
9,25	0	1	274	-27,5

Tabla 4.6 Valores tomados del diseño del controlador en Matlab.

Fuente: [Los autores]

Labview				
Entradas			Salidas	
Inductivo	Opt_Izq	Opt_Der	<i>Velocidad</i>	<i>Giro</i>
2,025	0	0	582,47	0
3	0	0	578,859	0
5	0	0	493,792	0
7,6	0	0	382,854	0
2,89	1	0	579,199	0
3,75	1	0	559,8	5,717
4,96	1	0	493,792	6,56348
6,23	1	0	380,249	17,96
8,75	1	0	293,329	27,5064
2,5	0	1	581,497	0
3,25	0	1	580,077	0
5,65	0	1	382,361	-18,138
7,8	0	1	318,851	-25,645
9,25	0	1	274,275	-27,523

Tabla 4.7 Valores tomados del diseño del controlador en Labview.

Fuente: [Los autores].

4.3.4 Valores de velocidad y ángulo de giro obtenidos en la simulación

Para obtener los valores de salida del controlador en la simulación (Simulink), se introduce varios datos para tener diferentes entradas y a la vez tener diferentes valores en las salidas.

En Simulink los datos de entrada son ingresados mediante las propias entradas para los sensores que posee el robot móvil y a través del modelo numérico implementado se obtiene las salidas del controlador y la distancia a la que el móvil se encuentra alejado del centro de la línea.

Simulink				
Entradas			Salidas	
Inductivo	Opt_Izq	Opt_Der	<i>Velocidad</i>	<i>Giro</i>
2,025	0	0	584	0
3	0	0	578.8	0
5	0	0	493.7	0
7,6	0	0	384.3	0
2,89	1	0	579.2	0
3,75	1	0	559.3	5,861
4,96	1	0	493.7	6,667
6,23	1	0	380.2	18
8,75	1	0	293.3	27,5
2,5	0	1	581	0
3,25	0	1	579.7	0
5,65	0	1	382.4	-18
7,8	0	1	319.1	-25,55
9,25	0	1	274.2	-27,5

Tabla 4.8 Datos obtenidos de la simulación.

Fuente: [Los autores].

4.3.5 Comparación entre el recorrido utilizando controlador FIS Vs recorrido utilizando controlador convencional

Para realizar esta prueba se procede a medir el tiempo que tarda el robot móvil Robotino en recorrer la pista diseñada de principio a fin.

La finalidad de esta prueba es demostrar que el controlador difuso puede proporcionar algún tipo de ventaja, respecto al tiempo que tarda el robot en cumplir dicho recorrido, sin el controlador difuso implementado.

A continuación se presentarán una serie de datos y cálculos estadísticos que darán respuesta a la duda planteada.



Figura 4.24 Pista diseñada para pruebas de recorrido.

Fuente: [Los autores].

4.3.5.1 Tiempos de recorrido seguido usando el controlador difuso

Al robot móvil Robotino se lo hizo recorrer la pista, usando el controlador difuso, diez veces. Cada vuelta recorrida fue cronometrada.

La velocidad que utiliza el controlador es variable y va desde los 200 mm/s hasta los 700 mm/s, de igual forma el valor del ángulo de giro es variable y va desde los -35° hasta los 35° y los datos obtenidos son los siguientes:

VUELTA	TIEMPO	
	minutos	segundos
1	1	14.56
2	1	13.78
3	1	11.00
4	1	13.38
5	1	7.98
6	1	8.44
7	1	8.22
8	1	3.46
9	1	3.75
10	1	4.05

Tabla 4.9 Tiempo de recorrido con control difuso.

Fuente: [Los autores].

4.3.5.2 Tiempos de recorrido seguido usando un control convencional

Se procede a cronometrar las vueltas realizadas por el móvil implementando un controlador convencional, se obtiene como resultado los siguientes datos:

Velocidad Aplicada: 280 mm/s Angulo de Giro: $\pm 20^\circ$		
VUELTA	TIEMPO	
	minutos	segundos
1	3	46.84
2	3	46.81
3	3	47.35
4	5	00.43
5	5	00.04
6	4	50.98
7	3	6.94
8	3	6.20
9	3	7.06
10	4	47.97

Tabla 4.10 Tiempo de recorrido con velocidad de 280 mm/s y ángulo de $\pm 20^\circ$ usando controlador convencional.

Fuente: [Los autores]

Velocidad Aplicada: 350 mm/s Angulo de Giro: $\pm 20^\circ$		
VUELTA	TIEMPO	
	minutos	segundos
1	1	54.44
2	1	56.71
3	1	56.66
4	2	1.86
5	1	55.54
6	1	57.78
7	2	12.34
8	2	13.74
9	2	12.08
10	2	12.98

Tabla 4.11 Tiempo de recorrido con velocidad de 350 mm/s y ángulo de $\pm 20^\circ$ usando controlador convencional

Fuente: [Los autores].

A continuación en la tabla 4.16 se indica el error absoluto y relativo que existió entre el tiempo tomado con el controlador FIS implementado al móvil y el tiempo tomado con el controlador convencional, estableciéndole a este último una velocidad constante de 280 mm/s.

La tabla muestra claramente las muestras tomadas en las columnas a la izquierda, seguidas por las columnas que contienen tanto las formulas como los resultados de los errores.

Tiempo Control FIS (min) X2	Tiempo Control Convencional a 280 mm/s (min) X1	Error absoluto $\varepsilon_a = x_1 - x_2 $	Eficiencia
1:15	3:47	2,32	66,8587
1:14	3:47	2,33	67,1469
1:11	3:48	2,37	68,1034
1:13	5:01	3,88	77,4451
1:08	5:01	3,93	78,4431
1:08	4:51	3,43	76,0532
1:08	3:7	1,99	64,8208
1:03	3:6	2,03	66,3398
1:04	3:7	2,03	66,1237
1:04	4:5	3,01	74,3209
Total			705,656
Eficiencia			70,565

Tabla 4.12 Eficiencia del controlador FIS en comparación a un controlador convencional.

Fuente: [Los autores].

4.3.5.3 Cálculos Estadísticos

Estadística Descriptiva para muestras tomadas con el controlador FIS implementado:

La siguiente tabla muestra los datos con los que se va a trabajar, en total se hizo recorrer al robot móvil 10 veces la pista diseñada, en la tabla mostrado como *Número de vuelta*, seguido de este valor se encuentra el tiempo que demoró el robot móvil en culminar la pista en cada vuelta, este tiempo está registrado en minutos y segundos. En la tabla denominado como *Tiempo (min) usando control FIS*.

Número de vuelta	Tiempo (min) usando control FIS
1	1:15
2	1:14
3	1:11
4	1:13
5	1:08
6	1:08
7	1:08
8	1:03
9	1:04
10	1:04

Tabla 4.13 Muestras tomadas con el controlador implementado en Labview.

Fuente: [Los autores].

Medidas de Centralización:

- Media Aritmética:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n a_i = \frac{a_1 + a_2 + \dots + a_n}{n}$$

$$\bar{x} = \frac{1,15 + 1,14 + 1,11 + 1,13 + 1,08 + 1,08 + 1,08 + 1,03 + 1,04 + 1,04}{10} = 1,088$$

- Mediana

$$me = \frac{n}{2} = \frac{10}{2} = 5$$

El número 5 indica la posición en la cual se encuentra la mediana. Es decir me=1,08

- Moda

Dato con más incidencia de repetición en la muestra obtenida.

$$Mo = 1.08$$

Medidas de Dispersión:

- Varianza:

$$s^2 = \frac{\sum (Xi - \bar{x})^2}{(n - 1)}$$

$$s^2 = \frac{1,15 - 1,088 + 1,14 - 1,088 + 1,11 - 1,088 + 1,13 - 1,088 + 1,08 - 1,088 + 1,08 - 1,088 + \dots + 1,04 - 1,088}{9}$$

$$s^2 = 6,66 * 10^{-16}$$

- Desviación Estándar

$$s = \sqrt{s^2}$$

$$s = 2.58 * 10^{-8}$$

Estadística Descriptiva para muestras tomadas con un controlador convencional implementado, usando como velocidad 280 mm/s y un ángulo de $\pm 20^\circ$.

Número de vuelta	Control convencional a 280 mm/s
1	3:47
2	3:47
3	3:48
4	5:01
5	5:01
6	4:51
7	3:7
8	3:6
9	3:7
10	4:5

Tabla 4.14 Muestras tomadas con un control convencional de 280 mm/s de velocidad y un ángulo de $\pm 20^\circ$.

Fuente: [Los autores].

Medidas de Centralización:

- Media Aritmética:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n a_i = \frac{a_1 + a_2 + \dots + a_n}{n}$$

$$\bar{x} = \frac{3,47 + 3,47 + 3,48 + 5,01 + 5,01 + 4,51 + 3,7 + 3,6 + 3,7 + 4,5}{10} = 4,05$$

- Mediana

$$me = \frac{n}{2} = \frac{10}{2} = 5$$

El número 5 indica la posición en la cual se encuentra la mediana, según los datos ordenados en forma creciente, es decir:

$$me = 3,7$$

- Moda

Dato con más incidencia de repetición en la muestra obtenida.

$$Mo = 3,47$$

Medidas de Dispersión:

- Varianza:

$$s^2 = \frac{\sum (X_i - \bar{x})^2}{(n-1)}$$

$$s^2 = \frac{3,47 - 4,05 + 3,47 - 4,05 + 3,48 - 4,05 + 5,01 - 4,05 + 5,01 - 4,05 + 4,51 - 4,05 + 3,7 + \dots + 4,05}{9} = 0,41$$

$$s^2 = 0,41$$

- Desviación Estándar

$$s = \sqrt{s^2}$$

$$s = 0,6413$$

Estadística Descriptiva para muestras tomadas con un controlador convencional implementado, usando como velocidad 350 mm/s y un ángulo de $\pm 20^\circ$.

Número de vuelta	Control convencional de 350 mm/s
1	1,55
2	1,57
3	1,57
4	2,2
5	1,56
6	1,58
7	2,12
8	2,14
9	2,12
10	2,13

Tabla 4.15 Muestras tomadas con un control convencional de 350 mm/s de velocidad y un ángulo de $\pm 20^\circ$.

Fuente: [Los autores].

Medidas de Centralización:

- Media Aritmética:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n a_i = \frac{a_1 + a_2 + \dots + a_n}{n}$$

$$\bar{x} = \frac{1,55 + 1,57 + 1,57 + 2,2 + 1,56 + 1,58 + 2,12 + 2,14 + 2,12 + 2,13}{10} = 1,9$$

- Mediana

$$me = \frac{n}{2} = \frac{10}{2} = 5$$

El número 5 indica la posición en la cual se encuentra la mediana, según los datos ordenados en forma creciente, es decir:

$$me = 1,85$$

- Moda

Dato con más incidencia de repetición en la muestra obtenida.

$$Mo = 1,57$$

Medidas de Dispersión:

- Varianza:

$$s^2 = \frac{\sum (Xi - \bar{x})^2}{(n - 1)}$$

$$s^2 = \frac{1,55 - 1,9 + 1,57 - 1,9 + 1,57 - 1,9 + 2,2 - 1,9 + 1,56 - 1,9 + 1,58 - 1,9 + 2,13 - 1,9 + \dots - 1,9}{9} = 0,092$$

$$s^2 = 0,09$$

- Desviación Estándar

$$s = \sqrt{s^2}$$

$$s = 0,3$$

La desviación estándar es una medida de dispersión la cual se emplea en estadística, ésta medida indica cuánto tienden a alejarse los valores del promedio de la muestra total.

Tipo Control	Desviación Estándar (s)	Resultado
Controlador FIS	$2.58 \cdot 10^{-8}$.	En este caso se tiene que la desviación estándar es menor en la muestra tomada utilizando el controlador FIS. Esto da como conclusión que los datos tomados mientras el robot móvil está usando el controlador FIS son menos distantes entre sí, en consecuencia son más lineales, lo cual demuestra que el error entre ellos es mínimo.
Controlador normal	0,64 y 0,3.	La desviación estándar calculada, para los datos tomados cuando, el robot móvil recorre la pista haciendo uso de un controlador convencional, es mayor, esto indica que están principalmente dispersos entre sí, generando un error más grande. Esto se debe a que con un controlador convencional, la probabilidad de que pierda la línea el robot es más alta, por lo tanto el tiempo que tardará el robot en culminar el recorrido de la pista, variará indefinidamente en cada medición.

Tabla 4.16 Resultados de la desviación estándar.

Fuente: [Los autores].

En la tabla 4.21 se describe las principales diferencias encontradas durante las pruebas realizadas entre el controlador convencional y el controlador difuso aplicado al robot móvil Robotino.

Característica	Control Difuso	Control Convencional	Observaciones
Velocidad	Variable	Constante (280 mm/s)	La velocidad con el control difuso varía según las condiciones de la pista a seguir
Giro	Variable	Constante ($\pm 20^\circ$)	El giro cambia según las curvaturas de la pista
Tiempo	Corto (1:088)	Largo (4:05)	El tiempo en el que termina su trayectoria es mucho menor con el control difuso
Seguimiento	Continuo	Discontinuo	El seguimiento con el controlador es continuo ya que acelera y desacelera según las rectas las condiciones de la pista.

Tabla 4.17 Comparación Control difuso vs. Control convencional.

Fuente: [Los autores].

CAPITULO 5

5.1 CONCLUSIONES

- En todas las pruebas realizadas, se demostró la efectividad del controlador FIS al implementárselo en un robot móvil seguidor de línea, su uso da ventaja en cuanto a tiempo de recorrido de la pista (Tiempo fis y t convencional), fineza en el seguimiento de línea y decisión correcta de la aceleración o frenado que debe efectuar el robot según el tipo de curva que está atravesando.
- El sistema implementa un controlador difuso que emite los valores de velocidad y ángulo de giro para el seguimiento de una trayectoria, mediante el robot móvil Robotino, el cual recibe por parte del controlador velocidades por encima de los 250 mm/s hasta los 700 mm/s y valores de ángulo de giro que van desde -35° hasta 35° , con el fin de que el robot se desplace por la trayectoria establecida sin perder línea y en el menor tiempo posible.
- El error relativo tanto para la velocidad como para el ángulo de giro del controlador implementado en Labview con el controlador diseñado en Matlab es menor al 3%, es decir la variación entre uno y otro es muy pequeña razón por la cual el funcionamiento del sistema está acorde a lo diseñado.
- Con un control convencional para que se complete la trayectoria, la velocidad debe ser muy baja y el ángulo de giro alto de esta forma el tiempo en el que se completa el trayecto es muy largo; conforme se eleva el valor de la velocidad se pierde fidelidad en el seguimiento de la línea ya que el móvil pierde la capacidad de tomar ciertas curvas. En contraste con el controlador difuso, al tener la capacidad de decidir los valores de velocidad y ángulo de giro debe tener en cada instante, la velocidad será elevada en trayectos rectilíneos y solamente disminuirá en curvas logrando un seguimiento continuo y con un tiempo muy corto.

- Al tener un sistema dinámico, para obtener el modelo numérico que permita describir su comportamiento, se parte de un modelo experimental y de la utilización de la regresión lineal múltiple, de esta forma se obtiene la ecuación que describe el comportamiento del sistema, el cual depende de la velocidad y del ángulo giro aplicado al móvil.
- La implementación del controlador difuso en un robot móvil, ayudó a disminuir de manera notable la inestabilidad que posee el robot al cumplir un recorrido a una velocidad que la normalmente establecida en un control convencional.
- Al realizar la comparación entre los tiempos que le toma al robot culminar la trayectoria impuesta con un controlador convencional y con un controlador Fis, se obtiene que en controlador Fis tiene una eficiencia del 69.85% en comparación al controlador convencional.

5.2 RECOMENDACIONES

- El controlador difuso con el cual trabaja el robot móvil Robotino fue diseñado e implementado en base a sus sensores (Ópticos e Inductivo), se puede obtener un control más preciso si se trabajara con el módulo de la cámara web que tiene el robot, ya que al tener la imagen se la puede procesar para obtener la coordenada en la cual se encuentra el centro de la línea (ver anexo 1) y de esta forma sirva como entrada para el controlador. Por lo tanto se recomienda encontrar una forma de conectar el módulo de cámara web de Robotino y el lenguaje de programación Labview de tal forma que le permita trabajar de forma directa con la imagen que emite el robot y a la vez trabajar con el módulo de Vision and motion que presenta Labview.
- Para un adecuado funcionamiento del robot móvil Robotino y que el mismo no presente desconexión, algún tipo de desvío o pérdida de pista durante su trabajo se recomienda que las baterías se encuentren cargadas al 100%.

- Diseñar e implementar un sistema de iluminación propio para Robotino, ya que al estar realizando el recorrido por una pista, muchas de las veces pierde la línea a razón de la sombra que produce el chasis propio del robot.
- Investigar y realizar el control del robot móvil Robotino utilizando Matlab, sería una experiencia nueva llena de aprendizaje ya que la manipulación del robot se vuelve mucho más interesante al realizarla con un software que no sea el propio del robot, agregando retos que se pueden ir superando con la interacción del robot y el entorno.
- Realizar un mantenimiento general al robot móvil Robotino, ya que el mal uso lo ha ido deteriorando, dando como consecuencia que en el transcurso del desarrollo del presente proyecto, la constante falla en las baterías ha dado más de un inconveniente.

BIBLIOGRAFÍA

- Martin del Brío, Bonifacio; Sanz Molina Alfredo. “Redes Neuronales y sistemas borrosos”, Tercera edición, México D.F. - México, Alfaomega Grupo Editor; 2007.

[Del Brío].

- Barrientos, Antonio; Penin, Luis Felipe; Balaguer, Carlos; Aracil Rafael, “Fundamentos de Robótica”, Segunda edición, Madrid-España, Mc Graw-Hill; 2007.

[Barrientos].

- CHAPRA, Steven, Métodos numéricos para Ingenieros, Quinta edición, Editorial McGraw-Hill Interamericana, México D.F.-México, 2007.

[Chapra].

- Ollero Barturone A., “Robótica. Manipuladores y Robots móviles”, Primera edición, Editorial Macombo, Barcelona-España, 2001

[Ollero].

- XiaoQi Chen, Y.Q. Chen and J.G. Chase. Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions. Published by In-Teh; 2009.

[XiaoQi].

- Pajares Martinsanz, Gonzalo; De la Cruz García, Jesús. Visión por computador. Imágenes digitales y aplicaciones. 2^{da} edición. México: Alfaomega Grupo Editor, 2008.

[Pajares].

- Siegwart, Roland; Nourbakhsh, Illah. Introduction to Autonomous Mobile Robots. A Bradford Book. Cambridge, London England. 2004.

[Siegwart].

Páginas Web

- Tutorial de Introducción de Lógica Borrosa, 2008, María de los Ángeles Olmo Castillo, disponible en la web en: <http://www.dma.fi.upm.es/java/fuzzy/tutfuzzy/>, consulta del 2011-10-01.
[Web_01].
- Lógica difusa blog donde se encuentra información importante sobre lógica difusa, disponible en la web en <http://logicadifusatgs.blogspot.com/>, consulta del 2011-10-03.
[Web_02].
- Página web donde encontramos la biografía de Karel Capek, disponible en la web en: <http://www.epdlp.com/escritor.php?id=154>, consulta del 2011-10-06.
[Web_03].
- BBC mundo noticias. Página web donde se publican noticias de la actualidad. Artículo “Nacen las roboto-rachas”, disponible en la web en: <http://news.bbc.co.uk/>, consulta del 2011-10-06.
[Web_04].
- Inventos a lo largo de la historia, disponible en la web en: <http://www.fi.edu/learn/scitech//automaton/automaton.php?cts=instrumentation>, consulta del 2011-10-06.
[Web_05].
- Valiosa información sobre la robótica, resumida en una pagina web. http://cfievalladolid2.net/tecno/cyr_01/robotica/intro, consulta del 2011-10-06.
[Web_06].

- Robótica. Blog donde se encuentra información sobre la clasificación de los robots, disponible en la web en: <http://robotik-jjlg.blogspot.com/2009/03/tipos-de-robots-2>, consulta del 2011-1-06.
[Web_07].
- Blog de gadgets y electrónica de consumo, con una categoría de robótica, disponible en la web en: <http://www.xataka.com/robotica/llega-el-primer-robot-que-aprende-toma-sus-propias-decisiones-y-consulta-sus-dudas-con-otros-robots> , consulta del 2011-10-06.
[Web_08].
- Presentación sobre sistemas de locomoción de robots móviles, disponible en la web en: www.esi2.us.es/~vivas/ayr2iaei/LOC_MOV.pdf, Consulta del 2011-10-06.
[Web_09].
- Epistemowikia. Revista «Hiperenciclopédica» de Divulgación del Saber. Artículo: Locomoción en robots, disponible en la web en: [http://campusvirtual.unex.es/cala/epistemowikia/index.php?title=Locomoci%C3%B3n en robots#Robots con orugas](http://campusvirtual.unex.es/cala/epistemowikia/index.php?title=Locomoci%C3%B3n%20en%20robots#Robots%20con%20orugas), Consulta del 2011-1-06.
[Web_10].
- Lógica Difusa, aplicaciones, disponible en la web en: <http://cial.110mb.com/LADY/Aplicaciones>, consulta del 2011-10-08.
[Web_11].
- Autómatas en la historia, disponible en la web en: http://automata.cps.unizar.es/Historia/Webs/automatas_en_la_historia.htm, consulta del 2011-10-09
[Web_12].

- Blog que trata acerca de la Robótica, la visión artificial y sus avances en la industria y en la medicina, disponible en la web en: <http://laroboticaindustriaymedicina.blogspot.com/>, consulta del 2011-10-09.
[Web_13].
- Web que contiene noticias de actualizadas sobre la tecnología. Artículo referido; Robots bípedos compiten en un maratón de cuatro días en Japón., disponible en la web en: <http://es.kioskea.net/news/>. Consulta del 2011-10-12.
[Web_14].
- Artículo sobre BigDog el robot todo terreno, disponible en la web en: <http://www.xataka.com/robotica/bigdog-el-robot-todoterreno.> consulta del 2011-10-12.
[Web_15].
- Artículo y video sobre robot hexápodo. Blog de avances de robots. <http://blog.makezine.com/archive/2011/08/morphex-conventional-hexapod-2.html>, disponible en la web en: consulta del 2011-10-12.
[Web_16].
- Blog sobre tecnología, <http://www.madboxpc.com/foro/topic/156645-disena-japon-robot-que-levanta-pacientes-de-hasta-80-kgs/>, consulta del 2011-10-12.
[Web_17].
- Página de irobot, disponible en la web en: <http://www.irobot.com.ec/productos.html>, consulta del 2011-10-12.
[Web_18].
- Página de negotiator robot, disponible en la web en: http://www.irobot.com/gi/ground/210_Negotiator, consulta del 2011-10-12.
[Web_19].

- Introducción a los sensores inductivos, disponible en la web en: [http://galia.fc.uaslp.mx/~cantocar/automatas/PRESENTACIONES PLC PDF S/24 SENORES INDUCTIVOS.PDF](http://galia.fc.uaslp.mx/~cantocar/automatas/PRESENTACIONES_PLC_PDF_S/24_SENORES_INDUCTIVOS.PDF), consulta del 2011-11-04.

[Web_20].

- REC Robotics Equipment Corporation GmbH Dr. Christian Verbeek, Foro de Robotino en línea, Alemania, disponible en la web en: <http://forum.openrobotino.org.>, consultas de 2011-11-05 - 2011-11-30.

[Web_21].

Manuales Técnicos

- Manual de Robotino, disponible en la web en: http://www.festo-didactic.com/int-en/services/printed-media/technicaldocumentation/robotino-manual_544305.htm?fbid=aW50LmVuLjU1Ny4xNy4zMj45MDguNjcwOA, consulta del 2011-11-06.

[Manual Robotino].

Tesis

- Gualotuña Villavicencio, Roberto Jairo; Llinin Rea, Edison Orlando. “Implementación de un algoritmo de búsqueda informada en el robot móvil Robotino de Festo para la obtención de la trayectoria más óptima en tiempo real dentro de un entorno controlado”. Quito; Universidad Politécnica Salesiana, Facultad de Ingenierías, Carrera de Sistemas, Mención Robótica e Inteligencia Artificial. 2011.

[Tesis_01].

- BAUTISTA LOZA, Doris Anabelle; TUFÍÑO CARDENAS, Rodrigo Efraín. “Diseño y construcción de un robot cartesiano para la manipulación e identificación de figuras geométricas a través de medios ópticos utilizando redes neuronales”. Quito; Universidad Politécnica Salesiana, Facultad de Ingenierías, Carrera de Sistemas, Mención Robótica e Inteligencia Artificial.

[Tesis_02].

ANEXO 1

Procesamiento digital de imágenes

Procesamiento de imágenes

El Procesamiento de imágenes es un conjunto de operaciones utilizadas para mejorar un set de datos de una imagen con el fin de mejorarlas de alguna forma, para ayudar a su interpretación o para extraer algún tipo de información útil de ella.

Binarización

La binarización de una imagen consiste en tener una imagen en blanco y negro, preservando las propiedades esenciales de la imagen.

Segmentación

Pretende separar los objetos del fondo de la imagen, para luego diferenciarlos unos de otros.

Bordes

Los bordes caracterizan generalmente los límites de los objetos y serán útiles en la segmentación e identificación de objetos en escenas.

Detección de bordes

Una buena detección de bordes es fundamental para la visión artificial orientada al reconocimiento de objetos. Si la detección es pobre o da bordes donde no los hay, el programa puede llegar a reconocer un objeto donde no lo hay.³⁴

Procesamiento digital de imágenes en Labview

Para obtener un procesamiento digital de imágenes en Labview sin el uso de una librería propia para la adquisición de imágenes, es fundamental el uso de registros de desplazamiento (Shift Register), pues mediante estos es posible recorrer una imagen de igual forma se utilizan bucles, sentencias case, manejo de imágenes así como se

³⁴ Detección de bordes, disponible en <http://members.fortunecity.es/davidweb2/visart/bordes.htm>

hace uso de múltiples variables de distintos tipos para ir guardando valores que se necesitan durante el programa.

Al no tener una conexión directa entre el módulo de cámara web de Robotino y Labview, lo que se hace para procesar las imágenes es capturar la imagen y guardarla a la misma en disco y de esta forma procesarla.

Para procesar las imágenes se hace uso de los siguientes bloques:

<p style="text-align: center;">Registros de desplazamiento</p> 	<p>Los registros de desplazamiento guardan la información que queremos de una iteración anterior para que podamos usarla cuando queramos, para utilizar un <i>Shift Register</i> se debe de crear un bucle, una vez que se tiene el bucle creado se da clic derecho y se selecciona <i>Add Shift Register</i>:</p> <p>Se inicializa con la conexión exterior del lado izquierdo es decir lo que va a valer en la primera iteración, la interior de este mismo lado es el valor de la variable que en la iteración anterior hayamos conectado a la conexión interior del derecho.</p> <p>Finalmente la conexión exterior del lado derecho es el resultado de la última iteración</p>
<p style="text-align: center;">Leer archivo .JPEG</p> <p style="text-align: center;">Read JPEG File.vi</p>  <p>Reads a JPEG file and creates the data necessary to display the file in a picture control.</p>	<p>Las funciones que se necesitan para trabajar con imágenes dentro de Labview se encuentran en <i>Programming</i> → <i>Graphics & Sound</i> → <i>Graphics Formats</i>:</p> <p>Para leer imágenes se recibe como parámetro de entrada la dirección de la imagen, y se devuelve un conjunto</p>

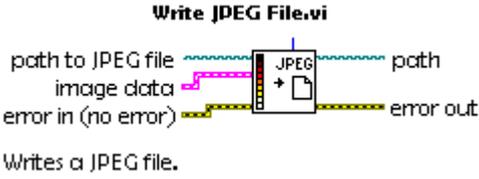
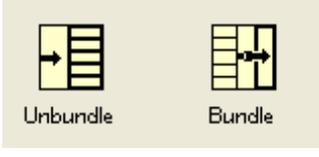
	<p>(cluster) con los datos de la imagen en caso de que la lectura haya resultado exitosa, o un error en caso de que algún error a la hora de abrir la imagen.</p>
<p style="text-align: center;">Escribir archive .JPEG</p> <p style="text-align: center;">Write JPEG File.vi</p> 	<p>Para la escritura de una imagen se recibe como parámetro de entrada los datos de la imagen y la dirección donde queremos guardar la nueva imagen, y devuelve la existencia de algún error o no a la hora de escribir.</p>
<p style="text-align: center;">Cluster</p> 	<p>Se utiliza la función unbundle de clusters para poder acceder al array donde se encuentran los datos de la imagen, y la función bundle para crear un arreglo con los datos de la imagen una vez procesada</p>

Tabla A-1 Descripción de procesamiento de imágenes

Fuente: [Los autores].

Se recorre las líneas de la imagen esto a través de los registros de desplazamiento y de lazos for: un para recorrer el ancho de la imagen así como uno para recorrer el alto de la imagen, se escoge un valor RGB umbral para decidir que es negro (línea) y que es blanco mediante las pruebas con distintos valores con un valor de 60 de umbral es suficiente además no hay que iluminar la línea la que esto crea un reflejo y se procede a buscar píxeles negros que estén por debajo del umbral.

Una vez que se encuentre dentro de línea se continua buscando píxeles hasta que se detecte el otro borde, se tiene que guardar tanto el número del primer píxel así como el del último píxel de la línea para calcular el punto medio. Una vez obtenido el punto medio se realiza un cálculo para ver la distancia del punto medio con la del

centro de la imagen, la diferencia entre el punto medio y el centro de la imagen indica que tan desviado se encuentra el robot de la línea negra y según el valor se conoce si la desviación es hacia la derecha o a la izquierda.

A continuación se muestra el orden de los bloques a seguir para obtener el valor de la coordenada en donde se encuentra en centro de la línea con respecto al tamaño de la imagen

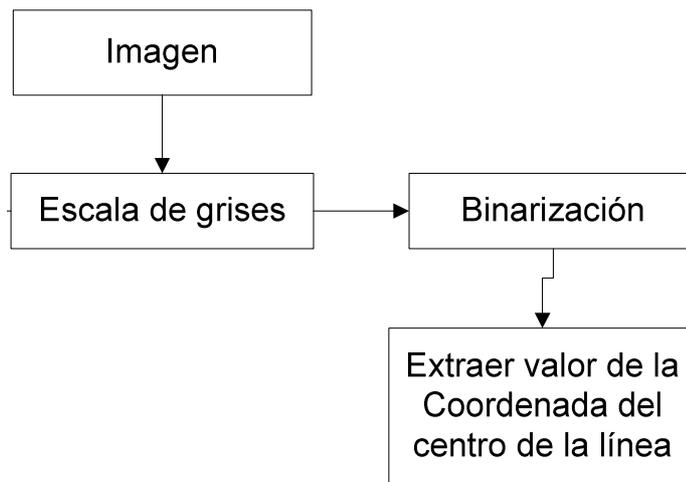


Figura A1-1 Diagrama de bloque para procesamiento de la imagen.

Fuente: [Los autores].

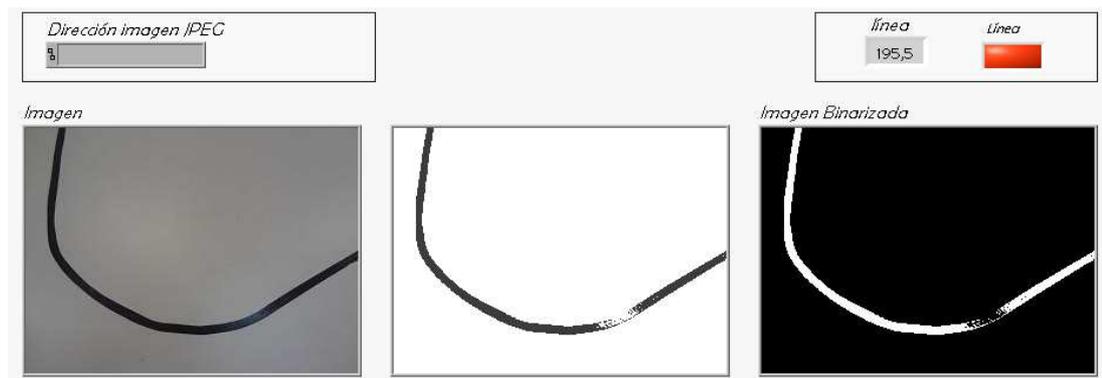


Figura A1-2 Imagen binarizada con umbral de 60.

Fuente: [Labview 2010].

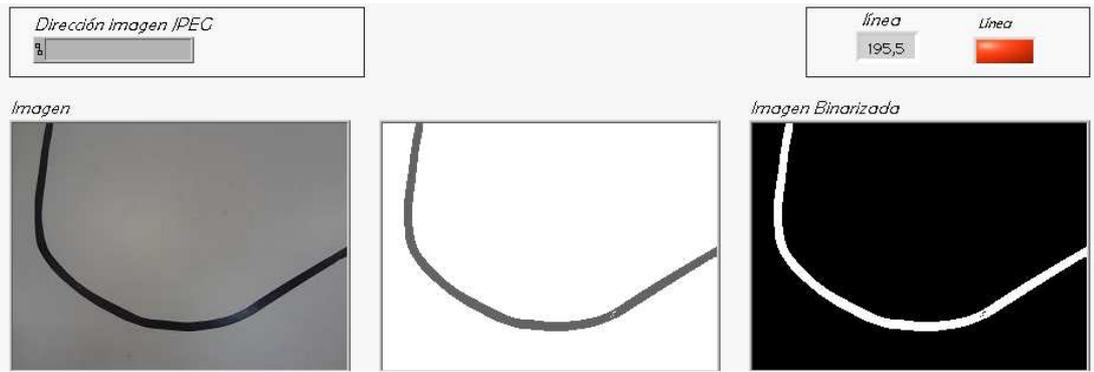


Figura A1-3 Imagen binarizada con umbral de 100.

Fuente: [Labview 2010].

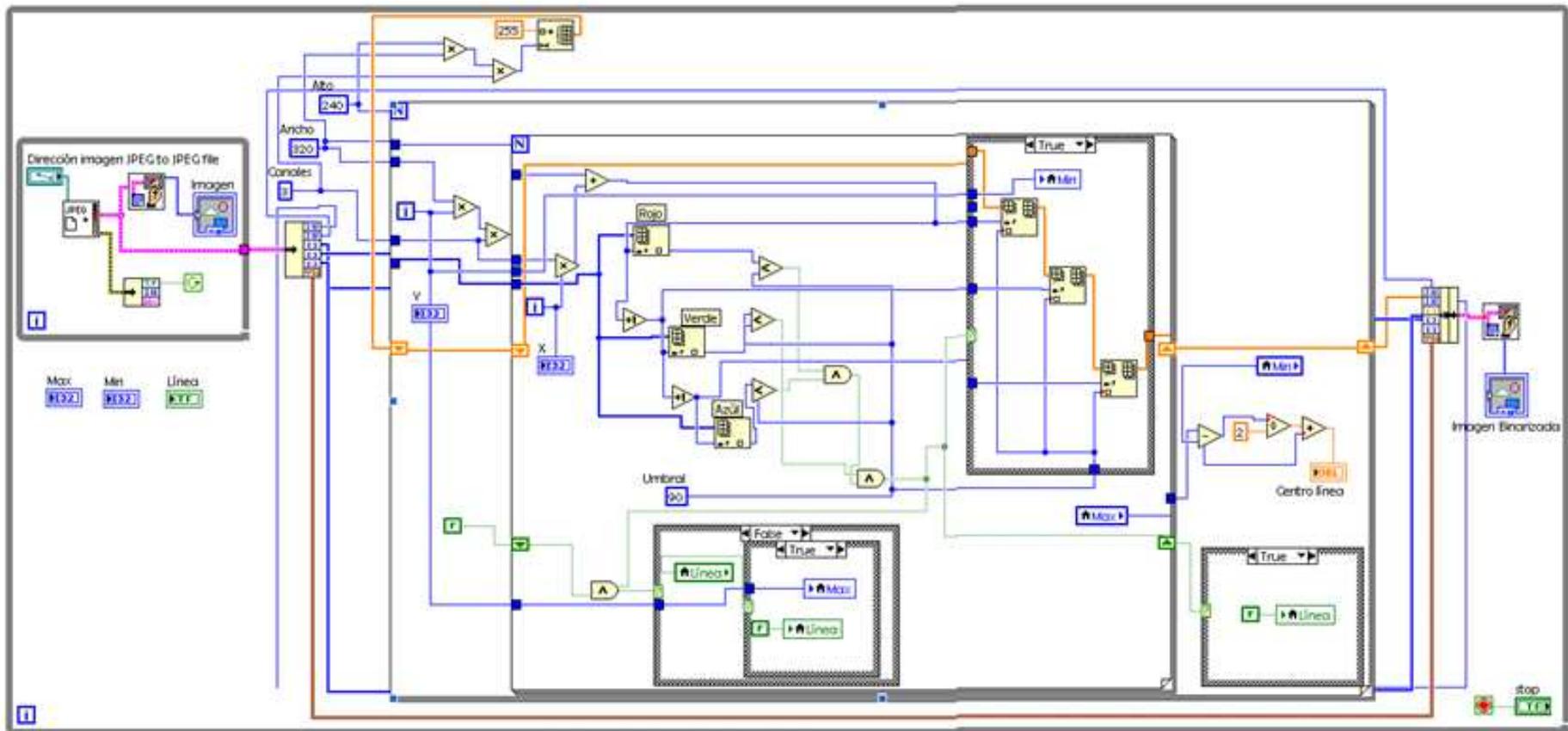


Figura A1-4 Diagrama de bloques de para el procesamiento de la imagen.

Fuente: [Labview 2010].

Operaciones morfológicas

La morfología matemática es una herramienta muy utilizada en el procesamiento de imágenes. Las operaciones morfológicas pueden simplificar los datos de una imagen, preservar las características esenciales y eliminar aspectos irrelevantes. Teniendo en cuenta que la identificación y descomposición de objetos, la extracción de rasgos, la localización de defectos e incluso los defectos en líneas de ensamblaje están sumamente relacionados con las formas.

La morfología matemática se puede usar, entre otros, con los siguientes objetivos:

- Pre procesamiento de imágenes (supresión de ruido, simplificación de formas).
- Destacar la estructura de objetos (extraer el esqueleto, marcado de objetos, envolvente convexa, ampliación, reducción).
- Descripción cualitativa de objetos (área, perímetro, diámetro, etc).

Dilatación y Erosión

Estas operaciones son fundamentales en el procesamiento morfológico. De hecho, la mayoría de los algoritmos morfológicos están basados en estas dos operaciones aunque también se utilizan las siguientes:

- Apertura y Clausura (Opening and Closing)
- Transformación Hit-or-Miss (o Ganancia - Pérdida)
- Extracción de Frontera (Boundary Extraction)
- Afinado o Adelgazamiento (Thinning)
- Engrosamiento (Thickening)
- Relleno de Región (Region Filling)
- Esqueleto
- Poda (Pruning)
- Traslación
- Reflexión
- Resta de imágenes

Las operaciones que serán efectuadas en este trabajo son:

Dilatación

La dilatación de imágenes es un operador conocido como 'relleno', 'expansión' o 'crecimiento'. Puede ser usado para rellenar 'huecos' de tamaño igual o menor que el elemento estructurante con el que se opera la dilatación.

Usado con imágenes binarias, donde cada píxel es 1 ó 0, la dilatación es similar a la convolución. Sobre cada píxel de la imagen se superpone el origen del elemento estructurante. Si el píxel de la imagen no es cero, cada píxel que cae en la estructura es añadido al resultado aplicando el operador 'or'.

Con la notación:

$$A \oplus B$$

Representando la dilatación de una imagen A por un elemento estructurante B, se puede escribir:

$$C = A \oplus B = \cup_B (A)_b$$

Donde $(A)_b$ representa la traslación de A por b. Intuitivamente, para cada elemento no cero b_{ij} de B, A es trasladado i, j y sumado a C usando el operador 'or'.

Ejemplo:

$$\begin{pmatrix} 0100 \\ 0100 \\ 0110 \\ 1000 \\ 0000 \end{pmatrix} \oplus [\cdot 11] = \begin{pmatrix} 0110 \\ 0110 \\ 0111 \\ 1100 \\ 0000 \end{pmatrix}$$

Figura A1-5 Origen del elemento estructurante está en (0,0).

Fuente: [Los autores].

Usado con imágenes en escala de grises, la dilatación se efectúa tomando el máximo de una serie de sumas. Puede ser usado para implementar el operador de 'máxima vecindad' con la forma de la vecindad dada en el elemento estructurante.

Erosión

La erosión es lo opuesto a la dilatación; realiza con el fondo lo que la dilatación realiza al objeto de la imagen. También en este caso, existe un elemento estructurante que se utiliza para operar con la imagen. Los efectos son de 'encogimiento', 'contracción', o 'reducción'. Puede ser utilizado para eliminar islas menores en tamaño que el elemento estructurante.

Sobre cada píxel de la imagen se superpone el origen del elemento estructurante. Si cada elemento no cero de dicho elemento está contenido en la imagen, entonces el píxel de salida es puesto a 1. Haciendo $A \otimes B$ como representación de la erosión de una imagen A por el elemento estructurante B, se puede definir:

$$C = A \otimes B = \cap_B (A)_{-b}$$

Donde $(A)_{-b}$ representa la traslación de A por b . B puede ser visto como una 'sonda' que se desliza a lo largo de toda la imagen A, testando la naturaleza espacial de A en cada punto. Si B trasladado i, j puede ser contenido en A (poniendo el origen de B en i, j), entonces i, j pertenece a la erosión de A por B.

Ejemplo:

$$\begin{pmatrix} 0100 \\ 0100 \\ 1110 \\ 1000 \\ 0000 \end{pmatrix} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{pmatrix} 0000 \\ 0000 \\ 1100 \\ 0000 \\ 0000 \end{pmatrix}$$

Figura A1-6 Origen del elemento estructurante está en (0,0).

Fuente: [Los autores].

Usado en imágenes en escala de grises, la erosión se efectúa tomando el mínimo de una serie de diferencias. Puede ser usado para implementar el operador de 'mínima vecindad' con la forma de la 'vecindad' dada por el elemento estructurante.

Detección de bordes

La detección de bordes es una parte fundamental de la mayoría de sistemas de visión puesto que el éxito de los niveles siguientes de procesamiento depende fuertemente de la fidelidad de las características, en este caso los bordes.

La mayoría de las técnicas para detectar bordes emplean operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen.

Operadores basados en la primera derivada (gradiente)

El gradiente de una imagen $f(x, y)$ en un punto (x, y) , se define como un vector bidimensional dado por:

$$\mathbf{G}[f(x, y)] = \begin{bmatrix} \mathbf{G}_x \\ \mathbf{G}_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix}$$

Dónde el vector \mathbf{G} apunta en la dirección de variación máxima de f en el punto (x, y) por unidad de distancia, con la magnitud y dirección dadas por:

$$|\mathbf{G}| = \sqrt{G_x^2 + G_y^2}; \quad \theta(x, y) = \tan^{-1} \frac{G_x}{G_y}$$

Para la práctica es común aproximar la magnitud del gradiente con valores absolutos, porque el valor de magnitud del gradiente no es tan importante como la relación entre diferentes valores, es decir para decidir si un punto es de borde según que la magnitud del gradiente supere o no un determinado umbral, solo es cuestión de ajustar dicho umbral para que el resultado de extracción de bordes sea el mismo tanto si se calcula la magnitud del gradiente.

$$|\mathbf{G}| \approx |G_x| + |G_y|$$

Operadores de Sobel

El operador de Sobel tiene la propiedad de suavizar la imagen, eliminando parte del ruido y por lo tanto, minimiza la aparición de falsos bordes debido al efecto de magnificación del ruido por parte de los operadores de primera derivada.

Los valores de G_x y G_y pueden implementarse por convolución de la imagen con las máscaras mostradas a continuación conocidas como operadores de Sobel.

$$\begin{matrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -2 & -1 \\ -0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \\ (a) & (b) \end{matrix}$$

Figura A1-7 (a) Máscara usada para obtener G_x en el punto central de la región 3×3 , (b) Máscara usada para obtener G_y en el mismo punto.

Fuente: [Pajares].

Aplicación de operaciones morfológicas en Labview

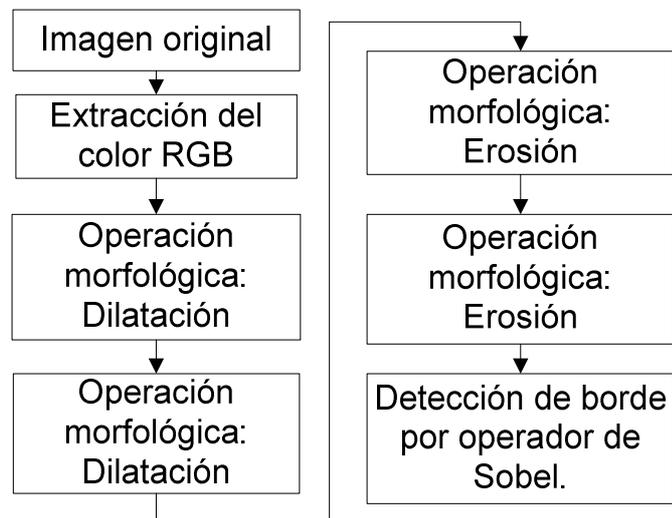


Figura A1-8 Diagrama para la aplicación de operaciones morfológicas sobre las imágenes.

Fuente: [Los autores].

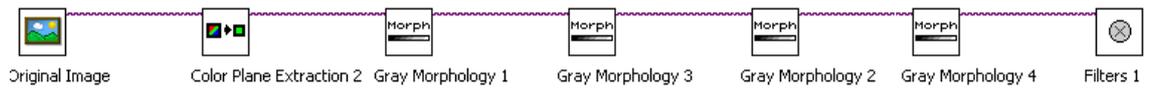
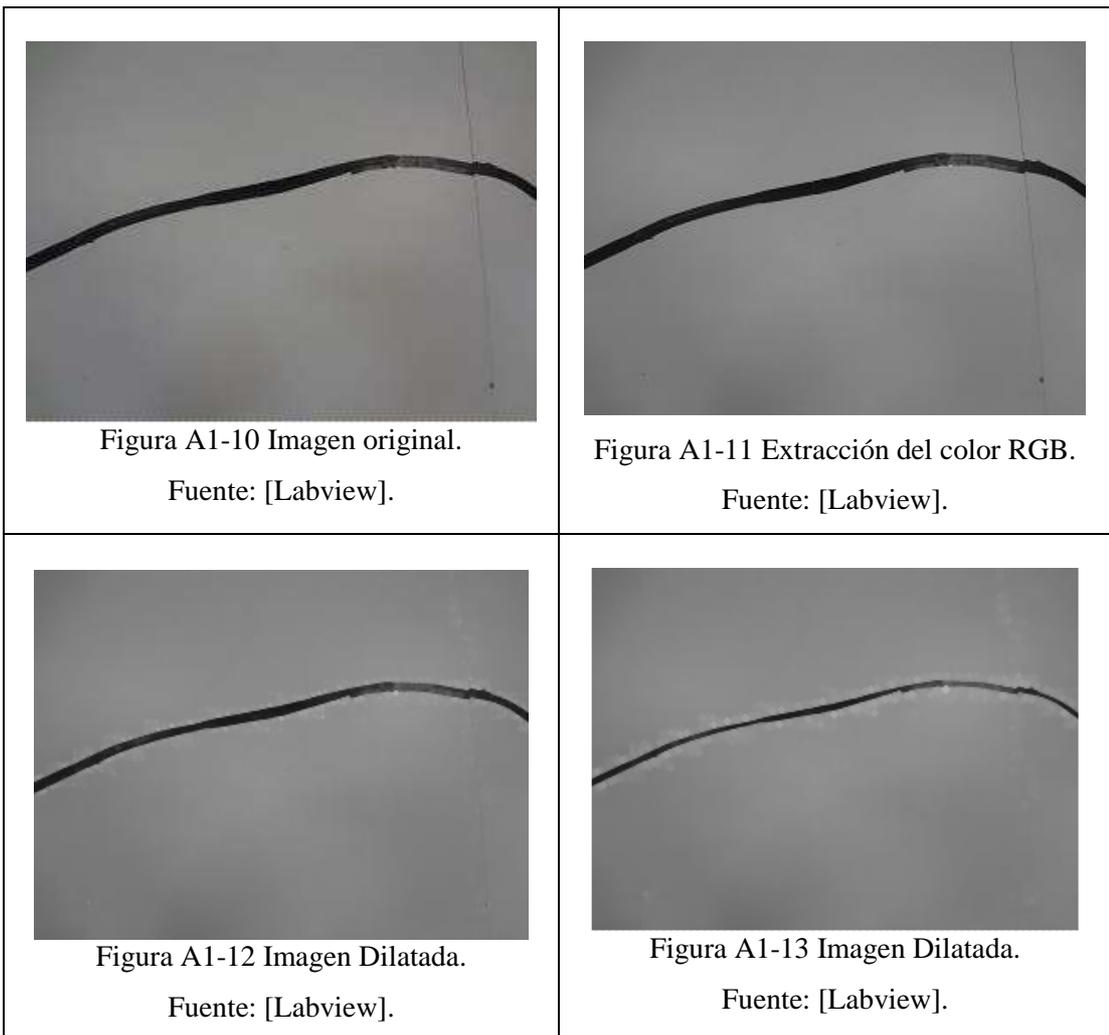


Figura A1-9 Secuencia para la aplicación de operaciones morfológicas sobre las imágenes en Labview.

Fuente: [Labview].

Tal como se muestra en los diagramas anteriores a la imagen original se le aplico una extracción del color RGB, para tener una imagen binarizada, luego se le aplica la operación morfológica dilatación con el fin de que cualquier línea o punto sobre el fondo blanco que no corresponda a la pista sea imperceptible, una vez que la imagen solo muestra la pista se le aplica la operación morfológica de erosión para que la imagen muestre la pista sin ningún tipo de basura con el tamaño original y por último aplicar el filtro de Sobel para detectar el borde de la pista.



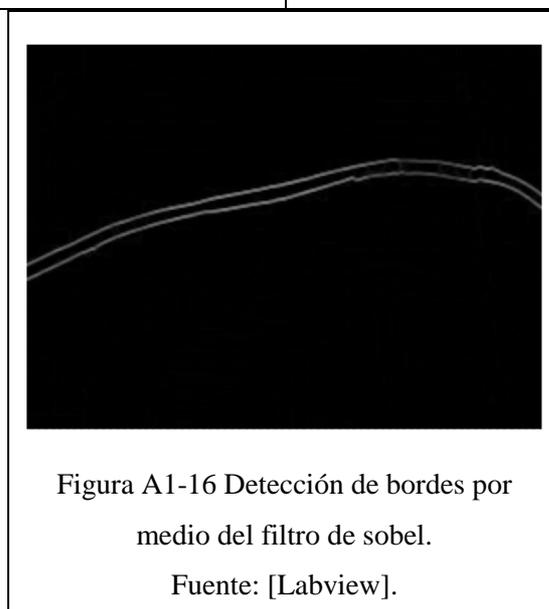
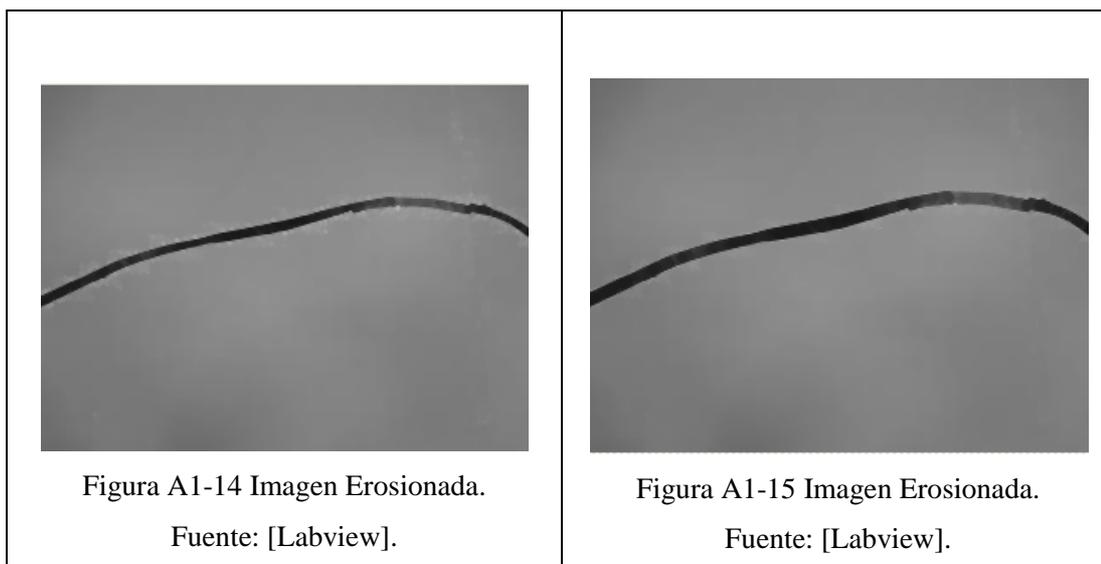


Tabla A-2 Operaciones morfológicas realizadas sobre una imagen.

Fuente: [Los autores].

En la siguiente tabla se muestra cada una de las imágenes correspondientes a la pista luego de ser aplicadas las operaciones morfológicas de dilatación y erosión, así como el filtro de Sobel para la detección de bordes.

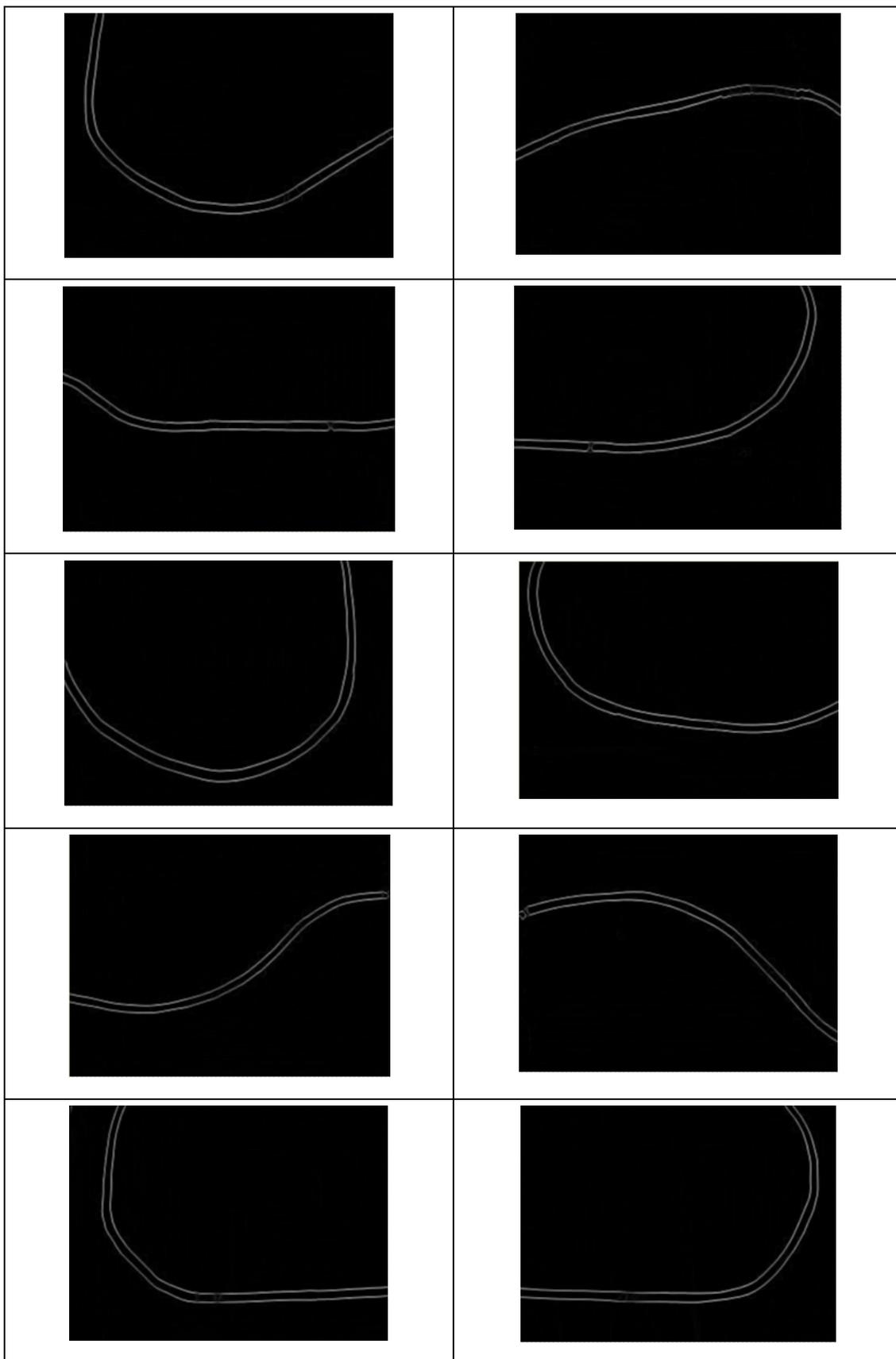


Tabla A-3 Imágenes de la pista aplicadas operaciones morfológicas.

Fuente: [Los autores].

ANEXO 2

Diseño en Fuzzy System Designer de Labview

Esta librería cuenta con un asistente para poder crear el diseño del controlador difuso dependiendo de las necesidades del programador. Para hacer uso de dicho asistente de control difuso en Labview 2010, se deben seguir los siguientes pasos:

En la barra de herramientas seleccionar Tools.



Figura A2-1 Barra de herramientas

Fuente: [Labview 2010].

A continuación emerge un menú en el cual se debe seleccionar Control Design and Simulation.

Luego de seleccionar dicho ítem, emergerá otro menú denominado Fuzzy System Designer, el cual es el asistente de control difuso.

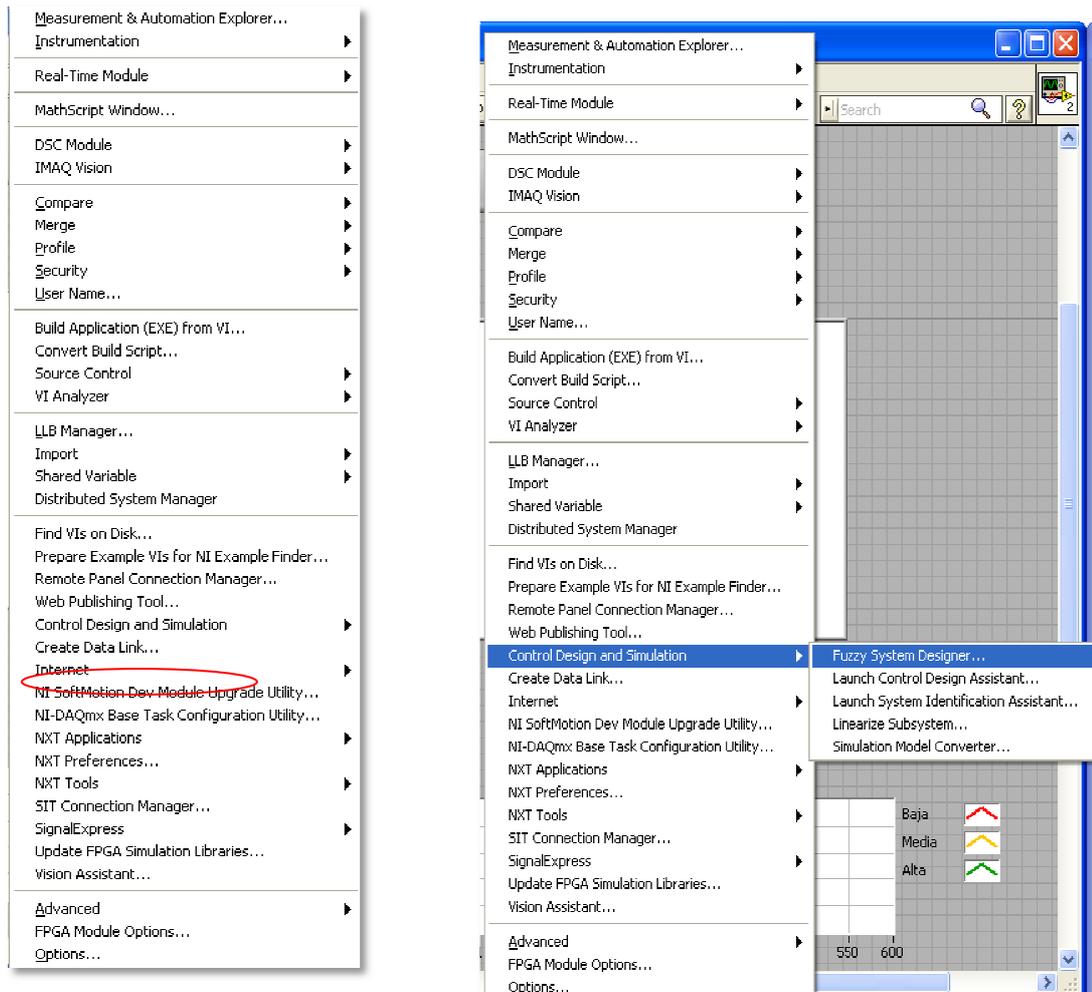


Figura A2-2 FuzzySystemDesigner

Fuente: [Labview 2010].

Inmediatamente después de dar clic en Fuzzy System Designer, aparecerá la siguiente ventana, en la cual se empezará a colocar las entradas y salidas del sistema, para esto se da clic en el icono (+).

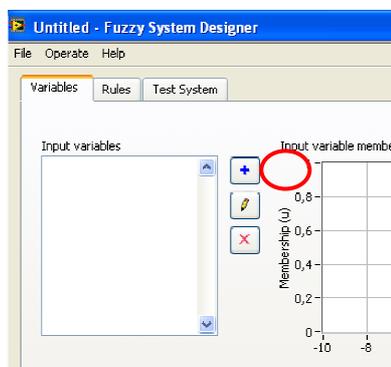


Figura A2-3 Asignar entradas y salidas

Fuente: [Labview 2010].

Aparecerá el editor de variables.

Ahí se procede a colocar las diferentes variables con el rango correspondiente de datos que ocupe.

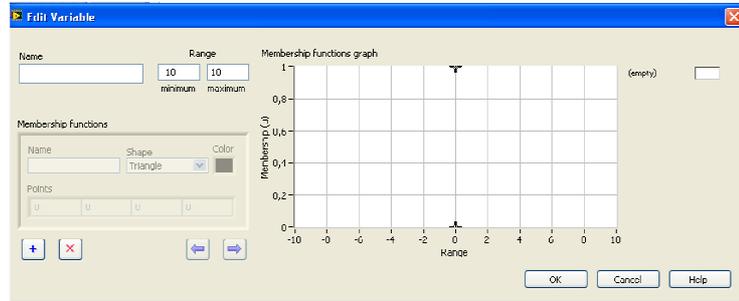


Figura A2-4 Editor de variable

Fuente: [Labview 2010].

Como ejemplo se ha creado la variable “velocidad”, por defecto la grafica de la función es triangular, pero se pueden cambiar sus propiedades, es decir su forma y color.

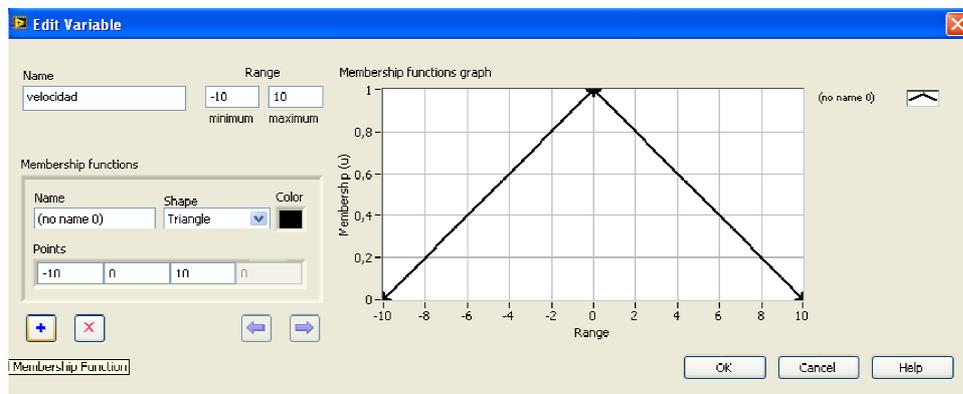


Figura A2-5 Función

Fuente: [Labview 2010].

Cambiando la forma de la grafica de la función.

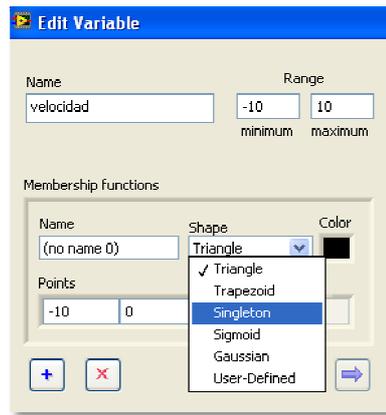


Figura A2-6 Forma de Función

Fuente: [Labview 2010].

Cambiando el color de la grafica de la función.

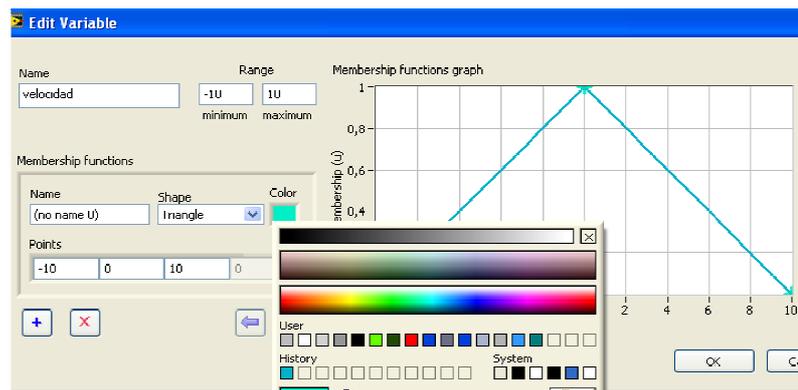


Figura A2-7 Color de Gráfica

Fuente: [Labview 2010].

Para eliminar una variable solamente se dará clic en el símbolo (X)

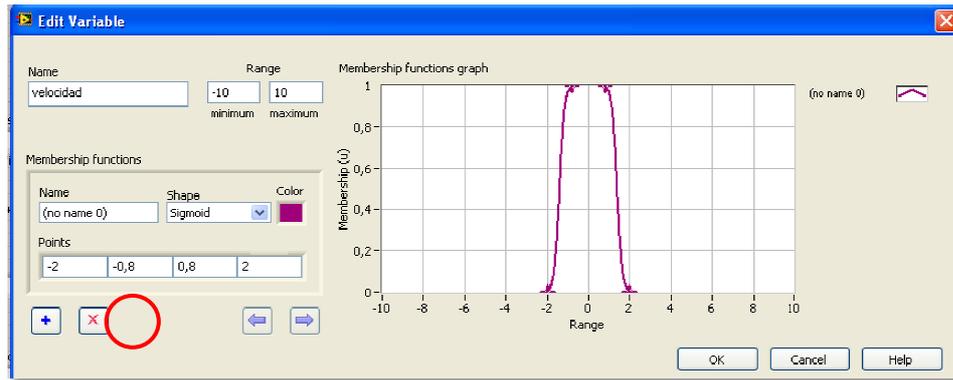


Figura A2-8 Eliminar variable

Fuente: [Labview 2010].

Así se eliminará la variable que no se desee, dando lugar a un campo vacío para poder seguir asignando las variables requeridas.

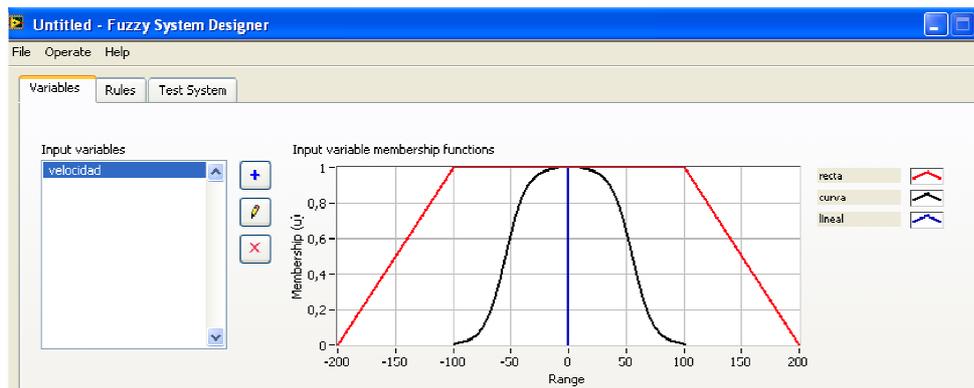


Figura A2-9 Variables creadas

Fuente: [Labview 2010].

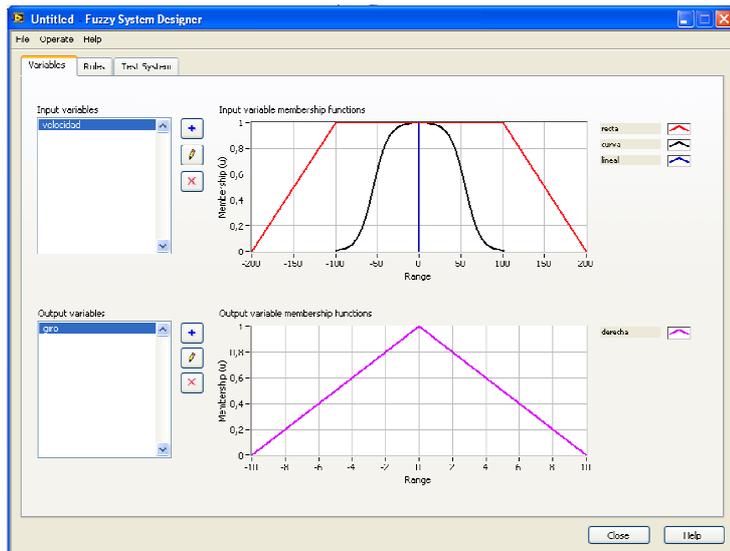


Figura A2-10 Entradas y Salidas

Fuente: [Labview 2010].

Una vez que tengamos las variables creadas, tanto de entrada como de salida, se procede a crear las reglas.

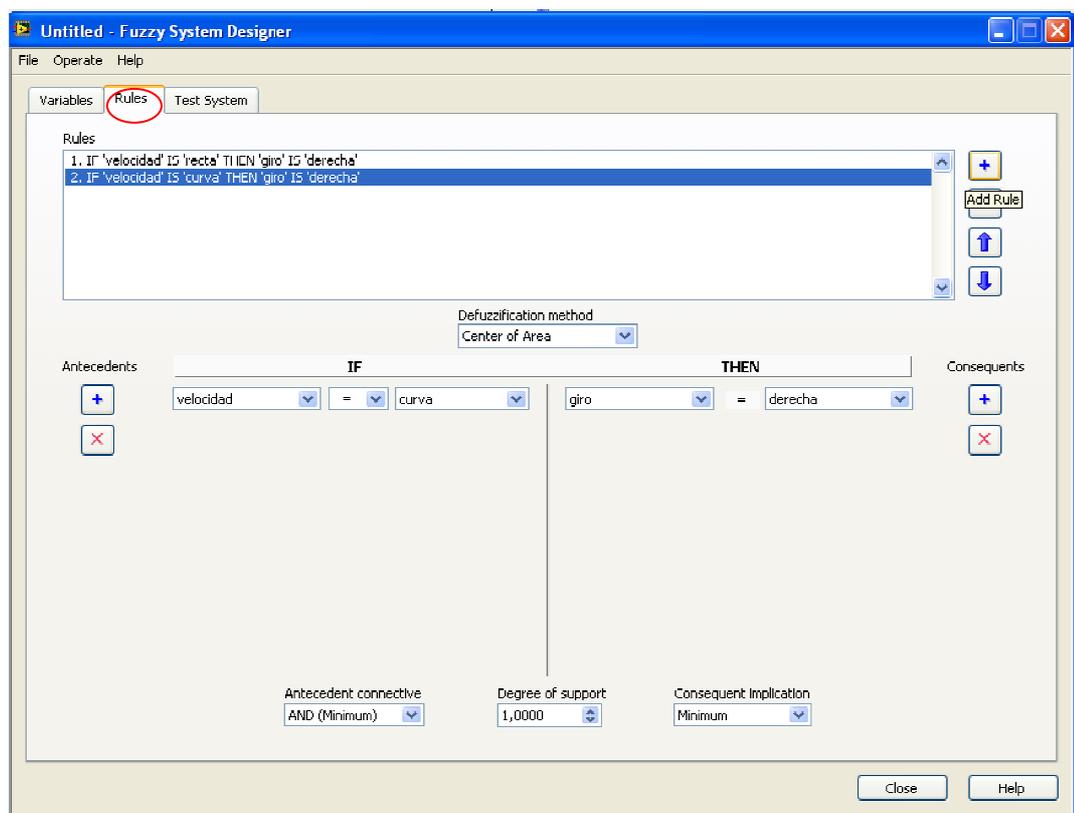


Figura A2-11 Reglas

Fuente: [Labview 2010].

ANEXO 3

Conexión Robotino con Matlab

Para establecer la conexión entre Matlab y Robotino es necesario descargar un Toolbox³⁵, mediante la instalación de este se puede conectar al robot móvil vía wireless con Matlab y a su vez con Simulink.

Al descargar este Toolbox se tiene el siguiente archivo ejecutable.

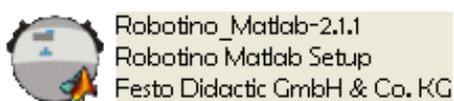


Figura A3-1 Icono instalador Toolbox Robotino-Matlab V2.1.1.

Fuente: [Los autores].

Una vez iniciada la instalación se procede a seleccionar el lenguaje para el programa.



Figura A3-2 Seleccionar lenguaje.

Fuente: [Los autores].

Una vez seleccionado el lenguaje aparece una ventana que da la bienvenida al asistente de instalación, clic en continuar.

³⁵ Página para descargar Toolbox que permite conectar Robotino con Matlab.

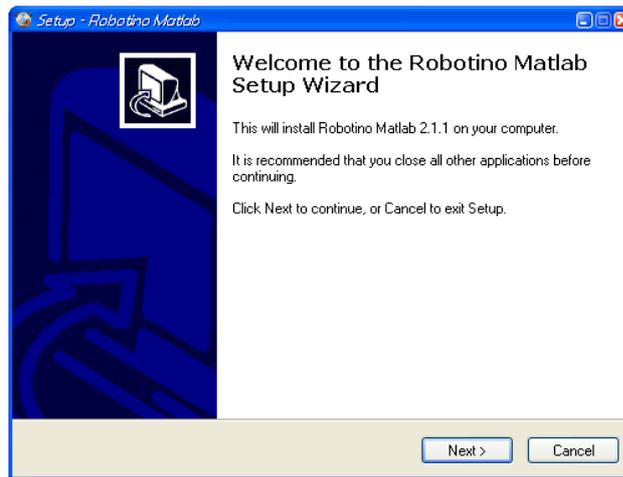


Figura A3-3 Asistente para la instalación.

Fuente: [Los autores].

Luego se muestra una ventana con los términos de la licencia del Toolbox a aceptar.

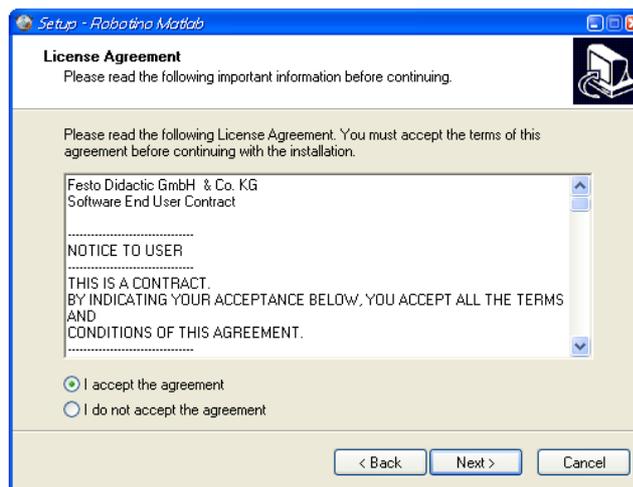


Figura A3-4 Términos de la licencia.

Fuente: [Los autores].

Una vez aceptados los términos de la licencia, se muestra una ventana en la cual se selecciona la carpeta donde se desea que se instale el Toolbox.

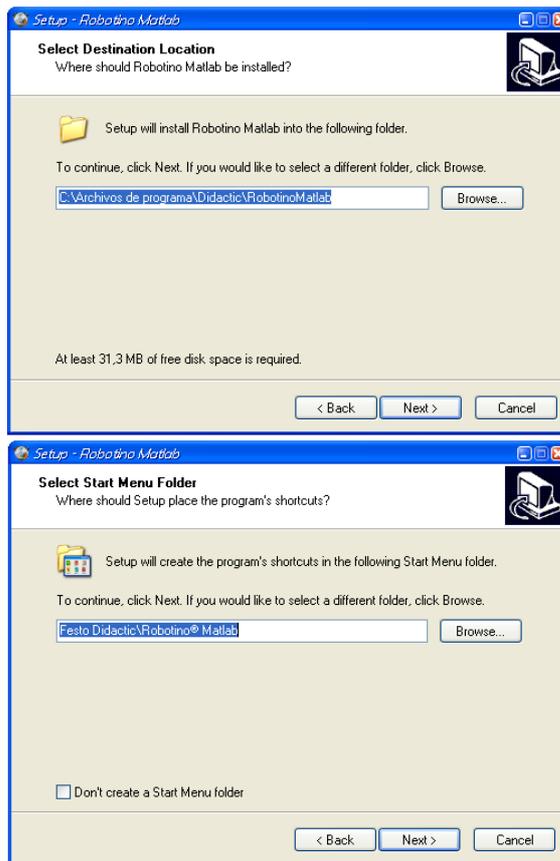


Figura A3-5 Seleccionar carpeta para la instalación.

Fuente: [Los autores].

Finalmente el asistente de instalación muestra la información seleccionada previamente e indica que se encuentra listo para iniciar la instalación.

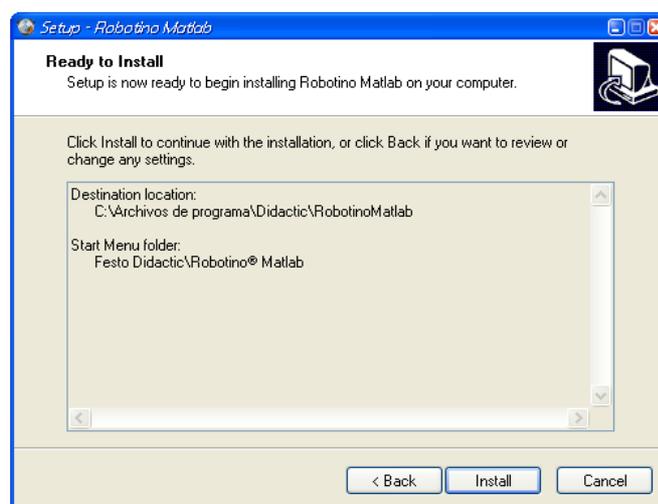


Figura A3-6 Asistente listo para la instalación.

Fuente: [Los autores].

Una vez finalizada la instalación del Toolbox Robotino-Matlab en la carpeta seleccionada se guarda toda la información y los bloques a utilizarse tanto en Matlab como en Simulink.

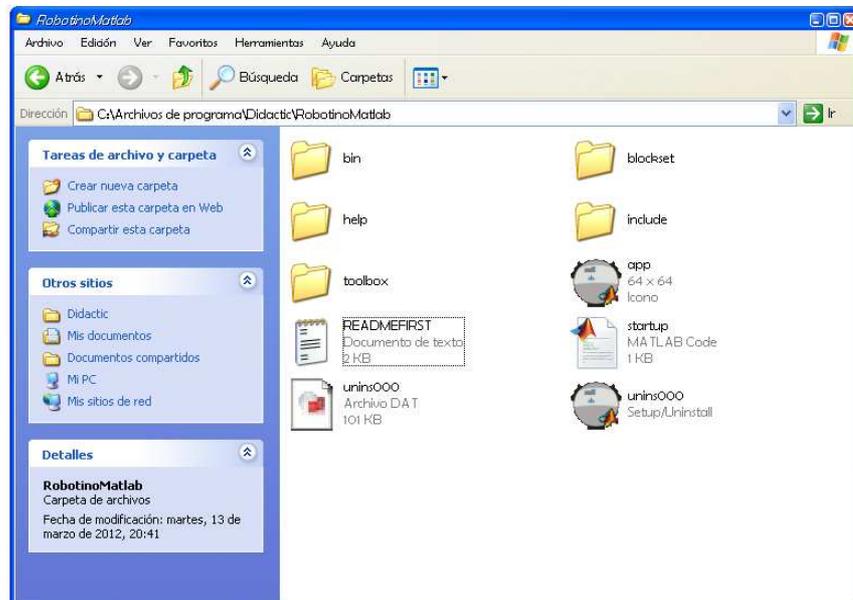


Figura A3-7 Carpeta dónde se encuentra instalado el Toolbox.

Fuente: [Los autores].

Para utilizar el Toolbox, se inicia Matlab y se debe cambiar el directorio de trabajo al de Robotino, lo cual se logra digitando el siguiente texto en el Command window de Matlab.

```
>>cd (getenv ('ROBOTINOMATLAB_DIR'))
```

Además es necesario añadir los bloques y Toolbox para conectar a Robotino y Matlab esto se realiza digitando el siguiente texto:

```
>>addpath (strcat (getenv ('ROBOTINOMATLAB_DIR'),  
'/blockset'));
```

```
>>addpath (strcat (getenv ('ROBOTINOMATLAB_DIR'),  
'/toolbox'));
```

Luego de ingresado este texto, en Matlab se cargan todos los bloques y funciones que permiten conectar a Matlab con Robotino.

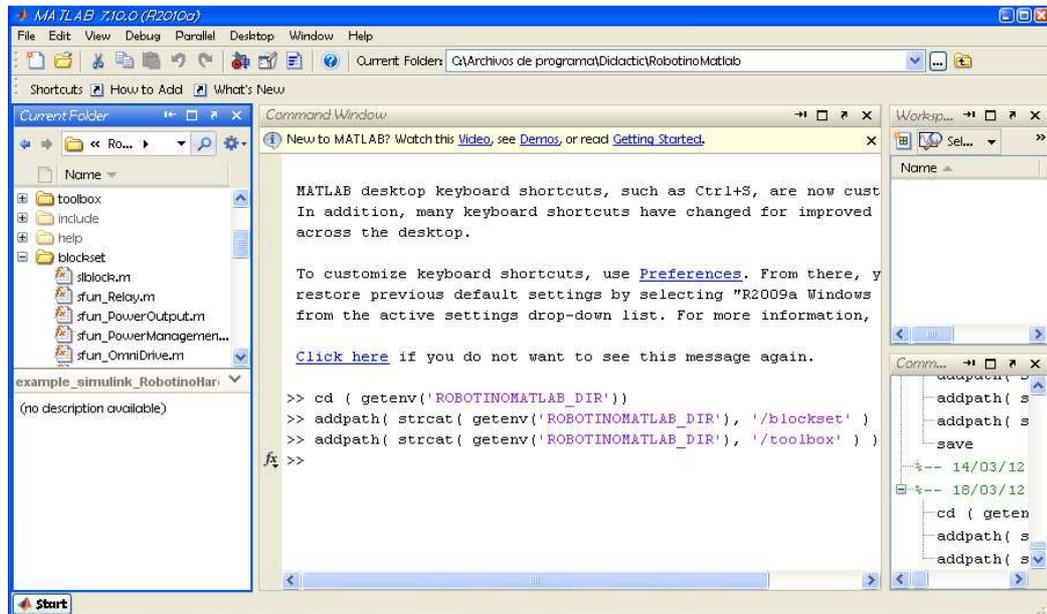


Figura A3-8 Toolbox disponible en Matlab.

Fuente: [Matlab 2010].