

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE NAVEGACIÓN  
AUTONOMA BASADO EN LÓGICA DIFUSA PARA UN ROBOT MÓVIL.**

**CARLOS HERNAN CAICEDO SAAVEDRA  
NELSON OSWALDO CAICEDO SAAVEDRA**

**UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
SAN JUAN DE PASTO  
2014**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE NAVEGACIÓN  
AUTONOMA BASADO EN LÓGICA DIFUSA PARA UN ROBOT MÓVIL.**

**CARLOS HERNAN CAICEDO SAAVEDRA  
NELSON OSWALDO CAICEDO SAAVEDRA**

**Trabajo de grado presentado como requisito parcial para optar al título de  
Ingeniero Electrónico.**

**DIRECTOR  
M.Sc. CAMILO ARTURO LAGOS MORA**

**UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
SAN JUAN DE PASTO  
2014**

## **NOTA DE RESPONSABILIDAD**

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Acuerdo 1. Artículo 324. Octubre 11 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

**Nota de aceptación**

---

---

---

---

---

---

---

---

**Presidente de Tesis**

---

**Jurado 1**

---

**Jurado 2**

**San Juan de Pasto, junio de 2014.**

## **AGRADECIMIENTOS**

Muchas han sido las personas que han contribuido para lograr el desarrollo de este proyecto, participando de una u otra forma en cada una de nuestras etapas formativas en la universidad. En primer lugar queremos agradecer a Dios, por darnos la sabiduría, la fuerza y la salud para trabajar con esmero y dedicación. A todo el personal docente de la Universidad de Nariño, por su entrega a la enseñanza y la investigación, ellos con sus conocimientos brindaron pautas fundamentales para el desarrollo de este trabajo. A las personas que laboran en el departamento de electrónica, por preocuparse porque todo marche en orden y hacer más amenos nuestros días en las aulas de clase. Queremos brindar también un agradecimiento especial al ingeniero Camilo Arturo Lagos M, docente de robótica de la facultad de ingeniería, él con su amplia trayectoria y compromiso orientó nuestro trabajo, resolvió nuestras dudas y colaboró para que los objetivos planteados se logaran con éxito. Gracias a nuestros familiares por apoyarnos en este sueño, ustedes han sido y seguirán siendo nuestra inspiración y por quienes luchamos día a día, gracias por confiar en nosotros, brindarnos su amor y su cariño, gracias por sus consejos y por permitirnos formar parte de la Universidad de Nariño.

## DEDICATORIA

Grandes han sido los esfuerzos realizados en la búsqueda de todos los objetivos propuestos a lo largo de estos años, muchos amigos que han pasado por mi vida saben lo mucho que he soñado con este momento, ellos con su alegría han hecho que me sienta feliz de haber elegido este camino y conocerlos, mi familia con su amor y sus consejos, me han dado la fuerza para afrontar los retos que la vida me impone, ellos han fortalecido mi espíritu cuando se torna difícil el camino, para todos ustedes quiero dedicar este logro más en mi caminar por este mundo; para ustedes Papá y Mamá, quienes sembraron en mí ese deseo de superación, responsabilidad y respeto. Seguiré buscando triunfos, compartiré conocimientos y tratare que todo lo aprendido no se quede allí, demostrando que la educación es lo que enriquece al ser humano, lo llena de valores y le permite luchar por una sociedad con equidad y justicia para todos...

Carlos Hernán Caicedo.

Para mi padre y madre, que con su ejemplo, esfuerzo y sacrificio, siempre buscan y desean el bienestar de todos sus hijos, para toda mi familia que durante estos años hemos luchado juntos, para ellos con cariño y amor.

Nelson Oswaldo Caicedo.

## TABLA DE CONTENIDO

	<b>Pág.</b>
INTRODUCCIÓN.....	20
OBJETIVOS.....	21
1 FUNDAMENTOS TEÓRICOS.....	22
1.1 ANTECEDENTES.....	22
1.2 ROBÓTICA.....	24
1.2.1 Breve reseña histórica.....	23
1.2.2 Clasificación de robots.....	24
1.3 ROBÓTICA MÓVIL.....	24
1.3.1 Grado de libertad (GDL).....	25
1.3.2 Centro instantáneo de rotación (ICR).....	27
1.3.3 Configuración de robots móviles con ruedas.....	27
1.4 SISTEMA DE LOCALIZACIÓN.....	31
1.4.1 Estimación de la posición en un robot móvil.....	31
1.4.2 Estimación de la posición mediante odometría.....	32
1.4.3 Estimación basada en estaciones de transmisión.....	35
1.4.4 Algoritmos de localización utilizando radiofrecuencia.....	36
1.5 SENSORES UTILIZADOS EN ROBÓTICA MÓVIL.....	39
1.5.1 Sensores para medir desplazamiento y posición.....	39
1.6 ACTUADORES.....	41
1.6.1 Motores de corriente continua (DC).....	41
1.6.2 El driver.....	42

1.6.3	Modulación por ancho de pulso PWM.....	43
1.7	MICROCONTROLADORES.....	43
1.7.1	Microcontrolador PIC 18F4550.....	43
1.8	SISTEMA DE CONTROL.....	44
1.8.1	Control PID (Proporcional, Integral y Derivativo).....	44
1.9	SISTEMAS DIFUSOS.....	46
1.9.1	Inteligencia Artificial (IA). ....	46
1.9.2	Lógica difusa.....	46
1.9.3	Conjuntos difusos.....	47
1.9.4	Operaciones entre conjuntos difusos.....	49
1.9.5	Controlador lógico difuso.....	51
1.9.6	Estructura de un FLC.....	51
1.10	ESTUDIO DE LA TECNOLOGÍA ZIGBEE.....	56
1.10.1	Estándar IEEE 802.15.4.....	56
1.10.2	Zigbee.....	56
1.10.3	Dispositivos presentes en una red Zigbee.....	56
1.10.4	Topologías de red.....	57
2	METODOLOGÍA.....	59
2.1	FASE I: REVISIÓN BIBLIOGRÁFICA.....	59
2.2	FASE II: DISEÑO Y CONSTRUCCIÓN DE LA PLATAFORMA ROBÓTICA..	59
2.2.1	Diseño de la plataforma del robot.....	59
2.2.2	Elección de los motores.....	60

2.2.3	Diseño de la plataforma en AUTOCAD.....	60
2.2.4	Construcción de la plataforma.....	61
2.2.5	Construcción del encoder incremental.....	62
2.3	FASE III: DESARROLLO DE LAS SOLUCIONES PLANTEADAS.....	64
2.3.1	Caracterización de los motores.....	64
2.3.2	Medición de la velocidad de los motores.....	64
2.3.3	Control PID en microcontrolador.....	65
2.3.4	Sintonización PID mediante el método de Ziegler – Nichols.....	67
2.3.5	Localización del robot mediante odometría.....	70
2.3.6	Sensores.....	74
2.4	DISEÑO DEL CONTROLADOR LÓGICO DIFUSO.....	77
2.4.1	Diseño del FLC en MATLAB.....	78
2.4.2	Modelado de los conjuntos difusos.....	79
2.4.3	Prueba de funcionamiento del controlador en MATLAB.....	82
2.4.4	Implementación del controlador lógico difuso en lenguaje C.....	84
2.5	SISTEMA DE NAVEGACIÓN FINAL.....	90
2.5.1	Algoritmo de navegación.....	91
2.5.2	Comunicación entre el robot y el PC.....	93
2.5.3	Interfaz gráfica de usuario.....	94
2.6	SISTEMA DE LOCALIZACIÓN BASADO EN RADIOFRECUENCIA.....	94
2.6.1	Módulos XBee de Digi.....	95
2.6.2	Modo de operación.....	95
2.6.3	Desarrollo del sistema de localización basado en RSSI.....	97

2.6.4 Propagación de la onda RF.....	98
2.6.5 Pasos para el desarrollo del sistema.....	101
2.6.6 Desarrollo del algoritmo de posicionamiento.....	104
2.7 ESTRUCTURA FINAL DEL SISTEMA.....	105
2.7.1 Diseño y construcción de circuitos impresos.....	106
2.7.2 Baterías utilizadas.....	108
3 PRUEBAS Y RESULTADOS.....	109
3.1 PLATAFORMA ROBÓTICA.....	109
3.2 RESULTADOS CONTROLADOR PID.....	110
3.3 PRUEBA DE NAVEGACIÓN DEL ROBOT.....	112
3.4 PRUEBAS DEL SISTEMA DE POSICIONAMIENTO RF.....	117
3.5 ALGORITMO DE POSICIONAMIENTO.....	120
4 CONCLUSIONES.....	129
5 RECOMENDACIONES.....	130
BIBLIOGRAFIA.....	131
ANEXOS.....	133

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1. Tipos de robots móviles.....	25
Figura 2 Ruedas en un robot móvil.....	26
Figura 3 Tipos de ruedas utilizadas en robótica móvil.....	26
Figura 4 Centro instantáneo de rotación en diferentes robots.....	27
Figura 5 Triciclo clásico con una rueda de tracción.....	28
Figura 6 Robot móvil en configuración diferencial.....	28
Figura 7 Movimientos de robot en configuración diferencial.....	29
Figura 8 Robot tipo oruga.....	29
Figura 9 Robot tipo oruga.....	29
Figura 10 Robot en configuración omnidireccional.....	30
Figura 11 Métodos de estimación de la posición.....	32
Figura 12 Desplazamiento de un robot móvil en configuración diferencial.....	33
Figura 13 Posición de un robot móvil en configuración diferencial.....	35
Figura 14 Sistema de posicionamiento mediante balizas.....	36
Figura 15 Localización de un objeto por triangulación.....	37
Figura 16 Potencia de señal recibida (RSSI).....	38
Figura 17 Localización de un nodo mediante trilateración.....	38
Figura 18 Encoder utilizado para medir desplazamiento y velocidad.....	40
Figura 19 Sensores infrarrojos.....	41
Figura 20 Motor de corriente continua (DC).....	42
Figura 21 Circuito integrado driver L293B.....	42
Figura 22 Esquema interno de un microcontrolador.....	43
Figura 23 Esquema controlador PID(Proporcional Integral Derivativo).....	45
Figura 24 Funciones de pertenencia para representar conjuntos difusos.....	48
Figura 25 Grado de pertenencia en diferentes conjuntos difusos.....	49
Figura 26 Intersección mediante operador AND.....	50
Figura 27 Unión entre dos conjuntos difusos mediante OR.....	50
Figura 28 Complemento NOT de un conjunto difuso.....	50
Figura 29 Estructura de un controlador lógico difuso.....	52
Figura 30 Fuzificación de una entrada definida.....	52
Figura 31 Proceso de agregación de entradas.....	53
Figura 32 Proceso de implicación.....	54
Figura 33 Agregación de los conjuntos de las reglas activadas.....	55
Figura 34 Defuzificación con COA.....	56
Figura 35 Topologías de red Zigbee.....	58
Figura 36 Diseño general de la plataforma.....	59
Figura 37 Motor DC con reducción incluida.....	60
Figura 38 Diseño del robot en AutoCad.....	61
Figura 39 Estructura de la plataforma en acrílico.....	61
Figura 40 Ubicación de los sensores de obstáculos.....	62
Figura 41 Construcción encoder incremental.....	63

Figura 42	Circuito conformador de pulsos para el encoder.....	63
Figura 43	Respuesta de los motores en función del PWM aplicado.....	65
Figura 44	Lazos de control PID de los motores.....	66
Figura 45	Curva de respuesta sistema de segundo orden.....	67
Figura 46	Función de transferencia de un motor.....	69
Figura 47	Respuesta real del motor y función de transferencia.....	69
Figura 48	Respuesta en SIMULINK del controlador PID.....	70
Figura 49	Diagrama de flujo algoritmo de odometría.....	71
Figura 50	Desplazamiento en línea recta.....	72
Figura 51	Desplazamiento describiendo un arco de circunferencia.....	73
Figura 52	Rotación sobre el centro de gravedad.....	74
Figura 53	Sensor Sharp GP2Y0A02YK.....	74
Figura 54	Curva de respuesta Sensor Sharp GP2Y0A02YK.....	75
Figura 55	Función creada para el sensor Sharp.....	76
Figura 56	Ubicación de los sensores Sharp en la plataforma.....	76
Figura 57	Entradas y salidas del controlador difuso.....	79
Figura 58	Conjuntos difusos sensores.....	79
Figura 59	Conjuntos entrada trayectoria.....	80
Figura 60	Conjuntos de salida.....	80
Figura 61	Evaluación de las reglas en MATLAB.....	83
Figura 62	Simulación del FLC en MATLAB.....	83
Figura 63	Función de membrecía triangular.....	85
Figura 64	Aplicación del fuzificador a una entrada.....	86
Figura 65	Proceso de implicación de una regla.....	87
Figura 66	Proceso de agregación de las reglas activadas.....	88
Figura 67	Defuzificación mediante COA.....	89
Figura 68	Proceso realizado por el FLC.....	89
Figura 69	Comparación respuesta FLC en MATLAB y PROTEUS.....	90
Figura 70	Simulación algoritmo de navegación inicial.....	92
Figura 71	Sistemas de referencia asociados al robot.....	93
Figura 72	Comunicación entre el robot y el PC.....	93
Figura 73	Interfaz gráfica de usuario.....	94
Figura 74	Estructura general de una trama API.....	96
Figura 75	Estructura general del sistema de posicionamiento.....	98
Figura 76	Multitrayecto de la onda RF en interiores.....	99
Figura 77	Posición de los módulos XBee en las medidas.....	101
Figura 78	Dispersión encontrada RSSI Vs distancia.....	102
Figura 79	Regresión logarítmica al modelo encontrado.....	103
Figura 80	Método de trilateración con XBee.....	105
Figura 81	Estructura final del sistema.....	106
Figura 82	Diseño y construcción del PCB.....	106
Figura 83	Tarjeta de control y baliza XBee terminadas.....	107
Figura 84	Baterías utilizadas.....	108
Figura 85	Dimensiones del robot móvil.....	109
Figura 86	Robot terminado.....	110

Figura 87	Respuesta del controlador de velocidad.....	110
Figura 88	Respuesta a un escalón multinivel y a un tren de pulsos.....	111
Figura 89	Ciclos de trabajo de los PWM izquierdo y derecho.....	112
Figura 90	Navegación entorno libre de obstáculos.....	113
Figura 91	Posición real del robot en el escenario.....	113
Figura 92	Navegación en distintas posiciones iniciales.....	114
Figura 93	Navegación entorno con obstáculos.....	115
Figura 94	Navegación entorno con 3 obstáculos.....	115
Figura 95	Aplicación de MTH en la navegación.....	116
Figura 96	Estaciones baliza y nodo móvil del sistema.....	119
Figura 97	Área de posicionamiento.....	122
Figura 98	Posicionamiento robot en el centro.....	124
Figura 99	Posicionamiento robot cercano a baliza 1.....	125
Figura 100	Posicionamiento robot cercano a baliza 2.....	126
Figura 101	Posicionamiento robot cercano a baliza 3.....	127
Figura 102	Zona válida para posicionamiento.....	128

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1 Velocidad máxima y mínima de los motores.....	65
Tabla 2 Valores de los parámetros PID con Ziegler Nichols.....	68
Tabla 3 Valores de las constantes PID modificadas.....	69
Tabla 4 Medidas de distancias a diferentes obstáculos.....	77
Tabla 5 Reglas del controlador lógico difuso.....	83
Tabla 6 Funciones triangulares para la entrada posición.....	85
Tabla 7 Identificadores de tramas API más comunes.....	9797
Tabla 8 Resultados de la navegación entorno sin obstáculos.....	117
Tabla 9 Entorno con obstáculos.....	117
Tabla 10 Prueba RSSI a 1 m de distancia con distinta orientación.....	118
Tabla 11 Asociación de potencia a distancia para la baliza 1.....	120
Tabla 12 Variación de la potencia con la altura.....	121
Tabla 13 Posiciones de las balizas emisoras.....	122
Tabla 14 Valores RSSI con distintos ángulos de emisión/recepción.....	122
Tabla 15 Posicionamiento robot en el centro (200, 200).....	123
Tabla 16 Resultado posicionamiento punto (100, 100).....	124
Tabla 17 Resultado posicionamiento punto (300, 100).....	125
Tabla 18 Resultado posicionamiento punto (200, 300).....	126

## LISTA DE ANEXOS

	<b>pág</b>
Anexo A Circuito eléctrico tarjeta de control robot.....	119
Anexo B Circuito eléctrico módulo XBee robot y balizas XBee.....	120
Anexo C Esquema general del robot.....	121
Anexo D Algoritmo de navegación del robot.....	122
Anexo E Código fuente robot móvil.....	137

## GLOSARIO

**Robot:** Máquina controlada por ordenador y programada para manipular objetos y realizar trabajos a la vez que interacciona con su entorno.

**Teleoperación:** Es el conjunto de tecnologías que comprende la operación o el gobierno a distancia de un dispositivo por un ser humano.

**Sistema autónomo:** Son sistemas completos que actúan por sí solos, desarrollando tareas con la mínima intervención humana.

**IA (Inteligencia Artificial):** Capacidad que tienen las máquinas para realizar tareas que en el momento son realizadas por el ser humano, en términos de procesamiento de información para lograr formas de visión, movimiento o comunicación.

**Lógica Difusa (Fuzzy Logic):** técnica de inteligencia artificial que trata de emular la forma de razonamiento humano.

**FLC (Fuzzy Logic Controller):** Controlador lógico difuso, también se conoce como FIS (Sistema de inferencia difuso), aplica la teoría difusa al control de procesos.

**Cinemática de un robot:** Estudia el movimiento del robot con respecto a un sistema de referencia.

**Trayectoria:** Es el camino que debe seguir el robot para realizar una tarea y alcanzar una meta, en los robots autónomos dicha trayectoria es generada por sí mismo, así como las acciones de control para mantenerse y recuperar dicha trayectoria.

**ICC (Centro Instantáneo De Curvatura):** Hace referencia al punto donde se cortan los ejes proyectados de las ruedas de un robot móvil y es sobre este en el cual realiza su rotación.

**PID (Proporcional, integral, derivativo):** Control de un proceso mediante la interacción de tres modificaciones del error dado por una acción proporcional, una integral y una derivativa; con la que se busca establecer una salida en un rango deseado.

**GPS:** Sistema de posicionamiento global, con el cual mediante el uso de satélites se logra estimar la posición geográfica en cualquier parte del planeta de un dispositivo dotado con esta tecnología.

**RSSI (received signal strength indicator):** Valor de la potencia recibida del último paquete enviado con el que cuentan algunos dispositivos emisores y receptores de radiofrecuencia.

**Transeiver:** Dispositivo que combina en un solo paquete el transmisor y receptor de señales de radiofrecuencia.

**RPM:** Acrónimo de revoluciones por minuto a la que gira un motor.

**Zigbee:** Conjunto de protocolos extendidos del estándar IEEE 802.15.4, usado en redes de área personal de baja transmisión de datos, como redes de sensores inalámbricos.

**API (Application Programming Interface):** Interfaz de programación de aplicaciones.

**dBm:** Decibeles referenciados a milivoltios. Unidad de medida utilizada en los sistemas de radiofrecuencia para medir la cantidad de potencia recibida o transmitida entre antenas de un enlace radioeléctrico.

## RESUMEN.

Este trabajo describe el estudio y la aplicación de las técnicas utilizadas para el navegado autónomo de robots móviles en ambientes reales. Se presenta el diseño y construcción de una plataforma robótica que puede ser utilizada para enseñanza e investigación y con la que se pueden aplicar técnicas de navegación y control. El sistema de navegación propuesto incluye el desarrollo de subsistemas fundamentales como son: Sistema de localización, percepción, planificación y control de movimientos y un sistema de comunicación, los cuáles son indispensables cuando se desea lograr el navegado autónomo en ambientes interiores.

Se inicia con el sistema de localización en éste se utilizan métodos para estimar la posición y orientación del robot, conocer su posición en todo momento es fundamental para aplicar la técnica de control requerida. En cuanto a percepción se investiga e implementa sensores para medición de distancias y detección de obstáculos brindando al sistema la capacidad de obtener información del entorno.

Para la planificación y control de movimientos, se diseña e implementa un controlador difuso, capaz de conducir al robot sobre una trayectoria definida, dependiendo de las características del ambiente de navegación, configurando al robot en velocidad y orientación adecuadas, sin perder nunca su destino hacia la meta.

La monitorización del sistema se realiza mediante la tecnología Zigbee, ésta establece la comunicación entre la plataforma y el ordenador desde el cuál se supervisa el comportamiento del robot; adicionalmente el estudio de esta tecnología sirve para mejorar el sistema de posicionamiento, logrando implementar un sistema de localización en interiores basado en la potencia de la señal recibida (RSSI) y el método de triangulación para estimar la posición.

## ABSTRACT

This paper describes the study and application of techniques used for autonomous mobile robot navigated in real environments, the design and construction of a robotic platform which can be used for teaching and research and the techniques that can be applied is presented navigation and control. The proposed navigation system includes the development of key subsystems such as: location system, perception, planning and control of movement and communication system, which is essential when you want to achieve the autonomous kite indoors.

It starts with the location system in which methods for estimating the position and orientation of the robot are used. Knowing your position at all times is essential to apply the technique required control. Regarding perception investigates and implements sensors for distance measurement and obstacle detection system providing the ability to obtain information from the environment. For planning and control of movements, designs and implements a fuzzy controller capable of driving the robot on a defined path, depending on the characteristics of the navigation environment, the robot speed setting and guidance, without ever losing its destination to the goal.

The monitoring system is realized by Zigbee technology, it establishes communication between the platform and the computer from which the behavior of the robot is monitored; further study of this technology is used to improve the positioning system, achieving implement an indoor location system based on the received signal strength (RSSI) and the triangulation method to estimate the position.

## INTRODUCCIÓN

La robótica es una de las aplicaciones más apasionantes de la electrónica, ya que permite la interacción de conocimientos de distintas ramas de ingeniería como son los sistemas informáticos, mecánicos, eléctricos, electrónicos y comunicaciones.

El desarrollo de robots móviles es un campo que en la actualidad ha cobrado gran importancia, debido a la necesidad que tiene el ser humano de desarrollar nuevas tecnologías que replacen sus labores logrando mejoras en la eficiencia y productividad de los procesos, además de la realización de tareas complejas o peligrosas que pongan en riesgo su integridad. Ésta necesidad ha llevado a la construcción de robots cada vez más inteligentes, que puedan desenvolverse en distintos ambientes. Según su nivel de inteligencia éstos pueden clasificarse en teleoperados, de funcionamiento repetitivo y los autónomos ó inteligentes. En los teleoperados, las acciones de control son tomadas por un operario, los de funcionamiento repetitivo trabajan con un programa de control específico e invariante con limitada percepción del entorno, los autónomos perciben y modelan su entorno y desarrollan tareas por si solos, planifican y actúan para alcanzar un objetivo tomando decisiones ante situaciones inesperadas.

El concepto de autonomía ha cobrado gran importancia, hace referencia a la capacidad de realizar las tareas con la mínima intervención humana, los robots autónomos se consideran los más evolucionados en cuanto a procesamiento de información y hacia los cuales apuntan las investigaciones de los últimos años, estos sistemas procesan señales, establecen relaciones, toman decisiones en tiempo de ejecución y adaptan sus planes a las diferentes circunstancias externas.

En el documento se plantea el diseño de un sistema de navegación autónoma para un robot móvil con ruedas, el cuál posee una configuración diferencial para desplazarse en una superficie plana en ambientes interiores. Esta plataforma puede servir para la realización de futuras investigaciones y aplicaciones como técnicas de control, visión por computadora, exploración, inspección, vigilancia, entre otras. La incorporación de técnicas de inteligencia artificial en éste caso la lógica difusa se presenta como una solución al problema de navegado en la toma de decisiones, intentando emular en cierta forma el razonamiento humano.

## **OBJETIVOS**

### **OBJETIVO GENERAL.**

- Diseñar e implementar un sistema de navegado para un robot móvil autónomo.

### **OBJETIVOS ESPECÍFICOS.**

- Realizar una revisión bibliográfica del material existente en el área de robots móviles autónomos, y de esta manera conocer los avances realizados en la actualidad, cuáles son los temas en que se debe investigar y cuáles son los requerimientos y exigencias de la robótica actual.
- Investigar e implementar sensores para la medición de distancias y la detección de obstáculos durante la navegación.
- Diseñar y construir una plataforma para el robot móvil.
- Estudiar y aplicar técnicas para la estimación de la posición y la orientación de un robot móvil.
- Diseñar e implementar un controlador para la navegación con técnicas de inteligencia artificial logrando dotar de autonomía al robot y de esta manera obtener un controlador inteligente que procese información presente en sus entradas (sensores), analice éstos datos y en sus salidas se obtenga un comportamiento adecuado de los actuadores del robot (motores) para alcanzar una meta.
- Implementar un módulo de comunicación inalámbrica para comunicación y monitorización del comportamiento del robot y una interfaz gráfica de usuario en la que se pueda visualizar la información proveniente del sistema.

# 1. FUNDAMENTOS TEÓRICOS

## 1.1 ANTECEDENTES.

En distintas universidades del país y del mundo, el desarrollo de robots móviles autónomos es un campo de investigación muy trabajado en la actualidad, diversos proyectos se han realizado en sobre este tema. A continuación se describen brevemente algunos que utilizan técnicas de inteligencia artificial para controlar sus sistemas, como también implementan sistemas de posicionamiento relativos y absolutos.

- En el 2007, la Universidad Tecnológica de Pereira publicó el artículo: “Navegación de robots móviles mediante comportamientos utilizando lógica difusa”<sup>1</sup>. En éste la lógica difusa es usada para el guiado del robot, pero se trabaja con base a comportamientos lo que significa pequeñas acciones de control para la toma de decisiones a la hora de alcanzar un objetivo, en este trabajo no se tiene en cuenta la evasión de obstáculos.
- En la Universidad de Nariño en el 2009, se desarrolló el trabajo de investigación denominado: “Diseño de un robot oruga teleoperado para exploración visual y toma de muestras ambientales con sensores en lugares de difícil acceso”<sup>2</sup>, en el cuál se implementa un sistema de tracción mediante orugas, se trabajan sensores de ultrasonido e infrarrojo en la detección de colisiones. En este trabajo no se trabaja el nivel de autonomía por ser un robot teleoperado.
- En el 2012 se desarrolló el proyecto: “Indoor location systems based on Zigbee networks”<sup>3</sup>, donde se investiga el potencial de la tecnología Zigbee para crear un sistema de localización en interiores, este sistema es aplicado a la localización de objetos y nodos y no a la robótica móvil.
- Gilaberte Esther desarrolló el proyecto denominado: “Implementación de sensores de ultrasonido en un sistema autónomo de tiempo real”<sup>4</sup>, en el que se investigan técnicas para la detección de obstáculos y medida de distancias en robots móviles, se tratan temas sobre los ultrasonidos, sensores ópticos,

---

<sup>1</sup> H. Parra, L. Ríos, M. Bueno, Navegación de Robots móviles mediante comportamientos utilizando lógica difusa, Scientia Et Technia, vol. XIII, número 034, 2007. 7p.

<sup>2</sup> CHARFUELAN BURBANO, Diego. Diseño de un robot oruga teleoperado para exploración visual y toma de muestras ambientales con sensores en lugares de difícil acceso. Pasto, 2009, 161p. Trabajo de grado, (ingeniería electrónica). Universidad de Nariño. Facultad de ingeniería.

<sup>3</sup> GUTIERREZ, Maria. Indoor location systems based on Zigbee networks. Mikkeli University of Applied Sciecies. Abril 2012. 46p.

<sup>4</sup> GILABERTE, Esther. Implementación de sensores de ultrasonido en un sistema autónomo de tiempo real, tesis, 2003. 143p.

fotorresistencias, logrando con éxito la detección de obstáculos y colisiones. En este proyecto no se aplican técnicas de inteligencia artificial (IA).

## **1.2 ROBÓTICA.**

### **1.2.1 Breve reseña histórica.**

Desde épocas remotas, el hombre ha deseado crear vida e inteligencia artificial, se ha empeñado en dar vida a seres inanimados, construyendo una serie de mecanismos para todo tipo de fines: científicos, investigación o para realizar y agilizar tareas: repetitivas, pesadas y difíciles que por siglos ha tenido que efectuar, o simplemente con el propósito de compañía y entretenimiento. Generalmente tratando que estas máquinas y dispositivos sean capaces de imitar las funciones y movimientos de los seres vivos; particularmente las del ser humano<sup>5</sup>.

El concepto de robótica es muy utilizado en la actualidad; la mayor parte de la gente tiene una idea de lo que es la robótica, sabe de sus aplicaciones y los alcances que puede llegar a lograr; sin embargo, muy poca gente conoce el origen de la palabra robot o tiene idea de los fundamentos de la robótica como ciencia. Resulta interesante como la sociedad tiene una gran familiaridad con el concepto "robot", pero llama más la atención como carece de esta familiaridad con aparatos que son más antiguos y más utilizados que éste, tal es el caso del osciloscopio o las que se consideran parientes cercanos de los robots: las máquinas de control numérico. Tal vez una de las causas principales de esta popularidad hacia los robots, es la mitificación que se les ha dado a través de la historia, coadyuvada y ampliada por la literatura, el teatro y el cine de ciencia ficción.

El término robot aparece por primera vez en 1921, en la obra teatral R.U.R (Rossum's Universal Robots) del novelista y autor dramático checo Karel Capek en cuyo idioma la palabra "robota" significa fuerza del trabajo o servidumbre. Esta obra se refiere a un brillante científico llamado Rossum y su hijo, quienes desarrollan una sustancia química, utilizándola para fabricar robots, teniendo como planes que los robots sirvan a la humanidad de forma obediente para realizar todos los trabajos físicos. Rossum sigue haciendo mejoras para llegar a crear el robot perfecto, pero al realizarlo la historia de la novela experimenta un giro desagradable, cuando los robots "perfectos" comienzan a desobedecer las órdenes para las cuales se habían construido, sublevándose contra sus dueños y en consecuencia destruyendo a toda la humanidad. Por aquellos años la producción en grandes series se había introducido en numerosas fábricas. Se discute ya del poder de las máquinas y la dominación de los hombres por las

---

<sup>5</sup> MEDRANO, Roman. Diseño y construcción de robot móvil autónomo. Universidad michoacana de San Nicolas de Hidalgo. Facultad de ingeniería mecánica. Morelia, Michoacán. Enero, 2009, p.7.

máquinas, argumento de esta obra y de las obras teatrales y películas de los años veinte en los que aparecen trabajadores robóticos.

El término tiene amplia aceptación y pronto se aplica a autómatas construidos en los años veinte y treinta que se exhiben en ferias, promociones de productos, películas y otras aplicaciones más o menos festivas imitando movimientos de seres vivos, incluyéndose en algunos casos funciones sensoriales primarias.

### **1.2.2 Clasificación de robots.**

Dado a la gran variedad de robots que existen en la actualidad, se puede distinguir los robots cuya actividad consiste en fabricar piezas o bienes mediante procesos de manufactura, considerados como robots de uso industrial y los que no se dedican a esta tarea conocidos de manera genérica con el término de robots de servicio<sup>6</sup>.

La federación internacional de robótica (IFR)<sup>7</sup>, define de manera provisional a los robots de servicio como: Robot que opera de manera semi o totalmente autónoma, para realizar servicios útiles a los humanos y equipos, excluidas operaciones de manufactura. Los robots de servicio a menudo consisten en plataformas móviles que son controladas de la misma manera que los robots industriales.

Según su nivel de autonomía, los robots se pueden clasificar en robots teleoperados donde las acciones de control son tomadas por un operario, de funcionamiento repetitivo los cuales trabajan con un programa de control específico e invariante con limitada percepción del entorno. Los autónomos o inteligentes los cuales perciben su entorno y desarrollan tareas por si solos.

El desarrollo de robots móviles responde a la necesidad de extender el campo de aplicación de la robótica, restringido inicialmente al alcance de una estructura mecánica anclada en uno de sus extremos (manipuladores). Se trata también de incrementar la autonomía limitando en lo posible la intervención humana.

## **1.3 ROBÓTICA MÓVIL.**

La movilidad en un robot es lo que lo hace cada vez más útiles y se empieza su utilización en diferentes campos, no es lo mismo un brazo robótico anclado a una estructura, que uno que pueda desplazarse, es allí donde éste está obligado a interactuar con su entorno. Existen diferentes tipos de robots móviles, según el

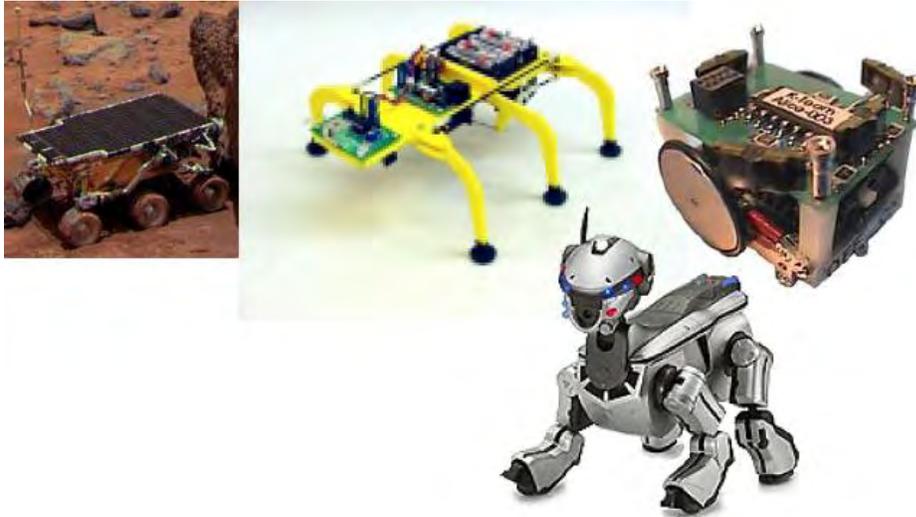
---

<sup>6</sup> OLLERO BATURONE, Aníbal. Robótica: Manipuladores y robots móviles. 1 ed. Barcelona, Marcombo, 2001. p.3.

<sup>7</sup> IFR: Federación internacional de robótica. Organización sin fines de lucro, conformada por organizaciones de robótica de más de 15 países. Su propósito es promover y fortalecer la industria de la robótica en todo el mundo.

medio donde se desplazan, éstos se clasifican en: terrestres, acuáticos o aéreos<sup>8</sup>. La Figura 1 muestra algunos tipos de robots.

**Figura 1. Tipos de robots móviles.**



*Fuente: ACUÑA, Carlos. Diseño y construcción de un robot móvil para desplazarse dentro de un laberinto. Quito. Escuela politécnica nacional. Agosto, 2006. p.12.*

### **1.3.1 Grado de libertad (GDL).**

Es cada uno de los movimientos de desplazamiento y rotación que puede realizar el robot. Un cuerpo que se mueve en dos dimensiones tiene 3 GDL (una rotación y 2 traslaciones). Un cuerpo que se mueve en tres dimensiones tiene 6 GDL (3 rotaciones y 3 traslaciones).<sup>9</sup>

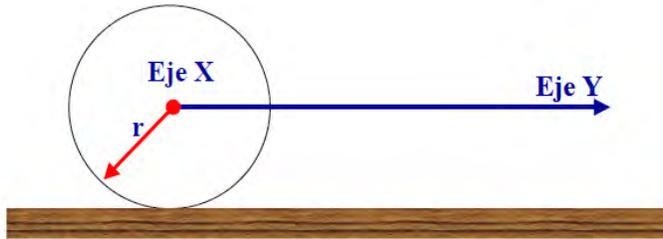
- **Ruedas.** Las ruedas son el elemento que proporciona la capacidad de movilidad en un robot móvil, las cuales se mueven por el contacto superficial (o fricción) con la superficie), idealmente, se desplaza  $2\pi r$  por vuelta, (Ver Figura 2).

---

<sup>8</sup> UNIVERSIDAD DE CHILE. Introducción a la robótica. Departamento de ingeniería eléctrica. Chile. 2010. p.3.

<sup>9</sup> INSTITUTO POLITÉCNICO NACIONAL. Estudio de robots móviles con ruedas. Julio, 2011. p.23.

**Figura 2. Ruedas en un robot móvil.**



*Fuente: UNIVERSIDAD DE CHILE. Introducción a la robótica. Departamento de ingeniería eléctrica. Chile. 2010. p.4.*

Las ruedas se clasifican como:

- **Rueda fija.** El movimiento se produce en la dirección de la rueda.
- **Orientación centrada.** Además del giro de la rueda, existe rotación alrededor del eje vertical que esta dirigido al dentro de la rueda.
- **Orientación descentrada.** Gira sobre el eje de la rueda y rota alrededor del eje vertical situado a una distancia  $d$  desde el centro de la rueda.
- **Rueda sueca.** En este tipo de rueda además de moverse en la dirección de la rueda, se mueve en dirección perpendicular a la dirección de ésta. La Figura 3 ilustra esta clasificación<sup>10</sup>.

**Figura 3. Tipos de ruedas utilizadas en robótica móvil.**



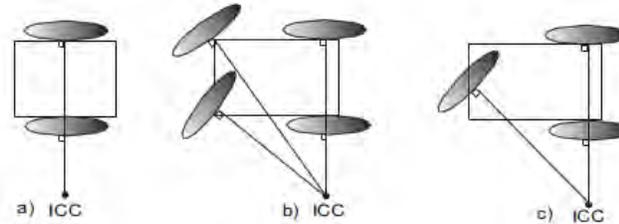
*Fuente: INSTITUTO POLITÉCNICO NACIONAL. Estudio de robots móviles con ruedas. Julio, 2011. p 25.*

<sup>10</sup> UNIVERSIDAD DE CHILE. Introducción a la robótica, Op.cit., p.5.

### 1.3.2 Centro instantáneo de rotación (ICR).

Llamado también centro instantáneo de curvatura (ICC), se define como el punto por el cual cruzan los ejes de todas las ruedas y es el punto alrededor del cual el robot gira en un instante determinado<sup>11</sup>. La Figura 4 muestra como el centro de rotación varía de acuerdo al número de ruedas y disposición de los ejes en la plataforma.

**Figura 4. Centro instantáneo de rotación en diferentes robots.**



*Fuente: INSTITUTO POLITÉCNICO NACIONAL. Estudio de robots móviles con ruedas. Julio, 2011. p 27.*

### 1.3.3 Configuración de robots móviles con ruedas.

Aunque existen diferentes tipos de robots móviles (bípedos, hexápodos, acuáticos, entre otros), este punto solo se enfocará en robots móviles sobre ruedas y para aplicaciones en ambientes controlados (interiores), a continuación se presentan los tipos y características más importantes que rigen el comportamiento de tales robots.

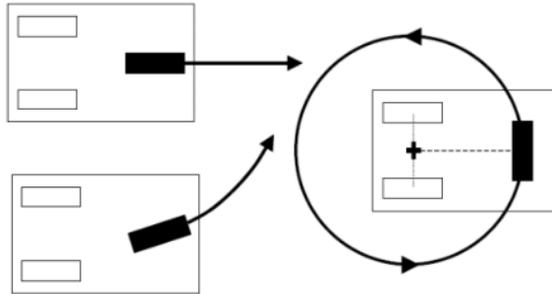
- **Triciclo clásico con conducción sincronizada.** El concepto de diseño más simple se basa en una llanta que tiene la capacidad de impulsar y direccionar, esta configuración requiere dos ruedas pasivas en la parte posterior, dado que siempre son requeridos por los menos tres puntos de contacto para estabilizar la plataforma<sup>12</sup>. La Figura 5 muestra ésta configuración.

- **Guiado en configuración diferencial.** Consta de dos ruedas colocadas en el eje perpendicular a la dirección del robot. Cada rueda es controlada por un motor, de tal forma que el giro del robot queda determinado por la diferencia de velocidad de las ruedas. Así para girar a la izquierda hay que darle una velocidad mayor a la rueda derecha y para girar a la derecha hay que darle una velocidad mayor a la rueda izquierda. La Figura 6 presenta una plataforma en configuración diferencial.

<sup>11</sup> Ibid., p.5.

<sup>12</sup> MÚJICA, Dante. Navegación autónoma de un robot móvil dentro de entornos real/virtual. Instituto politécnico nacional. Escuela superior de ingeniería mecánica y eléctrica. México D.F. 2008. p.10.

**Figura 5. Triciclo clásico con una rueda de tracción.**



*Fuente: MÚJICA, Dante. Navegación autónoma de un robot móvil dentro de entornos real/virtual. Instituto politécnico nacional. Escuela superior de ingeniería mecánica y eléctrica. México D.F. 2008. p.11.*

Puesto que siempre es necesario tres puntos de soporte, este tipo de diseños requiere de una o dos llantas pasivas adicionales, dependiendo de la localización de los motores.

**Figura 6. Robot móvil en configuración diferencial.**



*Fuente: Robot móvil en configuración diferencial [en línea].*

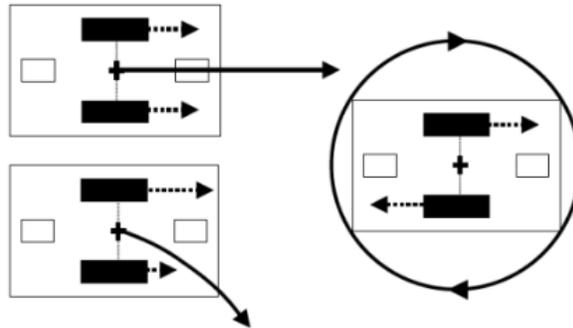
*<<http://www.roboticamovil/configuracióndiferencial.html>> [Citado el 10 de julio de 2013]*

El guiado diferencial es mecánicamente más simple que un robot configurado como triciclo, porque no se requiere que uno de los motores se haga rotar para cambiar la orientación, sin embargo el control de desplazamiento es más complejo debido a que se requiere siempre de la coordinación de ambos motores. La Figura 7 muestra los desplazamientos logrados en el cambio de velocidades de una configuración de este tipo.

La configuración con una sola rueda pasiva no puede disponer de una llanta con motor en el centro por razones de estabilidad. Así que cuando se hace girar, el robot rotará en el punto medio donde están los motores. El diseño con dos ruedas pasivas, una en la parte frontal y otra en la parte posterior, permite que el robot gire sobre su centro de gravedad. Sin embargo, este diseño puede introducir

problemas en el direccionamiento debido a que el contacto con la superficie tiene cuatro puntos de soporte.

**Figura 7. Movimientos de robot en configuración diferencial.**



*Fuente: MÚJICA, Dante. Navegación autónoma de un robot móvil dentro de entornos real/virtual. Instituto politécnico nacional. Escuela superior de ingeniería mecánica y eléctrica. México D.F. 2008. p.13.*

• **Tracción mediante orugas.** Un caso especial de robots con guiado diferencial son los que emplean orugas en lugar de llantas, en esta configuración, el deslizamiento en los giros es muy grande, por esta razón se pierde la exactitud en los cálculos por odometría. Se emplea donde el piso es muy irregular ya que presentan la diferencia de una mejor maniobrabilidad en terrenos rugosos y alta fricción al girar, debido a sus múltiples puntos de contacto con la superficie<sup>13</sup>. Las orugas pueden ser correas o cadenas de acuerdo a las características del terreno. En la Figura 8 se muestra la tracción mediante orugas.

**Figura 8. Robot tipo oruga.**



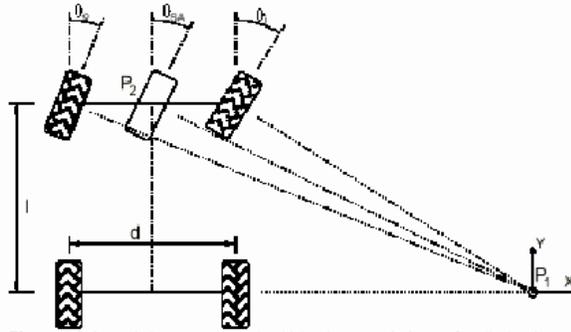
*Fuente: Robot móvil tipo oruga [en línea].*

*<<http://www.xatakaciencia.com/robotica/robots-moviles-i> > [Citado el 10 de julio de 2013]*

<sup>13</sup> MÚJICA, Dante, Op. cit., p.12.

- **Direccionamiento tipo ackermann.** Este sistema de locomoción es llamado también tipo automóvil, consiste de dos ruedas combinadas en la parte trasera que permiten impulsar y dos ruedas también combinadas en la parte delantera cuya función es dar orientación al móvil. El desplazamiento en línea recta no presenta ningún problema dado que las llantas de atrás son impulsadas por un mismo motor (ver Figura 9). Una limitante que tiene el robot es que no puede girar sobre su centro de gravedad<sup>14</sup>.

**Figura 9. Robot en configuración ackermann.**



*Fuente: OLLERO BATURONE, Aníbal. Robótica: Manipuladores y robots móviles. 1 ed. Barcelona: Marcombo, 2001. p.28.*

- **Configuración omnidireccional.** Este tipo de configuración esta provista de ruedas omnidireccionales (rueda sueca), las cuales poseen una corona de rodillos en forma perpendicular al desplazamiento normal de la rueda. Por lo tanto, al aplicar fuerzas laterales a la plataforma, ésta logra moverse sobre los rodillos que giran sobre sí mismos, brindando la ventaja del desplazamiento lateral. A diferencia de las anteriores configuraciones, un robot de este tipo es capaz de moverse en todas las direcciones<sup>15</sup>. La Figura 10 muestra un tipo de rueda omnidireccional y un robot con esta configuración.

**Figura 10. Robot en configuración omnidireccional.**



<sup>14</sup> MÚJICA, Dante, Op. cit., p.12.

<sup>15</sup> Ibid., p.13.

*Fuente: Robot móvil configuración omnidireccional [en línea].*  
< [http://metaltech.gronerth.com/triciclo\\_robot\\_con\\_ruedas\\_omnidireccionales-p36668.html](http://metaltech.gronerth.com/triciclo_robot_con_ruedas_omnidireccionales-p36668.html) > [Citado el 11 de julio de 2013]

## **1.4 SISTEMA DE LOCALIZACIÓN.**

El sistema de localización en robots móviles hace referencia a la capacidad de conocer la posición del robot respecto a un sistema de coordenadas. Para que un robot móvil pueda afrontar tareas como generar trayectorias, evitar obstáculos, etc, se requiere que éste sea capaz de determinar su posición y orientación con respecto a un sistema de referencia.

Se pueden definir dos tipos de sistemas de localización: sistema de localización relativo y sistemas de localización absoluto. Los sistemas de posicionamiento relativo son los que permiten estimar la posición del robot desde un punto de referencia inicial conocido, a partir de esto se realizan medidas internas para calcular el desplazamiento respecto al punto inicial. La localización autónoma de un vehículo se plantea como un problema de estimación de los parámetros X, Y,  $\theta$ , a partir de una serie de medidas proporcionadas por los sensores. Los sistemas de posicionamiento absoluto por su parte permiten conocer la posición y orientación de robot con respecto al entorno en el cual se desenvuelven, para ello se trata de conocer la localización respecto a un sistema de referencia absoluto, el cual esta delimitado por estaciones de transmisión o balizas que emiten señales de referencia periódicamente<sup>16</sup>.

### **1.4.1 Estimación de la posición en un robot móvil.**

La mayoría de robots móviles disponen de encoders para detectar su movimiento, lo que les permite estimar en cada instante de tiempo su posición, por lo cual se suelen usar sistemas de posicionamiento externo<sup>17</sup>. Las principales técnicas utilizadas para la estimación de la posición y orientación de un robot autónomo pueden verse en la Figura 11.

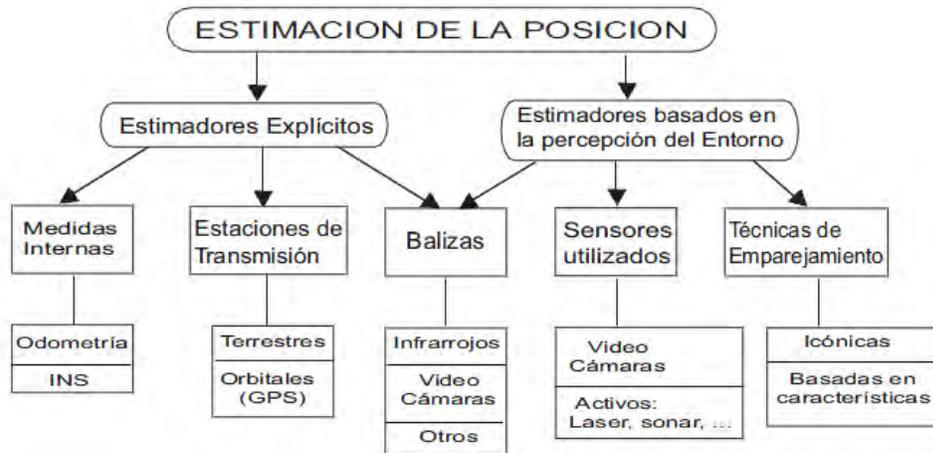
Para la estimación de la posición se distinguen los estimadores explícitos y los estimadores basados en la percepción del entorno.

---

<sup>16</sup> MARTIN, José. Posicionamiento por radiofrecuencia basado en robots colaborativos. Universidad autónoma de Madrid. España. 2012. p.9.

<sup>17</sup> GONZALES JIMENEZ, Javier. Estimación de la posición de un robot móvil. España, 2010. p.3.

**Figura 11. Métodos de estimación de la posición.**



Fuente: GONZALES JIMENEZ, Javier. *Estimación de la posición de un robot móvil*. España, 2010, p.4.

- **Estimadores explícitos.** Estos proporcionan directamente la posición del robot por medio de medidas, sin tener que realizar procesamiento de información para interpretación del entorno. En los sistemas de estimación explícita se distinguen la estimación basada en medidas internas y la estimación basada en estaciones de transmisión. La estimación basada en medidas internas permite el cálculo de valores con los elementos que el robot dispone: encoders, giroscopios, brújulas, tacómetros, etc. La estimación de la posición mediante estaciones de transmisión, se basa en el empleo de elementos que emiten señales, una unidad montada sobre el vehículo y unidades externas que necesitan ser colocadas en posiciones conocidas del entorno<sup>18</sup>. Generalmente la unidad móvil actúa como receptor, mientras que las externas actúan como emisores o señales de referencias.

- **Estimación basada en medidas internas.** La forma más simple de estimar la posición y orientación de un robot móvil, consiste en integrar la trayectoria recorrida por éste a partir de una serie de medidas internas lo que significa el conteo de vueltas dadas por las ruedas, medida de velocidad, aceleración, cambios de orientación, etc.

#### 1.4.2 Estimación de la posición mediante odometría.

La odometría es un método sencillo para estimación de posición y ampliamente utilizado para el cálculo de posicionamiento de los robots móviles, es un tipo de posicionamiento relativo, ya que calcula la posición del móvil, respecto a un punto de partida conocido, se basa en ecuaciones simples que pueden implementarse fácilmente y utilizan información generada por encoders colocados en los motores del robot con lo cual es posible calcular el desplazamiento y por ende su

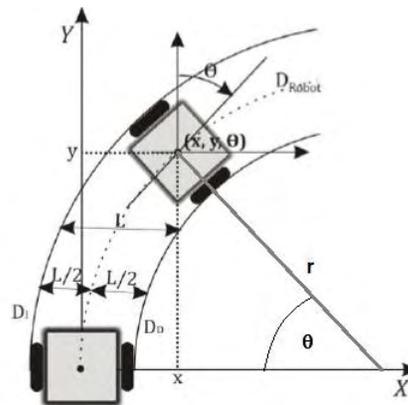
<sup>18</sup> GONZALES JIMENEZ, Javier. Op. cit., p.7.

localización en un punto determinado<sup>19</sup>. La simplicidad y el bajo costo es una gran ventaja que nos ofrece un sistema odométrico, sin embargo es necesaria una calibración constante, debido al desgaste y errores provenientes de la presión de la rueda, desajuste de los ejes, etc. Una desventaja de esta técnica es la acumulación de errores debido a la resolución de los encoders utilizados, imperfecciones en el suelo, deslizamiento de las ruedas y a la carga transportada. La acumulación de errores en el cálculo de orientación causa grandes errores en la estimación de la posición, los cuales se incrementan proporcionalmente con la distancia recorrida por el robot. A pesar de estas limitaciones, la odometría es una parte importante en los sistemas de navegación del robot, las tareas de navegación serán simplificadas, si se mejora la precisión de la odometría. Los encoders como sensores que son, puede estar basados en distintas tecnologías o ser de distintos tipos, pero todos toman medidas del número de vueltas que realiza el eje del móvil. La forma del cálculo del desplazamiento varía de acuerdo al tipo de configuración elegida para el robot.

• **Odometría en un robot móvil con configuración diferencial.**

La configuración diferencial es una de las más usadas, su estructura es muy empleada como se explicó anteriormente, por lo tanto el análisis matemático que a continuación se presenta permite conocer el desplazamiento y determinar la posición para un robot de este tipo<sup>20</sup>. Para calcular el desplazamiento de un robot en configuración diferencial como el mostrado en la Figura 12, se hace uso de la información de los pulsos provenientes de encoders colocados en las ruedas.

**Figura 12. Desplazamiento de un robot móvil en configuración diferencial.**



*Fuente: MÚJICA, Dante. Navegación autónoma de un robot móvil dentro de entornos real/virtual. Instituto politécnico nacional. Escuela superior de ingeniería mecánica y eléctrica. México D.F. 2008. p.16.*

<sup>19</sup> GONZALES JIMENEZ, Javier. Op. cit., p.8.

<sup>20</sup> MUJICA, Dante. Navegado autónomo de Robots móviles en ambientes real/virtual, México, mayo 2008, p.15.

La configuración diferencial consta de dos ruedas, izquierda y derecha. El desplazamiento de cada rueda se calcula teniendo en cuenta el diámetro y el número de vueltas que ésta ha dado. El desplazamiento esta dado por:

$$D_{Rd} = \pi * d * N^{\circ} \text{ de revoluciones derecha} \quad (1)$$

$$D_{Ri} = \pi * d * N^{\circ} \text{ de revoluciones izquierda} \quad (2)$$

Donde  $D_{Rd}$ ,  $D_{Ri}$  son el desplazamiento de la rueda derecha e izquierda y  $d$  es el diámetro de la rueda. El número de revoluciones es la razón entre los pulsos contados por el encoder y el número de pulsos por revolución.

$$N^{\circ} \text{ de revoluciones} = \frac{\text{Número de pulsos contados por el encoder}}{\text{Número de pulsos por revolución}} \quad (3)$$

El desplazamiento total del robot ( $D_{robot}$ ) se calcula sumando los aportes individuales de las ruedas y dividiendo este resultado entre dos.

$$D_{robot} = \frac{D_{Rd} + D_{Ri}}{2} \quad (4)$$

En la Figura 12 se puede observar como el desplazamiento en el plano depende del radio de curvatura  $r$  y la longitud  $L$  entre las ruedas del robot. Entonces el desplazamiento de la rueda se puede calcular de la siguiente forma:

$$D_{Ri} = \theta \left( r + \frac{L}{2} \right) \quad (5)$$

$$D_{Rd} = \theta \left( r - \frac{L}{2} \right) \quad (6)$$

Al restar las ecuaciones (5) y (6), se elimina la variable  $r$  y se obtiene:

$$D_{Ri} - D_{Rd} = \theta * L \quad (7)$$

Despejando  $\theta$  podemos conocer la orientación del robot en radianes respecto a la posición inicial  $\theta_0$ .

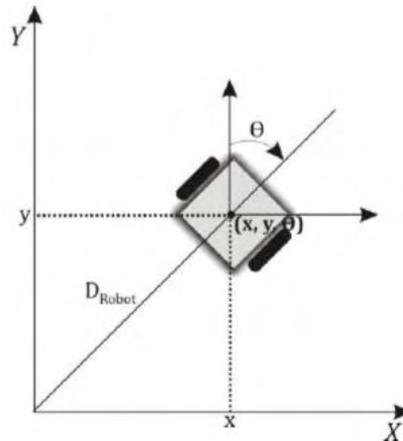
$$\theta = \frac{D_{mi} - D_{md}}{L} + \theta_0 \quad (8)$$

Para encontrar las coordenadas  $x$ ,  $y$  de posición del robot en el plano (ver Figura 13), se utilizan funciones trigonométricas y el ángulo de orientación encontrado en la ecuación (8). La posición se calcula con respecto a las coordenadas iniciales de la siguiente manera:

$$x = D_{robot} \cdot \sin \theta + X_0 \quad (9)$$

$$y = D_{robot} \cdot \cos \theta + Y_0 \quad (10)$$

**Figura 13. Posición de un robot móvil en configuración diferencial.**



Fuente: MÚJICA, Dante. Navegación autónoma de un robot móvil dentro de entornos real/virtual. Instituto politécnico nacional. Escuela superior de ingeniería mecánica y eléctrica. México D.F. 2008. p.17.

#### **1.4.3 Estimación basada en estaciones de transmisión.**

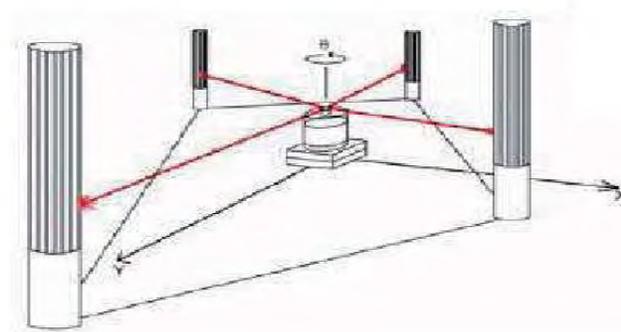
Son un tipo de posicionamiento absoluto, que ha sido utilizado desde hace varias décadas en aplicaciones marítimas y aeronáuticas. En los últimos años con la aparición de robots móviles para exteriores y en particular de vehículos que navegan todo terreno, este tipo de estimación ha cobrado gran importancia. Su ventaja más destacada es proporcionar una posición absoluta de un vehículo en un área suficientemente grande sin tener que estructurar el entorno, permitiendo la navegación en diversos escenarios con el recorrido de grandes distancias<sup>21</sup>. Dentro de esta clasificación se encuentra el denominado GPS, diseñado por el Departamento de Defensa de los Estados Unidos, el cual es operado mediante 24 satélites en 6 órbitas casi circulares (4 satélites por órbita), para realizar el posicionamiento estos transmiten señales de radio en alta frecuencia y utilizando al menos 3 satélites, el receptor calcula por triangulación la altitud, latitud y altura del vehículo de forma "instantánea y continua".

• **Estimación mediante Balizas:** Este tipo de sistemas permite determinar la posición del vehículo en un entorno restringido, mediante la ubicación en el escenario de navegación de un determinado número de balizas ubicadas en posiciones conocidas como se ilustra en la Figura 14. Una baliza es un sistema que emite algún tipo de señal, dependiendo de la tecnología utilizada puede ser radiofrecuencia, ultrasonidos e infrarrojos. La posición es determinada en base al principio de triangulación, bien a partir de medidas de distancias, medidas de ángulos o combinaciones de las dos. El número mínimo de balizas requeridas dependerá del tipo de sistema empleado y de la precisión requerida<sup>22</sup>.

<sup>21</sup> OLLERO BATURONE, Aníbal, Op. cit., p.22.

<sup>22</sup> GONZALES JIMENEZ, Javier. Op. cit., p.33.

**Figura 14. Sistema de posicionamiento mediante balizas.**



*Fuente: Estudio de robots móviles con ruedas. Capítulo 1. p.28.*

Se debe garantizar que un número mínimo de balizas estén visibles o disponibles para poder encontrar la posición por triangulación, este método es uno de los más precisos, con la desventaja de que se debe configurar apropiadamente el entorno de trabajo y de garantizar que en todo momento un suficiente número de señales estén libres de obstrucciones y dentro del campo visual del sensor.

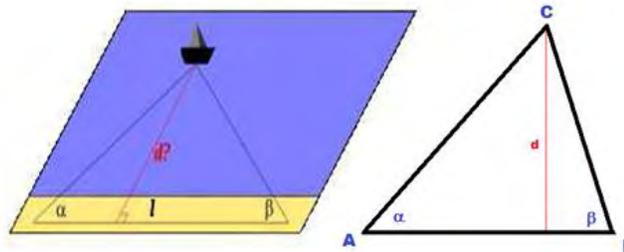
#### **1.4.4 Algoritmos de localización utilizando radiofrecuencia.**

Hay diferentes algoritmos de localización bien conocidos y basados en principios geométricos, la mayoría de los métodos empleados basan su funcionamiento en uno de dos principios: Basados en ángulos (triangulación) y basados en distancias (trilateración), pero la palabra triangulación se usa indistintamente cuando se refiere a cualquiera de los dos métodos.

- **Basados en ángulos (triangulación).** Este método basa la localización del nodo móvil en la capacidad del receptor de conocer los ángulos de recepción de las señales emitidas y en el conocimiento de la posición de los emisores. El principal inconveniente de este método es la forma de calcular el ángulo con el que el emisor está enviando una señal, ya que se requieren sistemas de antenas direccionales muy precisas y la aplicación posterior de algoritmos de localización basados en fórmulas trigonométricas, lo que hace que el coste computacional sea bastante elevado. Todo esto repercute de manera muy negativa en el coste final de cada uno de los nodos<sup>23</sup>. Un ejemplo práctico de la implementación de este método es su utilización para la localización de los barcos en alta mar. (Ver Figura 15).

<sup>23</sup> MARTIN, José. Posicionamiento por radiofrecuencia basado en robots colaborativos. Universidad autónoma de Madrid. 2012, p.23.

**Figura 15. Localización de un objeto por triangulación.**



*Fuente: DIAZ, Santiago. Redes de sensores inalámbricas aplicada a robótica colaborativa. Universidad autónoma de Madrid. Escuela politécnica superior España.2009. p. 14.*

La Figura 15 muestra el principio de cálculo de la posición utilizado por el método de triangulación. Se conocen las posiciones de los nodos A y B y se pretende encontrar la posición del nodo C, los receptores A y B son capaces de detectar el ángulo con el que llega la señal, hallando la distancia  $d$  se habrá posicionado localmente el nodo C. Dicha distancia se halla usando la siguiente fórmula trigonométrica, siendo  $l$  la longitud entre los nodos A y B<sup>24</sup>.

$$d = \frac{l}{\frac{1}{\tan \alpha} + \frac{1}{\tan \beta}} \quad (11)$$

- **Basados en distancias (trilateración).** Este método se basa en obtener las distancias absolutas desde el nodo móvil a cada emisor y calcular la posición del receptor basándose en los datos de distancias medidas. Para ello es necesario conocer la posición de los emisores, las distancias entre estos y el nodo a posicionar, dichas distancias se calculan haciendo uso de la magnitud física medida de la señal. Esta magnitud puede ser tiempo de llegada (ToA) o potencia recibida (RSSI)<sup>25</sup>.

- **Medida de distancia mediante TOA (Time of Arrival).** También llamado tiempo de vuelo, se basa en la medida del tiempo que tarda la señal en llegar desde el nodo emisor al receptor, conociendo la velocidad de propagación, se puede calcular la distancia. Una desventaja de esta técnica es la necesidad de la sincronización exacta de los relojes de los nodos emisores y receptor, ya que alguna diferencia de tiempos produce incertidumbre y errores en la estimación de la posición.

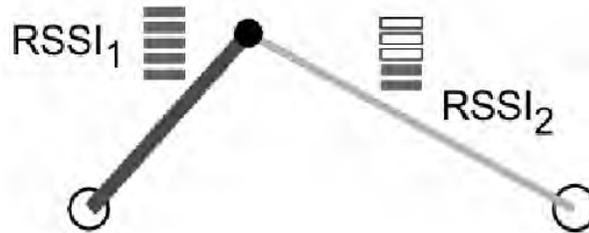
<sup>24</sup> MARTIN, José. Op cit., p.35.

<sup>25</sup> Ibid., p.18.

• **Potencia de señal recibida RSSI (Received Signal Strength Indication).**

Cuando se utiliza emisores de radiofrecuencia, el cálculo de distancia se puede realizar mediante potencia de señal recibida RSSI, se puede estimar la distancia entre dos nodos ya que existe una relación logarítmica entre la potencia recibida y la distancia recorrida<sup>26</sup>, aunque esta relación es puramente teórica porque en la práctica esta potencia sufre además de la atenuación por la distancia, perdidas por otros factores como reflexiones, difracciones, etc. El principio funciona así: Si se recibe una alta potencia de un nodo, significa que el objeto móvil se encuentra cercano, en cambio un nivel bajo de potencia indica que el objeto se encuentra alejado del nodo (Ver Figura 16), para estimar la distancia se aplica un modelo de propagación adecuado al caso.

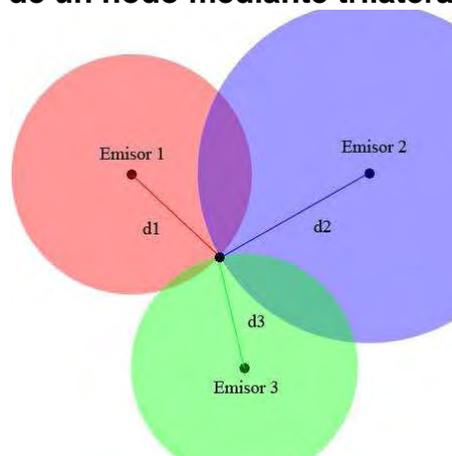
**Figura 16. Potencia de señal recibida (RSSI).**



*Fuente: MARTIN, José. Posicionamiento por radiofrecuencia basado en robots colaborativos. Universidad autónoma de Madrid. España. 2012. p.19.*

Una vez estimada la distancia a la que se encuentra el nodo móvil se puede aplicar el método de trilateración para calcular la posición mediante intersección de circunferencias. La Figura 17 muestra la representación de este método.

**Figura 17. Localización de un nodo mediante trilateración.**



*Fuente: MARTIN, José. Posicionamiento por radiofrecuencia basado en robots colaborativos. Universidad autónoma de Madrid. España. 2012. p.24.*

<sup>26</sup> Ibid., p.19.

Para esto el receptor obtiene la distancia 'dx' a la que se encuentra uno de los emisores, esta distancia es utilizada como radio para trazar una circunferencia alrededor del nodo; el receptor puede estar en cualquier punto de dicha circunferencia. Se necesita por lo menos tres circunferencias para aplicar este método y por ende tres balizas emisoras. Mediante la intersección de las tres circunferencias es posible calcular la posición del receptor.

Si consideramos que uno de los emisores está en las coordenadas (0, 0), el segundo está en las coordenadas (a, 0) y el tercer emisor está en las coordenadas (c, b)<sup>27</sup>. La posición en la que se encuentra el receptor se calcula de la siguiente manera:

$$x = \frac{d_1^2 - d_2^2 + a^2}{2a} \quad (12)$$

$$y = \frac{c^2 + b^2 + d_1^2 - d_3^2}{2b} - \frac{c \cdot x}{b} \quad (13)$$

## 1.5 SENSORES UTILIZADOS EN ROBÓTICA MÓVIL.

Un sensor se puede definir como un transductor que convierte algún fenómeno físico en señales eléctricas, que un microcontrolador u otro tipo de sistema pueda procesar y analizar, éste es un elemento que permite al robot percibir ciertas características de su entorno.<sup>28</sup>

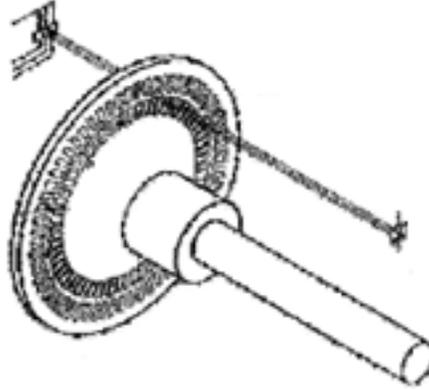
### 1.5.1 Sensores para medir desplazamiento y posición.

- **Encoder:** También llamados codificadores ópticos, estos convierten un desplazamiento rotacional en una señal digital sin la necesidad de un conversor análogo digital. Está compuesto por un disco perforado o con algún tipo de marcas que van colocadas en el eje al cual se desea medir (ver Figura 18). La medida del desplazamiento se realiza mediante el conteo de las interrupciones de un haz de luz las cuales son captadas por un receptor, a medida que el disco gira se generan pulsos, los cuales entran a un circuito eléctrico adicional o microcontrolador que permite contarlos y guardarlos calculando el desplazamiento y la velocidad a la cual gira el eje.

<sup>27</sup>DIAZ, Santiago. Redes de sensores inalámbricas aplicada a robótica colaborativa. Universidad autónoma de Madrid. España. 2009, p.20.

<sup>28</sup>CHARFUELAN BURBANO, Diego. Diseño de un robot oruga teleoperado para exploración visual y toma de muestras ambientales con sensores en lugares de difícil acceso. Pasto, 2009, p 35.

**Figura 18. Encoder utilizado para medir desplazamiento y velocidad.**



*Fuente: OLLERO BATURONE, Aníbal. Robótica: Manipuladores y robots móviles. 1 ed. Barcelona, Marcombo, 2001. p.171.*

Se puede distinguir encoders absolutos y encoders relativos. Un **encoder absoluto** tiene grabado un código binario que permite conocer la posición exacta en la cual se encuentra el eje.

Los **encoders relativos o incrementales** producen por su parte dos ondas cuadradas que permiten contar el número de pulsos y la dirección de movimientos. Mediante el desfase de las señales se puede medir también la dirección de movimiento. Tanto los encoders absolutos como los incrementales pueden llegar a presentar problemas debido a la gran precisión que es necesaria en el proceso de fabricación, además son dispositivos especialmente realizados muy sensibles a golpes, vibraciones, polvo y fricciones.

- **Sensores Infrarrojos:** Este tipo de sensores se basa en la emisión de luz infrarroja y en la detección de ésta por fotodetectores. Los infrarrojos tienen limitaciones ya que pueden recibir interferencias tanto de luz ambiente sobre todo de tipo neón, como también por fuentes de calor. Según la forma como se produzca la emisión y detección de luz, se puede dividir en: Captadores por barrera y captadores por reflexión<sup>29</sup>.

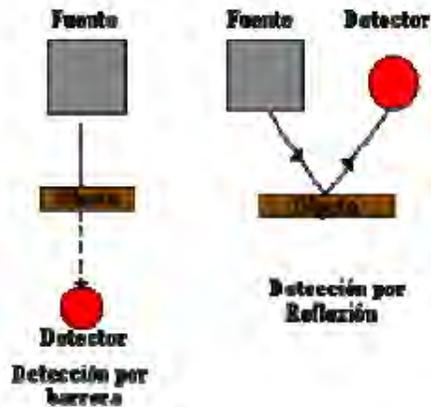
- **Captadores de barrera.** Detectan la existencia de un objeto, porque este interfiere en la recepción de la señal luminosa.

- **Captadores por reflexión.** La señal luminosa es reflejada por el objeto recibida por un captador fotoeléctrico, lo que indica al sistema la presencia de un objeto. La Figura 19 ilustra este tipo de sensores.

---

<sup>29</sup> ACUÑA, Carlos. Diseño y construcción de un robot móvil para desplazarse dentro de un laberinto. Quito. Escuela politécnica nacional. Agosto, 2006. p.16.

**Figura 19. Sensores infrarrojos.**



*Fuente: ACUÑA, Carlos. Diseño y construcción de un robot móvil para desplazarse dentro de un laberinto. Quito. Escuela politécnica nacional. Agosto, 2006. p.16.*

## **1.6 ACTUADORES.**

Los actuadores son elementos que acciona el robot para reaccionar a los estímulos producidos por otras fuentes como por ejemplo los sensores o el sistema de control, sirven para animar la estructura mecánica, éstos convierten la energía de alimentación en movimiento. Existen diferentes tipos de actuadores hidráulicos, neumáticos, eléctricos<sup>30</sup>. En esta parte del trabajo se hace referencia a los actuadores eléctricos (motores), ya que sus características de control, sencillez, precisión y bajo costo hacen que sean muy utilizados tanto en la tracción como en el control de dirección del vehículo. Los motores más utilizados en robótica móvil son: Motores DC, servomotores y motores paso a paso. Para el desarrollo del proyecto se utilizaron motores DC.

### **1.6.1 Motores de corriente continua (DC).**

Son generalmente muy usados en robótica. Existen de distintos tamaños, formas y potencias, pero todos se basan en el mismo principio de funcionamiento, son los más simples, fáciles de encontrar y económicos en el mercado, pero se debe tener en cuenta separar la lógica de control debido a su consumo, para esto se hace necesario usar la técnica de puente H o la utilización de circuitos integrados como el driver. Los motores DC no pueden ser posicionados o enclavados en una posición específica, simplemente giran a la máxima velocidad y sentido de acuerdo al control realizado<sup>31</sup>. La Figura 20 muestra este tipo de motor.

<sup>30</sup> MUJICA, Dante. Op.cit., p.18.

<sup>31</sup> CHARFUELAN BURBANO, Diego. Op cit., p.38.

**Figura 20. Motor de corriente continua (DC).**



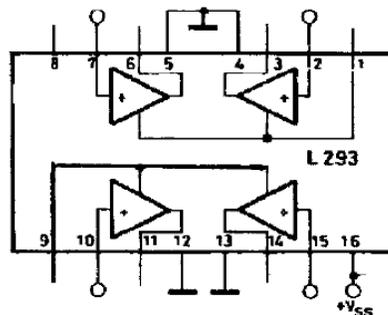
*Fuente: MÚJICA, Dante. Navegación autónoma de un robot móvil dentro de entornos real/virtual. Instituto politécnico nacional. Escuela superior de ingeniería mecánica y eléctrica. México D.F. 2008. p.19.*

### 1.6.2 El driver.

Es un circuito integrado que brinda la posibilidad de conexionar cargas que consumen gran cantidad de corriente, por ejemplo los diferentes tipos de motores. Este permite el control del sentido de giro y velocidad gracias a su construcción interna, el control se logra con el uso de señales de niveles TTL.

• **Driver L293B.** Es un dispositivo integrado de 4 canales capaz de proporcionar una corriente de salida de hasta 1 A por canal, cada canal es controlado por señales de entrada compatibles TTL y dispone de habilitadores de canal que desconecta las salidas de los mismos, posee además una patilla para alimentación de las cargas a controlar<sup>32</sup>. La Figura 21 muestra el encapsulado de 16 pines.

**Figura 21. Circuito integrado driver L293B.**



*PALACIOS E, DOMÍNGUEZ F., LÓPEZ J., Microcontrolador PIC16F84-Desarrollo de proyectos, 1 ed. Alfaomega, 2004, p.469*

<sup>32</sup> THOMSON MICROELECTRONICS. Datasheet Driver L293B [en línea].  
< <http://www.alldatasheet.com/view.jsp?Searchword=L293b> > [Citado 17 de julio de 2013].

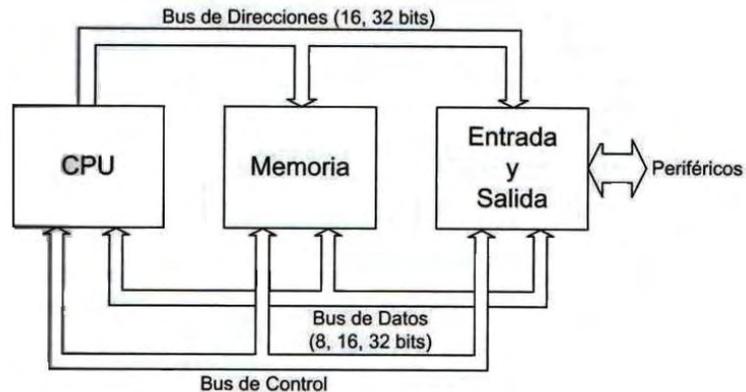
### 1.6.3 Modulación por ancho de pulso PWM.

La técnica PWM (Pulse Width Modulation) o modulación por ancho de pulso permite generar ondas cuadradas con una frecuencia y ciclo trabajo determinado (duty cycle), es decir puede variarse el tiempo en que la señal se encuentra en nivel alto ( $T_{on}$ ), frente al tiempo permanece en bajo ( $T_{off}$ ). De esta forma la tensión media aplicada a la carga es proporcional al tiempo  $T_{on}$ , controlando así la velocidad de motores, luminosidad de lámparas, etc<sup>33</sup>.

## 1.7 MICROCONTROLADORES.

Un microcontrolador es un dispositivo integrado que incluye una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida, convertidores A/D, puerto de comunicación, etc.). Un microcontrolador es definitivamente un circuito integrado que incluye todos los componentes de un computador y debido a su reducido tamaño se puede montar en el propio dispositivo a gobernar. La Figura 22 muestra el esquema general de un microcontrolador y se observan los tres bloques fundamentales: CPU, unidad de memoria, entrada y salida, todos estos interconectados entre sí mediante líneas eléctricas denominadas buses.

**Figura 22. Esquema interno de un microcontrolador.**



Fuente: MICROCHIP. Arquitectura interna PIC [en línea].

< [http://perso.wanadoo.es/luis\\_ju/pic/pic03.html](http://perso.wanadoo.es/luis_ju/pic/pic03.html) > [Citado 18 de julio de 2013].

### 1.7.1 Microcontrolador PIC 18F4550.

Es un microcontrolador de la empresa Microchip, es ideal en aplicaciones donde se trabaja a baja potencia ya que está construido con tecnología nanoWatt, utilizado en múltiples aplicaciones de conectividad ya que dispone de tres puertos seriales y un puerto serial asíncrono. Su ventaja frente a otros de su misma gama

<sup>33</sup> GARCÍA, Eduardo. Compilador C CCS y simulador PROTEUS para microcontroladores PIC. Editorial Alfa-Omega. México. 2008, p.137.

es la gran cantidad de memoria RAM para almacenamiento, posee una memoria de programa Flash mejorada, lo que lo hace ideal en aplicaciones de control embebido, monitoreo y procesos en los cuales se requiere una conexión periódica con el ordenador vía USB<sup>34</sup>. Sus características son:

- Arquitectura tipo RISC avanzada Harvard.
- Memoria de programa flash de 32 Kbytes.
- Memoria RAM de 2 Kbytes.
- Velocidad de la CPU de 12MIPS.
- Periféricos de comunicación USART, SPI, I2C.
- Cuatro temporizadores (Timers 0, 1, 2, 3).
- Conversor ADC.
- 1 Puerto USB full speed, USB 2.0.
- 77 instrucciones de programación.
- Encapsulado de 40 pines
- Pila de 32 niveles.
- 2 módulos CCP (Comparación, captura, PWM).
- Frecuencia máxima de reloj de 48 MHz.

## **1.8 SISTEMA DE CONTROL.**

El motivo principal del uso de un sistema de control es mantener cierta variable de algún tipo en un valor o rango de valores previamente definidos. Los sistemas de control más utilizados en la actualidad son los sistemas en lazo cerrado. La característica primordial de éstos es el hecho que la variable de salida regresa o retroalimenta a la entrada, con lo que se logra comparar continuamente el error existente entre la referencia y la variable controlada, logrando realizar acciones correctivas para alcanzar una referencia dada.

### **1.8.1 Control PID (Proporcional, Integral y Derivativo).**

Se le denomina de esta forma a un proceso controlado por un sistema de compensación en lazo cerrado, basado en un regulador de acciones proporcional, integral y derivativa, de tal forma que se logran características dinámicas estables, dicho de otra forma se logra que el sistema responda a cambios en sus variables en una forma estable<sup>35</sup>. El esquema general de un controlador PID se muestra en la Figura 23, éste hace parte de un sistema de control realimentado, cuyo propósito es hacer que el error en estado estacionario, entre la señal de referencia y la señal de salida del proceso sea cero, lo que se logra mediante el uso de la acción integral. Además el controlador tiene la capacidad de anticipar el futuro a

---

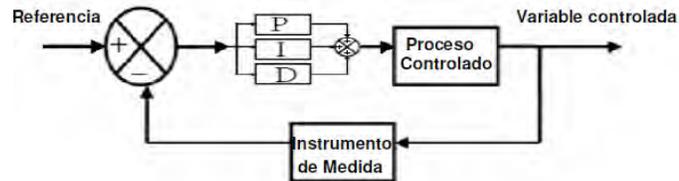
<sup>34</sup> MICROCHIP. Datasheet PIC18F4550 [en línea].

<<http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>>[Citado 20 de agosto 2013]

<sup>35</sup> BARCO JIMENEZ, John. Diseño e implementación de un controlador PID adaptativo basado en un procesador digital de señales dsPIC aplicado al control de procesos industriales. Pasto, 2009, p.40.

través de la acción derivativa que tiene un efecto predictivo sobre la salida del proceso.

**Figura 23. Esquema controlador PID(Proporcional Integral Derivativo)**



OGATA, Katsuhiko. *Sistemas de control en tiempo discreto*. Traducido por José Miguel Aranda Ramírez. 3 Ed. México: Prentice Hall. 1996. p.680.

La acción de control PID en controladores analógicos esta dada por:

$$m(t) = K \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (14)$$

Donde  $e(t)$  es la señal de error actuante que entra al controlador,  $m(t)$  es la salida del controlador (señal manipulada),  $K$  es la ganancia proporcional,  $T_i$  es el tiempo integral y  $T_d$  es el tiempo derivativo.

**Controlador PID digital.** Al discretizar la ecuación (14) se obtiene la función de transferencia de un controlador PID digital, como se explica en Ogata<sup>36</sup>, para el proceso de discretización el término integral se aproxima mediante la sumatoria trapezoidal y el derivativo mediante la diferencia de dos puntos obteniendo la siguiente expresión:

$$M(z) = K \left[ 1 - \frac{T}{2T_i} + \frac{T}{T_i} \frac{1}{1-z^{-1}} + \frac{T_d}{T} (1 - z^{-1}) \right] E(z) \quad (15)$$

$$M(z) = \left[ Kp + \frac{Ki}{1-z^{-1}} + Kd(1 - z^{-1}) \right] E(z) \quad (16)$$

Se puede observar que la relación existente entre las constantes de sintonía del regulador PID en tiempo continuo, y las constantes de sintonía para el PID en tiempo discreto son:

$$Kp = K - \frac{KT}{2T_i} = K - \frac{Ki}{2} \quad \text{Proporcional.} \quad (17)$$

<sup>36</sup> OGATA, Katsuhiko. *Sistemas de control en tiempo discreto*. Traducido por José Miguel Aranda Ramírez. 3 Ed. México: Prentice Hall. 1996. p.679-681.

$$K_i = \frac{KT}{T_i} \quad \text{Integral.} \quad (18)$$

$$K_d = \frac{KT_d}{T} \quad \text{Derivativa.} \quad (19)$$

Donde  $T$  es el tiempo de muestreo de la señal. Nótese que la ganancia  $K_p$  para el controlador PID digital es más pequeña que la ganancia  $K$  del controlador PID analógico por un factor de  $K_i / 2$ .

## 1.9 SISTEMAS DIFUSOS

### 1.9.1 Inteligencia Artificial (IA).

En el transcurso de la humanidad diseñar y construir máquinas capaces de realizar procesos con cierta inteligencia ha sido uno de los principales objetivos de los científicos a lo largo de la historia. La inteligencia artificial se encuentra definida de forma general como la capacidad que tienen las máquinas para realizar tareas que en el momento son hechas por los seres humanos. La inteligencia artificial supone un serio esfuerzo por entender la complejidad de la experiencia humana en términos de procesamiento de información: Visión, movimientos, comunicación, etc. Como ciencia estudia el modo de mejorar las capacidades cognoscitivas de las máquinas, de modo que se acerquen lo más posible al razonamiento humano<sup>37</sup>. En la actualidad existen tres paradigmas en cuanto al desarrollo de la IA.

- Redes neuronales.
- Lógica difusa.
- Algoritmos genéticos.

### 1.9.2 Lógica difusa.

La lógica difusa o borrosa es una técnica de computación flexible, que permite clasificar información del mundo real en una escala infinita, acotada por los valores falso y verdadero. Es básicamente una lógica multivaluada que proporciona un medio para enfrentar situaciones del mundo real, situaciones complejas y dinámicas, que son más fácilmente caracterizadas por palabras que por modelos matemáticos. La lógica difusa es una de las formas más eficientes a la hora de representar el conocimiento empírico humano y de esta manera, obtener un sistema basado en un modelo físico o matemático<sup>38</sup>. En nuestro medio industrial es común encontrar la realización de acciones en las que solo se toma en cuenta información cualitativa poco precisa de un proceso, por ejemplo muy cálido, poco rápido, demasiado grande; estos modelos como respuesta al conocimiento

<sup>37</sup> SAMANIEGO, Javier. Diseño e implementación de un prototipo de asistente de hogar, caso práctico aspiradora autónoma. Escuela superior politécnica de Chimborazo. Facultad de informática y electrónica. Ecuador, Riobamba. 2009, p.13.

<sup>38</sup> CHAHUARA, José. Control neuro-difuso aplicado a una grúa torre. Tesis USMSM, p.12.

empírico adquirido por un operador humano, observaciones fáciles de entender por un experto en el mismo proceso, pero creando un entorno de incertidumbre e imprecisión, porque lo que para un operador es cálido, puede ser que para otro sea poco cálido.

La importancia de la lógica difusa se deriva del hecho de que existen múltiples aplicaciones del mundo real que cumplen con estas condiciones, especialmente el campo de los sistemas y procesos basados en el conocimiento para la toma de decisiones y control<sup>39</sup>.

Algunas características de la lógica difusa que la hacen de tanto interés son:

- Es fácil de entender, sus conceptos matemáticos son sencillos.
- Presenta tolerancia a datos imprecisos.
- Es flexible, su escalamiento es sencillo.
- En ésta se pueden modelar funciones lineales y no lineales.
- Ésta puede ser construida sobre la información de la experiencia de un experto u operarios que manejen un determinado proceso o tengan conocimiento del sistema a controlar.
- Puede ser combinada con otras técnicas de control.

### 1.9.3 Conjuntos difusos

Si analizamos la teoría clásica de conjuntos, un conjunto tiene unos límites bien definidos, denominados “límites crisp” (limitados por el sí o el no), por ejemplo el conjunto B al que pertenecen los números mayores que 10, se representa como:

$$B = \{x/x > 10\} \quad (20)$$

Sin embargo en la teoría de los conjuntos difusos, la transición desde “*pertenecer a un conjunto*” hasta “*no pertenecer al conjunto*” es gradual y se podría decir que un conjunto difuso es un conjunto sin un límite definido ya que contiene elementos solo con un cierto grado de pertenencia. A diferencia de los conjuntos clásicos que se pueden caracterizar ya sea por sus funciones de pertenencia, una descripción o un listado de sus elementos, los conjuntos difusos solo pueden ser caracterizados por sus funciones de pertenencia, la única condición que se debe satisfacer es que deben variar entre 0 y 1. En la lógica difusa un elemento (x) no tiene pertenencia absoluta sino gradual al conjunto A y su función característica no adopta valores en el conjunto discreto {0,1}, sino en el intervalo cerrado [0,1] de su función de pertenencia  $\mu_A$ <sup>40</sup>. Matemáticamente un conjunto difuso se define como:

$$A = \{(x, \mu_A(x))/x \in S\} \quad (21)$$

---

<sup>39</sup> CHAHUARA, José. Op.cit., p.12.

<sup>40</sup> Ibid., p.15.

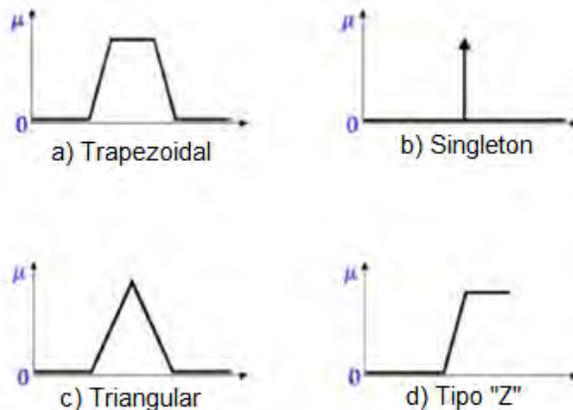
Donde  $A$  es el conjunto que se puede definir como un identificador lingüístico,  $S$  es el universo de discurso y  $\mu_A$  es la función de pertenencia.

- **Variables lingüísticas.** Las variables lingüísticas son elementos fundamentales en un sistema difuso y se las puede encontrar en diferentes categorías, se utilizan variables para nombrar tanto al universo de discurso como también a los subconjuntos que lo componen, por ejemplo en el caso de la variable lingüística “ESTATURA”, existirán también las categorías: “BAJO, MEDIANO, ALTO”, que son llamados términos lingüísticos y representan los posibles valores que puede tomar la variable lingüística “ESTATURA”<sup>41</sup>.

- **Universo de discurso.** El universo de discurso es el que contiene todos los elementos que pueden ser tomados bajo consideración para asignar valores a las variables del sistema, un conjunto difuso toma sus elementos del universo de discurso, por ejemplo el conjunto “Gente Joven” puede tener a los seres humanos como su universo de discurso.

- **Función de pertenencia.** Un conjunto difuso está definido por su función de pertenencia, también llamada función de membresía, ésta es la curva que define con qué grado cada elemento se encuentra incluido en el conjunto difuso. Todos los elementos que se encuentran dentro del universo de discurso, son miembros de algún conjunto difuso en el grado definido por la curva de pertenencia. Esta función puede ser una curva arbitraria, usualmente se utilizan formas estándar como la función triangular, trapezoidal, tipo “S”, exponencial, singleton y pi ( $\pi$ )<sup>42</sup>. La Figura 24 representa algunas funciones de pertenencia frecuentemente utilizadas.

**Figura 24. Funciones de pertenencia para representar conjuntos difusos.**



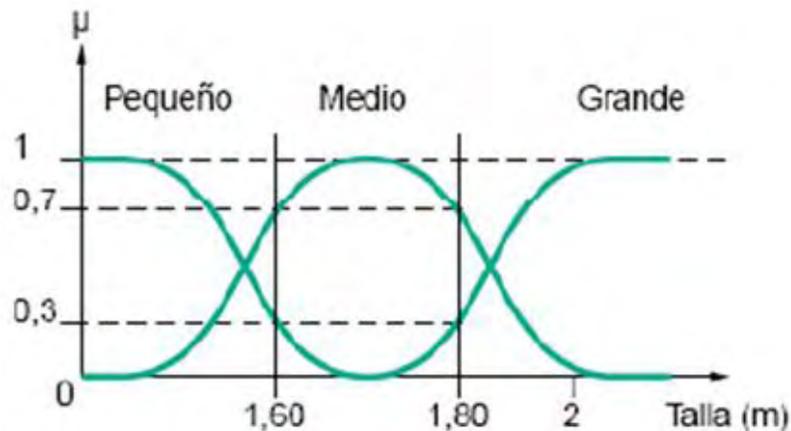
Fuente: CHAHUARA QUISPE, José. Control neuro-difuso aplicado a una grúa torre. Tesis digitales USMSM, p.16.

<sup>41</sup> CHAHUARA, José. Op.cit., p.13.

<sup>42</sup> Ibid., p.14.

- **Grado de pertenencia.** Es el grado con el cual una entrada bien definida es compatible con una función de pertenencia, puede tomar valores entre 0 y 1. Tomando el ejemplo de la variable ESTATURA, a cada valor le corresponderá un grado de pertenencia a los conjuntos difusos PEQUEÑO, MEDIO, GRANDE. Una persona de 1.80 m pertenece al conjunto “talla alta” con un grado 0.3 y al conjunto “talla media” con un grado de 0.7 y al conjunto “talla baja” con un grado de 0.0 (Ver Figura 25). Con esto se podría decir que una entrada tiene un grado de pertenencia en todos los conjuntos difusos<sup>43</sup>.

**Figura 25. Grado de pertenencia en diferentes conjuntos difusos.**



*Fuente: FAJARDO, Darío. Lógica difusa aplicada al control. Universidad de Nariño. Facultad de ingeniería. Departamento de electrónica. 2010, p.31.*

#### 1.9.4 Operaciones entre conjuntos difusos.

A los conjuntos difusos se les puede aplicar diferentes operadores o bien realizar operaciones entre ellos de igual manera que se realizan con los conjuntos clásicos Si se tienen dos conjuntos difusos  $\mu_A(x)$ ,  $\mu_B(x)$  se pueden definir tres operaciones básicas<sup>44</sup>.

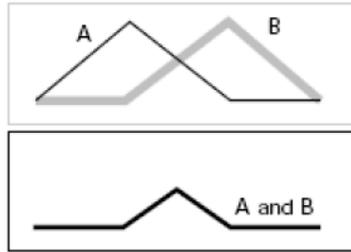
- **Intersección (AND).** El operador correspondiente es *AND*, el resultado de la proposición es el mínimo de los grados de los elementos operados como se ilustra en la Figura 26.

$$\mu_A(x) \text{ And } \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \quad (22)$$

<sup>43</sup> Ibid., p.14.

<sup>44</sup> CHAHUARA, José. Op.cit., p.16.

**Figura 26. Intersección mediante operador AND.**

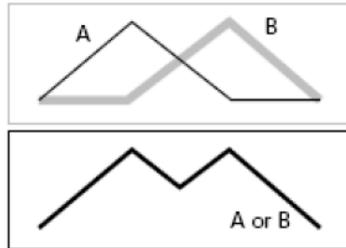


Fuente: The Mathwork inc. Fuzzy Logic Toolbox User's Guide, 2009.

• **Unión (OR).** El operador utilizado es OR, y el resultado es el máximo de los grados de pertenecía de los elementos (Ver figura 27)<sup>45</sup>.

$$\mu_A \text{ Or } \mu_B(x) = \max(\mu_A(x), \mu_B(x)) \quad (23)$$

**Figura 27. Unión entre dos conjuntos difusos mediante OR.**

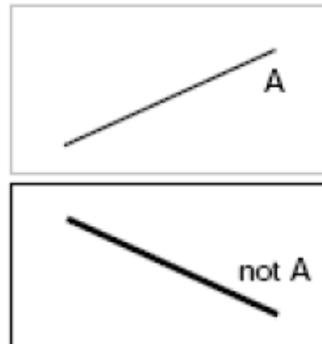


Fuente: The Mathwork inc. Fuzzy Logic Toolbox User's Guide, 2009.

• **Complemento (NOT).** El operador es NOT y el resultado del operador es lo que le falta al grado de verdad para ser 1<sup>46</sup>.

$$\mu_{\text{not } A}(x) = 1 - (\mu_A(x)) \quad (24)$$

**Figura 28. Complemento NOT de un conjunto difuso.**



Fuente: The Mathwork inc. Fuzzy Logic Toolbox User's Guide, 2009.

<sup>45</sup> BARCO JIMENEZ, John. Op.cit., p.59.

<sup>46</sup> Ibid., p.61.

### 1.9.5 Controlador lógico difuso.

Los controladores lógicos difusos *FLC* (Fuzzy Logic Controller) también conocidos como *FIS* (Fuzzy Inference System) son sin duda alguna la aplicación más extendida que ha tenido la lógica difusa y el área donde más se ha extendido es el control de procesos<sup>47</sup>, los sistemas difusos son sistemas basados en reglas, éstas son expresadas como implicaciones lógicas, es decir en forma de sentencias **SI-ENTONCES**. Cuando se utiliza el término *implicación*, en éste se refleja la relación que guarda un hecho derivado de otro. Este hecho puede ser verdadero o falso como es manejado en la lógica Crisp. Si se hace una extensión en la teoría de los conjuntos difusos, pueden existir hechos que van más allá de cumplir solo estas dos condiciones, sino que tienen un cierto grado de verdad. Es aquí donde cabe decir que en el universo no existe un hecho, sino que lo que existe es un grado de verdad en el que el hecho se cumple.

En el uso de técnicas difusas para el control automático, se trata de imitar el comportamiento consciente de un operador humano, controlando procesos no-lineales complejos o plantas de producción las cuales difícilmente pueden ser modeladas por reflexiones físico-matemáticas. La estructura típica de un controlador basado en un sistema difuso consta de un primer bloque que realiza un preprocesado de las variables de entrada que son proporcionadas por un vector al controlador difuso, esta entrada es aplicada a la base de reglas para obtener la salida. Finalmente, la salida puede requerir un procesado final (post procesado) con el fin de adecuarla al proceso que se ha de controlar<sup>48</sup>.

### 1.9.6 Estructura de un FLC.

La estructura interna de un controlador difuso se aprecia en la Figura 29 y consta de:

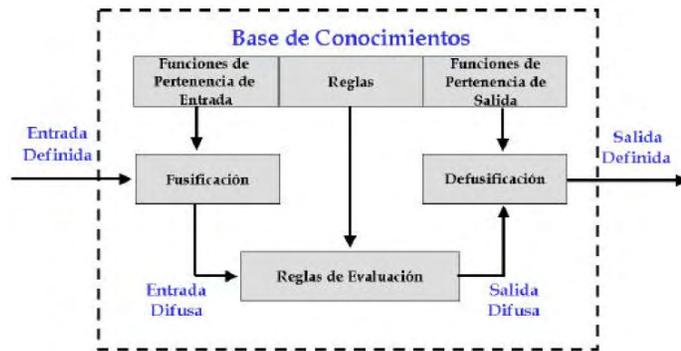
- Fuzificador.
- Base de las reglas difusas.
- Mecanismo de inferencia difusa.
- Defuzificador.

---

<sup>47</sup> DEL BRIO, Martin. Redes neuronales y sistemas borrosos. AlfaOmega, 3 ed. México DF. 2007, p.300.

<sup>48</sup> FERNÁNDEZ, Guillermo. Sistema de navegación autónoma para un robot utilizando lógica difusa. Universidad Argentina de la empresa, Ingeniería en informática. Buenos aires. 2009, p.19.

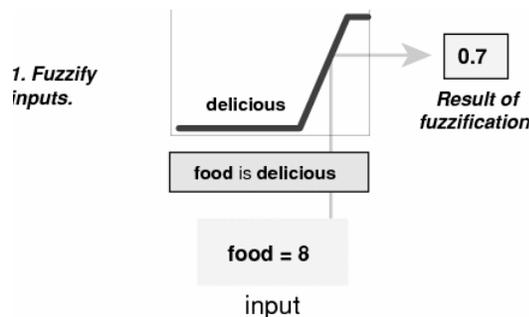
**Figura 29. Estructura de un controlador lógico difuso.**



Fuente: CHAHUARA QUISPE, José. Control neuro-difuso aplicado a una grúa torre. Tesis digitales USMSM, p.22.

- **Fuzificación.** Es la primera tarea del controlador difuso, en esta parte se realiza la traducción de las entradas numéricas en variables lingüísticas las cuales van a ser posteriormente utilizadas, esta operación permite pasar la información del campo real al difuso, consiste en determinar el grado de pertenencia de un valor a un conjunto difuso<sup>49</sup>. Con este procedimiento se establece una relación entre los puntos de entrada y sus correspondientes conjuntos difusos en el universo de discurso de entrada (ver Figura 30).

**Figura 30. Fuzificación de una entrada definida.**



Fuente: The Mathwork inc. Fuzzy Logic Toolbox User's Guide, 2009.

- **Base de las reglas difusas.** La lógica difusa tiene por objetivo simular la manera de pensar de los seres humanos, en este caso se redacta y se almacena el conocimiento en una base de reglas difusas compuestas por sentencias **SI-ENTONCES**, en éstas se describen las condiciones (antecedentes) y las acciones (consecuentes) que deben existir a la hora de tomar una decisión<sup>50</sup>. Se puede decir que las variables de discurso, el universo de discurso, los conjuntos difusos y operadores entran a formar parte de los sujetos y verbos de la lógica difusa. La sintaxis de las reglas es la siguiente:

<sup>49</sup> CHAHUARA, José. Op.cit., p.22.

<sup>50</sup> Ibid., p.22.

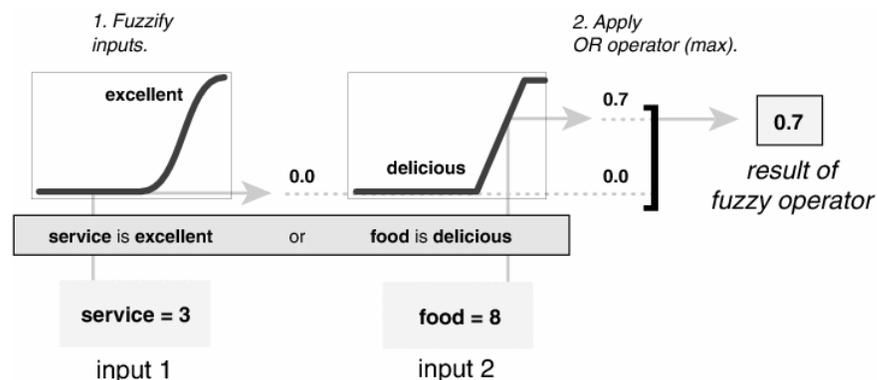
### SI antecedente **ENTONCES** consecuente.

El antecedente de una regla puede tener muchas partes, en todo caso todas las partes del antecedente son calculadas simultáneamente y transformadas en un número utilizando los operadores lógicos. Por su parte el consecuente de una regla puede tener también partes múltiples, pero todos son afectados de la misma manera por el antecedente. Para expresar el conocimiento de forma completa normalmente se hace necesaria la utilización de varias reglas con pesos asociados que se agrupan formando una base o bloque de reglas y tomar una decisión con más exactitud.

• **Mecanismo de inferencia difusa.** Es el proceso que formula la relación de las entradas con las salidas, es aquí donde se involucran todas las piezas del controlador difuso, analiza el efecto de las entradas sobre cada regla, entonces para cada grado de pertenencia asociado a la medición se generan conclusiones. El mecanismo de inferencia puede activar una o varias reglas que al final se agregan formando un solo conjunto para entregar una respuesta<sup>51</sup>. Existen numerosas estrategias de inferencia, pero la técnica de Mamdani es la más utilizada en los sistemas difusos de control existentes debido a su funcionalidad y simplicidad. La inferencia es un cálculo que consiste en tres pasos: Una agregación de las variables lingüísticas de entrada, una composición o implicación y una agregación del resultado.

• **Agregación de variables lingüísticas de entrada ó aplicación del operador.** La agregación determina el grado en el que se cumple el antecedente de cada regla, ahora si la regla contiene varias premisas, éstas suelen estar relacionadas por operadores lógicos como AND y OR (ver Figura 31).

**Figura 31. Proceso de agregación de entradas.**

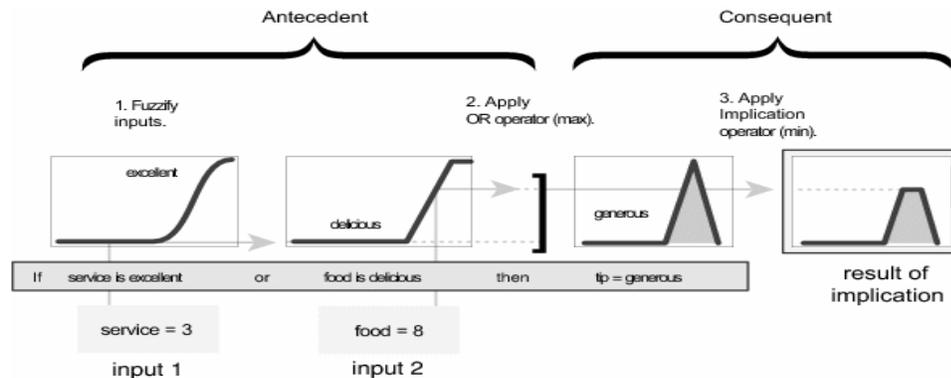


Fuente: The Mathwork inc. Fuzzy Logic Toolbox User's Guide, 2009.

<sup>51</sup> BARCO JIMENEZ, John. Op.cit., p.63.

• **Implicación.** Es el segundo paso en el proceso de inferencia, en donde se comprueba la validez de la conclusión de una regla al relacionar el grado con que se cumple el antecedente con el peso de la misma, es decir indica la medida en que el predicado afecta la conclusión; como el predicado tiene un grado de verdad, entonces la conclusión también se hace verdadera en ese mismo grado. El operador utilizado para el método de implicación es el operador *mínimo* (*min*). La Figura 32 muestra cómo se desarrolla el proceso de implicación de una regla.

**Figura 32. Proceso de implicación.**



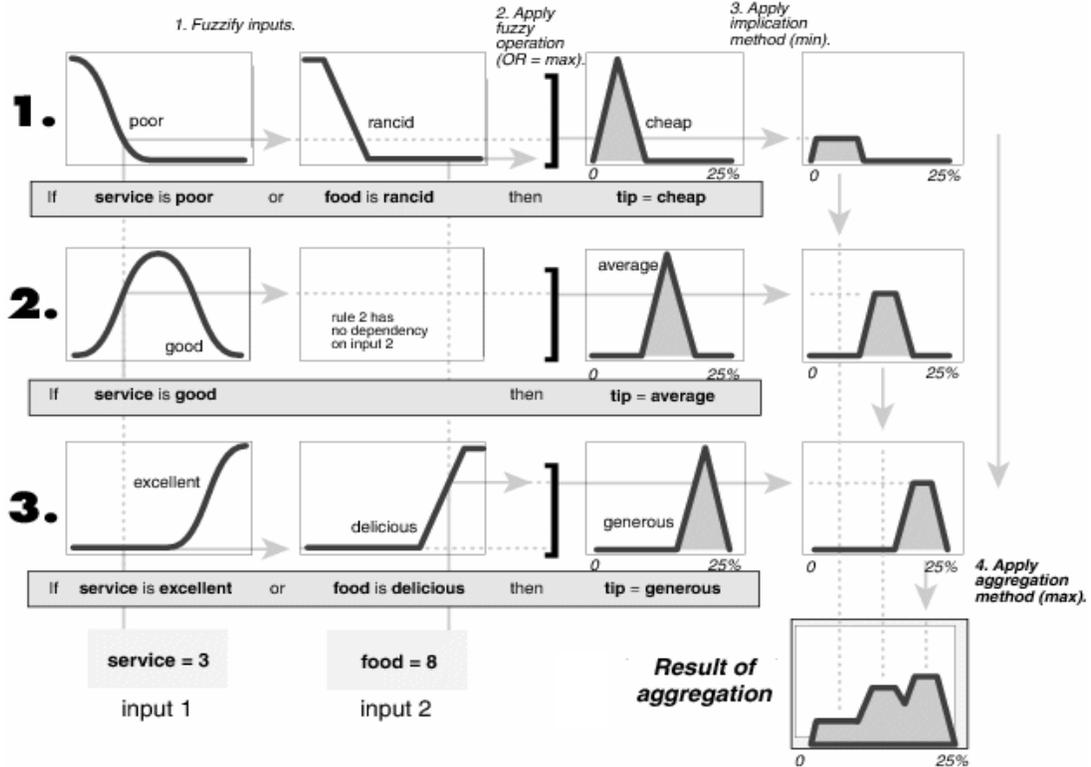
Fuente: The Mathwork inc. Fuzzy Logic Toolbox User's Guide, 2009.

• **Agregación para componer un resultado.** Ya que las decisiones están basadas en la prueba de todas las reglas que forman parte de un sistema difuso, los consecuentes de las reglas deben ser combinados de alguna manera para tomar una decisión; en la agregación los conjuntos difusos que representan las salidas de las reglas son combinados en un único conjunto difuso de salida. El método de agregación utiliza el operador *máximo* (*max*), lo que sería igual a realizar la unión de los conjuntos de salida. La Figura 33 ilustra el proceso de agregación.

• **Defuzificación.** Una vez se ha agregado todos los resultados en un solo conjunto difuso se debe transformar en un valor numérico; cuando se habla de defuzificar se habla de convertir un valor difuso obtenido en el proceso de inferencia en un valor no difuso útil para desarrollar la acción de control respectiva. Existen muchos métodos para realizar éste proceso,<sup>52</sup> en general no hay una técnica que supere a las demás, por lo tanto la elección del método depende de las necesidades de funcionamiento y exigencias de diseño. Las técnicas más utilizadas en el proceso de defuzificación son:

<sup>52</sup> CHAHUARA QUISPE, José. Op. cit., p.50.

Figura 33. Agregación de los conjuntos de las reglas activadas.



Fuente: The Mathwork inc. Fuzzy Logic Toolbox User's Guide, 2009.

- Centroide o centro de áreas (COA).
- Centro de sumas (COS).
- media de los máximos (MOM).
- Promedio de centros (CA).

El método de defuzificación mas utilizado es el centroide o centro de áreas (COA), que consiste en hallar para cada conjunto difuso su área y centro geométrico, es el punto que divide el área bajo la curva del conjunto en dos regiones de iguales (ver Figura 34). Se calcula como el peso promedio de los elementos del conjunto resultante<sup>53</sup>:

$$COA = \frac{\sum \mu(x_i) * x_i}{\sum \mu(x_i)} \quad (25)$$

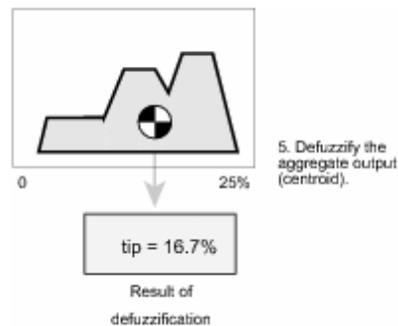
Donde:

$x_i$ : Punto cualquiera en el universo de discurso.

$\mu(x_i)$ : Grado de pertenencia del conjunto.

<sup>53</sup> JUÁREZ, José. Control, capítulo 14. Universidad de Murcia. 2011, p. 25.

**Figura 34. Defuzificación con COA.**



*Fuente: The Mathwork inc. Fuzzy Logic Toolbox User's Guide, 2009.*

## **1.10 ESTUDIO DE LA TECNOLOGÍA ZIGBEE.**

### **1.10.1 Estándar IEEE 802.15.4**

Es un estándar del organismo IEEE para la transmisión de datos a baja velocidad, con bajas latencias y bajo consumo, soporta múltiples topologías para su conexión en red, entre ellas la topología tipo estrella y la topología punto a punto. La topología a escoger es una elección de diseño y va a estar dada por la aplicación a la que se desee orientar<sup>54</sup>.

### **1.10.2 Zigbee.**

La tecnología Zigbee es un estándar de comunicaciones inalámbricas diseñado por la Zigbee Alliance. Es un conjunto de soluciones que pueden ser implementadas por cualquier fabricante. Zigbee está basado en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (Wireless Personal Area Network, WPAN) que por su poca introducción al mercado no es muy conocida a pesar de que no es muy reciente. Las comunicaciones se realizan en la banda libre ISM (Industrial, Scientific and Medical) de 2.4GHz, con una velocidad de transferencia máxima de 250kbps, tiene como objetivo las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías; es especialmente útil para redes de sensores en entornos industriales, médicos, aplicaciones domóticas y robóticas<sup>55</sup>.

### **1.10.3 Dispositivos presentes en una red Zigbee.**

- **El Coordinador:** Es el tipo de dispositivo más completo de la red, es el único que tiene la función de formar la red, es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red. Una vez

<sup>54</sup> PEREZ CAMPANERO, Juan. Clasificador de actividades, Universidad pontificia comillas. Ingeniería en informática. 2009, p.25.

<sup>55</sup> Ibid., p.24.

establecidos estos parámetros, el coordinador puede formar una red, permitiendo a dispositivos routers y end devices unirse a él. Una vez formada la red, el coordinador hace las funciones de router, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información<sup>56</sup>.

- **Router:** Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para transmitir un paquete de información, Interconecta dispositivos separados en la topología de la red, se debe aclarar que un router debe unirse a una red Zigbee antes de poder actuar como router retransmitiendo paquetes de otros destinos.

- **End device:** Son los dispositivos finales y no tienen capacidad de enrutar paquetes, deben interactuar siempre a través de su nodo padre ya sea éste un coordinador o un router, es decir, no pueden enviar información directamente a otro end device. Este tipo de nodo puede estar dormido la mayor parte del tiempo, normalmente estos equipos van alimentados con baterías y su consumo es menor al no tener que realizar funciones de enrutamiento aumentando la vida útil de las baterías.

#### 1.10.4 Topologías de red.

De acuerdo al tipo de aplicación que se quiere desarrollar y los requerimientos de la red a implementar se pueden elegir diferentes topologías. La Figura 35 ilustra las topologías de red que se pueden implementar en el estándar Zigbee<sup>57</sup>.

- **Topología estrella:** En esta topología de red todos los dispositivos que forman la red solo pueden comunicarse con el nodo coordinador, el cual es único y es el encargado de controlar la red, en ésta existen numerosos dispositivos finales los cuales no pueden comunicarse entre ellos.

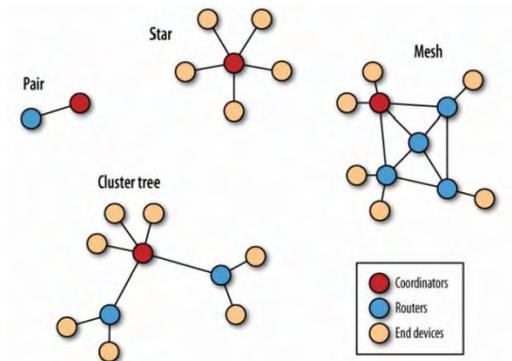
- **Topología árbol:** Es una topología maximizada de la topología estrella, esta formada por varios números de arreglos tipo estrella, lo cual permite aumentar la cobertura de la red. Se utilizan dispositivos intermedios que actúan como un “router coordinador”, estos manejan su parte de la red pero tienen comunicación directa con el nodo coordinador. Existe la desventaja de que los dispositivos finales solo pueden comunicarse con el coordinador principal a través de los router locales, en el caso de fallar alguno de estos los dispositivos finales a su cargo quedaran incomunicados de la red.

---

<sup>56</sup> OYARCE, Andrés. XBee guía de usuario. Ingeniería MCI Ltda, Chile, 2009, p.7.

<sup>57</sup> TREVIÑO, Jesús. Red de sensores inalámbrica genérica. Universidad Autónoma de Ciudad Juárez. México. Mayo, 2011, p.16.

**Figura 35. Topologías de red Zigbee.**



*Fuente: FORTUÑO, Antón. Desarrollo e implementación de una red de sensores Zigbee mediante el dispositivo XBee de Digi. Escuela técnica superior de ingeniería. Mayo, 2012, p.21.*

- **Topología tipo malla:** En este tipo de topología todos los dispositivos de la red mantienen comunicación entre ellos, ya sea directa o a través de enrutadores, manejados todos por un nodo coordinador, solucionando el problema anteriormente mencionado ya que si el nodo router falla el dispositivo final busca un nuevo camino de los múltiples posibles para llegar el paquete a su destino.

## 2. METODOLOGÍA

Para el desarrollo de este trabajo de investigación, se dividió el proyecto en cuatro fases, con las cuales se buscó alcanzar los objetivos planteados en la propuesta inicial. En su orden se desarrollaron las siguientes fases: Revisión bibliográfica, diseño y construcción de la plataforma, desarrollo de las soluciones propuestas y una fase final de pruebas y resultados. A continuación se describe el desarrollo de éstas y las actividades realizadas.

### 2.1 FASE I: REVISIÓN BIBLIOGRÁFICA.

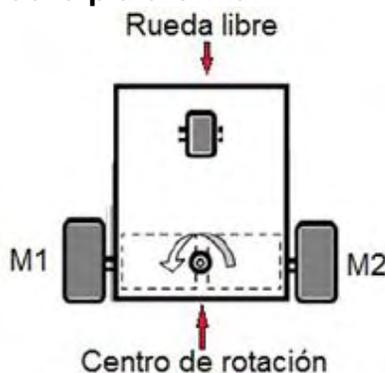
En esta fase inicial del proyecto se revisó en diversas fuentes de información como fueron: libros, documentos digitales, páginas web, artículos de investigación, monografías, etc. Se realizaron también entrevistas a docentes con alta experiencia en el área de la robótica y el control de procesos, esto con el fin de conocer los avances realizados en la actualidad, cuales son los temas en que se debía profundizar, logrando así conocer los requerimientos y exigencias de la robótica actual. Con esto se obtuvieron los conocimientos necesarios sobre las técnicas y tecnologías para aplicarlas posteriormente en desarrollo del proyecto.

### 2.2 FASE II: DISEÑO Y CONSTRUCCIÓN DE LA PLATAFORMA ROBÓTICA

#### 2.2.1 Diseño de la plataforma del robot.

La plataforma del robot hace referencia a la estructura donde van colocados todos los componentes físicos del sistema. Se eligió la configuración diferencial que consta de dos ruedas de tracción colocadas a lado y lado del robot y una rueda pasiva o rueda libre que esta ubicada en la parte posterior, el centro de rotación del robot se localiza en el intermedio de las ruedas de tracción. La Figura 36 muestra el esquema general de la plataforma.

**Figura 36. Diseño general de la plataforma.**



*Fuente: Esta investigación.*

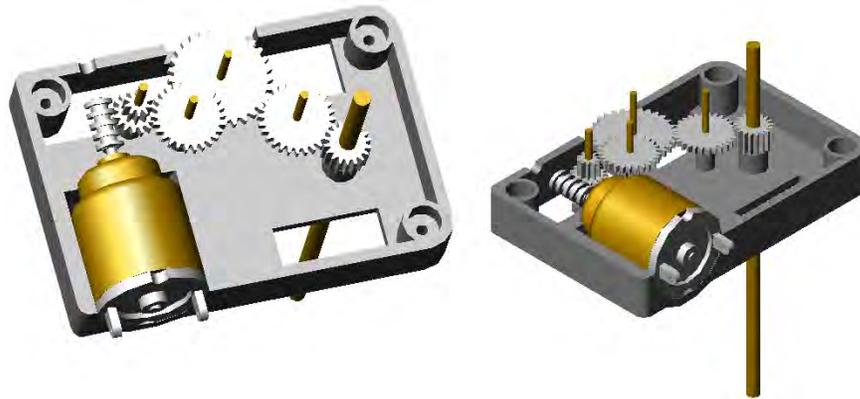
Para este proyecto se construyó una plataforma robótica desde cero, ya que no se contaba con una que cumpla con los requerimientos del proyecto, en el mercado

existen disponibles diferentes tipos de plataformas robóticas, las cuales son costosas lo que aumentaría el costo del proyecto.

### 2.2.2 Elección de los motores.

Después de realizar varias pruebas con diferentes tipos de motores con y sin reducción, se concluyó que se debe trabajar con motores que incluyan un sistema de caja reductora mediante engranajes. En la práctica se trató de construir un sistema de reducción, pero se necesitaban dos reductoras idénticas, que ofrezcan buen torque y velocidades de salida trabajables por lo que esta idea fue descartada en primera instancia. Debido a que en el medio es difícil encontrar este tipo de motores fue necesario buscarlos en varios tipos de juguetes (carros de control remoto, muñecos de movimiento, etc.), de esta manera se consiguieron dos motores con reducción incluida que se acercan a los requerimientos del proyecto. El motor seleccionado se muestra en la Figura 37, donde se puede apreciar su construcción interna. Cabe mencionar la importancia del tornillo sin fin y engranajes cuando se desea desplazarse en terrenos inclinados, ya que este frena el motor si no se aplica voltaje.

**Figura 37. Motor DC con reducción incluida.**



*Fuente: Esta investigación.*

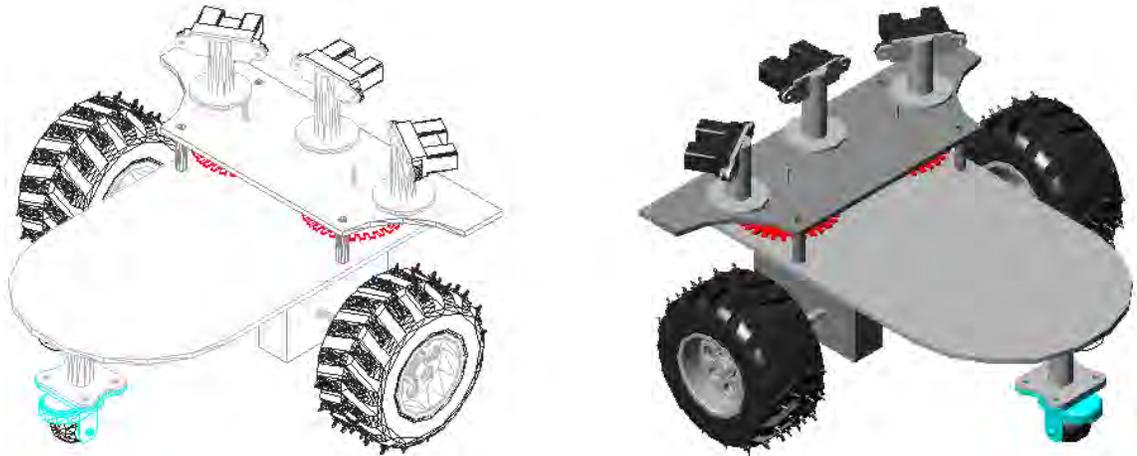
El motor elegido tiene las siguientes características:

- Motor de corriente continua.
- Tornillos sin fin incluido.
- Voltaje nominal de 6V.
- Caja reductora de 75:1.

### 2.2.3 Diseño de la plataforma en AUTOCAD.

Con el fin de facilitar el diseño de la estructura mecánica y tener una idea clara y más precisa de las dimensiones y forma del robot, se diseñó la plataforma en AUTOCAD con las ruedas y motores elegidos. Este programa además de la realización de los diseños a escala real permite visualizar el diseño en 3D. El diseño de la plataforma final se muestra en la Figura 38.

**Figura 38. Diseño del robot en AutoCad.**



*Fuente: Esta investigación.*

#### **2.2.4 Construcción de la plataforma.**

Para esto se toman como base las medidas de los planos realizados en AUTOCAD. Se inicia con una estructura, la cual debe ser resistente y liviana, para reducir al máximo el peso del robot. Sobre esta estructura se fijan los motores DC y la rueda libre. En la caja reductora del motor se realizan algunas modificaciones; se cambia el eje del último engranaje de salida por uno de mayor longitud, lo que permite fijar en este la rueda de tracción con mayor facilidad, se decidió trabajar en acrílico ya que este es un material resistente y muy liviano, los motores se fijan en la cara inferior de la estructura mediante tornillos. Para brindar mayor rigidez y evitar el movimiento de los motores, en la parte trasera del vehículo se fija la rueda libre mediante una estructura cilíndrica. Los elementos para la construcción se muestran en la Figura 39.

**Figura 39. Estructura de la plataforma en acrílico.**



*Fuente: Esta investigación.*

Se creó un segundo piso para la ubicación de los sensores de distancia, dejando un espacio intermedio para la ubicación de los encoders. Este segundo piso se realiza con el fin de proteger los encoders, ya que al encontrarse en el exterior del motor, pueden sufrir roces o golpes con cables que afectarían el elemento y también la medida realizada. Para ubicar los tres sensores de obstáculos sharp, se construyeron tres estructuras cilíndricas con base circular, que permiten sujetar los elementos. Además estas estructuras permiten girar los sensores en ángulos deseados para apuntar en distintas direcciones y mejorar el rango de cobertura (Ver Figura 40).

**Figura 40. Ubicación de los sensores de obstáculos.**



*Fuente: Esta investigación.*

### **2.2.5 Construcción del encoder incremental.**

Para poder medir la velocidad y el desplazamiento del robot, se hace necesaria la utilización de un encoder incremental. Para este proyecto se construyó un encoder con un disco perforado, un emisor y receptor de infrarrojos y un circuito conformador de pulsos. En el disco de plástico se realizaron 32 ranuras, como se muestra en la Figura 41, es recomendable que el espaciado y tamaño de las ranuras sean equidistantes y de iguales dimensiones ya que con esto se mejora la precisión y se obtienen ondas cuadradas con un ciclo de trabajo del 50%.

El componente emisor y receptor de infrarrojos se conforma por el sensor óptico de barrera CNY-36<sup>58</sup>, éste consta internamente de un diodo emisor y un fototransistor receptor el cual es mostrado en la Figura 42. Para acondicionar la señal de salida se utiliza un circuito conformador de pulsos en este caso una

---

<sup>58</sup> OPTOELECTRONICS. Datasheet CNY-36 [en línea].  
<[http://www.datasheetcatalog.com/datasheets\\_pdf/C/N/Y/3/CNY36.shtml](http://www.datasheetcatalog.com/datasheets_pdf/C/N/Y/3/CNY36.shtml)> [Citado 15 de mayo 2012]

compuerta Schmitt Trigger 74LS14, de esta manera se obtiene una señal digital pura para la cual será procesada por el microcontrolador.

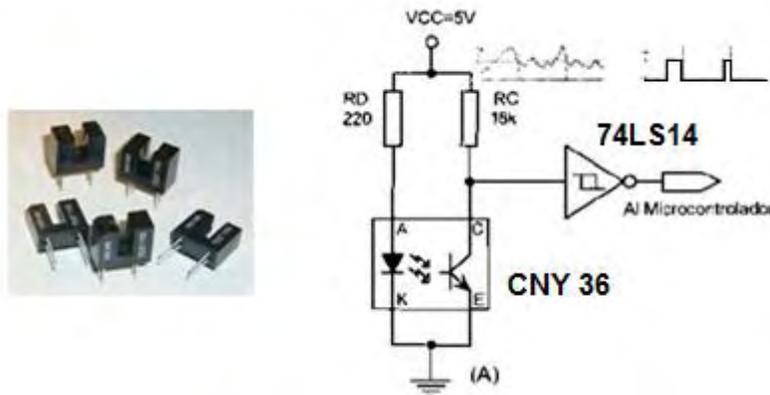
**Figura 41. Construcción encoder incremental.**



*Fuente: Esta investigación.*

Los encoders se colocan en el eje del motor y no en la rueda, ya que así se obtiene una mayor resolución de giro de la rueda y mayor cantidad de pulsos que garantizan precisión en los movimientos.

**Figura 42. Circuito conformador de pulsos para el encoder.**



PALACIOS E, DOMÍNGUEZ F., LÓPEZ J., *Microcontrolador PIC16F84-Desarrollo de proyectos*, 1 ed. Alfaomega, 2004, p.9.

• **Resolución del encoder.** El número de pulsos en una revolución de la rueda se calcula multiplicando la cantidad de ranuras por el factor de reducción. Al poseer 32 ranuras y una reducción de 74.88, la resolución del encoder se calcula como el perímetro de la rueda sobre el número de pulsos por revolución.

$$N^{\circ}\text{pulsos por revolución} = 32 * 74.88 = 4160 \text{ pulsos/revolución.} \quad (26)$$

$$\text{Resolución} = \frac{\pi * \text{DiametroRueda}}{N^{\circ}\text{pulsos por revolución}} = \frac{\pi * 7.5\text{cm}}{4160} = 0.0557\text{mm /pulso} \quad (27)$$

## 2.3 FASE III: DESARROLLO DE LAS SOLUCIONES PLANTEADAS.

En esta fase se desarrollaron los componentes del sistema de navegación, los cuales son los subsistemas de tracción, localización, percepción, control de movimientos y comunicación. A continuación se exponen las actividades realizadas para cada caso.

### 2.3.1 Caracterización de los motores.

Para lograr un correcto desempeño del sistema, se hace necesario conocer las características del motor a utilizar, ya que de esto depende el correcto control y a la vez el cálculo de la posición del robot móvil. Se pudo constatar que cada motor tiene un comportamiento diferente cuando es sometido a una entrada escalón determinada, esto a pesar de que sean del mismo fabricante o tengan referencias iguales, lo que dificulta el control de los actuadores.

### 2.3.2 Medición de la velocidad de los motores.

Para medir la velocidad de los motores se utiliza encoders, para ello primero se debe tener en cuenta la relación entre la velocidad del motor y la frecuencia de los pulsos provenientes del encoder, realizando el conteo de dichos pulsos en un tiempo de muestreo constante. Para calcular la frecuencia de giro se divide el número de pulsos contados entre el tiempo de muestreo, con esto se obtiene la frecuencia o pulsos por segundo (pps). Si tenemos un tiempo de muestreo de 25 ms, entonces la frecuencia se calcula con la ecuación (28). La relación que existe entre la frecuencia y las revoluciones esta dada por:

$$frecuencia = \frac{Pulsos\ contados}{25\ ms} = Pulsos\ contados * 40 \quad (28)$$

$$RPM = \frac{frecuencia * 60}{32} \quad (29)$$

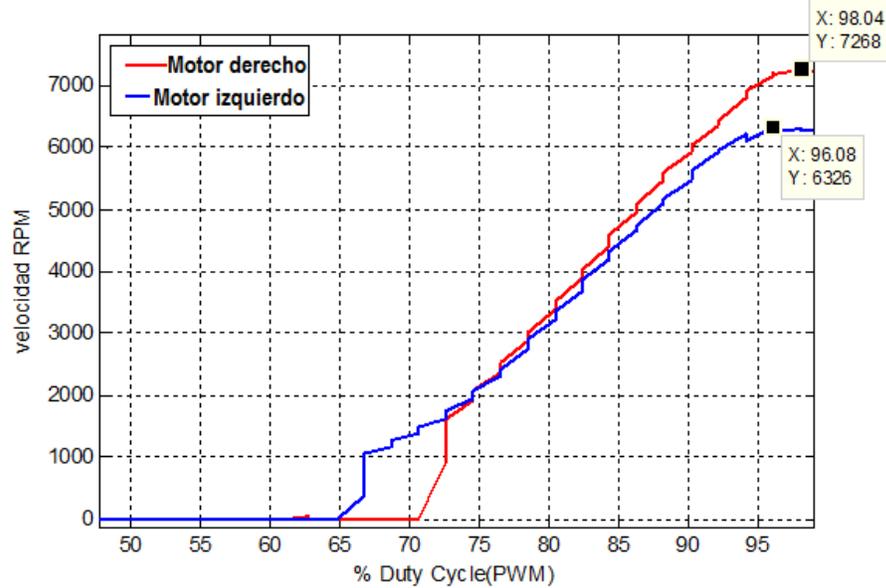
- **Regulación de la velocidad mediante PWM.** Se utiliza la modulación por ancho de pulso (PWM) para controlar la cantidad de energía que se le entrega a los motores y regular su velocidad.

- **Etapas de potencia.** Para controlar simultáneamente los dos motores se utiliza el driver L293B, el cual fue tratado en la sección anterior. Se trabajó con un voltaje de cargas diferente al de alimentación, ya que los motores funcionan a 9 voltios y la lógica de funcionamiento del circuito se trabajó a 5 V para una conexión directa con el PIC. Para regular la velocidad de los motores, las dos señales PWM del microcontrolador son conectadas al habilitador de cada canal del driver, además cuatro líneas del puerto A son utilizadas para controlar el sentido de giro y paro de los motores.

- **Respuesta de los motores.** Para realizar un control de velocidad de los motores, se midieron las velocidades de giro en función del ancho de pulso PWM,

aumentando progresivamente el ciclo de trabajo (duty cycle) de 0% hasta 100%, como se muestra en la Figura 43. El resultado de la medición se muestra en la Tabla 1.

**Figura 43. Respuesta de los motores en función del PWM aplicado.**



Fuente: Esta investigación.

**Tabla 1. Velocidad máxima y mínima de los motores.**

Motor	Velocidad máxima (RPM)	Duty cycle (%)	Velocidad mínima (RPM)	Duty cycle min (%)
Izquierdo	7268	100	200	70.5
Derecho	6326	100	200	64.7

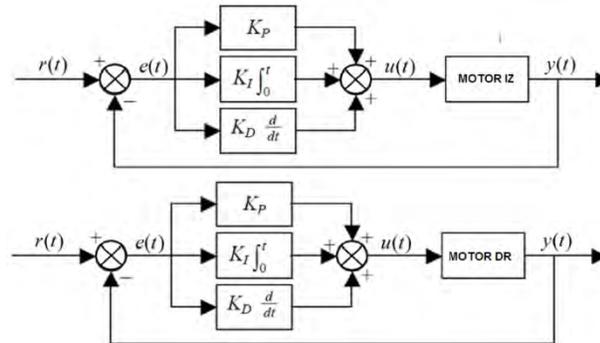
Conociendo la velocidad máxima de los motores y el número de engranajes de la caja reductora se aplica el factor de reducción para obtener la velocidad de salida de las ruedas. Las velocidades máximas fueron 57.5 RPM para el motor derecho y 54.7 RPM para el motor izquierdo.

### 2.3.3 Control PID en microcontrolador.

Como se ha discutido anteriormente, la configuración diferencial necesita de un óptimo control de velocidades para tener un correcto desplazamiento especialmente en línea recta. El algoritmo de control implementado toma una velocidad definida, aplica la potencia necesaria al motor, llevándolo rápidamente hasta la velocidad deseada y la mantiene a pesar de perturbaciones y factores externos que podrían incrementarla o disminuirla. Aunque existen una gran variedad de algoritmos, se implementó un control PID de velocidad mediante dos

lazos de control para cada motor: derecho (*md*) e izquierdo (*m*), esto se muestra en la Figura 44.

**Figura 44. Lazos de control PID de los motores.**



Fuente: Esta investigación.

• **Discretización de algoritmo PID.** El controlador PID digital responde a la ecuación:

$$M(z) = \left[ Kp + \frac{Ki}{1-z^{-1}} + Kd(1 - z^{-1}) \right] E(z) \quad (30)$$

Para implementar el algoritmo en un dispositivo digital, en este caso un microcontrolador, se hace necesario aproximar la derivada y la integral para que tengan mayor similitud con el modelo continuo<sup>59</sup>. La acción proporcional (*Ap*) en su versión muestreada se calcula de la siguiente forma:

$$Ap(k) = Kp * E(k) \quad (31)$$

Para el cálculo de la acción integral (*Ai*), se utiliza la integración rectangular hacia adelante, en donde la acción integral se aproxima por:

$$Ai(k) = Ki * E(k) + Ai(k - 1) \quad (32)$$

La acción derivativa (*Ad*) en su versión muestreada es calculada mediante la regla de diferencia hacia atrás, de la siguiente manera:

$$Ad(k) = Kd(E(k) - E(k - 1)) \quad (33)$$

La acción de control final esta dada por la suma de las tres acciones *Ap*, *Ai*, *Ad*.

$$Control(k) = Ap(k) + Ai(k) + Ad(k) \quad (34)$$

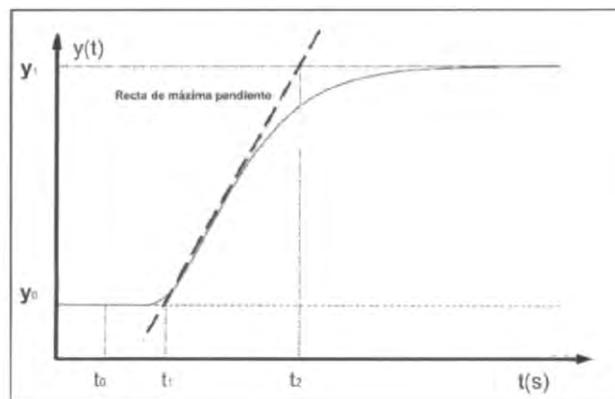
<sup>59</sup> BARCO JIMENEZ, John. Op. cit., p.50.

El proceso de discretización de controladores PID se desarrolla en Kuo<sup>60</sup>, donde se explica con claridad temas referentes a controladores PID, así como formas de calcular las variables que lo componen.

#### 2.3.4 Sintonización PID mediante el método de Ziegler – Nichols.

En los distintos procesos de control es posible aplicar diversas técnicas de diseño, con el fin de determinar los parámetros del controlador que cumplan con las especificaciones de diseño requeridas como son: Tiempo de establecimiento, sobrepaso, etc. Para ello se hace necesaria una buena selección de los parámetros PID, lo que significa establecer valores adecuados para  $K_p$ ,  $T_i$  y  $T_d$ ,<sup>61</sup> esto se conoce como sintonización del controlador. En este trabajo se realizó la sintonización mediante las reglas de Ziegler – Nichols (Z&N), el cual es un método empírico para sintonización. Este proceso requiere la respuesta de la planta en lazo abierto ante una entrada escalón, así que se midió la velocidad ante una entrada escalón de 0 a 5 voltios la cual describió una respuesta tipo “S” como la mostrada en la Figura 45, por esta razón es posible aplicar el primer método de sintonización. En la curva de transferencia se traza una recta de máxima pendiente para aproximar el modelo a un sistema de segundo orden y de dicha respuesta se encuentran los parámetros necesarios para sintonización.

**Figura 45. Curva de respuesta sistema de segundo orden.**



*Fuente: GARCIA, Eduardo. Compilador C CCS y simulador PROTEUS para microcontroladores PIC. Editorial Alfa-Omega, 2008, p.154.*

Una vez obtenidos los parámetros, se puede calcular los valores sugeridos por Ziegler – Nichols para el controlador PID de la siguiente manera:

<sup>60</sup> KUO, Benjamín. Sistemas de control Automático. Traducido por Guillermo Aranda Pérez. Ed 7. México. 1996. p.838.

<sup>61</sup> OGATA, Katsuhiko. Ingeniería de control moderna. Traducido por Miguel Ángel Martínez Ramírez. 3 Ed. México. 1998. p.670.

$$Kp = (1.2 * \gamma_o)/(k_o * \tau_o) \quad (35)$$

$$Ti = 2 * \tau_o \quad (36)$$

$$Td = 0.5 * \tau_o \quad (37)$$

Estas constantes del controlador PID corresponden al tiempo continuo, luego se encuentra su equivalente en el tiempo discreto teniendo en cuenta el tiempo de muestreo de 25ms, se encuentran los valores de sintonización los cuales se muestran en la Tabla 2.

**Tabla 2. Valores de los parámetros PID con Ziegler Nichols.**

Parámetro	Motor izquierdo	Motor derecho
<b>Kp</b>	0.0482	0.051
<b>Ki</b>	0.03	0.04
<b>Kd</b>	0.005	0.005

Estos valores obtenidos se utilizan como parametros iniciales de sintonización, ya que estas reglas están diseñadas para obtener un sobrepaso máximo del 25%, por lo que se hace necesario una sintonización más fina mediante pruebas de ensayo- error con lo que es posible corregir o mejorar la respuesta variando los parámetros de acuerdo a la respuesta obtenida. De todas formas las reglas de sintonización de Ziegler – Nichols, pueden utilizarse para encontrar directamente los parámetros PID o también como valores iniciales de sintonización.

• **Validación de la función de transferencia.**

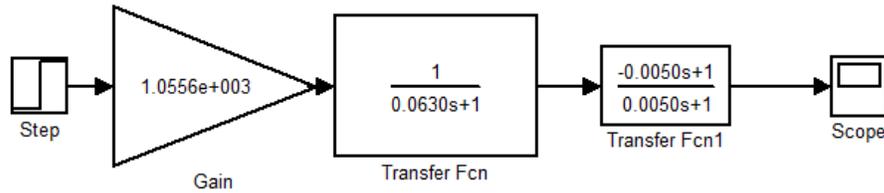
Para realizar la validación del modelo encontrado se realiza la simulación en MATLAB, para ello la respuesta del motor es aproximada a un sistema de segundo orden,<sup>62</sup> de la forma.

$$G(s) = K_0 * \frac{e^{-s\tau}}{(1+s\gamma_o)} \approx K_0 * \frac{1}{(1+s\gamma_o)} * \frac{(1-s\frac{\tau_o}{2})}{(1+s\frac{\tau_o}{2})} \quad (38)$$

Con el modelo del motor encontrado, se verifica la respuesta de las funciones de transferencias mediante la herramienta SIMULINK de MATLAB, para comparar el modelo matemático encontrado y la respuesta real ante la misma entrada escalón como se muestra en la Figura 46.

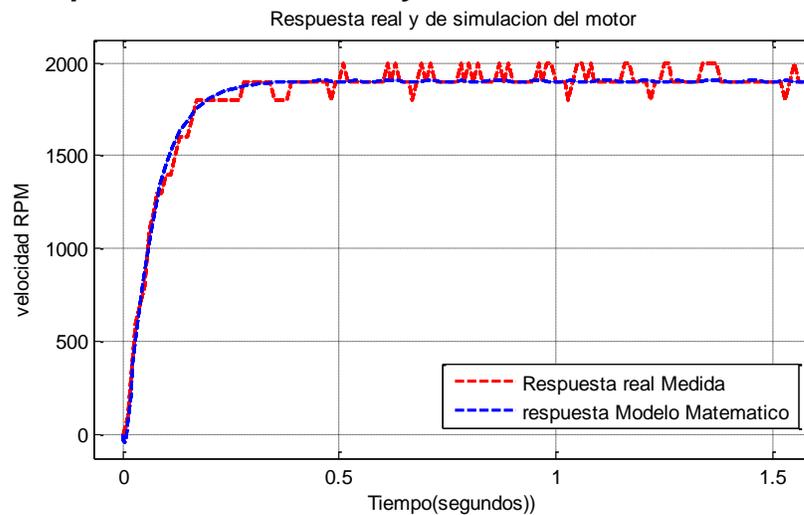
<sup>62</sup> KUO, Benjamín. Sistemas de control Automático. Traducido por Guillermo Aranda Pérez. Ed 7. . Mexico. 1996. p.134.

**Figura 46. Función de transferencia de un motor.**



Fuente: Esta investigación.

**Figura 47. Respuesta real del motor y función de transferencia**



Fuente: Esta investigación.

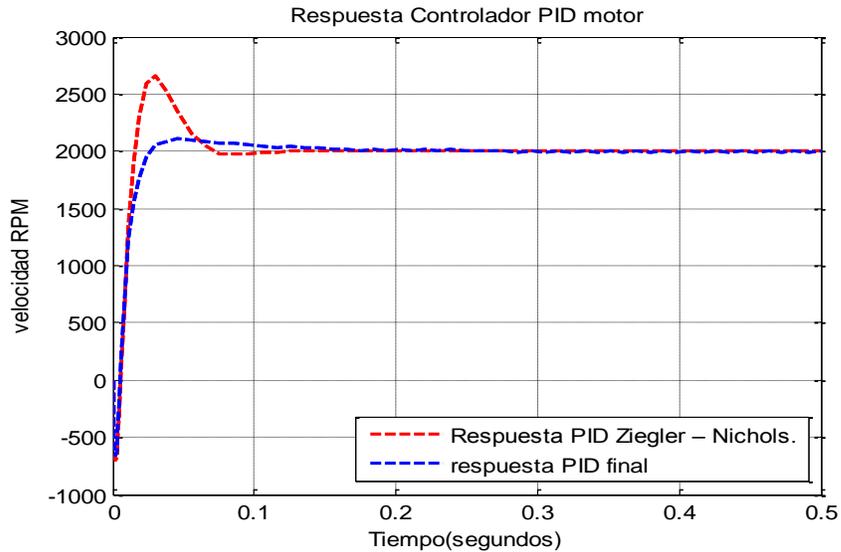
Se mira que el modelo matemático aproxima de manera eficiente la dinámica real del motor, luego se procede a verificar el controlador PID en SIMULINK con las constantes  $K_p$ ,  $T_i$ ,  $T_d$  encontradas mediante los parámetros de Ziegler–Nichols (Z&N). Recordemos que las reglas de sintonización Z&N, están diseñadas para obtener un sobrepaso del 25%, lo que se comprobó en simulación, por esta razón los valores obtenidos sirvieron como parámetros iniciales para posteriormente realizar una sintonización más fina. El resultado de la sintonización Z&N y con la que se logró un mejor desempeño se muestran en la Figura 48.

Se encontró el equivalente PID discreto que es implementado en el algoritmo del microcontrolador y se realizaron pruebas experimentales. De estas pruebas, los valores PID finales con los que se obtuvo un mejor comportamiento en sobrepaso y tiempo de respuesta se muestran en la Tabla 3.

**Tabla 3. Valores de las constantes PID modificadas.**

Parámetro	Motor izquierdo	Motor derecho
<b>Kp</b>	0.0521	0.0485
<b>Ki</b>	0.03	0.035
<b>Kd</b>	0.005	0.005

**Figura 48. Respuesta en SIMULINK del controlador PID.**



Fuente: Esta investigación.

### 2.3.5 Localización del robot mediante odometría.

Para conocer en cada momento la posición en la que el robot se encuentra, se implementó el sistema de localización mediante odometría, éste junto al sistema de control de los actuadores permiten al robot desplazarse y posicionarse correctamente en el espacio. Este sistema se implementa en el microcontrolador y para medir el desplazamiento se hace uso de la información proveniente de los encoders. De las ecuaciones odométricas tenemos que el desplazamiento y orientación del robot se calculan de la siguiente manera:

$$D_{robot} = \frac{\pi * D_R}{2 * kppr * n} (C_i + C_d) \quad (39)$$

$$\theta = \frac{\pi * D_R}{L * kppr * n} (C_i - C_d) + \theta_0 \quad (40)$$

Donde:

$D_R$ : Diámetro de la rueda.

$Kppr$ : Constante de pulsos por revolución o resolución del encoder.

$L$ : Longitud del robot.

$n$ : Factor de reducción de los engranajes entre el motor y la rueda.

$C_i$ : Número de pulsos capturados contador izquierdo.

$C_d$ : Número de pulsos capturados contador derecho.

$X_o, Y_o$ : Posición inicial del robot en el plano.

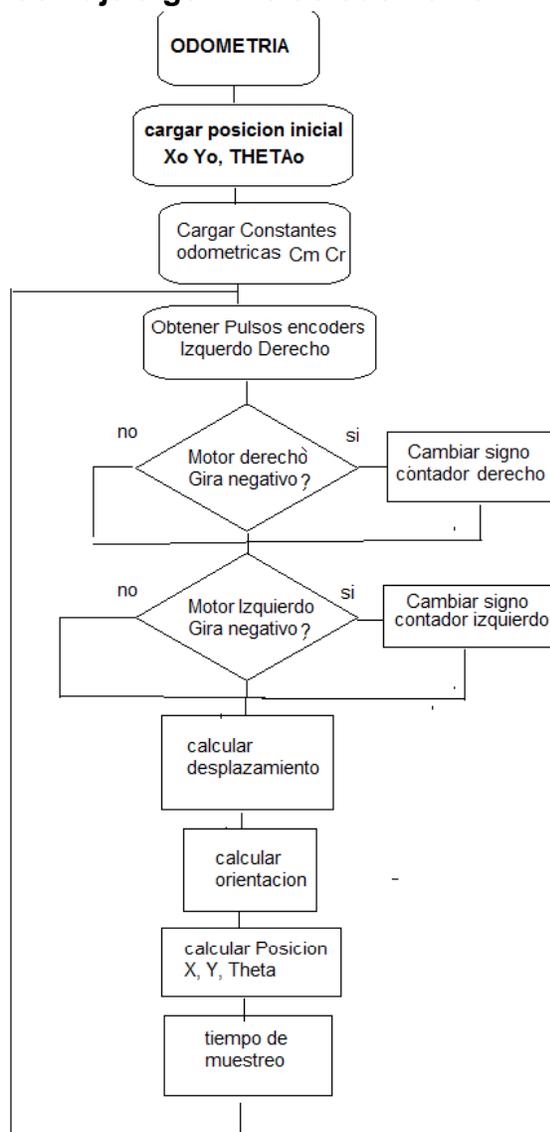
$\theta_0$ : Orientación inicial del robot en el plano.

Como la resolución del encoder, la longitud del robot, el factor de reducción y el diámetro de la rueda permanecen constantes, podemos agrupar estos términos en dos contantes a las cuales les llamaremos  $C_m$  y  $C_r$ , la constante  $C_m$  es utilizada para el cálculo del desplazamiento y  $C_r$  para el cálculo de orientación. El algoritmo que calcula el desplazamiento y la orientación del robot se muestra en la Figura 49. La posición  $(x, y)$  del robot esta determinada por.

$$x = C_m * (C_i + C_d) * \sin \theta + X_0 \quad (41)$$

$$y = C_m * (C_i + C_d) * \cos \theta + Y_0 \quad (42)$$

**Figura 49. Diagrama de flujo algoritmo de odometría.**

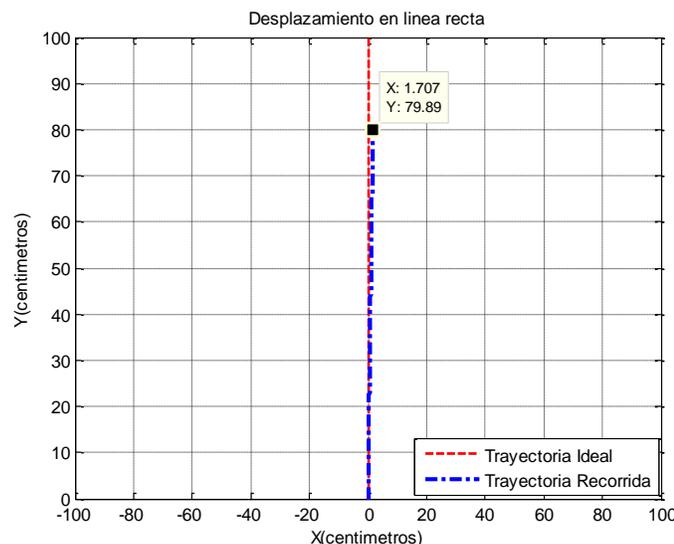


*Fuente: Esta investigación.*

- **Desplazamiento y control del robot.** Para lograr la autonomía de un robot móvil, éste debe ser capaz de describir trayectorias básicas que serán necesarias a la hora de cumplir con una tarea. Simultáneamente el robot debe tener la capacidad de localizarse en el espacio y controlar la velocidad de los actuadores para alcanzar un objetivo. Para ello se han establecido tres comportamientos o formas básicas de desplazamiento en las que el robot puede moverse: Desplazamiento en línea recta, desplazarse formando un arco de radio R y girar sobre su centro de gravedad.

- **Desplazamiento en línea recta:** Dada la configuración diferencial del robot, para que éste se desplace en línea recta, la velocidad de la rueda izquierda y derecha deben ser las mismas y en igual sentido. El algoritmo implementado coloca dos velocidades iguales en cada motor y calcula mediante odometría la distancia recorrida. De esta manera se logró probar el funcionamiento del controlador, ya que las velocidades deben ser iguales para obtener un óptimo desplazamiento en línea recta. El controlador PID garantiza que las velocidades sean lo más iguales posible y la odometría realiza la medida del desplazamiento y orientación en cada instante. Se realizó una prueba experimental de desplazamiento en línea recta partiendo desde la posición de origen (0,0), y desplazándose 80 cm hacia adelante a una velocidad de 30 RPM. La Figura 50 muestra la trayectoria ideal y la trayectoria real.

**Figura 50. Desplazamiento en línea recta.**



*Fuente: Esta investigación.*

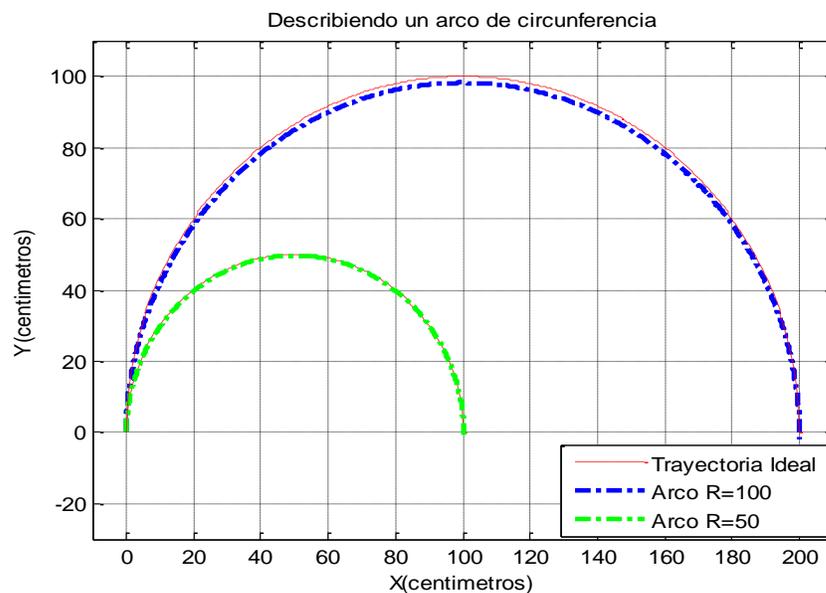
- **Desplazamiento describiendo un arco de circunferencia de radio R:** Para desplazarse formando una curva se hace necesario que una velocidad sea mayor que la otra y para que se describa un arco perfecto las velocidades deben

permanecer constantes, así el robot describirá una curva alrededor del centro de curvatura instantáneo. De la cinemática del robot se puede calcular el radio hacia el centro de curvatura instantáneo (R), dependiendo de las velocidades izquierda (Vi) y derecha (Vd) y la longitud del robot (L)<sup>63</sup>.

$$R = \frac{L}{2} \left( \frac{V_d + V_i}{V_d - V_i} \right) \quad (43)$$

Si se asume un valor para una velocidad y un radio de curvatura, podemos calcular la velocidad de la rueda contraria. El programa realizado permite realizar curvas con distintos radios, tomando una velocidad constante y calculando la velocidad de la otra rueda. Por ejemplo, si se desea describir un arco con un radio de curvatura de 50 cm, el programa calcula que la velocidad izquierda debe ser de 40 RPM y la velocidad derecha de 20 RPM con lo que se logra describir el arco de circunferencia deseado de manera exitosa, esto indica el buen funcionamiento del controlador PID (ver Figura 51).

**Figura 51. Desplazamiento describiendo un arco de circunferencia.**



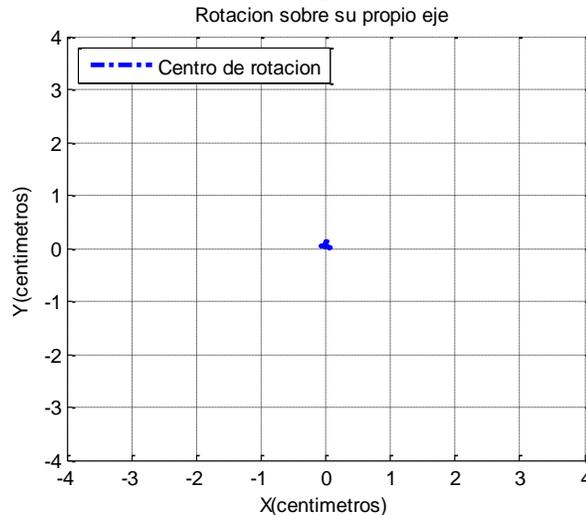
*Fuente: Esta investigación.*

- **Rotación sobre su centro de gravedad:** Una propiedad que posee la configuración diferencial es la capacidad de rotar sobre su centro de gravedad cuando la velocidad de sus ruedas son iguales pero giran en sentido contrario, en este caso no habrá desplazamiento y solo cambiara la orientación del robot. Esto es de gran importancia si se desea cambiar de orientación sin necesidad de

<sup>63</sup> OLLERO, Anibal. Op. cit., p.105.

desplazarse como si sucede con otras configuraciones. En la Figura 52 se aprecia que el robot se mantiene sobre un punto.

**Figura 52. Rotación sobre el centro de gravedad.**



*Fuente: Esta investigación.*

### 2.3.6 Sensores.

El sensor utilizado para medir desplazamiento y velocidad es el encoder y el sensor utilizado para la medición de distancias y detección de obstáculos es el sensor Sharp GP2Y0A02YK.

- **Sensor Infrarrojo GP2Y0A02YK.** Es un sensor para medir distancias con procesamiento de señal integrado y un voltaje análogo de salida proporcional a la distancia. Este tipo de sensor es muy utilizado en robótica, con el cual es posible medir distancias entre el robot móvil y los obstáculos que se encuentren en su camino, consta de un emisor infrarrojo y un sensor lineal el cual detecta el haz de luz emitido por el diodo infrarrojo. su imagen se aprecia en la Figura 53.

**Figura 53. Sensor Sharp GP2Y0A02YK.**



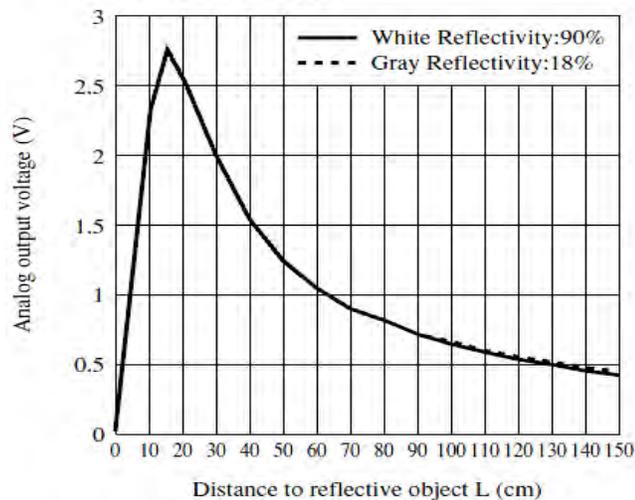
*Fuente: bluetip- Electrónica y Robótica [en línea].*  
< <http://www.bluetip.com.uy/GP2Y0A02YK> > [citado el 20 de agosto de 2013].

Sus principales características son:

- Rango de alcance efectivo: Entre 20 y 150 cm.
- Tiempo de respuesta típico: 39ms
- Salida analógica.
- Alimentación 5V.

El voltaje de salida varía entre 0.4 y 2.7 V y su curva de respuesta tiene forma exponencial (ver Figura 54). En la gráfica se observa que en la medición de distancias inferiores a 20 cm el sensor no es operativo y la salida de voltaje no puede ser considerada como válida para el cálculo de la distancia<sup>64</sup>.

**Figura 54. Curva de respuesta Sensor Sharp GP2Y0A02YK.  
Analog Output Voltage vs. Distance to Reflective Object**



Fuente: SHARP. Datasheet GP2Y0A02YK [en línea].

< [http://www.erasme.org/IMG/gp2y0a02\\_e.pdf](http://www.erasme.org/IMG/gp2y0a02_e.pdf) > [Citado 17 de agosto de 2013].

• **Caracterización del sensor.** La curva de respuesta del sensor no tiene un funcionamiento lineal por eso fue necesario crear una función que se aproxime al modelo real de medición, ya que el sensor entrega un voltaje análogo se utiliza el módulo conversor ADC que tiene incorporado el PIC.

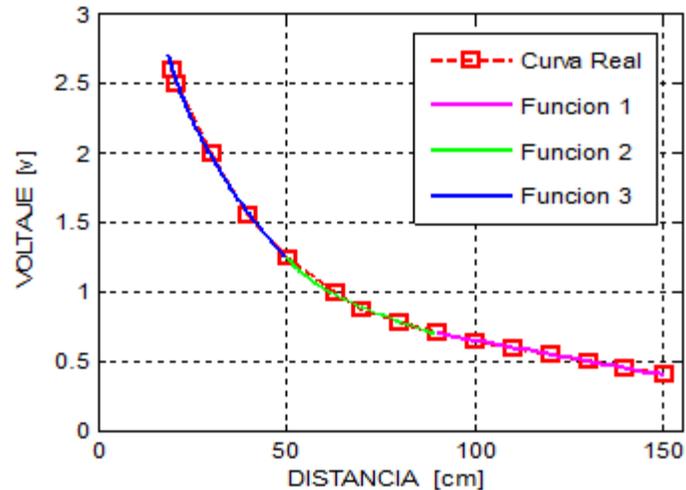
Para obtener una mejor resolución del ADC, se trabajó con un voltaje de referencia de 3V. La función implementada en el microcontrolador convierte el voltaje medido en un valor de distancia, se realizó un muestreo de distancia y voltaje entregado, a estos datos se aplicó regresión cuadrática con la ayuda de MATLAB, se dividió la respuesta en tres funciones cuadráticas de la forma

<sup>64</sup> SHARP. DataSheet Sensor GP2Y0A02YK. [en línea].

< [http://www.erasme.org/IMG/gp2y0a02\\_e.pdf](http://www.erasme.org/IMG/gp2y0a02_e.pdf) > [Citado 10 de junio de 2013].

$ax^2+bx+c$ , obteniendo los coeficientes necesarios. La función creada se muestra en la Figura 55.

**Figura 55. Función creada para el sensor Sharp.**



*Fuente: Esta investigación.*

El robot móvil consta de tres sensores infrarrojos GP2Y0A02YK en la parte frontal del vehículo como se ilustra en la Figura 56, esto con el fin de tener una mayor cobertura del espacio de navegación, de esta manera pueden ser detectados obstáculos en el frente y en los extremos del robot, algo de gran importancia cuando se desea girar en algún sentido. Los sensores de los extremos se ubicaron con un ángulo de inclinación de 15 grados, mejorando el campo de medición cuando el robot se desplaza.

**Figura 56. Ubicación de los sensores Sharp en la plataforma.**



*Fuente: Esta investigación.*

• **Prueba de funcionamiento de los sensores.** Se realizaron las pruebas para validar la función creada y mirar si las distancias medidas por el sensor son correctas, se tomaron medidas con objetos de diferentes formas, colores, materiales y texturas. La Tabla 4 muestra los resultados obtenidos con el sensor y algunos objetos de prueba a diferentes distancias.

**Tabla 4. Medidas de distancias a diferentes obstáculos.**

Distancia real (cm)	Madera	Caja nylon brillante	Caja cartón rojo	Caja plástico blanco	Caja metal	Caja papel azul
20	20.1	23.7	20.9	20.5	22.3	20.8
30	29.3	35.2	30.8	31.5	34.2	30.4
40	40.4	48.5	41.02	41.6	47.3	41.2
50	49.4	67.7	51.3	51.3	63.1	52.1
60	58.5	88.3	60.7	61.7	80.6	62.8
70	69.8	99.9	71.11	75.7	99.9	75.2
80	79.6	103.3	84.4	80.69	103.8	88.4
90	91.9	127.7	94.3	95.6	154.3	109.2
100	99.3	142.2	103.03	107.26	160	122.1
110	112	160	115.7	118.8	160	116.7
120	122	160	128.9	132.5	160	128.9
130	138	160	138.9	146.9	160	151.1
140	151	160	155	157	160	160
150	160	160	160	160	160	160

De esta prueba se concluye que el color del objeto tiene poca influencia en la medición, en general su comportamiento es bueno especialmente con el color blanco. Cabe mencionar que esta versión del sensor Sharp trae incorporado un circuito el cual ayuda a disminuir la influencia de los colores en la medición<sup>65</sup>. Con respecto a la textura y al material de fabricación de los objetos como papel, madera y tela son elementos que presentan buenas mediciones, pero materiales cubiertos con nylon, materiales brillantes o lisos presentan disminución en el valor del voltaje lo que conlleva a pequeñas desviaciones en la medida.

## 2.4 DISEÑO DEL CONTROLADOR LÓGICO DIFUSO.

Para el control de navegación del robot móvil, se hace necesario que exista un sistema que brinde al robot la posibilidad de tomar sus propias decisiones, de esto depende en gran parte el concepto de autonomía, para ello se planteó la utilización de la lógica difusa, una técnica de inteligencia artificial como método

<sup>65</sup> SHARP. Datasheet GP2Y0A02YK [en línea].  
< [http://www.erasme.org/IMG/gp2y0a02\\_e.pdf](http://www.erasme.org/IMG/gp2y0a02_e.pdf) > [Citado 17de agosto de 2013].

para controlar la navegación, lograr la evasión de obstáculos y por ende alcanzar un objetivo en un ambiente desconocido.

El controlador lógico difuso tendrá dos funciones principales, la primera es mantener al robot en una trayectoria definida que lo llevará directo a la meta, controlando en cada momento la orientación. En caso de encontrar obstáculos en su camino, éste le permitirá evadirlos sin perder su objetivo y recuperar su trayectoria. Dependiendo de la ubicación del obstáculo respecto a la trayectoria se tratará de rodearlo, ya sea por la derecha o la izquierda, para esto el controlador tomará la mejor decisión según sea el caso. La segunda función es controlar la velocidad del vehículo de acuerdo a la información sobre su entorno, reduciendo su velocidad ante la presencia de obstáculos e incrementándola al máximo ante la ausencia de los mismos.

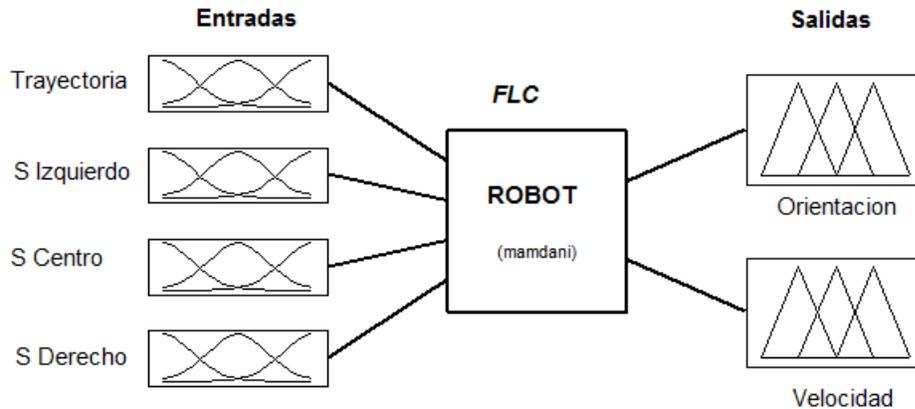
#### **2.4.1 Diseño del FLC en MATLAB.**

MATLAB es un entorno de desarrollo que permite realizar aplicaciones de lógica difusa mediante la utilización del toolbox *fuzzy logic*. El controlador diseñado tiene cuatro entradas, de las cuales tres son suministradas por los sensores de distancia Sharp y una entrada es el valor de la posición respecto a la trayectoria, las entradas del sistema difuso son procesadas en el controlador pasando por las etapas de fuzificación, mecanismo de inferencia y defuzificación, obteniendo como salida dos valores, orientación para alcanzar un objetivo y la velocidad de los actuadores. Se implementó un sistema difuso del tipo Mamdani, este es un método intuitivo y robusto, en las reglas de Mamdani los cálculos son considerablemente simplificados y permiten una interpretación gráfica simple<sup>66</sup>. El sistema difuso de tipo Mamdani utiliza como método de *and* el operador **mínimo**, para el proceso de implicación se utiliza el **mínimo**, en el proceso de agregación se utiliza el operador **máximo** y como método de *defuzificación* el **centro de área**. El controlador diseñado con sus entradas y salidas se muestra en la Figura 57.

---

<sup>66</sup> CHAHUARA QUISPE, José. Op. Cit., p12.

**Figura 57. Entradas y salidas del controlador difuso.**



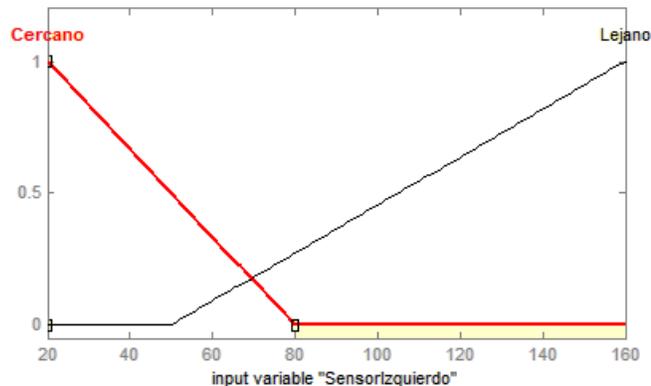
*Fuente: Esta investigación.*

### 2.4.2 Modelado de los conjuntos difusos.

*Fuzzy logic toolbox* mediante su editor gráfico (*FIS editor*) permite realizar este proceso de forma rápida e intuitiva, editando las funciones de pertenencia de forma gráfica. Así se diseñaron los conjuntos de entrada y salida del controlador.

- **Conjuntos difusos de entrada.** En el diseño cada una de las entradas referidas a la medición de los sensores de distancia se han denominado: Sensor izquierdo (SI), sensor centro (SC), sensor derecho (SD). Cada una de estas entradas se ha dividido en dos conjuntos difusos: *Objeto cercano*, *objeto lejano*, las cuales son modeladas con funciones de membrecía triangulares, el rango de definió entre 0 y 150 cm, que es el alcance máximo de los sensores de obstáculos. Las funciones de pertenencia son las mismas para los tres sensores. La Figura 58 muestra la entrada difusa SI y sus conjuntos.

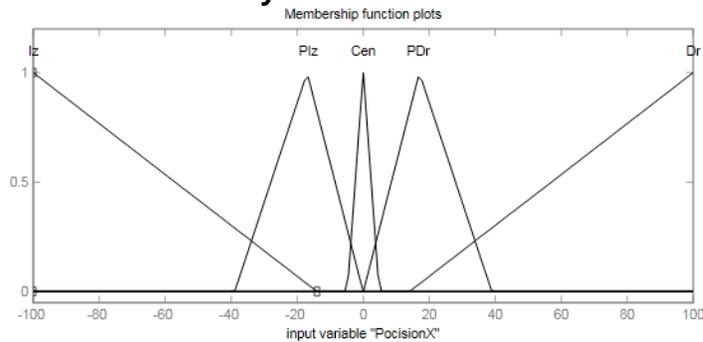
**Figura 58. Conjuntos difusos sensores.**



*Fuente: Esta investigación.*

La entrada trayectoria varía entre un rango de -100 cm y 100 cm, modelado mediante cinco conjuntos difusos de forma triangular: *izquierda*, *poco a la izquierda*, *centro*, *poco derecha*, *derecha*, como se muestra en la Figura 59. Con la incorporación de este número de conjuntos difusos, se logra tener una gran eficiencia del controlador, ya que se pueden tomar decisiones mucho más acertadas y obtener salidas más precisas para controlar la navegación del robot.

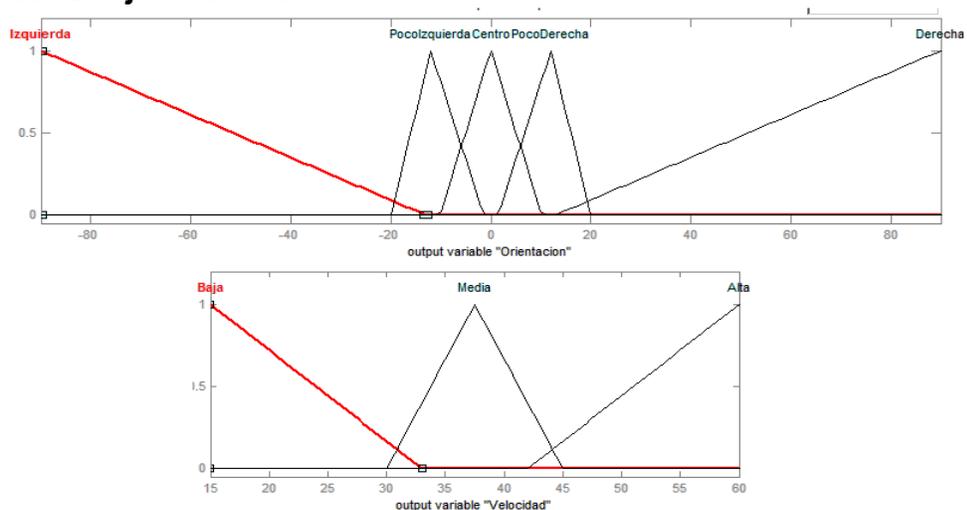
**Figura 59. Conjuntos entrada trayectoria.**



Fuente: Esta investigación.

• **Conjuntos de salida.** La salida velocidad fue modelada en un rango entre 0 y 60 RPM, que son la velocidad mínima y máxima a la que el robot puede desplazarse, se crearon tres conjuntos difusos para la variable velocidad denominados: *baja*, *media* y *alta*. La variable de salida orientación se modeló en un rango entre  $-90^\circ$  y  $90^\circ$  y se definieron cinco conjuntos difusos: *izquierda*, *poco izquierda*, *centro*, *poco derecha* y *derecha*, este rango garantiza cambios moderados en la orientación, sobre todo cuando se detecta un obstáculo. La Figura 60 muestra los conjuntos de salida velocidad y orientación.

**Figura 60. Conjuntos de salida.**



Fuente: Esta investigación.

• **Base de las reglas SI-ENTONCES.** Durante la navegación del robot móvil pueden ocurrir diferentes sucesos, en cada uno de estos se debe configurar apropiadamente al robot en velocidad y orientación, teniendo en cuenta lo anterior se diseñó la base de reglas. Como primer paso se obtuvieron comportamientos individuales, sobre cómo responder a una determinada situación, por ejemplo:

*Si posición es izquierda entonces orientarse a la derecha.*

*Si sensor derecho es lejano y sensor centro es lejano y sensor izquierdo es lejano, entonces velocidad del robot es alta.*

Se combinaron las entradas de los sensores y la trayectoria hacia la meta, en cada caso se evaluó cual sería la respuesta de un operador humano ante esas situaciones, determinando cuánto rotar para evadir un obstáculo y velocidad ante la presencia de éstos. Se debe tener en cuenta que para cada situación existirá una respuesta y que no deben quedar situaciones sin planear. Se definieron en total 40 reglas las cuales se especifican en la Tabla 5. Para implementar un sistema difuso con dos salidas, se trabaja como si se tratará de dos sistemas individuales de una salida los cuales trabajarán con las mismas condiciones en sus entradas<sup>67</sup>.

**Tabla 5. Reglas del controlador lógico difuso.**

Regla	SI				ENTONCES	
	Sensor izquierdo	Sensor centro	Sensor derecho	Trayectoria	Orientación	Velocidad
1	Cercano	Cercano	Cercano	Izquierda	Derecha	Baja
2	Cercano	Cercano	Cercano	Poco izquierda	Derecha	Baja
3	Cercano	Cercano	Cercano	Centro	Derecha	Baja
4	Cercano	Cercano	Cercano	Poco derecha	Izquierda	Baja
5	Cercano	Cercano	Cercano	Derecha	Izquierda	Baja
6	Cercano	Cercano	Lejano	Izquierda	Derecha	Baja
7	Cercano	Cercano	Lejano	Poco izquierda	Derecha	Baja
8	Cercano	Cercano	Lejano	Centro	Derecha	Baja
9	Cercano	Cercano	Lejano	Poco derecha	Derecha	Baja
10	Cercano	Cercano	Lejano	Derecha	Derecha	Baja
11	Cercano	Lejano	Cercano	Izquierda	Derecha	Baja
12	Cercano	Lejano	Cercano	Poco izquierda	Derecha	Baja
13	Cercano	Lejano	Cercano	Centro	Derecha	Baja
14	Cercano	Lejano	Cercano	Poco derecha	Izquierda	Baja
15	Cercano	Lejano	Cercano	Derecha	Izquierda	Baja
16	Cercano	Lejano	Lejano	Izquierda	Derecha	Media
17	Cercano	Lejano	Lejano	Poco izquierda	Poco derecha	Media
18	Cercano	Lejano	Lejano	Centro	Poco derecha	Media

<sup>67</sup> MÚJICA, Dante. Navegación autónoma de un robot móvil dentro de entornos real/virtual. Instituto politécnico nacional. Escuela superior de ingeniería mecánica y eléctrica. México D.F. 2008. p.38.

19	Cercano	Lejano	Lejano	Poco derecha	Poco derecha	Media
20	Cercano	Lejano	Lejano	Derecha	Poco derecha	Media
21	Lejano	Cercano	Cercano	Izquierda	Izquierda	Baja
22	Lejano	Cercano	Cercano	Poco izquierda	Izquierda	Baja
23	Lejano	Cercano	Cercano	Centro	Izquierda	Baja
24	Lejano	Cercano	Cercano	Poco derecha	Izquierda	Baja
25	Lejano	Cercano	Cercano	Derecha	Izquierda	Baja
26	Lejano	Cercano	Lejano	Izquierda	Derecha	Media
27	Lejano	Cercano	Lejano	Poco izquierda	Derecha	Media
28	Lejano	Cercano	Lejano	Centro	Poco derecha	Media
29	Lejano	Cercano	Lejano	Poco derecha	Izquierda	Media
30	Lejano	Cercano	Lejano	Derecha	Izquierda	Media
31	Lejano	Lejano	Cercano	Izquierda	Poco izquierda	Media
32	Lejano	Lejano	Cercano	Poco izquierda	Poco izquierda	Media
33	Lejano	Lejano	Cercano	Centro	Poco izquierda	Media
34	Lejano	Lejano	Cercano	Poco derecha	Poco izquierda	Media
35	Lejano	Lejano	Cercano	Derecha	Izquierda	Media
36	Lejano	Lejano	Lejano	Izquierda	Derecha	Alta
37	Lejano	Lejano	Lejano	Poco izquierda	Poco derecha	Alta
38	Lejano	Lejano	Lejano	Centro	Centro	Alta
39	Lejano	Lejano	Lejano	Poco derecha	Poco izquierda	Alta
40	Lejano	Lejano	Lejano	Derecha	Izquierda	Alta

El FIS editor de MATLAB se encarga de realizar todo el proceso de inferencia, como fuzificación, evaluación de las reglas, implicación, agregación y defuzificación obteniendo en las salidas un valor numérico para velocidad y orientación. La Figura 61 muestra el proceso de evaluación de las reglas en MATLAB.

### 2.4.3 Prueba de funcionamiento del controlador en MATLAB.

Se simuló en MATLAB el proceso de navegación del robot utilizando el controlador difuso diseñado, en éste se creó un entorno con obstáculos, se implementaron funciones para simular el alcance de los sensores y graficar el desplazamiento del robot, este programa brinda la posibilidad de agregar obstáculos en el ambiente y mirar el desempeño del controlador. Los resultados obtenidos se muestran en la Figura 62.

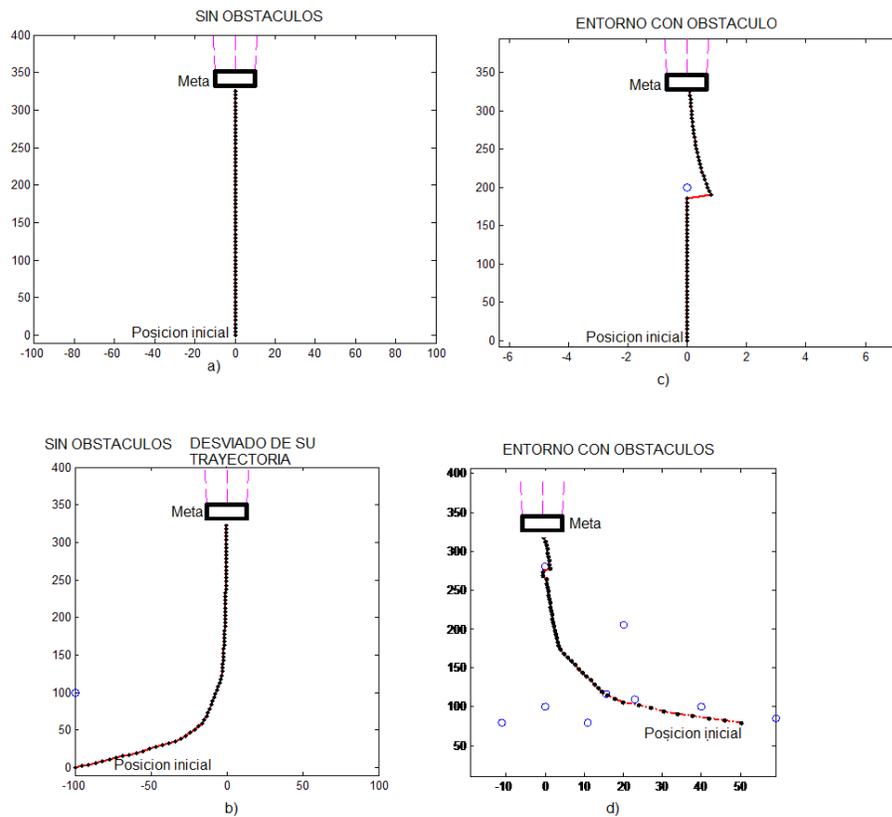
Con la simulación se pudo comprobar el buen funcionamiento el controlador, en la Figura 62 en a) se puede mirar que el robot llega a su objetivo sin realizar ninguna maniobra de evasión, en b) el robot llega a la meta desviado de su trayectoria en un ambiente sin obstáculos, en c) y d) se observa claramente la maniobra de evasión de obstáculos realizada por el robot móvil, la cual es ordenada por el controlador difuso.

**Figura 61. Evaluación de las reglas en MATLAB.**



*Fuente: Esta investigación.*

**Figura 62. Simulación del FLC en MATLAB.**



*Fuente: Esta investigación.*

#### 2.4.4 Implementación del controlador lógico difuso en lenguaje C.

El diseño del controlador difuso en MATLAB se llevó a cabo de una manera intuitiva y rápida, obteniendo buenos resultados en la simulación, ahora se debe implementar este sistema en el microcontrolador PIC18F4550. Se escogió el lenguaje C para desarrollar el controlador difuso ya que estos dispositivos pueden ser programados en este lenguaje, además de que la implementación de dicho sistema en código ensamblador sería muy extensa y tediosa, lenguaje C ofrece gran flexibilidad en la programación y cálculos aritméticos para estos dispositivos. Para implementar el controlador difuso se estudió el proceso de inferencia de Mamdani entendiendo su forma de operación, para posteriormente realizar su implementación de forma numérica en el PIC, buscando ante todo obtener una velocidad de procesamiento óptima en la toma de decisiones y conseguir una respuesta rápida. El controlador implementado tiene las mismas características que el diseñado en el editor FIS de MATLAB.

- Sistema difuso tipo Mamdani.
- Número de entradas: 4.
- Número de salidas: 2.
- Tipo de funciones de membresía: Triangular, trapezoidal.
- Número de reglas: 40.
- Método AND: Mínimo.
- Método implicación: Mínimo.
- Método de agregación: Máximo.
- Método de defuzificación: Centro de área.

Para el diseño del FLC se crearon.

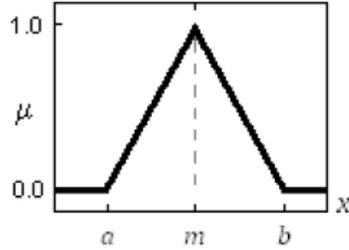
- Variables de entrada y salida.
- Funciones de pertenencia de entrada y salida.
- Fuzificador.
- Base de las reglas.
- Mecanismo de inferencia.
- Defuzificador.

Como primer paso se crearon las variables de entrada y salida necesarias, luego se definen las etiquetas lingüísticas de los conjuntos difusos con el mismo nombre de los realizados en MATLAB, por ejemplo: para la entrada *posición* se crean las siguientes etiquetas: *izquierda I*, *poco a la izquierda PI*, *centro C*, *poco derecha PD* y *derecha D*.

• **Funciones de pertenencia de entrada y salida.** El siguiente paso es elegir una forma para los conjuntos difusos, se utilizó las funciones trapezoidal y triangular por ser las que menos memoria y menor cálculo necesitan. Se creó una rutina que

permite implementar funciones de membrecía triangular, al definir tres puntos  $a$ ,  $m$ ,  $b$ . como se muestra en la Figura 63.

**Figura 63. Función de membrecía triangular.**



Fuente: Esta investigación.

En el caso de una función triangular el grado de pertenencia  $\mu(x)$  para una entrada  $x$  se define con la siguiente expresión:

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a, \\ \frac{x-a}{m-a} & \text{si } a < x < m \\ \frac{b-x}{b-m} & \text{si } m < x < b \\ 0 & \text{si } x \geq b \end{cases} \quad (44)$$

Se almacenan los parámetros  $a$ ,  $m$  y  $b$  que forman los triángulos de cada conjunto difuso en un arreglo o matriz. Siguiendo con el ejemplo de la variable de entrada Posición se guardan los valores mostrados en la Tabla 5:

**Tabla 6. Funciones triangulares para la entrada posición**

Variable de entrada	Conjunto difuso	Parámetros del triangulo
Posición	izquierda	Triangulo [-100 -100 -13.2]
	poco a la izquierda	Triangulo [ -39 -17 0]
	centro	Triangulo [ -5 0 5]
	poco derecha	Triangulo [ 0 17 39]
	derecha,	Triangulo [ 13.2 100 100]

En la matriz *Posición* se almacena los triángulos de la siguiente manera:

$$Posición = \begin{bmatrix} -100 & -100 & -13.2 \\ -39 & -17 & 0 \\ -5 & 0 & 5 \\ 0 & 17 & 39 \\ 13.2 & 100 & 100 \end{bmatrix} \quad (45)$$

El mismo procedimiento se realizó para cada una de las funciones de membrecía, de las variables de entrada y de salida. Con las funciones guardadas como arreglos de matrices se logra ahorrar memoria y se optimiza el código. Todos estos valores serán utilizados en el proceso de fuzificación.

- **Fuzificador:** Se creó una rutina que evalúa una entrada definida calculando el grado de pertenencia a cada uno de los conjuntos difusos, aquí se hace uso de los parámetros de las funciones triangulares guardadas anteriormente, por ejemplo en la Figura 64 se ilustra el proceso de fuzificación de la entrada *Posición*, para obtener el grado de pertenencia a cada uno de sus conjunto difusos, los cuales se guardan en un nuevo arreglo para ser utilizados en el proceso de inferencia. De igual forma se realiza la fuzificación para todas las entradas del sistema.

**Figura 64. Aplicación del fuzificador a una entrada.**



*Fuente: Esta investigación.*

- **Mecanismo de inferencia.** Se almacenan las reglas del FLC en un arreglo. Para la realización del proceso de inferencia o evaluación de las reglas se tienen en cuenta tres pasos: El primer paso es aplicar el operador AND, calculando el mínimo de los grados de pertenencia de los antecedentes de las reglas SI-ENTONCES, este valor será el grado de activación de cada regla. Por ejemplo para la regla uno se tiene:

R1=Si (sensor1 lejano = 0.7) y (sensor2 cercano = 0.5) y (sensor3 lejano = 0.6)

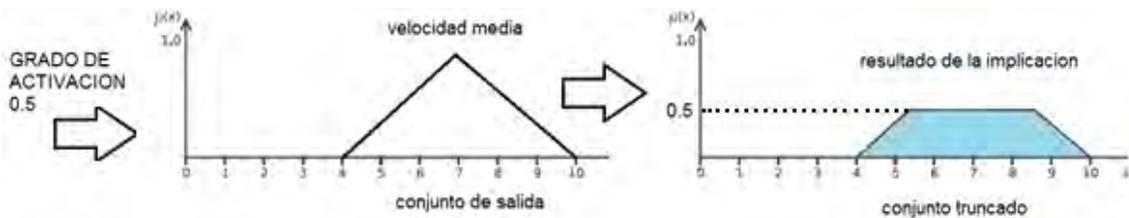
**ENTONCES...** R1=min (0.7, 0.5, 0.6)=0.5

Para que una regla se active, se necesita que todas las condiciones se cumplan en un grado de verdad diferente de cero, (operador and), si una condición no se cumple la regla se desactiva. De esta manera las cuatro entradas se evalúan en

cada una de las cuarenta reglas, se pueden activar varias reglas a la vez según sea el caso. El grado de activación de cada regla se guarda en un vector para su utilización en el proceso de implicación.

El segundo paso es realizar el proceso de implicación, para lo que se utiliza el mínimo (*min*). Para ello se calcula el valor mínimo entre el conjunto de salida triangular y el grado de activación de la regla calculado en el primer paso, esto modifica el conjunto de salida truncándolo en el grado de activación de la regla (ver Figura 65). El resultado de la evaluación de cada regla se guarda en una matriz.

**Figura 65. Proceso de implicación de una regla.**

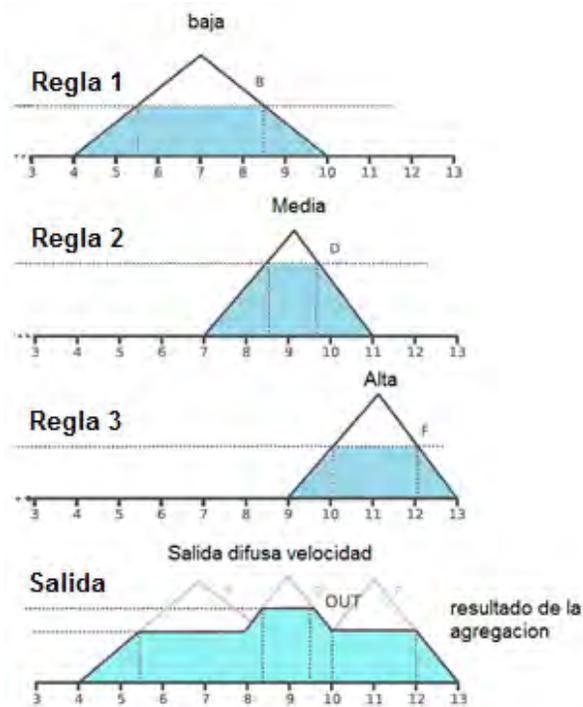


Fuente: Esta investigación.

El tercer paso consiste en unir los resultados de las reglas activadas mediante el proceso de agregación. Se utilizó el método del máximo (*max*), con lo que se unen todos los conjuntos truncados en un único conjunto de salida, para lograrlo se hace un barrido a lo largo del universo de discurso de salida sumando las contribuciones de cada regla activada. Por ejemplo para la variable velocidad, se suman los conjuntos de tres reglas que fueron activados, como se aprecia en la Figura 66.

Todo el mecanismo de inferencia descrito anteriormente, se realiza mediante el llamado de una función, ésta evalúa una a una todas las reglas mediante iteraciones de *ciclos for* anidados, con lo cual se logró optimizar el código al máximo y consumir los mínimos recursos de memoria en el microcontrolador.

Figura 66. Proceso de agregación de las reglas activadas.

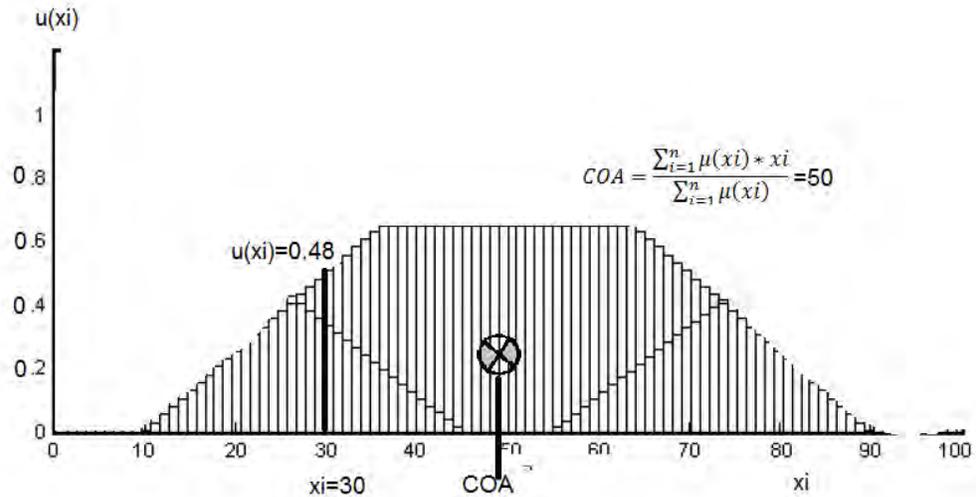


Fuente: Esta investigación.

• **Implementación del defuzificador.** Una vez obtenido el resultado de la agregación en un solo conjunto, se debe pasar el resultado difuso a un valor numérico válido para las salidas de velocidad y orientación, esto se realizó mediante el método del centro de área (COA). Para una función discreta se utiliza la ecuación (25).

Para encontrar el centro de área, se implementa un algoritmo que discretiza el conjunto de salida obtenido en el proceso de agregación, de igual forma esta rutina hace un barrido a lo largo del universo de discurso para conocer el numerador y el denominador de la ecuación (25). El numerador se calcula sumando el producto de cada punto evaluado con su grado de pertenencia respectivo, para el término del denominador se suman los grados de pertenencia del conjunto (ver Figura 67). Debido a que el controlador diseñado tiene dos salidas, se maneja el proceso de defuzificación por separado para la salida de orientación y velocidad.

**Figura 67. Defuzificación mediante COA.**



*Fuente: Esta investigación.*

En resumen, el proceso implementado en el microcontrolador se puede describir así: Se obtienen las entradas y se fuzifican, luego se evalúan en cada una de las cuarenta reglas, el resultado de cada regla se suma en solo conjunto de salida, se aplica la defuzificación, y se obtienen las salidas para controlar la navegación del robot, dependiendo de la posición que éste se encuentre y de su entorno. La Figura 68 ilustra este proceso.

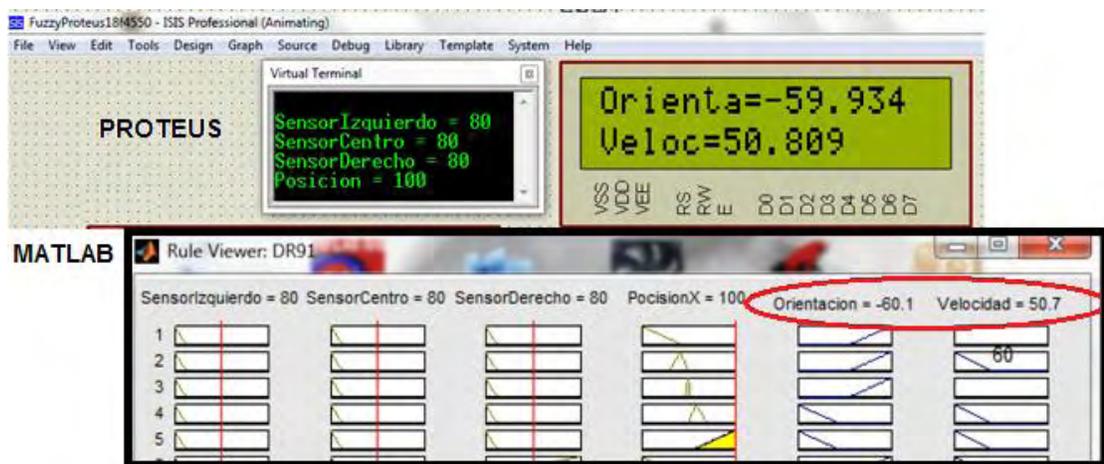
**Figura 68. Proceso realizado por el FLC.**



*Fuente: Esta investigación.*

- **Validación del controlador.** El código fue implementado en el compilador CCS y probado mediante el simulador PROTEUS, se configuró el PIC 18F4550 para que trabaje a su máxima velocidad (48MHz), verificando el correcto funcionamiento. Se obtuvo un tiempo de respuesta promedio de 300ms dependiendo de la cantidad de reglas activadas, posteriormente se compararon los resultados obtenidos con la respuesta del controlador implementado en MATLAB, obteniendo muy buenos resultados en el proceso con una variación solo en algunos decimales. La Figura 69 muestra la comparación de los resultados entre MATLAB y el simulador PROTEUS.

**Figura 69. Comparación respuesta FLC en MATLAB y PROTEUS.**



Fuente: Esta investigación.

## 2.5 SISTEMA DE NAVEGACIÓN FINAL.

Para el sistema de navegación se realizó la integración de todos los sistemas creados a lo largo del proyecto con el fin de facilitar la gestión de todos los elementos disponibles en el robot. Cada sistema se implementa mediante funciones y se guarda en librerías, luego se hace un llamado en el momento en el que se requiera su aplicación. Las funciones creadas fueron las siguientes:

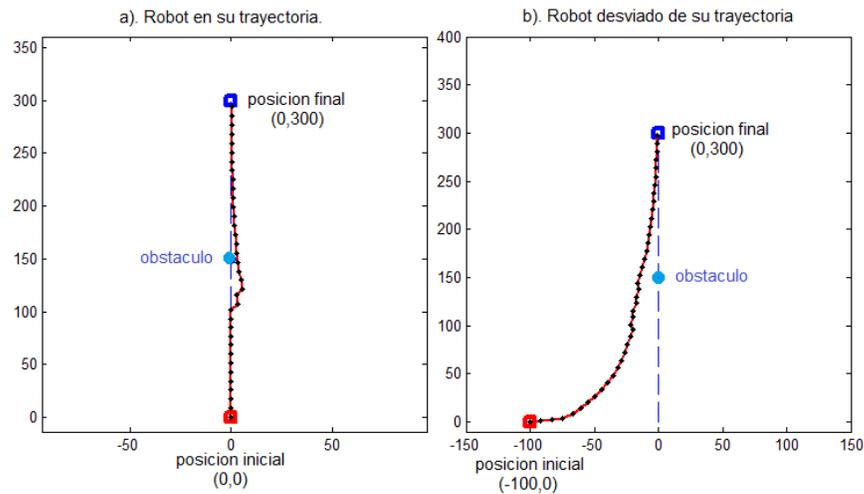
- **Sensores\_Obstaculos():** Esta función permite conocer la lectura de los sensores de obstáculos conectados al robot, al invocar esta función el programa devuelve la distancia en centímetros a la que se encuentra el objeto detectado.
- **Función\_Odometría():** Permite la medida del desplazamiento de las ruedas del robot mediante la información suministrada por los encoders, calcula la nueva posición y orientación del robot con respecto a un valor anterior.

- ***PID\_Motores()***: Esta instrucción permite tomar la medida de velocidad de las ruedas, aplicar el algoritmo de control PID y fijar la velocidad en un valor deseado.
- ***Sentido\_de\_Giro()***: Permite configurar el sentido de giro de los actuadores para lograr distintas formas de desplazamiento, por ejemplo: desplazarse hacia adelante, rotar sobre su propio eje, girar hacia la derecha o izquierda.
- ***Función\_Rotar\_Angulo()***: Con esta función se logra el robot rote sobre su eje hasta alcanzar un ángulo deseado.
- ***Función\_Desplazarse()***: Esta instrucción permite desplazarse una distancia determinada.
- ***Motor\_Stop()***: Permite detener el movimiento del robot en cualquier momento.
- ***Enviar\_Paquete()***: Sirve para enviar información desde el robot hacia la estación central, se transmiten datos de posición, orientación y mediadas de los sensores de obstáculos.
- ***Funcion\_Evaluar\_FIS()***: Esta función permite la utilización del controlador lógico difuso diseñado, evalúa en sus entradas la información recogida por los sensores de obstáculos y datos de posición, para devolver como salida la velocidad adecuada y la mejor orientación según sea el caso.

### **2.5.1 Algoritmo de navegación.**

El algoritmo de navegación permite al robot navegar desde una posición inicial ( $X_o$ ,  $Y_o$ ) hasta una final ( $X_f$ ,  $Y_f$ ), inicialmente se pensó en un sistema que resumiera el problema en dos pasos: Trazar una trayectoria paralela al eje Y en el punto  $X_f$ ,  $Y_f$  y alcanzar dicha trayectoria mediante la utilización del controlador difuso. En la simulación del controlador se observó que el controlador responde eficazmente cuando el robot se encuentra en línea con la trayectoria trazada por  $X_f$ ,  $Y_f$ , también si éste empieza desviado de su trayectoria, el controlador difuso le permite recuperarla. Esto se muestra en la Figura 70.

**Figura 70. Simulación algoritmo de navegación inicial.**



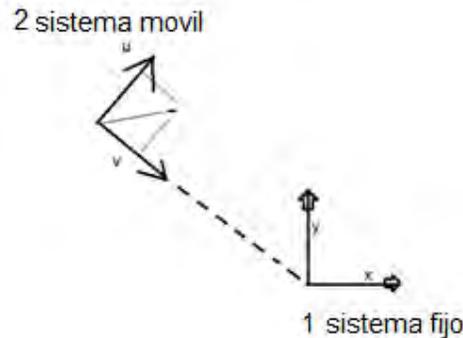
*Fuente: Esta investigación.*

El problema que se detectó es que la navegación se realiza mediante iteraciones, en éstas se busca siempre alcanzar el eje Xf de la trayectoria con el controlador difuso, mientras que la posición Yf es verificada en cada iteración mediante un bucle de control, lo que ocasionó que el robot no alcance la meta deseada, ya que éste detenía su movimiento cuando la posición Y final era alcanzada sin verificar si la posición en X final era correcta. Otro inconveniente sucede cuando la posición Y final es menor que la Y inicial, en este caso el robot tendría que girar y desplazarse en sentido negativo en el eje Y por lo que posiciones negativas no serían alcanzables. Por este motivo se propone realizar un algoritmo de navegación el cual posee dos sistemas de referencia: un sistema fijo (x y), asociado a las coordenadas absolutas del plano en el cual se localiza el robot y un sistema de referencia móvil (u v) asociado al robot (ver Figura 71).

Para relacionar el sistema fijo con el sistema móvil se utiliza una **matriz de transformación homogénea (MTH)**,<sup>68</sup> ésta permite la representación conjunta de la posición y orientación de los objetos, se puede observar en la Figura 71 que el sistema 2 se relaciona con sistema 1 mediante una traslación en el eje x, otra en y, terminando con una rotación en el eje z. La representación de la posición y la orientación mediante MTH es muy utilizada en robots manipuladores (brazos robóticos), en donde el problema es trabajado en tres dimensiones, en nuestro caso el problema de la traslación se reduce a dos dimensiones ya que el robot solo puede desplazarse en los ejes x, y.

<sup>68</sup> BARRIENTOS, Antonio. Fundamentos de robótica. Madrid, 1996, p.62.

**Figura 71. Sistemas de referencia asociados al robot.**



*Fuente: Esta investigación.*

El algoritmo de navegación se lo dividido en los siguientes pasos.

**Paso 1:** Generación de una trayectoria recta entre la posición inicial y la meta.

**Paso 2:** Rotar hasta quedar en línea con la trayectoria.

**Paso 3:** Crear dos sistemas de referencia, uno asociado a la posición inicial del robot y un segundo sistema asociado al origen de coordenadas del plano absoluto en el cual se trabaja.

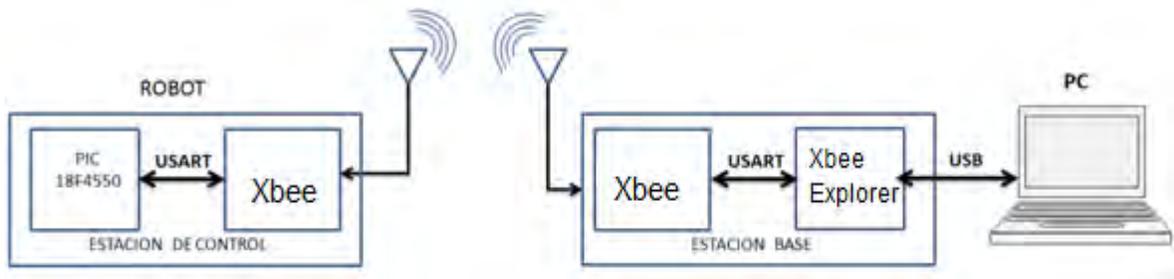
**Paso 4:** Crear una MTH que relacione los dos sistemas.

**Paso 5:** Desplazarse hacia la meta siguiendo la trayectoria creada, aplicando el controlador difuso en el sistema de referencia móvil y transformando mediante la MTH los valores de posición y orientación al sistema fijo, verificando en todo momento si la meta fue alcanzada.

### 2.5.2 Comunicación entre el robot y el PC.

El sistema de comunicación permite mantener informado al usuario de los cambios que suceden en el proceso de navegación del robot móvil, conociendo así obstáculos encontrados en el camino, datos de su posición y hacer posible el intercambio de información. Para ésto se utilizan los módulos XBee S2, la comunicación con la tarjeta de control se realiza a través del puerto serial del PIC, por su parte en la estación base la información es enviada al PC mediante la tarjeta XBee Explorer, la cual es una interfaz USB- SERIAL con la utilización de un VIRTUAL COM. El proceso de comunicación implementado se muestra en la Figura 72.

**Figura 72. Comunicación entre el robot y el PC.**

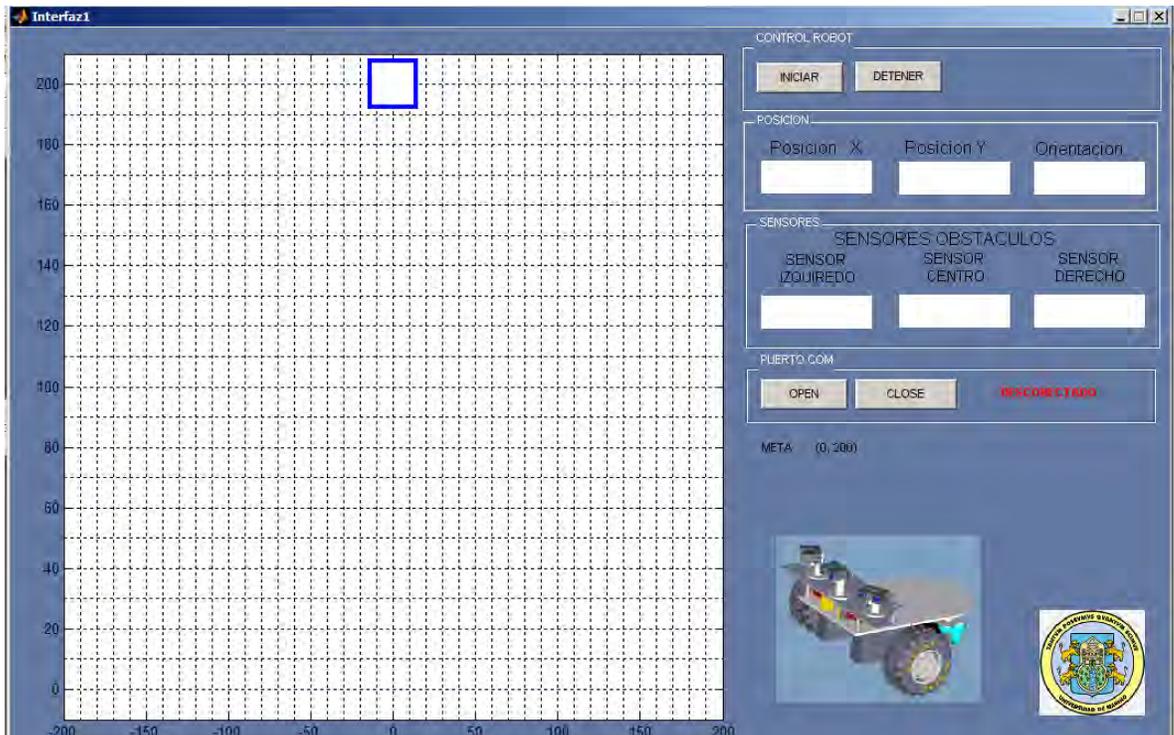


*Fuente: Esta investigación.*

### 2.5.3 Interfaz gráfica de usuario.

Se creó una interfaz gráfica de usuario GUI en MATLAB, con ésta se logró monitorear el comportamiento del robot, además se puede mirar la información proveniente de los sensores de obstáculos, posición y orientación en la que se encuentra el móvil y la gráfica de la trayectoria recorrida. Otra función es comunicarse con el nodo móvil vía RF mediante el puerto serial virtual. La interfaz gráfica diseñada se muestra en la Figura 73.

**Figura 73. Interfaz gráfica de usuario.**



*Fuente: Esta investigación.*

## 2.6 SISTEMA DE LOCALIZACIÓN BASADO EN RADIOFRECUENCIA.

Esta parte del trabajo trata sobre la creación de una red inalámbrica de sensores cuya función principal es el auto-posicionamiento del robot móvil con respecto a nodos baliza de posiciones conocidas, haciendo uso de la medida de potencia de las señales de radiofrecuencia recibidas entre los mismos. El sistema de localización por excelencia más conocido y extendido actualmente en casi todo el mundo es el GPS, éste presenta una solución “casi universal” para obtener localización precisa y rápida en cualquier punto del planeta, pero lamentablemente existen determinados entornos exteriores, y en la mayoría de interiores como hogares, hospitales, universidades, etc, donde el GPS no es operativo, sin embargo, en dichos lugares transcurre gran parte de la actividad humana.

Si se desea realizar un sistema de posicionamiento absoluto para el robot, se debe pensar en una solución que sea confiable y a la vez económica por los costos del proyecto. Para desarrollar el posicionamiento por RF se escogió trabajar con la tecnología Zigbee y se utilizó los módulos XBee *serie2*.

### **2.6.1 Módulos XBee de Digi.**

Los módulos XBee son dispositivos que integran un transmisor-receptor de radiofrecuencia, utilizan la tecnología Zigbee como protocolo para la transmisión de información, además contienen un procesador en el mismo módulo, lo que permite a los usuarios desarrollar aplicaciones adicionales de manera rápida y sencilla como conexión de sensores (usando su conversor ADC), muestreos y toma de datos, salidas PWM, entradas y salidas digitales, entre otras. En la actualidad existen dos series de estos módulos: La serie1 y la serie2. La presentación del encapsulado de las dos series es la misma, además en las dos series existe la denominación "PRO", la cual hace referencia a un mayor alcance y cobertura de la señal<sup>69</sup>.

- **Configuración del XBee:** La configuración de los módulos XBee, se puede realizar mediante el envío de comandos AT por algún programa que utilice conexión a puerto serial, por ejemplo hiperterminal de Windows, conectado al modulo UART del XBee. Sin embargo el fabricante provee y recomienda el software XCTU con el cual se pueden realizar una configuración de manera rápida y fácil, además de que permite cambiar parámetros y actualizar el firmware del módulo.

### **2.6.2 Modo de operación.**

Entre los diferentes modos de operación se pueden destacar dos diferentes:

---

<sup>69</sup> CHAMORRO, Luis. Diseño e implementación de una red de sensores inalámbricos para el control de riegos por goteo. Universidad de Nariño. Facultad de ingeniería. Pasto.2012, p.48.

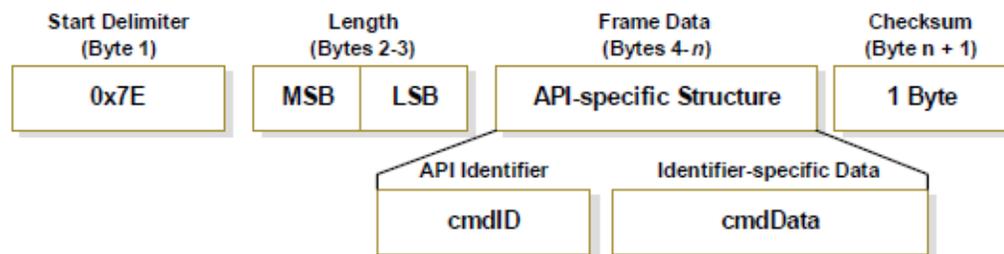
• **Modo transparente:** Es la configuración por defecto y es la más sencilla de realizar, en este modo todos los datos recibidos a través del pin 3 (data in), son guardados en el buffer de entrada y transmitidos, lo que ingresa por la antena como paquete RF, es guardado en el buffer de salida y luego enviado por el pin 2 (data out). Es ideal cuando se desea reemplazar alguna conexión serial por cable, se llama modo transparente porque la transmisión y empaquetado de los datos es transparente para el usuario y en la salida el módulo no agrega información adicional al dato transmitido<sup>70</sup>.

• **Modo API (Application Programming Interface):** Este modo es más complejo, pero permite el uso de tramas con cabeceras que aseguran la entrega de los datos al estilo TCP, extiende el nivel en el cual la aplicación del cliente puede interactuar con las capacidades de red de los módulos<sup>71</sup>. Entre las opciones que permite el modo API, se tienen:

- Transmitir información a múltiples destinatarios sin entrar al modo de comandos.
- Recibir estado de éxito/falla de cada paquete RF transmitido.
- Identificar la dirección de origen de cada paquete recibido.
- Enviar comandos AT para configuración de módulos remotos.
- Permite obtener los datos de las entradas analógicas y digitales de un módulo remoto, lo cual no es posible realizar en modo transparente.

Cuando el XBEE se encuentra en modo API, toda la información que entra y sale, es empaquetada en tramas o frames, que definen operaciones y eventos dentro del módulo. La Figura 74 muestra la estructura general de una trama API.

**Figura 74. Estructura general de una trama API.**



*Fuente: DIGI, International Inc. API Operation. En XBee/XBee PRO ZB RF Modules. 2012, p.35.*

Existen diferentes tipos dependiendo de la función que se realice, pero todas se basan en la misma estructura. La función de la trama API se especifica siempre en

<sup>70</sup> DIGI. Guía de usuario. XBee series 1. Junio. 2010, p.19.

<sup>71</sup> Ibid., p.20.

el cuarto byte el cual es el identificador API (API ID)<sup>72</sup>. La Tabla 6 muestra el valor del identificador API de acuerdo al contenido de la trama.

**Tabla 7. Identificadores de tramas API más comunes.**

API Frames Names	API ID
AT command	0x08
AT command-queue parameter value	0x09
Zigbee transmit request	0x10
Explicit addressing Zigbee command frame	0x11
Remote command request	0x17
AT command response	0x88
Modem Status	0x8A
Zigbee receive packet	0x90
Remote command response	0x97

### 2.6.3 Desarrollo del sistema de localización basado en RSSI.

El sistema de posicionamiento diseñado, utiliza la medida del nivel de potencia recibido (RSSI) y con el procesamiento de esta información se infiere la distancia entre los módulos, para posteriormente aplicar la técnica de triangulación y encontrar la posición del robot.

• **Estructura del sistema de posicionamiento RF.** El sistema de posicionamiento consta de las siguientes partes:

Tres nodos fijos con posiciones conocidas que actúan como balizas emisoras.

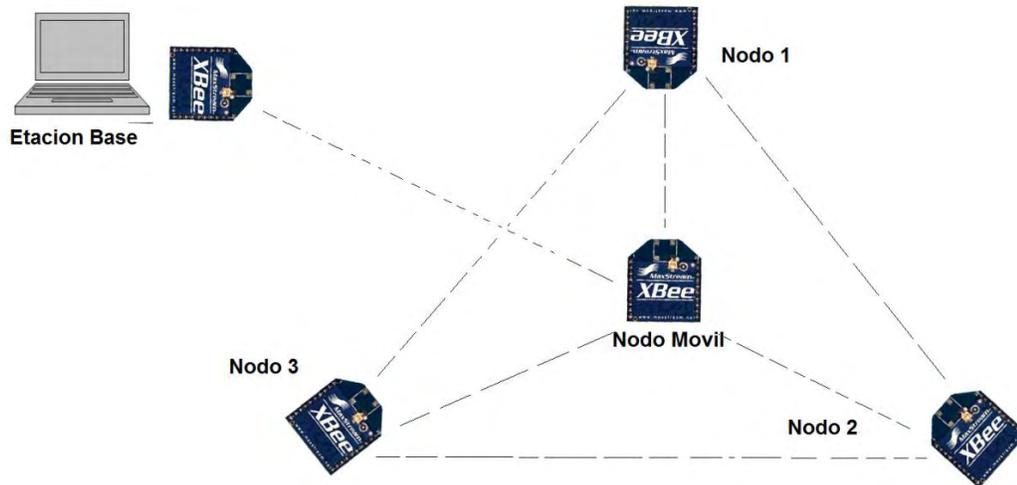
Un nodo móvil (robot) al cual se pretende localizar en el espacio delimitado por las balizas.

Un nodo que hace las veces de interfaz con el PC y permite enviar y recibir información.

Los tres nodos fijos están compuestos por un módulo XBee y su respectiva alimentación mediante baterías de 3 V, el nodo móvil esta compuesto por el modulo XBee y el microcontrolador de la tarjeta de control, en el cual se implementó el algoritmo de posicionamiento. El nodo interfaz-PC, esta formado por un modulo XBee y la tarjeta XBee Explorer que permite la comunicación con el PC en donde se monitorea el comportamiento de la red y del robot. La estructura general del sistema de posicionamiento se muestra en la Figura 75.

<sup>72</sup> DIGI, International Inc. API Operation. En XBee/XBee PRO ZB RF Modules.2012.

**Figura 75. Estructura general del sistema de posicionamiento.**



*Fuente: Esta investigación.*

#### **2.6.4 Propagación de la onda RF.**

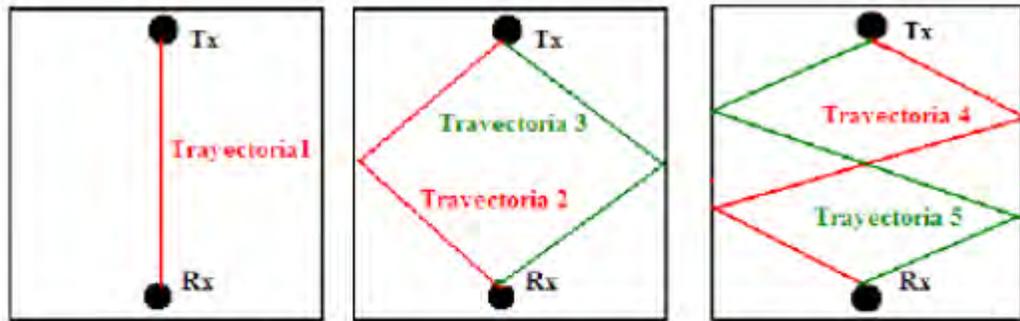
Se llama propagación al conjunto de fenómenos físicos que conducen a las ondas del transmisor al receptor. Cuando una onda de radio choca con un obstáculo, parte de ella se refleja y experimenta una cierta pérdida de la intensidad, este fenómeno se denomina reflexión y hace que el ángulo de la onda incidente sea equivalente al ángulo de la onda reflejada<sup>73</sup>.

Por definición, una onda de radio es susceptible de propagarse en varias direcciones. Después de reflejarse varias veces, una señal puede llegar a una estación o punto de acceso luego de tomar muchas rutas diferentes llamadas multitrayecto<sup>74</sup>. En el caso de los escenarios en interiores, la situación general es tener 9 trayectorias de propagación significativas. La Figura 76 muestra este fenómeno.

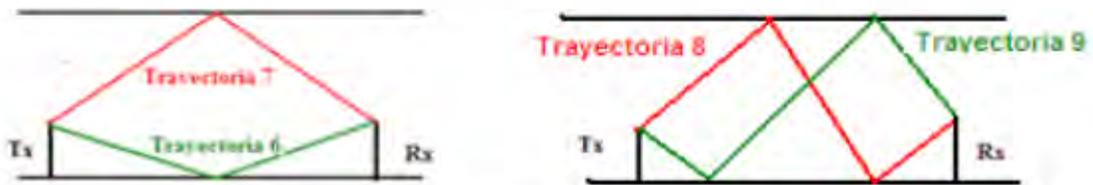
<sup>73</sup> MARTIN, José. Op. cit., p.29.

<sup>74</sup> Ibid., p.30.

**Figura 76. Multitrayecto de la onda RF en interiores.**



**Trayectorias de propagación en un escenario interior visto de planta**



**Trayectorias de propagación en un escenario**

*Fuente: MARTIN, José. Posicionamiento por radiofrecuencia basado en robots colaborativos. Universidad autónoma de Madrid. España. 2012. p.30.*

En la figura anterior, la trayectoria 1 es la que mayor intensidad de señal tiene ya que no se refleja en ningún sitio. Si se tuviera en cuenta esta trayectoria, los resultados obtenidos serían similares a la propagación en espacio libre, pero en realidad el aporte de las otras trayectorias que sufren reflexión en algún punto de los planos superior, inferior y lateral, provoca que el valor de la potencia recibida sufra constantes cambios<sup>75</sup>.

La variación de la potencia recibida en función de la distancia depende del entorno en el que se encuentran el transmisor y el receptor. El ejemplo más simple es cuando la transmisión se realiza en el espacio libre, es decir en un caso ideal en el que no encontramos ningún tipo de obstáculos. No obstante, en la práctica suele encontrarse entornos que contienen objetos que reflejan, absorben u obstruyen la señal electromagnética, modificando de esta manera la potencia recibida respecto la potencia que se recibiría en el caso ideal.

La potencia recibida no es solamente función de la distancia, ya que la propagación radioeléctrica especialmente en el interior de edificios, sigue unos patrones difíciles de predecir y provoca variaciones aleatorias de la señal. La potencia recibida varía también en función de la frecuencia utilizada, y por tanto en

<sup>75</sup> Ibid., p.49.

función del canal utilizado. Realmente la potencia recibida será la suma de una serie de señales que llegan de distintas direcciones, debido a las reflexiones con objetos cercanos y objetos que bloquean parcialmente la señal. De esta manera, la potencia recibida resultante puede ser mayor o menor que la potencia medida en espacio libre. Además, si los objetos reflectantes se mueven respecto a los receptores, la potencia recibida variará con el tiempo para una misma distancia<sup>76</sup>.

• **Modelo de propagación para interiores.** Para inferir la distancia entre las antenas de los módulos XBee se utiliza los modelos de propagación de RF en interiores. Un modelo de propagación es una formulación empírico matemática para la caracterización de la propagación de ondas de radio en un ambiente dado. Los modelos de propagación predicen la pérdida por trayectoria que una señal RF puede tener entre una estación base y un receptor, sea móvil o fijo<sup>77</sup>. Un enlace de radio se puede expresar de la forma.

$$P_{rx} = P_{tx} + G_{tx} + G_{rx} + L_p + L_{otras} \quad (46)$$

Donde  $P_{rx}$  es la potencia recibida,  $P_{tx}$  es la potencia transmitida,  $G_{tx}$  es la ganancia de la antena transmisora,  $G_{rx}$  es la ganancia de la antena receptora,  $L_p$  son las pérdidas de potencia de la señal debida a la propagación y  $L_{otras}$  son pérdidas de otra naturaleza. Por otro lado, las pérdidas debido a la propagación en interiores con base en el modelo de pendiente única, vienen definidas en la siguiente relación:

$$L_p[dB] = L_0 + 10 \cdot n \cdot \log_{10}\left(\frac{d}{d_0}\right) \quad (47)$$

Teniendo en cuenta que el módulo de RF obtiene la potencia recibida  $P_{rx}$  de la señal y se quiere obtener la distancia  $d$ , se puede reorganizar la expresión para obtener la ecuación (48), Siendo  $Cnt$  una constante que involucra los valores de potencias y ganancias de las antenas emisoras y receptoras<sup>78</sup>. De este modo se consigue la relación teórica entre  $P_{rx}$  (RSSI) y distancia, que se ha de buscar empíricamente con los módulos de RF disponibles.

$$P_{rx}[RSSI] = Cnt - 10 \cdot n \cdot \log_{10}\left(\frac{d}{d_0}\right) \quad (48)$$

Si siguiendo este modelo, cada escenario en el que realice la propagación tendrá una constante  $Cnt$  y una pendiente  $n$  distinta, dependiendo de las características

<sup>76</sup>ROCA, Daniel. Posicionamiento con una red de sensores inalámbricos (WSN): implementación para una aplicación real, Universidad Autónoma de Barcelona (UAB), Departamento de Telecomunicaciones e ingeniería de sistemas. España, 2009, p35.

<sup>77</sup> MARTIN, José. Op. cit., p.31

<sup>78</sup> Lbid., p.33

propias del entorno; como materiales de construcción, hierro, concreto, vidrio, elementos metálicos presentes, como sillas y otros objetos.

La relación entre distancia  $d$  y potencia  $RSSI$  está dada por la expresión:

$$d = 10^{\frac{cnt-RSSI}{10*n}} \quad (49)$$

### 2.6.5 Pasos para el desarrollo del sistema.

El sistema de posicionamiento se lo dividió en dos etapas:

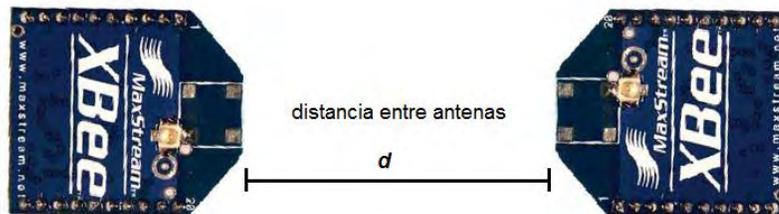
Obtención y calibración del modelo de propagación.

Creación de la red Zigbee para el posicionamiento.

#### • Obtención y calibración del modelo de propagación.

Para hallar la relación entre la potencia recibida  $RSSI$  y la distancia entre módulos XBee, se debe obtener el modelo de propagación. Para ello se colocaron los dos módulos frente a frente como muestra en la Figura 77, se tomó la medida de potencia entre los módulos y se fue incrementando la distancia de separación cada 5cm, tomando en cada caso el valor de potencia. El valor  $RSSI$  en los módulos XBee varía entre -26dBm y -90dBm, siendo el primero el valor máximo de potencia recibida cuando los dos módulos se encuentran lo más cercano posible.

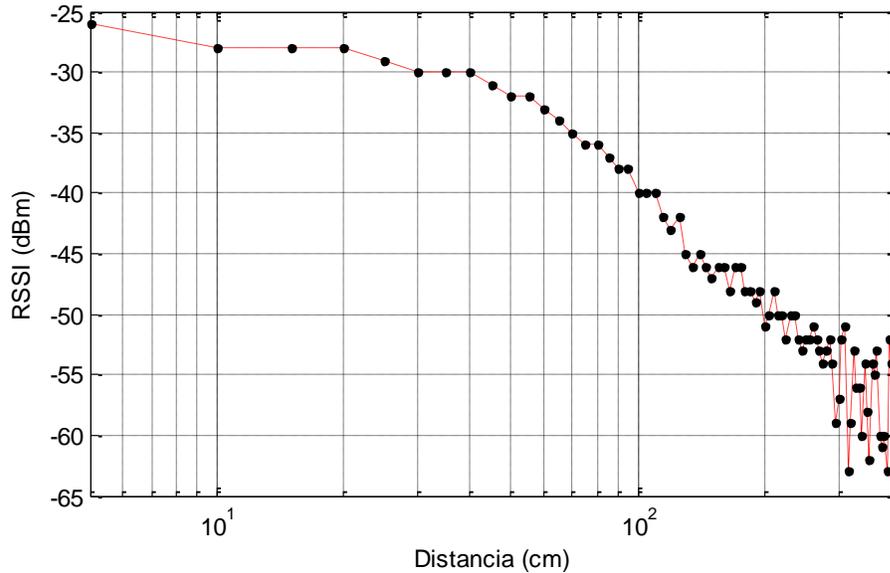
**Figura 77. Posición de los módulos XBee en las medidas.**



*Fuente: Esta investigación.*

Con la ayuda de MATLAB se grafica en escala logarítmica la relación de la potencia recibida respecto a la distancia, logrando apreciar mejor la forma de propagación de la onda RF. La Figura 78 muestra el modelo de dispersión de la potencia en relación a la distancia entre antenas.

**Figura 78. Dispersión encontrada RSSI Vs distancia.**



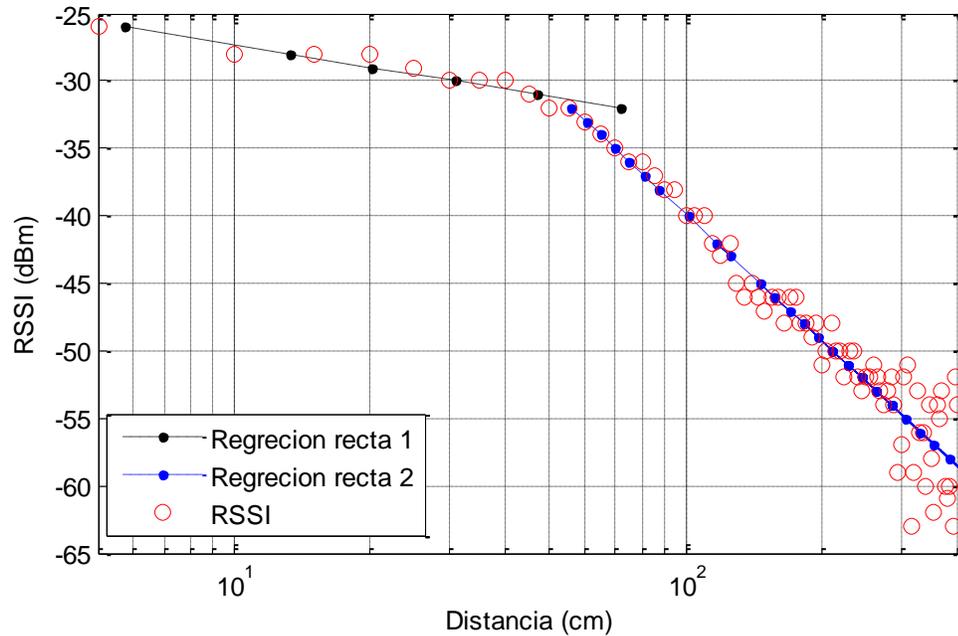
*Fuente: Esta investigación.*

Al mirar el comportamiento de la potencia cuando se aumentó la distancia entre módulos, se eligió aplicar un modelo de propagación de doble pendiente, ya que la curva presenta un punto de inflexión en donde la pendiente se vuelve más pronunciada a partir de una distancia específica, observando que después de este valor la pendiente es mucho mayor debido a que la potencia decrece más rápidamente. En la práctica el punto de cambio fue a partir de 100 cm. La ecuación (50) muestra el modelo de pendiente doble aplicado.

$$P_{rx}[dBm] = \begin{cases} Cnt1 - 10 \cdot n_1 \cdot \log_{10}(d) & d \leq 1m \\ Cnt2 - 10 \cdot n_2 \cdot \log_{10}(d) & d > 1m \end{cases} \quad (50)$$

Con la ayuda de MATLAB se realizaron dos regresiones logarítmicas para aproximar el modelo a uno de doble pendiente, obteniendo así las constantes necesarias y las pendientes de las ecuaciones. Esto se aprecia en la Figura 79.

**Figura 79. Regresión logarítmica al modelo encontrado**



*Fuente: Esta investigación.*

El modelo de propagación para el lugar donde se realizó el muestreo y que relaciona la potencia RSSI respecto a la distancia quedaría de la siguiente manera:

$$P_{rx}[dBm] = \begin{cases} -5.4660 - 10.(-0.5466). \log_{10}(d) & d \leq 100 \text{ cm} \\ -31.1310 - 10.(-3.1131). \log_{10}(d) & d > 100 \text{ cm} \end{cases} \quad (51)$$

Con el modelo encontrado y conociendo la potencia de la señal se puede inferir la distancia entre módulos. Una vez calibrado el modelo de propagación para el lugar específico en donde se realizó la prueba, se procede a diseñar el algoritmo de posicionamiento en el microcontrolador.

• **Creación de la red Zigbee para el posicionamiento.** Después de analizar la mejor topología de red se decidió utilizar una red tipo estrella, en donde el coordinador de red estará ubicado en el nodo móvil que irá en el robot y los cuatro nodos fijos se configuran como routers. Se trabajó esta topología debido a que se busca posicionar el nodo móvil que es el encargado de gestionar la red. Los módulos fueron configurados para trabajar en modo API, ya que es en este modo de operación donde se puede aprovechar todo el potencial de los Xbee-S2 y en donde se puede adquirir el valor RSSI de los distintos nodos baliza, además solo en el modo API es posible trabajar con las direcciones de red de los dispositivos para realizar la comunicación con todos los módulos presentes. En este punto, se

crearon las librerías necesarias para el trabajar con las tramas API que se utilizaron para configurar y comunicarse con los XBee.

Para trabajar los Xbee-S2 en Modo API, se implementó en lenguaje C librerías que permitan la creación de tramas específicas, que tengan el formato correcto y calculen automáticamente parámetros propios de cada trama como el largo de trama, el API ID y el checksum. Igualmente se realizaron funciones para la recepción de Tramas API sobre todo para garantizar que la trama recibida sea correcta y extraer la información útil de las mismas en este caso el valor de la potencia recibida y la dirección del módulo quien envía el paquete.

### **2.6.6 Desarrollo del algoritmo de posicionamiento.**

El algoritmo de posicionamiento diseñado esta dividido en tres pasos:

#### **Paso1. Obtener indicador RSSI del nodo móvil a cada uno de los nodos fijos.**

Para obtener el valor RSSI de cada uno de los XBee, se procede a enviar un mensaje desde el coordinador hacia el módulo al cual se desea medir la potencia de recepción. El proceso es el mismo para cada uno de los nodos baliza. Cuando se trabaja en modo API, se puede trabajar con las respuestas que generan los módulos confirmando cuándo se ha recibido un paquete (respuesta ASK), lo cual no es posible de realizar en modo transparente. En el momento en el que el módulo recibe una respuesta o un paquete cualquiera, éste mide el nivel de intensidad con el cual se recibe el mensaje y lo guarda en un registro RSSI. Esta sería una opción para los módulos XBee-s2, en los cuales la intensidad de potencia no viene incluida en los paquetes, contrario a los XBee-s1 donde en sus paquetes ya esta incluido el valor de RSSI.

El comando "ATDB" puede ser enviado en las tramas API ***AT\_comand*** para obtener el valor RSSI en módulos locales y ***AT\_comand\_request*** para módulos remotos. Si se desea más información sobre la estructura de dichas tramas se puede encontrar en la hoja de características de los módulos XBee suministrada por el fabricante<sup>79</sup>. Mediante el manejo de estas tramas API y comandos AT se mide la potencia recibida cuando el coordinador recibe un paquete o respuesta. De igual forma se envían tramas a los módulos remotos preguntando el nivel RSSI recibido por éstos.

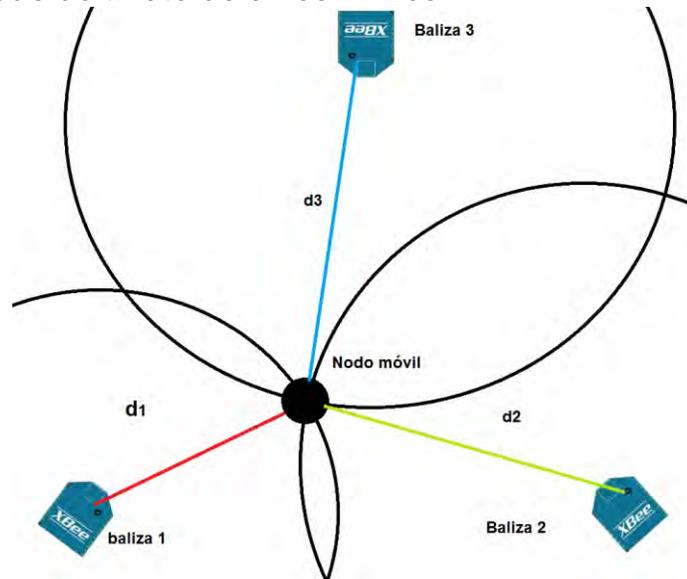
**Paso2. Asociación de cada potencia a una distancia.** Mediante el modelo de propagación encontrado, que relaciona la intensidad de potencia con la distancia entre antenas, se estima la distancia de separación que existe entre el nodo móvil y cada uno de los nodos fijos. Se necesita como mínimo obtener tres distancias para poder estimar la posición de un objeto en dos dimensiones.

---

<sup>79</sup> DIGI, International INC. XBee Series 2 OEM RF Modules. Product Manual. En XBee/XBee PRO ZB RF Modules.2012.

**Paso3. Aplicar la técnica de triangulación:** para estimar la posición  $(X_e, Y_e)$  en la cual se encuentra el nodo móvil, se aplica la técnica de trilateración conociendo la posición de cada nodo emisor, la distancia del nodo móvil a cada nodo fijo, se traza una circunferencia con centro en la posición conocida de cada módulo, con lo que se obtiene un sistema de tres ecuaciones. Solucionando el sistema se puede encontrar la posición en la que se encuentra el nodo móvil y en consecuencia el robot (ver Figura 80).

**Figura 80. Método de trilateración con XBee.**

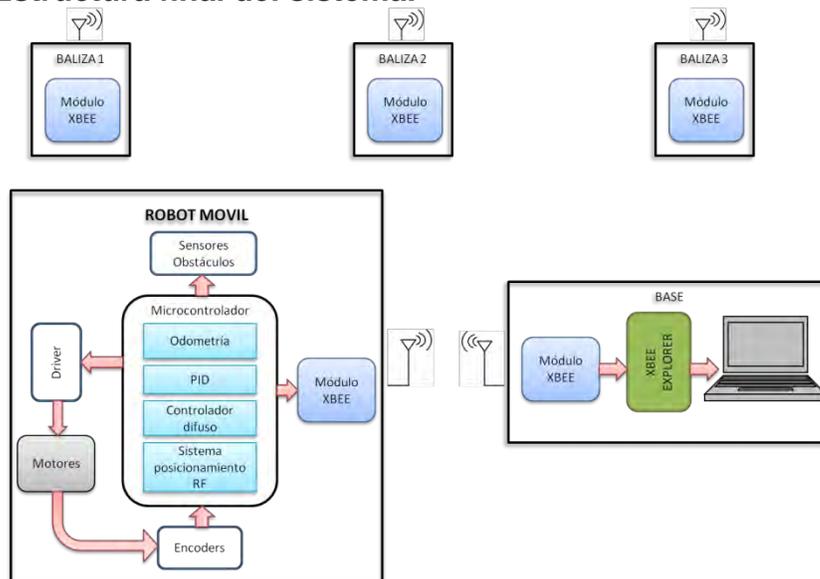


*Fuente: Esta investigación.*

## 2.7 ESTRUCTURA FINAL DEL SISTEMA.

El sistema final se conforma por cinco bloques: el robot móvil, la estación base-PC y las tres estaciones baliza. El robot móvil conformado por los sensores, los motores, el módulo XBee y la tarjeta de control que la información a la estación base-PC en donde se visualiza y monitorea el comportamiento del sistema. Además se tienen las tres balizas XBee cuando se desea utilizar un sistema de posicionamiento mediante RSSI. La Figura 81 muestra la estructura del sistema con los elementos que lo conforman.

**Figura 81. Estructura final del sistema.**

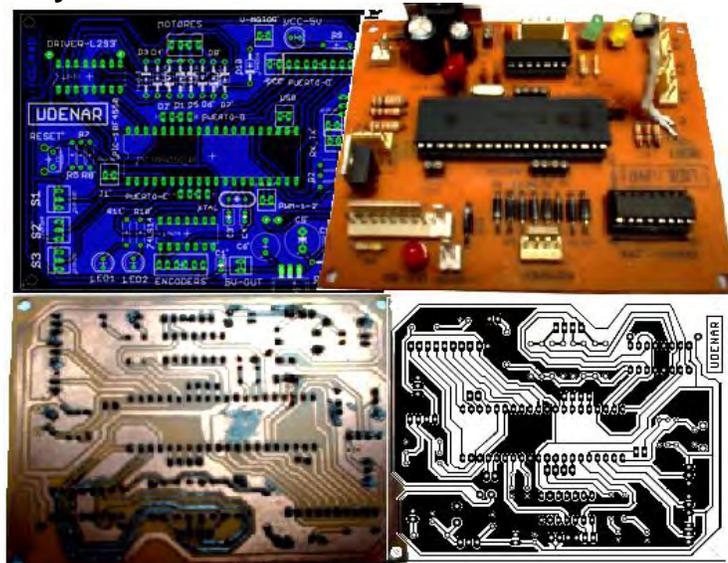


*Fuente: Esta investigación.*

### 2.7.1 Diseño y construcción de circuitos impresos.

El diseño de la tarjeta de control y de las estaciones balizas XBee utilizadas para el proyecto se realizó en el software EAGLE, en el diseño final de la tarjeta de control, se tuvo en cuenta que el robot puede ser utilizado como base para futuras investigaciones, por lo que los puertos libres del microcontrolador también están disponibles en conectores, la placa también posee un conector para voltaje 3V y 5V por si se requiere su utilización. La Figura 82 muestra el proceso de diseño y fabricación de las placas.

**Figura 82. Diseño y construcción del PCB.**

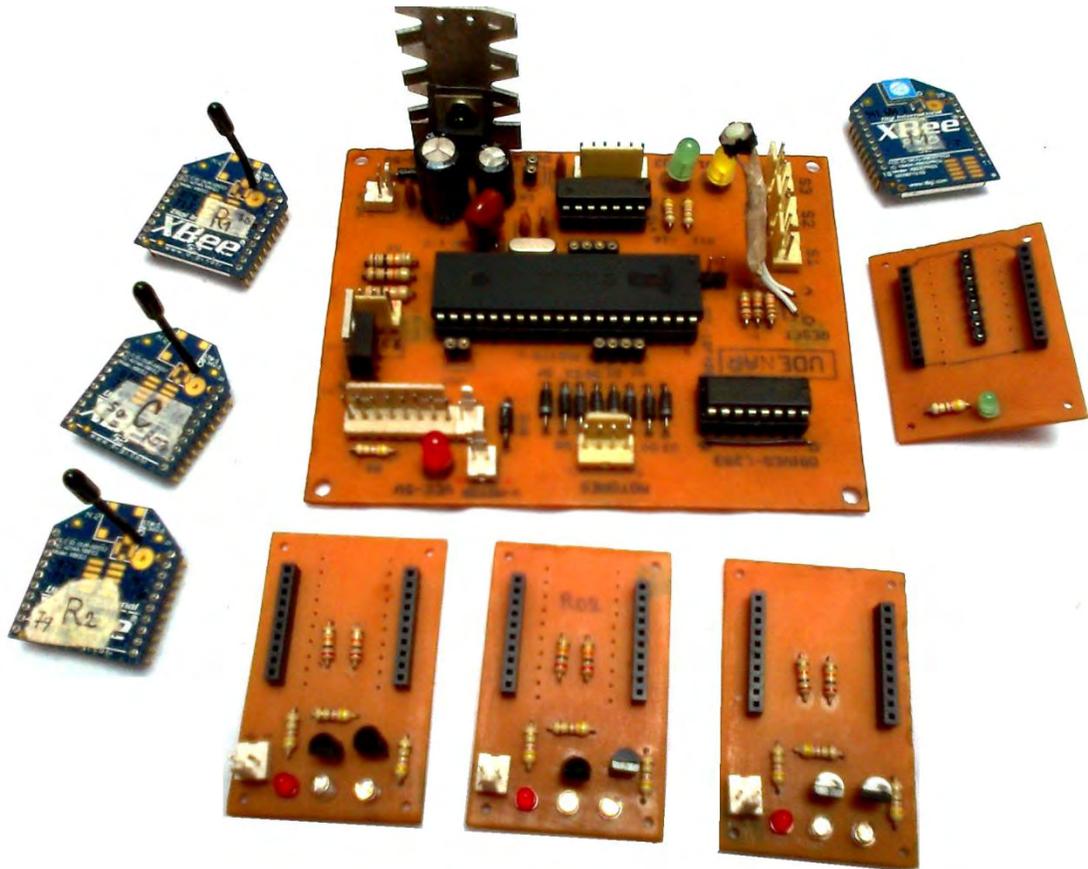


*Fuente: Esta investigación.*

De igual forma se realizaron las tarjetas XBee para el sistema de posicionamiento, con el fin de obtener información del comportamiento de la red, en estas se colocaron leds indicadores de asociación, recepción de paquetes y alimentación, éstos permiten conocer si se ha formado la red y cuándo se ha recibido un paquete.

Se aplicó la técnica del planchado para traspasar el diseño a la placa de cobre o baquelita, posteriormente se realizaron los agujeros y soldadura de componentes. Pensando en un diseño flexible a la hora de las conexiones, los sensores de distancia poseen conectores mólex individuales para una fácil conexión o cambio. De igual forma, este tipo de conectores se utilizan en la entrada de los encoders, salida de los motores y el modulo XBee. La Figura 83 muestra la placa final del robot y de las estaciones XBee terminadas.

**Figura 83. Tarjeta de control y baliza XBee terminadas.**



*Fuente: Esta investigación.*

### 2.7.2 Baterías utilizadas.

Para energizar la tarjeta de control del robot, se utilizó una batería de 9V de ion de litio recargable, ésta alimenta los componentes electrónicos PIC, driver, sensores, encoder y modulo XBee. La alimentación de los motores se realizó mediante una batería recargable de ion de litio de 9 voltios de la marca Marshall. Se realizó esta elección ya que los motores son elementos que consumen mucha corriente lo que ocasiona una descarga rápida de la batería, las baterías de ion de litio tienen mayor durabilidad. Para energizar cada una de las estaciones baliza del sistema de posicionamiento RF del robot, se utilizan un par de baterías de 1.5V AA, debido a que los XBee presentan un bajo consumo de corriente brindando larga durabilidad de las baterías utilizadas. La Figura 84 muestra las baterías utilizadas en el proyecto.

**Figura 84. Baterías utilizadas**



*Fuente: Esta investigación.*

### 3. PRUEBAS Y RESULTADOS.

Tras el desarrollo de la investigación y las actividades propuestas, se logró diseñar y construir una plataforma robótica, como base para realizar las pruebas del funcionamiento de los sistemas diseñados. En ésta se probaron el funcionamiento del controlador PID de velocidad de las ruedas, el sistema odométrico, el rango y alcance de los sensores y el más importante el controlador de navegación mediante lógica difusa. Adicionalmente, el desarrollo de esta plataforma fue de gran ayuda para adentrarse en el estudio de los sistemas de localización en interiores mediante la onda RF. A continuación se profundiza sobre las características y el desempeño de los sistemas implementados.

#### 3.1 PLATAFORMA ROBÓTICA.

El prototipo final posee las siguientes características físicas. La Figura 85 muestra las dimensiones del robot móvil. En el anexo C, se especifica más detalladamente esta plataforma.

Largo: 22 cm.

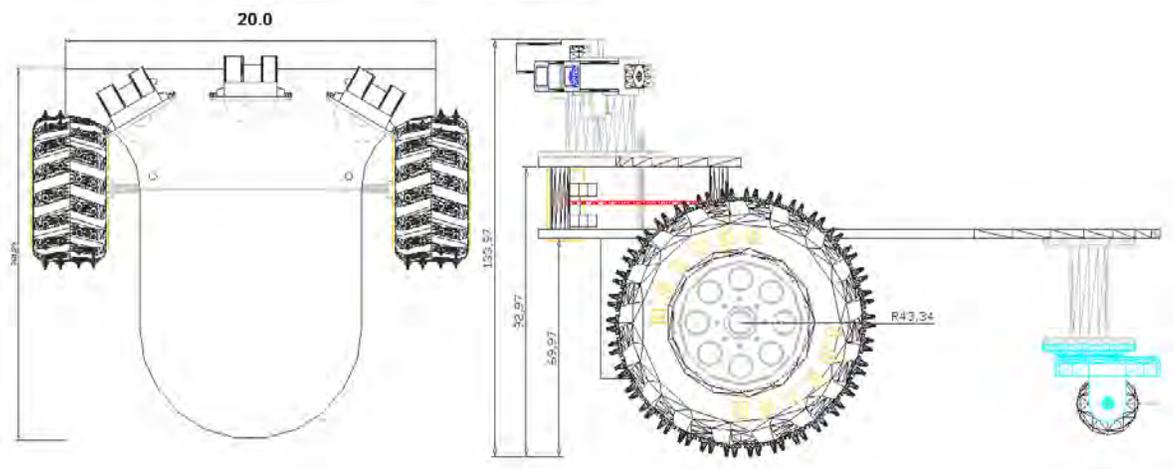
Ancho: 20 cm.

Alto: 15 cm.

Peso: 1500 gramos.

Diámetro de la rueda: 7.84 cm

**Figura 85. Dimensiones del robot móvil.**



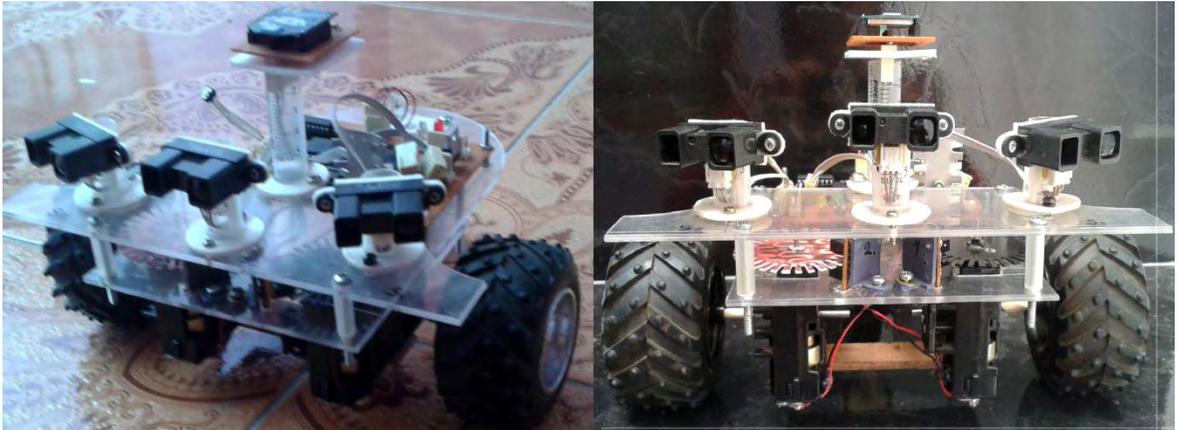
*Fuente: Esta investigación.*

#### Características generales.

- 2 encoders para medida del desplazamiento.
- 3 sensores GP2Y0A02YK para detección de obstáculos.
- Modulo XBee para comunicación inalámbrica.
- PIC18F4550 para control.

- Puertos libres en conectores independientes para múltiples aplicaciones.
- Driver L293B para control de motores.
- Dos motores DC con reducción de 75:1.
- Salida de voltaje regulado de 5V, 3.3V para sensores.

**Figura 86. Robot terminado.**

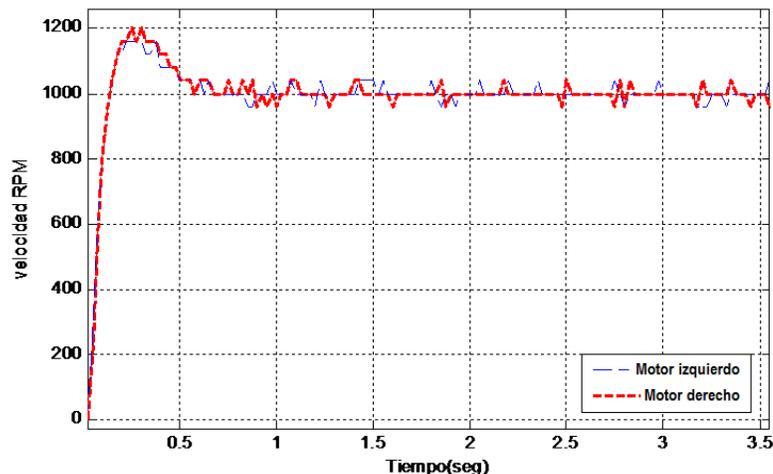


*Fuente: Esta investigación.*

### 3.2 RESULTADOS CONTROLADOR PID.

Se verificó el funcionamiento del controlador PID diseñado que regula la velocidad de las ruedas y las mantiene en el valor deseado. La Figura 87 muestra el funcionamiento del controlador aplicado a los motores ante una entrada de referencia de 1000 RPM. De las pruebas realizadas se obtuvieron tiempos de establecimiento menores a 1000ms y sobrepasos no superiores al 25%, algo de mucha importancia ya que cuando se desea navegar en línea recta las velocidades deben ser lo más iguales posible.

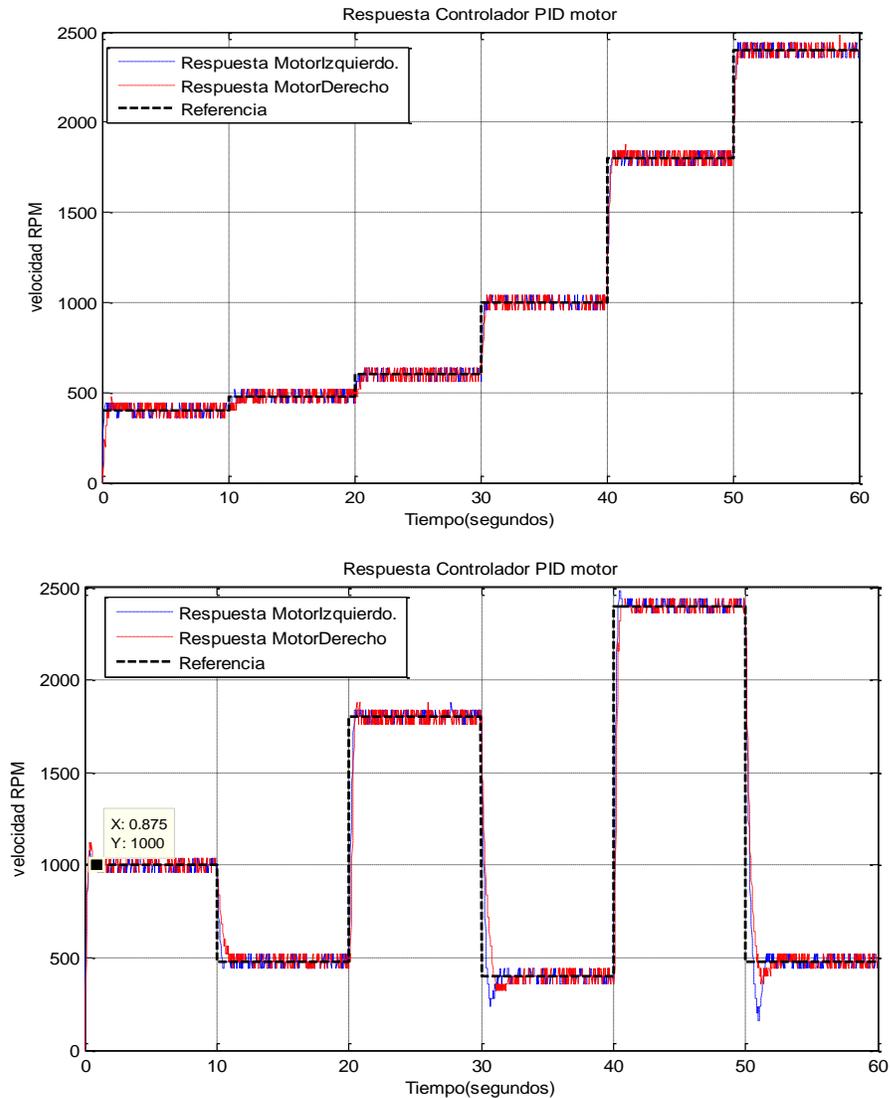
**Figura 87. Respuesta del controlador de velocidad.**



*Fuente: Esta investigación.*

Se probó el controlador diseñado ante diferentes velocidades (escalón multinivel), verificando el comportamiento de éste ante cambios repentinos de velocidad, pensando ya en un ambiente real de navegación, donde el robot necesite disminuir o incrementar su velocidad ante la presencia de obstáculos. El resultado se muestra en la Figura 88, donde se puede apreciar que los dos motores siguen la referencia de manera exitosa.

**Figura 88. Respuesta a un escalón multinivel y a un tren de pulsos.**

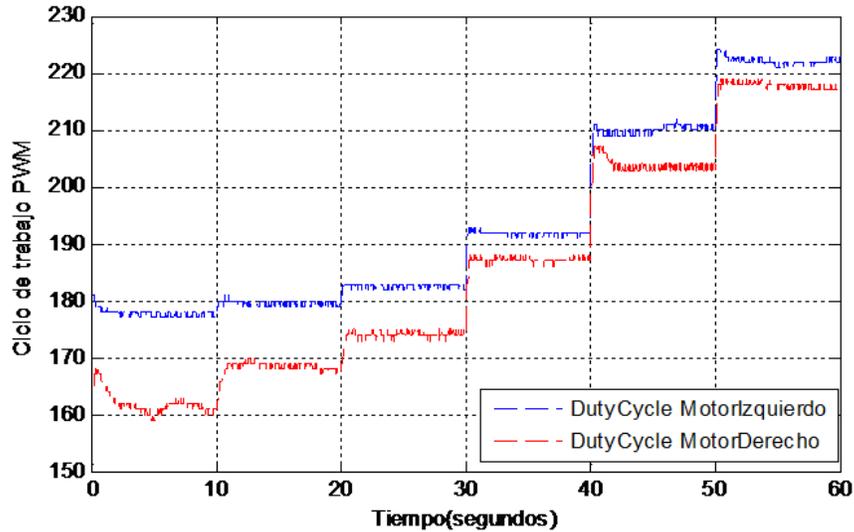


*Fuente: Esta investigación.*

Se puede observar que el controlador responde satisfactoriamente a la entrada de referencia, modificando la variable de control para alcanzar la velocidad de salida requerida, las dos velocidades son constantes una vez se estabiliza el sistema. Es interesante resaltar que aunque se obtiene dos velocidades iguales en los

actuadores, los valores de PWM son diferentes con distintos ciclos de trabajo para cada motor. Esto sucede debido a que los dos motores tienen comportamientos diferentes a pesar de que son de iguales referencias. La Figura 89 muestra los ciclos de trabajo PWM aplicados tanto a los motores izquierdo y derecho.

**Figura 89. Ciclos de trabajo de los PWM izquierdo y derecho.**



*Fuente: Esta investigación.*

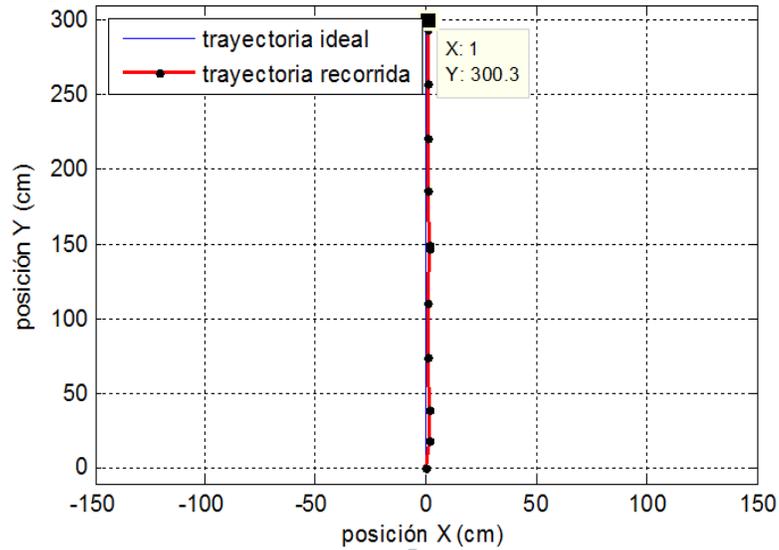
### **3.3 PRUEBA DE NAVEGACIÓN DEL ROBOT.**

Después de comprobar el funcionamiento de los sistemas diseñados individualmente, se realiza la integración de éstos en un mismo programa, el objetivo de las pruebas fue lograr el navegado del robot desde una posición inicial hasta una posición final o meta en un entorno con y sin obstáculos, verificando el comportamiento real con el obtenido en simulaciones. Se desarrollaron diversos experimentos en diferentes condiciones, cambiando la orientación y posición inicial, con obstáculos de diferentes formas y tamaños. Con fines de documentación se presentan cuatro experimentos realizados. Con esto se logró verificar el comportamiento real del controlador lógico difuso para mantener al robot en una trayectoria definida.

#### **3.3.1 Seguimiento de una trayectoria en un entorno libre de obstáculos.**

En esta prueba se eligió la posición inicial en el origen de coordenadas (0, 0) sin desviación hacia la meta, la posición final se ubicó a 3 metros del origen en línea recta, el ángulo de orientación inicial fue de 0 grados. Los datos son enviados a la interfaz gráfica de MATLAB desde la cual se monitorea el comportamiento del robot. La Figura 90 muestra la trayectoria descrita por el robot.

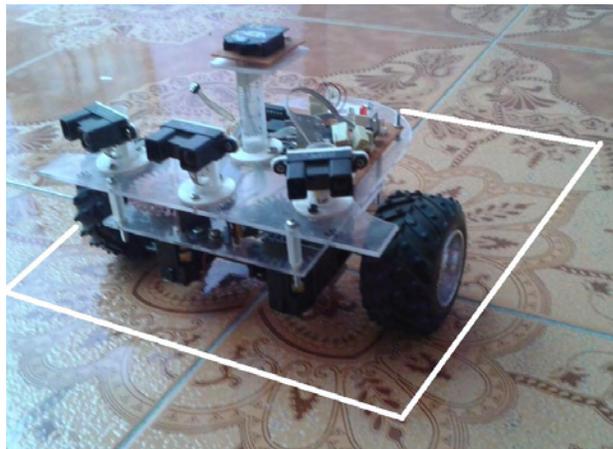
**Figura 90. Navegación entorno libre de obstáculos.**



*Fuente: Esta investigación.*

La posición final obtenida por odometría fue de (1, 303.3) y la orientación final de 0.5 grados. Se mira como el controlador mantiene al robot en su trayectoria a la velocidad máxima. El resultado real de la posición obtenida en la prueba fue de (3, 305), con una orientación de 5 grados. Esto debido a que en la práctica existen factores externos que afectan la medida como la fricción de las ruedas con el piso, el deslizamiento de las mismas y la respuesta de los motores para vencer la inercia inicial. Otro problema que influye en la lectura por odometría es el error en el tren de engranajes, en el cual todo el movimiento suministrado por el motor no es transmitido a la ruedas debido al acople de engranajes y a la holgura entre los mismos. La posición real obtenida en el escenario de pruebas se observa en la Figura 91.

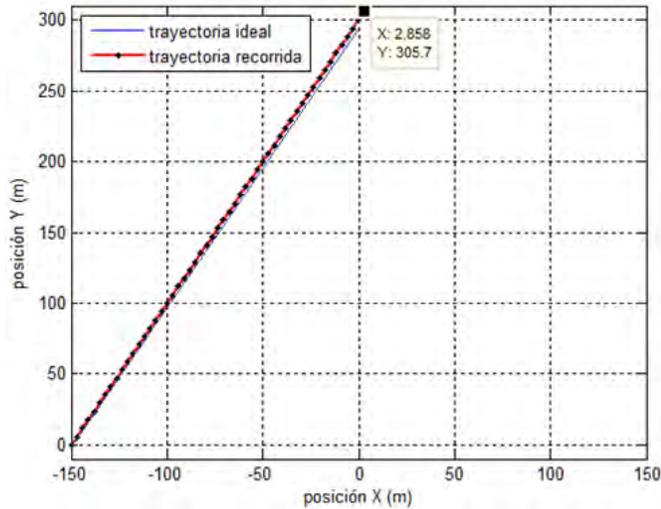
**Figura 91. Posición real del robot en el escenario.**



*Fuente: Esta investigación.*

Para la segunda prueba, la posición inicial del robot es de (-150, 0) con una orientación inicial de 90 grados, la posición final deseada fue de (0, 300). Con esto se logró probar también el funcionamiento con la MTH diseñada, la cual trabaja con dos sistemas de referencia y permite al robot trazar una trayectoria recta desde cualquier punto inicial hasta el final. La Figura 92 muestra dicho comportamiento. La posición final fue de (2.8, 305).

**Figura 92. Navegación en distintas posiciones iniciales.**

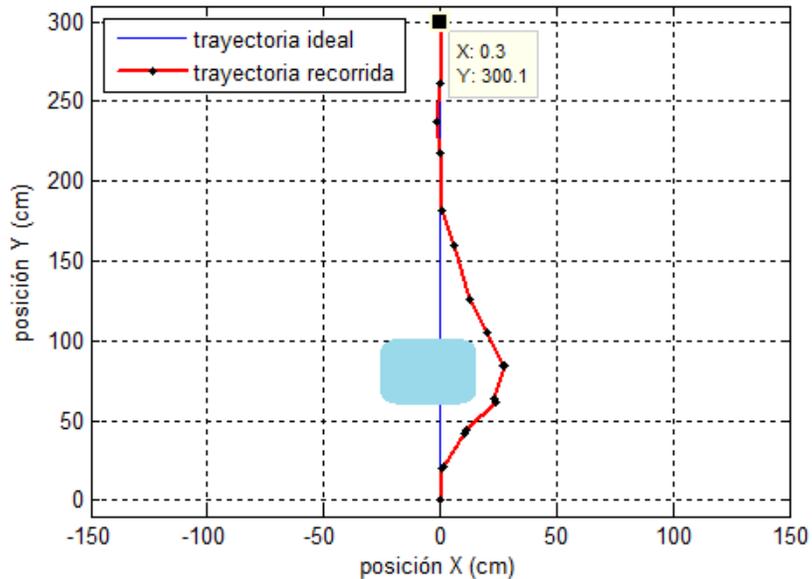


*Fuente: Esta investigación.*

### **3.3.2 Entorno con obstáculos.**

Se realizaron pruebas con obstáculos en la trayectoria del robot, con el fin de comprobar la capacidad que tiene el controlador difuso de tomar una decisión para evadirlos y posteriormente recuperar su camino hacia la meta variando su velocidad según sea el caso. En la prueba se inicia el robot en el origen de coordenadas con una orientación de 0 grados y se coloca un obstáculo en la línea de trayectoria. El comportamiento se muestra en la Figura 93, en ésta se observa como el robot detecta el obstáculo, disminuye su velocidad para posteriormente rodearlo y seguir su camino en línea recta hacia la meta.

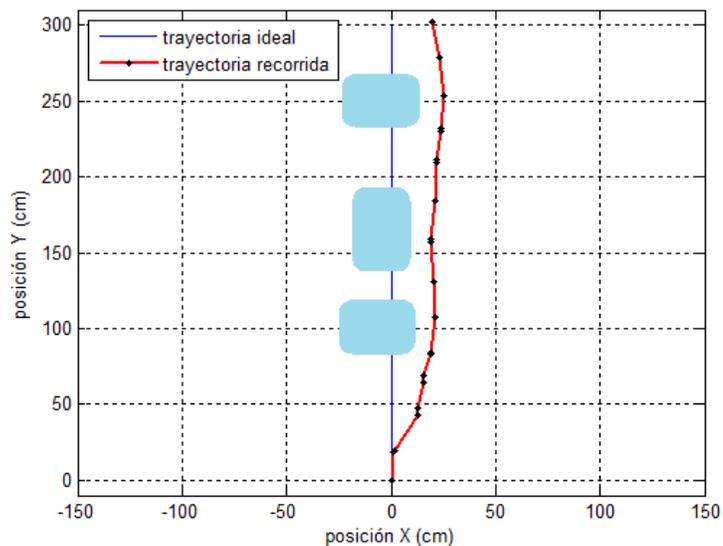
**Figura 93. Navegación entorno con obstáculos.**



*Fuente: Esta investigación.*

En otra prueba se inicia el robot en el origen con una orientación de 0 grados y se colocan 3 obstáculos en la línea de trayectoria (en línea recta). El comportamiento se muestra en la Figura 94, se observa como el control funciona eficazmente y el programa responde a la información suministrada por los sensores. Este tipo de pruebas se tornan mucho más complejas, ya que la orientación cambia constantemente debido a las medidas suministradas por los sensores y la velocidad del robot experimenta cambios continuos.

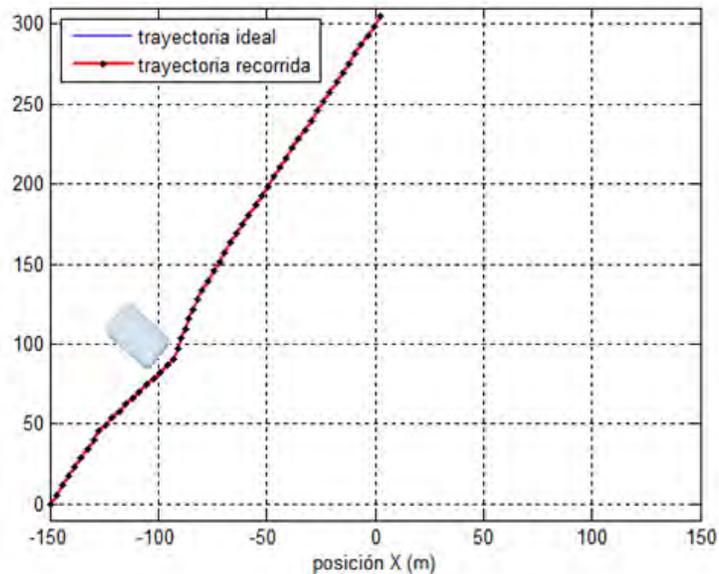
**Figura 94. Navegación entorno con 3 obstáculos.**



*Fuente: Esta investigación.*

La tercera prueba se realiza con una orientación inicial de 90 grados y una posición inicial de (-150,0), la meta es colocada en (0,300), se ubicó un obstáculo por donde se asume la trayectoria ideal del robot. Se observó que la MTH y permite trabajar el sistema como si estuviera en línea recta con la meta con una orientación de 0 grados. El comportamiento se muestra en la Figura 95. La posición final fue de (5,305).

**Figura 95. Aplicación de MTH en la navegación.**



*Fuente: Esta investigación.*

### **3.3.3 Análisis y discusión de resultados.**

De las diferentes pruebas realizadas, se pudo observar que todos los sistemas implementados, como son localización mediante odometría, control PID de velocidad, sistema de percepción, transformación mediante MTH funcionan correctamente, gracias a la información que brindan dichos sistemas se pudo corroborar el funcionamiento del controlador difuso para mantener una trayectoria. En todas las pruebas se observó que el robot alcanza la meta deseada, con una proporción de error aceptable para cada caso.

#### **• Errores en la posición y orientación.**

La principal fuente de errores en la posición es la medida de desplazamiento con odometría, ya que por si alguna razón las ruedas patinan, el robot tomará esta información como datos de desplazamiento. Este error es acumulativo y se incrementa cada vez que se realiza un nuevo cálculo de posición, por lo cual la odometría no es recomendable cuando se desea desplazarse distancias prolongadas. En cuanto al cálculo de orientación mediante odometría el proceso es aun más crítico, ya que la acumulación de errores produce una desviación muy notable en la medida del ángulo de orientación, de esta manera el robot calcula estar en una orientación cuando en realidad lleva otra muy diferente; esto resulta

muy serio si nos damos cuenta que en cualquier vehículo con ruedas lo que se debe controlar con precisión es la orientación. La Tabla 7 muestra los resultados obtenidos en la navegación para alcanzar la meta en un entorno sin obstáculos, en cada caso se estima el error obtenido. Para el cálculo del error se utilizó el error cuadrático:

$$Error = \sqrt{(x_e - x_r)^2 + (y_e - y_r)^2} \quad (52)$$

**Tabla 8. Resultados de la navegación entorno sin obstáculos.**

Pruebas	Meta						Error obtenido	
	Posición real [cm]			Posición obtenida [cm]				
	$x_r$	$y_r$	Theta $_r$ °	$x_e$	$y_e$	Theta $_e$ °	Posición [cm]	orientación grados
Prueba 1	0	300	0	3	303	0	4.2	0
Prueba 2	100	400	0	110	408	5	12.8	5
Prueba 3	-100	400	0	-105	405	-6	7.07	6

La Tabla 8 muestra los resultados obtenidos en la navegación para alcanzar la meta en un entorno con obstáculos.

**Tabla 9. Entorno con obstáculos.**

Pruebas	Meta						Error obtenido	
	Posición real [cm]			Posición obtenida [cm]				
	$x_r$	$y_r$	Theta $_r$ °	$x_e$	$y_e$	Theta $_e$ °	Posición [cm]	orientación grados
Prueba 1	0	300	0	10	305	15	11.1	15
Prueba 2	100	400	0	105	404	13	6.4	13
Prueba 3	-100	400	0	-110	412	-10	15.6	10

En las pruebas se observó que cuando se trabaja con cambios de orientación continuos, es más recomendable realizar giros con ángulos de bajo valor, esto debido a que el controlador funciona de manera eficaz y en el momento en que el robot no detecte obstáculos puede seguir avanzando; en el inicio se pensó trabajar con ángulos grandes pero se presentó el inconveniente que el vehículo se desvió de su trayectoria por errores en la medida de orientación.

### 3.4 PRUEBAS DEL SISTEMA DE POSICIONAMIENTO RF.

Para probar el sistema de posicionamiento basado en el análisis del valor RSSI, se escogió un escenario de prueba, el cual fue el aula 305 de la facultad de ingeniería de la Universidad de Nariño, a éste se le encontró el modelo de

propagación respectivo. Para medir la distancia entre los módulos XBee se utilizó la relación empírica entre la potencia recibida y la distancia entre antenas. En la práctica se comprobó que para minimizar errores del sistema, el escenario no debe sufrir cambios, por ejemplo si en el escenario se agrega un objeto metálico como una silla ésta influye en la medida de potencia recibida por los fenómenos propios de propagación de la onda RF.

### 3.4.1 Recepción de potencia.

Un factor clave para poder realizar el posicionamiento es que las potencias recibidas deben ser las mismas si la distancia entre los módulos XBee se mantiene sin importar la orientación entre éstos. Las pruebas desarrolladas arrojaron que esto no es así ya que las potencias recibidas son diferentes si la orientación del módulo cambia. La Tabla 10 muestra el experimento realizado con los XBee separados 1 metro de distancia, cambiando la orientación en cada caso. Se tomó la potencia recibida por el XBee 1 y el XBee 2.

**Tabla 10. Prueba RSSI a 1 m de distancia con distinta orientación.**

Posición relativa entre antenas		RSSI (dBm)	
XBee1	XBee2	XBee 1	XBee 2
		-42	-43
		-37	-36
		-36	-35
		-35	-34

*Fuente: Esta investigación.*

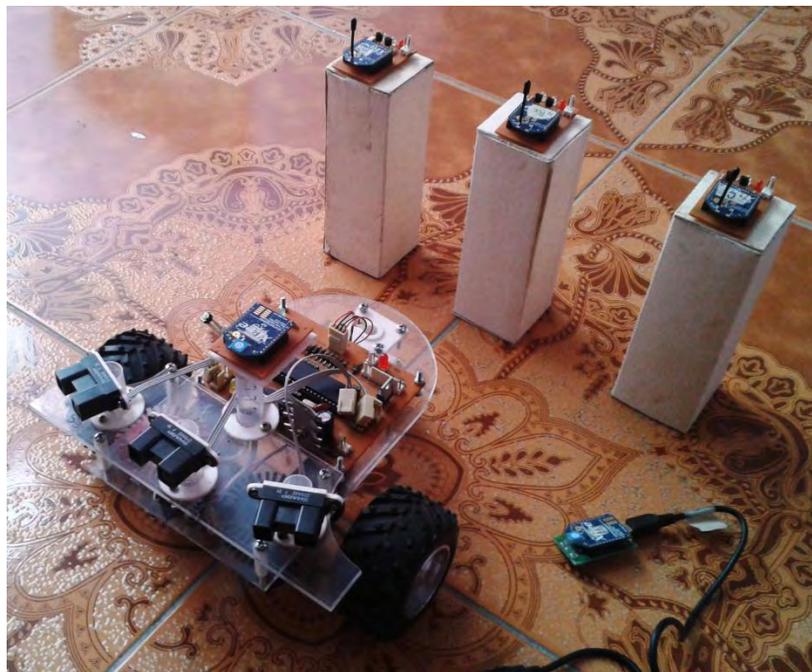
Con esta prueba se miró el comportamiento del nivel RSSI de un nodo fijo con respecto a uno móvil, detectando que el lado por el que el XBee tiene visión directa con el otro módulo afecta en la recepción. Se encontró una diferencia de hasta 4 dBm en el nivel de RSSI lo que afecta directamente la conversión de potencia a distancia y por ende la precisión del sistema. Investigando se encontró que esto puede suceder por el patrón de radiación de la antena dipolo, el cual puede no ser perfectamente omnidireccional debido a la construcción interna de los módulos XBee ya que poseen un plano de masa el cual suele funcionar como antena produciendo fenómenos de interferencia constructiva o destructiva

dependiendo de la fase con que llega la señal<sup>80</sup>. Esta característica también se evidenció en transmisión, obteniendo mayor potencia por lados específicos del XBee pero se debe trabajar con este inconveniente, ya que es una característica propia de este tipo de módulos.

Para solucionar este inconveniente se tomó la decisión de que a la hora de realizar el posicionamiento del robot éste gire sobre su centro de rotación para hacer un barrido de potencias en los 360 grados, luego de cada iteración solo se tomaran los valores de máxima potencia obtenida a cada baliza, al final de este proceso se obtendrán tres potencias máximas con las que se realiza la estimación de distancia y posteriormente la trilateración.

Para probar el algoritmo de posicionamiento, las tres balizas XBee emisoras se elevaron 15 cm del suelo mediante una base rectangular, ésto para lograr la misma altura que tiene la antena del nodo móvil ubicado en el robot. La Figura 96 muestra las bases utilizadas para la implementación del sistema.

**Figura 96. Estaciones baliza y nodo móvil del sistema.**



*Fuente: Esta investigación.*

---

<sup>80</sup> MARTIN, José. Op. cit, p.119.

### 3.5 ALGORITMO DE POSICIONAMIENTO.

El lugar escogido para realizar las pruebas fue el aula 305 de la facultad de ingeniería de la Universidad de Nariño, previamente fue necesario encontrar la relación entre el valor RSSI y la distancia para dicho escenario específico.

#### 3.5.1 Aplicación del modelo para estimación de una distancia.

Para cada baliza se encontró su modelo de propagación respectivo. Una vez realizado este proceso, se creó un programa que asocia la potencia recibida con la distancia, Se realizó una prueba con una sola baliza emisora y el robot móvil desplazándose en línea recta frente a ésta, los datos se tomaron incrementando la distancia cada 10 cm. Se tuvo en cuenta que los módulos se ubiquen de tal forma que puedan transmitir y recibir la máxima potencia. Para fines de documentación en la Tabla 10 se muestra el resultado obtenido con la baliza 1, en ella se mira la distancia real y la estimada mediante el modelo de propagación. El mismo procedimiento se realizó para cada baliza aplicando en cada caso el modelo de propagación respectivo.

**Tabla 11. Asociación de potencia a distancia para la baliza 1.**

<b>RSSI [dBm]</b>	<b>Distancia real [cm]</b>	<b>Distancia estimada [cm]</b>	<b>Error [cm]</b>
-28	20	20	0
-28	30	20	10
-29	40	40.1	0,1
-30	50	51.04	1,04
-31	60	57.1	2,9
-32	70	63.2	6,8
-32	80	63.2	16,8
-34	90	78.1	11,9
-35	100	87.9	12,1
-35	110	87.9	22,1
-37	120	109.1	10,9
-38	130	121.6	8,4
-38	140	121.6	18,4
-40	150	151.02	1,02
-41	160	172.2	12,2
-41	170	172.2	2,2
-42	180	185.1	5,1
-40	190	172.2	17,8
-43	200	199.91	0,09
-41	210	172.2	37,8
-46	220	265.3	45,3
-46	230	265.3	35,3

-46	240	265.3	25,3
-42	250	185.1	64,9
-47	260	265.6	5,6
-46	270	247.1	22,9
-48	280	285.5	5,5
-48	290	285.4	4,6
-49	300	306.9	6,9

En la tabla anterior se puede observar como el error aumenta proporcional con la distancia y que se incrementa más a partir de los 2 metros por lo que se puede decir que cuando el robot se encuentra cerca de la baliza tendrá mayor precisión que cuando se encuentra más alejado de éstas.

### 3.5.2 Comportamiento de la potencia al aumentar la altura de las balizas.

Esta prueba se realizó con el fin de observar el comportamiento del valor RSSI cuando los nodos baliza se encuentran a diferentes alturas del nodo móvil pensando en trabajos futuros de posicionamiento donde se podría colocar las tarjetas baliza en alturas superiores por ejemplo paredes, postes, etc. Se pudo comprobar que la potencia disminuye cuando se incrementa la altura de las balizas. La Tabla 11 muestra el resultado de la prueba, donde se empezó con una altura de 15 cm para un nodo baliza, la distancia de separación con respecto al robot fue de 1 metro. Posteriormente se incrementó la altura del nodo fijo y se fueron calculando los valores RSSI.

**Tabla 12. Variación de la potencia con la altura.**

Altura desde el piso [cm]	Valor RSSI [dBm]
15	-33
35	-34
47	-34
55	-36
75	-38
115	-44
140	-48

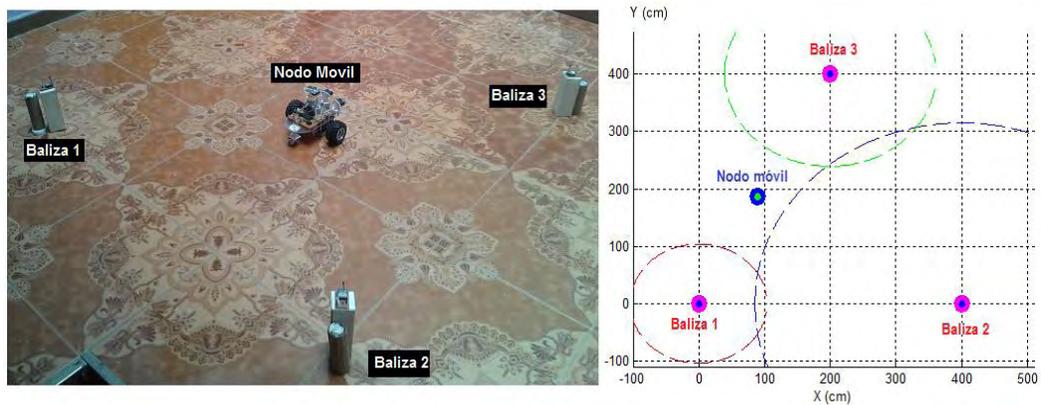
### 3.5.3 Resultados posicionamiento.

Para realizar la prueba de posicionamiento, se formó un área de  $4 m^2$  delimitada por las balizas emisoras. Las posiciones fijas de las balizas se muestran en la Tabla 13. La Figura 97 muestra el espacio delimitado por los módulos.

Tabla 13. Posiciones de las balizas emisoras.

Baliza	Posición (x,y) [cm]
Router 1	(0,0)
Router 2	(400,0)
Router 3	(200,400)

Figura 97. Área de posicionamiento.



Fuente: Esta investigación.

### 3.5.4 Comportamiento del valor RSSI de acuerdo al ángulo de emisión/recepción de las balizas.

Como se mencionó anteriormente el valor RSSI varía de acuerdo al ángulo por donde el módulo emite o recibe la señal de radiofrecuencia, esto debido al patrón de radiación de los XBee. Para ello se realizó una prueba para cada baliza en la cual se colocó el módulo receptor a una distancia de 300 cm y se fue variando el ángulo por el que éste tiene visión directa con el emisor, comprobando que el valor RSSI sufre cambios debido a la orientación entre módulos. La Tabla 14 muestra el resultado de este experimento.

Tabla 14. Valores RSSI con distintos ángulos de emisión/recepción.

Ángulo [grados]	RSSI [-dBm]		
	Baliza 1	Baliza 2	Baliza 3
0	51	52	53
15	50	52	52
30	50	51	50
45	51	53	49
60	51	52	49
75	51	50	48

90	50	50	49
120	49	53	49
135	49	50	48
180	52	52	54
195	52	52	59
210	51	52	61
225	52	51	62
240	56	53	57
255	58	50	60
270	55	49	57
300	51	50	63
315	50	50	60
330	51	51	57
360	51	52	52

### 3.5.5 Posicionamiento en distintas coordenadas de prueba.

Posteriormente se colocó el robot en el espacio delimitado por las balizas y se ejecutó el algoritmo de posicionamiento en diferentes puntos de prueba. Para cada caso se calculó el error cuadrático para la posición estimada con respecto a la real.

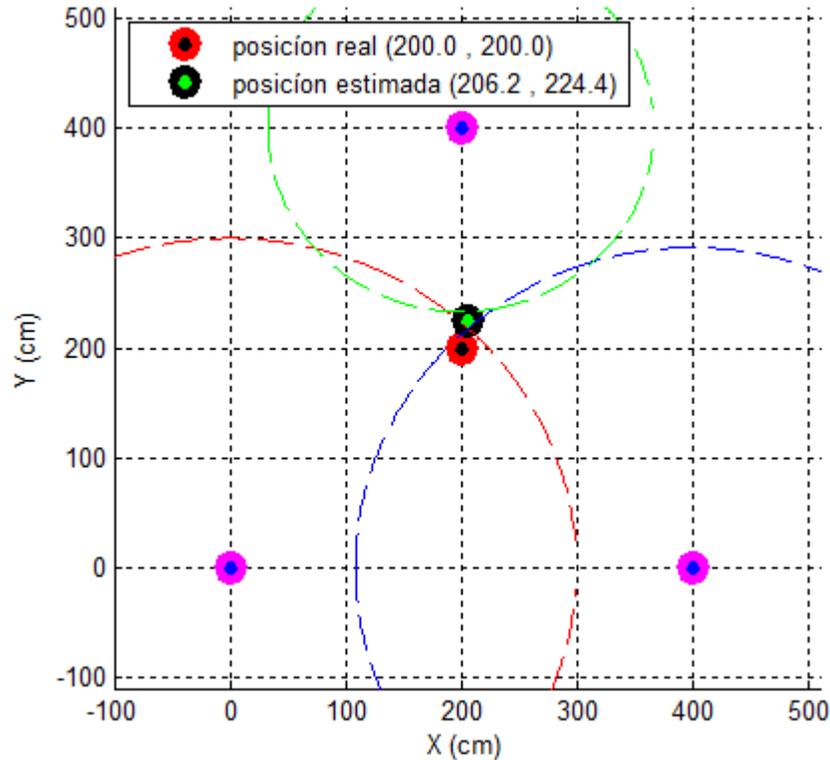
#### Prueba1: Posicionamiento robot en centro (200,200).

En esta prueba se calculó por triangulación la ubicación en la que se encuentra el robot, eligiendo la posición central en centímetros (200,200) como punto de prueba. Después de realizar el giro sobre su centro de gravedad y un barrido para elegir la máxima potencia se encontró la posición estimada. En la Figura 98 se observa la ubicación real y la calculada del nodo móvil. Se realizaron cuatro repeticiones sobre el mismo punto y se cálculo del error en posicionamiento en cada caso. La Tabla 15 muestra los resultados de la prueba, en ésta se encuentran los valores RSSI obtenidos, la conversión de potencia a distancia para cada baliza y la posición estimada del robot.

**Tabla 15. Posicionamiento robot en el centro (200, 200).**

Prueba N°	Posición real [cm]		RSSI [dBm]			Distancia estimada [cm]			Posición estimada [cm]		Error [cm]
	X	Y	P1	P2	P3	d1	d2	d3	X	Y	
1	200	200	-49	-46	-47	359	221	166	206.2	224.4	25
2			-48	-45	-42	279	283	173	197.4	247.9	47
3			-47	-45	-43	279	283	180	196.9	211.1	11
4			-48	-46	-47	359	221	166	206.2	224.4	25

**Figura 98. Posicionamiento robot en el centro.**



*Fuente: Esta investigación.*

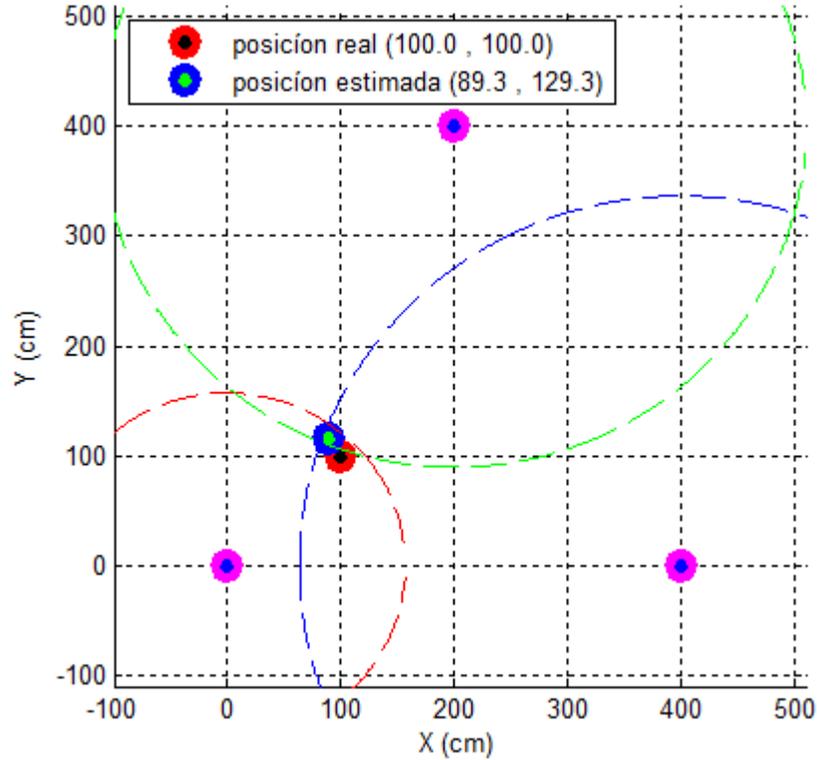
**Prueba 2: Posicionamiento robot cercano a baliza 1. (100, 100).**

Para esta prueba se eligió el punto (100, 100) como posición a calcular, comprobando que en ese punto el valor RSSI emitido por la baliza 1 fue aproximadamente de -46 dBm en todas las repeticiones, el cual es mayor que el emitido por las estaciones 2 y 3 (-50 dBm y -51 dBm respectivamente). Al obtener una potencia más alta en la estación 1, se puede asumir que el resultado estará en una zona cercana al primer módulo. Se realizaron cuatro repeticiones en el mismo punto y se registraron los resultados en la Tabla 16. La Figura 99 muestra el resultado de esta prueba.

**Tabla 16. Resultado posicionamiento punto (100, 100).**

Prueba N°	Posición real [cm]		RSSI [dBm]			Distancia estimada [cm]			Posición estimada [cm]		Error [cm]
	X	Y	P1	P2	P3	d1	d2	d3	X	Y	
1	100	100	-42	-46	-48	153	314	305	89.3	129.3	31
2			-40	-44	-40	126	255	284	148.0	149.0	68
3			-38	-46	-45	115	314	112	92.8	204.4	104
4			-37	-46	-45	103	314	160	89.3	186.3	86

**Figura 99. Posicionamiento robot cercano a baliza 1.**



Fuente: Esta investigación.

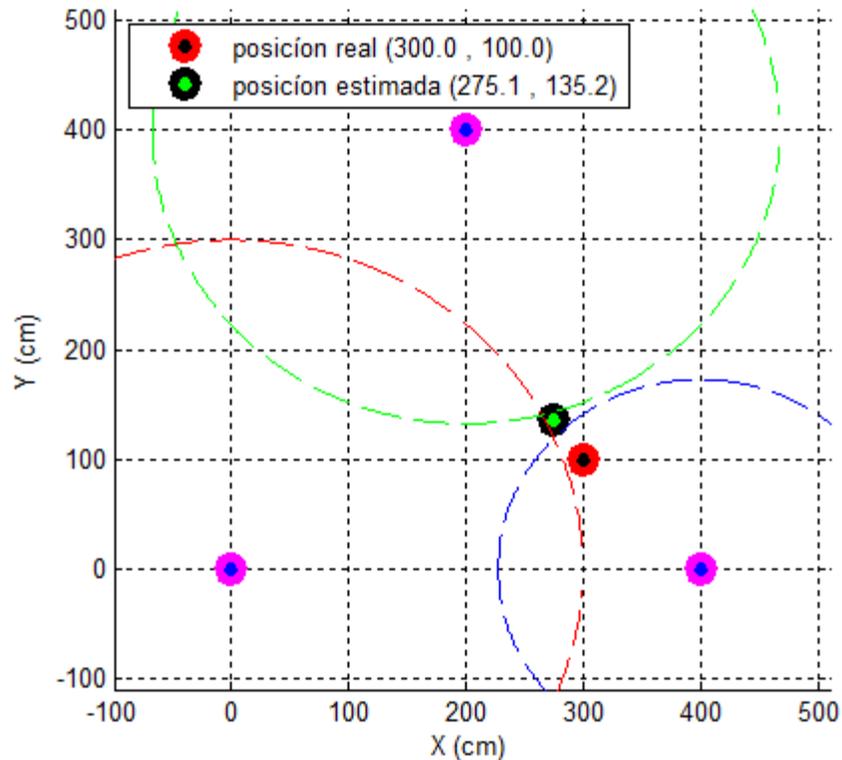
**Prueba 3: Posicionamiento robot cercano a baliza 2. (300, 100).**

En esta prueba se calculó el posicionamiento en un punto cercano a la baliza 2 (300, 100). Se registraron los valores obtenidos los cuales se muestran en la Tabla 17. La Figura 100 muestra los resultados de esta prueba.

**Tabla 17. Resultado posicionamiento punto (300, 100).**

Prueba N°	Posición real [cm]		RSSI [dBm]			Distancia estimada [cm]			Posición estimada [cm]		Error [cm]
	X	Y	P1	P2	P3	d1	d2	d3	X	Y	
1	300	100	-50	-39	-55	300	160	270	275.1	135.2	43
2			-44	-37	-43	244	137	353	250.9	42.8	75
3			-46	-38	-55	256	158	271	251.0	114.5	51
4			-44	-38	-43	244	158	353	243.2	46.7	77

**Figura 100. Posicionamiento robot cercano a baliza 2.**



*Fuente: Esta investigación.*

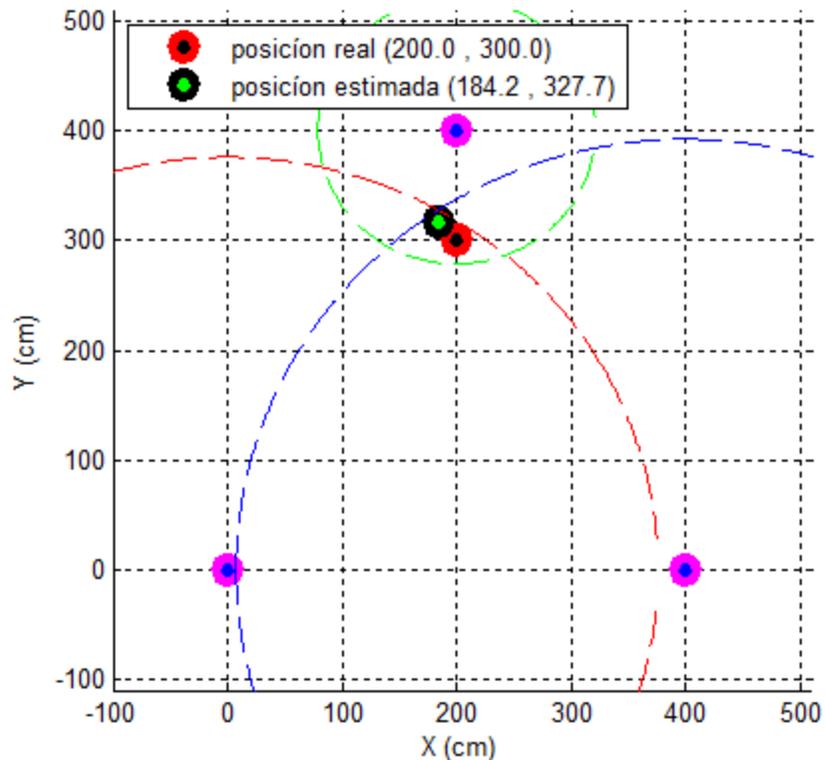
**Prueba 4: Posicionamiento robot cercano a baliza 3. (200, 300).**

Se realiza el proceso de triangulación para la posición (200,300), la Tabla 18 muestra los valores RSSI obtenidos en cada prueba, su respectiva conversión de potencia a distancia y el valor estimado de posicionamiento. La Figura 101 ilustra el resultado obtenido en este punto.

**Tabla 18. Resultado posicionamiento punto (200, 300).**

Prueba N°	Posición real [cm]		RSSI [dBm]			Distancia estimada [cm]			Posición estimada [cm]		Error [cm]
	X	Y	P1	P2	P3	d1	d2	d3	X	Y	
1	200	300	-55	-49	-47	361	385	137	184.2	327.7	31
2			-48	-49	-34	291	356	149	147.2	254.8	69
3			-46	-47	-45	256	349	112	129.3	251.8	85
4			-45	-47	-32	255	349	127	128.5	246.8	89

**Figura 101. Posicionamiento robot cercano a baliza 3.**

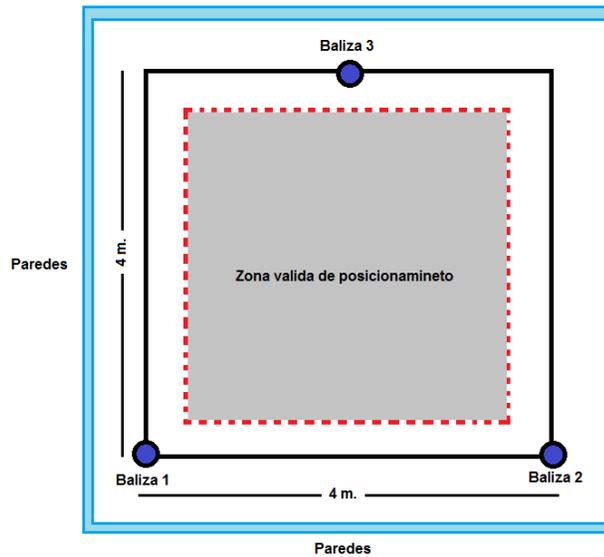


*Fuente: Esta investigación.*

En la práctica se observó que cuando el nodo móvil se encuentra más cercano a las paredes, la potencia se vuelve mucho más aleatoria debido a las reflexiones provocadas por las mismas, por esta razón se decidió alejar las balizas por lo menos 1 metro de las paredes y trabajar una zona válida en donde los valores RSSI no se miren tan afectados por los fenómenos de propagación de la onda. En Figura 102 se puede apreciar mejor la zona elegida para realizar el posicionamiento.

Por ejemplo cuando el receptor se encuentra ubicado con respecto a la baliza 1 a una distancia de 2 m en el centro de la habitación, el valor RSSI es de -46 dBm, pero cuando se encuentra a la misma distancia pero cercano a las paredes el valor fue de -50 dBm.

**Figura 102. Zona válida para posicionamiento.**



*Fuente: Esta investigación.*

Cabe resaltar que la rotación realizada por el robot móvil sobre su centro de gravedad para realizar el barrido y muestreo de potencias es de gran importancia, ya que así se logró obtener el valor de máxima potencia de cada módulo. De no ser así el error sería aun mayor por el inconveniente comentado anteriormente con los módulos XBee, los cuales no reciben el mismo valor de potencia por todos sus lados. Como era de esperarse, las pruebas presentaron más error cuando el robot estuvo más alejado de una baliza, ya que en el modelo de propagación, a partir de los 2 metros de distancia aproximadamente, el valor de potencia se hace más aleatorio. En este caso se obtuvieron potencias erróneas que el algoritmo asoció a otras distancias, al tener tres balizas emisoras se obtienen tres medidas de distancia con incertidumbre lo que incrementa el error en el posicionamiento del nodo móvil. De todas formas dependiendo de la aplicación que se le quiera dar al sistema de posicionamiento, los primeros resultados son aceptables y permiten pensar en la mejora del sistema para futuras investigaciones.

#### 4. CONCLUSIONES.

La implementación de un controlador lógico difuso, es un método válido para dotar de autonomía a un robot móvil, se comprobó el correcto funcionamiento del sistema cuando se desea mantener al robot en una trayectoria definida, ayudando a la toma de decisiones sobre todo ante la presencia de obstáculos.

Se logró implementar en lenguaje C el controlador difuso diseñado en MATLAB, algo de gran importancia cuando se desea dar soluciones a distintos procesos de control, ya que se pueden implementar en microcontrolador de manera fácil y económica técnicas de control inteligente. Un FLC es propicio cuando no se tiene un modelo matemático o el proceso es difícil de modelar.

En un sistema robótico que base su posicionamiento mediante odometría, hay que tener en cuenta que la acumulación de errores en el cálculo de orientación causa grandes errores en la estimación de la posición, los cuales se incrementan proporcionalmente con la distancia recorrida por el robot, así que esta técnica no sería recomendable para el desplazamiento de grandes distancias o se debe tener un sistema adicional que permita verificar con precisión la posición y orientación del robot. No obstante en robótica móvil, el sistema de posicionamiento por odometría siempre será importante, debido a que entrega información constante del desplazamiento, sin requerir de una difícil implementación.

Se probó que se puede implementar un sistema de posicionamiento para interiores utilizando la tecnología Zigbee, a través del procesamiento del valor RSSI en los módulos XBee, para ello es fundamental trabajar en modo API, en donde se aprovechan al máximo la funcionalidad de los módulos.

Con el desarrollo de la investigación, se conoció que la potencia recibida por los módulos XBee, varía dependiendo mucho del lado por el cual se recibe la señal, también se observó que existe un lado por el cual emite mayor potencia. Se piensa que esto puede suceder por la ubicación de la antena chip, además puede ser que el patrón de radiación no es completamente omnidireccional, lo que afecta el posicionamiento del nodo móvil.

En la realización del sistema de posicionamiento basado en RSSI, no se puede obtener una gran precisión, debido a que el valor de la potencia no solo varía con la distancia entre antenas, sino que depende de muchos factores como fenómenos de reflexión, absorción y difracción de la onda RF, lo que hace que se produzcan errores en la conversión de potencia a distancia que influyen en el posicionamiento final del nodo móvil.

## 5. RECOMENDACIONES.

Para mejorar la precisión del sistema odométrico y del control de velocidad del robot, se recomienda utilizar un encoder de mayor resolución, preferiblemente uno que esté incorporado en el motor, debido a que al ser construido industrialmente, son de tamaño reducido, presentan una gran cantidad de pulsos por vuelta y ocupan un mínimo espacio en la plataforma, además pueden ser fácilmente acoplados en cualquier eje.

El procedimiento llevado a cabo en el diseño del FLC, puede servir de guía a proyectos que requieran implementar un control difuso en otros procesos, como control de temperatura, luminosidad de interiores, control de motores, etc. Ya que el código desarrollado fue optimizado para ocupar el mínimo de recursos en el microcontrolador y las librerías diseñadas pueden ser utilizadas para otras aplicaciones en la universidad.

El sistema de posicionamiento RF diseñado, no es exclusivo solo para robótica móvil, también puede ser aplicado en distintos campos como: Localización en interiores de hospitales, vigilancia, geriátricos, guiado de turistas en museos. Se recomienda seguir trabajando en este campo desde la región y particularmente desde la Universidad de Nariño, ya que es un área de investigación actual.

En el diseño de FLC implementados en microcontrolador, se recomienda que la salida esté compuesta por varios conjuntos difusos, con ello se garantiza más resolución y precisión en la salida tomando decisiones acertadas. Otra ventaja evidenciada es que al tener varios conjuntos de salida el proceso de defuzificación es más rápido debido a la activación de reglas concretas.

Si se piensa en la realización de sistemas de posicionamiento para interiores, los módulos XBee serie 1 serían una buena opción ya que el valor RSSI viene incluido por defecto en las tramas API, sin tener que realizar procesos adicionales. También se recomienda utilizar un mayor número de balizas emisoras para mejorar la precisión del sistema.

Para mejorar la precisión del sistema de posicionamiento, se recomienda realizar un filtrado y refinamiento de los datos de potencia, ésto permitirá descartar potencias invalidas y por ende mejorar el posicionamiento del nodo móvil.

## BIBLIOGRAFIA

OLLERO BATURONE, Aníbal. Robótica: Manipuladores y robots móviles. 1 ed. Barcelona: Marcombo. 2001. 447p.

DEL BRIO, Martin. Redes neuronales y sistemas borrosos. 3 ed. México DF. AlfaOmega. 2007. 440p.

OGATA, Katsuhiko. Sistemas de control en tiempo discreto. Traducido por José Miguel Aranda Ramírez. 3 Ed. México. Prentice Hall. 1996. 756p.

ACUÑA REGALADO Carlos, Diseño y construcción de un robot móvil que pueda desplazarse dentro de un laberinto. Escuela politécnica nacional, Quito. 2006. 11p.

PALACIOS E, DOMÍNGUEZ F, LÓPEZ J. Microcontrolador PIC16F84-Desarrollo de proyectos, 1 ed. Alfaomega, 2004, 515p.

MUJICA, Dante. Navegado autónomo de Robots móviles en ambientes real/virtual, Tesis (ingeniería electrónica). Instituto politécnico nacional. Escuela superior de ingeniería mecánica y eléctrica. México. Mayo. 2008. 81p.

GONZALES JIMENEZ, Javier. Estimación de la posición de un robot móvil. España. 2010. 25p.

GILABERTE Esther. Implementación de sensores de ultrasonido en un sistema autónomo de tiempo real. Tesis. Marzo. 2003. 146p.

Parra H, Ríos L, Bueno M. Navegación de Robots móviles mediante comportamientos utilizando lógica difusa. Scientia Et Technia. Mayo. 2007. 8p.

MARTINEZ, Juárez, C. Control Difuso, capítulo 14. Universidad de Murcia. 2008. 58p.

GARCIA, Eduardo. Compilador C CCS y simulador PROTEUS para microcontroladores PIC. Editorial Alfa-Omega. 2008. 270p.

OGATA, Katsuhiko. Ingeniería de control moderna. Traducido por Miguel Ángel Martínez Ramírez, 3 ed. México. 1998. 1015p.

KUO, Benjamín. Sistemas de control Automático. Traducido por Guillermo Aranda Pérez. 7 ed. México. 1996. 930p.

GUTIERREZ, Maria. Indoor location systems based on zigbee networks. Mikkelin University of Applied Scieces. April 2012. 46p.

MARTIN, José. Posicionamiento por radiofrecuencia basado en robots colaborativos. Universidad autónoma de Madrid. España. 2012. 183p.

DIAZ, Santiago. Redes de sensores inalámbricas aplicada a robótica colaborativa. Universidad autónoma de Madrid. Escuela politécnica superior. España 2009. 110p.

CORTEZ, Isabel. Robot Autónomo como nodo móvil de redes de sensores. Universidad Autónoma de Madrid. Junio. 2011. 113p.

CHAHUARA QUISPE, José. Control neuro-difuso aplicado a una grúa torre. Tesis digitales USMSM. 38p.

PAZ, Leobardo. Localización en interiores y descubrimiento de servicios. Universidad Autónoma de Guadalajara. Tecnologías de Internet. México. 2011. 6p.

RUIZ, Isidro. Sistema de posicionamiento para interiores. Universidad pontificia comillas. Escuela técnica superior de ingeniería. Madrid. Septiembre. 2010. 132p.

DIGI, International Inc. API Operation XBee/XBee PRO ZB RF Modules. 2012. 60p

CHARFUELAN BURBANO, Diego. Diseño de un robot oruga teleoperado para exploración visual y toma de muestras ambientales con sensores en lugares de difícil acceso. Trabajo de grado, (ingeniería electrónica). Universidad de Nariño. Facultad de ingeniería. Pasto. 2009. 161p.

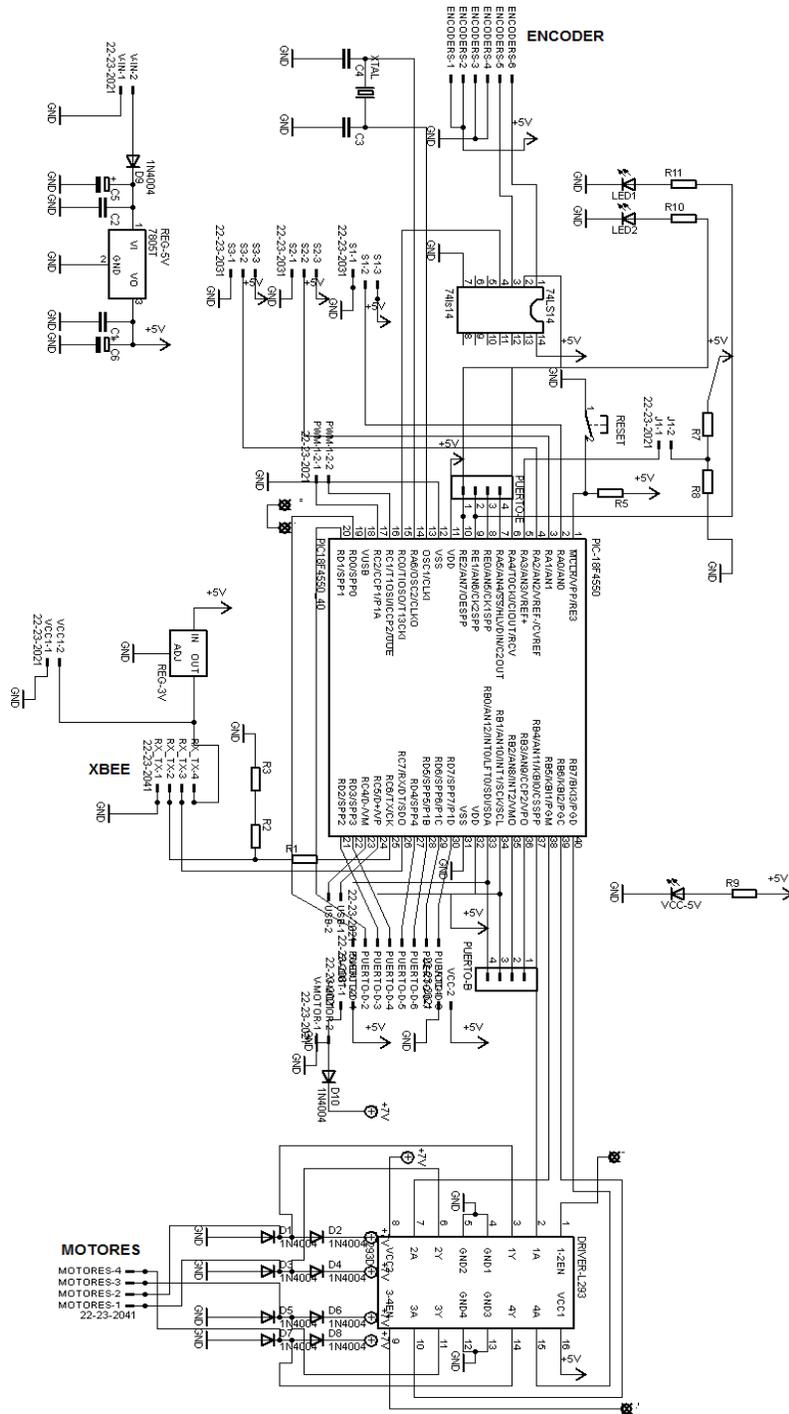
BARCO JIMENEZ, John. Diseño e implementación de un controlador PID adaptativo basado en un procesador digital de señales dsPIC aplicado al control de procesos industriales. Trabajo de grado, (ingeniería electrónica). Universidad de Nariño. Facultad de ingeniería. Pasto. 2009. 265p.

PEREZ CAMPANERO, Juan. Clasificador de actividades. Universidad Pontificia Comillas. Ingeniería en informática. España. 2009. 119p.

ROCA, Daniel. Posicionamiento con una red de sensores inalámbricos (WSN): implementación para una aplicación real. Universidad Autónoma de Barcelona (UAB). Departamento de Telecomunicaciones e ingeniería de sistemas. España. 2009. 113p.

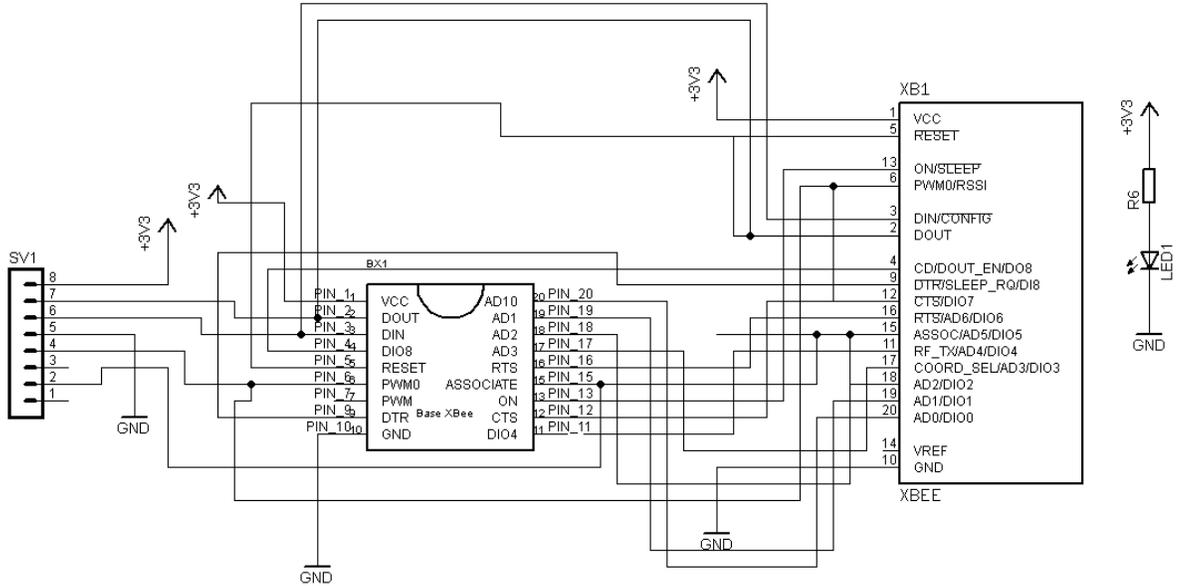
# ANEXOS.

## Anexo A. Circuito Eléctrico Tarjeta de control Robot.

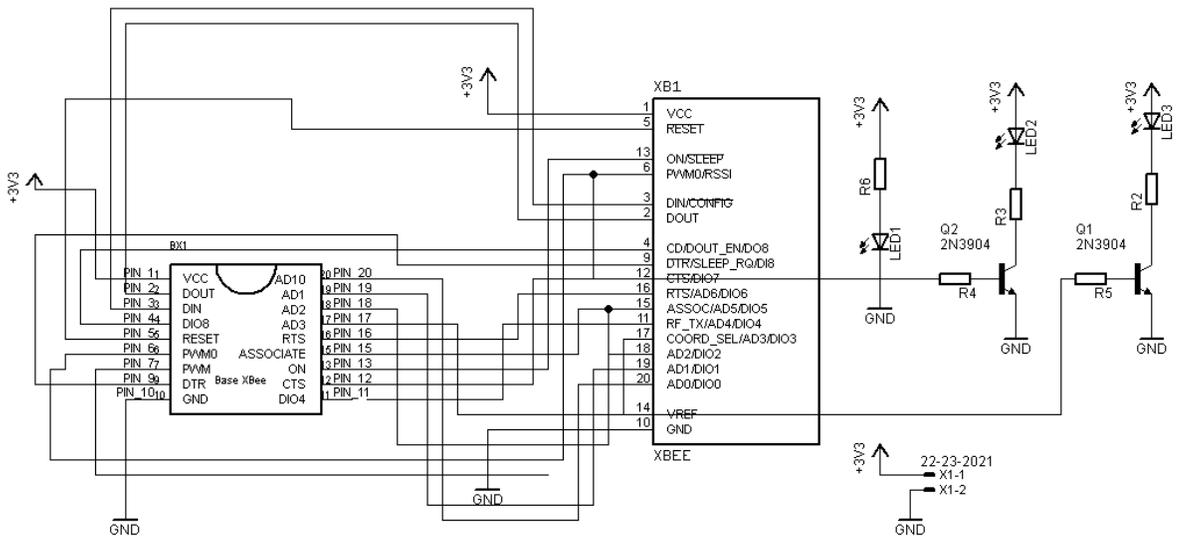


## Anexo B. Circuito Eléctrico modulo XBee Robot y balizas XBee.

### Diagrama esquemático XBee Robot.

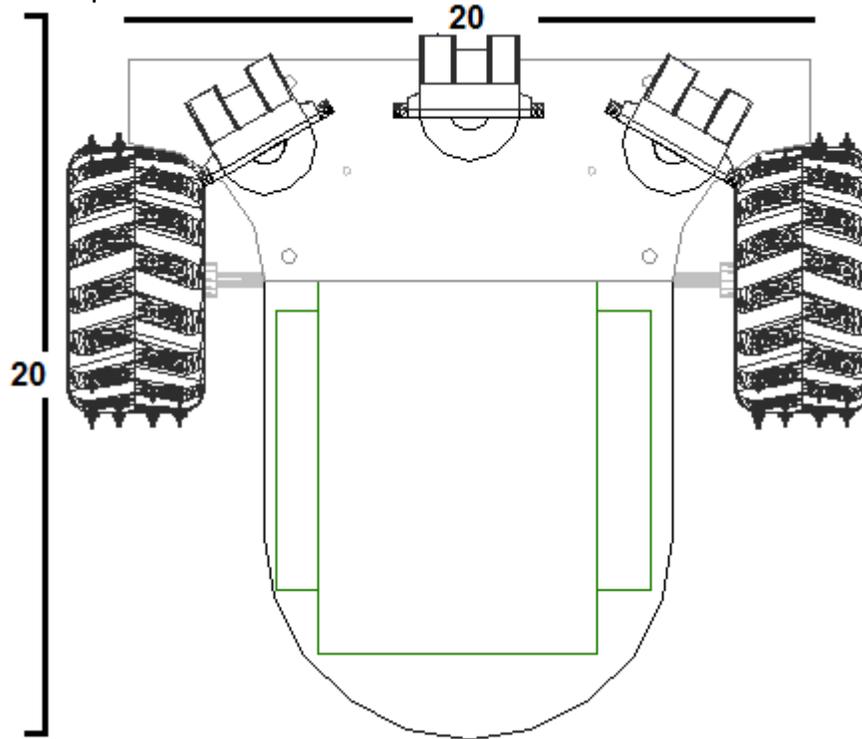


### Diagrama esquemático balizas XBee.

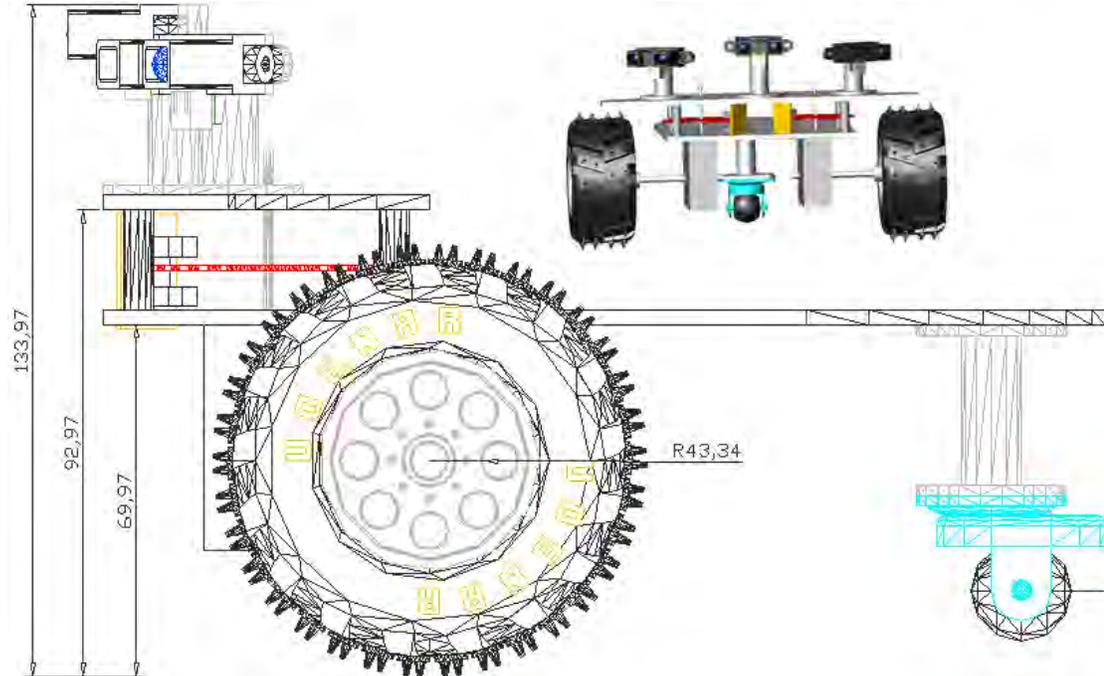


### Anexo C. Esquema general del robot.

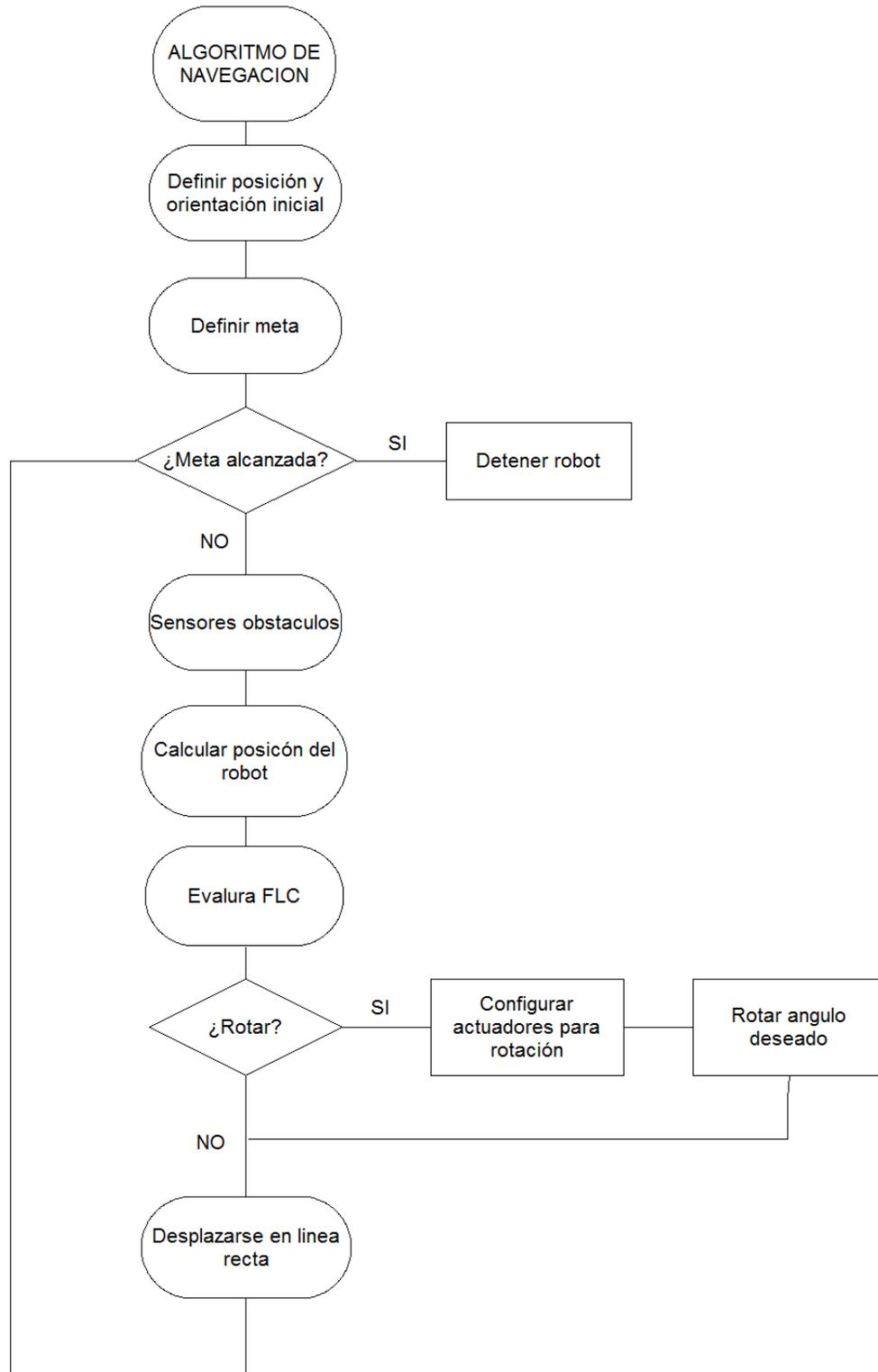
Vista superior



Robot móvil vista lateral.



## Anexo D. Algoritmo de navegación del robot.



## Anexo E. Código fuente robot móvil.

```
//-----UNIVERSIDAD DE NARIÑO-----
//-----FACULTAD DE INGENIERÍA-----
//-----DEPARTAMENTO DE ELECTRONICA-----
//-----
//El siguiente es el código fuente utilizado para la realización del trabajo de
//grado: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE NAVEGACIÓN AUTÓNOMA BASADO EN
// LÓGICA DIFUSA PARA UN ROBOT MÓVIL. En este se encuentran anexas todas las
//librerías creadas para el funcionamiento y pruebas del robot.
//-----
#include <18f4550.h>
#define adc=10
#include <math.h>
#include <Triangular.h>
#include <MaxMin2.h>
#include <difuso5.c>
#define HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGEN
#define delay (clock=48000000)
#define rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8, parity=N)
#define TOCON_REG =0xFD5
#define T1CON_REG =0xFCD
#define TRISC =0xF94
#define standard_io(a)
#define standard_io(b)
#define standard_io(e)
#define LED3 PIN_E1
#define LED4 PIN_E2
#define LED_ON(x) output_high(x)
#define LED_OFF(x) output_low(x)
//-----
//-----tramas api recibidas-----

//7E 00 12 10 01 00 13 A2 00 40 A6 20 79 FF FE 00 00 58 42 45 45 99
int8 RespuestaDB[21]={};
int8 Iniciar[18]= {0x7E, 0x7E, 0x7E, 0x7E, 0x00,0x00,0x00,0x7E, 0x7E, 0x7E, 0x7E, 0x00,0x00,0x00};
char DB[11] = {0x31,0x31,0x31,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0};
char ASO[11] = {0x31,0x31,0x31,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0};
//-----tramas api enviadas-----
int8 TramaDB[8]= {0x7E, 0x00, 0x04, 0x08, 0x03, 0x44, 0x42, 0x6E};
int8 comp[44]= {};
int8 Paquete[17]= {0x7E, 0x00, 0x12,
                  0x10, 0x00, 0x00, 0x13, 0xA2, 0x00, 0x40, 0x91, 0x9C, 0x3B, 0xFF, 0xFE, 0x00, 0x00};
int8 Prem[19]= {0x7E, 0x00, 0x0F,
               0x17, 0x03, 0x00, 0x13, 0xA2, 0x00, 0x40, 0xA6, 0x20, 0x80, 0xFF, 0xFE, 0x02,
               0x44, 0x42, 0x25 };
```



```

        {
        counterP++;
        Rssi[counterP]=RespuestaDB[18];
        pasolnt=15;
        PreguntarPotencia();
        pasolnt=4;
        break;
        }
    else{
        paso=1;
        break;}//volver a enviar trama remota
case 4:maxi=11;
tramaCorrecta=get_string1(DB,0x06,maxi,0x88);//recibir potencia del último paquete
if (tramaCorrecta==1){
    Potencias[counterP]=DB[8];
    Xbee=Xbee+1;
    paso=1;
    if (Xbee==4){
        continuar=0;
        paso=0;
        pasolnt=0;
    }
    break;
}
else{
    PreguntarPotencia();
    pasolnt=4;
    break;}
case 31:
    maxi=6;
    tramaCorrecta=get_string1(ASO,0x10, maxi,0x8A);
    if (tramaCorrecta==1)
        {paso=1;}
    else{
        }
    pasolnt=31;
    break;
default:
    break;
}
}
//-----Sensores SHARP-----
void Sensores(){
    set_adc_channel(0);
    delay_us(20);
    qa=read_adc();

```

```

    s1=Distancia(qa);
//-----
    set_adc_channel(1);
    delay_us(20);
    qa=read_adc();
    s2=Distancia(qa);
//-----
    set_adc_channel(2);
    delay_us(20);
    qa=read_adc();
    s3=Distancia(qa);
    obstaculo=0;
//-----
    if(( s1<60 || s2<60) || s3<60)
    {
        obstaculo=1;
    }
    conrecta=1;
    if (obstaculoAnt==1)
    {
        conrecta=0;
        r1=0;
    }
}
//-----Detener el robot-----
void motor_stop()
{ set_pwm1_duty(0);
  set_pwm2_duty(0);
  //output_b(0b00000000);
  //delay_ms(500)
  iT0=180;
  //eT0=0;
  iT0d=176;
  //eT0d=0;
  LED_ON(LED4);
  c=0;
}
//-----Control PID-----
void Odometria_PID(){
while(mover==1)
{
    ci=get_timer0();
    cd=get_timer1();
    set_timer0(0);
    set_timer1(0);
    delay_ms(25);
}
}

```

```

fmi=ci*40;
fmd=cd*40;
yT=fmi;
rT=fi;
eT=rT-yT;
iT=Ki*eT+iT0;
dT=Kd*(eT-eT0);
uT=iT+Kp*eT+dT;
if(uT>maximo){
    uT=maximo;}
else if (uT<minI){
    uT=minI;
}
DutyI=uT;
set_pwm1_duty(DutyI);
iT0=iT;
eT0=eT;
//PID Motor derecho
yTd=fmd;
rTd=fd;
eTd=rTd-yTd;
iTd=Ki2*eTd+iT0d;
dTd=Kd2*(eTd-eT0d);
uTd=iTd+Kp2*eTd+dTd;
if(uTd>maximo){
    uTd=maximo;}
else if (uTd<minD){
    uTd=minD;
}
DutyD=uTd;
set_pwm2_duty(DutyD);
iT0d=iTd;
eT0d=eTd;
//-----
void main(){
    output_b(0b00000000);
    TOCON_REG=0b10101000;
    T1CON_REG=0b00000111;
    set_timer0(0);
    set_timer1(0);
    fi=0;
    fd=0;
    DutyI=0;
    DutyD=0;
    minI=10.0;
    minD=10.0;

```

```

maximo=255.0;
Kp =0.0042 ;Ki =0.0041;Kd =0.0024;//Constantes PID para motor izquierdo
Kp2 =0.0042 ;Ki2 =0.0041;Kd2 =0.0024;//Constantes PID para motor derecho
iT0=180;
eT0=0;
iT0d=176;
eT0d=0;
bit_clear(TRISC,1);
bit_clear(TRISC,2);
setup_timer_2(t2_div_by_4,249,1);
setup_ccp1(ccp_pwm);
setup_ccp2(ccp_pwm);
set_pwm1_duty(0);
set_pwm2_duty(0);
LED_OFF(LED3);
LED_OFF(LED4);
//-----
Cmo=(pi*Diametro)/(2*ppr*Reduccion);
Cro=(pi*Diametro)/(Longitud*ppr*Reduccion);
Cr=Cro;
Cm=Cmo;
//-----
setup_adc_ports(AN0_TO_AN2|VSS_VDD);
setup_adc(ADC_CLOCK_INTERNAL);
set_adc_channel(0);

Vinicio=0;
//-----
Xbee=1;
x=-100.11;
y=-111.21;
theta=-333.33;
thetagrados=-333.33;
enviar_paquete();
pasolnt=10;
OrdenInicio=0;
//-----
enable_interrupts(INT_RDA);
enable_interrupts(GLOBAL);
mover=0;
set_timer0(0);
set_timer1(0);
x=0;
y=0;
theta=0.0;
thetagrados=0.0;

```

```

terminator=0;
pasolnt=10;
while(1){
  if (Inicio ==1)
  {
    enviar_paquete();
    while(y< 300){
      if (OrdenInicio==1){
        xdif=x;
        Sensores();//evaluar sensores
        if (evaluarCentro==3)//Modifica entrada difusa x
          {xdif=0;} //evalua obstaculos como si x==0;y Theta==0
        evaluarfis(s1,s2,s3,xdif);//evaluo el FLC
        orient=Res1;
        oradian=(orient*pi)/180;//convierto de grados a radianes
        Vel2=Res2;// convierto de rpm-pps(F=Vrpm*Kppr*reduccion)/60
//-----
        if (evaluarCentro==3){//modifica la salida FLC orientacion
          if ((x<0 && obstaculo==1) || Derecha==1)
          {
            if(oradian>0){
              oradian=-1*oradian;//oradian-0;
            }
          }
          anrotacion=oradian;//oradian-0;
          oradian=oradian+theta;

        }
        else if(evaluarCentro==2)
        {
          anrotacion=oradian-theta;
          if( r1==2){
            contrecta=0;
            oradian=0.0;
            anrotacion=(-1*theta);
            r1=0;
            conteogiro=0;
            conteA=2;
            Evacion=0;
          }
        }
      }
    }
  }
}

```