



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“NAVEGACIÓN AUTÓNOMA DE UN ROBOT MÓVIL DENTRO DE
ENTORNOS REAL/VIRTUAL”

TESIS

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS EN INGENIERÍA ELECTRÓNICA

PRESENTA:
ING. DANTE MÚJICA VARGAS

ASESOR:
DR. FRANCISCO JAVIER GALLEGOS FUNES



MÉXICO, D. F. MAYO DE 2008



INSTITUTO POLITÉCNICO NACIONAL
SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D. F. siendo las 17:00 horas del día 6 del mes de Mayo del 2008 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de la E. S. I. M. E. para examinar la tesis de grado titulada:

"NAVEGACIÓN AUTÓNOMA DE UN ROBOT MÓVIL DENTRO DE ENTORNOS REAL/VIRTUAL"

Presentada por el alumno:

MUJICA

Apellido paterno

VARGAS

Apellido materno

DANTE

Nombre(s)

Con registro: A 0 6 0 4 1 5

Aspirante al grado de:

MAESTRO EN CIENCIAS EN INGENIERIA ELECTRONICA

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de tesis

DR. FRANCISCO JAVIER GALLEGOS FUNES

Segundo Vocal

DR. RAÚL CASTILLO PÉREZ

Secretario

DR. JOSÉ MANUEL DE LA ROSA VÁZQUEZ

Presidente

DR. ALEXANDRE MICHTCHENKO

Tercer Vocal

DR. JESÚS YALJÁ MONTIEL PÉREZ

Suplente



DR. RAÚL PEÑA RIVERO

El Presidente del Colegio

DR. JAIME RÓBLES GARCÍA



**INSTITUTO POLITECNICO NACIONAL
SECRETARIA DE INVESTIGACION Y POSGRADO**

CARTA CESION DE DERECHOS

En la Ciudad de México, D. F., el día 26 del mes Mayo del año 2008

el(la) que suscribe Dante Mújica Vargas alumno (a) del Programa de
Maestría en Ciencias en Ingeniería Electrónica
con número de registro A060415 adscrito a la Sección de Estudios de Posgrado e Investigación de la
E.S.I.M.E. Unidad Zacatenco, manifiesta que es autor(a) intelectual del presente Trabajo de Tesis
bajo la dirección del Dr. Francisco Javier Gallegos Funes y cede los derechos del
trabajo intitulado: Navegación autónoma de un robot móvil dentro de entornos real/virtual al
Instituto Politécnico Nacional para su difusión, con fines Académicos y de Investigación.

Los usuarios de la información no deben reproducir el contenido textual, graficas o datos del trabajo
sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la
siguiente dirección: adorinam@yahoo.com.mx, fgallegosf@ipn.mx

Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente
del mismo.

Nombre y Firma

DANTE MÚJICA VARGAS

Contenido

Índice de figuras	iii
Índice de tablas	vi
Nomenclatura	vii
Resumen	ix
Abstract	x
Capítulo 1 Introducción	1
1.1 Introducción	1
1.2 Objetivo	2
1.3 Justificación	2
1.4 Contenido de la tesis	2
1.5 Referencias	3
Capítulo 2 Estado del arte	4
2.1 Introducción	4
2.2 Fundamentos de realidad virtual	4
2.2.1 <i>Concepto de realidad virtual</i>	5
2.2.2 <i>Modelo de un sistema de realidad virtual</i>	5
2.2.3 <i>Tipos de realidad virtual</i>	6
2.2.4 <i>Aplicaciones de la realidad virtual</i>	8
2.3 Fundamentos de Robótica móvil	9
2.3.1 <i>Concepto de robótica móvil</i>	10
2.3.2 <i>Configuración de los robots móviles</i>	10
2.3.3 <i>Odometría de un robot móvil con configuración diferencial</i>	15
2.3.4 <i>Sensores y actuadores utilizados en la robótica móvil</i>	17
2.3.5 <i>Algoritmo para control de motores</i>	20
2.3.6 <i>Controladores inteligentes empleados en la robótica móvil</i>	22
2.4 Conclusiones	24
2.5 Referencias	24
Capítulo 3 Propuesta	26
3.1 Introducción	26
3.2 Creación de la realidad virtual	27

3.3 Manipulación de la realidad virtual empleando el MatLab	29
3.4 Arquitectura del robot móvil	30
3.5 Dimensiones del robot móvil	31
3.6 Electrónica empleada	32
3.6.1 <i>Electrónica de control / Transmisor de radio frecuencia</i>	33
3.6.2 <i>Receptor de radio frecuencia</i>	34
3.7 Descripción y programación del controlador difuso	36
3.8 Incorporación del robot móvil dentro del entorno virtual	42
3.9 Conclusiones	44
3.10 Referencias	44
Capítulo 4 Resultados experimentales	45

4.1 Introducción	45
4.2 Caracterización de los actuadores y sintonización del controlador	45
4.3 Caracterización de los sensores	47
4.4 Comportamiento del robot para la recuperación de su trayectoria	51
4.5 Comportamiento del robot para evadir objetos durante su navegación y recuperar su trayectoria	54
4.6 Discusión de resultados	58
4.7 Conclusiones	59
4.8 Referencias	59

Capítulo 5 Conclusiones

5.1 Conclusiones generales	60
5.2 Recomendaciones y trabajo futuro.	61

Anexos

A Diagrama de estados-eventos de la interfaz gráfica de usuario.	62
B Diagrama de flujo del programa del módulo de control y transmisión.	63
C Diagrama de flujo del programa del módulo de recepción.	66

Índice de figuras

Figura 2.1	Modelo genérico de un ambiente virtual.	6
Figura 2.2	Accesorios empleados en la realidad virtual inmersiva.	6
Figura 2.3	Realidad virtual semi-inmersiva.	7
Figura 2.4	Realidad virtual no inmersiva.	7
Figura 2.5	Sistema de telepresencia.	7
Figura 2.6	Aplicación de realidad virtual en las Ciencias de la Tierra.	8
Figura 2.7	Aplicación de realidad virtual en la Medicina.	9
Figura 2.8	Aplicación de realidad virtual en el arte.	9
Figura 2.9	Robots móviles.	10
Figura 2.10	Robot móvil con configuración de triciclo.	11
Figura 2.11	Conducción y rotación de un robot con configuración de triciclo.	11
Figura 2.12	Robot con conducción síncrona.	12
Figura 2.13	Direccionamiento de un robot síncrono.	12
Figura 2.14	Robots móviles con configuración diferencial.	12
Figura 2.15	Conducción y rotación de un robot con configuración diferencial.	13
Figura 2.16	Robot con orugas.	13
Figura 2.17	Robot con direccionamiento de Ackermann.	13
Figura 2.18	Llantas omni-direccionales.	14
Figura 2.19	Robot omni-direccional con tres llantas.	14
Figura 2.20	Robot omni-direccional con cuatro llantas.	14
Figura 2.21	Movimientos básicos de un robot omni-direccional.	15
Figura 2.22	Distribución de los motores del robot utilizado.	15
Figura 2.23	Trayectoria recorrida por un robot con configuración diferencial.	16
Figura 2.24	Posición del robot con configuración diferencial.	17
Figura 2.25	Sensor ultrasónico.	18
Figura 2.26	Tipos de encoders. a) Encoder incremental, b) Encoder de cuadratura.	18
Figura 2.27	Tipos de motores. a) Corriente directa, b) A pasos, c) Servomotor	19
Figura 2.28	Control de servomotores.	20
Figura 2.29	Lazo de un controlador PID.	21
Figura 2.30	Controlador PI fusionado.	21
Figura 2.31	Diagrama a bloques de un controlador difuso para un robot.	23
Figura 3.1	Diagrama a bloques del sistema propuesto.	26
Figura 3.2	Bloques requeridos para crear mundos virtuales.	27

Figura 3.3	Instalaciones de SEPI-Electrónica.	27
Figura 3.4	Creación de patrones.	28
Figura 3.5	Estructura básica del entorno virtual.	28
Figura 3.6	Pasillo principal del entorno virtual.	28
Figura 3.7	Pasillo baño.	29
Figura 3.8	Pasillo salón.	29
Figura 3.9	Pasillo interior.	29
Figura 3.10	Interfaz gráfica de usuario.	30
Figura 3.11	Arquitectura del robot móvil.	31
Figura 3.12	Robot móvil.	31
Figura 3.13	Dimensiones del robot móvil.	32
Figura 3.14	Contenido de la trama de información.	32
Figura 3.15	Diagrama esquemático del módulo de control.	34
Figura 3.16	Círculo impreso del módulo de control.	34
Figura 3.17	Diagrama esquemático del módulo receptor.	35
Figura 3.18	Círculo impreso del módulo receptor.	35
Figura 3.19	Controlador difuso propuesto.	36
Figura 3.20	Conjuntos de la entrada Sensor-Derecho.	37
Figura 3.21	Conjuntos de la entrada Sensor-Centro.	37
Figura 3.22	Conjuntos de la entrada Sensor-Izquierdo.	37
Figura 3.23	Conjuntos de la entrada Posición-Meta.	37
Figura 3.24	Conjuntos de la salida Rotación.	40
Figura 3.25	Conjuntos de la salida Velocidad.	40
Figura 3.26	Centroides de la salida Rotación.	41
Figura 3.27	Centroides de la salida Velocidad	41
Figura 3.28	Resultados obtenidos con MatLab	42
Figura 3.29	Resultados obtenidos con Lenguaje C.	42
Figura 3.30	Perspectivas del robot virtual. a) Vista frontal, b) Vista posterior, c) Vista lateral derecha, d) Vista superior.	43
Figura 3.31	Sistema virtual.	43
Figura 4.1	Curvas de transferencia de los motores.	46
Figura 4.2	Distribución de los motores.	46
Figura 4.3	Desplazamientos del robot. a) Desplazamiento hacia delante, b) Desplazamiento hacia atrás.	49
Figura 4.4	Respuesta del controlador PI.	47
Figura 4.5	Detección de diferentes formas geométricas. a) Detección de un cilindro, b) Detección de un cubo, c) Detección de una arista del cubo, d) Detección de un objeto diminuto.	49
Figura 4.6	Detección de objetos mediante el Sensor 1. a) Objeto: cilindro, b) Objeto: rectángulo (caja).	49
Figura 4.7	Detección de objetos mediante el Sensor 2. a) Objeto: cilindro, b) Objeto: rectángulo (caja).	49

Figura 4.8	Detección de objetos mediante el Sensor 3. a) Objeto: cilindro, b) Objeto: rectángulo (caja).	50
Figura 4.9	Determinación del cono de medición real.	50
Figura 4.10	Conos de medición reales de los sensores. a) Medición real del sensor 1, b) Medición real del sensor 2, c) Medición real del sensor 3.	51
Figura 4.11	Distribución de los sensores de ultrasónicos.	51
Figura 4.12	Comportamiento del robot para recuperar su trayectoria y alcanzar la meta con salida en el origen. a) Trayectorias ideal y recorrida para alcanzar la meta con el robot puesto en el origen, b) Posición final del robot dentro del entorno virtual, c) Posición final del robot dentro del entorno real.	52
Figura 4.13	Comportamiento del robot para recuperar su trayectoria y alcanzar la meta con salida a -0.92 metros del origen. a) Trayectorias ideal y recorrida para alcanzar la meta con el robot puesto a -0.92 metros del origen, b) Posición final del robot dentro del entorno virtual, c) Posición final del robot dentro del entorno real.	53
Figura 4.14	Comportamiento del robot para recuperar su trayectoria y alcanzar la meta con salida a 0.92 metros del origen. a) Trayectorias ideal y recorrida para alcanzar la meta con el robot puesto a 0.92 metros del origen, b) Posición final del robot dentro del entorno virtual, c) Posición final del robot dentro del entorno real.	54
Figura 4.15	Posicionamiento de objetos en la evasión y recuperación de trayectoria. a) Primer prueba, b) Segunda prueba, c) Tercer prueba.	55
Figura 4.16	Comportamiento del robot para evadir objetos y recuperar su trayectoria, en la primer prueba. a) Trayectorias ideal y recorrida para evadir objetos y recuperar su trayectoria en la primer prueba, b) Posición final del robot dentro del entorno virtual, c) Posición final del robot dentro del entorno real.	56
Figura 4.17	Comportamiento del robot para evadir objetos y recuperar su trayectoria, en la segunda prueba. a) Trayectorias ideal y recorrida para evadir objetos y recuperar su trayectoria en la segunda prueba, b) Posición final del robot dentro del entorno virtual, c) Posición final del robot dentro del entorno real.	57
Figura 4.18	Comportamiento del robot para evadir objetos y recuperar su trayectoria, en la tercer prueba. a) Trayectorias ideal y recorrida para evadir objetos y recuperar su trayectoria en la tercer prueba, b) Posición final del robot dentro del entorno virtual, c) Posición final del robot dentro del entorno real.	57
Figura A-1	Diagrama de estados-eventos de la interfaz gráfica de usuario.	62
Figura B-1	Diagrama de flujo del programa del módulo de control y transmisión. a) Rutina principal, b) Rutinas auxiliares, c) Rutinas de las interrupciones.	65
Figura C-1	Diagrama de flujo del programa del módulo de recepción. a) Rutina principal, b) Rutina auxiliar, c) Rutina de la interrupción.	68

Índice de tablas

Tabla 2.1	Movimientos básicos de un robot con guiado omni-direccional con cuatro llantas.	14
Tabla 3.1	Descripción de los botones de la interfaz gráfica de usuario.	30
Tabla 3.2	Reglas del controlador lógico difuso.	38
Tabla 4.1	Constantes del controlador PI.	47
Tabla 4.2	Dimensiones de los objetos.	48
Tabla 4.3	Configuración de los sensores.	48
Tabla 4.4	Resultados obtenidos.	58
Tabla 4.5	Repetibilidad de las pruebas realizadas.	58

Nomenclatura

ANFIS	Sistema Neuro-Fuzzy de inferencia adaptiva.
GPS	Sistema de posicionamiento global.
GUI	Interfaz gráfica de usuario.
GUIDE	Entorno para desarrollar interfaces gráficas de usuario.
I ² C	Interfaz serial para circuitos inter-integrados.
LCD	Pantalla de cristal líquido.
MIPS	Millones de instrucciones por segundo.
PCB	Tarjeta de circuito impreso.
PI	Proporcional Integral.
PID	Proporcional Integral Diferencial.
PWM	Modulación de ancho del pulso.
RS-232	Estándar recomendado 232.
RV / VR	Realidad virtual.
TTL	Lógica transistor transistor.
TWI	Interfaz serial de dos cables.
USART	Transmisor y receptor serial síncrono y asíncrono.
D _D	Desplazamiento del motor derecho.
D _I	Desplazamiento del motor izquierdo.
DR	Diámetro de las llantas.
D _{Robot}	Desplazamiento del robot.
Error _{MD}	Error del motor derecho.
Error _{MI}	Error del motor izquierdo.
I _{SUM}	Variable auxiliar.
Kd	Constante de derivación.
Ki	Constante de integración.
Ki _{MD}	Constante de integración para ambos motores.
Kp	Constante de proporcionalidad.
Kp _{MD}	Constante de proporcionalidad del motor derecho.
Kp _{MI}	Constante de proporcionalidad del motor izquierdo.
L	Separación de las llantas.
V _C	Variable de control.
V _{c_{MD}}	Variable de control del motor derecho.
V _{c_{MI}}	Variable de control del motor izquierdo.

V_D	Velocidad deseada.
V_P	Variable del proceso.
V_{PMD}	Variable del proceso del motor derecho.
V_{PMI}	Variable del proceso del motor izquierdo.
$u(t)$	Entradas del controlador difuso.
x	Posición en el eje X.
x_0	Posición inicial en el eje X.
$x(t)$	Salidas del controlador difuso.
y	Posición en el eje Y.
y_0	Posición inicial en el eje Y.
y_{salida}	Variable de salida del controlador difuso.
$y(t)$	Salidas del robot.
y_k	Centroide de cada conjunto difuso de la salida.
θ	Orientación del robot.
θ_0	Orientación inicial del robot.
$\mu(x)$	Grado de pertenencia a cada uno de los conjuntos de las entradas.
$\mu(y)$	Grado de pertenencia a cada uno de los conjuntos de las salidas.

Resumen

Este trabajo describe el diseño e implementación de un controlador lógico difuso para dotar de inteligencia y autonomía a un robot móvil. El controlador difuso, el cual se basa en la evasión de objetos y recuperación de trayectoria, posee un mecanismo para combinar información proveniente de sensores ultrasónicos con expresiones odométricas para estimar la posición del robot, así como coordinar el comportamiento del robot para alcanzar una meta. El robot móvil opera en una configuración diferencial.

El seguimiento a la toma de decisiones y acciones de control del robot durante su navegación, se hace de forma remota mediante realidad virtual no inmersiva, creada a partir del entorno real en el que el robot se encuentra operando. Con tal fin se utiliza una interfaz gráfica de usuario, la cual permite adquirir la información de la posición del robot a través de radio frecuencia, y actualizar su posición dentro de la realidad virtual.

Abstract

This work describes the design and implementation of logic fuzzy controller to provide of intelligence and autonomy to a mobile robot. The fuzzy controller, which is based on avoiding obstacles and recovering path, it has a mechanism to combine information from ultrasonic sensors and odometry expressions to estimate the robot position, as well as to coordinate the robot behavior to achieve a goal. The mobile robot operates in a differential configuration.

The follow-up to decision-making and actions to control the robot during its navigation is done remotely by non-immersive virtual reality, created from real-world environment in which the robot is operating. To this end uses a graphical user interface, which allows acquiring information from the position of the robot via radio frequency, and update its position within the virtual reality.

Capítulo 1

Introducción

1.1 Introducción

La definición de un robot móvil autónomo es ambigua y abierta aun al debate. Es difícil definir qué es exactamente un móvil autónomo, porque los términos usados no están aun delimitados. En este trabajo se define a un móvil autónomo, como un robot que puede navegar inteligentemente y por sí mismo dentro de cualquier ambiente sin la intervención humana [1].

Cuando escuchamos la palabra robot, diferentes imágenes se presentan en nuestra mente, muchas de las cuales están influenciadas por la industria del cine. Por ejemplo, en películas como Star Wars, I robot y Terminator, puede apreciarse que los robots juegan un papel muy importante y son altamente sofisticados, a tal grado que muchos de ellos tienen cualidades humanas. Sin embargo, todos estos ejemplos son una pobre representación de los robots que existen y operan en la actualidad.

Dada esa errada concepción de un robot, es necesario definirlo dentro del contexto de nuestra realidad. El término robot describirá a una máquina que tiene cierto nivel de inteligencia, por lo tanto, puede interpretar la información presente en sus entradas (sensores) y es capaz de responder a éstas en la forma más conveniente [2]. Para este trabajo, un robot recopilará información del ambiente en el que se encuentra a partir de sus sensores, la procesará y entonces responderá a ésta por medio del control de sus elementos actuadores. En base a esto, para un observador externo, el robot parecerá ser inteligente, pues es capaz de tomar decisiones a partir de las situaciones que se presenten en su entorno.

El término entorno también puede ser definido. Con un mundo demasiado complejo, no podemos recrear cada evento o situación que se suscita en él. Sin embargo, creando un modelo simple del mundo alrededor del robot, esperamos tener las situaciones más significativas que se pueden presentar. Los robots han sido diseñados para operar en diferentes entornos, por ejemplo tierra, aire, mar y espacio exterior. En este trabajo delimitamos el entorno del robot a un ambiente interior, por lo que éste se moverá en una superficie plana por lo que sus movimientos estarán restringidos a dos dimensiones.

1.2 Objetivo

Construcción de un sistema móvil capaz de tomar decisiones y acciones de control cuando estas sean requeridas en su navegación.

1.3 Justificación

Desarrollo de un sistema móvil con inteligencia para ser usado en diferentes aplicaciones de visión por computadora, por ejemplo en inspección, seguridad o entretenimiento.

1.4 Contenido de la tesis

El Capítulo 2 presenta en primera instancia los fundamentos relacionados con la realidad virtual. Para ello se conceptualiza este término, se explica cómo se encuentra constituido un sistema de realidad virtual, se clasifican estos sistemas en función directa con el hardware empleado y se mencionan aplicaciones en diferentes áreas. Posteriormente se expone el concepto de robótica móvil, se profundiza en la configuración de robots móviles sobre ruedas, se establecen las expresiones matemáticas para determinar la posición relativa para un robot con configuración diferencial, y finalmente se describe el hardware de uso común en la robótica móvil y algunas técnicas para dotar de inteligencia artificial a un robot móvil.

El Capítulo 3 desglosa el sistema propuesto. Primeramente describe la secuencia gráfica para desarrollar el entorno virtual, así como la interfaz gráfica de usuario diseñada para manipular dicho entorno. Despues se presenta grosso modo la arquitectura empleada en el robot móvil, así como la electrónica empleada. Posteriormente se detalla el diseño e implementación del controlador lógico difuso utilizado. Por último se indica la forma en que el robot móvil se incorporó al entorno virtual.

El capítulo 4 muestra los resultados obtenidos durante el proceso de caracterización de los elementos sensores y actuadores. También describe las pruebas de navegación realizadas para cada uno de los comportamientos propuestos, así como la fusión de estos.

El Capítulo 5 presenta las conclusiones obtenidas durante el desarrollo de este trabajo y proporciona algunas recomendaciones para lograr mejores resultados.

Finalmente se presentan los anexos con los diagramas de flujo de los programas empleados, así como el diagrama de estados-eventos que describe la ejecución de la interfaz gráfica de usuario. También, se presentan los videos de las pruebas realizadas.

1.5 Referencias

- [1] Hyungsuk Cho. "Intelligent Mobile Robots: The state of art and perspectives". Department of Mechanical Engineering. Korea Advanced Institute of Science and Technology. Daejeon Korea. 2004.
- [2] "Introducción a la Robótica basada en comportamientos". Departamento de Computación. Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires. www.exa.unicen.ar/catedras/irobotic.

Capítulo 2

Estado del Arte

2.1 Introducción

Este capítulo presenta los fundamentos relacionados con la realidad virtual, para ello se conceptualiza este término, se explica cómo se encuentra constituido un sistema de realidad virtual, se clasifican estos sistemas en función directa con el hardware empleado y se mencionan aplicaciones en diferentes áreas. Posteriormente se expone el concepto de robótica móvil, se profundiza en la configuración de robots móviles sobre ruedas, se establecen las expresiones matemáticas para determinar la posición relativa para un robot con configuración diferencial, y finalmente se describen el hardware de uso común en la robótica móvil y algunas técnicas para dotar de inteligencia artificial a un robot móvil.

2.2 Fundamentos de la Realidad Virtual

El avance de la informática en los últimos años propició la creación de un nuevo término, **Realidad Virtual**. Iniciado en los programas de entrenamiento militares, simuladores de vuelo, centros de investigación y académicos, pasando a los programas más variados en el ámbito profesional y doméstico. Hoy se empieza a aplicar en ingeniería, arquitectura, educación, juegos, entre otros campos. Cuál será el futuro de la realidad virtual, es aún una interrogante con bastantes aristas, dado lo impredecible que es acotar una disciplina incipiente, que depende de los avances que se pueden obtener tanto en software como en hardware, así como de nuevas aplicaciones inexploradas. La realidad virtual, evidentemente, se encuentra en su estado inicial y sus posibles aplicaciones pueden ser tan diversas como la creatividad y necesidad del hombre permita.

El concepto agrupa ideas aparentemente opuestas: “*realidad*” y “*virtual*”, después de muchos recorridos y discusiones semánticas, el término realidad parece establecido en su sentido positivo, es “*real*” aquello que tiene existencia verdadera y efectiva, en cuanto al término “*virtual*” define todo aquello que tiene la virtud de producir un efecto, también señala con frecuencia como la aposición a lo real. La conjunción de estas dos palabras trae consigo una paradoja, pues define a la *realidad virtual* como “*realidad no real*” [1].

2.2.1 Concepto de Realidad Virtual

Definir el término Realidad Virtual (RV, o simplemente VR, del inglés *Virtual Reality*) es difícil. Existen tantas definiciones como investigadores haya, dado su reciente y rápida evolución no ha permitido establecerse una definición clara. De tal forma que no resulta extraño que la realidad virtual resulte ser relativa para diferentes personas y en diferentes situaciones. Más que “definirla” se citarán un conjunto de concepciones donde cada una entrega una idea de lo que debería entenderse por Realidad Virtual [2]:

- **Realidad Virtual** es la experiencia de telepresencia, siendo ésta la sensación de presencia utilizando un medio de comunicación.
- **Realidad Virtual** es un paso más allá de lo que sería la simulación por computadora, tratándose más bien de una simulación interactiva, dinámica y en tiempo real de un sistema.
- **Realidad Virtual** es una simulación por computadora, dinámica y tridimensional, con alto contenido gráfico, acústico y táctil, orientado a la visualización de situaciones complejas, durante la cual el usuario ingresa, a través del uso de dispositivos de entrada sofisticados, a mundos que aparentan ser reales, resultando inmerso en ambientes altamente participativos de origen artificial.
- **Realidad Virtual** es una simulación tridimensional interactiva por computadora en la cual un usuario se siente introducido en un ambiente artificial, y lo que percibe como real está basado en estímulos de sus órganos sensoriales.
- **Realidad Virtual** es el medio que proporciona una visualización participativa en tres dimensiones y la simulación de mundos virtuales, siendo dichos mundos el elemento fundamental de un sistema de realidad virtual. La realidad virtual es un entorno generado por computadora en el que los usuarios pueden entrar físicamente e interactuar con él, desplazándose en su interior o modificándolo de cualquier forma. En su forma más simple, un mundo virtual podría estar compuesto por un edificio tridimensional por el que podríamos desplazarnos, aunque sin modificar nada. Sin embargo, con los accesorios adecuados, el usuario podría ver, desplazarse e interactuar con estos entornos gráficos.

Aunque el término realidad virtual es el más empleado existen otras expresiones para referirse al mismo concepto, tal es el caso de *Realidad Sintética*, *Mundos Virtuales* y *Ciberespacio*, palabras diferentes con un mismo significado e idénticas indefiniciones.

2.2.2 Modelo de un sistema de realidad virtual

Las partes básicas de la realidad virtual son: el modelo de simulación, la representación del ambiente virtual, la entrada/salida y por supuesto el usuario [3]. La Figura 2.1 presenta el modelo genérico de un ambiente virtual.

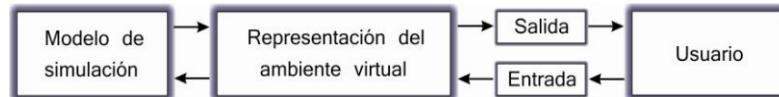


Figura 2.1. Modelo genérico de un ambiente virtual.

- **Modelo de simulación.** Es la representación matemática del sistema que se está visualizando. Un modelo virtual necesita responder dinámicamente en respuesta de la entrada del usuario. Se pueden crear modelos matemáticos sofisticados de la realidad virtual, pero lo importante es la manera en que estos modelos están asociados con un sistema visual y auditivo.
- **Dispositivos de Entrada/Salida.** A través de los dispositivos de entrada y salida de datos, se facilita la comunicación hombre-máquina. Con los dispositivos de entrada, el usuario puede transmitir sus órdenes al sistema de realidad virtual, indicándole que desea desplazarse, cambiar el punto de vista e interactuar con algún objeto del mundo virtual, y los dispositivos de salida permiten que el usuario se sienta inmerso dentro del mundo creado.
- **Usuario.** El usuario es la razón de existir de un sistema de realidad virtual, pues es él quien recibe los estímulos de parte del sistema y a su vez se encarga de retroalimentarlo y definir su comportamiento.

2.2.3 Tipos de realidad virtual

Existen diferentes tipos de realidad virtual, los cuales se describen a continuación [3]:

- **Sistemas inmersivos** son aquellos donde el usuario se siente dentro del mundo virtual que está explorando. En este tipo de sistemas se utilizan diferentes dispositivos denominados accesorios, como pueden ser guantes, trajes especiales, visores o cascos (ver Figura 2.2), estos últimos le permiten al usuario visualizar los mundos a través de ellos, haciéndolo sentirse inmerso dentro de estos mundos. Estos sistemas son ideales para aplicaciones de entrenamiento o capacitación.



Figura 2.2. Accesorios empleados en la realidad virtual inmersiva.

- **Sistemas semi-inmersivos o inmersivos de proyección** se caracterizan por ser cuatro pantallas en forma de cubo, tres pantallas forman las paredes como se muestra en la Figura 2.3, las cuales rodean al observador, el

usuario usa lentes y dispositivos de seguimiento de movimientos de la cabeza, de esta manera al moverse el usuario las proyecciones perspectivas son calculadas por el motor de RV para cada pared y se despliegan en proyectores que están conectados a la computadora. Este tipo de sistemas son usados principalmente para visualizaciones donde se requiere que el usuario se mantenga en contacto con elementos del mundo real.



Figura 2.3. Realidad virtual semi-inmersiva. Figura 2.4. Realidad virtual no inmersiva.

- **Sistemas no inmersivos o de escritorio** (ver Figura 2.4) son aquellos donde el monitor es el portal de entrada hacia el mundo virtual y la interacción es por medio del teclado, micrófono, mouse o joystick. Este tipo de sistemas son idóneos para visualizaciones científicas, también son usados como medio de entretenimiento (como es el caso de los juegos de arcade¹) y aunque no ofrecen una inmersión total son una buena alternativa.
- **Sistemas de Telepresencia** utilizan generalmente elementos como cámaras, micrófonos, dispositivos táctiles y de fuerza como elementos de retroalimentación, ligados a elementos de control remoto para permitir al usuario manipular robots u otros dispositivos distantes en forma virtual. La Figura 2.5 muestra un sistema de telepresencia.

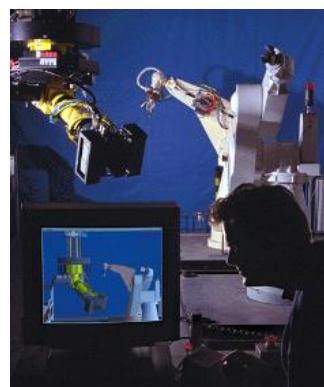


Figura 2.5. Sistema de telepresencia.

¹ Juegos de arcade/arcade: Grandes máquinas disponibles en los sitios comerciales. Cuando es referido como género agrupa a los juegos con gráficos poco complicados donde la dificultad aumentaba progresivamente con cada pantalla superada (Pac-Man), por ende no hay profundidad en cuanto a historia, mitología, personalidad en los personajes.

2.2.4 Aplicaciones de la realidad virtual

En un principio la realidad virtual fue usada principalmente en aplicaciones militares o incluso de entretenimiento, sin embargo, en los últimos años se ha diversificado su uso, por ende, en lugar de describirlas todas a detalle sólo se mencionarán las más destacadas [2].

Realidad virtual en la Ingeniería

Dentro de las aplicaciones hay proyectos de manipulación remota de robots (ver Figura 2.5), o procesos de ensamblado, también existen áreas dedicadas al desarrollo de prototipos virtuales. Todas estas aplicaciones facilitan la automatización dentro de diferentes áreas.

Realidad virtual en las Ciencias de la Tierra

Dentro de las Ciencias de la Tierra se realizan proyectos para la visualización de los fenómenos volcánicos o la modelación de relieves topográficos, dichas aplicaciones se presentan en la Figura 2.6.

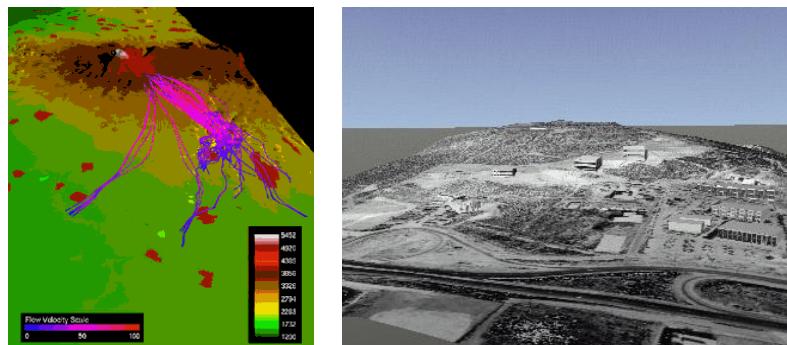


Figura 2.6. Aplicación de realidad virtual en las Ciencias de la Tierra.

Realidad virtual en la Medicina

La medicina es uno de los campos más importantes para las aplicaciones de la realidad virtual. Así como la medicina cuenta con una diversidad de áreas de estudio, la realidad virtual ofrece implementaciones para cada una de éstas. Por ejemplo, en la Psicología permite el tratamiento de fobias que van desde el miedo a las alturas, a volar, a las arañas, claustrofobia, etc. Con la incorporación de modelos virtuales (elevadores, simuladores de vuelo, entre otros) se incrementa la seguridad o confianza del paciente, puesto que él sabe que mientras se pruebe este tipo de modelos nunca se encontrará realmente en peligro, y tendrá la capacidad de detenerlo cuando lo deseé. Una de las aplicaciones donde más beneficio representa es sin duda en la simulación de cirugías virtuales (ver Figura 2.7), la idea es proveer al cirujano de una herramienta que le permita experimentar procedimientos quirúrgicos en un ambiente artificial, también este tipo de aplicaciones se pueden utilizar como entrenadores para estudiantes de medicina.

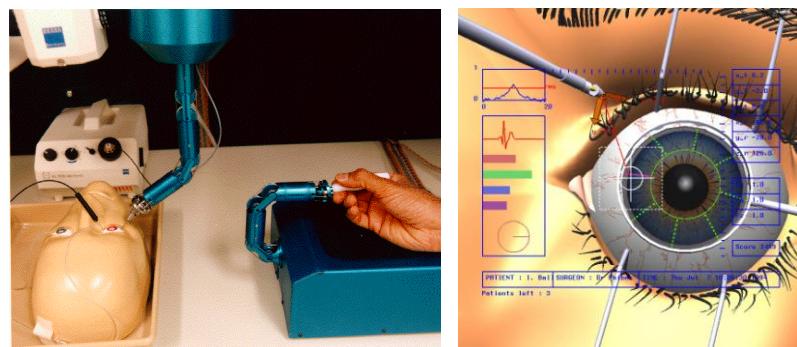


Figura 2.7. Aplicación de realidad virtual en la Medicina.

Realidad virtual en el arte

El mundo del arte es y siempre ha sido importante para el ser humano y hoy en día la computación tiene influencia en casi cualquier área de trabajo o de interés, la realidad virtual se ha convertido en una herramienta de trabajo para el campo de las artes. La realidad virtual juega un papel importante para el conocimiento, es utilizada por museos (como la que se muestra en la Figura 2.8), planetarios y centros de ciencia. Estos centros realizan exposiciones virtuales donde se puede hacer recorridos en templos antiguos, palacios, galaxias, aprender de diversas áreas del conocimiento, entre otras.



Figura 2.8. Aplicación de realidad virtual en el arte.

2.3 Fundamentos de Robótica móvil

La robótica móvil es un campo que se encuentra en estado de rápida expansión en el mundo académico e industrial. Con el paso del tiempo los robots móviles han llegado a ser más inteligentes y capaces de realizar tareas en forma autónoma, haciendo posible su incorporación en lugares distantes e inaccesibles, como los que se muestran en la Figura 2.9 [4].



Figura 2.9. Robots móviles.

Un concepto que va de la mano con la robótica móvil es la llamada teleoperación robótica [5]. Esta rama constituye una tecnología idónea para aplicaciones que se desarrollan en entornos peligrosos u hostiles a los que una persona no puede accesar directamente. Una ayuda inestimable en la mejora de resultados obtenidos en tareas de este tipo es la realidad virtual, aunque se mencionó con anterioridad la forma en que estas dos tecnologías se interrelacionan, un aspecto importante que debe considerarse y es motivo para que estas áreas se unan, es el problema de retardo o pérdida en la transferencia de información que fluye desde el entorno remoto hacia el local, esta situación puede presentarse cuando un robot está provisto de cámaras como instrumentos para explorar el entorno en estudio. Bajo esta condición resulta ser más práctico generar un entorno de navegación basado en modelos tridimensionales o virtuales.

2.3.1 Concepto de Robótica móvil

La robótica móvil es un área de intensa investigación cuyo objetivo es el desarrollo de robots móviles autónomos que puedan desenvolverse en ambientes estructurados (estáticos) y no estructurados (dinámicos) [6].

2.3.2 Configuración de robots móviles

Aunque existen diferentes tipos de robots móviles (bípedos, hexápodos, acuáticos, entre otros), este punto sólo se enfocará sobre 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 [7].

Triciclo básico 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 lo menos tres puntos de contacto para estabilizar una plataforma [8]. Dicha configuración se presenta en la Figura 2.10.



Figura 2.10. Robot móvil con configuración de triciclo.

La velocidad lineal y angular de este tipo de robots están completamente desacopladas, así que para impulsarlo en línea recta, la llanta frontal debe estar posicionada en el centro y debe ser impulsado a la velocidad deseada. La Figura 2.11 muestra el guiado para algunas configuraciones de direccionamiento. Este tipo de diseño no tiene la capacidad de girar sobre su centro de gravedad, con la llanta delantera puesta a 90° el robot girará cerca del punto medio entre las dos llantas posteriores, así que el radio mínimo de giro es la distancia entre la llanta delantera y el punto medio de las llantas de atrás.

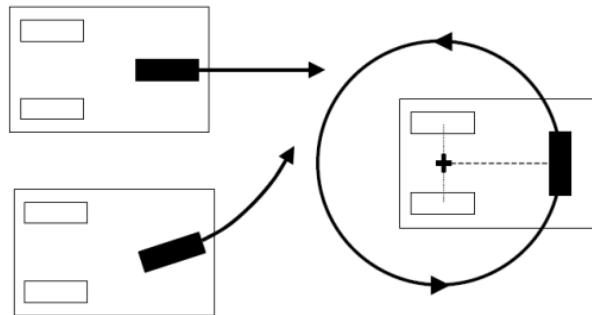


Figura 2.11. Conducción y rotación de un robot con configuración de triciclo.

Una extensión de los robots configurados como triciclo son los de *conducción sincronizada* (ver Figura 2.12), sin embargo aquí se tienen tres llantas las cuales permiten el desplazamiento y controlan la orientación del robot, todas las llantas se rotan de tal forma que apunten al mismo punto (ver Figura 2.13), esto se logra usando un motor y una cadena para controlar la orientación y otro motor para controlar el desplazamiento. Un robot de este tipo puede ser considerado como *Omnidireccional* debido a que tiene la capacidad de ser orientado en cualquier dirección, sin embargo, debe detenerse y realinear sus llantas cuando va desplazándose en línea recta y se quiere hacer rotar, puesto que no puede desplazarse y girar al mismo tiempo.



Figura 2.12. Robot con conducción síncrona.

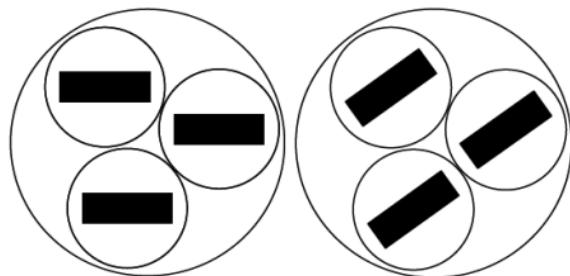


Figura 2.13. Direccionamiento de un robot síncrono.

Guiado diferencial con llantas y orugas

Este diseño tiene dos motores fijos, uno sobre el lado derecho y otro en el izquierdo del robot y ambos tienen la capacidad de impulsarlo [8]. Puesto que siempre es necesario contar con tres puntos de soporte, este diseño requiere una o dos llantas pasivas adicionales, dependiendo de la localización de los motores (ver Figura 2.14). 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 del desplazamiento es más complejo que la configuración de triciclo, debido a que se requiere siempre de la coordinación de ambos motores.



Figura 2.14. Robots móviles con configuración diferencial.

La configuración con una sola rueda pasiva no puede disponer de una llanta con motor en el centro del robot 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 posterior permite que el robot gire sobre su centro de gravedad. Sin embargo, este diseño puede introducir problemas de direccionamiento debido a que el contacto con la superficie tiene cuatro puntos de soporte. La Figura 2.15 muestra el desplazamiento del robot en función de la velocidad de sus motores, si ambos motores tienen la misma velocidad, el robot se desplazará en línea recta ya sea hacia adelante o atrás, si uno de los motores tiene una mayor velocidad que el otro, el robot se desplazará formando un arco, si ambos motores giran a la misma velocidad pero en dirección opuesta, el robot girará sobre su centro de gravedad.

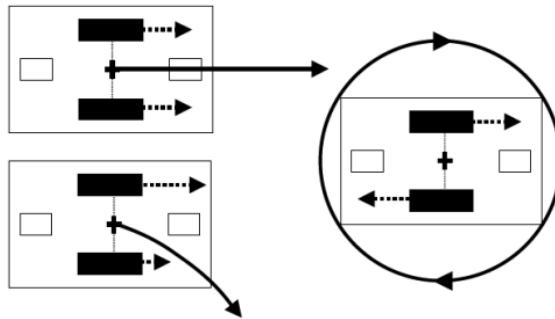


Figura 2.15. Conducción y rotación de un robot con configuración diferencial.

Un caso especial de robots con guiado diferencial son los que emplean orugas en lugar de llantas, la única diferencia que presentan, es una mejor maniobrabilidad en terrenos rugosos y alta fricción al girar, debido a sus múltiples puntos de contacto con la superficie. La Figura 2.16 muestra un robot de este tipo.

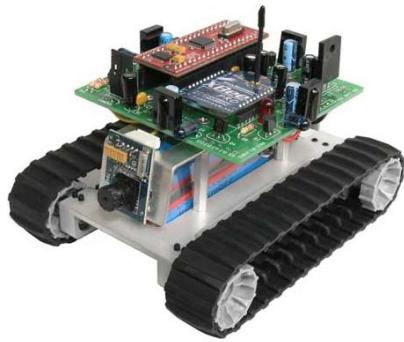


Figura 2.16. Robot con orugas.

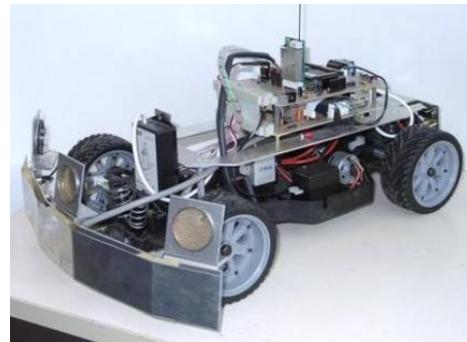


Figura 2.17. Robot con direccionamiento de Ackermann.

Direccionamiento de Ackermann

El sistema de locomoción de un automóvil, consiste de dos llantas combinadas en la parte trasera que permiten impulsar y dos llantas también combinadas en la parte delantera cuya función es dar orientación al móvil. Esto es conocido como direccionamiento de Ackermann (ver Figura 2.17) [8]. El desplazamiento en línea recta no presenta ningún problema dado que las llantas de atrás son impulsadas por un mismo motor, sin embargo éstas presentan deslizamiento en las curvas, y la mayor limitante que tiene un robot de este tipo es que no puede girar sobre su centro de gravedad.

Guiado Omni Direccional

Hasta el momento las plataformas móviles presentan la misma deficiencia, no poder desplazarse en cualquier dirección, en contraste a esto, el robot con guiado omni-direccional es capaz de moverse en todas direcciones, ésta propiedad está relacionada con el tipo de llantas (ver Figura 2.18) que utiliza [8]. Dichas llantas están cubiertas con

cilindros que rotan libremente, es importante indicar que las llantas son manipuladas por los motores, pero los cilindros no. Un robot de este tipo puede ser construido ya sea con tres o cuatro llantas, sin embargo la orientación de los cilindros en las llantas difiere, en el caso de un robot con tres llantas (ver Figura 2.19) los cilindros deben estar a 90° con respecto al eje de la llanta, en cambio en un modelo con cuatro llantas (ver Figura 2.20) se requiere que los cilindros estén a 45° .

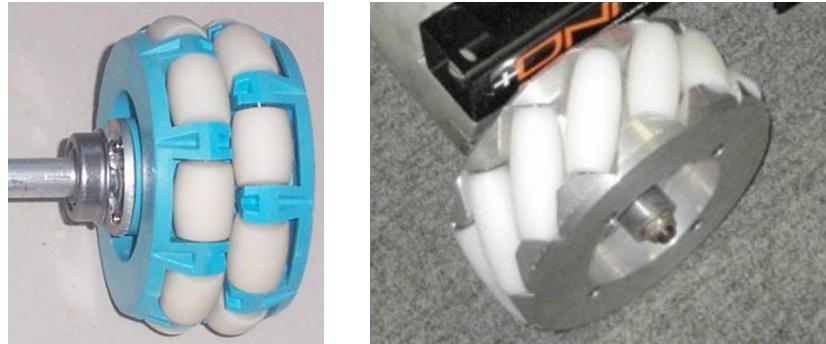


Figura 2.18. Llantas omni-direccionales.



Figura 2.19. Robot omni-direccional con tres llantas.

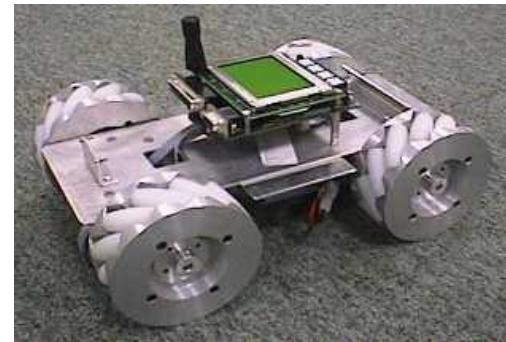


Figura 2.20. Robot omni-direccional con cuatro llantas.

La Tabla 2.1 muestra los movimientos básicos que rigen la cinemática de un robot con cuatro llantas, es decir, los desplazamientos hacia adelante, atrás, hacia un lado y rotación sobre su centro de gravedad [8]. La Figura 2.21 ilustra dichos movimientos de acuerdo a la Tabla 2.1.

Tabla 2.1. Movimientos básicos de un robot con guiado omni-direccional con cuatro llantas.

Desplazamiento	Movimiento de motores
Hacia delante	Todos los motores hacia adelante
Hacia atrás	Todos los motores hacia atrás
Deslizarse hacia la izquierda	1,4: atrás y 2,3: adelante
Deslizarse hacia la derecha	1,4: adelante y 2,3: atrás
Girar con las manecillas del reloj	1,3: adelante y 2,4: atrás
Girar contra las manecillas del reloj	1,3: atrás y 2,4: adelante

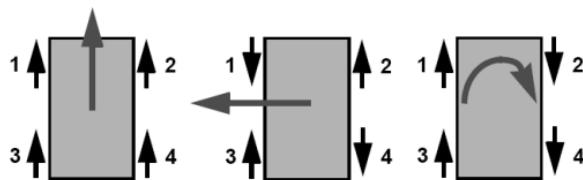


Figura 2.21. Movimientos básicos de un robot omni-direccional.

2.3.3. **Odometría de un robot móvil con configuración diferencial**

La odometría es uno de los métodos más usados para estimar la posición relativa de un robot móvil con respecto a un punto de partida conocido, se basa en ecuaciones simples que pueden implementarse fácilmente y utilizan la información generada por encoders colocados generalmente en los motores de los robots [9]. Dependiendo de la configuración de los robots móviles estas expresiones pueden variar considerablemente.

El robot empleado en éste trabajo tiene como base dos motores (Figura 2.22). Como puede apreciarse, hay un motor en cada lado de la plataforma, permitiendo impulsar y orientar al robot independientemente uno con respecto al otro. Considerando lo anterior y la información presentada en 2.2.2 la cinemática de este móvil obedece a una configuración con guiado diferencial.

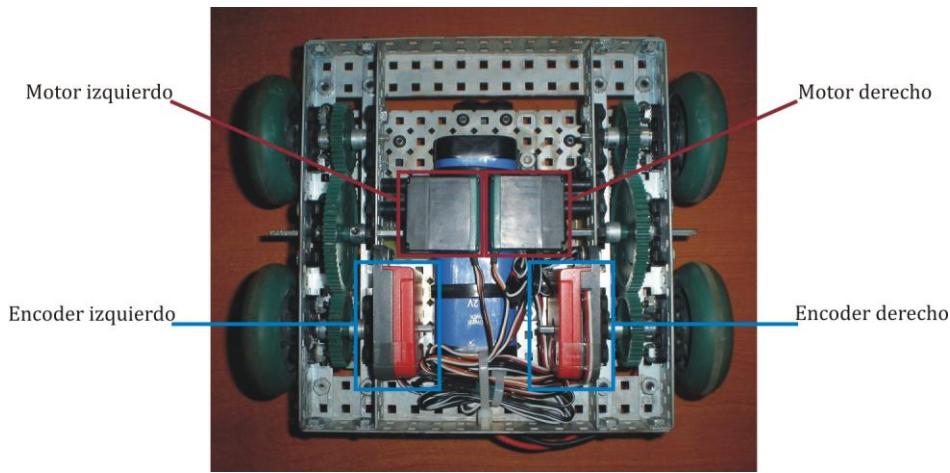


Figura 2.22. Distribución de los motores del robot utilizado.

Por lo tanto el análisis matemático que a continuación se presenta permite conocer el desplazamiento y determinar la posición de un robot de este tipo. Considerando la distancia que el robot viaja (ver Figura 2.23), el primer paso requerido es conocer el desplazamiento del robot. En términos numéricos estos datos son calculados con las siguientes expresiones [10].

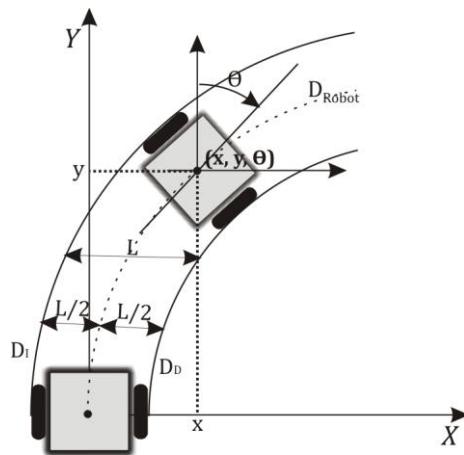


Figura 2.23. Trayectoria recorrida por un robot con configuración diferencial.

$$D_D = D_I = \pi \cdot DR \cdot \text{Número de revoluciones} \quad (2.1)$$

$$D_{\text{Robot}} = \frac{D_I + D_D}{2} \quad (\text{metros}) \quad (2.2)$$

$$\text{Número de revoluciones} = \frac{\text{Número de pulsos contados por el encoder}}{\text{Números de pulsos por revolución}} \quad (2.3)$$

donde D_{Robot} es el desplazamiento del robot, D_D es el desplazamiento del motor derecho, D_I es el desplazamiento del motor izquierdo, DR es el diámetro de las llantas y L es la separación entre las llantas.

Tomando como referencia el desplazamiento del robot, el desplazamiento de cada llanta puede expresarse de la siguiente forma,

$$D_I = \theta \left(r + \frac{L}{2} \right) \quad (2.4)$$

$$D_D = \theta \left(r - \frac{L}{2} \right) \quad (2.5)$$

Restando ambas ecuaciones se elimina la variable r ,

$$D_I - D_D = \theta \cdot L \quad (2.6)$$

Despejando θ podemos conocer la orientación del robot en radianes.

$$\theta = \frac{D_I - D_D}{L} + \theta_0 \quad (\text{radianes}) \quad (2.7)$$

Utilizando funciones trigonométricas y el ángulo de orientación, la posición x e y (ver Figura 2.24) se puede determinar de la siguiente forma,

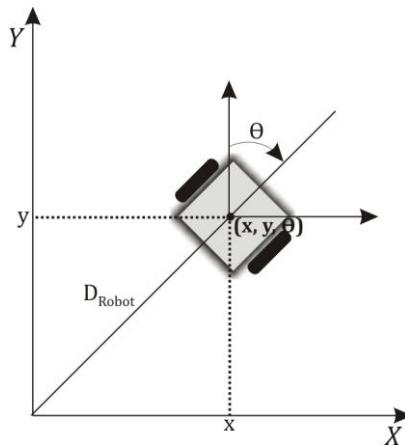


Figura 2.24. Posición del robot con configuración diferencial.

$$x = \left(\frac{D_I + D_D}{2} \right) \operatorname{sen}(\theta) + x_0 \quad (\text{metros}) \quad (2.8)$$

$$y = \left(\frac{D_I + D_D}{2} \right) \cos(\theta) + y_0 \quad (\text{metros}) \quad (2.9)$$

2.3.4 Sensores y actuadores utilizados en la robótica móvil

Los seres vivos cuentan con una serie de sentidos que les sirven para obtener información del ambiente y su sistema de navegación utiliza la información obtenida de estos sentidos para poder conocer y desplazarse en el mundo. Al igual que los seres vivos un robot autónomo debe contar con sentidos que le permitan conocer su entorno, para ello debe estar provisto de una serie de sensores cuya información será utilizada por el sistema de control y por el algoritmo de navegación para definir la ruta a seguir. Entre los sensores más usados se tienen: sensores ópticos (fotorresistencias, infrarrojos), sensores de ultrasonido, visión artificial basada en una cámara (detección de contornos, colores).

Sensor de Ultrasonido

Este tipo de sensores son muy empleados para detección de obstáculos, dado que son capaces de dar la información de la distancia a la que estos se encuentran. Estos sensores utilizan las propiedades de las ondas de sonido, a partir de su velocidad y el tiempo transcurrido entre la emisión de una señal acústica (a una frecuencia ultrasónica de 50 a 250 KHz) y su recepción, se calcula la distancia [11]. Por lo general se usa un transductor como emisor y otro como receptor como se muestra en la Figura 2.25, aunque también se puede usar un solo transductor

para ambos propósitos, para este último caso tras la emisión del ultrasonido se espera un determinado tiempo a que las vibraciones del sensor desaparezcan y esté preparado para recibir el eco producido por el obstáculo.

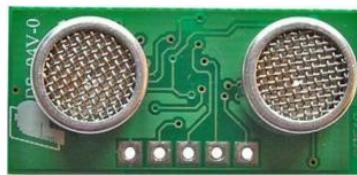


Figura 2.25. Sensor ultrasónico.

Otra necesidad presente en la navegación autónoma de los robots es determinar su posición a cada instante. Se puede conocer la posición relativa con respecto a un punto de referencia (por lo general es el punto de partida), o la posición absoluta del móvil. Para ello se debe equipar al robot con sensores como encoders, brújulas electrónicas (compás magnético), acelerómetros, giroscopios y GPS.

Encoders

Cuando se usan motores a pasos para el desplazamiento de un robot se pueden contar las vueltas a partir de los pasos que da cada llanta y con ellos se puede determinar la distancia recorrida por el robot, sin embargo muchos robots utilizan motores de cd ya que éstos consumen menos potencia, para medir la distancia recorrida por cada llanta se utilizan los encoders. Hay básicamente dos tipos de sensores, los cuales se muestran en la Figura 2.25 [8]:

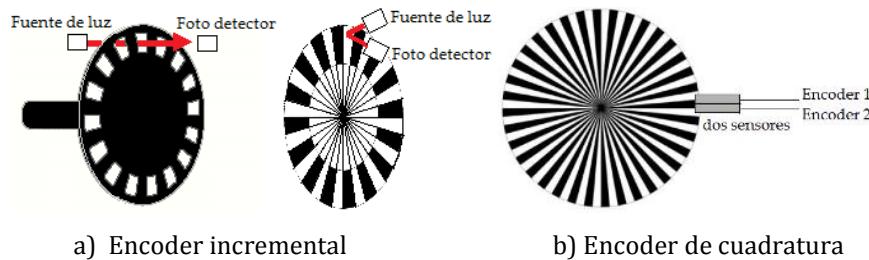
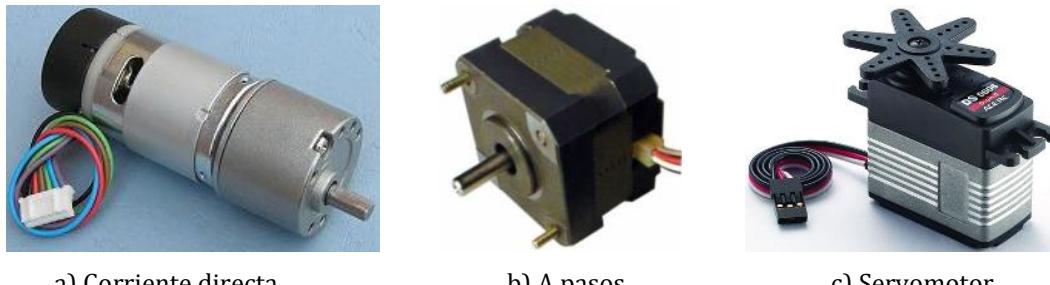


Figura 2.26. Tipos de encoders.

- Incrementales: Permiten contar el número de ranuras por las que pasa el haz o las marcas blancas en que éste se refleja (ver Figura 2.25a), a partir de ello podemos conocer la velocidad o el desplazamiento de las llantas.
- Cuadratura: Pueden hacer lo mismo que los sensores incrementales, pero tienen la capacidad de determinar el sentido de giro de un motor. Para ello están provistos de otro sensor (ver Figura 2.25b), de tal forma que si el Encoder 1 recibe primero la señal el movimiento es en sentido de las manecillas del reloj, en cambio si Encoder 2 recibe primero la señal el movimiento será en contra de las manecillas del reloj.

Los actuadores son los músculos de un robot, es decir son los elementos que convierten la energía de alimentación en movimiento. En la robótica móvil existen básicamente tres tipos de motores de uso común (mostrados en la Figura 2.27): Los motores de corriente directa, los motores a pasos y los servomotores.



a) Corriente directa

b) A pasos

c) Servomotor

Figura 2.27. Tipos de motores.

La selección de un motor obedece al tipo de robot que se pretende desarrollar. Por ejemplo, un motor de corriente directa puede ser utilizado para cualquier diseño, aunque es ideal para robots grandes. Éste presenta los inconvenientes de ser demasiado rápido (por lo tanto requiere un tren de engranaje reductor) y un control complejo (PWM – Modulación de Ancho del Pulso). Un motor a pasos es recomendado para robots muy ligeros, en general seguidores de líneas o solucionadores de laberintos, estos motores ofrecen un control preciso de velocidad, sus limitantes son el gran tamaño para la baja potencia que proporcionan, demandan demasiada corriente y necesitan un control muy complejo. Un servomotor es usado principalmente en robots pequeños o articulados, básicamente son un motor de cd que cuenta con un tren de engranaje dentro de un encapsulado, en general son los más versátiles para ser implementados físicamente y ser manipulados mediante un controlador que genere una señal PWM, no obstante, el control de la velocidad y su capacidad para soportar un bajo peso son sus principales inconvenientes.

Servo motores

Un servo motor (también llamado servo) es un dispositivo que tiene la capacidad de ser ubicado en cualquier posición dentro de su rango de operación y mantenerse en dicha posición. Conformado por un motor de corriente directa muy pequeño, un tren de engranaje reductor que permite incrementar su torque, una tarjeta controladora que convierte la señal PWM recibida en movimiento del eje de salida y un potenciómetro que mide la posición de salida del eje. La tarjeta controladora dispone de retroalimentación y monitorea constantemente la información generada por el resistor variable, de tal forma que permite posicionar y mantener el eje del motor [12]. La señal PWM utilizada para manipular los servos tiene una frecuencia de 50 Hz, así que los pulsos son generados cada 20 ms. El ancho del pulso especifica la posición deseada en el eje del motor (Figura 2.28). Por ejemplo, una señal con un ancho de 1.25 ms llevará el motor a 0°, una de 1.5 ms lo pondrá a 90° y con una de 1.75 ms hará que el motor este en 180°.

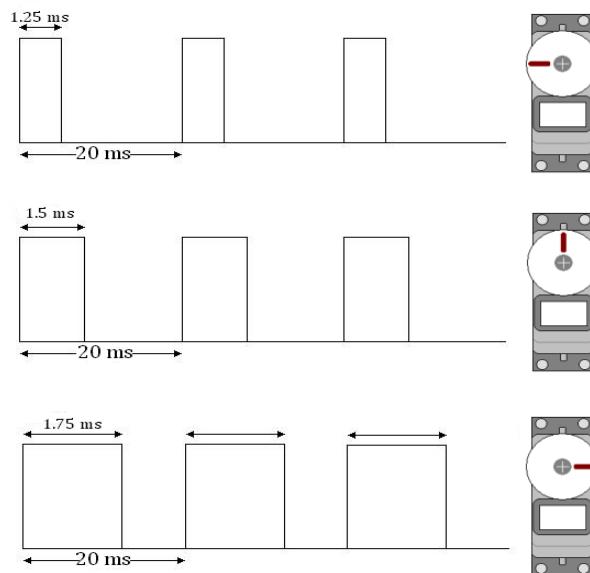


Figura 2.28. Control de servomotores.

Servomotores modificados (Analógicos/Digitales)

Es posible modificar un servomotor para eliminar su restricción de giro y permitirle una rotación continua (el proceso para llevar a cabo esta operación puede encontrarse en la literatura) [13], sin embargo, es posible adquirir servos modificados por el fabricante. Estos dispositivos alterados permiten incrementar la velocidad de rotación del motor dependiendo del pulso en su entrada PWM. Al aplicar un pulso de 1.5 ms hará que el motor se detenga. Un pulso menor a 1.5 ms provocará que el motor gire en sentido contrario a las manecillas del reloj, y cuando el pulso es mayor a 1.5 ms girará con las manecillas del reloj. Los servomotores modificados pueden ser analógicos o digitales. En el caso de los analógicos, éstos disponen de electrónica analógica y ofrecen la ventaja de poderse calibrar (llevándolo al punto de reposo) utilizando un potenciómetro. En el caso de los digitales, cuentan con un circuito integrado el cual ya no permite hacer modificaciones, sin embargo no requieren de calibración puesto que el ajuste es preciso.

2.3.5 Algoritmo para control de motores

Un algoritmo de control debe ser capaz de tomar una velocidad definida, aplicar potencia al motor para llevarlo rápidamente hasta la velocidad requerida y mantenerla a pesar de factores externos e inesperados que podrían incrementarla o decrementarla. Aunque existe una gran variedad de algoritmos, en aplicaciones de robótica móvil se utiliza el control PID (Proporcional Integral Derivativo), así como sus variantes. Un lazo de un controlador de este tipo es ilustrado en la figura 2.29.

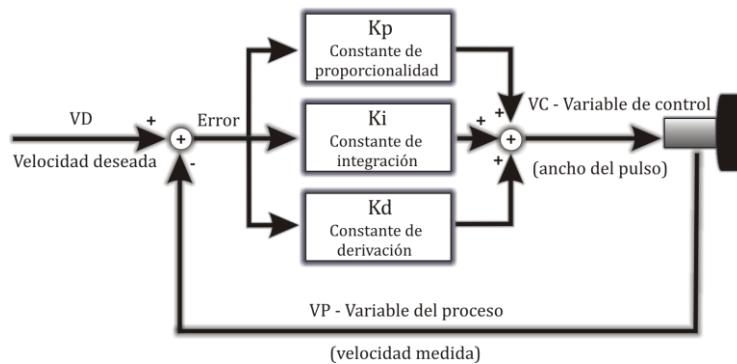


Figura 2.29. Lazo de un controlador PID.

Para controlar la velocidad de un motor mediante un lazo PID, como primer paso se debe establecer la velocidad requerida, después debe medirse la velocidad actual del motor, con estos datos puede determinarse el error, para ello debe sustraerse la velocidad medida de la velocidad deseada, si el error no es cero, éste debe ser transformado en un valor válido para la variable de control que cambiará la salida del proceso a un valor más cercano a la velocidad deseada. La transformación del error se hace simultáneamente mediante un módulo proporcional (multiplicando al error por la constante de proporcionalidad, haciendo que el error sea mínimo), un módulo integral (multiplicando al error por la constante de integración, con el propósito de disminuir y eliminar el error en estado estacionario provocado por módulo proporcional) y un módulo derivativo (multiplicando al error por la constante de derivación, evitando que el error se incremente). Las salidas de estos tres módulos son sumadas y asignadas a la variable de control, generando una nueva velocidad del motor.

Considerando que un robot con configuración diferencial está basado en dos motores, es necesario emplear dos lazos de control. Sin embargo, para tal caso existe un modelo [12] que modifica los lazos a controladores PI (Proporcional Integral) y los une (Figura 2.30). Este método tiene la capacidad de mejorar el desplazamiento de un robot permitiendo hacer un recorrido en línea recta más preciso y compensar el error que puede existir en los giros.

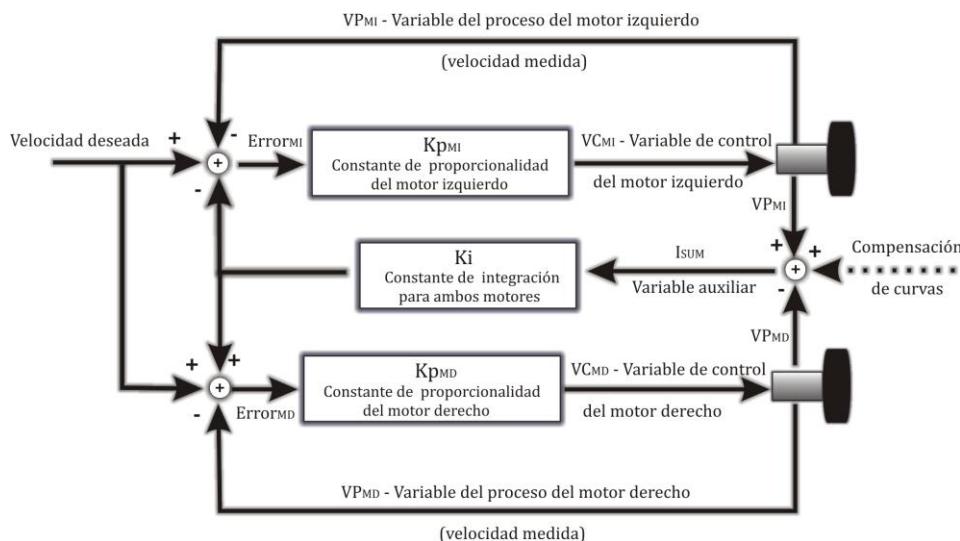


Figura 2.30. Controlador PI fusionado.

Como puede apreciarse, hay dos lazos de control separados, uno para manipular el motor derecho y otro para el izquierdo, cada uno de los cuales involucra retroalimentación vía un módulo proporcional. La diferencia de movimiento que puede presentarse entre los motores es regresada a ambos lazos de control a través de un módulo integral. Éste módulo integra la diferencia de velocidades, la cual posteriormente es eliminada por los módulos proporcionales. Para compensar los efectos que pueden presentarse durante la rotación del robot se incluye el término compensación de curvas, éste permite incrementar o decrementar el efecto del módulo integral sobre los lazos de control. Sin embargo, la incorporación de ésta entrada trae consigo que éste término mitigue el efecto del módulo integral (si hay una mala elección de su valor), presentándose los siguientes efectos: requerir más tiempo durante la rotación del robot o cancelar totalmente al módulo integral (haciendo que el robot vibre durante la rotación).

En otras palabras, el funcionamiento de éste controlador hace lo siguiente: cuando los dos motores están corriendo a la misma velocidad ningún efecto ocurre. Cuando el motor izquierdo es más lento más potencia le es aplicada, al mismo tiempo que al motor derecho se le suministra menor potencia. Si la velocidad del motor izquierdo no puede ser incrementada, no existe ningún problema, porque la velocidad del motor derecho será minimizada. Pasa lo mismo en el caso contrario, más potencia es aplicada al motor derecho mientras es reducida para el motor izquierdo.

2.3.6 Controladores inteligentes empleados en la robótica móvil

La navegación autónoma de robots móviles en entornos no estructurados constituye uno de los retos más importante en el campo de la robótica móvil, quizás por esto se han propuesto muchas estrategias para solucionar este problema [14]. Con el paso del tiempo se han desarrollado una gran variedad de controladores inteligentes, los más usados son los siguientes:

Redes neuronales

Las redes neuronales artificiales son estructuras computacionales que emulan a las redes neuronales naturales [13]. Frecuentemente son usadas para problemas de clasificación o toma de decisiones que no tienen una solución algorítmica simple o directa. La virtud de las redes neuronales es su habilidad para generar la salida deseada en función de una entrada y un conjunto de casos de entrenamiento sin necesidad de programación explícita, siendo capaz de generalizar casos no vistos previamente. Una red neuronal está formada por un número de unidades individuales llamadas neuronas que están ligadas unas con otras vía conexiones. Cada neurona tiene un número de entradas, un nodo de procesamiento y una salida, mientras cada conexión entre neuronas está asociada con un peso. Cada neurona constantemente evalúa sus entradas, calcula su valor local de activación acorde a una fórmula (umbral) y produce un valor de salida [7].

Algoritmos genéticos

Es una técnica basada en los principio de la teoría de Darwin de la evolución para progresar hacia una solución. Operan iterativamente evolucionando una solución desde una historia de soluciones potenciales, las cuales son manipuladas por un número de operaciones inspiradas biológicamente. Aunque sólo es una aproximación biológica real del proceso evolutivo, está demostrado que es un poderoso y robusto medio para resolver problemas [7].

Lógica difusa

La lógica difusa provee una metodología formal para representar, manipular e implementar el conocimiento heurístico humano acerca de cómo controlar un sistema [15].

Como puede apreciarse en el diagrama a bloques para el control de un robot (Figura 2.32), un controlador difuso opera como un sistema de lazo cerrado en tiempo real. Para ello toma la información presente en sus entradas $u(t)$ (en éste caso las mediciones realizadas por los sensores del robot $y(t)$), la analiza, toma las decisiones de control pertinentes y genera nuevos datos $x(t)$ para configurar al robot (velocidad, orientación, etc.)

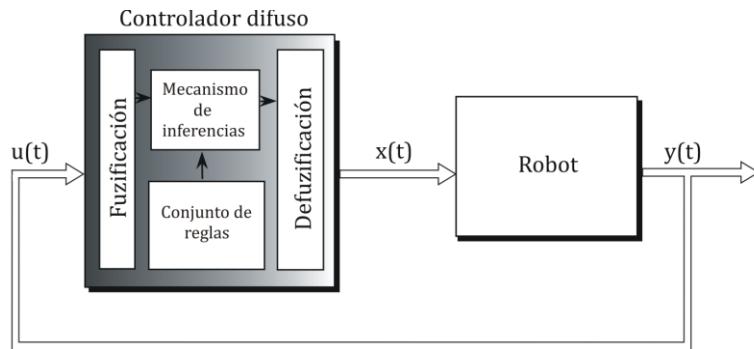


Figura 2.31. Diagrama a bloques de un controlador difuso para un robot.

Puede observarse que un controlador difuso está constituido por cuatro módulos. La función de cada módulo es descrita a continuación:

- **Fuzzificación** convierte las entradas $u(t)$ del controlador a información que el mecanismo de inferencia pueda usar fácilmente para activar y aplicar reglas.
- Un conjunto de **reglas Sí-Entonces** las cuales contienen la descripción lingüística para lograr un buen control.
- **Mecanismo de inferencias** lleva a cabo la toma de decisiones, para ello toma la información actual en las entradas y selecciona una o más reglas que sean las apropiadas para el control del robot.
- **Defuzzificación** convierte las señales de salida $x(t)$ en información de entrada para el robot.

2.4 Conclusiones

Este capítulo presentó un panorama básico sobre dos temas en concreto, realidad virtual y robótica móvil. Información requerida para conocer y entender sobre las capacidades de dichas tecnologías. En el caso de la realidad virtual, se mostró cómo ha comenzado a ser una herramienta útil en diferentes áreas de la ciencia y cómo ha comenzado a ser parte de nuestra vida cotidiana. En cuanto a la robótica móvil se explicó por qué ha sido motivo de interés en el ámbito académico e industrial y porque el afán de hacer robots más inteligentes y autónomos. Partiendo de este búsquedas, se describieron algunas formas de locomoción, así como sus capacidades y limitantes relacionadas directamente con los sensores y actuadores empleados. Al final se citaron algunas de las técnicas de inteligencia artificial usadas para dotar de autonomía a un robot móvil.

2.5 Referencias

- [1] María Jesús Ledesma Carbayo. "Introducción a la realidad virtual". Ingeniería Neurosensorial. Departamento de Electrónica. Universidad Politécnica de Madrid. www.lorien.die.upm.es/isn/docs/VR0304.pdf.
- [2] Juan Carlos Parra Márquez, Rodrigo García Alvarado e Iván Santelices Malfanti. "Introducción práctica a la realidad virtual". Departamento de Computación. Universidad del Bío-Bío, Concepción Chile. 2001.
- [3] Erika Corrado Padilla, Julián J. Delgado y Salvador Castañeda. "Tecnologías de realidad virtual: Modelo edificio inteligente". Departamento de Cómputo Dirección Telemática. Centro de Investigación Científica y Educación Superior de Ensenada. Baja California México. 2001.
- [4] Hyungsuk Cho. "Intelligent Mobile Robots: The state of art and perspectives". Department of Mechanical Engineering. Korea Advanced Institute of Science and Technology. Daejeon Korea. 2004.
- [5] Caprari Gilles. "Autonomous Micro-Robots: Applications and limitations". École Polytechnique Fédérale de Lausanne, Switzerland, 2003.
- [6] Cristian Ignacio García Barrales. "Monte Carlo aplicado a la auto-localización de robots". Departamento de Ingeniería en Sistemas Computacionales. Universidad de las Américas Puebla. 2005.
- [7] Thomas Bräunl. "Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems". School of Electrical, Electronic and Computer Engineering. University of Western Australia. 2006.
- [8] Omar Sánchez Pérez. "Cinemática de Robots Móviles". Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática. Universidad de Huelva. España. 2007.
- [9] David P. Anderson. "IMU Odometry". 2006. www.geology.smu.edu/~dpa-www/robo/Encoder/limu_odo.
- [10] G. W. Lucas. "A tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators". 2001. www.rossum.sourceforge.net/papers/DiffSteer.
- [11] Andrés Felipe Orozco Duque. "Sensores, métodos y técnicas para la navegación de robots móviles autónomos". Control Digital – Grupo de Investigación en Automática. Universidad Pontificia Bolivariana. Medellín Colombia. 2003.

- [12] Dennis Clark and Michael Owings. "Building Robot Drive Trains". TAB Robotics. 2003.
- [13] Seattle Robotics Society. <http://www.seattlerobotics.org/guide/servohack.html>
- [14] Francisco Blanes, Ginés Benet. "Herramientas CASD para diseño de controladores inteligentes en robótica móvil". Department d'Informática de Sistemes i Computadors. Universidad Politécnica de Valencia. 2002.
- [15] Kevin M. Pasino, Stephen Yurkovich. "Fuzzy Control". Addison-Wesley. 1998.

Capítulo 3

Sistema Propuesto

3.1 Introducción

Este capítulo desglosa el sistema propuesto. Primeramente describe la secuencia gráfica para desarrollar el entorno virtual, así como la interfaz gráfica de usuario diseñado para manipular dicho entorno. Después se presenta grosso modo la arquitectura empleada en el robot móvil, así como la electrónica empleada. Posteriormente se detalla sobre el diseño e implementación del controlador lógico difuso utilizado. Por último se presenta la forma en que el robot móvil se incorporó al entorno virtual.

El sistema propuesto está constituido por dos bloques, los cuales están relacionados tal como lo ilustra la Figura 3.1. El robot móvil navega dentro de un entorno real y conocido, para hacerlo está provisto de un controlador difuso que se encarga de tomar decisiones y acciones de control cuando diferentes situaciones se van presentando durante la navegación, para ello utiliza la información presente en sus sensores. A medida que el robot se va desplazando, la información de su posición y las mediciones de los sensores ultrasónicos son enviadas vía radio frecuencia hacia el bloque de realidad virtual. Este segundo bloque opera como un medio remoto para dar seguimiento al comportamiento del robot dentro del entorno real. La información recibida es presentada al usuario mediante la interfaz gráfica. Sin embargo, para que el usuario experimente la sensación de presencia (telepresencia) dentro del entorno real, la información de la posición del robot es utilizada para actualizar la posición del robot virtual dentro del entorno virtual (ambos creados a partir del robot y entorno real).

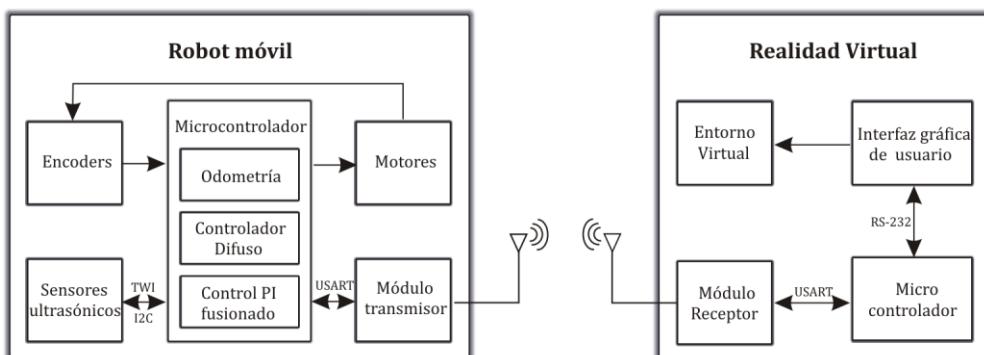


Figura 3.1. Diagrama a bloques del sistema propuesto.

3.2 Creación de la Realidad Virtual

El entorno virtual diseñado es del tipo no inmersivo (el portal de es por medio del monitor) y pasivo (no se crean objetos durante su ejecución). Para el diseño y construcción se utilizó V-Realm Builder 2.0 [1], éste es el editor predeterminado del Toolbox de Realidad Virtual que MatLab 7.4 [2] emplea. La Figura 3.2 muestra la interacción de los bloques requeridos en la creación de un mundo virtual.

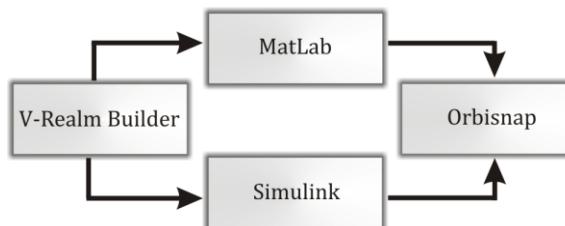


Figura 3.2. Bloques requeridos para crear mundos virtuales.

El entorno real utilizado como modelo para crear el mundo virtual son las instalaciones de SEPI-Electrónica (ver Figura 3.3), así que se tomaron algunas fotos, se obtuvieron las dimensiones de los objetos existentes y se midieron el ancho y largo de los pasillos. De tal forma que el mundo virtual tuviera las mismas dimensiones del mundo real. Es necesario indicar que la unidad dimensional que el editor maneja corresponde a un metro de la realidad.



Figura 3.3. Instalaciones de SEPI-Electrónica.

Tras el proceso de familiarización se observó que el editor V-Realm Builder permite manipular cuatro formas geométricas básicas (cuadrados, conos, cilindros y esferas), así como sus atributos, es decir las dimensiones, posición, textura, entre otros. Para crear un objeto existente en el mundo real es necesario relacionar estas formas y agruparlas. Esto permite generar pequeños modelos que pueden ser utilizados posteriormente como patrones.

Como puede apreciarse en la Figura 3.3, existen cuatro objetos que se repiten continuamente (pilares, puertas, paredes con y sin ventana). Considerando esto se crearon los modelos de dichos objetos (ver Figura 3.4) y se establecieron como patrones para dar forma al mundo virtual.

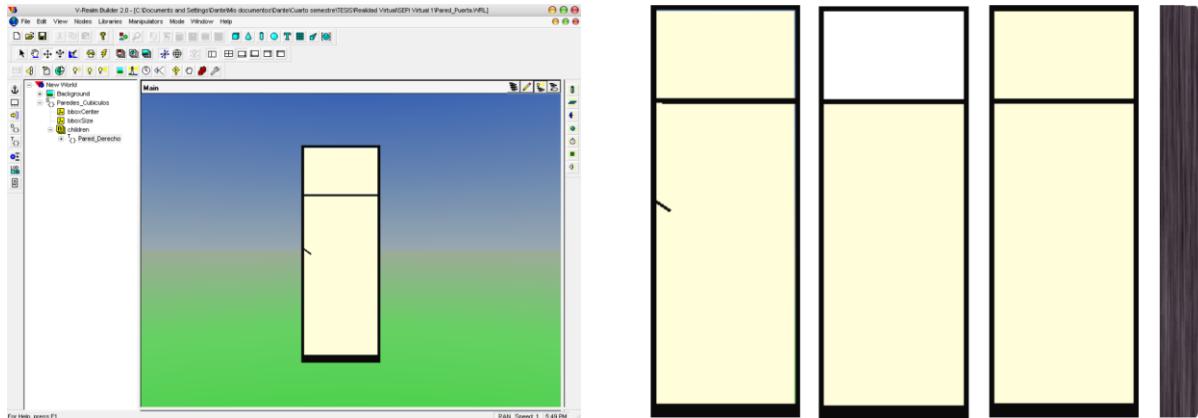


Figura 3.4. Creación de patrones.

Con estos objetos se inició la creación del entorno virtual, comenzándose con el esqueleto de las instalaciones (ver Figura 3.5), para ello se hizo la distribución del piso, techo, paredes y pilares. Posteriormente se incorporaron y colocaron los patrones de las paredes y puertas tal como se muestra en la Figura 3.6, creándose el pasillo principal.

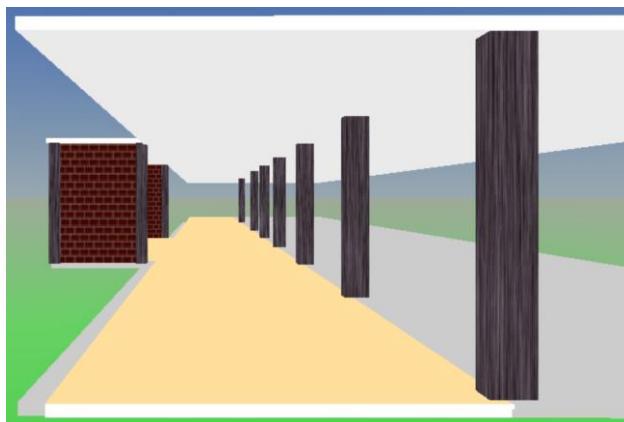


Figura 3.5. Estructura básica del entorno virtual.



Figura 3.6. Pasillo principal del entorno virtual.

La siguiente etapa fue la creación de los pequeños pasillos, primero se crearon nuevos objetos (persianas y puerta metálica) y se dio forma al pasillo cercano al sanitario (ver Figura 3.7), después se creó una puerta especial para la entrada a los laboratorios de la sección y con esto se estructuró el segundo pasillo (ver Figura 3.8).



Figura 3.7. Pasillo baño.



Figura 3.8. Pasillo salón.

Finalmente se trabajó con el pasillo interior de las instalaciones, la Figura 3.9 presenta la distribución de esta parte.

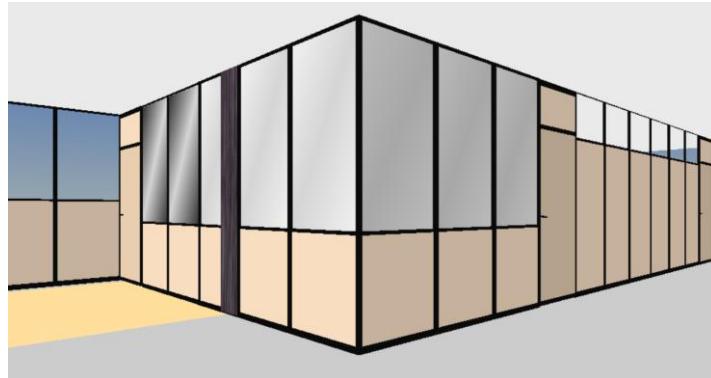


Figura 3.9. Pasillo interior.

3.3 Manipulación de la realidad virtual empleando MatLab

Utilizando el ambiente de programación GUI / GUIDE [3] se diseñó y codificó una interfaz gráfica de usuario para manipular la realidad virtual y el flujo de información (posición y mediciones de los sensores) proveniente del robot móvil (ver anexo A). La adquisición de esta información se realiza a través del puerto serial. Para ello se programó una rutina que configurara el puerto a una velocidad de 38400 bits/s y que permitiera intercambiar bidireccionalmente información. El entorno de operación se muestra en la Figura 3.10. Cuenta con tres cuadros de texto en la parte superior sobre los cuales son desplegadas las mediciones realizadas por cada uno de los sensores de ultrasonido. Tres cuadros de texto en la parte inferior, los cuales presentan al usuario la posición en X e Y, así como la orientación del robot y cuatro botones que permiten realizar las acciones mostradas en la Tabla 3.1.

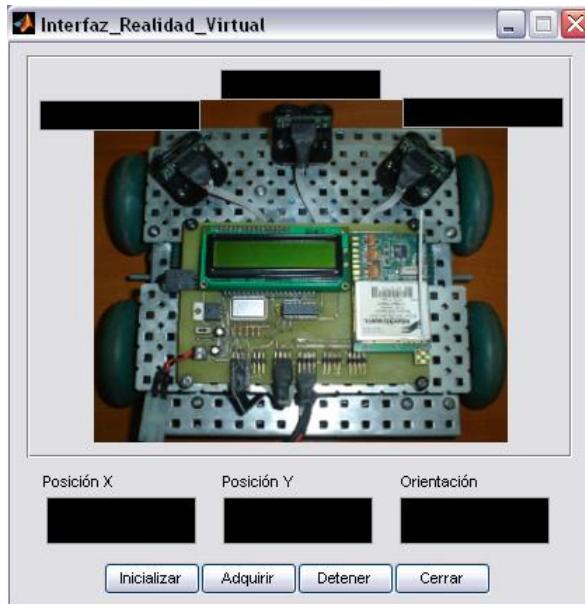


Figura 3.10. Interfaz gráfica de usuario.

Tabla 3.1. Descripción de los botones de la interfaz gráfica de usuario.

Acción	Descripción
Inicializar:	Configura el puerto serial y manipula a Orbisnap para desplegar el entorno virtual.
Adquirir:	Envía una bandera al robot para iniciar el flujo del programa cargado en el microcontrolador. Inicia la transferencia de información proveniente del robot y actualiza su posición dentro del entorno virtual.
Detener:	Envía una bandera al robot para detener el flujo del programa cargado en el microcontrolador. Detiene la adquisición. Cierra el puerto serial.
Cerrar:	Cierra la ventana de Orbisnap y termina la ejecución de la interfaz de usuario.

3.4 Arquitectura del robot móvil

La arquitectura del sistema se basa en tres niveles como se muestra en la Figura 3.11. El primer nivel provee información acerca de la localización del robot y su entorno de navegación, para ello está provisto de un par de encoders ópticos los cuales permiten conocer el desplazamiento del móvil y en función de este dato conocer su posición X e Y, así como su orientación. También, dispone de tres sensores de ultrasonido ubicados en la parte frontal del robot, a través de los cuales se monitorea el entorno de navegación. El segundo nivel se encarga de adquirir la información proveniente de los sensores, tomar las decisiones pertinentes, controla los actuadores, hace desplegar la información en el LCD y la enviar vía radiofrecuencia hacia la computadora. Finalmente, el tercer nivel permite el desplazamiento del robot.

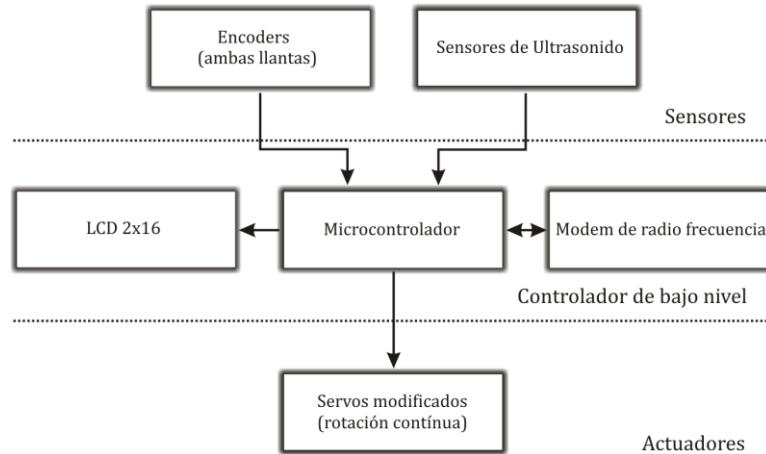


Figura 3.11. Arquitectura del robot móvil.

La implementación física del sistema se presenta en la Figura 3.12.

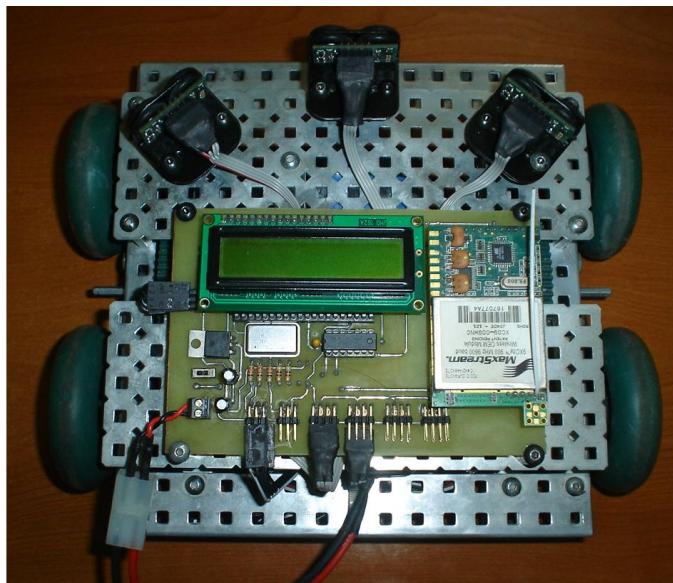


Figura 3.12. Robot móvil.

3.5 Dimensiones del robot móvil

Como se pudo observar en las expresiones odométricas (ver 2.3.3) el desplazamiento del robot está relacionado directamente con el diámetro de las llantas (DR) del móvil y la orientación está en función de la separación de las mismas (L). Siendo así, resulta necesario conocer las dimensiones del robot. La figura 3.13 presenta las dimensiones del robot empleado en éste trabajo.

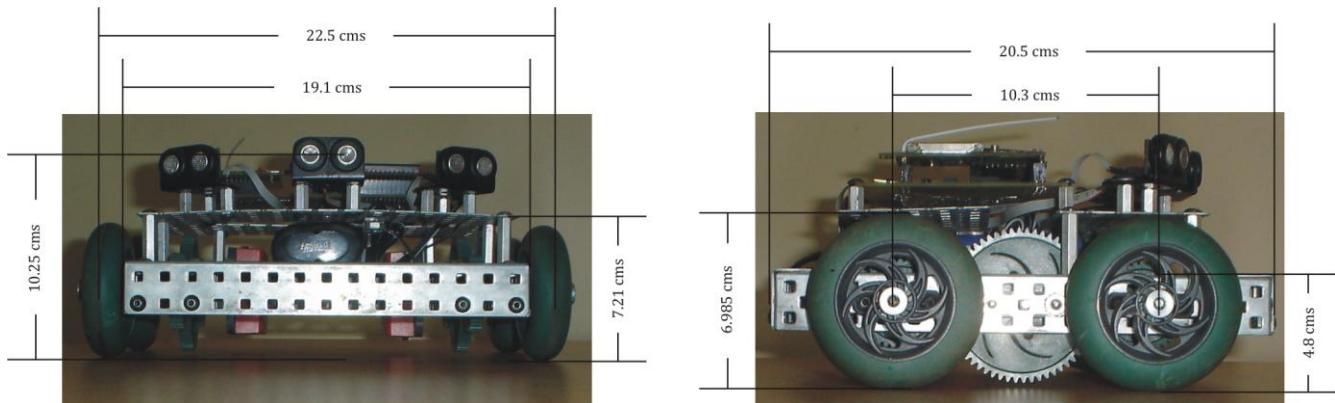


Figura 3.13. Dimensiones del robot móvil.

A partir de las imágenes se tiene que $DR=6.985$ cms y $L=22.5$ cms.

3.6 Electrónica empleada

Para el desarrollo de este trabajo se implementaron dos módulos electrónicos. El primero de ellos lleva a cabo el control del robot móvil y el envío mediante radio frecuencia de la trama de información (Figura 3.14), la cual contiene las mediciones de los sensores ultrasónicos y la posición actual del robot. El siguiente bloque recibe la trama de información a través de radio frecuencia, la manipula y espera una petición de envío por parte de la interfaz gráfica de usuario.

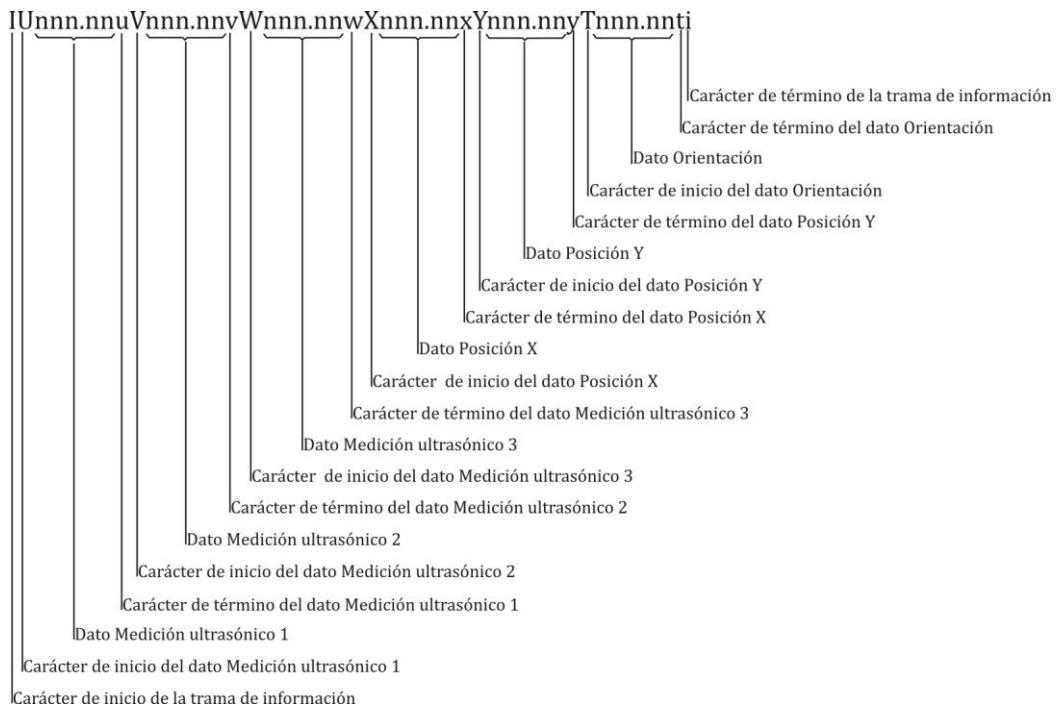


Figura 3.14. Contenido de la trama de información.

3.6.1 Electrónica de control/Transmisor de radio frecuencia

El sistema mínimo tiene como elemento principal al microcontrolador de 8 bits ATMega32 [4]. Considerando que la tarea que más recursos puede demandar es la lógica de control, se configuró este dispositivo a su máxima capacidad de procesamiento, siendo capaz de desarrollar 16 MIPS (millones de instrucciones por segundo), para ello fue necesario utilizar un oscilador de 16 MHz. Aunado a esto, las demás tareas que el microcontrolador debe realizar, así como los recursos y la programación que les son requeridas se resumen a continuación:

- *Control de motores:*

Módulos PWM configurados para generar pulsos de 1 a 2 ms con un periodo de 18 ms.

- *Manipulación de los sensores de ultrasonido*

Bus TWI/I²C configurado a una velocidad de transferencia de 100 KHz para programar y adquirir la distancia medida por los sensores de ultrasonido (modelo SRF10) [5].

- *Transferencia de las mediciones realizadas mediante radio frecuencia*

Configuración bidireccional del módulo USART (Transmisor y receptor serial síncrono y asíncrono) a una velocidad de 9600 baudios/s, lo cual permite transmitir la trama de información a través del modem de radio frecuencia (modelo 9Xcite) [6].

- *Manipulación del LCD*

Puerto A configurado como bus bidireccional de datos y Puerto C como bus de control .

En el anexo B se presentan los diagramas de flujo y el código implementado en éste módulo.

La Figura 3.15 presenta el diagrama esquemático de la distribución de este módulo.

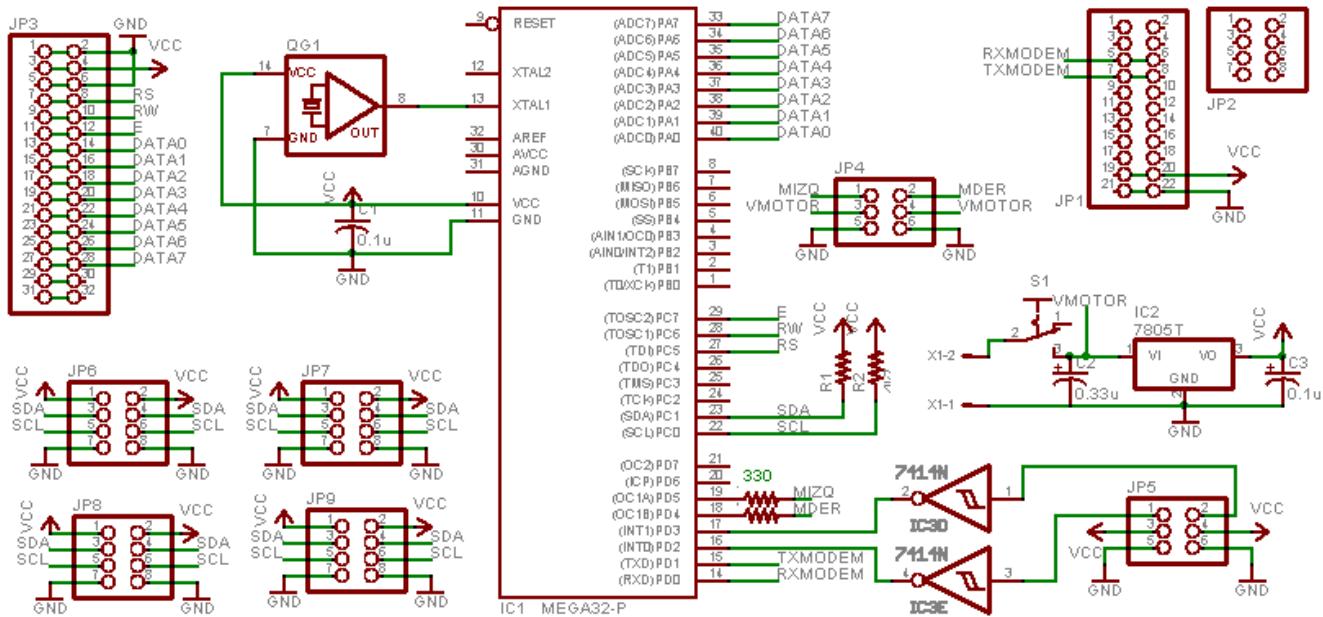


Figura 3.15. Diagrama esquemático del módulo de control.

Posteriormente se hizo el diseño y la elaboración el circuito impreso, la Figura 3.16 muestra la implementación física del módulo de control.



Figura 3.16. Circuito impreso del módulo de control.

3.6.2 Receptor de radiofrecuencia

Este módulo recibe vía radiofrecuencia la trama de información proveniente del robot. Considerando que el modem utiliza un módulo USART (a una velocidad de 9600 bits por segundo) para interactuar con otros dispositivos y que la interfaz gráfica de usuario fue diseñada para adquirir información mediante la interfaz RS-232 (a una velocidad de 38400 bits por segundo) era necesario utilizar un dispositivo con dos módulos USART.

Con el argumento anterior se utilizó un microcontrolador ATMega128 [7]. Utilizando el método del peleo¹ se monitorea la recepción de la trama de información (ver 3.6) proveniente del robot, posteriormente se extrae cada dato de las mediciones realizadas y es asignado a diferentes variables; cuando la interfaz de usuario envía una petición de información se envía cada dato por separado (en el anexo C se presentan el diagrama de flujo y el código implementado). La Figura 3.17 ilustra el diagrama esquemático de este módulo. De la misma forma que el sistema de control se hizo el diseño y la elaboración del circuito impreso que se muestra en la Figura 3.18.

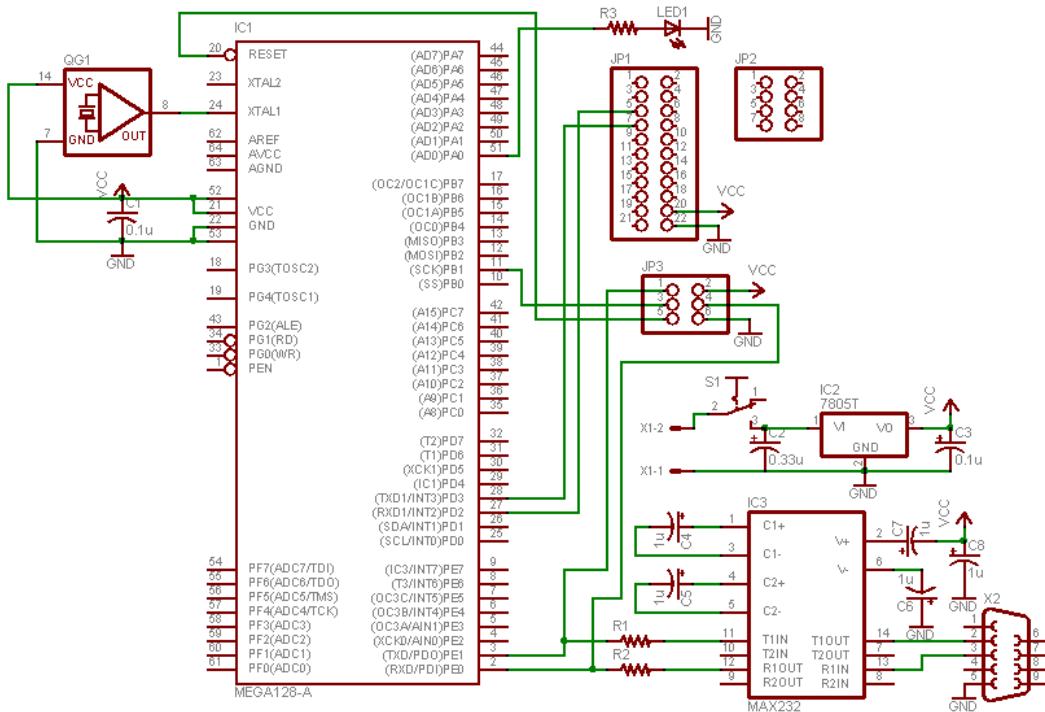


Figura 3.17. Diagrama esquemático del módulo receptor.

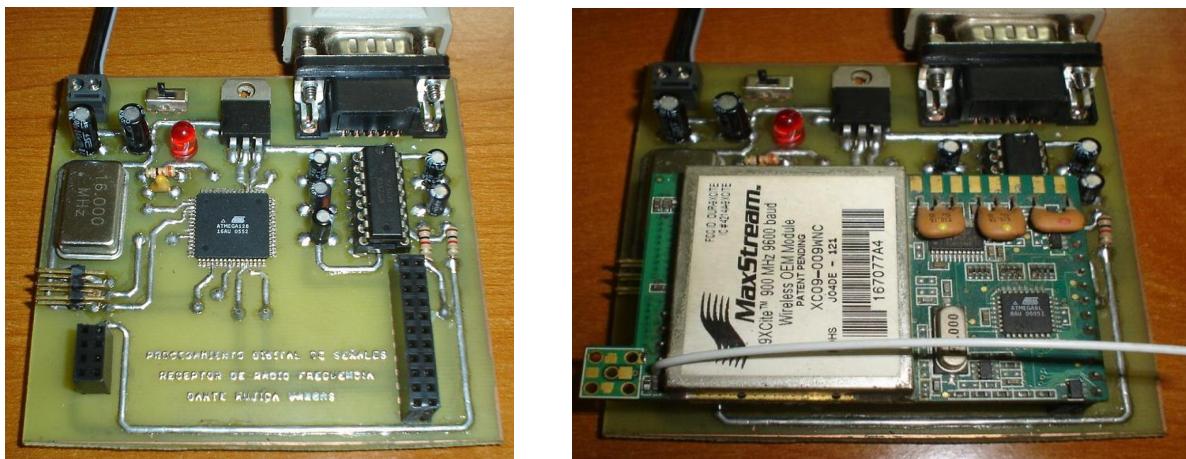


Figura 3.18. Circuito impreso del módulo receptor.

¹ Poleo/Polling: Es una operación de consulta constante hacia un dispositivo de hardware para crear una actividad síncrona, sin el uso de interrupciones, aunque también puede suceder lo mismo para recursos de software.

3.7 Descripción y programación del controlador lógico difuso

Para controlar la navegación del robot dentro del entorno real se optó por emplear lógica difusa, dentro de los argumentos que se tomaron en cuenta para dicha elección se pueden citar: la simplicidad y facilidad de implementación; puede llevar varias tareas a la vez (evadir objetos, determinar la posición de un objetivo, etc.), coordinarlas y darles prioridad dependiendo de la situación en la se encuentre el robot, permitiendo con esto no utilizar métodos especiales para cada una de éstas tareas; considera la información presente en sus entradas y en base a esta proporciona información útil que permite configurar a los actuadores. En base a lo anterior, para controlar la navegación del móvil se propone un controlador difuso con cuatro entradas y dos salidas (ver Figura 3.19) para controlar la navegación del robot móvil. Tres entradas corresponden a la distancia medida por los sensores de ultrasonido y la cuarta es la posición de la meta. Las salidas son la velocidad con que el robot debe desplazarse y el ángulo de rotación para alcanzar la meta o evadir obstáculos.

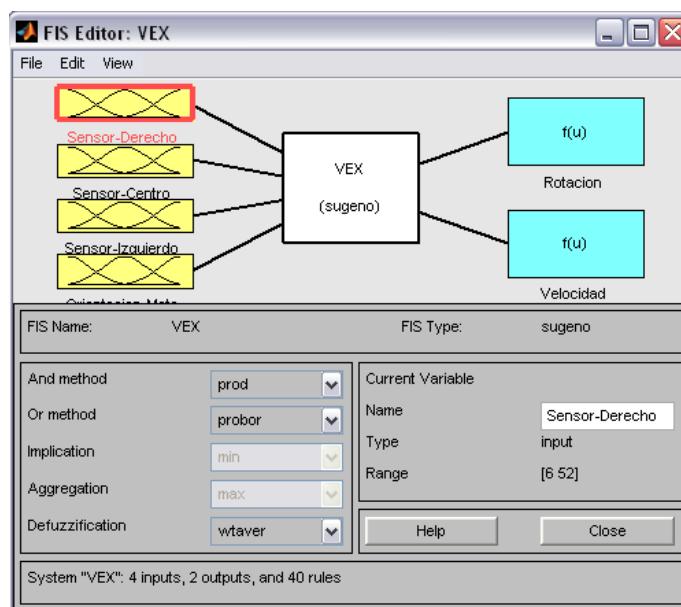


Figura 3.19. Controlador difuso propuesto.

Las entradas referidas a la medición de los sensores (ver Figuras 3.20, 3.21 y 3.22) fueron modeladas a partir de dos conjuntos difusos (objeto cercano y objeto lejano). La entrada correspondiente a la posición de la meta (ver Figura 3.23) cuenta con cinco grupos (izquierda, poco-izquierda, centro, poco-derecha y derecha). La evaluación de todos estos grupos proporciona una gran flexibilidad durante la toma de decisiones.

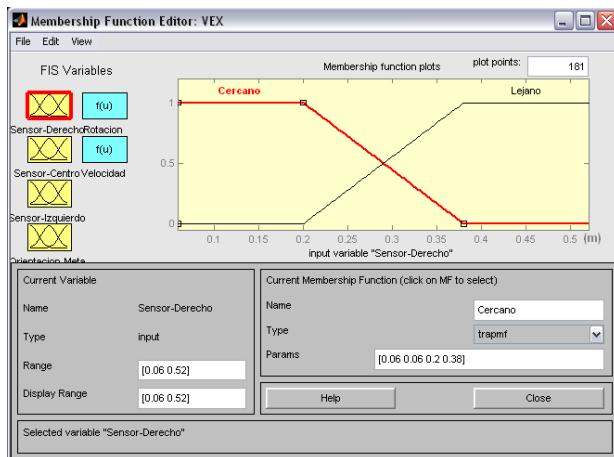


Figura 3.20. Conjuntos de la entrada Sensor-Derecho.

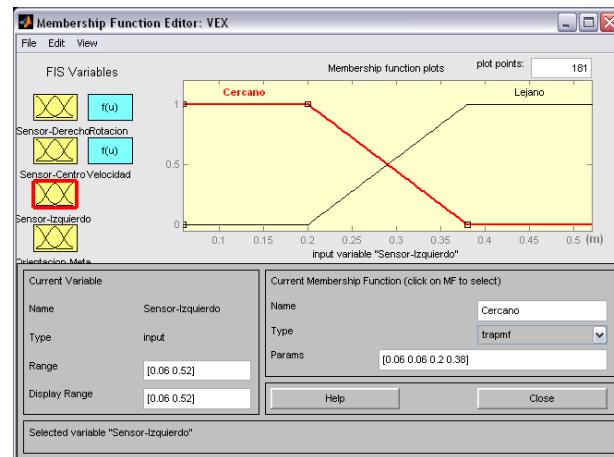


Figura 3.21. Conjuntos de la entrada Sensor-Centro.

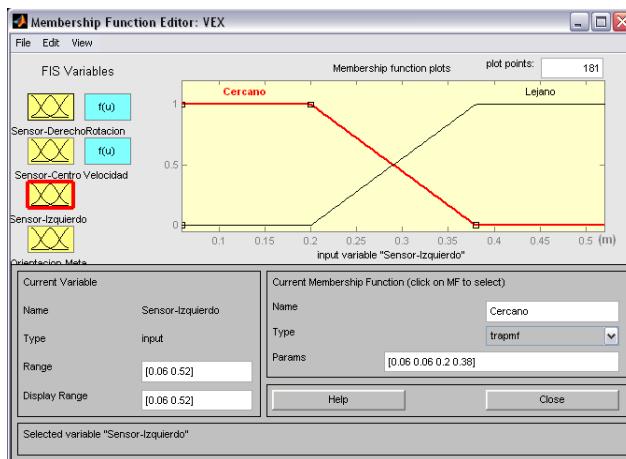


Figura 3.22. Conjuntos de la entrada Sensor-Izquierdo.

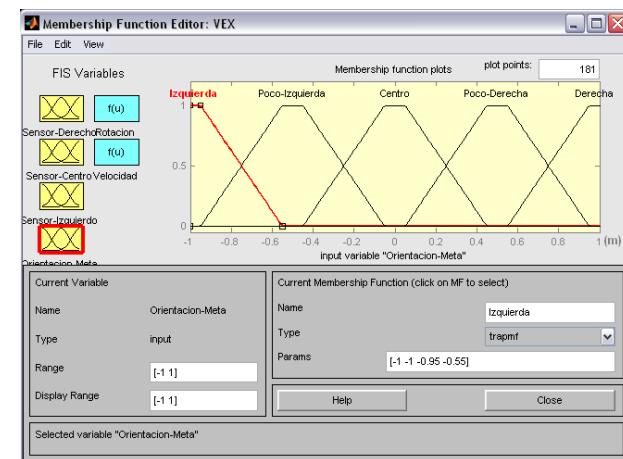


Figura 3.23. Conjuntos de la entrada Posición-Meta.

Para determinar los valores de Rotación y Velocidad del robot es necesario llevar a cabo las tres partes fundamentales en el funcionamiento de un sistema de control difuso.

1) Fuzzificación

Mediante este proceso se toma la información presente en cada una de las entradas y se calcula el grado de pertenencia $\mu(x)$ a uno o varios de los conjuntos difusos en que se divide el rango de valores posibles para su correspondiente entrada. Formalizando la notación y considerando, en general el dominio de valores de entrada dividido en N conjuntos difusos, denominados A_1, A_2, \dots, A_N , los valores que se obtienen después de ésta etapa son los grados de pertenencia del valor de entrada x a cada uno de estos conjuntos difusos: $\mu_{A1}(x), \mu_{A2}(x), \dots, \mu_{AN}(x)$. Por ejemplo, si en la entrada *Sensor-Derecho* fue detectado un objeto a 0.2 metros, los grados de pertenencia para los dos conjuntos difusos son:

$$\begin{aligned}\mu_{Cercano}(0.2) &= 1 \\ \mu_{Lejano}(0.2) &= 0\end{aligned}$$

2) Evaluación de las reglas de control

En la navegación de un robot móvil basada en comportamientos difusos es necesario hacer una descomposición en tareas simples (comportamientos independientes), cada comportamiento esta compuesto por un conjunto de reglas que permiten lograr un conjunto de objetivos bien definidos, por ejemplo, se pueden tener las siguientes reglas:

- *SI (Sensor_Derecho es Objeto lejano) Y (Sensor_Centro es Objeto lejano) Y (Sensor_Izquierdo es Objeto lejano) ENTONCES (Velocidad será Alta).*
- *SI (Meta es Izquierda) ENTONCES (Rotación será Izquierda).*

En forma general, una regla con varias entradas y salidas puede tener la siguiente notación:

R:SI $\underbrace{\psi_1 \text{ es } A \text{ Y } \psi_2 \text{ es } B \text{ Y } \psi_3 \text{ es } C \text{ Y } \dots}_{\text{Antecedente}}$ ENTONCES $\underbrace{\psi_1 \text{ será } C \text{ Y } \psi_2 \text{ será } D \text{ Y } \psi_3 \text{ será } E \text{ Y } \dots}_{\text{Consecuente}}$

Una vez que los comportamientos individuales están diseñados, el siguiente paso es la coordinación de comportamientos con un mismo objetivo. Para llevarlo a cabo es necesario definir el conjunto de reglas de control (Tabla 3.2) para el sistema.

Tabla 3.2. Reglas del controlador lógico difuso.

SI				ENTONCES	
Sensor Derecho (Objeto)	Sensor Centro (Objeto)	Sensor Izquierdo (Objeto)	Posición Meta	Velocidad	Rotación
Lejano	Lejano	Lejano	Izquierda	Alta	Izquierda
Lejano	Lejano	Lejano	Poco Izquierda	Alta	Poco Izquierda
Lejano	Lejano	Lejano	Centro	Alta	Centro
Lejano	Lejano	Lejano	Poco Derecha	Alta	Poco Derecha
Lejano	Lejano	Lejano	Derecha	Alta	Derecha
Lejano	Lejano	Cercano	Izquierda	Media	Derecha
Lejano	Lejano	Cercano	Poco Izquierda	Media	Poco Derecha
Lejano	Lejano	Cercano	Centro	Media	Poco Derecha
Lejano	Lejano	Cercano	Poco Derecha	Alta	Poco Derecha
Lejano	Lejano	Cercano	Derecha	Alta	Derecha
Cercano	Lejano	Lejano	Izquierda	Alta	Izquierda
Cercano	Lejano	Lejano	Poco Izquierda	Alta	Poco Izquierda
Cercano	Lejano	Lejano	Centro	Media	Poco Izquierda
Cercano	Lejano	Lejano	Poco Derecha	Media	Poco Izquierda

Cercano	Lejano	Lejano	Derecha	Media	Izquierda
Lejano	Cercano	Lejano	Izquierda	Alta	Izquierda
Lejano	Cercano	Lejano	Poco Izquierda	Alta	Poco Izquierda
Lejano	Cercano	Lejano	Centro	Media	Poco Derecha
Lejano	Cercano	Lejano	Poco Derecha	Alta	Poco Derecha
Lejano	Cercano	Lejano	Derecha	Alta	Derecha
Cercano	Cercano	Cercano	Izquierda	Baja	Izquierda
Cercano	Cercano	Cercano	Poco Izquierda	Baja	Poco Izquierda
Cercano	Cercano	Cercano	Centro	Baja	Centro
Cercano	Cercano	Cercano	Poco Derecha	Baja	Poco Derecha
Cercano	Cercano	Cercano	Derecha	Baja	Derecha
Cercano	Cercano	Lejano	Izquierda	Alta	Izquierda
Cercano	Cercano	Lejano	Poco Izquierda	Alta	Izquierda
Cercano	Cercano	Lejano	Centro	Alta	Izquierda
Cercano	Cercano	Lejano	Poco Derecha	Media	Izquierda
Cercano	Cercano	Lejano	Derecha	Media	Izquierda
Lejano	Cercano	Cercano	Izquierda	Media	Derecha
Lejano	Cercano	Cercano	Poco Izquierda	Media	Derecha
Lejano	Cercano	Cercano	Centro	Alta	Derecha
Lejano	Cercano	Cercano	Poco Derecha	Alta	Derecha
Lejano	Cercano	Cercano	Derecha	Alta	Derecha
Cercano	Lejano	Cercano	Izquierda	Media	Centro
Cercano	Lejano	Cercano	Poco Izquierda	Media	Centro
Cercano	Lejano	Cercano	Centro	Media	Centro
Cercano	Lejano	Cercano	Poco Derecha	Media	Centro
Cercano	Lejano	Cercano	Derecha	Media	Centro

La interpretación de este tipo de reglas se puede ejemplificar de la siguiente forma,

SI el sensor derecho detecta un objeto lejano y el sensor central detecta un objeto lejano y el sensor izquierdo detecta un objeto lejano, y la odometría determina que la posición de la meta está en la dirección izquierda. ENTONCES el robot debe girar a la izquierda y a una velocidad alta.

La evaluación de la base de reglas del sistema consiste en determinar que regla (o reglas) se activará ante un determinado valor de entrada. Para averiguarlo, se parte del grado de pertenencia de dicho valor a cada uno de los diferentes conjuntos difusos del dominio de entrada. A cada regla se le asocia un valor determinado *peso*, que en principio coincide con el grado de pertenencia de la entrada *x* al conjunto indicado en el antecedente $\mu_{Ai}(x)$. Este peso será el que permitirá establecer el grado de pertenencia de la salida *y* del sistema al conjunto indicado en el consecuente de la regla $\mu_{ci}(y)$.

Cuando el sistema se compone de varias entradas, el antecedente se compone de varias condiciones referidas a los conjuntos a los que pertenecen éstas. En éste caso para calcular el peso asociado a cada una de las reglas se pueden aplicar los operadores producto y suma lógica difusa o aritmética. Para éste trabajo se definieron todas las reglas utilizando el operador producto. El utilizar éste método de implicación significa que deben cumplirse todas las condiciones en el antecedente para tener una salida determinada. Siendo así, el peso de las reglas es determinado de la siguiente forma:

$$\text{Peso Regla } i = \mu_{A_i}(x_i) \times \mu_{B_i}(x_j) \times \mu_{B_i}(x_i) \times \mu_{C_i}(x_i) \times \dots \times \mu_{Z_i}(x_i)$$

$$\text{Peso Regla } j = \mu_{A_j}(x_j) \times \mu_{B_j}(x_j) \times \mu_{B_j}(x_j) \times \mu_{C_j}(x_j) \times \dots \times \mu_{Z_j}(x_j)$$

$$\vdots$$

$$\text{Peso Regla } n = \mu_{A_n}(x_n) \times \mu_{B_n}(x_n) \times \mu_{B_n}(x_n) \times \mu_{C_n}(x_n) \times \dots \times \mu_{Z_n}(x_n)$$

Finalmente, el grado de pertenencia de la salida $\mu(y)$ a un conjunto en el dominio de valores de una salida se corresponde con el resultado de la suma de todas las reglas en cuyo consecuente aparezca el conjunto.

$$\mu_c(y_i) = \text{Peso Regla } i + \text{Peso Regla } j + \dots + \text{Peso Regla } n$$

3) Defuzzificación

Consiste en obtener un valor numérico para cada una de las salidas del sistema a partir de los conjuntos difusos definidos para cada variable. En el caso de la salida Rotación se proponen cinco conjuntos (Izquierda, Poco-Izquierda, Centro, Poco-Derecha y Derecha) como se muestra en la Figura 3.24. Para tener la flexibilidad necesaria y lograr resultados más precisos se optó por trabajar con cinco conjuntos para esta salida. Finalmente la variable Velocidad (Figura 3.25) está modelada a partir de tres conjuntos, el resultado es esencial en el control de la rapidez del robot móvil ya que permite disminuir la velocidad cuando existen objetos alrededor o incrementarla cuando no existen cuerpos cercanos.

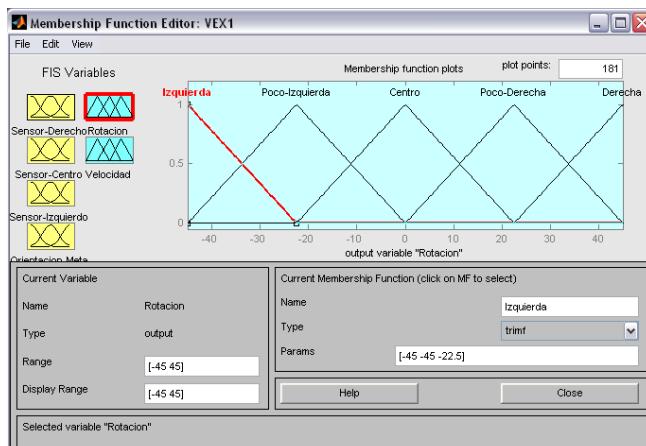


Figura 3.24. Conjuntos de la salida Rotación.

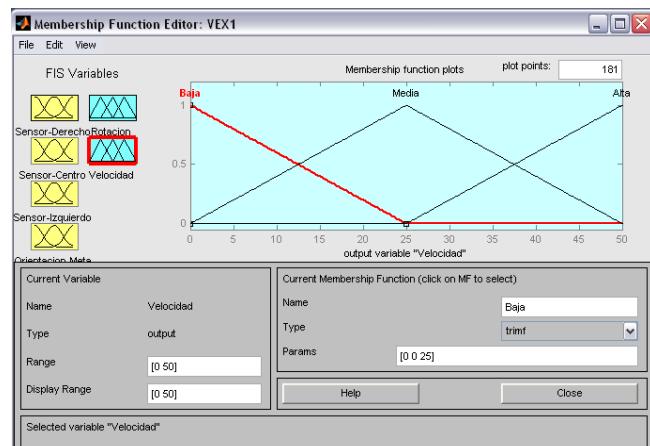


Figura 3.25. Conjuntos de la salida Velocidad.

Aunque existen varias técnicas para llevar a cabo la defuzzificación, se utilizó el método llamado Semifallo (Controlador Difuso Takagi-Sugeno) [9], éste es un método funcional que permite obtener el mismo resultado que otros métodos con un menor número de operaciones. Consiste en calcular el promedio de los centroides de las funciones de pertenencia de los conjuntos difusos activados para cada salida. Al ser funciones simétricas, los centroides (puntos donde el grado de pertenencia toma su valor máximo y es unitario) coinciden con el punto medio y pueden ser tomados como constantes, siendo así, las salidas pueden ser representadas como se muestra en las Figuras 3.26 y 3.27. Para la ponderación se utilizan los pesos de las reglas activadas, o grados de pertenencia de las salidas a los conjuntos que le corresponden. Esto puede expresarse matemáticamente de la siguiente forma [8]:

$$y_{\text{salida}} = \frac{\sum_{k=1}^n y_k \mu(y_k)}{\sum_{k=1}^n \mu(y_k)} \quad (3.1)$$

donde y_k es el centroide de cada conjunto difuso y $\mu(y_k)$ es el grado de pertenencia a cada uno de los conjuntos.

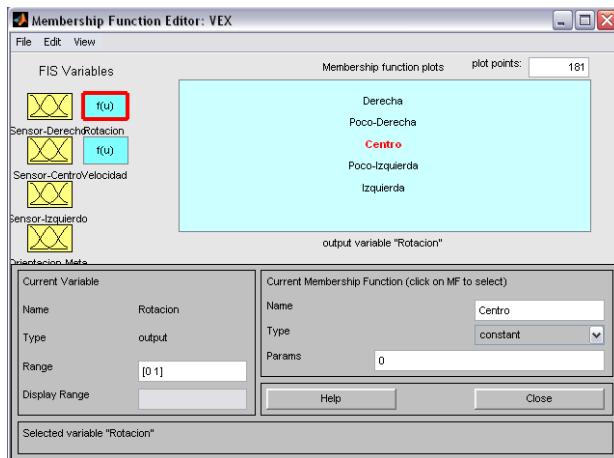


Figura 3.26. Centroides de la salida Rotación.

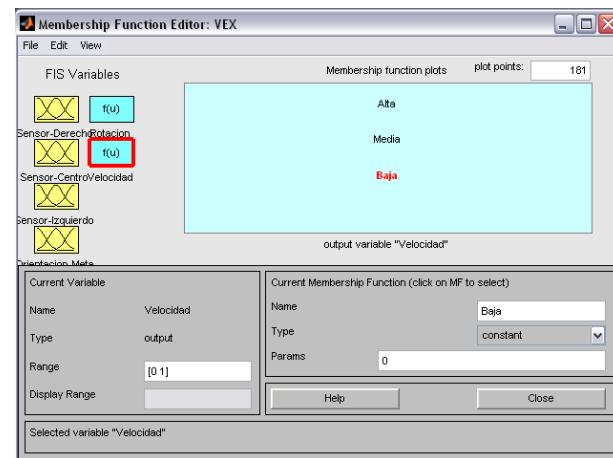


Figura 3.27. Centroides de la salida Velocidad.

Para llevar a cabo el diseño del controlador difuso se utilizó el programa ANFIS Editor GUI [10] que viene en el Toolbox de lógica difusa que MatLab 7.4 maneja; al ser una herramienta gráfica permitió llevar a cabo un diseño rápido. Sin embargo, fue requerido llevar todo esto a código que pudiera ser implementado dentro del microcontrolador, dado que estos dispositivos pueden ser programados en Lenguaje C, se escribió el código para este compilador. Las Figuras 3.28 y 3.29 muestran la comparación de resultados obtenidos en ambos casos.

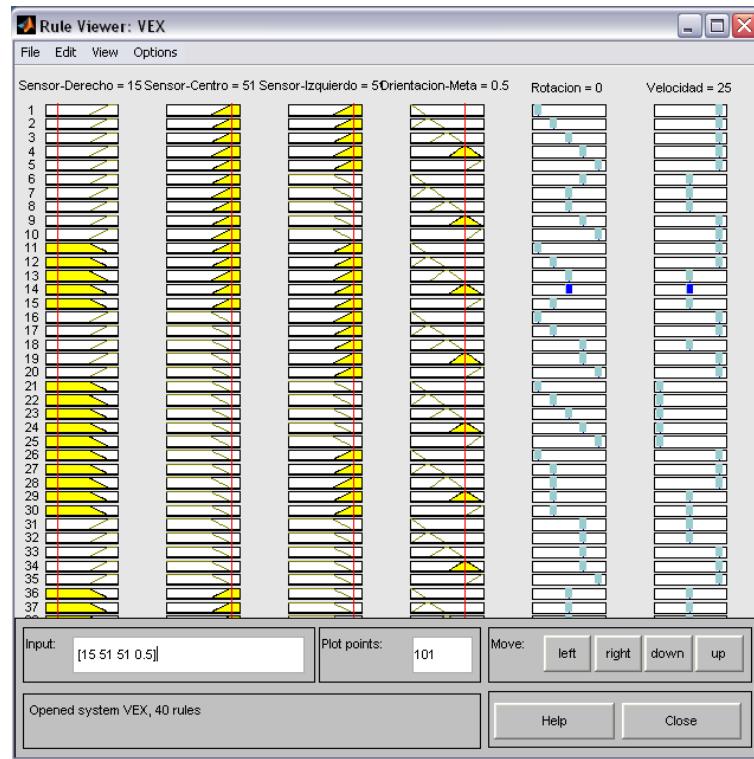


Figura 3.28. Resultados obtenidos con MatLab.

```

Simbolo del sistema - tc
File Edit Search Run Compile Debug Project Options Window Help
ROBOT.C
[(*]
    clrsr ();
    SensorDerecho = 15;
    SensorCentral = 51;
    SensorIzquierdo = 51;
    PosicionMeta = 0.5;
    Sensor Derecho Cercano = trapecio_n (SensorDerecho,6,0,36,0,50,0);
    Sensor_Derecho_Lejano = trapecio_p (SensorDerecho,36,0,50,0,56,0);
    Sensor_Central_Cercano = trapecio_n (SensorCentral,6,0,36,0,50,0);
    Sensor_Central_Lejano = trapecio_p (SensorCentral,36,0,50,0,56,0);
    Sensor_Izquierdo_Cercano = trapecio_n (SensorIzquierdo,6,0,36,0,50,0);
    Sensor_Izquierdo_Lejano = trapecio_p (SensorIzquierdo,36,0,50,0,56,0);
27.25                                     Output
Velocidad RPM: 25.000
Rotacion : 0.000
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```

Figura 3.29. Resultados obtenidos con Lenguaje C.

3.8 Incorporación del robot móvil dentro del entorno virtual

Siguiendo el procedimiento descrito en el punto 3.1 se diseñó el modelo del robot, para facilitar el procedimiento se realizó por separado del entorno virtual. La Figura 3.30 muestra las perspectivas del robot.

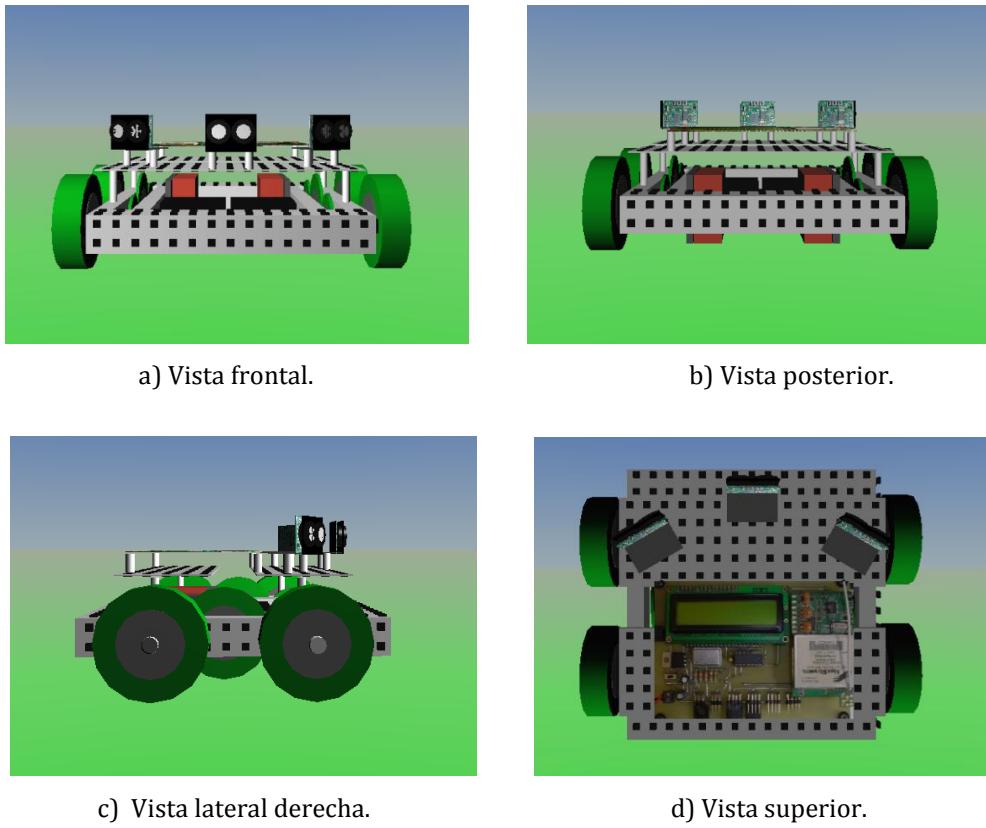


Figura 3.30. Perspectivas del robot virtual.

Posteriormente este modelo se incorporó al entorno virtual (Figura 3.31). Considerando las dimensiones del pasillo principal, el robot virtual fue puesto a 0.88 metros de distancia de la pared de la sección de Telecomunicaciones y centrado a 1.75 metros en ambos lados. Tomando en cuenta que la odometría del robot permite su desplazamiento en el plano cartesiano, esta posición del robot virtual se debe considerar como el punto de referencia o partida ($x=0, y=0$) en el mundo real y virtual.

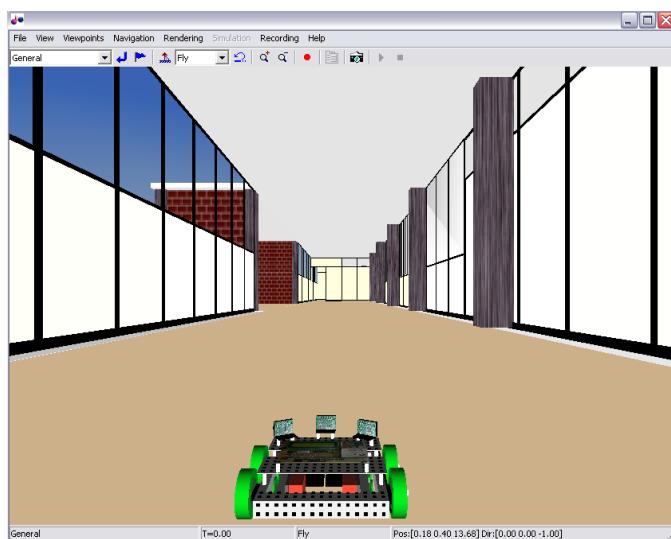


Figura 3.31. Sistema virtual.

3.9 Conclusiones

Este capítulo describió el sistema propuesto para la navegación del robot en los entornos real/virtual. Se presentó el procedimiento gráfico utilizado para desarrollar el entorno virtual, así como la interfaz gráfica de usuario para manipularla. Posteriormente se profundizó en la arquitectura del robot y la electrónica empleada, desglosándose en dos módulos, uno para controlar al robot móvil y el otro para recibir su posición actual vía radio frecuencia. Después se explicó la técnica para dotar de inteligencia artificial al robot, basando su diseño en la información adquirida por los sensores del robot y su posición. Por último se explicó cómo está conformado el sistema virtual y la posición que el robot real y virtual deben tener en ambos mundos.

3.10 Referencias

- [1] "V-Realm™ Builder: User's guide and reference". Ligos Corporation. 1996-1997.
- [2] MatLab | R2007a. The language of Technical Computing. 2007.
- [3] Graphical User Interface / Graphical User Interface Development Environment. MatLab help. 2007.
- [4] ATMega32 data sheet. 8-bit AVR Microcontroller with 32k bytes In-System Programmable flash. www.atmel.com
- [5] SRF10 Ultrasonic range finder. Technical specification. Davantech sonic range finders. www.robot-electronics.co.uk
- [6] 9Xcite-PKG™ RF Modems. MaxStream. www.maxstream.net
- [7] ATMega128 data sheet. 8-bit AVR Microcontroller with 128k bytes In-System Programmable flash. www.atmel.com.
- [8] José R. Hilera, Victor J. Martínez. "Redes neuronales artificiales: Fundamentos, modelos y aplicaciones". AlfaOmega, 2000.
- [9] Kevin M. Pasino, Stephen Yorkovich. "Fuzzy Control". Addison-Wesley. 1998.
- [10] Adaptive Neuro-Fuzzy Inference System. Fuzzy Logic Toolbox. MatLab 7.4. 2007.

Capítulo 4

Resultados experimentales

4.1 Introducción

En este capítulo se presenta la caracterización de los elementos sensores y actuadores del sistema. También se muestran las pruebas de navegación realizadas al sistema propuesto, así como la respuesta del sistema ante señales procesadas por sus sensores y actuadores.

4.2 Caracterización de los actuadores y sintonización del controlador

Una tarea presente en la robótica móvil es el control de motores (velocidad, posición y el sentido de giro). Aunque en la literatura existen métodos tradicionales para realizar dicho control, para este trabajo se optó por utilizar un controlador PI fusionado (ver 2.3.5), ya que éste permite controlar simultáneamente dos motores [1], siendo ideal para un robot con configuración diferencial como el utilizado en éste trabajo. Antes de iniciar el diseño fue necesario conocer el comportamiento de los motores, para ello se midieron la velocidad de giro en función del ancho de pulso que se les aplica. La Figura 4.1 muestra las curvas de transferencias de ambos motores. Es necesario indicar que por convención se toma como velocidad negativa cuando el eje del motor gira en el mismo sentido de las manecillas del reloj y positivo cuando gira en sentido contrario.

Dada la configuración diferencial del robot es importante tomar en cuenta la distribución de los dos motores (ver Figura 4.2). Por lo tanto, para que el robot tenga un desplazamiento hacia adelante es necesario que el motor situado en el lado derecho lleve a cabo una rotación negativa, en cambio el motor izquierdo requiere de una rotación positiva (ver Figura 4.3.a). En el caso contrario, el motor derecho requiere de una rotación positiva y el izquierdo una rotación negativa, tal como lo ilustra la Figura 4.3.b.

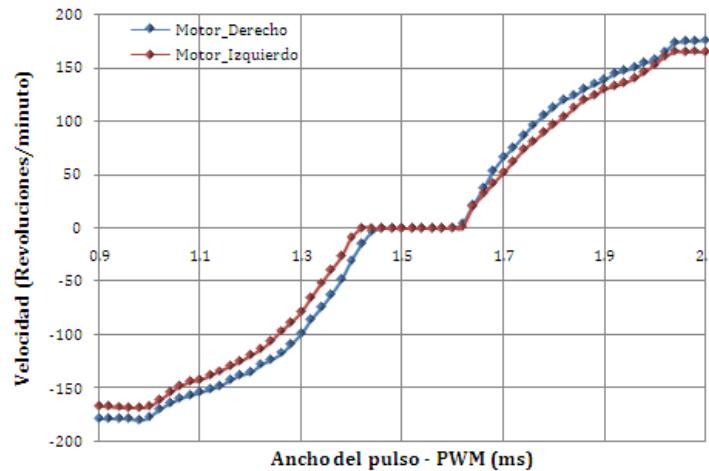
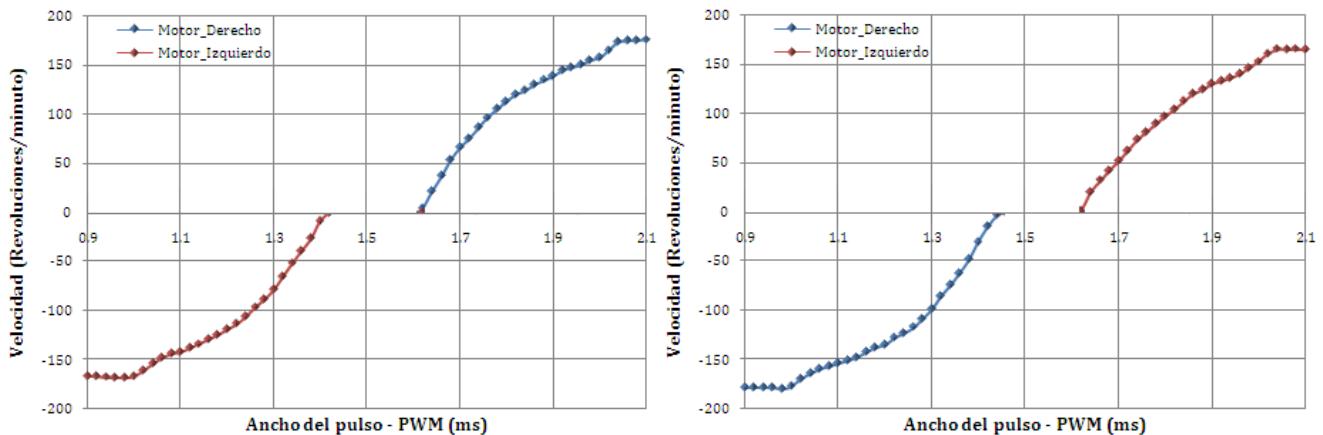


Figura 4.1. Curvas de transferencia de los motores.



Figura 4.2. Distribución de los motores.



a) Desplazamiento hacia delante.

b) Desplazamiento hacia atrás.

Figura 4.3. Desplazamientos del robot.

Aunque existen métodos basados en las técnicas de la Teoría del Control y métodos empíricos para conocer las constantes (K_p , K_i y K_d) de un controlador (ver 2.3.5) y se puede lograr un buen control de la velocidad de los motores (en forma separada), esto no garantiza que un robot sea capaz de lograr un desplazamiento preciso sobre una trayectoria recta [2]. Por lo tanto, si se utilizan estos métodos es necesario sintonizar el controlador bajo las

condiciones de operación en las que se encuentre el robot hasta lograr el mejor comportamiento o bien utilizar el método de prueba y error [1].

En base a lo anterior, la sintonización de las constantes del controlador PI se realizó a través de prueba y error. El desplazamiento más preciso sobre una trayectoria recta con desplazamiento hacia delante se logró con las constantes presentadas en la Tabla 4.1. Considerando estas constantes y la velocidad máxima de desplazamiento del robot (50 revoluciones/minuto), la respuesta gráfica del controlador PI se muestra en la Figura 4.4.

Tabla 4.1. Constantes del controlador PI.

Constante	Motor Derecho	Motor Izquierdo
K _p	0.138	0.195
K _i	0.05	0.05

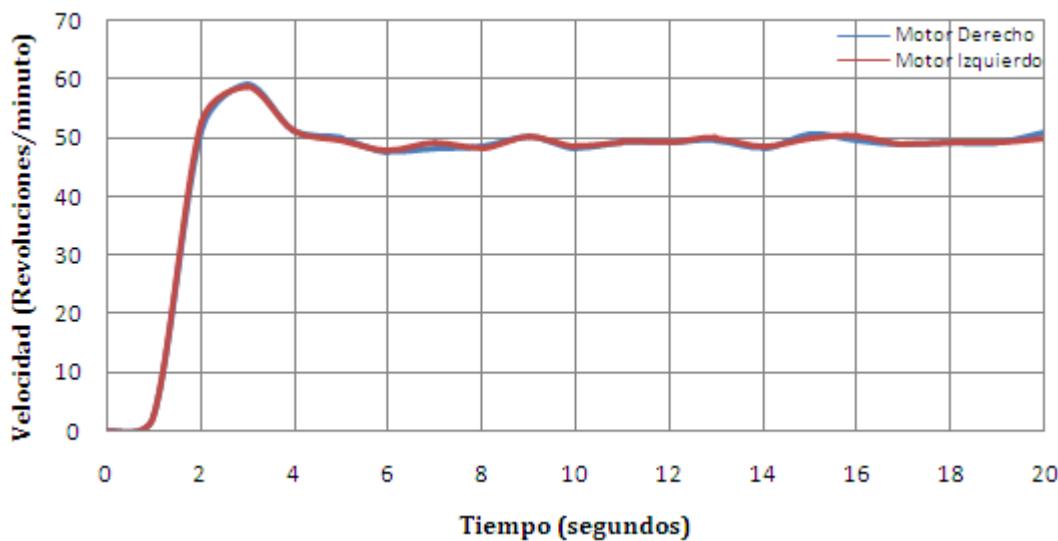


Figura 4.4. Respuesta del controlador PI.

4.3 Caracterización de los sensores

Puesto que todos los sensores vienen provistos de fábrica con la misma dirección, fue necesario implementar una rutina de programación que permitiera definir una nueva dirección a dos de los tres sensores utilizados en este trabajo. Se realizaron algunas pruebas para conocer el comportamiento de cada uno de los sensores ante la presencia de objetos con diferentes formas geométricas y dimensiones (ver Figura 4.5), las dimensiones de los objetos se presentan en la Tabla 4.2. Para ello se optó por configurar a los sensores como se muestra en la Tabla 4.3.

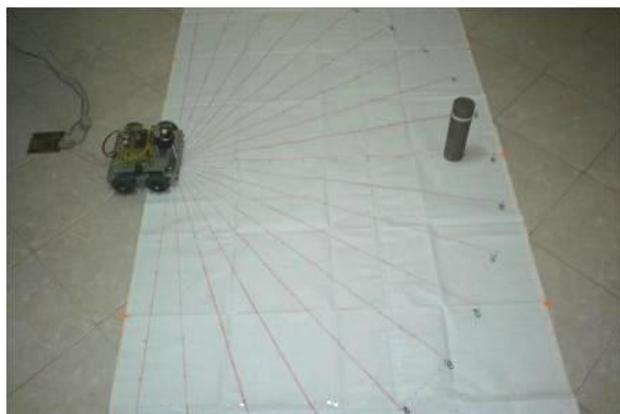
Tabla 4.2. Dimensiones de los objetos.

Objeto	Dimensiones	Material
Cilindro	Altura: 21 centímetros Diámetro: 6 centímetros	Cartón
Rectángulo	Altura: 23.7 centímetros Ancho: 10.1 centímetros	Cartón
Rectángulo	Altura: 30.5 centímetros Ancho: 0.35 centímetros	Metal
Rectángulo	Altura: 9.6 centímetros Ancho: 1.9 centímetros	Madera

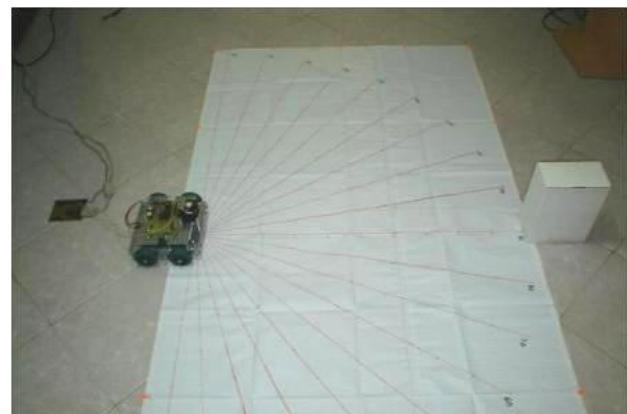
Tabla 4.3. Configuración de los sensores.

Alcance máximo de medición	1032 centímetros
Unidad de medición	Centímetros
Ganancia	8
Tiempo de muestreo	65 milisegundos

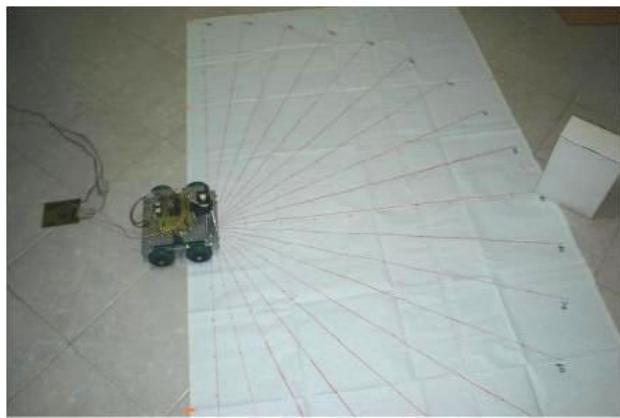
Es necesario indicar que la ganancia puede tomar valores de 0 a 16. Dado que no existe una relación lineal entre el alcance de medición y este parámetro, por lo tanto, se debe seleccionar a través de prueba y error, a la par debe irse modificando el tiempo de muestreo hasta que la distancia medida corresponda con la separación entre el sensor y el objeto.



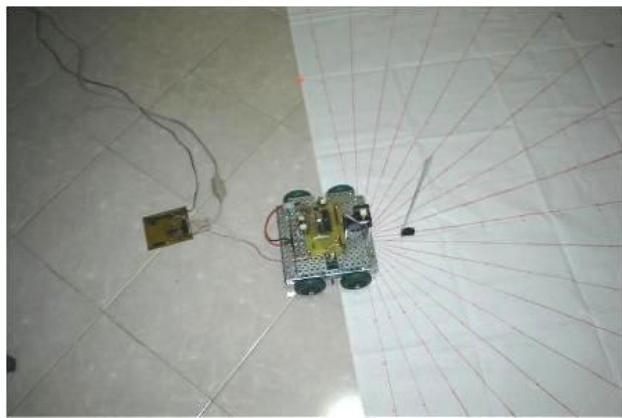
a) Detección de un cilindro.



b) Detección de un cubo.



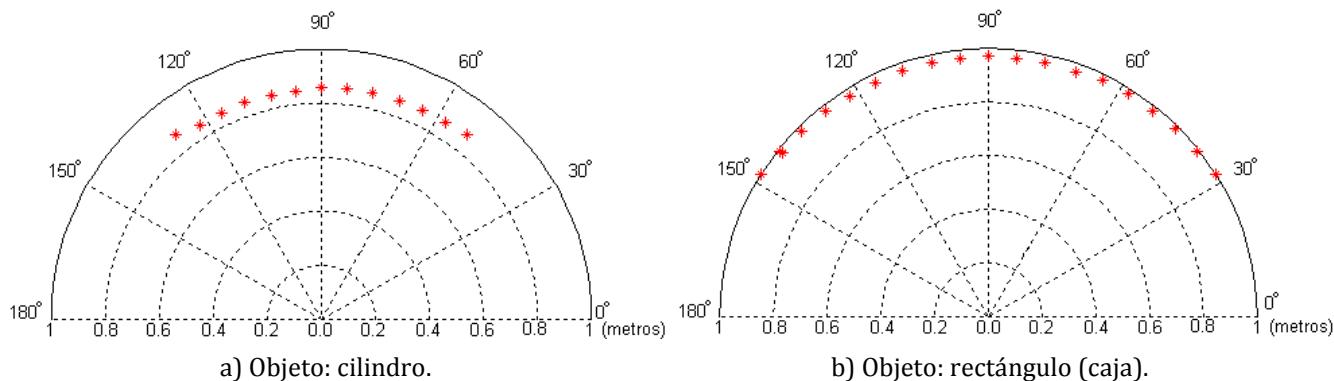
c) Detección de una arista del cubo.



d) Detección de un objeto diminuto.

Figura 4.5. Detección de diferentes formas geométricas.

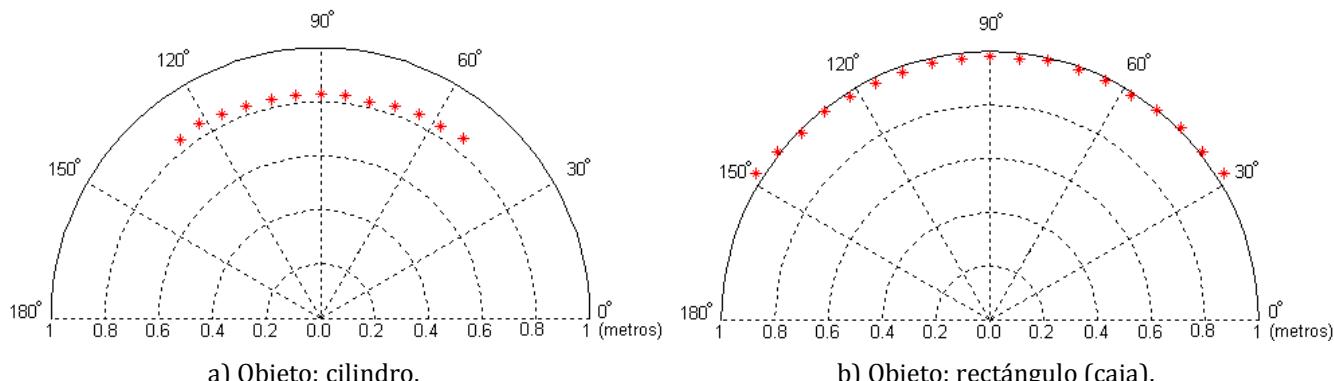
De este muestreo se observó la misma tendencia en los tres sensores: los objetos con una superficie plana son más detectables siempre y cuando ésta sea mayor a 2 cms, el objeto cilíndrico fue detectado a una distancia menor, los objetos con aristas fueron indetectables aún cuando estuvieran muy cerca del sensor y con los objetos diminutos (menores de 2 cms) se presentó la misma incapacidad para detectarlos. Los gráficos de las Figuras 4.6 a 4.8 muestran el ángulo y distancia a la que fueron detectados el cilindro y el cubo, con cada uno de los sensores.



a) Objeto: cilindro.

b) Objeto: rectángulo (caja).

Figura 4.6. Detección de objetos mediante el Sensor 1.



a) Objeto: cilindro.

b) Objeto: rectángulo (caja).

Figura 4.7. Detección de objetos mediante el Sensor 2.

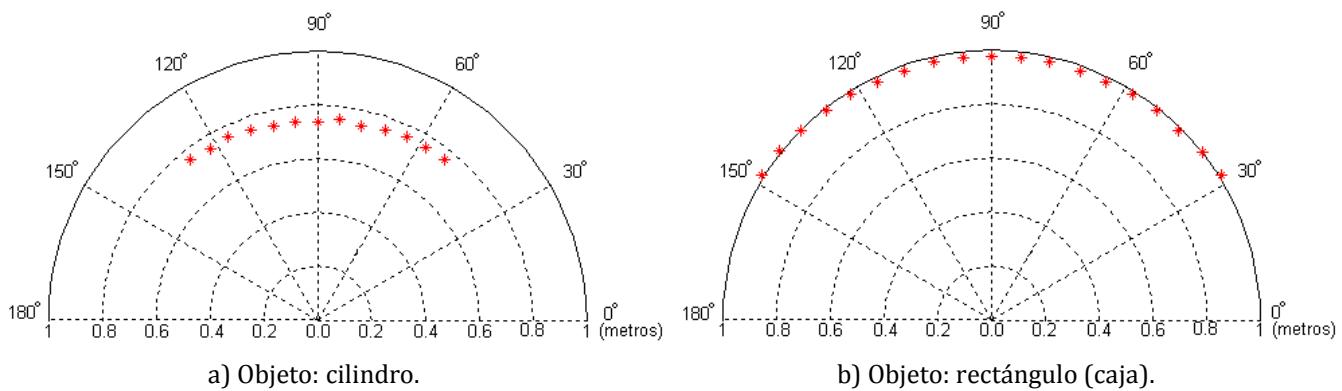


Figura 4.8. Detección de objetos mediante el Sensor 3.

Aunque el fabricante proporciona el cono de medición en forma generalizada, fue necesario conocer el área de trabajo para cada uno de los sensores, para ello se tomó un objeto con una superficie plana de 2 cm (objeto mínimo detectado) y se fue desplazando en el contorno de la media circunferencia (ver Figura 4.9), permitiendo conocer los puntos de detección del objeto y la distancia en la que se encuentra.

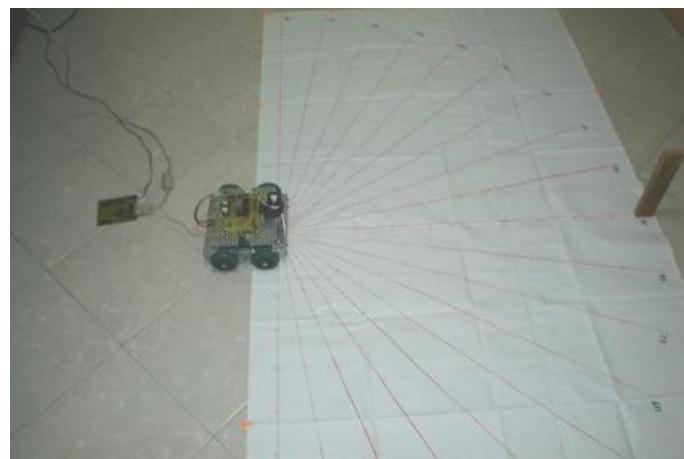
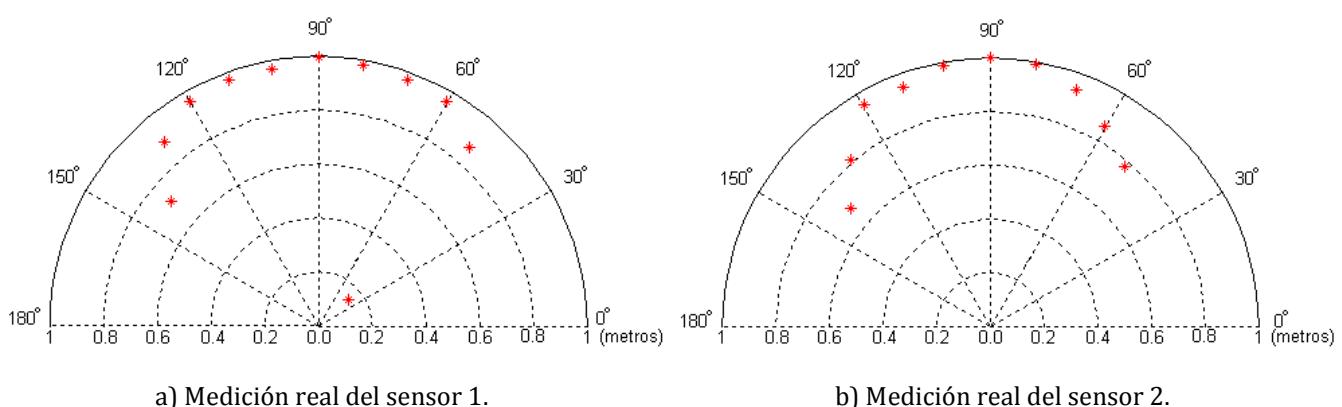


Figura 4.9. Determinación del cono de medición real.

La Figura 4.10 muestra el cono de medición real para cada uno de estos sensores empleados en este trabajo.



a) Medición real del sensor 1.

b) Medición real del sensor 2.

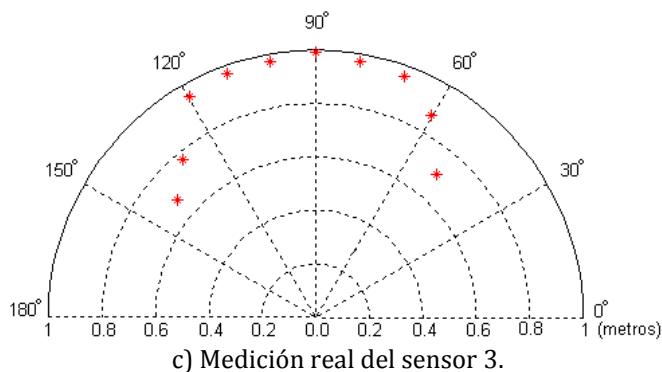


Figura 4.10. Conos de medición reales de los sensores.

Con el conocimiento del área de cobertura de cada sensor, se montaron en el robot y se hicieron algunas pruebas para determinar la mejor distribución de los sensores. Observándose el mejor resultado cuando los sensores estaban en la posición que se muestra en la Figura 4.11.

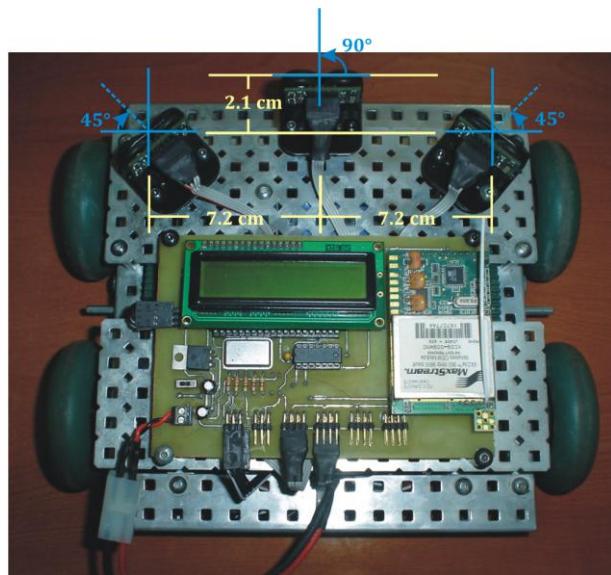
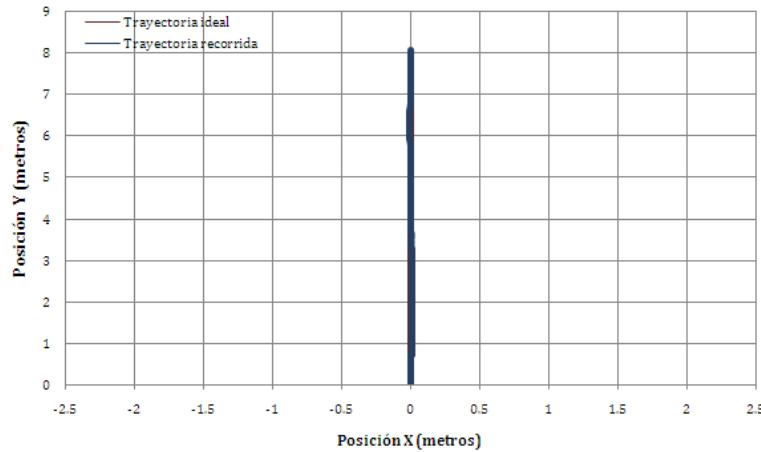


Figura 4.11. Distribución de los sensores de ultrasónicos.

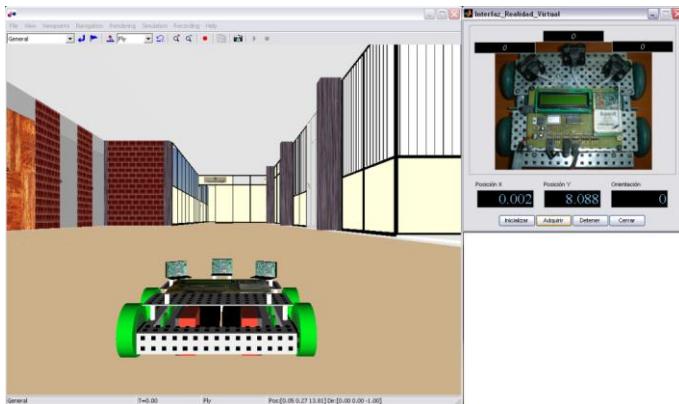
4.4 Comportamiento del robot para recuperación de la trayectoria

Para conocer la capacidad de navegación del robot dentro de su entorno, se evaluaron por separados los comportamientos propuestos en el controlador lógico difuso (ver 3.7), haciendo diferentes experimentos para cada caso, sin embargo con fines de documentación de éste trabajo sólo se presentan tres pruebas para ilustrar cada comportamiento del robot. El primero de estos comportamientos, consiste en la recuperación de trayectoria y el segundo, en la evasión de objetos, aunque durante la navegación ambos comportamientos son unificados y pueden ser tratados como un tercer comportamiento.

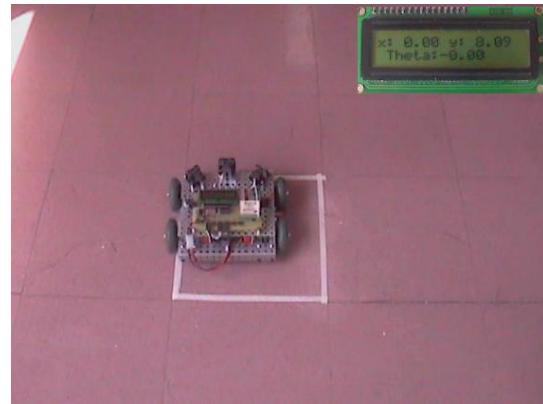
Para el primer comportamiento se realizaron tres experimentos, se estableció el punto de origen en ambos entornos (real/virtual) y se propuso que la meta estuviera a 8 metros a partir de éste y en línea recta. En la primera prueba, el robot fue puesto en el origen y en dirección hacia la meta (acorde a la odometría un ángulo de 0°), el seguimiento se hizo a través de la realidad virtual, obteniéndose el resultado de la Figura 4.12.



a) Trayectorias ideal y recorrida para alcanzar la meta con el robot puesto en el origen.



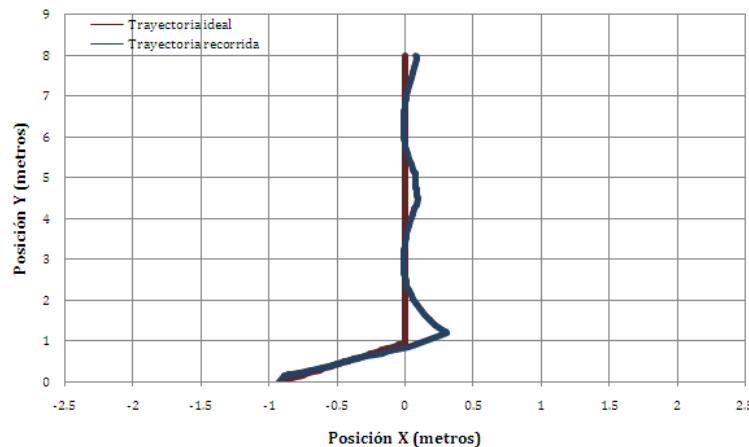
b) Posición final del robot dentro del entorno virtual.



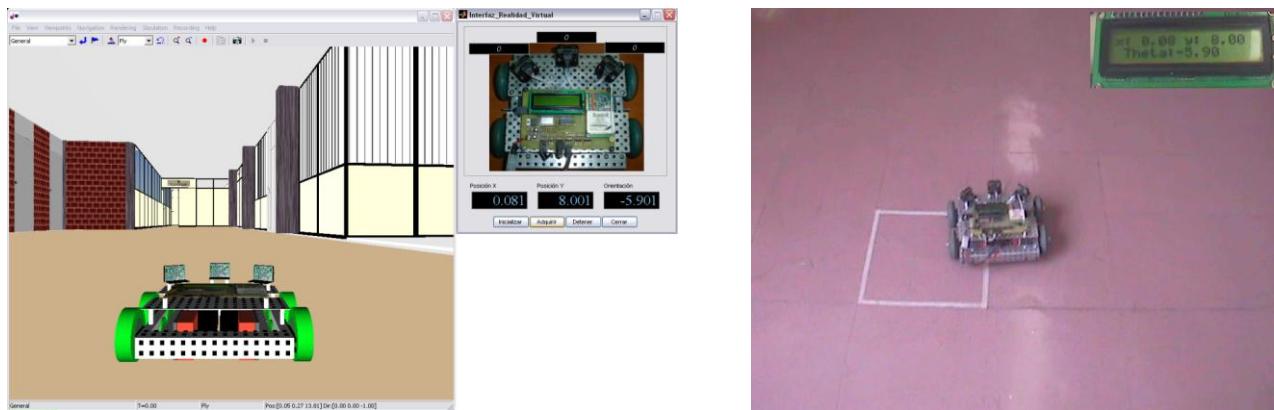
c) Posición final del robot dentro del entorno real.

Figura 4.12. Comportamiento del robot para recuperar su trayectoria y alcanzar la meta con salida en el origen.

En el segundo experimento el robot fue puesto a 0.92 metros hacia la izquierda del origen y orientado con un ángulo de 0° . En la Figura 4.12 se muestra el comportamiento del robot bajo esta condición.



- a) Trayectorias ideal y recorrida para alcanzar la meta con el robot puesto a -0.92 metros del origen.

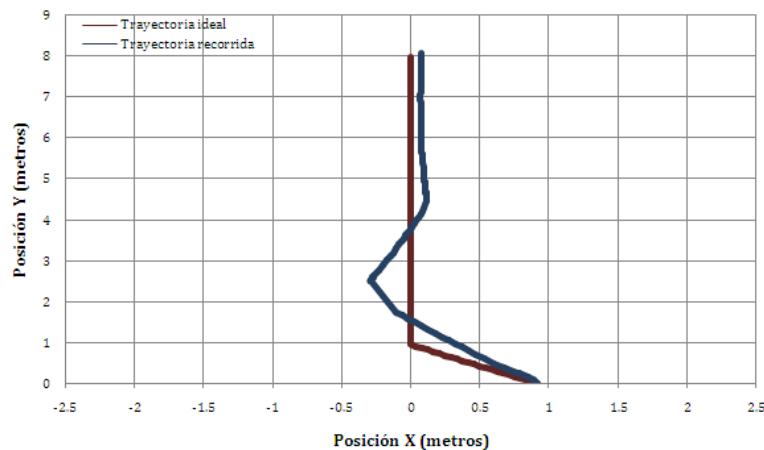


- b) Posición final del robot dentro del entorno virtual.

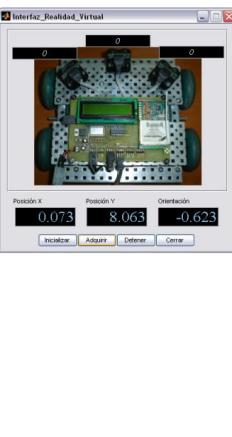
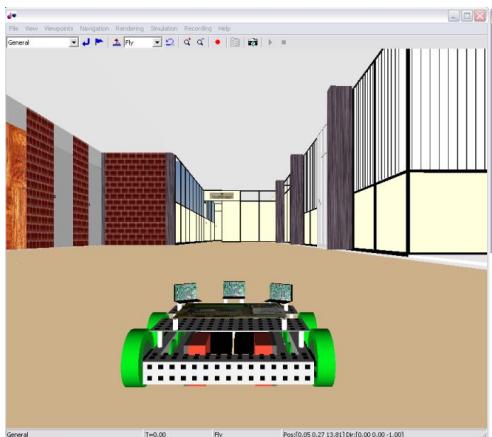
- c) Posición final del robot dentro del entorno real.

Figura 4.13. Comportamiento del robot para recuperar su trayectoria y alcanzar la meta con salida a -0.92 metros del origen.

El último experimento consistió en poner el robot a 0.92 metros hacia la derecha del origen y orientado a 0°. La Figura 4.14 ilustra el comportamiento del robot para recuperar su trayectoria para alcanzar la meta.



- a) Trayectorias ideal y recorrida para alcanzar la meta con el robot puesto a 0.92 metros del origen.



b) Posición final del robot dentro del entorno virtual.

c) Posición final del robot dentro del entorno real.

Figura 4.14. Comportamiento del robot para recuperar su trayectoria y alcanzar la meta con salida a 0.92 metros del origen.

4.5 Comportamiento del robot para evadir objetos durante su navegación y recuperar la trayectoria

Como se mencionó anteriormente el controlador lógico difuso unifica los comportamientos de evasión de objetos y recuperación de trayectorias. Para evaluar este tercer comportamiento se realizaron tres experimentos, para ello se fijó el punto de partida en ambos entornos, así como la meta a 8 metros de distancia con respecto al punto de origen. La Figura 4.15 presenta el posicionamiento de objetos para la evasión y recuperación de trayectoria.



a) Primer prueba.



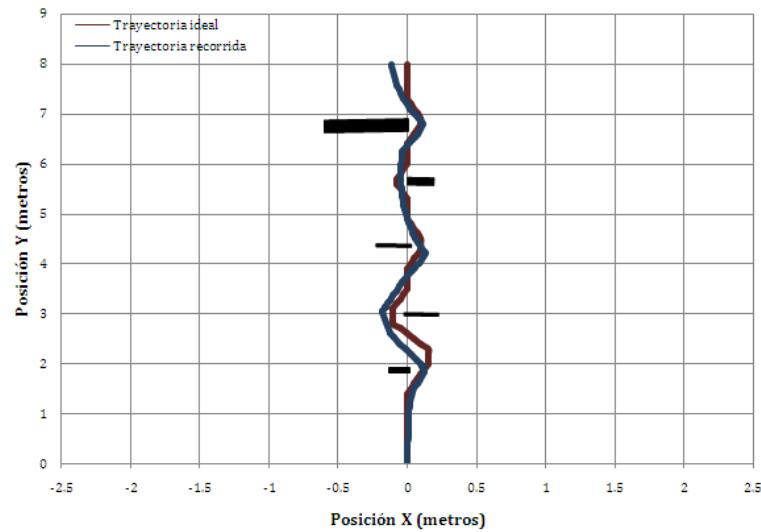
b) Segunda prueba.



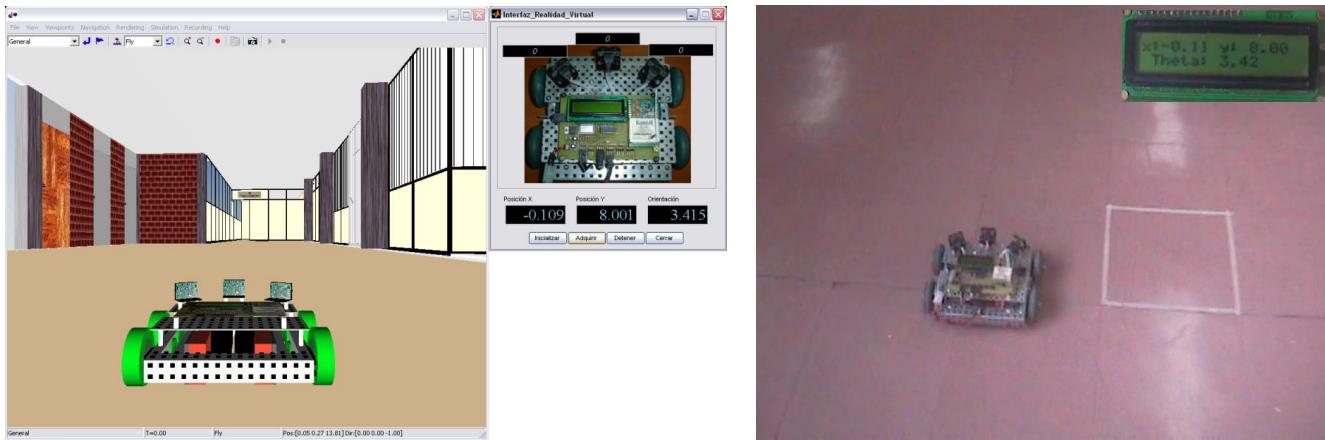
c) Tercer prueba.

Figura 4.15. Posicionamiento de objetos en la de evasión y recuperación de trayectoria.

En la primera de las pruebas se pusieron diferentes objetos sobre la trayectoria de navegación tal como lo muestra la Figura 4.15a. La razón de ponerlos de esta forma obedece a que todos estos objetos obstruyen directamente el paso del robot sobre la trayectoria recta e ideal para alcanzar la meta. La navegación del robot bajo estas condiciones presentes en su entorno se presenta en la Figura 4.16.

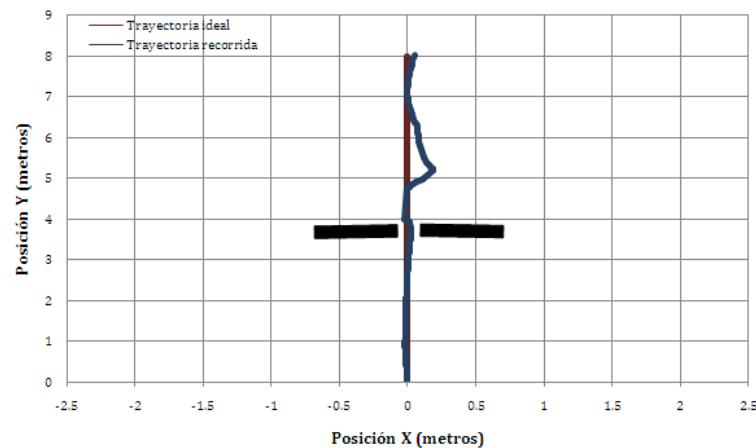


a) Trayectorias ideal y recorrida para evadir objetos y recuperar su trayectoria en la primera prueba.

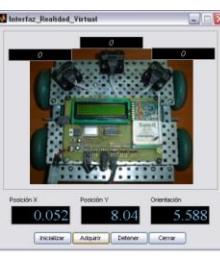
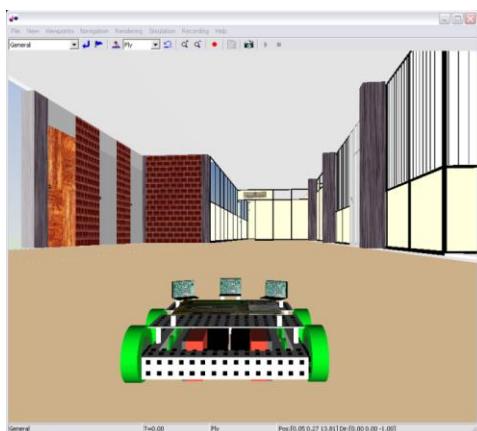


b) Posición final del robot dentro del entorno virtual. c) Posición final del robot dentro del entorno real.
Figura 4.16. Comportamiento del robot para evadir objetos y recuperar su trayectoria, en la primer prueba.

En la segunda prueba se utilizaron dos objetos con dimensiones superiores a las del robot, y se pusieron de tal forma que obstruyeran el paso del robot como lo muestra la Figura 4.15b. Tomando en cuenta que el robot dispone de tres sensores de ultrasonido en su parte frontal, y si los sensores detectan objetos muy cercanos puede interpretarse como la presencia de un objeto considerablemente grande, haciendo que la lógica difusa indique la detención del robot. Fue necesario entonces separar ambos objetos 40 cms uno con respecto al otro y hacer que el robot alcanzara la meta bajo esta condición, observándose el comportamiento mostrado en la Figura 4.17.



a) Trayectorias ideal y recorrida para evadir objetos y recuperar su trayectoria en la segunda prueba.

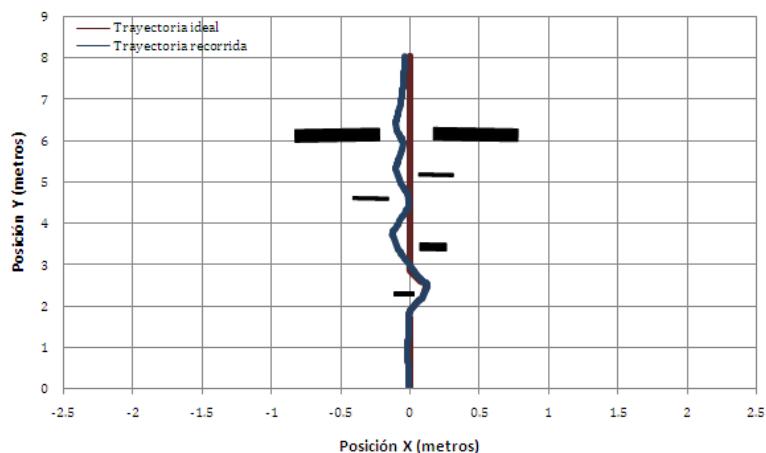


b) Posición final del robot dentro del entorno virtual.

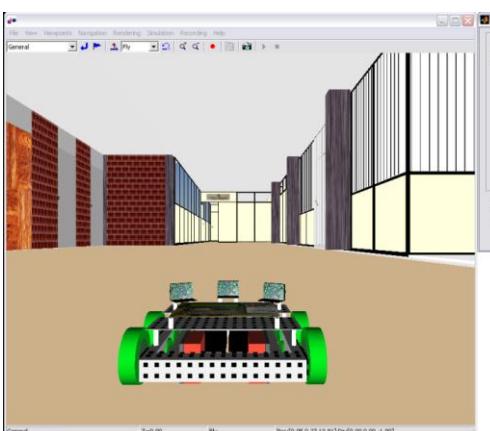
c) Posición final del robot dentro del entorno real.

Figura 4.17. Comportamiento del robot para evadir objetos y recuperar su trayectoria, en la segunda prueba.

En la tercera prueba los objetos se distribuyeron como lo ilustra la Figura 4.15c. El comportamiento del robot en base a esta distribución de objetos es ilustrado en la Figura 4.18.



a) Trayectorias ideal y recorrida para evadir objetos y recuperar su trayectoria en la tercera prueba.



b) Posición final del robot dentro del entorno virtual.

c) Posición final del robot dentro del entorno real.

Figura 4.18. Comportamiento del robot para evadir objetos y recuperar su trayectoria, en la tercera prueba.

Finalmente, para observar la navegación del robot dentro de los entornos real y virtual ante la presencia de objetos, en el disco adjunto a este trabajo se proporcionan las secuencias de video para ambos entornos.

4.6 Discusión de resultados

Con base a los resultados obtenidos en la evaluación de los comportamientos, se observó que en todas las pruebas el robot tuvo la capacidad de alcanzar la meta establecida. Para lograrlo fue de vital importancia la recuperación de la trayectoria, aun bajo la presencia de objetos existentes en su entorno de navegación. Debe aclararse que el entorno virtual es un medio remoto para dar seguimiento al robot durante su navegación dentro del entorno real por lo que no existe ningún error con respecto a las mediciones realizadas por el robot, ya que es ésta información la que se utiliza para ir actualizando la posición del robot virtual dentro del entorno virtual. El error como pudo observarse en las pruebas anteriores existe porque el robot no tuvo la exactitud para llegar a la meta definida. Cuantitativamente, la Tabla 4.4 presenta la posición de la meta que se definió y los resultados obtenidos para cada prueba.

Tabla 4.4. Resultados obtenidos.

	Meta			Recuperación de trayectoria			Evasión de objetos y recuperación de trayectoria		
	Posición X (metros)	Posición Y (metros)	Orientación (grados)	Posición X (metros)	Posición Y (metros)	Orientación (grados)	Posición X (metros)	Posición Y (metros)	Orientación (grados)
Prueba 1	0.0	8.0	0.0	0.002	8.088	0.0	-0.109	8.001	3.415
Prueba 2	0.0	8.0	0.0	0.081	8.001	-5.901	0.052	8.04	5.588
Prueba 3	0.0	8.0	0.0	0.073	8.063	-0.623	-0.038	8.062	-0.621

Dada la naturaleza de estas pruebas, al ser diferentes no es posible cuantificar y generalizar el error del robot para alcanzar una meta. Resulta conveniente realizar varios experimentos para cada una de las pruebas y determinar la repetibilidad que se obtiene en cada caso [3]. La tabla 4.5 presenta la repetibilidad obtenida para cada una de las pruebas realizadas.

Tabla 4.5. Repetibilidad de las pruebas realizadas.

	Recuperación de trayectoria		Evasión de objetos y recuperación de trayectoria	
	Posición X	Posición Y	Posición X	Posición Y
Prueba 1	± 0.086 %	± 0.105 %	± 0.206 %	± 0.051 %
Prueba 2	± 0.027 %	± 0.237 %	± 0.105 %	± 0.053 %
Prueba 3	± 0.25 %	± 0.142 %	± 0.096 %	± 0.055 %

4.7 Conclusiones

Este capítulo permitió conocer la forma en la que los motores están distribuidos en un robot basado en configuración diferencial, y la forma en que dicha distribución influye sobre el sentido de giro de los motores, así como el tipo de desplazamiento que puede generar. Se obtuvo la función de transferencia para cada motor, observándose que pueden generar una velocidad máxima de 170 RPM, aunque se consideró que para la navegación del robot una velocidad máxima de 50 RPM era conveniente. Posteriormente se hizo el estudio de los sensores de ultrasónicos, analizándose su cono máximo de medición, así como su capacidad para detectar objetos con diferentes formas geométricas. Finalmente se llevaron cabo las pruebas de navegación del robot, en el entorno real y se fue monitoreando el desenvolvimiento del robot a través de la realidad virtual. Dichas pruebas permitieron ponderar la capacidad y autonomía del robot, ante la presencia de objetos evadiéndolos, así como la virtud de poder recuperar su trayectoria y permitir con esto alcanzar una meta establecida.

4.8 Referencias

- [1] Thomas Bräunl. "Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems". School of Electrical, Electronic and Computer Engineering. University of Western Australia. 2006.
- [2] Dennis Clark and Michael Owings. "Building Robot Drive Trains". TAB Robotics. 2003.
- [3] Antonio Creus Sole. "Instrumentación Industrial". 7^a Edición. Marcombo. 2005.

Capítulo 5

Conclusiones

5.1 Conclusiones generales

A diferencia de un sistema de seguimiento basado bidimensional, la realidad virtual proporcionó una representación gráfica más clara del comportamiento del robot móvil ante la presencia de objetos existentes en su entorno real de navegación. Si se toma en cuenta que el seguimiento puede ser realizado desde una posición remota donde no es posible utilizar alguna cámara resulta de gran ayuda utilizar tecnología como ésta.

La plataforma diferencial del robot utilizado permitió la estimación de la posición sólo basada en la información proporcionada por los encoders y expresiones trigonométricas básicas. Sin embargo, dada la naturaleza de la odometría de un robot con configuración diferencial, ésta trae consigo la presencia de errores que se van incrementando constantemente (debido a la incorporación de las condiciones iniciales para estimar la nueva posición del robot).

Haber utilizado Lógica Difusa como técnica para dotar de inteligencia y autonomía al robot móvil trajo consigo ventajas sobre otras técnicas. Permitió llevar a cabo la lógica que un ser humano utilizaría para alcanzar un objetivo lo más pronto posible, para lograrlo solo se requirió evaluar dos comportamientos, la posición de la meta y la presencia de objetos en su trayectoria, Aunando a lo anterior, con el desarrollo de este trabajo se observó que la implementación de un controlador difuso es simple y no requiere de cálculos muy complejos es sus tres etapas (fuzzificación, evaluación de las reglas de control y defuzzificación), permitiendo entre otras cosas ser implementado a nivel de hardware sin consumir tantos recursos y tiempo de ejecución, poder expandirse tanto como sea requerido, sólo incorporando más comportamientos que doten de mayor inteligencia a un robot.

Con los experimentos realizados se demuestra que el controlador difuso propuesto proporciona resultados satisfactorios en las pruebas de navegación del robot, en todos los casos el robot fue capaz de llegar a la meta. Aunque en algunos casos existió un pequeño error, éste se considera como producto de agentes externos al controlador difuso, es decir irregularidades en el piso, derrapamiento del robot, así como la presencia del tren de engranes propio del diseño mecánico del robot.

5.2 Recomendaciones y trabajo futuro

Este trabajo puede crecer en su capacidad funcional u operacional, por ende proporcionar un mayor apoyo en el área de experimentación o cualquier otra área donde sea utilizado como referencia, por ser un sistema abierto a posibles mejoras se puede recomendar lo siguiente:

Aunque la plataforma del robot obedece a una configuración diferencial a pesar de disponer de cuatro llantas, se considera pertinente utilizar la configuración básica, es decir, dos llantas para desplazarse y una tercera de soporte. También se sugiere evitar el uso de engranes que influyan en el desplazamiento, así como utilizar motores de DC y no servos motores de rotación continua.

Se considera necesario utilizar los encoders como un medio para controlar la velocidad y hardware especial que permita estimar la posición global del robot, de tal forma que la autonomía del robot no esté limitada y supeditada a un par de encoders.

Disponer de un mayor número de sensores de ultrasonido con un cono de medición angosto que proporcionen al robot una mayor cobertura de su entorno de navegación, así como la detección de objetos cuando estos presentan un grado de rotación con respecto al robot.

La incorporación de más sensores trae consigo modificaciones del controlador lógico difuso, siendo entonces necesario utilizar algún dispositivo con mayor capacidad de procesamiento para realizar las operaciones de las etapas del controlador difuso.

Como trabajo futuro, se sugiere hacer la reconstrucción tridimensional en tiempo real de los objetos encontrados durante la navegación utilizando técnicas de procesamiento digital de señales o el hardware apropiado e incorporarlos en el entorno virtual.

Diagrama de estados-eventos de la interfaz gráfica de usuario

La figura A-1 muestra la sucesión de estados y eventos que se suscitan durante la ejecución y manipulación de la interfaz gráfica de usuario implementada. Como se mencionó anteriormente ésta interfaz se encarga de adquirir la información de la posición del robot dentro de su entorno real y en base a dichos datos actualizar la posición del robot virtual dentro del entorno virtual.

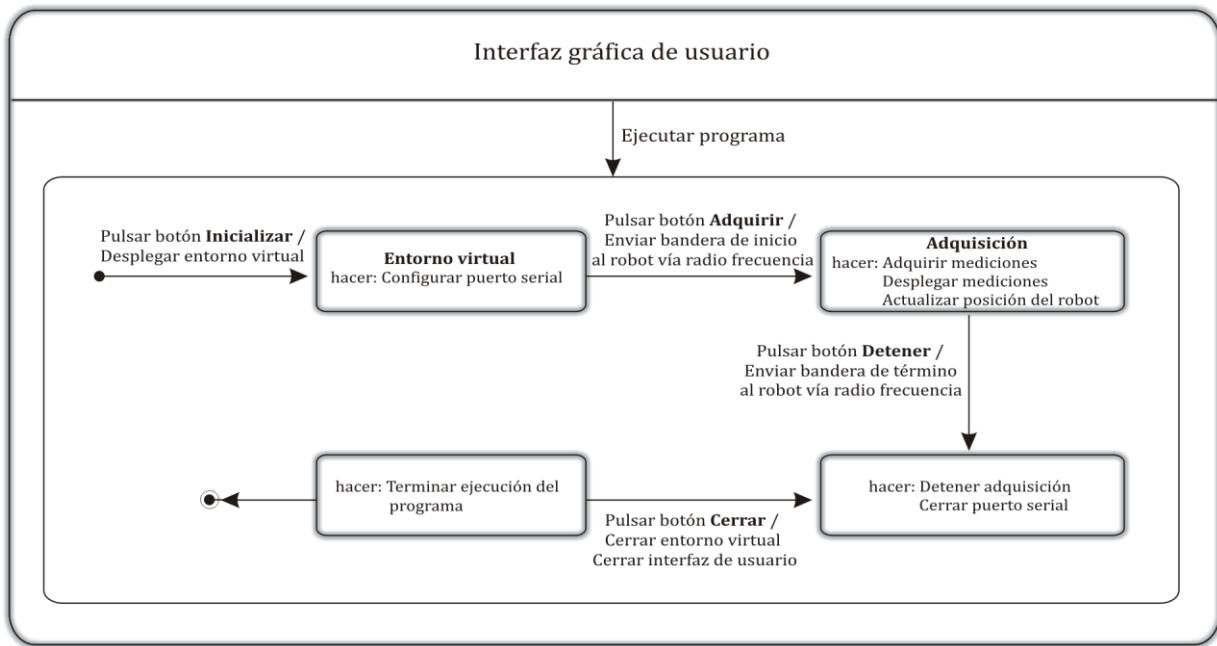
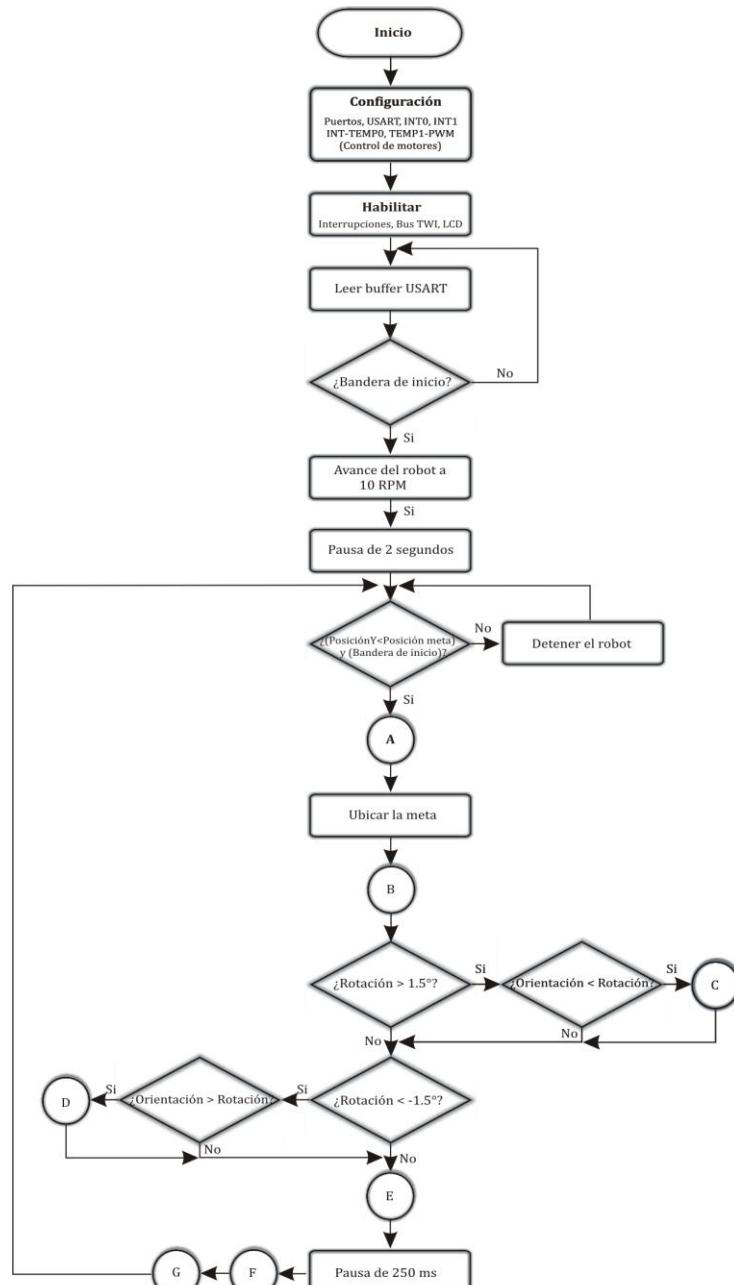


Figura A-1. Diagrama de estados-eventos de la interfaz gráfica de usuario.

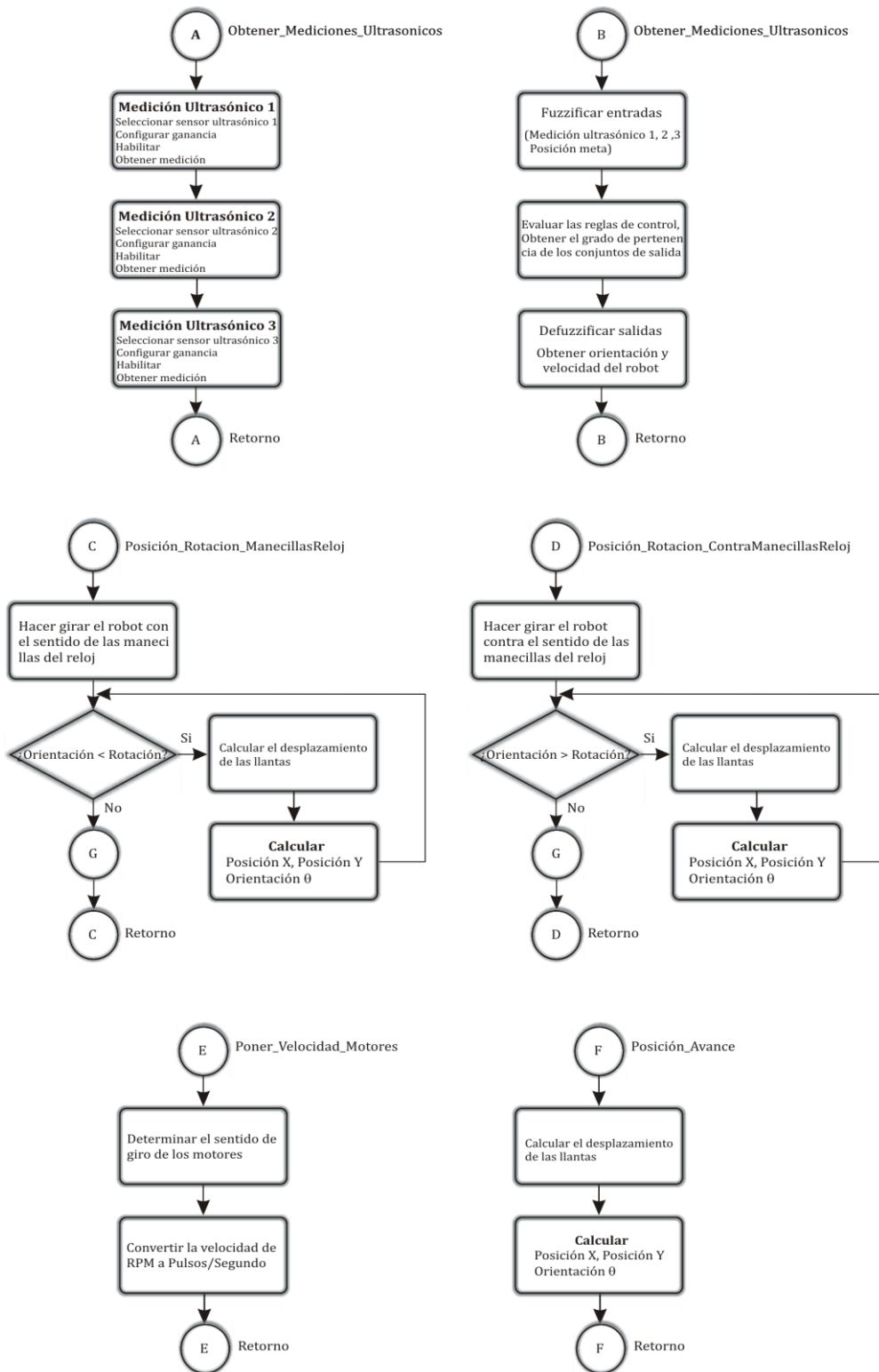
En el disco adjunto a éste trabajo se proporciona el código implementado en el editor M-File de MatLab.

Diagrama de flujo del programa del módulo de control y transmisión

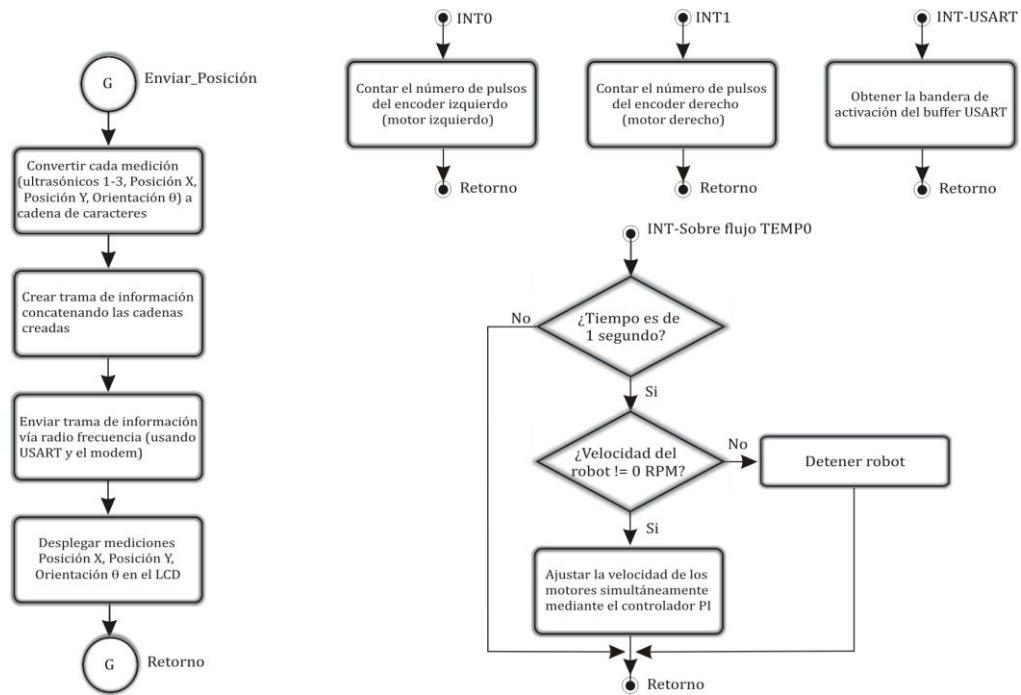
La figura B-1 muestra el diagrama de flujo del programa del módulo de control y transmisión de la posición del robot.



a) Rutina principal.



b) Rutinas auxiliares.



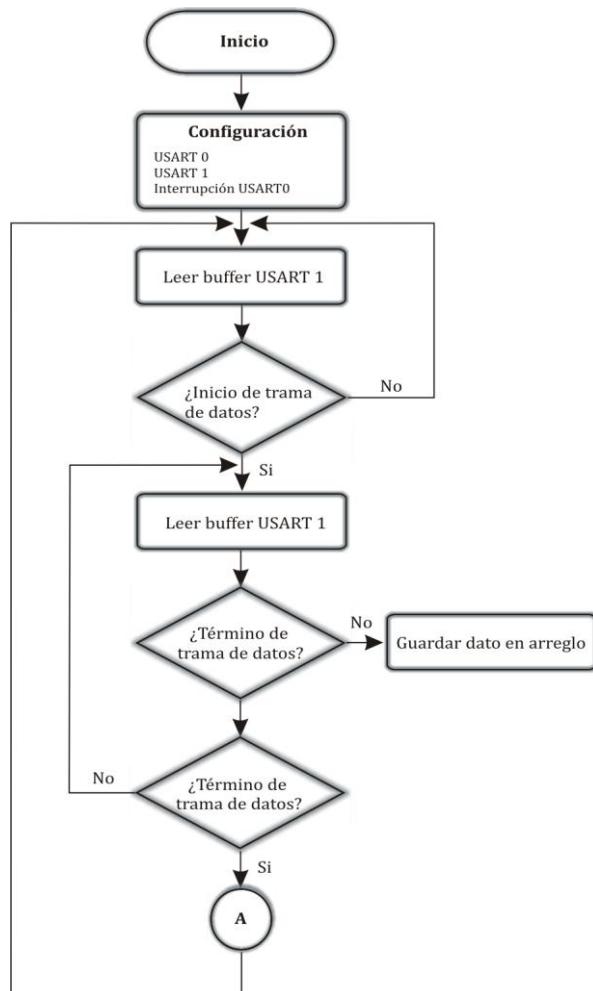
c) Rutinas de las interrupciones.

Figura B-1. Diagrama de flujo del programa del módulo de control y transmisión.

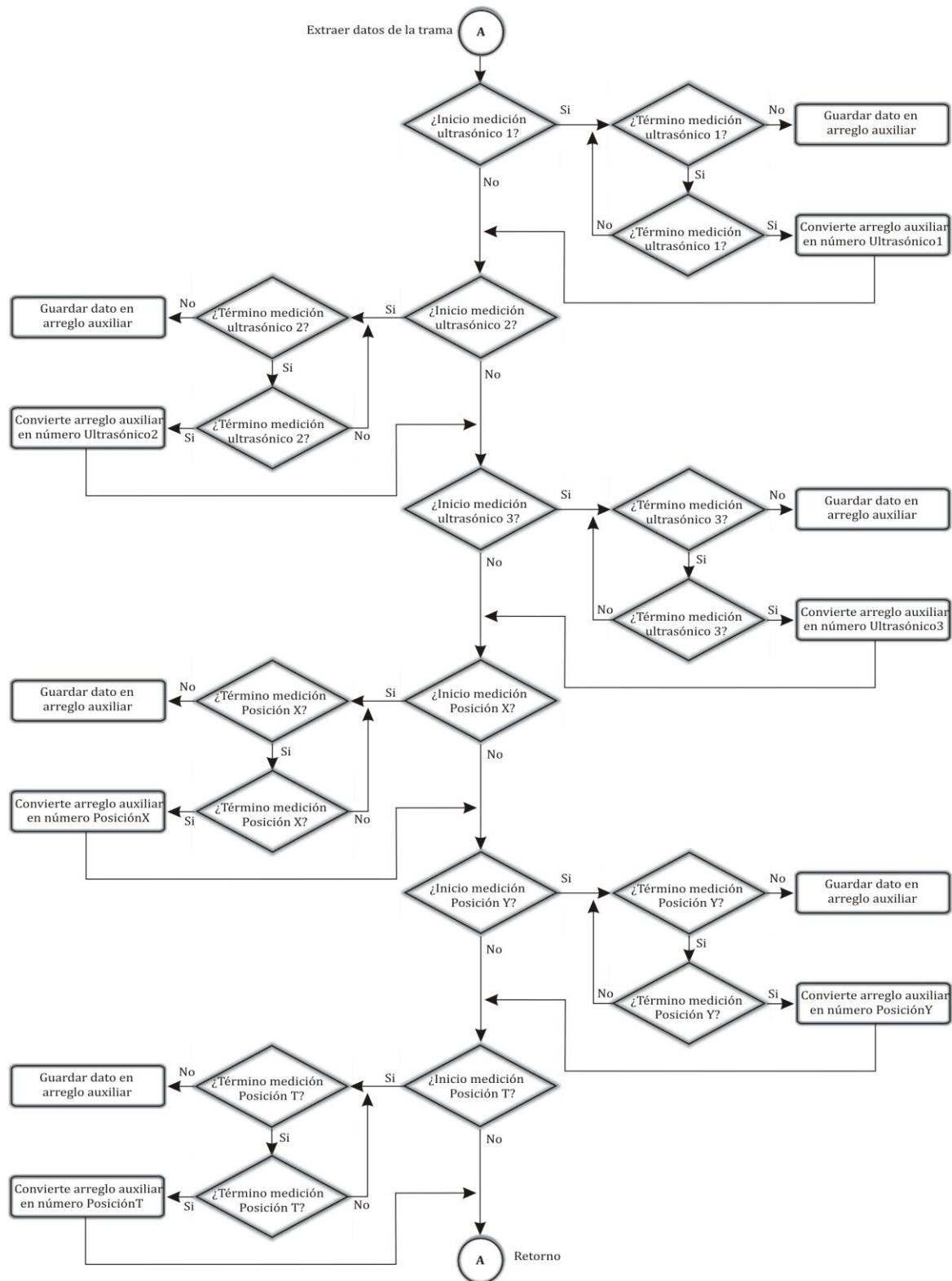
En el disco adjunto a éste trabajo se proporciona el código implementado en Lenguaje C utilizando el compilador AVR Studio 4.13.

Diagrama de flujo del programa del módulo de recepción

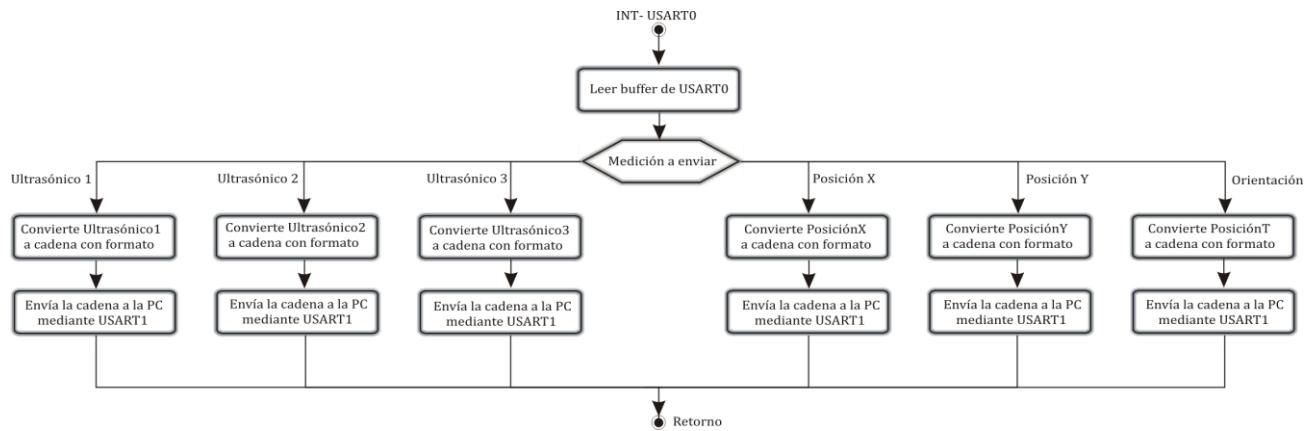
La figura C-1 muestra el diagrama de flujo del programa del módulo de recepción de la posición del robot.



a) Rutina principal.



b) Rutina auxiliar.



c) Rutina de la interrupción.

Figura C-1. Diagrama de flujo del programa del módulo de recepción.

En el disco adjunto a éste trabajo se proporciona el código implementado en Lenguaje C utilizando el compilador AVR Studio 4.13.