



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN Y DESARROLLO
DE TECNOLOGÍA DIGITAL



MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES

“DISEÑO DE UN CONTROLADOR DIFUSO EN FPGA APLICADO A
ROBÓTICA MÓVIL”

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS EN SISTEMAS DIGITALES

PRESENTA:

ING. JESÚS ANTONIO CAMACHO GONZÁLEZ

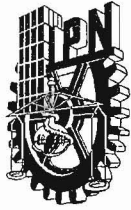
BAJO LA DIRECCIÓN DE:

DR. OSCAR H. MONTIEL ROSS

DR. ROBERTO SEPÚLVEDA CRUZ

MAYO DE 2010

TIJUANA, B.C., MÉXICO



INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de Tijuana, B.C. siendo las 12:00 horas del día 18 del mes de junio del 2010 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CITEDI para examinar la tesis de grado titulada:

DISEÑO DE UN CONTROLADOR DIFUSO EN FPGA APLICADO A ROBÓTICA MÓVIL.

Presentada por el alumno:

CAMACHO

Apellido paterno

GONZÁLEZ

materno

JESÚS ANTONIO

nombre(s)

Con registro:

B0	7	1	2	2	5
----	---	---	---	---	---

aspirante al grado de:

MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES

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

LA COMISION REVISORA

Director de tesis


DR. ROBERTO SEPULVEDA CRUZ

Director de tesis


DR. OSCAR HUMBERTO MONTIEL ROSS


DR. LUIS TUPAK AGUILAR BUSTOS


DR. JUAN JOSE TAPIA ARMENTA


DR. OSCAR CASTILLO LOPEZ

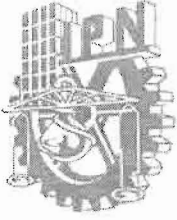
EL PRESIDENTE DEL COLEGIO


DR. LUIS ARTURO GONZÁLEZ HERNÁNDEZ



S. E. P.
INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACION Y DESARROLLO
DE TECNOLOGÍA DIGITAL
DIRECCION

INSTITUTO POLITÉCNICO NACIONAL



SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de Tijuana, Baja California, el día 21 del mes Junio del año 2010, el (la) que suscribe Jesús Antonio Camacho González

Alumno (a) del Programa TESIS DE MAESTRIA EN CIENCIAS EN SISTEMAS DIGITALES, con número de registro B071225, adscrito al CENTRO DE INVESTIGACIÓN Y DESARROLLO DE TECNOLOGÍA DIGITAL, manifiesta que es autor (a) intelectual del presente trabajo de Tesis, bajo la dirección de Dr. Oscar H. Montiel Ross y Dr. Roberto Sepúlveda Cruz

Y cede los derechos del trabajo intitulado Diseño de un Controlador Difuso en FPGA aplicada a Robótica Móvil.

Al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo, Este puede ser obtenido escribiendo a la siguiente dirección: Av. Del Parque 1310, Mesa de Otay, Tijuana, Baja California, México CP 22510. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Jesús A. Camacho
Jesús Antonio Camacho González

Nombre y firma

Dedicatoria

A mis padres Antonia González y Raymundo Camacho por ser lo que más quiero en esta vida, a mis hermanos Juan Pablo y Ana Emilia por que siempre van estar ahí para apoyarme, a mi hermano Raymundo que en donde quiera que se encuentre su presencia siempre me estará acompañando a donde vaya, y a todas aquellas personas que me han apoyado para lograr culminar una meta más en mi carrera.

Agradecimientos

A mis padres Raymundo Camacho y Antonia González por todo su cariño y apoyo incondicional que me han brindado toda mi vida para poder lograr todos los objetivos que me proponga y a mi familia que siempre han estado al pendiente de mí brindandome todos sus buenos deseos para que logre todo lo que me proponga.

A mis directores de tesis Dr. Oscar H. Montiel Ross y Dr. Roberto Sepúlveda Cruz por todas sus enseñanzas, por la paciencia que me tuvieron, por los consejos que me brindaron para realizar un buen trabajo de tesis y sobre todo por dejarme ser parte de la historia del Laboratorio de Sistemas Inteligentes.

A mis amigos Diego, Gabriel e Ismael por hacerme sentir en familia desde el primer día que llegué a Tijuana y a todos mis amigos por todos esos momentos que compartimos juntos dentro y fuera de CITEDI.

Al Centro de Investigación y Desarrollo de Tecnología Digital(CITEDI) por dejarme ser parte de esta gran institución, que es el Instituto Politécnico Nacional (IPN), y al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado para cumplir un escalón más en mi preparación profesional.

Diseño de un Controlador Difuso en FPGA aplicado a Robótica Móvil

Resumen

Este trabajo de tesis aborda el problema de regulación de postura de un robot móvil con ruedas no-holonómico que emplea un sistema de locomoción con direccionamiento diferencial, por lo cual se proponen dos sistemas difusos que se enfocan en controlar la posición y orientación del robot móvil con ruedas para cumplir con el objetivo de control propuesto. Estos sistemas difusos fueron codificados en el lenguaje de programación VHDL para ser implementados en un FPGA.

Los sistemas difusos propuestos en este trabajo de tesis se integraron en un sistema de control de movimiento del robot móvil con ruedas desarrollado en el ambiente de programación Simulink. De manera que se muestran los resultados obtenidos al realizar una serie de simulaciones computacionales donde se llevan a cabo las comparaciones entre los sistemas difusos desarrollados en Matlab y VHDL. Para realizar la simulación en el entorno Simulink de los sistemas difusos codificados en VHDL se utilizó la herramienta Xilinx System Generator.

Palabras clave: Problema de regulación de postura, robot móvil con ruedas, sistema difuso, VHDL, FPGA, Matlab, Simulink, Xilinx System Generator.

Design of a Fuzzy Controller in FPGA applied to Mobile Robotics

Abstract

This thesis addresses the posture regulation problem of a wheeled mobile robot nonholonomic system which uses a differential drive type of locomotion, so there were proposed two fuzzy systems focused on controlling the position and orientation of mobile robot wheeled to meet the control objective. These fuzzy systems were coded in VHDL programming language to be implemented in an FPGA.

The fuzzy systems proposed in this thesis, are integrated into a motion control system of wheeled mobile robot developed in Simulink programming environment. It is shown the results and comparisons obtained by performing a series of computer simulations which are carried out between fuzzy systems developed in Matlab and VHDL. To perform the simulation in Simulink environment for fuzzy systems coded in VHDL, there was used the Xilinx System Generator tool.

Keywords: Posture regulation problem, wheeled mobile robot, fuzzy system, VHDL, FPGA, Matlab, Simulink, Xilinx System Generator.

Contenido

Resumen	VII
Abstract	VIII
Contenido	IX
Lista de Figuras	XI
Lista de Tablas	XIV
Lista de símbolos y acrónimos	XV
1. Introducción	1
Introducción	1
1.1. Objetivo General	3
Objetivo General	3
1.1.1. Objetivos Específicos	3
1.2. Aportaciones del Trabajo	4
1.3. Organización de la Tesis	4
2. Marco Teórico	5
2.1. Sistema de Control Difuso	5
2.1.1. Fuzzificación	6
2.1.2. Inferencia	11
2.1.3. Defuzzificación	14
2.2. VHDL	16
2.2.1. Flujo de Datos	18
2.2.2. Comportamiento o Funcional	19
2.2.3. Estructural	19
2.3. FPGA's	20
2.3.1. Ventajas de los FPGA's	21
3. Sistema Difuso en VHDL para el Control de Movimiento de un Robot Móvil con Ruedas	24
3.1. Modelo Cinemático	24
3.2. Sistema de Control de Movimiento del Robot Móvil con Ruedas	26

3.3.	Control de Regulación de Postura	28
3.3.1.	Sistema Difuso de Posición	29
3.3.2.	Sistema Difuso de Orientación	37
3.4.	Codificación en VHDL del Control de Regulación de Postura	39
3.4.1.	Codificación en VHDL del Sistema Difuso de Posición	40
3.4.2.	Codificación en VHDL del Sistema Difuso de Orientación	41
4.	Experimentos	43
4.1.	Prueba del Sistema Difuso de Posición	44
4.1.1.	Prueba del Sistema Difuso de Posición con Salidas Velocidad Lineal y Velocidad Angular	45
4.1.2.	Prueba del Sistema Difuso de Posición con Salidas Velocidad An- gular Izquierda y Velocidad Angular Derecha	47
4.2.	Prueba del Sistema Difuso de Orientación	50
4.2.1.	Prueba del Sistema Difuso de Orientación con Salida Velocidad An- gular	51
4.2.2.	Prueba del Sistema Difuso de Orientación con Salidas Velocidad Angular Izquierda y Velocidad Angular Derecha	54
4.3.	Prueba del Sistema de Control de Movimiento del Robot Móvil con Ruedas	57
4.3.1.	Resultados Experimentales	58
5.	Conclusiones	75
5.1.	Conclusiones	75
5.2.	Trabajos Futuros	76
	Referencias	77

Lista de Figuras

2.1. Estructura básica de un sistema de control difuso.	6
2.2. Función de membresía triangular.	8
2.3. Función de membresía trapezoidal.	9
2.4. Función de membresía gaussiana.	10
2.5. Función de membresía campana.	10
2.6. Función de membresía sigmoideal.	11
2.7. Mecanismo de inferencia <i>max – min</i>	13
2.8. Centro de Gravedad.	15
2.9. Criterio del Máximo: a)Único valor, b)Intervalo de valores.	16
2.10. Método de las Alturas.	16
2.11. Modelo de hardware desarrollado en VHDL.	17
2.12. Esquema del modelado flujo de datos.	18
2.13. Esquema del modelado de comportamiento o funcional.	19
2.14. Esquema del modelado estructural.	19
2.15. Arquitectura básica de un FPGA.	21
3.1. Robot Móvil con Ruedas.	25
3.2. Problema de Regulación de Postura.	27
3.3. Sistema de control de movimiento del robot móvil con ruedas.	28
3.4. Etapa de control de potencia del motor de CD utilizando PWM y puente H.	28
3.5. Variable lingüística distancia.	30
3.6. Variable lingüística alfa.	31
3.7. Variable lingüística velocidad lineal.	32
3.8. Variable lingüística velocidad angular.	33
3.9. Variable lingüística velocidad angular izquierda.	35
3.10. Variable lingüística velocidad angular derecha.	36
3.11. Variable lingüística error theta.	38
3.12. Entidad del sistema de control difuso de posición en VHDL con salidas velocidad lineal y velocidad angular.	40
3.13. Entidad del sistema de control difuso de posición en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.	41

3.14. Entidad del sistema difuso de orientación en VHDL con salida velocidad angular.	42
3.15. Entidad del sistema difuso de orientación en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.	42
4.1. Simulación del sistema difuso de posición en VHDL con salidas velocidad lineal y velocidad angular.	45
4.2. Superficies de control del sistema difuso de posición en VHDL con salidas velocidad lineal y velocidad angular.	46
4.3. Simulación del sistema difuso de posición en Matlab-Simulink con salidas velocidad lineal y velocidad angular.	46
4.4. Superficies de control del sistema difuso de posición en Matlab-Simulink con salidas velocidad lineal y velocidad angular.	47
4.5. Error de superficies de control del sistema difuso de posición con salidas velocidad lineal y velocidad angular codificado en VHDL y en Matlab-Simulink.	48
4.6. Simulación del sistema difuso de posición en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.	48
4.7. Superficies de control del sistema difuso de posición en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.	49
4.8. Simulación del sistema difuso de posición en Matlab-Simulink con salidas velocidad angular izquierda y velocidad angular derecha.	49
4.9. Superficies de control del sistema difuso de posición en Matlan-Simulink con salidas velocidad angular izquierda y velocidad angular derecha.	50
4.10. Error de superficies de control del sistema difuso de posición con salidas velocidad angular izquierda y velocidad angular derecha codificado en VHDL y en Matlab-Simulink.	51
4.11. Simulación del sistema difuso de orientación en VHDL con salida velocidad angular.	52
4.12. Superficie de control del sistema difuso de orientación en VHDL con salida velocidad angular.	52
4.13. Simulación del sistema difuso de orientación en Matlab-Simulink con salida velocidad angular.	53
4.14. Superficie de control del sistema difuso de orientación en Matlab-Simulink con salida velocidad angular.	53
4.15. Error de superficies de control del Sistema difuso de orientación con salida velocidad angular codificado en VHDL y en Matlab-Simulink.	54
4.16. Simulación del sistema difuso de orientación en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.	55
4.17. Superficies de control del sistema difuso de orientación en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.	55

4.18. Simulación del sistema difuso de orientación en Matlab-Simulink con salidas velocidad angular izquierda y velocidad angular derecha.	56
4.19. Superficies de control del sistema difuso de orientación en Matlab-Simulink con salidas velocidad angular izquierda y velocidad angular derecha.	56
4.20. Error de superficies de control del sistema difuso de orientación con salidas velocidad angular izquierda y velocidad angular derecha codificado en VHDL y en Matlab-Simulink.	57
4.21. Sistema de Control del Robot Móvil de Ruedas desarrollado en Simulink.	58
4.22. Experimento 1. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.	59
4.23. Experimento 1. Error de posición y orientación del robot móvil de ruedas.	60
4.24. Experimento 2. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.	61
4.25. Experimento 2. Error de posición y orientación del robot móvil de ruedas.	62
4.26. Experimento 3. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.	63
4.27. Experimento 3. Error de posición y orientación del robot móvil de ruedas.	64
4.28. Experimento 4. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.	65
4.29. Experimento 4. Error de posición y orientación de robot móvil de ruedas.	66
4.30. Experimento 5. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.	67
4.31. Experimento 5. Error de posición y orientación del robot móvil de ruedas.	68
4.32. Experimento 6. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.	69
4.33. Experimento 6. Error de posición y orientación del robot móvil de ruedas.	70
4.34. Experimento 7. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.	71
4.35. Experimento 7. Error de posición y orientación del robot móvil de ruedas.	72
4.36. Experimento 8. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.	73
4.37. Experimento 8. Error de posición y orientación de robot móvil de ruedas.	74

Lista de Tablas

3.1. Parámetros de las funciones de membresía de las variables lingüísticas distancia(d) y alfa(α).	31
3.2. Parámetros de las funciones de membresía de las variables lingüísticas velocidad lineal(v_r) y velocidad angular(ω_r).	33
3.3. Base de reglas difusas con respecto a las variables lingüísticas velocidad lineal y velocidad angular.	34
3.4. Parámetros de las funciones de membresía de las variables lingüísticas velocidad angular derecha(v_{dr}) y velocidad angular izquierda(ω_{ir}).	36
3.5. Base de reglas difusas con respecto a las variables lingüísticas velocidad angular izquierda y velocidad angular derecha.	37
3.6. Parámetros de las funciones de membresía de las variable lingüística error theta(e_θ).	37
3.7. Base de reglas difusas con respecto a la variable lingüística velocidad angular.	39
3.8. Base de reglas difusas con respecto a las variables lingüísticas de salida velocidad angular izquierda y velocidad angular derecha.	39

Lista de símbolos y acrónimos

v	Velocidad lineal.
ω	Velocidad angular.
ω_i	Velocidad angular izquierda.
ω_d	Velocidad angular derecha.
ϑ	Vector de velocidades.
\dot{q}	Vector de velocidades generalizadas.
q	Vector de coordenadas generalizadas.
q_d	Posición y orientación deseada.
K_p	Ganancia proporcional.
K_i	Ganancia integral.
e_x	Error de posición en x .
e_y	Error de posición en y .
e_θ	Error de orientación θ .
d	Distancia.
α	Ángulo alfa.
VHDL	Acrónimo VHSIC y HDL.
FPGA	Field Programmable Gate Array.
CPLD	Complex Programmable Logic Device.
XSG	Xilinx System Generator.
DSP	Digital Signal Processor.
CD	Corriente directa.
Matlab	Matrix Laboratory.
COA	Centro de gravedad.
CM	Criterio del máximo.
IEEE	Institute of Electrical and Electronics Engineers.
PI	Controlador Proporcional-Integral.

Capítulo 1

Introducción

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

Un robot móvil necesita un sistema de locomoción que le permita moverse libremente en el espacio en el que se desarrolla. Por ello se desarrollan robots que puedan caminar, brincar, correr, deslizarse, patinar, nadar, volar y rodar [1].

Las ruedas han sido por mucho tiempo el sistema de locomoción más popular en robot móviles siendo la solución más simple y eficiente para conseguir la movilidad en terrenos suficientemente duros y libres de obstáculos, permitiendo conseguir velocidades relativamente altas, pero a su vez presenta limitaciones como el deslizamiento en el arranque y dependiendo de las características que posee el terreno se pueden presentar deslizamientos y vibraciones, de manera que este sistema de locomoción es poco eficiente en terrenos blandos [2].

Entre los robots móviles que emplean diferentes sistemas de locomoción mediante ruedas que les confieren características y propiedades respecto a la eficiencia energética, dimensiones, cargas útiles y maniobrabilidad, se encuentran los tipos: ackerman, triciclo clásico, direccionamiento diferencial, skid steer y síncronas por mencionar algunos [2]. En

este trabajo de tesis se considera un robot móvil con ruedas no-holonómico que emplea un sistema de locomoción con direccionamiento diferencial.

En las últimas décadas el control de robots móviles con ruedas ha sido objeto de interés continuo. Estos se enmarcan dentro de la teoría de control en el área que se conoce como control de sistemas no-holónomos, los cuales se caracterizan por estar sujetos a restricciones no integrables en las velocidades, es decir su plano de velocidades está restringido [3].

El problema de control de movimiento de un robot móvil con ruedas puede ser clasificado en tres problemas básicos [4] : seguimiento de trayectoria, seguimiento de una ruta y regulación de postura. En la literatura se reportan diversos trabajos que se enfocan en resolver estos problemas utilizando técnicas de control inteligente. En [5, 6] se propone un sistema de control difuso que controla el torque aplicado al modelo dinámico de un robot móvil con ruedas para dar solución al problema de regulación de postura. Al igual que en los trabajos mencionados anteriormente, en [4, 7, 8, 9] desarrollan un sistema de control para controlar el modelo dinámico del robot móvil de ruedas pero enfocados en resolver el problema de seguimiento de trayectoria utilizando lógica difusa tipo 2, algoritmos genéticos y redes neuronales. En [10] y [11] se enfocan en desarrollar controladores para controlar el modelo cinemático del robot móvil de ruedas utilizando lógica difusa y redes neuronales respectivamente para resolver el problema de seguimiento de trayectoria. Además se reportan trabajos, como es en el caso de [12] y [13], que se enfocan en resolver el problema de control de movimiento de robot móviles con ruedas dividiéndolo en subtarefas a realizar, de manera que se propone un sistema difuso por cada subtarea a realizar con el fin de lograr el objetivo propuesto.

La propuesta de este trabajo de tesis, es desarrollar sistemas difusos en VHDL para el control de posición y orientación de un robot móvil con ruedas, los cuales son integrados en un sistema de control de movimiento enfocado a resolver el problema de control de regulación de postura. Existen trabajos relacionados como [14, 15], que desarrollan sistemas difusos en VHDL enfocados en resolver el problema de control de movimiento

mediante la información proporcionada por el sistema sensorial, con el objetivo de que el robot móvil de ruedas evite los obstáculos que se le presenten en su camino. Además se reportan diferentes trabajos donde se desarrollan sistemas difusos en VHDL para ser implementados en un FPGA con diferentes fines como es el caso de [16, 17, 18], y en [19, 20, 21, 22] se enfocan en el control de velocidad de un motor de CD.

1.1. **Objetivo General**

Desarrollar sistemas difusos en VHDL para el control de posición y orientación de un robot móvil con ruedas, los cuales se integran en un sistema de control de movimiento que tiene como objetivo resolver el problema de regulación de postura.

1.1.1. **Objetivos Específicos**

- Desarrollar en Matlab-Simulink un sistema difuso para el control de posición de un robot móvil con ruedas.
- Desarrollar en Matlab-Simulink un sistema difuso para el control de orientación de un robot móvil con ruedas.
- Codificar en VHDL el sistema difuso para el control de posición de un robot móvil con ruedas desarrollado en Matlab-Simulink.
- Codificar en VHDL el sistema difuso para el control de orientación de un robot móvil con ruedas desarrollado en Matlab-Simulink.
- Realizar comparaciones entre los sistemas difusos desarrollados en Matlab-Simulink y en VHDL.
- Integrar los sistemas difusos propuestos en un sistema de control de movimiento de un robot móvil con ruedas desarrollado en Matlab-Simulink.

- Realizar diversas pruebas y analizar los resultados.

1.2. Aportaciones del Trabajo

De este trabajo de tesis se derivaron dos capítulos de libro titulados:

- Fuzzy System to Control the Movement of a Wheeled Mobile Robot.
- Embedding a Fuzzy Locomotion Pose Controller for a Wheeled Mobile Robot into an FPGA.

los cuales aparecerán publicados en el libro “Soft Computing for Intelligent Control and Mobile Robotics” de la editorial Springer-Verlag.

1.3. Organización de la Tesis

Este documento está compuesto por cinco capítulos. En el capítulo 2 se presenta el marco teórico que sustenta el desarrollo de este trabajo de tesis, donde se tratan aspectos de control difuso y lenguaje de programación VHDL. En el capítulo 3 se presenta la propuesta de este trabajo de tesis la cual es desarrollar sistemas difusos en VHDL para el control de posición y orientación de un robot móvil con ruedas, los cuales son integrados en un sistema de control de movimiento que se enfoca en resolver el problema de regulación de postura. El capítulo 4 presenta los resultados que se obtienen al hacer las comparaciones de los sistemas difusos desarrollados en Matlab-Simulink y VHDL. Además se muestran los resultados que se obtienen al realizar las simulaciones computacionales del sistema de control de movimiento del robot móvil con ruedas. Finalmente en el capítulo 5 se establecen las conclusiones a partir de los resultados obtenidos y los trabajos futuros.

Capítulo 2

Marco Teórico

2.1. Sistema de Control Difuso

Debido a su naturaleza multidisciplinaria, un sistema de control difuso se conoce por diferentes nombres, tales como sistema difuso basado en reglas, sistema experto difuso, modelo difuso, memoria asociativa difusa, sistema de inferencia difuso, o simplemente sistema difuso [23].

Los sistemas de control difuso se basan en la representación del conocimiento mediante reglas difusas *if – then* y razonamiento difuso, han sido aplicados exitosamente en diversos campos, como en control automático, robótica, sistemas expertos, predicción de series de tiempo, clasificación de datos, reconocimiento de patrones, entre otros [23].

Existen diferentes modelos de sistemas de control difuso, uno de los más conocidos y utilizados es el modelo de Mamdani. El sistema de inferencia difuso de Mamdani fue propuesto en un intento por controlar una máquina de vapor, por medio de un conjunto de reglas de control lingüísticas obtenidas de la experiencia de los operadores humanos [24]. En la Figura 2.1 se muestra la estructura básica de un sistema de control difuso que consta de tres etapas que son fuzzificación, inferencia y defuzzificación.

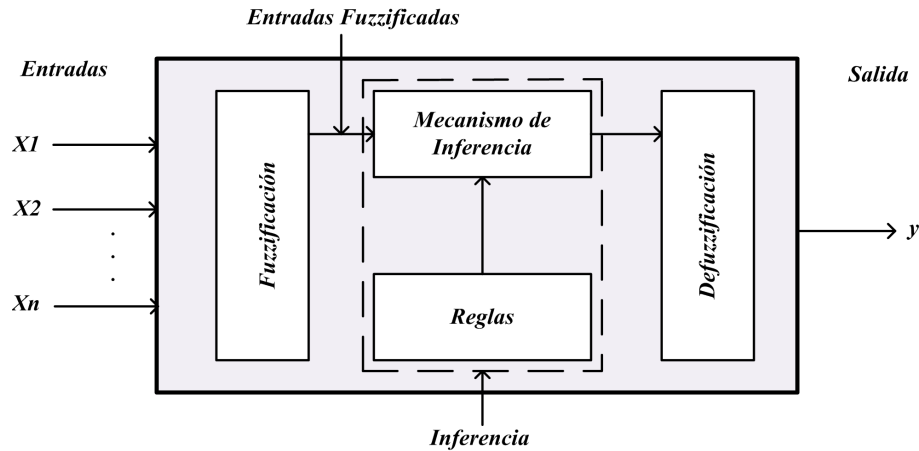


Figura 2.1: Estructura básica de un sistema de control difuso.

2.1.1. Fuzzificación

El proceso de fuzzificación es la interpretación de los datos de entrada por el sistema de control difuso y consiste de dos componentes principales:

- Variable Lingüística
- Funcion de membresía

Variable Lingüística

Una característica esencial en el pensamiento humano es la capacidad para resumir información. Los humanos podemos presentar y manejar datos con términos del lenguaje natural. Fundamentalmente, el razonamiento humano es aproximado más que exacto.

Haciendo una consideración sobre la representación del conocimiento, la unidad fundamental en razonamiento aproximado es el concepto de variable lingüística. Zadeh establece en su obra [25]:

“Por variable lingüística se entiende una variable cuyos valores son palabras u oraciones en un lenguaje natural o artificial. Por ejemplo, Edad es una variable lingüística

si sus valores son lingüísticos en lugar de numéricos: joven, no joven, muy joven, poco joven, viejo, no muy viejo y no muy joven, etc., en lugar de 20, 21, 22, 23,..."

Formalmente, un posible valor de una variable lingüística se representa mediante un conjunto difuso y se le llama *valor o término lingüístico*. La primera decisión que se debe tomar cuando se crea una variable lingüística es escoger el número de términos que se deben definir. La mayoría de las aplicaciones usan entre tres y siete términos lingüísticos. Normalmente se elige un número impar de términos, debido a que se definen simétricamente, donde un término describe la media entre dos extremos. La variable lingüística, permite la traslación de una medida (grados centígrados, amperes, volts, watts, etc.) a su descripción lingüística, mediante la función de membresía.

Función de Membresía

Una función de membresía es un conjunto difuso normal definido por el experto, que indica cómo cada punto en el espacio de entrada es mapeado a un valor de membresía (o grado de membresía) entre 0 y 1.

Las funciones de membresía agrupan en conjuntos los datos de entrada, tales como muy viejo, viejo, joven, muy joven, etc. Cada variable de entrada al sistema de control difuso puede tener varias funciones de membresía que definen a *cada valor o término lingüístico*. Por ejemplo, una variable de entrada, tal como distancia, pudiera tener cinco funciones de membresía correspondientes a los términos lingüísticos: muy lejos, lejos, medio lejos, cerca y muy cerca.

Existen diferentes formas de representar gráficamente una función de membresía, entre sus representaciones gráficas más comunes está la forma triangular, trapezoidal, campana, gaussiana, sigmoideal, entre otras. A continuación se definen los aspectos más importantes de algunas de las funciones de membresía mencionadas anteriormente.

La función de membresía triangular es descrita por tres parámetros $\{a, b, c\}$ como a continuación se muestra:

$$\text{triangular}(x; a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x. \end{cases} \quad (2.1)$$

En la Figura 2.2 se muestra gráficamente la función de membresía triangular con sus parámetros $\{a, b, c\}$ mencionados en la ecuación (2.1).

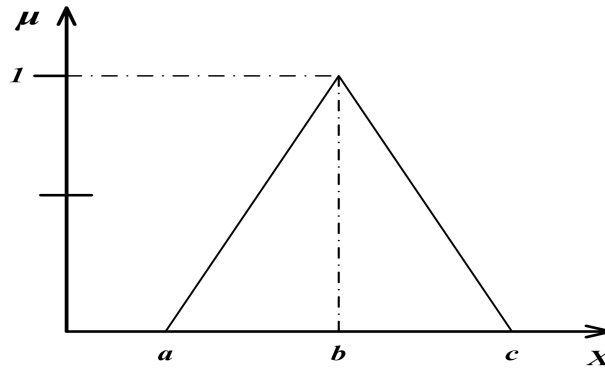


Figura 2.2: Función de membresía triangular.

Otra alternativa, para representar la función de membresía triangular es utilizando los operadores *max* y *min* tal como se muestra en la siguiente expresión:

$$\text{triangular}(x; a, b, c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right). \quad (2.2)$$

La función de membresía trapezoidal se especifica por cuatro parámetros $\{a, b, c, d\}$ como se presenta a continuación:

$$\text{trapezoidal}(x; a, b, c, d) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & d \leq x. \end{cases} \quad (2.3)$$

En la Figura 2.3 se muestra gráficamente la función de membresía trapezoidal con sus parámetros $\{a, b, c, d\}$ mencionados en la ecuación (2.3).

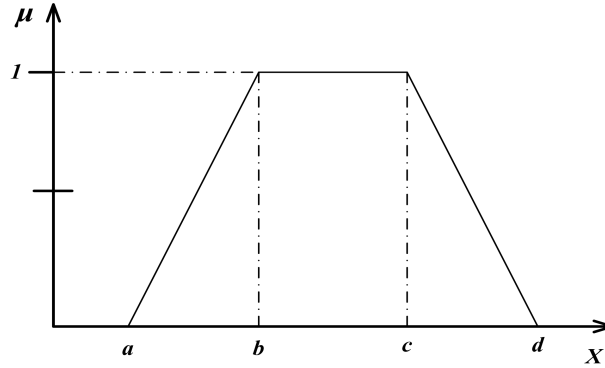


Figura 2.3: Función de membresía trapezoidal.

Utilizando los operadores *max* y *min*, la función de membresía trapezoidal también puede ser representada por la siguiente ecuación:

$$\text{trapezoidal}(x; a, b, c, d) = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{c-x}{c-b}, 0 \right), 0 \right). \quad (2.4)$$

La función gaussiana está definida por los parámetros c y σ . El parámetro c determina el centro, y el parámetro σ determina el ancho de la función. Esta función de membresía es representada gráficamente aplicando la siguiente ecuación:

$$\text{gaussiana}(x; c, \sigma) = e^{-\frac{1}{2} \left(\frac{x-c}{\sigma} \right)^2} \quad (2.5)$$

En la Figura 2.4 se representa gráficamente la función de membresía gaussiana en base a sus parámetros expresados en la ecuación (2.5).

La función de membresía campana se especifica mediante tres parámetros $\{a, b, c\}$. El parámetro a , que indica qué tan plana está la campana en su parte superior; el parámetro b indica qué tan abierta está la campana; y el parámetro c , que indica el centro de la campana. La ecuación que representa la función campana es:

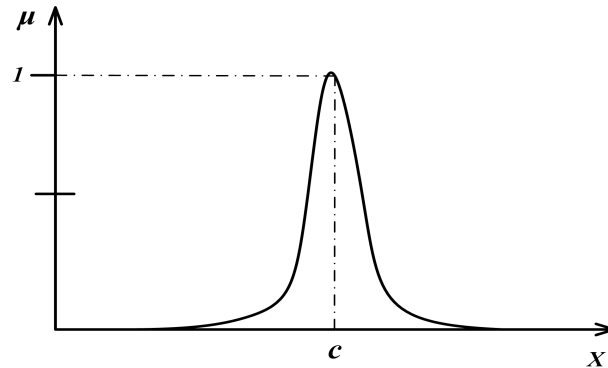


Figura 2.4: Función de membresía gaussiana.

$$\text{campana}(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \quad (2.6)$$

En la Figura 2.5 se muestra gráficamente la función campana en base a sus parámetros expresados en la ecuación (2.6).

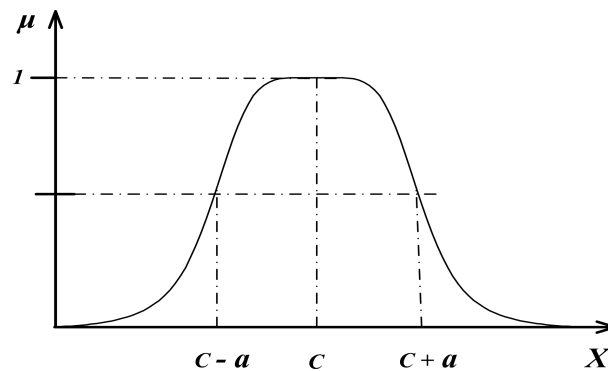


Figura 2.5: Función de membresía campana.

Una función de membresía sigmoideal es definida por

$$\text{sig}(x; a, c) = \frac{1}{1 + e^{-a(x-c)}} \quad (2.7)$$

donde el parámetro a controla la pendiente en el punto $x = c$.

En la Figura 2.6 se muestra la representación gráfica de la función sigmoïdal en base a sus parámetros expresados en la ecuación (2.7).

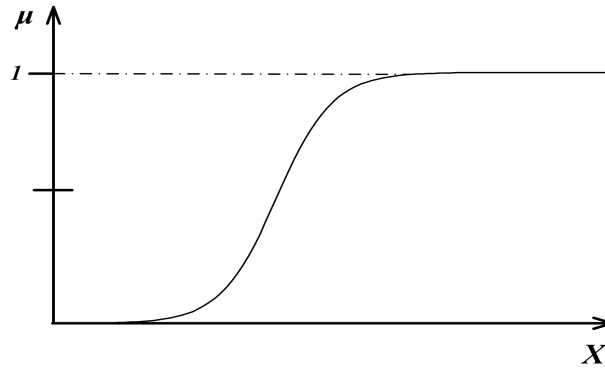


Figura 2.6: Función de membresía sigmoïdal.

2.1.2. Inferencia

La inferencia es la etapa que contiene la base de conocimiento del sistema difuso, la cual se divide básicamente en dos componentes: la base de reglas o de conocimiento, y el mecanismo de inferencia.

Base de Conocimiento

La base de conocimiento está formada a su vez por dos componentes. El primero es una base de datos que mantiene los parámetros y las características de cada una de las variables lingüísticas, el segundo es la base de reglas que es la forma en que el sistema difuso guarda el conocimiento experto que le permite resolver el problema para el cual ha sido diseñado.

Base de Datos. La base de datos proporciona información a la interfaz de fuzzi-ficación, al mecanismo de inferencia y a la interfaz de defuzzificación. Así mismo, proporciona los elementos del lenguaje para construir las reglas. Su contenido está estructurado con respecto a cada variable lingüística, proporcionando para cada una:

- El número de valores lingüísticos.
- Los parámetros de la función de membresía correspondientes a cada valor lingüístico.
- El factor de normalización y desnormalización (si lo hay) aplicado al universo de discurso.

Base de Reglas. Representa el comportamiento dinámico del sistema y contiene las reglas lingüísticas del control y la información referente a las funciones de membresía de los conjuntos difusos. Estas reglas, tienen típicamente la siguiente forma:

$$\text{Si } x_1 \text{ es } A \text{ y } x_2 \text{ es } B \text{ entonces } y \text{ es } C,$$

donde A , B y C son los conjuntos difusos de las variables de entrada “ x_1 ” y “ x_2 ”, y de la variable de salida “ y ” respectivamente.

Mecanismo de Inferencia

El mecanismo de inferencia es el encargado de efectuar el proceso de inferencia (razonamiento aproximado) sobre la base de reglas difusas y los hechos conocidos, que en este caso están representados por los conjuntos difusos obtenidos durante el mapeo realizado por la etapa de fuzzificación.

El mecanismo de inferencia *max – min* o también conocido como método de Mamdani, es actualmente el más utilizado por la sencillez y eficacia de su implementación; este método utiliza la t-norma *min* como función de implicación y la t-conorma *max* como operador de agregación.

Este método es un mecanismo de inferencia basado en reglas de la forma:

Regla 1: si x_1 es A_1 y x_2 es B_1 entonces y es C_1

Regla 2: si x_1 es A_2 y x_2 es B_2 entonces y es C_2

La conclusión inferida, en forma de función de membresía, de la aplicación de la base de reglas está dada por la siguiente expresión

$$\mu_{c'}(y) = \max(\min(\alpha^i, \mu_{c_i}(y))) \quad (2.8)$$

donde el grado de activación de la i -ésima regla es

$$\alpha^i = \min(\mu_{A_1}^i(x_1), \dots, \mu_{A_n}^i(x_n)) \quad (2.9)$$

donde α^i es el grado de activación de la i -ésima regla, $\mu_{A_1}^i(x_1), \dots, \mu_{A_n}^i(x_n)$ son las funciones de membresía de las entradas x_1, \dots, x_n respectivamente, y $\mu_{C_i}(y)$ son las funciones de membresía resultantes de la función de implicación (min).

El proceso de este mecanismo para dos variables de entrada y dos reglas es ilustrado en la Figura 2.7. La obtención de un valor concreto de la señal de control requiere de la aplicación de alguno de los métodos de defuzzificación.

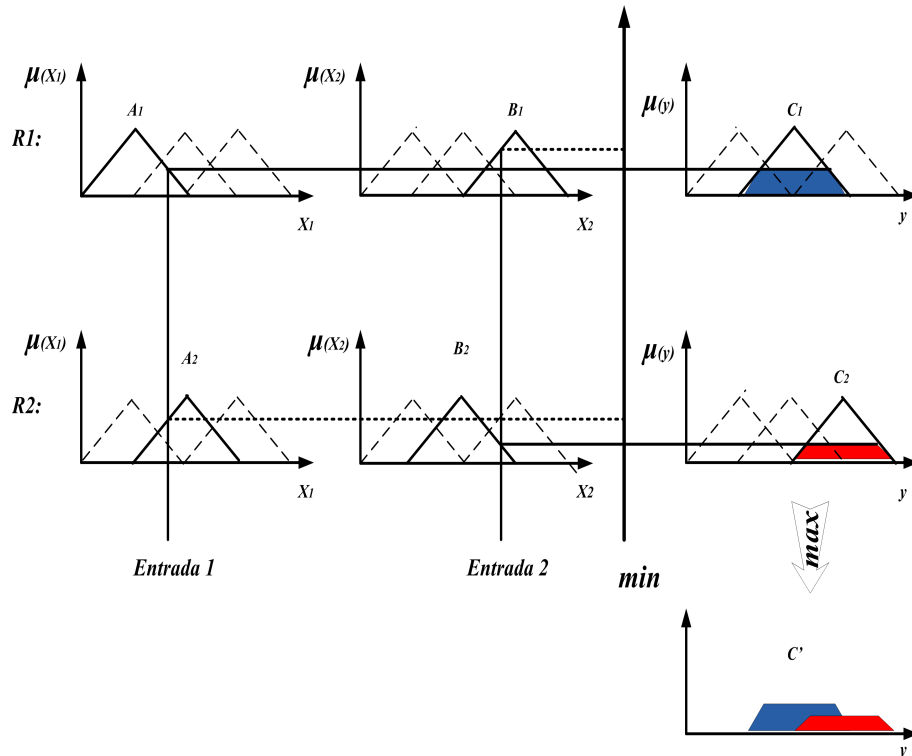


Figura 2.7: Mecanismo de inferencia $max - min$.

2.1.3. Defuzzificación

En todas las aplicaciones prácticas de los sistemas de control difuso se tiene un paso esencial después del proceso de inferencia. Se espera que en lugar de tener un conjunto difuso, la respuesta sea un valor real en el universo de discurso de la variable de salida. De ahí que hay necesidad de efectuar una traslación del conjunto difuso resultado del proceso de inferencia, a un valor real único representativo. El proceso que efectúa tal paso se conoce como defuzzificación el cual formalmente consiste en extraer un valor numérico de un conjunto difuso, para seleccionar un punto que sea el más representativo de la acción a realizar.

Existen diferentes métodos de defuzzificación, pero como ocurre con otros elementos del diseño de sistemas difusos, no existe aún un procedimiento sistemático para seleccionar el más adecuado, dependiendo del caso de aplicación particular. A continuación se presentan algunos de los métodos de defuzzificación más utilizados.

- Centro de Gravedad (COA).
- Criterio del Máximo (CM).
- Método de las Alturas.

La selección del método de defuzzificación puede jugar un papel decisivo en la síntesis de modelos difusos para muchas áreas de aplicación. Particularmente dentro del área del control difuso, su influencia puede ser determinante en el comportamiento y la robustez del controlador.

Centro de Gravedad (COA). Una vez dado el conjunto de salida se calcula su centro de gravedad mediante la ecuación

$$\bar{y} = \frac{\sum_{i=1}^N y_i \cdot \mu_B(y_i)}{\sum_{i=1}^N \mu_B(y_i)} \quad (2.10)$$

donde el universo de discurso de salida a sido discretizado en N puntos. Para cada entrada del sistema difuso podemos obtener un valor diferente de \bar{y} . En la Figura 2.8 se muestra la interpretación gráfica de este método de defuzzificación.

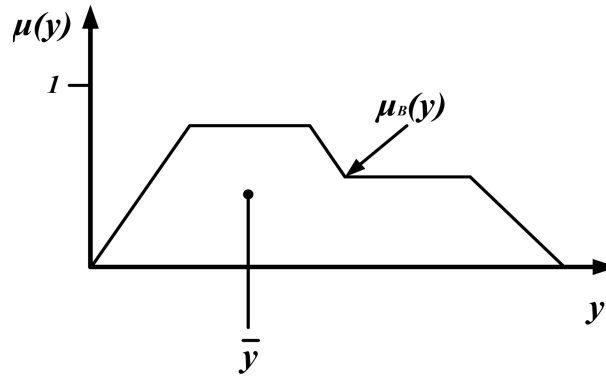


Figura 2.8: Centro de Gravedad.

Criterio del Máximo (CM). Este es el más simple de todos los métodos y el más económico desde el punto de vista computacional ya que encuentra el punto máximo donde $\bar{y} = \max(\mu_B(y))$. Si hay más de un punto máximo (intervalo), se calcula el punto medio de ese intervalo

$$\bar{y} = \frac{\bar{y}_{\text{máx}} - \bar{y}_{\text{mín}}}{2} \quad (2.11)$$

donde $\bar{y}_{\text{máx}}$ es el punto máximo y $\bar{y}_{\text{mín}}$ es el punto mínimo del intervalo respectivamente. En la Figura 2.9 se ilustra gráficamente el método de defuzzificación del criterio del máximo.

Método de las Alturas. El método de las alturas reemplaza cada conjunto difuso de salida por un valor numérico ubicado en el máximo grado de membresía del conjunto, y a continuación calcula el centroide del conjunto compuesto por estos valores numéricos. La salida del método de las alturas está dada por

$$\bar{y} = \frac{\sum_{i=1}^N \bar{y}_i \cdot \mu_{B^i}(\bar{y}_i)}{\sum_{i=1}^N \mu_{B^i}(\bar{y}_i)} \quad (2.12)$$

donde \bar{y}_i es el máximo grado de membresía del i -ésimo conjunto de salida (si este es más de un punto, puede tomarse el promedio de \bar{y}_i) y su grado de membresía en el i -ésimo

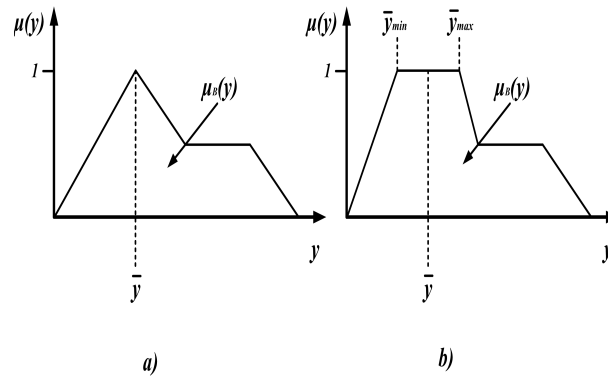


Figura 2.9: Criterio del Máximo: a) Único valor, b) Intervalo de valores.

conjunto de salida es $\mu_{B^i}(\bar{y}_i)$. En la Figura 2.10 se muestra el método de defuzzificación de las alturas gráficamente.

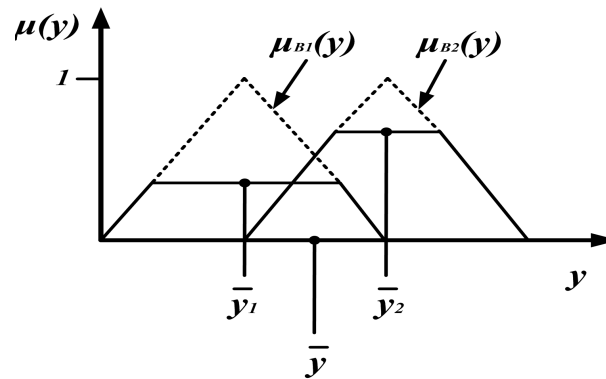


Figura 2.10: Método de las Alturas.

2.2. VHDL

VHDL es un lenguaje de programación y el significado de sus siglas es VHSIC (*Very High Speed Integrated Circuit Hardware Description Language*), es decir, lenguaje de descripción de hardware de circuitos integrados de muy alta velocidad. Este lenguaje fue desarrollado originalmente por el Departamento de Defensa de los Estados Unidos y

en el año de 1987 fue estandarizado por la IEEE (*Institute of Electrical and Electronics Engineers*) con el estándar IEEE Std 1076 [26].

Fue desarrollado como un lenguaje para el modelado y simulación lógica dirigida por eventos de sistemas digitales, y actualmente se utiliza también para la síntesis automática de circuitos. De manera que permite diseñar, modelar y comprobar un sistema desde un alto nivel de abstracción bajando hasta el nivel de definición estructural de compuertas lógicas [27].

VHDL divide los circuitos en entidades y arquitecturas. La entidad modela el circuito, componente o sistema externamente definiendo a éste mediante un nombre y sus conexiones que vienen siendo las entradas y salidas del circuito, en tanto que la arquitectura define el funcionamiento del circuito. Después de definir las interfaces de la entidad, otras entidades pueden utilizar a la primera como un subcircuito, al mismo tiempo que todas están siendo desarrolladas, es decir, están siendo detalladas en su funcionamiento. Así, una entidad es relativa a otras entidades a través de conexiones y comportamiento, por lo que es posible experimentar cada entidad con diferentes arquitecturas sin necesidad de cambiar el resto del diseño. De tal manera que cada entidad puede ser reutilizada en otros sistemas aunque no hayan sido diseñadas específicamente para éstos. En la Figura 2.11 se muestra un modelo de hardware desarrollado en VHDL.

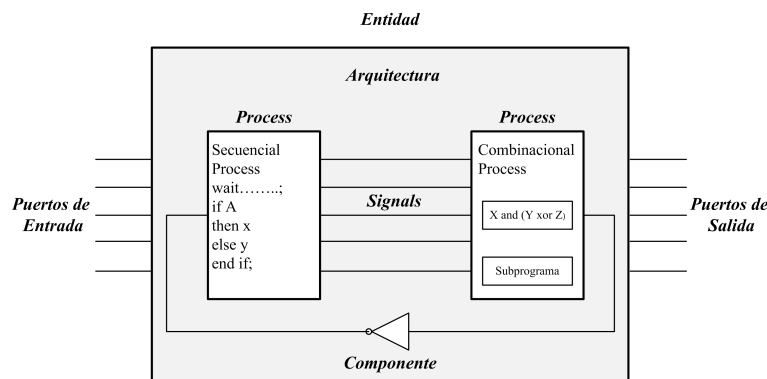


Figura 2.11: Modelo de hardware desarrollado en VHDL.

Una entidad en VHDL, que ya es todo un diseño, posee una o más conexiones

hacia los sistemas que la rodean. Una entidad puede estar compuesta de otras entidades, de procesos y por componentes, todos ellos trabajando concurrentemente. Cada entidad está definida por su arquitectura la cual se forma con instrucciones en VHDL, ya sean aritméticas, asignaciones a señales o de simples instanciaciones de componentes.

Los PROCESS en VHDL son utilizados para modelar tanto circuitos secuenciales como combinacionales utilizando un estilo de descripción secuencial. Para interconectar procesos distintos se utilizan SIGNALS que no son otra cosa que simples cables.

Una señal posee una fuente (driver) y uno o más destinos (receptores) y un tipo de dato que le proporciona características de interconexión. Por ejemplo, una señal que se define como tipo *bit* puede manejar los valores binarios 0 y 1 solamente, en tanto que una señal que se define como *bit vector* puede manejar más de una posición binaria.

La forma de diseñar circuitos en VHDL se divide en tres categorías de acuerdo a su complejidad: flujo de datos, de comportamiento o funcional, y estructural [28].

2.2.1. Flujo de Datos

En este estilo el diseño del circuito no es complicado por lo que solo basta con describir cómo fluyen los datos a través de la entidad, de las entradas hacia las salidas. La operación del sistema está definida en términos de un conjunto de transformaciones de datos expresadas como instrucciones concurrentes. En la figura 2.12 se muestra un esquema del estilo flujo de datos.

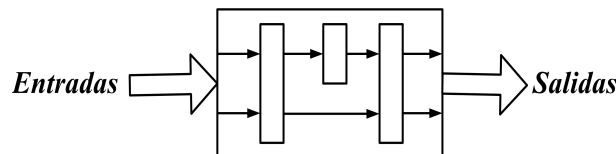


Figura 2.12: Esquema del modelado flujo de datos.

2.2.2. Comportamiento o Funcional

El diseño es un poco más complicado ya que requiere de varias decisiones antes de definir los datos de salida correctos. Por lo que se requiere de una descripción algorítmica del funcionamiento del circuito para facilitar el diseño del sistema. En VHDL esto se obtiene expresando el funcionamiento del diseño mediante una estructura PROCESS la cual se compone de instrucciones secuenciales. En la Figura 2.13 se muestra un esquema del estilo de comportamiento o funcional.

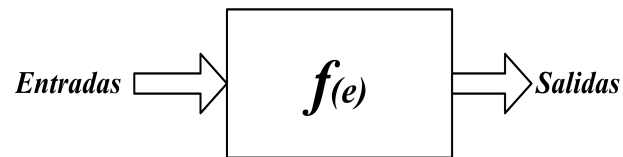


Figura 2.13: Esquema del modelado de comportamiento o funcional.

2.2.3. Estructural

Una descripción estructural se utiliza en circuitos que requieren de más de una función, hablando en términos de hardware, para realizar la finalidad del sistema. Para ello segmentamos el sistema en subcircuitos o componentes para facilitar el diseño, cada componente es caracterizado en particular ya sea utilizando una descripción de flujo de datos o funcional, y la entidad donde se describen las interconexiones de estos componentes recibe el nombre de descripción estructural. En la Figura 2.14 se muestra un esquema del estilo estructural.

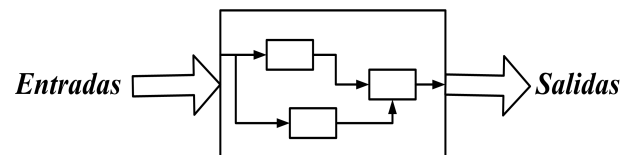


Figura 2.14: Esquema del modelado estructural.

Lo que ha hecho que VHDL sea en un tiempo tan corto el lenguaje de descripción de hardware más utilizado por la industria electrónica, es su independencia con la metodología utilizada por cada diseñador, su capacidad de descripción a diferentes niveles de abstracción, y en definitiva la posibilidad de poder reutilizar en diferentes aplicaciones un mismo código.

2.3. FPGA's

Un sistema de control difuso puede ser desarrollado usando software o hardware. Las implementaciones software son flexibles, aunque toman relativamente una gran cantidad de tiempo (del orden de los cientos de *ms*) para obtener el resultado de la inferencia, por ello esta solución es poco usada. Por otra parte, las implementaciones hardware poseen velocidades de procesamiento altas, son compactas y poco flexibles, características que posee un FPGA [29].

Un FPGA (del inglés Field Programmable Gate Array) es un dispositivo semiconductor que contiene componetes lógicos programables e interconexiones programables entre ellos. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una compuerta lógica o un sistema combinacional, hasta complejos sistemas en un circuito integrado [30].

La arquitectura de un FPGA consiste en arreglos de varias celdas lógicas las cuales se comunican unas con otras mediante canales de conexión verticales y horizontales tal como se muestra en la figura 2.15 [28].

Cada celda lógica es funcionalmente similar a los bloques lógicos de un CPLD. La diferencia está en que un FPGA normalmente utiliza generadores de funciones en vez de compuertas. Cada uno de estos generadores es como una memoria en donde en vez de implementar la función lógica mediante compuertas, se precalcula el resultado y se almacena en el generador. Las entradas al generador funcionan como un bus de direcciones, y mediante las diferentes combinaciones de las entradas del generador se selecciona el

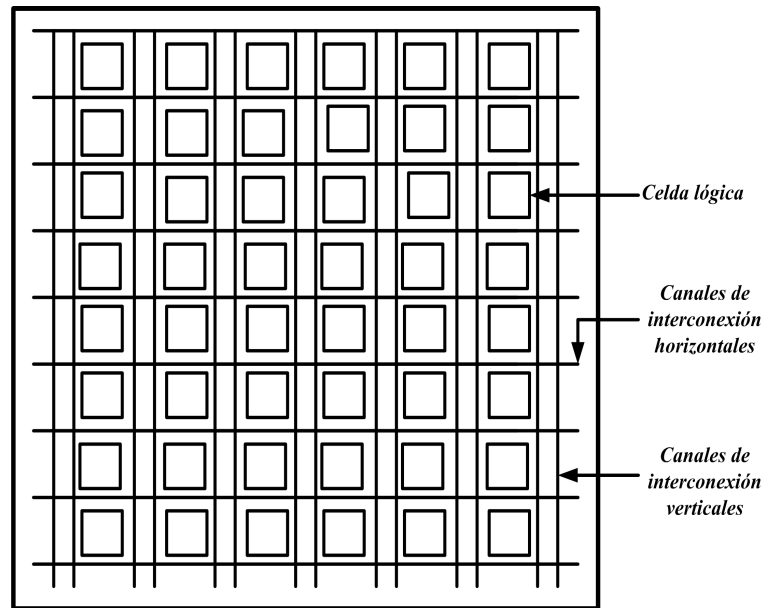


Figura 2.15: Arquitectura básica de un FPGA.

resultado correcto. Esto le da una gran densidad al dispositivo ya que se maneja un gran número de generadores, pero el tiempo de propagación al implementar una función lógica en estos generadores es menor al que se necesitaría si utilizáramos compuertas. La estructura de las celdas lógicas y las formas en que éstas pueden ser interconectadas, tanto salidas como entradas de la celda, varían de acuerdo al fabricante. En general una celda lógica tiene menos funcionalidad que la combinación de suma de productos y macroceldas de un CPLD, pero como cada FPGA tiene una gran cantidad de celdas lógicas es posible implementar grandes funciones utilizando varias celdas lógicas en cascada.

2.3.1. Ventajas de los FPGA's

Las ventajas al utilizar los dispositivos FPGA's en el diseño de sistemas digitales son las siguientes:

- **Hardware a la medida.** El diseñador no tiene que buscar los productos del mercado que mejor se adapten a sus diseños, sino que los diseña a la medida de sus

necesidades, o reutiliza o modifica los diseños ya existentes. En robótica, es muy común emplear un microcontrolador u otro en función de los periféricos que traiga integrados, y es muy común que sólo se utilicen unos pocos de ellos. En un sistema que incorpore FPGA's, es el diseñador el que implemente sólo los controladores necesarios. Se abren nuevas posibilidades de diseño, como por ejemplo el crear una CPU específica para una aplicación determinada junto con sus propios controladores de periféricos.

- **Acortamiento del ciclo de diseño.** El modelo del diseño hardware basado en HDL contiene muchas de las ventajas del diseño software. El circuito es ahora un archivo de texto, que se puede editar, simular, modificar y finalmente sintetizar. Se pueden crear repositorios hardware, con colecciones de diseños ya probados: controladores VGA, UART's, temporizadores, CPUs, etc. El diseñador puede crear prototipos muy rápidamente, probarlos, medirlos y modificarlos.
- **Flexibilidad.** Con el mismo hardware físico, conseguimos tener hardware con comportamientos diferentes. En la misma placa JPS, ahora podemos tener sintetizada una CPU, y más adelante podemos probar una UART.
- **Posibilidad de pasar algoritmos al hardware.** En los diseños mixtos FPGA-microcontrolador se puede optar por implementar una solución hardware frente a uno puramente software, consiguiéndose una mayor velocidad, por ejemplo la implementación de algoritmos de cifrado. En un microcontrolador de 8 bits esto sería un proceso lento, pero se convierte en viable si se añade un hardware que haga el cifrado.
- **Diseños hardware libre.** Posibilidad de realizar diseños de hardware libre que se compartan dentro de la comunidad hardware y que cualquier diseñador pueda utilizarlos, modificarlos y distribuir las modificaciones. Esto es especialmente útil en el campo de la docencia y la investigación. Esto se potencializa si el hardware

físico en el que se prueban los diseños es también libre, como el caso de las placas JPS. Desaparece la dependencia con el fabricante de la placa y cada universidad o diseñador puede fabricarse las placas que considere necesarias.

Capítulo 3

Sistema Difuso en VHDL para el Control de Movimiento de un Robot Móvil con Ruedas

El robot móvil con ruedas que se muestra en la Figura 3.1, es un ejemplo típico de un sistema mecánico no-holonómico, el cual se caracteriza por estar sujeto a restricciones no integrables en las velocidades, es decir su plano de velocidades está restringido, emplea un sistema de locomoción con direccionamiento diferencial, el cual posee dos ruedas montadas sobre un mismo eje actuadas independientemente y una rueda libre para mantener el equilibrio. Se asume que el robot móvil de ruedas considerado en este trabajo, está bajo la restricción de rodamiento puro y no deslizamiento.

3.1. Modelo Cinemático

El modelo cinemático del robot móvil con ruedas con direccionamiento diferencial se expresa mediante la siguiente ecuación [1]:

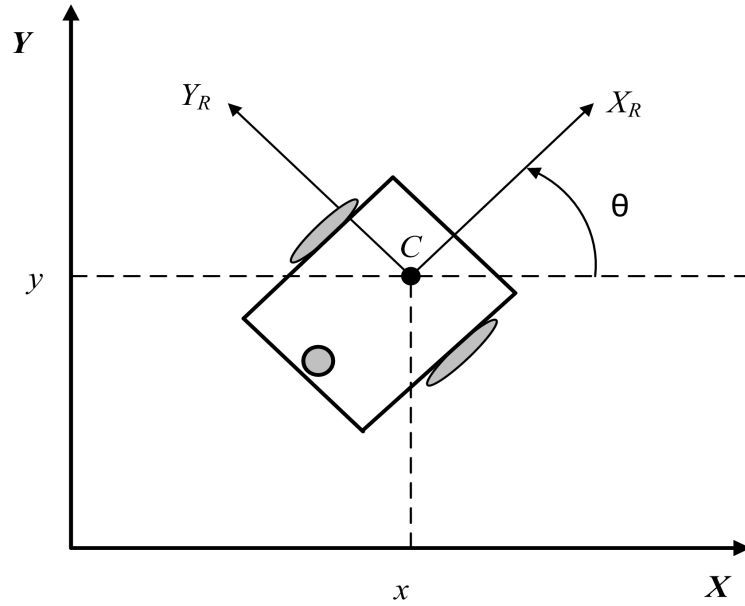


Figura 3.1: Robot Móvil con Ruedas.

$$\dot{q} = \underbrace{\begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}}_{S(q)} \underbrace{\begin{bmatrix} v \\ \omega \end{bmatrix}}_{\vartheta} \quad (3.1)$$

donde $q = (x, y, \theta)^T$ es el vector de coordenadas generalizadas; $\dot{q} = (\dot{x}, \dot{y}, \dot{\theta})^T$ es el vector de velocidades generalizadas, (x, y) representan la posición del punto de referencia C del robot móvil en las coordenadas cartesianas (X, Y) y θ es el ángulo de orientación con respecto a su dirección y el eje X ; $\vartheta = (v, \omega)$ es el vector de velocidades, donde v es la velocidad lineal y ω la velocidad angular.

Como se comentó anteriormente, el robot móvil de ruedas considerado en este trabajo emplea un sistema de locomoción direccionamiento diferencial, por ello en (3.1) se debe dejar en función de ω_d , que es la velocidad angular de la rueda derecha, y ω_i que corresponde a la velocidad angular de la rueda izquierda respectivamente del robot móvil. De manera que v y ω , en función de ω_d y ω_i , están dadas por [2]:

$$v = \frac{(\omega_d + \omega_i)}{2} \quad (3.2)$$

$$\omega = \frac{(\omega_d - \omega_i) r}{b}. \quad (3.3)$$

Al sustituir (3.2) y (3.3) en (3.1) se obtiene la siguiente ecuación:

$$\dot{q} = \begin{bmatrix} \frac{r \cos \theta}{2} & \frac{r \cos \theta}{2} \\ \frac{r \sin \theta}{2} & \frac{r \sin \theta}{2} \\ -\frac{r}{b} & \frac{r}{b} \end{bmatrix} \begin{bmatrix} \omega_i \\ \omega_d \end{bmatrix} \quad (3.4)$$

donde r es radio de la rueda y b es la distancia del eje que separa las dos ruedas del robot móvil.

3.2. Sistema de Control de Movimiento del Robot Móvil con Ruedas

El objetivo del presente trabajo es desarrollar un sistema de control difuso que nos permita resolver el problema de regulación de postura de un robot móvil con ruedas, el cual es representado formalmente como se muestra a continuación:

$$\lim_{t \rightarrow \infty} \|q_d - q(t)\| = 0. \quad (3.5)$$

El control de regulación de postura consiste en que la diferencia entre los valores constantes, para todo $t \geq 0$, de posición y orientación deseados $q_d = (x_d, y_d, \theta_d)$ y las coordenadas cartesianas de posición y orientación actuales del robot móvil con ruedas $q = (x(t), y(t), \theta(t))$ deben tender a cero. En la Figura 3.2 se ilustra el problema de regulación de postura del robot móvil con ruedas.

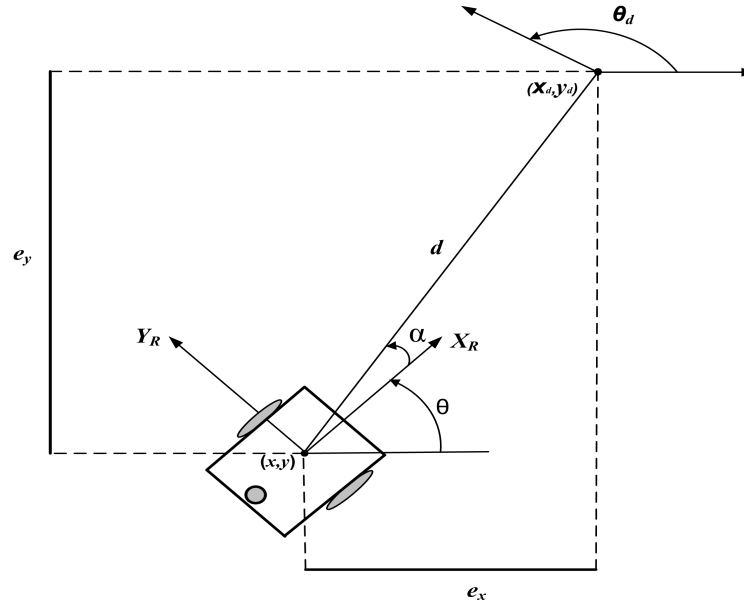


Figura 3.2: Problema de Regulación de Postura.

En la Figura 3.3 se muestra el sistema de control de movimiento del robot móvil con ruedas empleado para el desarrollo de este trabajo de tesis. Dicho sistema está compuesto por un lazo interno que se encarga de controlar la velocidad angular de los motores de las ruedas de tracción del robot móvil, y un lazo externo asociado a su modelo cinemático que se encarga de generar las velocidades de referencia que deben seguir los controladores de velocidad del lazo interno para alcanzar la posición y orientación deseada.

Los controladores empleados en el lazo interno para el control de velocidad de los motores de las ruedas del robot móvil, son controladores *PI* dados por $K(s) = K_p + \frac{K_i}{s}$ donde: $K_p = 0.008$ y $K_i = 0.06$, las cuales se obtuvieron de forma arbitraria (prueba y error). Cada rueda actuada del robot móvil utiliza un motor de CD Pittman GM9236S025 equipado con un encoder óptico para ser implementado en la retroalimentación del lazo de control [31], la velocidad es regulada utilizando un modulador de ancho de pulso (PWM) y la etapa de potencia se maneja por medio de un puente H utilizando el circuito integrado LMD18200 [32]. En la Figura 3.4 se muestra el diagrama a bloques del sistema de control de potencia del motor de CD utilizando el PWM y el puente H.

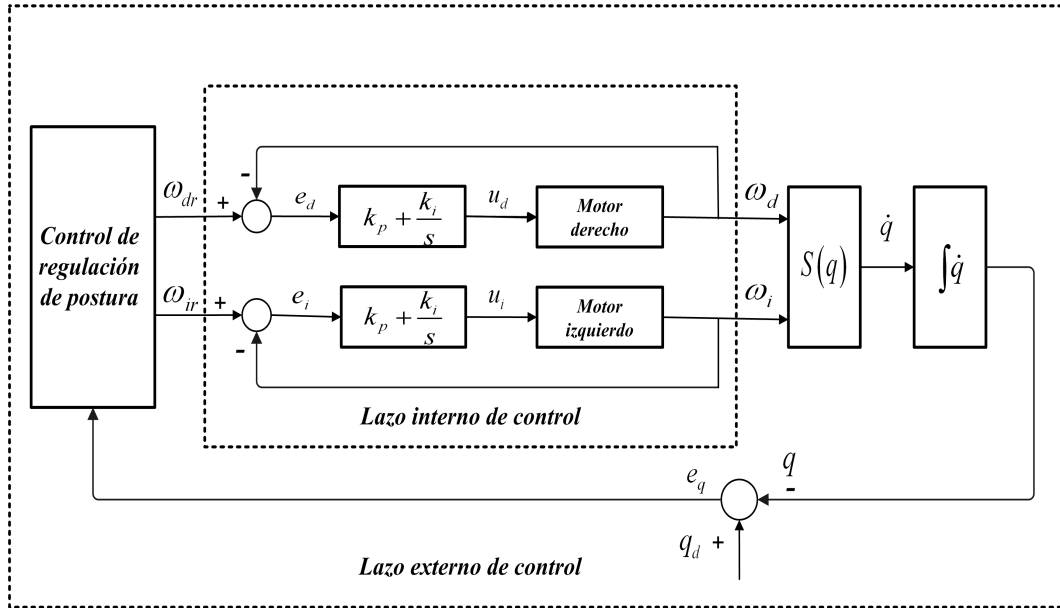


Figura 3.3: Sistema de control de movimiento del robot móvil con ruedas.

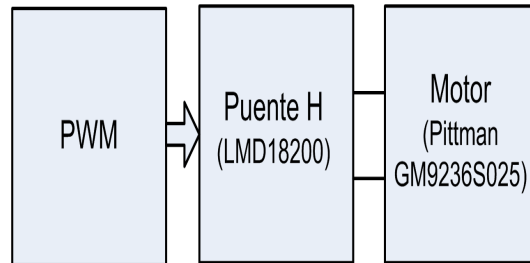


Figura 3.4: Etapa de control de potencia del motor de CD utilizando PWM y puente H.

La función de transferencia (3.6) representa el modelo del sistema formado por el motor de CD, el modulador de ancho de pulso (PWM) y el puente H.

$$G(s) = \frac{351.9}{s + 4.933}. \quad (3.6)$$

3.3. Control de Regulación de Postura

Para cumplir con el objetivo de control propuesto en (3.5) es necesario desarrollar un sistema difuso de tres entradas, lo que ocasiona que el número de reglas crezca difícil-

tando la sintonización [12]. Para simplificar el desarrollo se decidió dividir el control de regulación de postura en dos sistemas difusos:

- Sistema difuso de posición.
- Sistema difuso de orientación.

3.3.1. Sistema Difuso de Posición

El sistema difuso de posición tiene como objetivo hacer que el robot móvil con ruedas se desplace de la posición inicial a la posición deseada. Este sistema difuso es de tipo Mamdani con dos variables lingüísticas de entrada y dos de salida. Las variables lingüísticas de entrada son llamadas distancia(d) y alfa(α).

La variable lingüística distancia tiene dos términos lingüísticos que son pequeño(P) y grande(G), que se representan mediante una función de membresía tipo triangular y una tipo trapezoidal respectivamente, definidas en un universo de discurso en el intervalo $[0, 4]$ metros. En la Figura 3.5 se muestra la variable lingüística distancia con sus respectivos términos lingüísticos. Esta variable representa la magnitud de la distancia que existe entre la posición inicial y la deseada del robot móvil con ruedas. El cálculo de la distancia se obtiene mediante la siguiente ecuación [1]:

$$d = \sqrt{e_x^2(t) + e_y^2(t)} \quad (3.7)$$

donde: $e_x(t) = x_d(t) - x(t)$, $e_y(t) = y_d(t) - y(t)$.

La variable lingüística alfa tiene cinco términos lingüísticos los cuales son grande negativo(GN), mediano negativo(MN), cero(Z), mediano positivo(MP) y grande positivo(GP), los cuales se representan por tres funciones de membresía tipo triangular y dos de tipo trapezoidal, definidas en un universo de discurso dentro del intervalo $[-2\pi, 2\pi]$ radianes, que representa un giro completo en sentido horario y un giro completo en sentido antihorario del robot móvil con ruedas. En la Figura 3.6 se muestra la variable lingüística alfa con sus respectivos términos lingüísticos. Dicha variable representa el ángulo que se

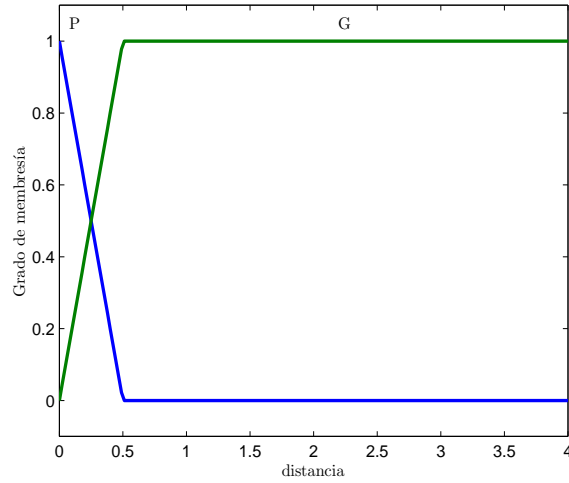


Figura 3.5: Variable lingüística distancia.

forma entre la orientación del robot móvil y la línea recta que conecta la posición actual del robot móvil con ruedas con su posición deseada (observar Figura 3.2). El cálculo de dicho ángulo se obtiene mediante la siguiente ecuación [1]:

$$\alpha = \text{atan2}(e_y(t), e_x(t)) - \theta(t) \quad (3.8)$$

donde

$$\text{atan2}(e_y, e_x) = \begin{cases} \arctan\left(\frac{e_y}{e_x}\right) & e_x > 0 \\ \pi + \arctan\left(\frac{e_y}{e_x}\right) & e_y \geq 0, e_x < 0 \\ -\pi + \arctan\left(\frac{e_y}{e_x}\right) & e_y < 0, e_x < 0 \\ \frac{\pi}{2} & e_y > 0, e_x = 0 \\ -\frac{\pi}{2} & e_y < 0, e_x = 0 \\ \text{No definido} & e_y = 0, e_x = 0 \end{cases}$$

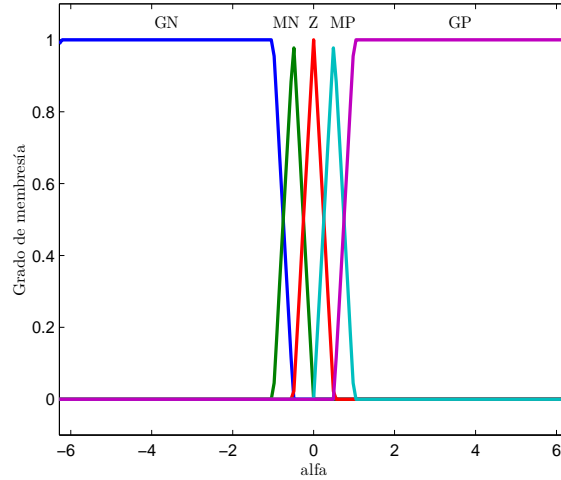


Figura 3.6: Variable lingüística alfa.

En la Tabla 3.1 se muestran los parámetros de las funciones de membresía de las variables lingüísticas distancia(d) y alfa(α).

Tabla 3.1: Parámetros de las funciones de membresía de las variables lingüísticas distancia(d) y alfa(α).

Variable lingüística	Término lingüístico	Parámetros
distancia(d)	Pequeño(P)	(-0.5, 0, 0.5)
	Grande(G)	(0, 0.5, 4, 4)
alfa(α)	Grande Negativo(GN)	(-2π , -2π , -1, -0.5)
	Mediano Negativo(MN)	(-1, -0.5, 0)
	Cero(Z)	(-0.5, 0, 0.5)
	Mediano Positivo(MP)	(0, 0.5, 1)
	Grande Positivo(GP)	(0.5, 1, 2π , 2π)

Se manejan dos tipos de variables lingüísticas de salida. De manera que se desarrollaron dos sistemas difuso de posición, uno que maneja como variables lingüísticas de salida velocidad lineal(v_r) y velocidad angular(ω_r) de referencia del robot móvil, y otro que maneja como variables lingüísticas de salida velocidad angular izquierda(ω_{ir}) y velocidad angular derecha(ω_{dr}) de referencia de sus ruedas respectivamente.

Variables Lingüísticas Velocidad Lineal y Velocidad Angular

En la Figura 3.7 se muestra la variable lingüística de salida velocidad lineal con sus respectivos términos lingüísticos. Dicha variable tiene cinco términos lingüísticos que son velocidad lineal grande negativa(VGN), mediana negativa(VMN), cero(VZ), mediana positiva(VMP) y grande positiva(VGP), los cuales son representados por tres funciones de membresía tipo triangular y dos de tipo trapezoidal, definidas en un universo de discurso dentro del intervalo de $[-0.8, 0.8]$ metros por segundo.

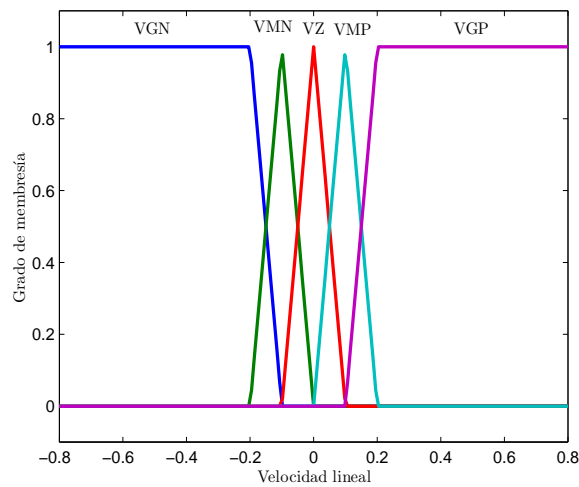


Figura 3.7: Variable lingüística velocidad lineal.

La variable lingüística de salida velocidad angular consta de cinco términos lingüísticos llamados velocidad angular grande negativa(WGN), mediana negativa(WMN), cero(WZ), mediana positiva(WMP) y grande positiva(WGP). En la Figura 3.8 se observa la variable lingüística velocidad angular con sus respectivos términos lingüísticos representados por tres funciones de membresía tipo triangular y dos tipo trapezoidal, las cuales se encuentran definidas en un universo de discurso dentro del intervalo de $[-3, 3]$ radianes por segundo.

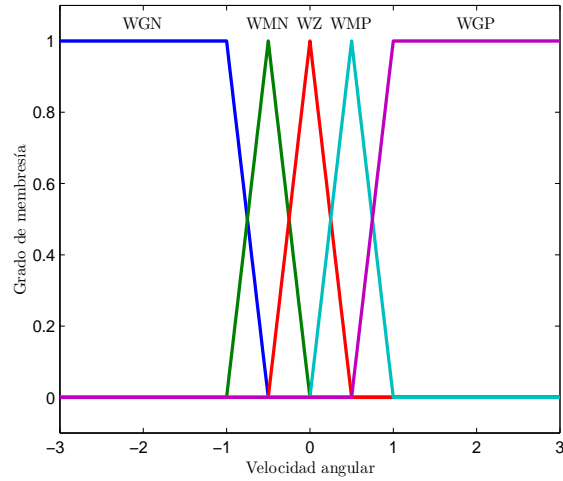


Figura 3.8: Variable lingüística velocidad angular.

En la Tabla 3.2 se muestran los parámetros de las funciones de membresía de las variables lingüísticas velocidad lineal(v_r) y velocidad angular(ω_r).

Tabla 3.2: Parámetros de las funciones de membresía de las variables lingüísticas velocidad lineal(v_r) y velocidad angular(ω_r).

Variable lingüística	Término lingüístico	Parámetros
velocidad lineal(v_r)	Grande Negativo(VGN)	(-0.8, -0.8, -0.2, -0.1)
	Mediano Negativo(VMN)	(-0.2, -0.1, 0)
	Cero(VZ)	(-0.1, 0, 0.1)
	Mediano Positivo(VMP)	(0, 0.1, 0.2)
	Grande Positivo(VGP)	(0.1, 0.2, 0.8, 0.8)
velocidad angular(ω_r)	Grande Negativo(WGN)	(-3, -3, -1, -0.5)
	Mediano Negativo(WMN)	(-1, -0.5, 0)
	Cero(WZ)	(-0.5, 0, 0.5)
	Mediano Positivo(WMP)	(0, 0.5, 1)
	Grande Positivo(WGP)	(0.5, 1, 3, 3)

A partir de la cantidad de términos lingüísticos asociados a las variables lingüísticas de entrada, se obtienen 10 reglas difusas las cuales se expresan en la Tabla 3.3, e indica la relación que existe entre las variables lingüísticas de entrada y salida.

Tabla 3.3: Base de reglas difusas con respecto a las variables lingüísticas velocidad lineal y velocidad angular.

		<i>alfa</i>					
		GN	MN	Z	MP	GP	
<i>distancia</i>	P	v_r	<i>VMN</i>	<i>VMP</i>	<i>VZ</i>	<i>VMP</i>	<i>VMN</i>
		ω_r	<i>WGN</i>	<i>WMN</i>	<i>WZ</i>	<i>WMP</i>	<i>WGP</i>
	G	v_r	<i>VGN</i>	<i>VGP</i>	<i>VGP</i>	<i>VGP</i>	<i>VGN</i>
		ω_r	<i>WGN</i>	<i>WMN</i>	<i>WZ</i>	<i>WMP</i>	<i>WGP</i>

Cuando se consideran estas dos salidas para el sistema difuso de posición, es necesario hacer la conversión de velocidad lineal y angular a velocidad angular izquierda y derecha de las ruedas del robot móvil. En la Figura 3.3 se observa que el sistema de control del robot móvil toma como referencia las velocidades angulares de sus ruedas. Para realizar dicha conversión se utilizan las siguientes ecuaciones [2]:

$$\omega_{ir} = \frac{v_r - (b/2)\omega_r}{r} \quad (3.9)$$

$$\omega_{dr} = \frac{v_r + (b/2)\omega_r}{r} \quad (3.10)$$

VARIABLES LINGÜÍSTICAS VELOCIDAD ANGULAR IZQUIERDA Y VELOCIDAD ANGULAR DERECHA

Ahora se propone que el sistema difuso de posición trabaje con las variables lingüísticas de salida velocidad angular izquierda y velocidad angular derecha. La razón por la cual se considera trabajar con estas variables, es que si se observa la Figura 3.3 el sistema de control del robot móvil considera las velocidades de sus ruedas como referencia, de manera que es más conveniente trabajar con dichas variables como salidas del sistema difuso y así evitar hacer la conversión de velocidad lineal y angular a velocidad angular izquierda y derecha de referencia de las ruedas del robot móvil.

En la Figura 3.9 se muestra la variable lingüística velocidad angular izquierda con sus cinco términos lingüísticos los cuales son velocidad angular grande negativa(WGN), mediana negativa(WMN), cero(WZ), mediana positiva(WMP) y grande positiva(WGP),

los cuales se representan por tres funciones de membresía triangulares y dos trapezoidales en un universo de discurso que se encuentra en el intervalo de $[-8, 8]$ radianes por segundo.

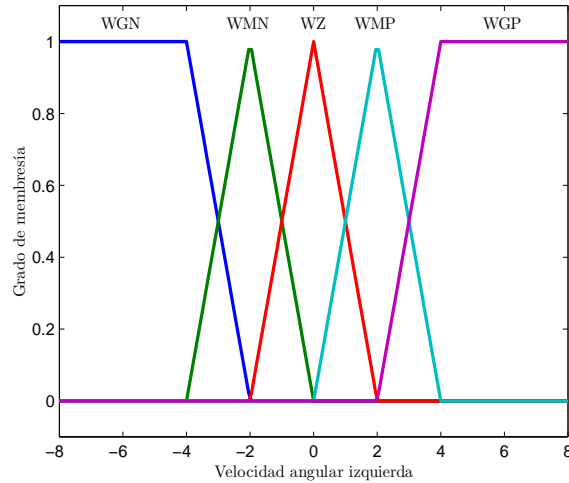


Figura 3.9: Variable lingüística velocidad angular izquierda.

Al igual que la variable lingüística de salida velocidad angular izquierda, la variable lingüística de salida velocidad angular derecha también considera cinco términos lingüísticos que son representados por dos funciones de membresía tipo trapezoidal y tres de tipo triangular, cuyos términos son velocidad angular derecha grande negativa(WGN), mediana negativa(WMN), cero(WZ), mediana positiva(WMP) y grande positiva(WGP). En la Figura 3.10 se muestra la variable lingüística de salida velocidad angular derecha con sus respectivos términos lingüísticos dentro del intervalo del universo de discurso de $[-8, 8]$ radianes por segundo.

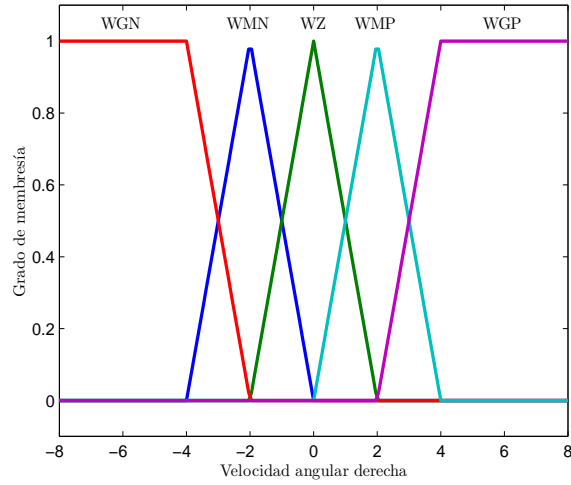


Figura 3.10: Variable lingüística velocidad angular derecha.

En la Tabla 3.4 se muestran los parámetros de las funciones de membresía de las variables lingüísticas velocidad angular izquierda (v_{ir}) y velocidad angular derecha (ω_{dr}).

Tabla 3.4: Parámetros de las funciones de membresía de las variables lingüísticas velocidad angular derecha (v_{dr}) y velocidad angular izquierda (ω_{ir}).

Variable lingüística	Término lingüístico	Parámetros
velocidad angular izquierda (v_{ir})	Grande Negativa(WGN)	(-8, -8, -4, -2)
	Mediana Negativa(VMN)	(-4, -2, 0)
	Cero(WZ)	(-2, 0, 2)
	Mediana Positiva(WMP)	(0, 2, 4)
	Grande Positiva(WGP)	(2, 4, 8, 8)
velocidad angular derecha (v_{dr})	Grande Negativa(WGN)	(-8, -8, -4, -2)
	Mediana Negativa(VMN)	(-4, -2, 0)
	Cero(WZ)	(-2, 0, 2)
	Mediana Positiva(WMP)	(0, 2, 4)
	Grande Positiva(WGP)	(2, 4, 8, 8)

En la Tabla 3.5, se muestra la base de reglas difusas que relaciona las variables lingüísticas de entrada distancia y alfa, con las de salida velocidad angular izquierda y velocidad angular derecha.

Tabla 3.5: Base de reglas difusas con respecto a las variables lingüísticas velocidad angular izquierda y velocidad angular derecha.

			<i>alfa</i>				
			GN	MN	Z	MP	GP
<i>distancia</i>	P	ω_{ir}	<i>WGP</i>	<i>WMP</i>	<i>WZ</i>	<i>WMN</i>	<i>WGN</i>
		ω_{dr}	<i>WGN</i>	<i>WMN</i>	<i>WZ</i>	<i>WMP</i>	<i>WGP</i>
	G	ω_{ir}	<i>WGP</i>	<i>WMP</i>	<i>WGP</i>	<i>WMN</i>	<i>WGN</i>
		ω_{dr}	<i>WGN</i>	<i>WMN</i>	<i>WGP</i>	<i>WMP</i>	<i>WGP</i>

3.3.2. Sistema Difuso de Orientación

El objetivo del sistema difuso de orientación es de orientar al robot móvil con ruedas hacia su orientación deseada. Este sistema difuso es de tipo Mamdani con una variable lingüística de entrada y a la salida está considerando una o dos variables de salida. La variable lingüística de entrada es llamada error theta (e_θ).

En la Figura 3.11 se muestra la variable lingüística error theta con sus respectivos términos lingüísticos, los cuales son representados por dos funciones de membresía tipo trapezoidal y tres tipo triangular, definidas en un universo de discurso dentro del intervalo de $[-2\pi, 2\pi]$ radianes. Los términos lingüísticos asociados a esta variable son error theta grande negativo (GN), mediano negativo(MN), cero(Z), mediano positivo(MP)y grande positivo(GP). Para calcular la magnitud de la entrada error theta se aplica la siguiente expresión:

$$e_\theta = \theta_d(t) - \theta(t). \quad (3.11)$$

En la Tabla 3.6 se muestran los parámetros de las funciones de membresía de la variable lingüística error theta(e_θ).

Tabla 3.6: Parámetros de las funciones de membresía de las variable lingüística error theta(e_θ).

Variable lingüística	Término lingüístico	Parámetros
error theta(e_θ)	Grande Negativo(GN)	$(-2\pi, -2\pi, -1, -0.5)$
	Mediano Negativo(MN)	$(-1, -0.5, 0)$
	Cero(Z)	$(-0.5, 0, 0.5)$
	Mediano Positivo(MP)	$(0, 0.5, 1)$
	Grande Positivo(GP)	$(0.5, 1, 2\pi, 2\pi)$

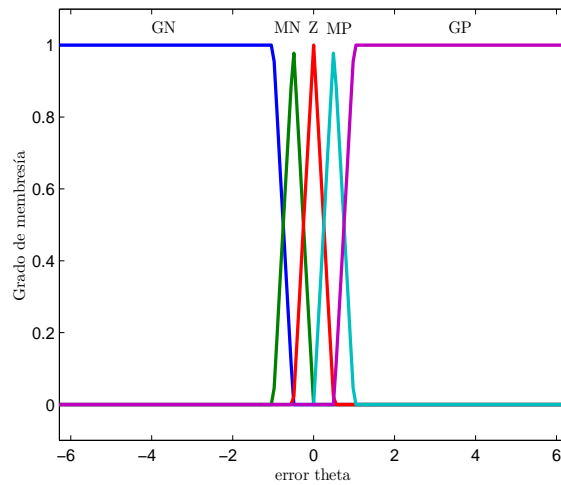


Figura 3.11: Variable lingüística error theta.

Como se mencionó anteriormente, el sistema difuso de orientación está considerando a su salida una o dos variables lingüísticas. Cuando se trabaja en la forma de una entrada y una salida, la variable lingüística asociada a dicha salida es velocidad angular, cuyas características son las mismas que se mencionaron anteriormente para la variable lingüística de salida velocidad angular del sistema difuso de posición mostrada en la Figura 3.8. De igual manera para realizar la conversión de velocidad angular a velocidad angular izquierda y derecha de las ruedas del robot móvil, se aplican (3.9) y (3.10) considerando a la velocidad lineal igual a cero.

A partir de la cantidad de los términos lingüísticos asociados a cada variable, se obtienen 5 reglas difusas las cuales se expresan en la Tabla 3.7, en donde se indica la relación que existe entre variable lingüística de entrada y salida.

Cuando se considera trabajar con dos variables lingüísticas de salida, las variables asociadas son velocidad angular izquierda y velocidad angular derecha. Estas variables poseen las mismas características de las variables de salida propuestas para el sistema difuso de posición mostradas en las Figuras 3.9 y 3.10.

En la Tabla 3.8 se muestra la base de reglas difusas que relaciona la variable

Tabla 3.7: Base de reglas difusas con respecto a la variable lingüística velocidad angular.

<i>error theta</i>	ω_r
GN	<i>WGP</i>
MN	<i>WMP</i>
Z	<i>WZ</i>
MP	<i>WMN</i>
GP	<i>WGN</i>

lingüística error theta con las variables de salida velocidad angular izquierda y velocidad angular derecha.

Tabla 3.8: Base de reglas difusas con respecto a las variables lingüísticas de salida velocidad angular izquierda y velocidad angular derecha.

<i>error theta</i>	ω_{ir}	ω_{dr}
GN	<i>WGN</i>	<i>WGP</i>
MN	<i>WMN</i>	<i>WMP</i>
Z	<i>WZ</i>	<i>WZ</i>
MP	<i>WMP</i>	<i>WMN</i>
GP	<i>WGP</i>	<i>WGN</i>

En el capítulo 4, serán mostradas las superficies de control de los sistemas difusos desarrollados en este trabajo de tesis, con el fin de realizar comparaciones entre sistemas difusos desarrollados en Matlab y VHDL.

3.4. Codificación en VHDL del Control de Regulación de Postura

Los sistemas difusos presentados en este trabajo se codificaron en el lenguaje de programación VHDL. Para desarrollar la codificación se emplearon los métodos que fueron propuestos en [33, 34, 35] en los cuales se desarrollan las etapas de fuzzificación, inferencia y defuzzificación de un sistema difuso en VHDL para su implementación en un FPGA Spartan 3.

3.4.1. Codificación en VHDL del Sistema Difuso de Posición

En la Figura 3.12 se muestra la entidad del sistema difuso de posición descrito en la sección 3.3.1 codificado en VHDL considerando las variables lingüísticas de salida velocidad lineal y velocidad angular, el cual está compuesto por cinco entradas y dos salidas. Las entradas son distancia (8:1) alfa (8:1), clock enable(ce), clock(clk) y reset(rst), y las salidas son vsal (8:1) y wsal (8:1). La entrada distancia y alfa, son las entradas del sistema difuso diseñadas para trabajar con datos de entrada de 8 bits, la entrada clock(clk) corresponde a la señal de reloj interno del FPGA, la entrada clock enable(ce) tiene como función habilitar o deshabilitar el funcionamiento del sistema y la entrada reset(rst) nos permite inicializar todos los valores del sistema. Las salidas vsal y wsal se diseñaron para trabajar con datos de salida de 8 bits, los cuales corresponden a la velocidad lineal y velocidad angular de referencia respectivamente a las cuales debe trabajar el robot móvil con ruedas para alcanzar la posición deseada.

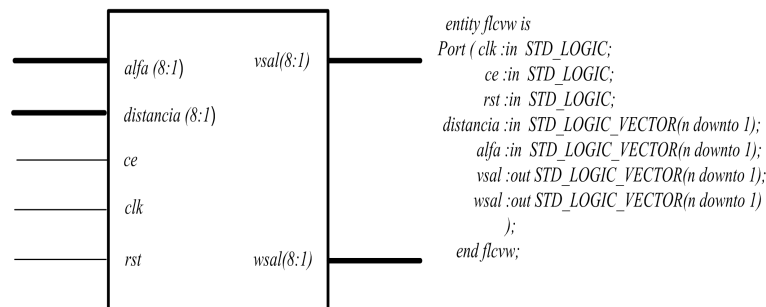


Figura 3.12: Entidad del sistema de control difuso de posición en VHDL con salidas velocidad lineal y velocidad angular.

Ahora se muestra en la Figura 3.13, la entidad del sistema difuso de posición codificado en VHDL considerando las variables lingüísticas de salida velocidad angular izquierda y velocidad angular derecha, el cual está compuesto por cinco entradas y dos salidas. Las entradas que se consideran para este sistema difuso son la mismas que fueron descritas en el sistema anterior solo cambian las salidas que son wdsal (8:1) y wisal (8:1). Las salidas wisal y wdsal se diseñaron para trabajar con datos de salida de 8 bits, los cuales

corresponden a la velocidad angular izquierda y velocidad angular derecha de referencia a las cuales tienen que trabajar las ruedas del robot móvil para alcanzar la posición deseada.

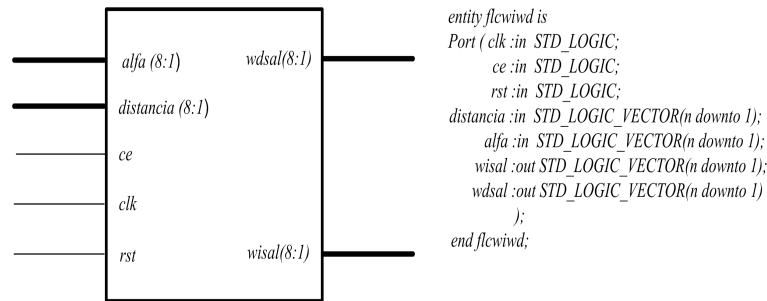


Figura 3.13: Entidad del sistema de control difuso de posición en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.

3.4.2. Codificación en VHDL del Sistema Difuso de Orientación

En la Figura 3.14 se muestra la entidad del sistema difuso de orientación descrito en la sección 3.3.2 codificado en VHDL considerando la variable lingüística de salida velocidad angular, el cual está compuesto por cuatro entradas y una salida. Las entradas son θ (8:1), clock enable (ce), clock (clk) y reset (rst), y la salida es w_{sal} (8:1). La entrada θ , es la entrada del sistema difuso diseñada para trabajar con datos de entrada de 8 bits, los cuales corresponden al cálculo del error de orientación entre la orientación deseada y la orientación actual del robot móvil con ruedas. La salida w_{sal} se diseñó para trabajar con datos de salida de 8 bits, los cuales corresponden a la velocidad angular de referencia a la cual tiene que trabajar el robot móvil para alcanzar la orientación deseada.

Como se comentó en la sección 3.3.2, el sistema difuso de orientación maneja dos tipos de variables lingüísticas de salida, de manera que en la Figura 3.15 se muestra la entidad del sistema codificado en VHDL con respecto a las variables lingüísticas de salidas velocidad angular izquierda y velocidad angular derecha, el cual está constituido por cuatro entradas y dos salidas. Para este sistema se manejan las mismas entradas

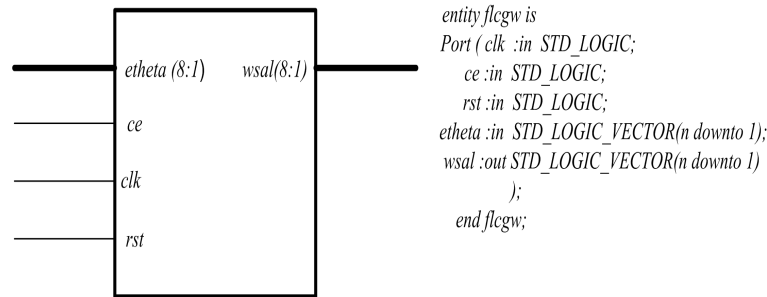


Figura 3.14: Entidad del sistema difuso de orientación en VHDL con salida velocidad angular.

descritas en el sistema anterior solo cambian las salidas que son wdsal (8:1) y wisal (8:1). Las salidas wisal y wdsal se diseñaron para trabajar con datos de salida de 8 bits, los cuales corresponden a la velocidad angular izquierda y velocidad angular derecha de referencia a las cuales tienen que trabajar las ruedas del robot móvil para alcanzar la orientación deseada.

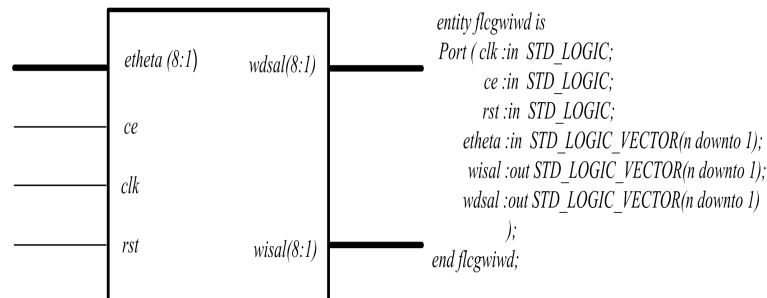


Figura 3.15: Entidad del sistema difuso de orientación en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.

Capítulo 4

Experimentos

En el desarrollo de este trabajo de tesis se utilizaron tres herramientas diferentes de software, las cuales son:

1. Simulink de Mathwork es una herramienta de trabajo que permite desarrollar simulaciones y diseños de modelos para sistemas dinámicos y embebidos en un lenguaje de alto nivel.
2. Xilinx Integrated Software Environment(Xilinx ISE) es un entorno de diseño que consiste en una herramienta que permite hacer un diseño completo basado en lógica programable (tanto CPLD como FPGA), es decir, incluye todas las etapas necesarias como son:
 - La entrada de diseño, bien a través de la captura esquemática, lenguajes de descripción de hardware como ABEL, VHDL o Verilog, o representación gráfica de diagramas de estado.
 - Herramientas de verificación para la obtención de una simulación del sistema, tanto a nivel funcional como de estimación de retardos.
 - Herramientas de implementación donde se permite la especificación de restricciones o indicaciones para realizar una implementación óptima sobre el dispos-

itivo lógico programable especificado.

- Herramientas de programación, para permitir descargar el diseño sobre el dispositivo físico, ya sea en una placa de evaluación o bien en la placa definitiva. De este modo, es posible probar y depurar el sistema sobre hardware de forma rápida y flexible, permitiendo tantos cambios como sean necesarios.
3. Xilinx System Generator (XSG) es una herramienta de diseño para DSP que permite el uso de Simulink para desarrollar diseños en FPGA. Esta herramienta permite generar el código en VHDL de los módulos contenidos dentro de la caja de herramientas de System Generator en Simulink; y viceversa, módulos desarrollados en VHDL pueden ser exportados a Simulink colocando el código generado en un System Generator “Black box”. Esta última característica es utilizada para realizar las pruebas de los sistemas difusos desarrollados en VHDL.

Es importante tener las versiones adecuadas de cada una de las herramientas de software utilizadas. En este caso se utilizan las siguientes versiones:

1. Matlab/Simulink versión 7.1 (R14).
2. Xilinx ISE versión 8.2.03i.
3. Xilinx System Generator versión 8.2.

4.1. Prueba del Sistema Difuso de Posición

En esta sección se muestran los resultados que se obtuvieron al realizar la comparación de los sistemas difusos de posición codificados en VHDL con respecto a los desarrollados en Matlab-Simulink.

4.1.1. Prueba del Sistema Difuso de Posición con Salidas Velocidad Lineal y Velocidad Angular

En la Figura 4.1 se muestra el modelo desarrollado en Matlab-Simulink para realizar las pruebas del sistema difuso de posición codificado en VHDL, con salidas velocidad lineal y velocidad angular, descrito en la sección 3.4.1. Este modelo está integrado por una serie de bloques que son descritos a continuación. El bloque “distancia” y “alfa” generan un vector de datos cada uno respectivamente que contienen todas las combinaciones posibles de los datos de entrada al sistema difuso que se requieren para obtener la respuesta del sistema y así poder generar las superficies de control de cada una de las salidas, los bloques “calculodistancia”, “calculoalfa”, “calculov” y “calculow” son funciones que nos permiten hacer la conversión de números reales a números binarios y viceversa de números binarios a números reales, el bloque “FLCVW” es un “Black Box” de la caja de herramientas de Xilinx System Generator con el cual se exportó a Simulink el código en VHDL del sistema difuso para ser simulado.

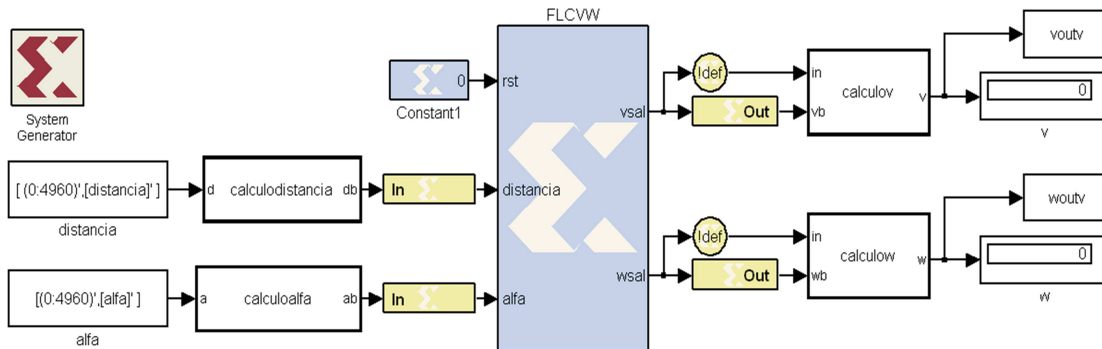
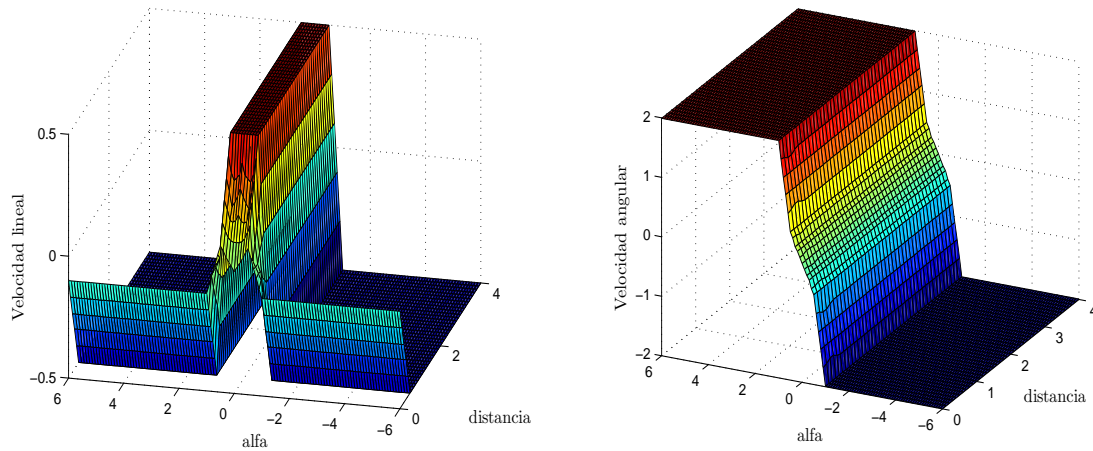


Figura 4.1: Simulación del sistema difuso de posición en VHDL con salidas velocidad lineal y velocidad angular.

En la Figura 4.2 se muestran las superficies de control que se obtuvieron al realizar la simulación del sistema difuso de posición codificado en VHDL con salidas velocidad lineal y velocidad angular.



(a) Velocidad lineal en VHDL

(b) Velocidad angular en VHDL

Figura 4.2: Superficies de control del sistema difuso de posición en VHDL con salidas velocidad lineal y velocidad angular.

Con el fin de comparar las superficies de control obtenidas del sistema difuso codificado en VHDL, se desarrolló en Matlab-Simulink el sistema descrito en la sección 3.3.1 considerando las salidas velocidad lineal y velocidad angular. En la Figura 4.3 se muestra el sistema difuso desarrollado en Matlab-Simulink. El bloque “FLCVWMATLAB”, que es un subsistema de Simulink, contiene el código del sistema difuso de posición con salidas velocidad lineal y velocidad angular.

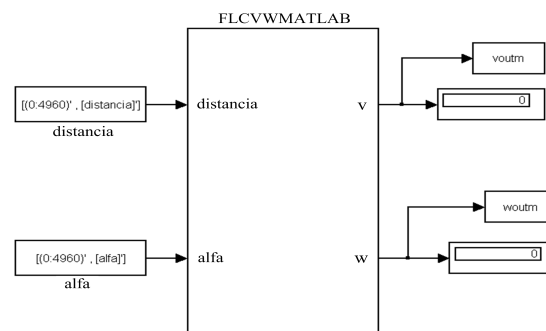
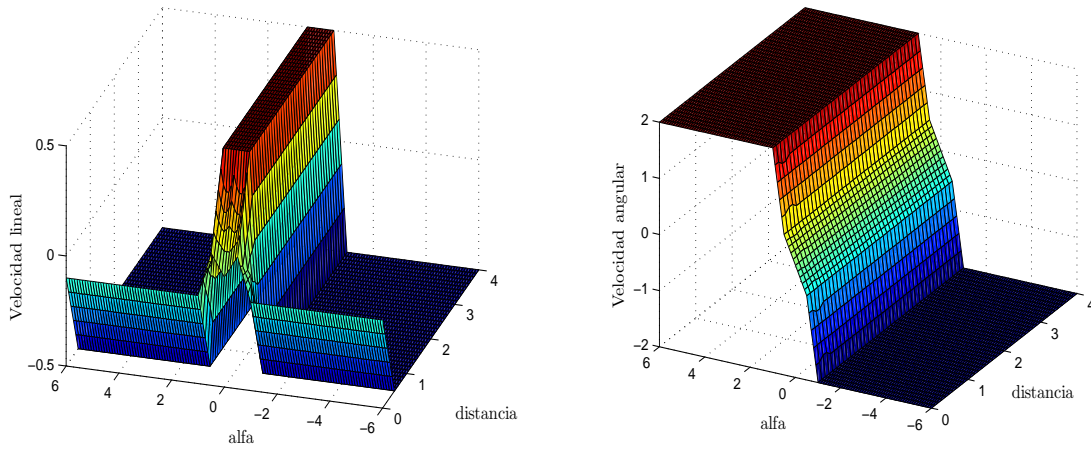


Figura 4.3: Simulación del sistema difuso de posición en Matlab-Simulink con salidas velocidad lineal y velocidad angular.

En la Figura 4.4 se muestran las superficies de control que se obtuvieron al realizar la simulación del sistema difuso desarrollado en Matlab-Simulink.



(a) Velocidad lineal en Matlab-Simulink

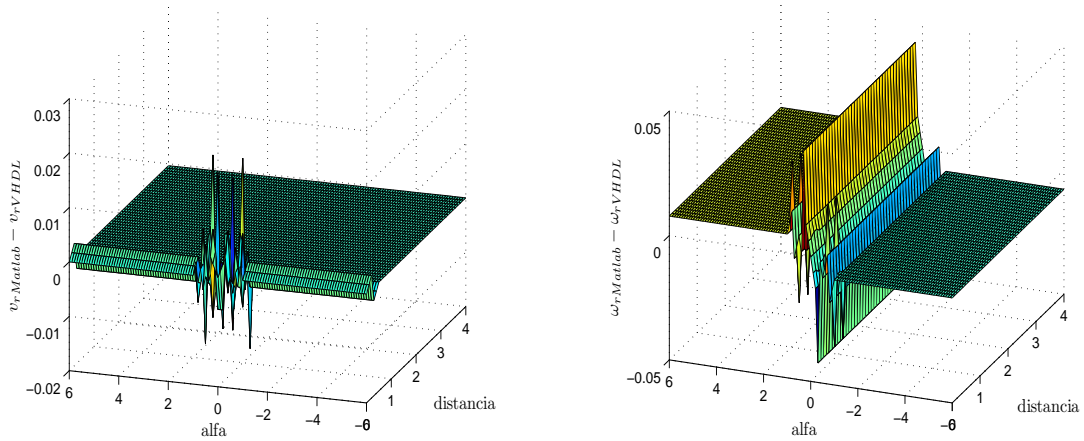
(b) Velocidad angular en Matlab-Simulink

Figura 4.4: Superficies de control del sistema difuso de posición en Matlab-Simulink con salidas velocidad lineal y velocidad angular.

Al observar las superficies de control que se obtuvieron de los sistemas difusos de posición codificados en VHDL y Matlab-Simulink, considerando las variables de salida velocidad lineal y velocidad angular, no se alcanzan a distinguir a simple vista las diferencias que puedan existir entre ellas, por ello se procedió a obtener el error que existe entre cada una de las superficies de control obtenidas. En la Figura 4.5 se muestra gráficamente el error obtenido de hacer las comparaciones de cada una de las superficies de control.

4.1.2. Prueba del Sistema Difuso de Posición con Salidas Velocidad Angular Izquierda y Velocidad Angular Derecha

Para realizar la simulación del sistema difuso de posición en VHDL con salidas velocidad angular izquierda y velocidad angular derecha descrito en la sección 3.4.1, se desarrolló un modelo en Simulink que se muestra en la Figura 4.6. El bloque “FLCWI-WD” contiene inmerso el código en VHDL del sistema difuso, los bloques “calculowi”



(a) Error velocidad lineal

(b) Error velocidad angular

Figura 4.5: Error de superficies de control del sistema difuso de posición con salidas velocidad lineal y velocidad angular codificado en VHDL y en Matlab-Simulink.

y “calculowd” son funciones que realizan la conversión de números binarios a números reales.

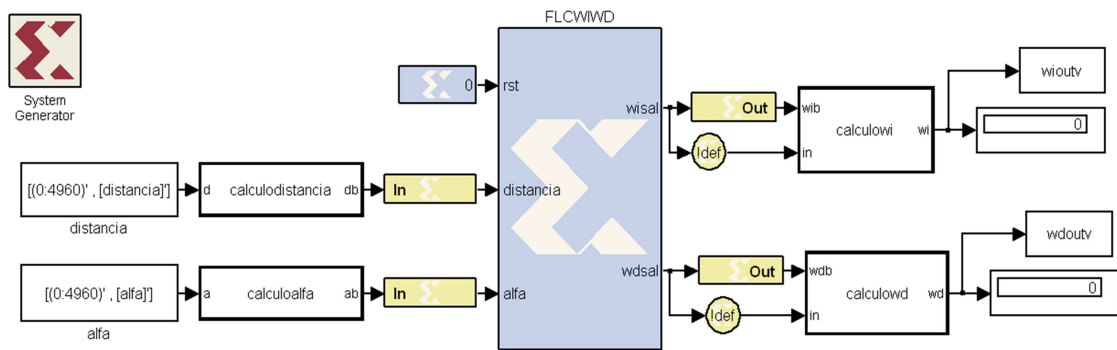
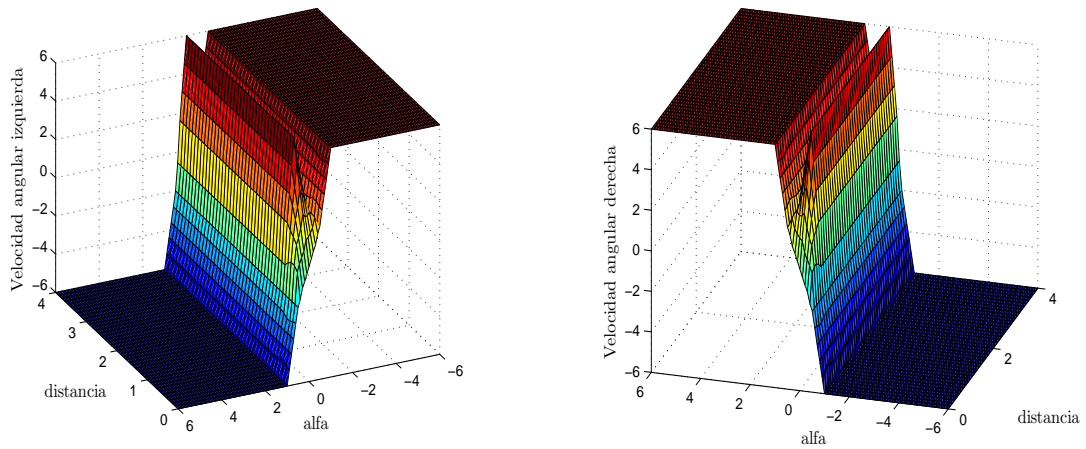


Figura 4.6: Simulación del sistema difuso de posición en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.

En la Figura 4.7 se muestran las superficies de control que se obtuvieron al realizar la simulación del sistema difuso de posición codificado en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.



(a) Velocidad angular izquierda en VHDL

(b) Velocidad angular derecha en VHDL

Figura 4.7: Superficies de control del sistema difuso de posición en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.

En la Figura 4.8 se muestra el modelo en Matlab-Simulink del sistema difuso de posición con salidas velocidad angular izquierda y velocidad angular derecha descrito en la sección 3.3.1, el cual se utiliza para realizar las comparaciones con el sistema difuso codificado en VHDL. Dentro del bloque “FLCWIWDMATLAB” se encuentra el código del sistema difuso realizado con funciones de Matlab.

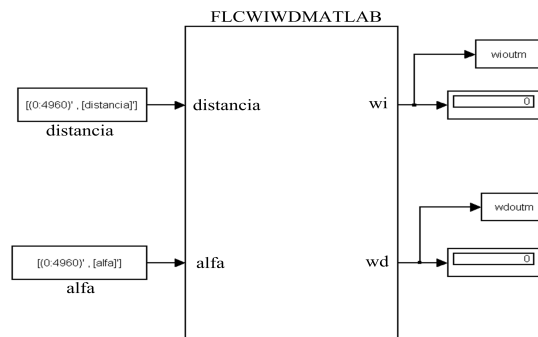
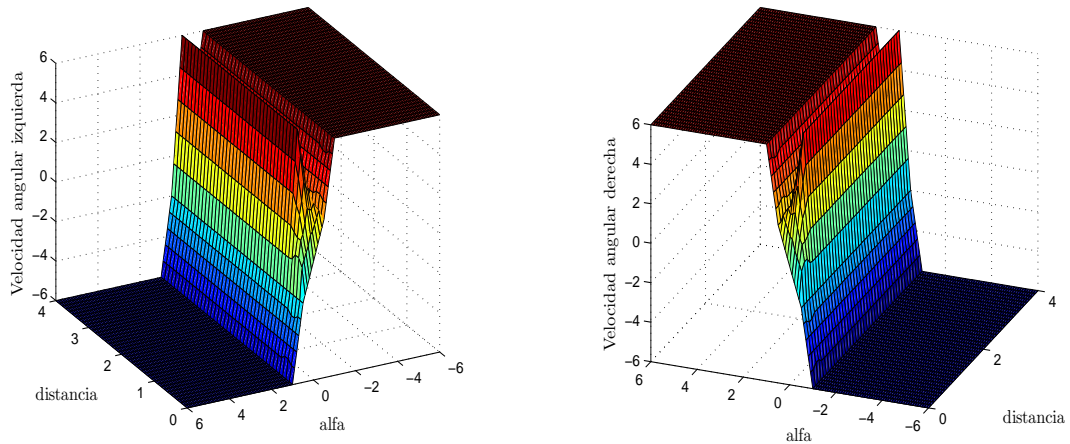


Figura 4.8: Simulación del sistema difuso de posición en Matlab-Simulink con salidas velocidad angular izquierda y velocidad angular derecha.

En la Figura 4.9 se observan las superficies de control que se obtuvieron para cada una de sus salidas al realizar la simulación del sistema difuso desarrollado en Matlab-Simulink.



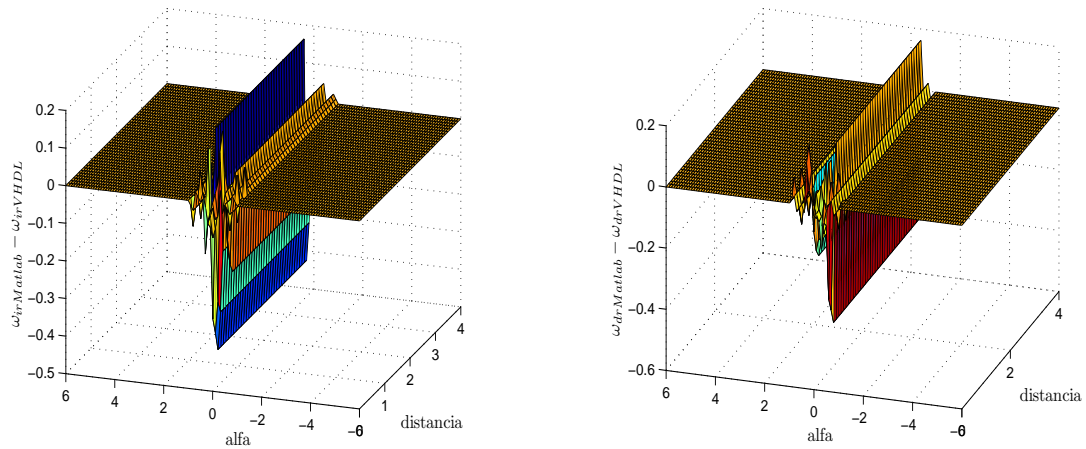
(a) Velocidad angular izquierda en Matlab-Simulink (b) Velocidad angular derecha en Matlab-Simulink

Figura 4.9: Superficies de control del sistema difuso de posición en Matlab-Simulink con salidas velocidad angular izquierda y velocidad angular derecha.

En la Figura 4.10 se muestra el error que se obtuvo al comparar las superficies de control que se obtuvieron al realizar las simulaciones de los sistemas difusos de posición con salidas velocidad angular izquierda y velocidad angular derecha codificado en VHDL y en Matlab-Simulink.

4.2. Prueba del Sistema Difuso de Orientación

En esta sección se muestran los resultados que se obtuvieron al realizar la comparación de los sistemas difusos de orientación codificados en VHDL y en Matlab-Simulink.



(a) Error velocidad angular izquierda

(b) Error velocidad angular derecha

Figura 4.10: Error de superficies de control del sistema difuso de posición con salidas velocidad angular izquierda y velocidad angular derecha codificado en VHDL y en Matlab-Simulink.

4.2.1. Prueba del Sistema Difuso de Orientación con Salida Velocidad Angular

En la Figura 4.11 se muestra el modelo en Simulink que se desarrolló para realizar las pruebas del sistema difuso de orientación codificado en VHDL con salida velocidad angular descrito en la sección 3.4.2, el cual está integrado por una serie de bloques que se describen a continuación. El bloque “etheta” es una función que genera un vector de datos de entrada para el sistema difuso que nos permita obtener su respuesta y así generar su superficie de control, el bloque “calculoetheta” convierte los datos de entrada reales a datos de entrada binario, y en el Black Box “FLCW” se encuentra inmerso el código en VHDL del sistema difuso.

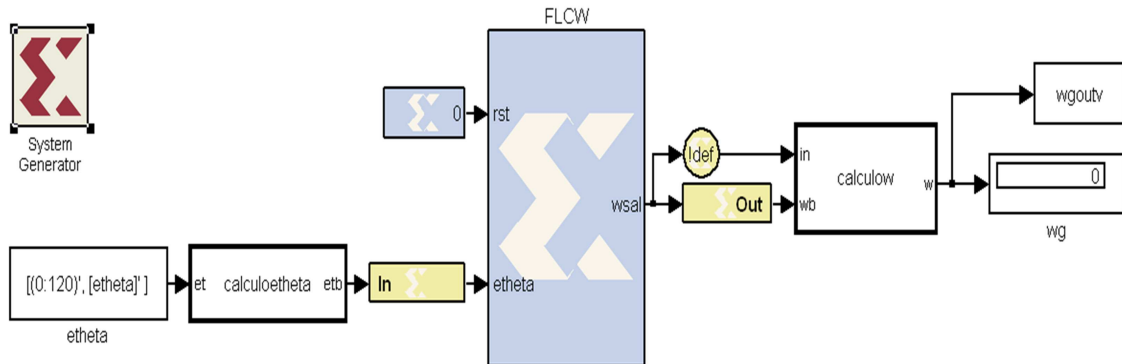


Figura 4.11: Simulación del sistema difuso de orientación en VHDL con salida velocidad angular.

En la Figura 4.12 se muestra la superficie de control que se obtuvo al realizar la simulación del sistema difuso de orientación codificado en VHDL con salida velocidad angular.

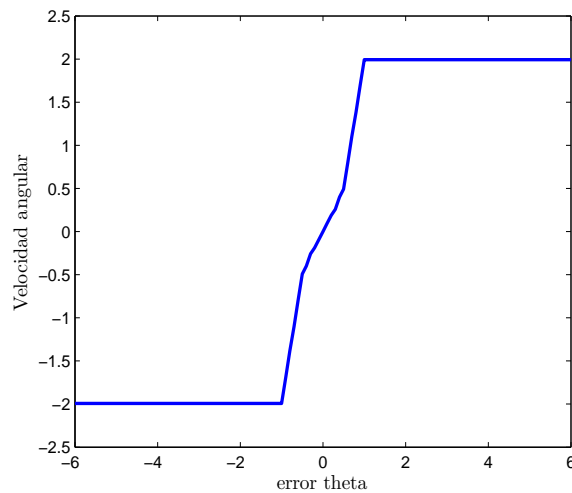


Figura 4.12: Superficie de control del sistema difuso de orientación en VHDL con salida velocidad angular.

En la Figura 4.13 se muestra el modelo desarrollado en Matlab-Simulink del sistema difuso de orientación con salida velocidad angular descrito en la sección 3.3.2 para realizar las comparaciones con el sistema difuso codificado en VHDL. Dentro del bloque

“FLCGWMATLAB”, que es un subsistema de Simulink, se encuentra el código del sistema difuso programado con funciones de Matlab.

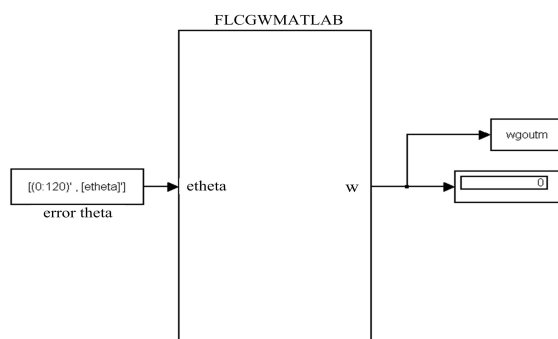


Figura 4.13: Simulación del sistema difuso de orientación en Matlab-Simulink con salida velocidad angular.

La superficie de control que se obtuvo al realizar la simulación del sistema difuso de orientación con salida velocidad angular desarrollado en Matlab-Simulink se presenta en la Figura 4.14.

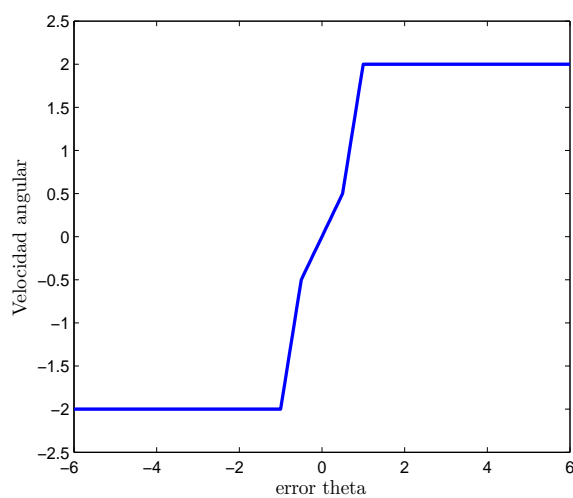


Figura 4.14: Superficie de control del sistema difuso de orientación en Matlab-Simulink con salida velocidad angular.

En la Figura 4.15 se muestra el error que se obtuvo al comparar las superficies de control que se obtuvieron al realizar las simulaciones de los sistemas difusos de orientación con salida velocidad angular codificado en VHDL y en Matlab-Simulink.

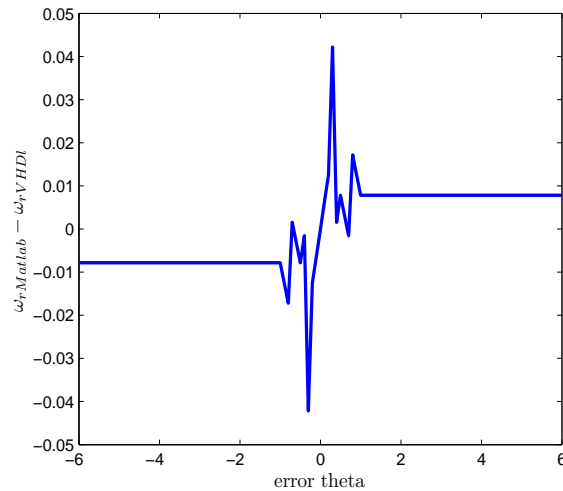


Figura 4.15: Error de superficies de control del Sistema difuso de orientación con salida velocidad angular codificado en VHDL y en Matlab-Simulink.

4.2.2. Prueba del Sistema Difuso de Orientación con Salidas Velocidad Angular Izquierda y Velocidad Angular Derecha

Para realizar las simulaciones del sistema difuso de orientación en VHDL con salidas velocidad angular izquierda y velocidad angular derecha descrito en la sección 3.4.2, se desarrolló un modelo en Simulink que se muestra en la Figura 4.16. El código en VHDL del sistema difuso se encuentra inmerso dentro del Black Box “FLCGWIWD”.

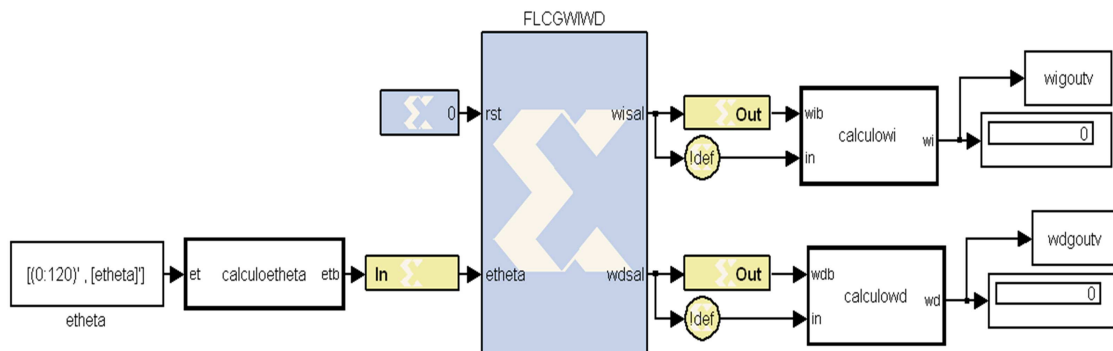
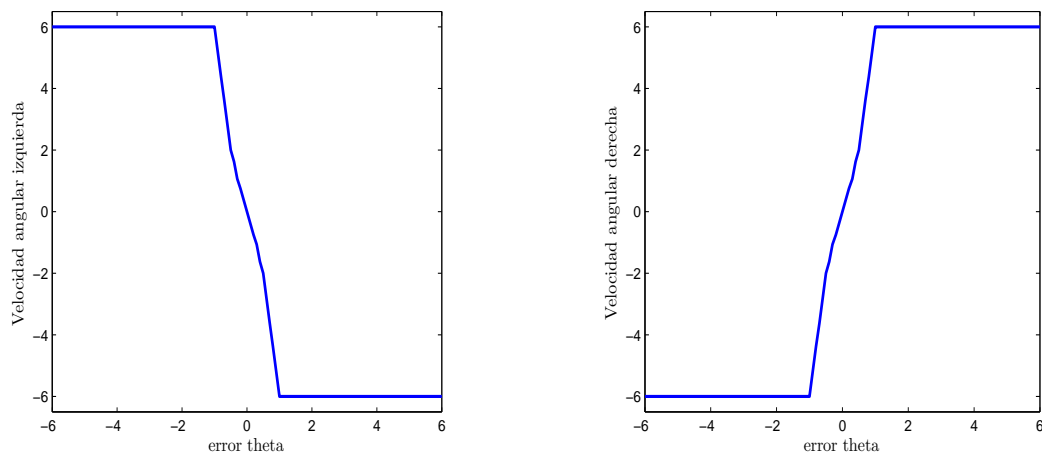


Figura 4.16: Simulación del sistema difuso de orientación en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.

En la Figura 4.17 se muestran las superficies de control que se obtuvieron al realizar la simulación del sistema difuso de orientación en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.



(a) Velocidad angular izquierda en VHDL (b) Velocidad angular derecha en Matlab-Simulink

Figura 4.17: Superficies de control del sistema difuso de orientación en VHDL con salidas velocidad angular izquierda y velocidad angular derecha.

En la Figura 4.18 se muestra el modelo en Matlab-Simulink del sistema difuso de orientación con salidas velocidad angular izquierda y velocidad angular derecha descrito en la sección 3.3.2, para realizar las comparaciones con el sistema difuso codificado en VHDL.

Dentro del bloque “FLCGWIWDMATLAB”, el cual es un subsistema de Simulink, se encuentra el código del sistema difuso realizado con funciones de Matlab.

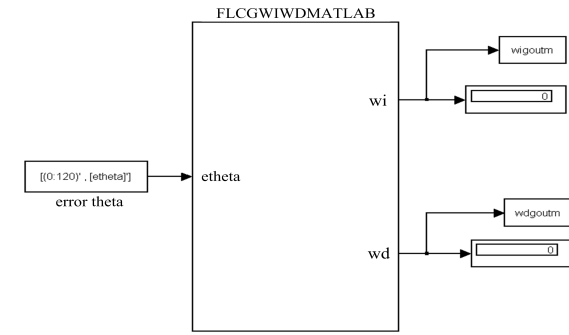
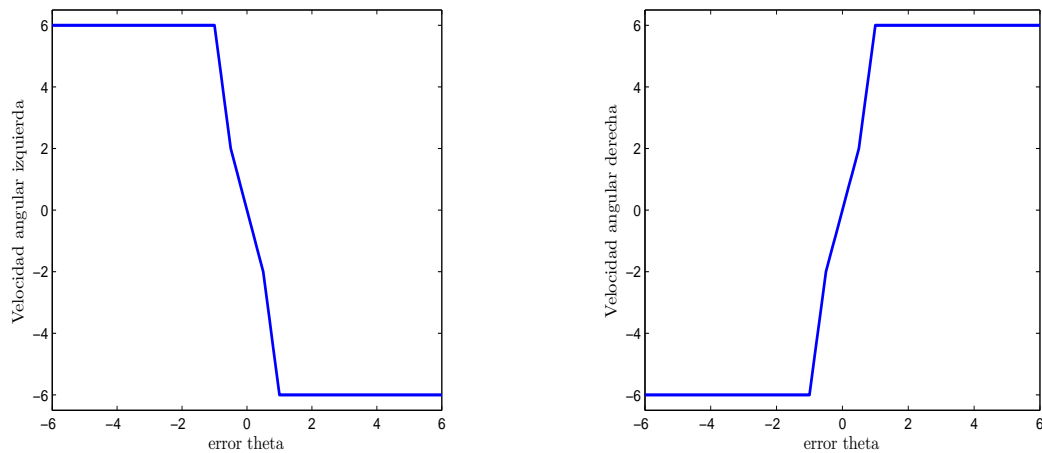


Figura 4.18: Simulación del sistema difuso de orientación en Matlab-Simulink con salidas velocidad angular izquierda y velocidad angular derecha.

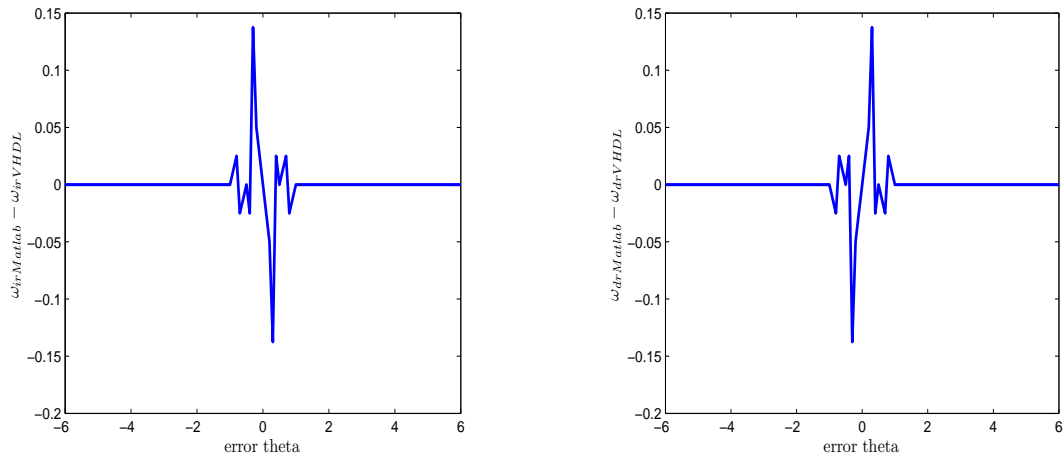
Las superficies de control que se obtuvieron al realizar la simulación del sistema difuso de orientación con salidas velocidad angular izquierda y velocidad angular derecha desarrollado en Matlab-Simulink se presentan en la Figura 4.19.



(a) Velocidad angular izquierda en Matlab-Simulink (b) Velocidad angular derecha en Matlab-Simulink

Figura 4.19: Superficies de control del sistema difuso de orientación en Matlab-Simulink con salidas velocidad angular izquierda y velocidad angular derecha.

En la Figura 4.20 se muestra el error que se obtuvo al comparar las superficies de control, las cuales se obtuvieron al realizar las simulaciones de los sistema difusos de orientación con salida velocidad angular izquierda y velocidad angular derecha codificado en VHDL y en Matlab-Simulink.



(a) Error velocidad angular izquierda

(b) Error velocidad angular derecha

Figura 4.20: Error de superficies de control del sistema difuso de orientación con salidas velocidad angular izquierda y velocidad angular derecha codificado en VHDL y en Matlab-Simulink.

4.3. Prueba del Sistema de Control de Movimiento del Robot Móvil con Ruedas

En la Figura 4.21 se muestra la implementación en Simulink del sistema de control de movimiento del robot móvil con ruedas descrito en el diagrama a bloques de la Figura 3.3, el cual se utiliza para realizar la parte experimental y así validar el funcionamiento de los sistemas difusos desarrollados en VHDL y Matlab-Simulink.

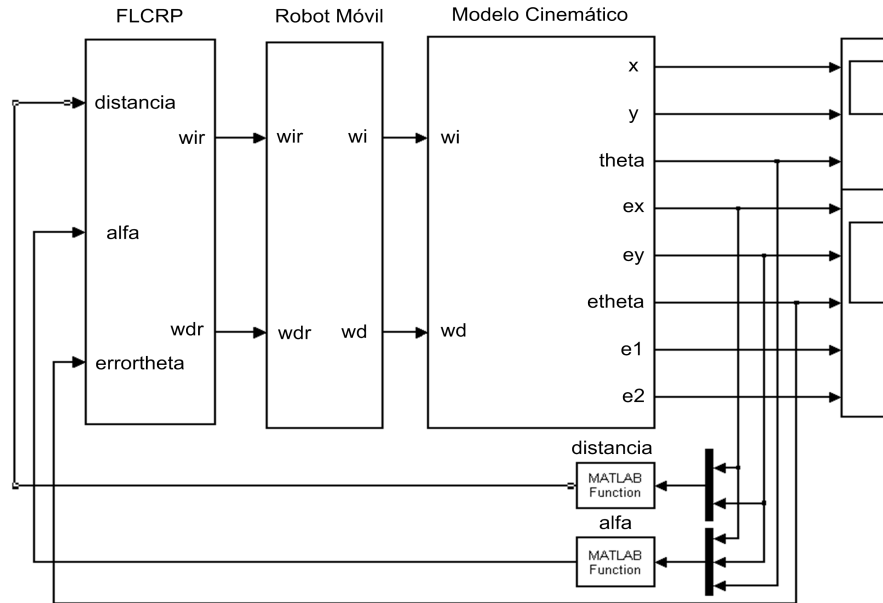


Figura 4.21: Sistema de Control del Robot Móvil de Ruedas desarrollado en Simulink.

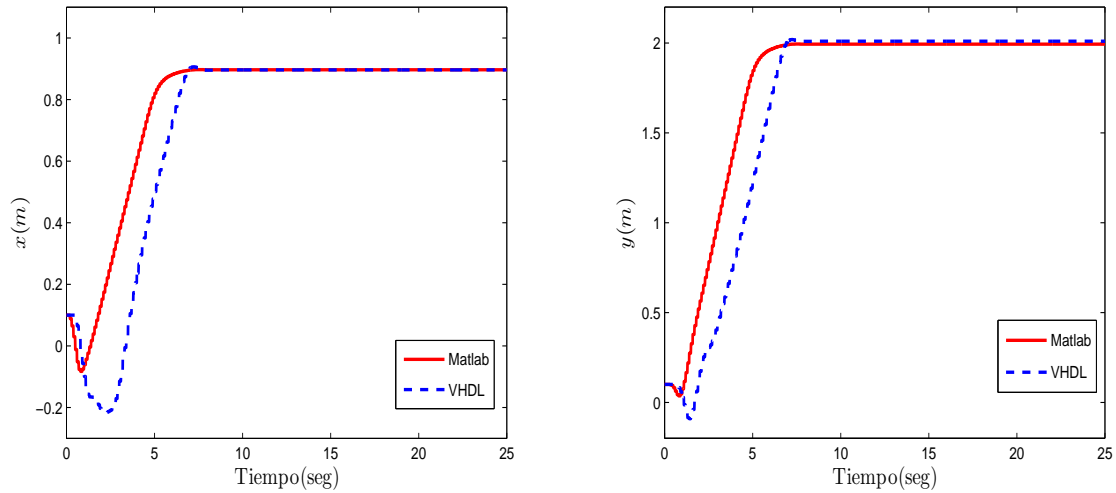
4.3.1. Resultados Experimentales

A continuación se presentan los resultados obtenidos de las simulaciones realizadas al sistema de control de movimiento del robot móvil de ruedas descrito en la sección 3.2. Para el desarrollo de las simulaciones se consideran las siguientes características:

1. La posición y orientación inicial del robot móvil de ruedas es igual a $q_0 = (0.1, 0.1, 0)^T$.
2. El radio de las ruedas del robot móvil es igual a $r = 0.08m$.
3. La distancia entre las ruedas actuadas del robot móvil es igual a $d = 0.45m$.

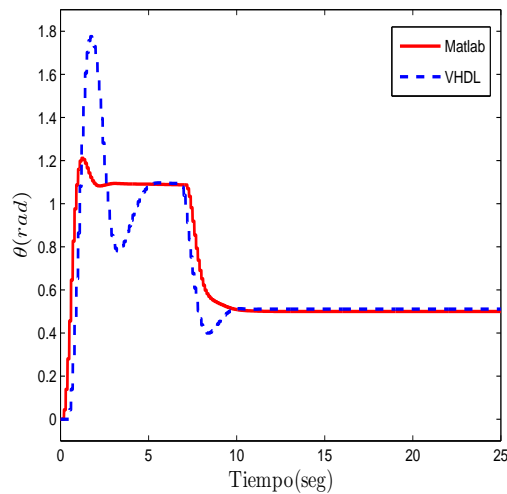
Experimento 1.- El robot móvil de ruedas debe alcanzar la posición y orientación deseada $q_d = (0.9, 2, 0.5)^T$. Para realizar este experimento se utiliza el sistema difuso de control de posición descrito en la sección 4.1.1 y el de control de orientación descrito en la sección 4.2.1, para resolver el problema de regulación de postura del robot móvil de ruedas, los cuales fueron desarrollados en VHDL y Matlab-Simulink para realizar comparaciones entre ambos. En la Figura 4.22 se observa la evolución en el tiempo

de la posición y orientación del robot móvil de ruedas utilizando los sistemas difusos desarrollados en VHDL y Matlab-Simulink.



(a) Posición final en x .

(b) Posición final en y

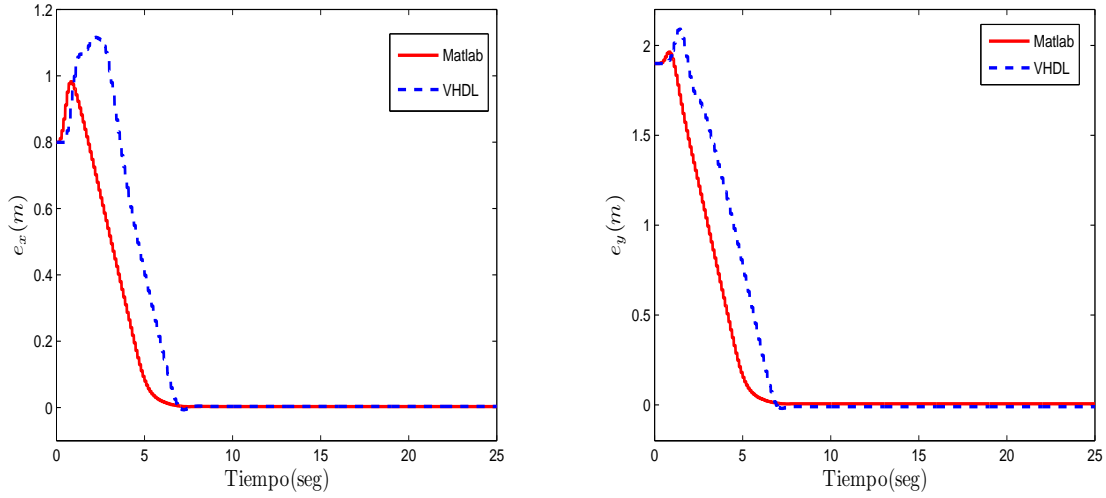


(c) Orientación final en θ

Figura 4.22: Experimento 1. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.

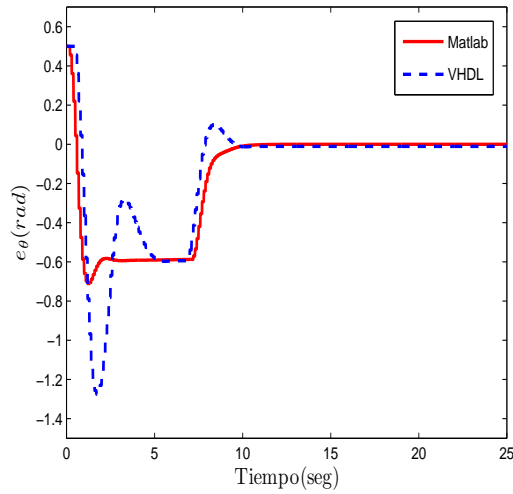
En la Figura 4.23 se muestra el error de posición en x y y , y el error de orientación θ del robot móvil de ruedas, observando claramente que estos tienden a acercarse a cero,

lo cual nos indica que se acerca a la posición y orientación deseada.



(a) Error de posición en x .

(b) Error de posición en y

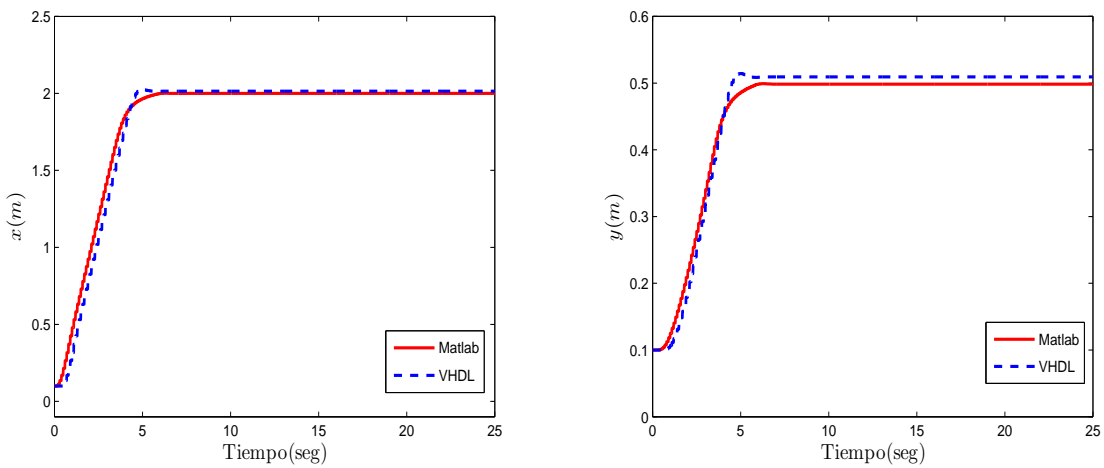


(c) Error de orientación en θ

Figura 4.23: Experimento 1. Error de posición y orientación del robot móvil de ruedas.

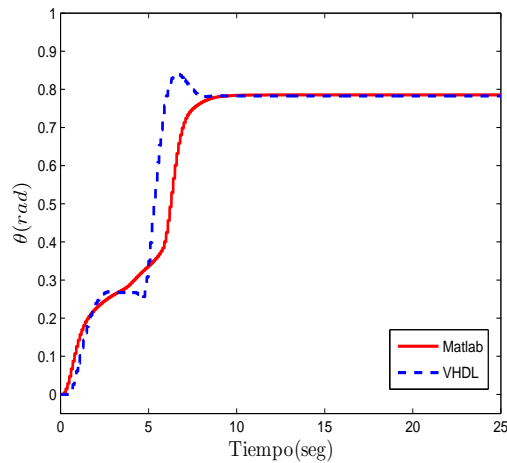
Experimento 2.- El robot móvil de ruedas debe alcanzar la posición y orientación deseada $q_d = (2, 0.5, \frac{\pi}{4})^T$. Al igual que el experimento anterior, en este también se hace uso del sistema difuso de control de posición descrito en la sección 4.1.1 y el

de control de orientación descrito en la sección 4.2.1, para resolver el problema de regulación de postura del robot móvil de ruedas, los cuales fueron desarrollados en VHDL y Matlab-Simulink para realizar comparaciones entre ambos. En la Figura 4.24 se muestra la evolución en el tiempo de la posición y orientación del robot móvil de ruedas utilizando los sistemas difusos desarrollados en VHDL y Matlab-Simulink.



(a) Posición final en x .

(b) Posición final en y



(c) Orientación final en θ

Figura 4.24: Experimento 2. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.

En la Figura 4.25 se observan los errores de posición en x y y , y orientación en θ que se obtuvieron al realizar el experimento 2.

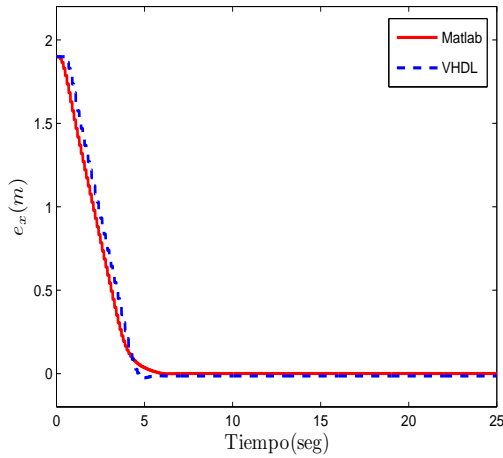
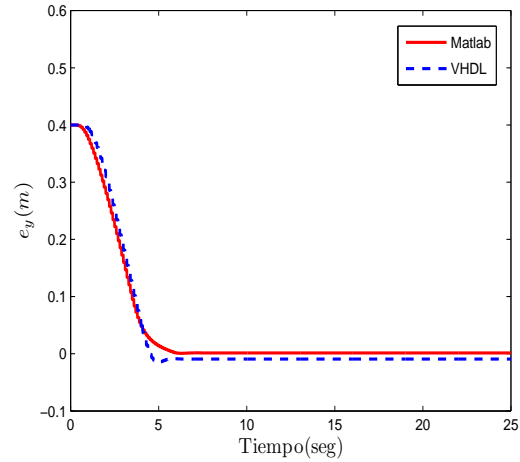
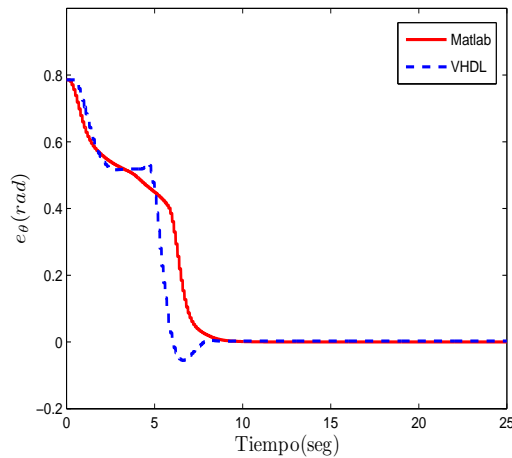
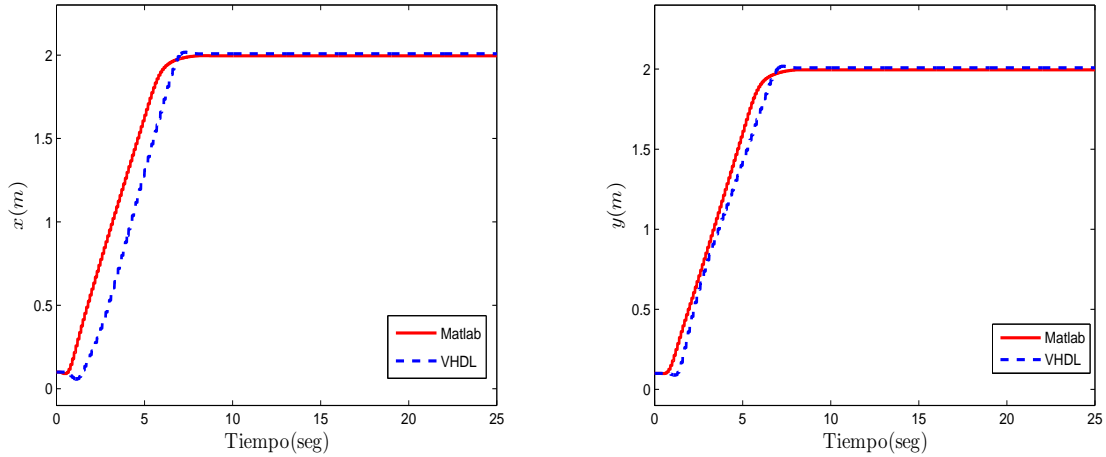
(a) Error de posición en x .(b) Error de posición en y (c) Error de orientación en θ

Figura 4.25: Experimento 2. Error de posición y orientación del robot móvil de ruedas.

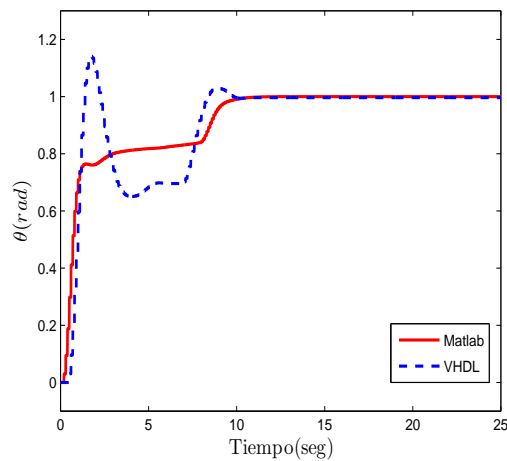
Experimento 3.- El robot móvil de ruedas debe alcanzar la posición y orientación deseada $q_d = (2, 2, 1)^T$. Se considera el sistema difuso de control de posición descrito en la sección 4.1.1 y el de control de orientación descrito en la sección 4.2.1, para resolver el problema de regulación de postura del robot móvil de ruedas, los cuales fueron desarro-

llados en VHDL y Matlab-Simulink para realizar comparaciones entre ambos. En la Figura 4.26 se muestra la evolución en el tiempo de la posición y orientación del robot móvil de ruedas utilizando los sistemas difusos desarrollados en VHDL y Matlab-Simulink.



(a) Posición final en x .

(b) Posición final en y

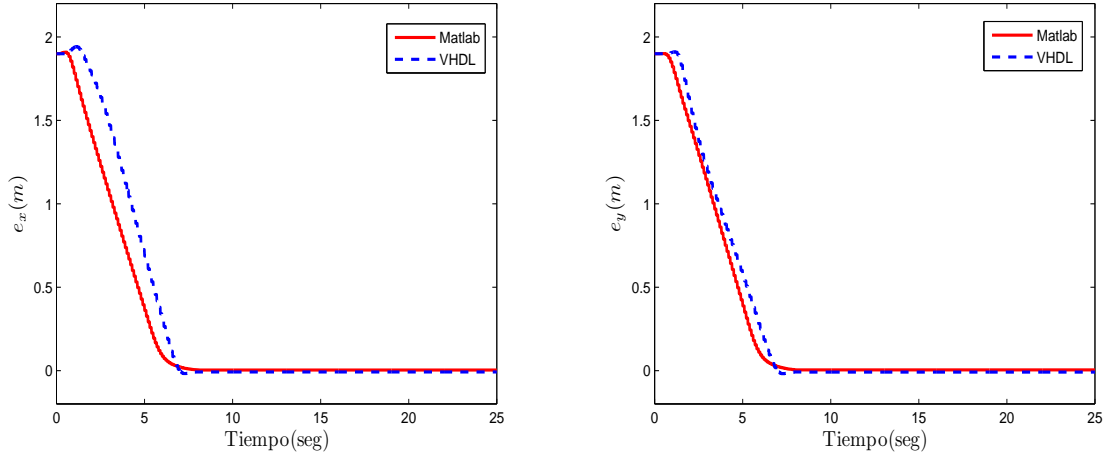


(c) Orientación final en θ

Figura 4.26: Experimento 3. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.

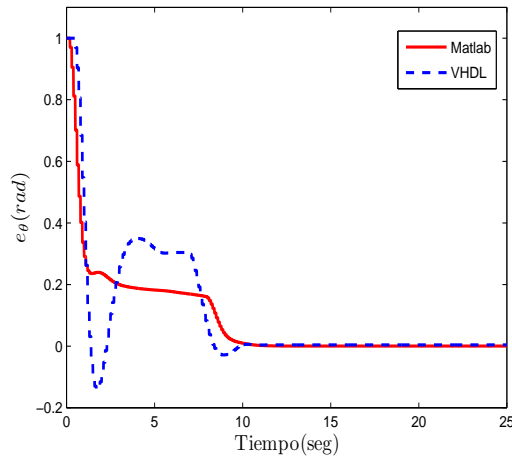
En la Figura 4.27 se muestra el error de posición en x y y , y de orientación en θ al realizar la simulación del sistema del control del robot móvil de ruedas en el experimento

3.



(a) Error de posición en x .

(b) Error de posición en y

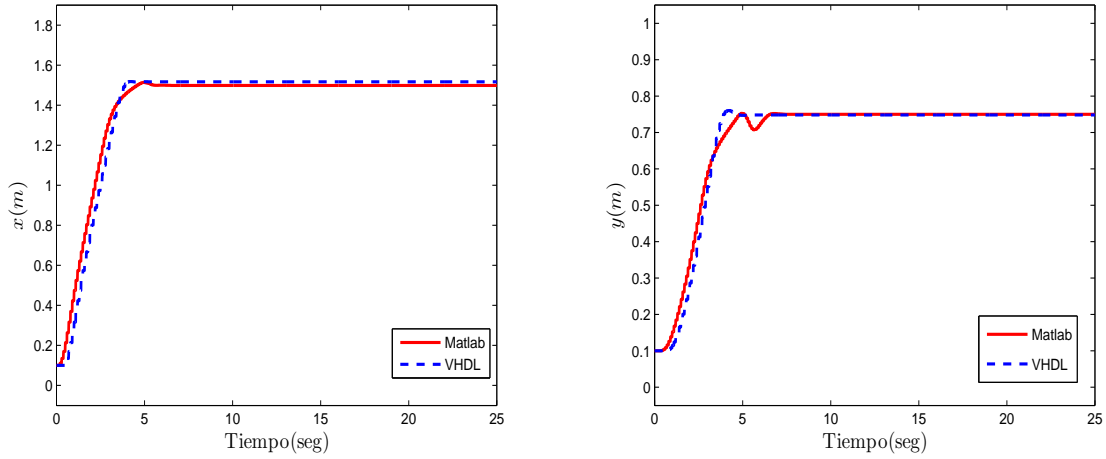


(c) Error de orientación en θ

Figura 4.27: Experimento 3. Error de posición y orientación del robot móvil de ruedas.

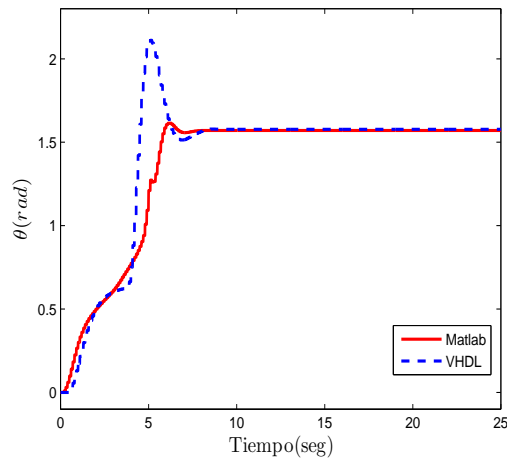
Experimento 4.- El robot móvil de ruedas debe alcanzar la posición y orientación deseada $q_d = (1.5, 0.75, \frac{\pi}{4})^T$. Al igual que en los experimentos anteriores se sigue considerando el sistema difuso de control de posición descrito en la sección 4.2.1 y el de control de orientación descrito en la sección 4.2.1, para resolver el problema de regulación de postura del robot móvil de ruedas, los cuales fueron desarrollados en VHDL y

Matlab-Simulink para realizar comparaciones entre ambos. En la Figura 4.28 se muestra la evolución en el tiempo de la posición y orientación del robot móvil de ruedas utilizando los sistemas difusos desarrollados en VHDL y Matlab-Simulink.



(a) Posición final en x .

(b) Posición final en y

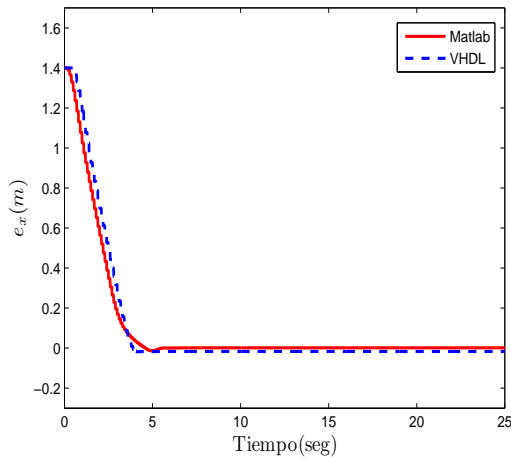


(c) Orientación final en θ

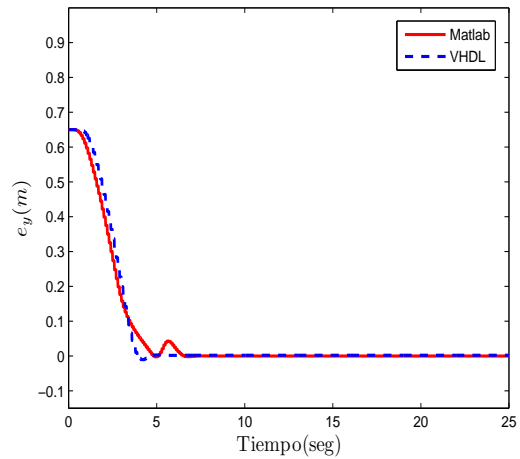
Figura 4.28: Experimento 4. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.

En la Figura 4.29 se observan los errores de posición en x y y , y el error de orientación θ al realizar la simulación del sistema de control del robot móvil de ruedas en

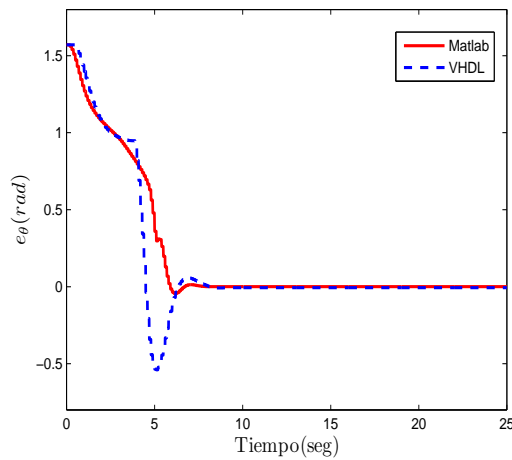
el experimento 4.



(a) Error de posición en x .



(b) Error de posición en y

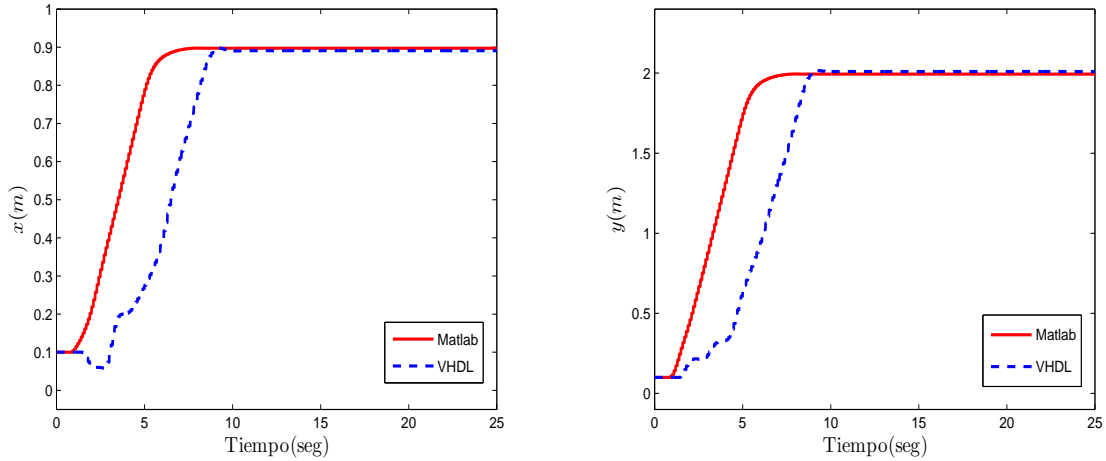


(c) Error de orientación en θ

Figura 4.29: Experimento 4. Error de posición y orientación de robot móvil de ruedas.

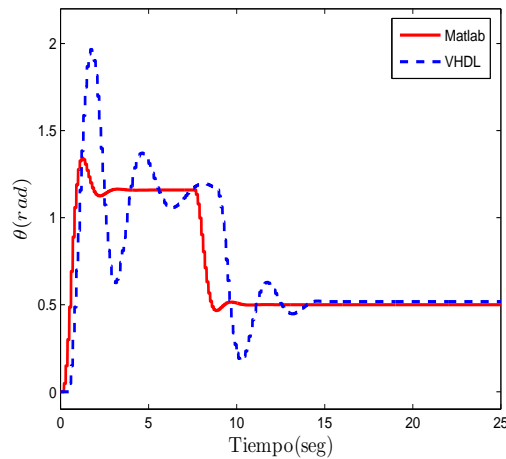
Experimento 5.- El robot móvil de ruedas debe alcanzar la posición y orientación deseada $q_d = (0.9, 2, 0.5)^T$. Para realizar este experimento se utiliza el sistema difuso de control de posición descrito en la sección 4.1.2 y el de control de orientación descrito en la sección 4.2.2 para resolver el problema de regulación de postura del robot móvil de ruedas, los cuales fueron desarrollados en VHDL y Matlab-Simulink para rea-

lizar comparaciones entre ambos. En la Figura 4.30 se observa la evolución en el tiempo de la posición y orientación del robot móvil de ruedas utilizando los sistemas difusos desarrollados en VHDL y Matlab-Simulink.



(a) Posición final en x .

(b) Posición final en y

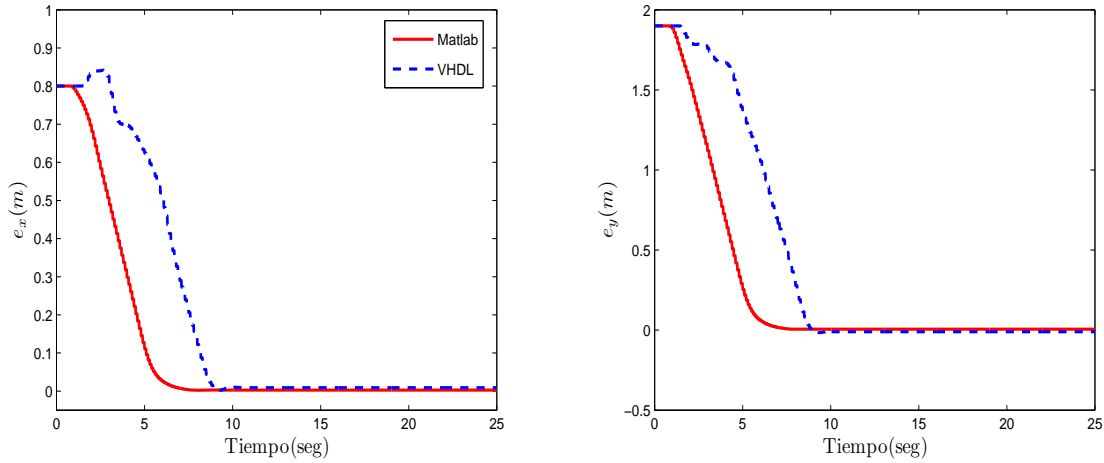


(c) Orientación final en θ

Figura 4.30: Experimento 5. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.

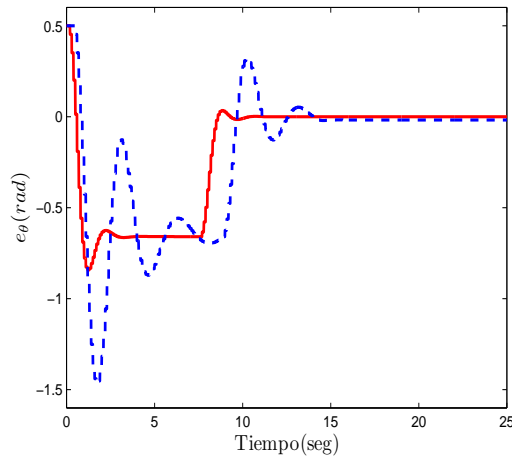
En la Figura 4.31 se muestra el error de posición en x y y , y el error de orientación θ del robot móvil de ruedas que se obtuvieron al realizar la simulación del sistema de

control del robot móvil de ruedas en el experimento 5.



(a) Error de posición en x .

(b) Error de posición en y

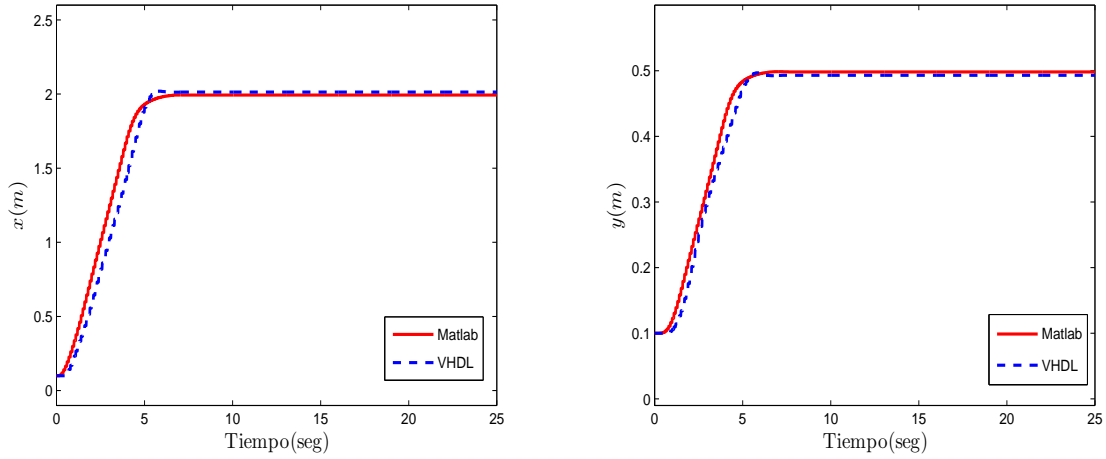


(c) Error de orientación en θ

Figura 4.31: Experimento 5. Error de posición y orientación del robot móvil de ruedas.

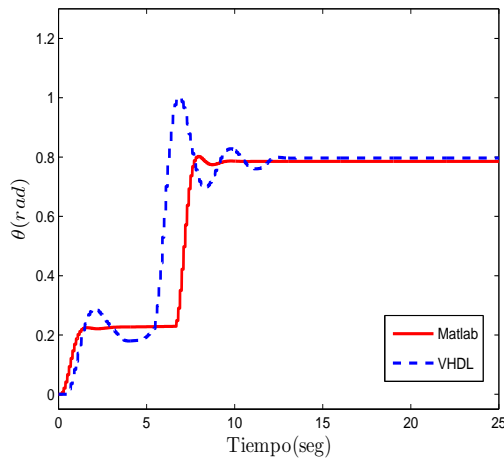
Experimento 6.- El robot móvil de ruedas debe alcanzar la posición y orientación deseada $q_d = (2, 0.5, \frac{\pi}{4})^T$. Al igual que el experimento anterior, en este también se hace uso del sistema difuso de control de posición descrito en la sección 4.1.2 y el de control de orientación descrito en la sección 4.2.2 para resolver el problema de regulación de postura del robot móvil de ruedas, los cuales fueron desarrollados en VHDL y

Matlab-Simulink para realizar comparaciones entre ambos. En la Figura 4.32 se muestra la evolución en el tiempo de la posición y orientación del robot móvil de ruedas utilizando los sistemas difusos desarrollados en VHDL y Matlab-Simulink.



(a) Posición final en x .

(b) Posición final en y



(c) Orientación final en θ

Figura 4.32: Experimento 6. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.

En la Figura 4.33 se observan los errores de posición en x y y , y orientación en θ que se obtuvieron al realizar el experimento 6.

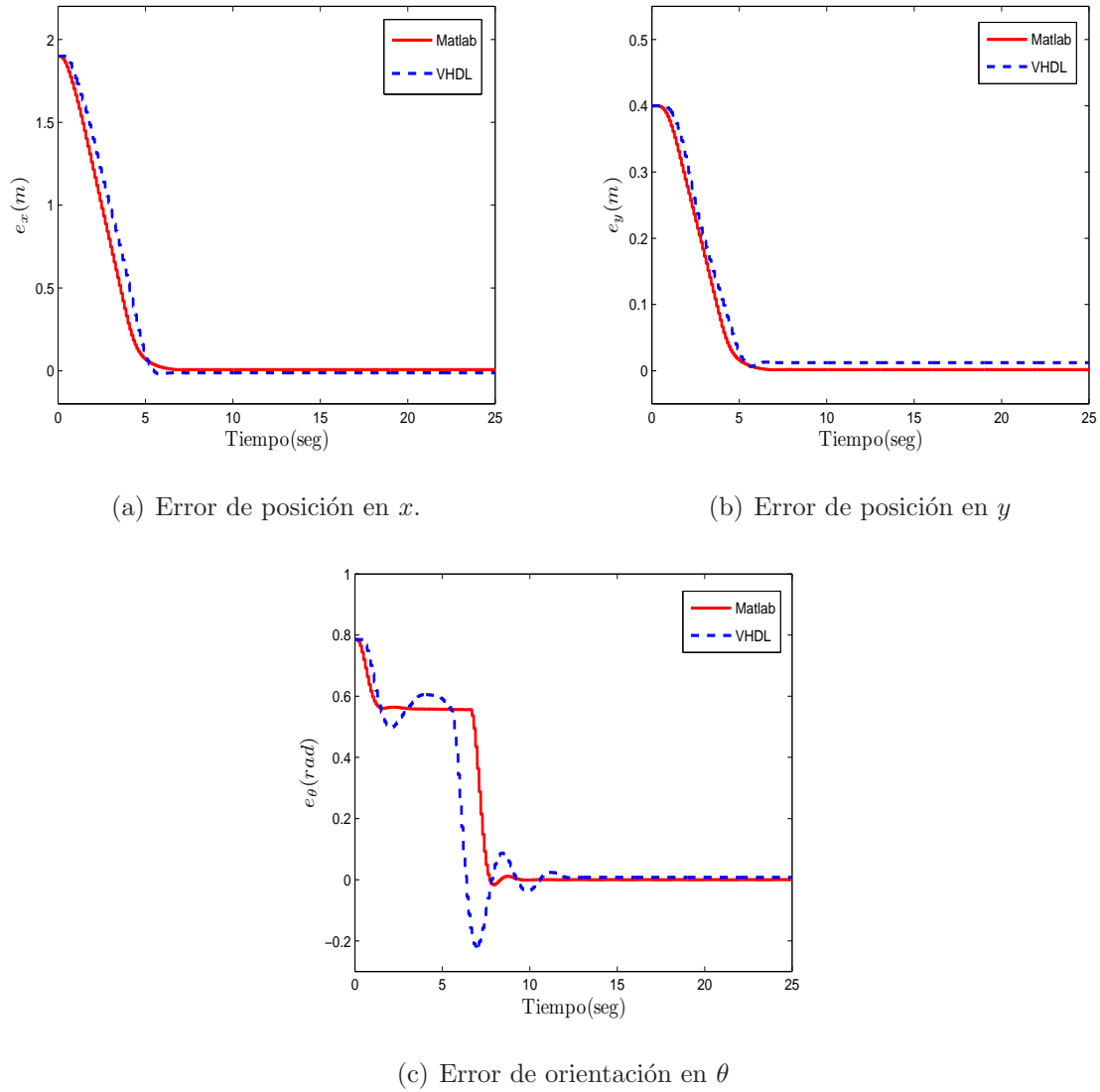
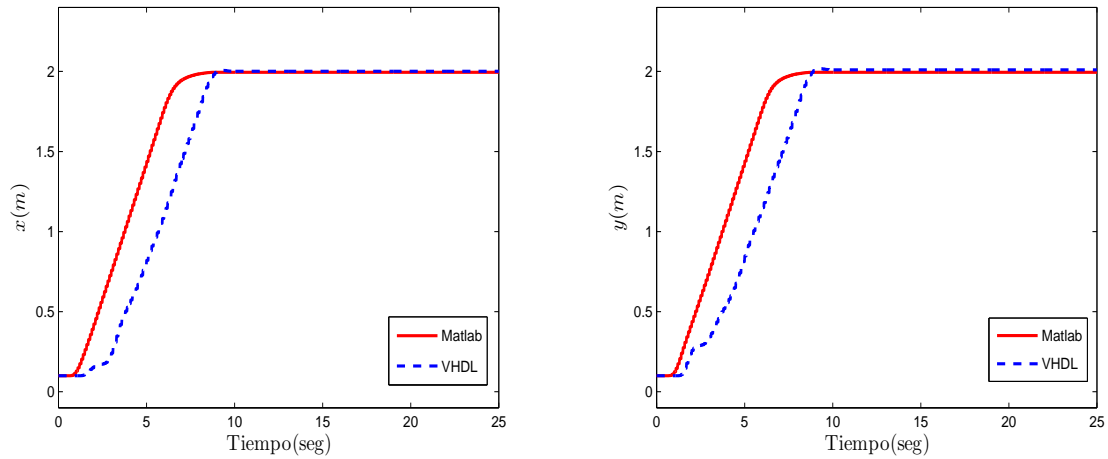


Figura 4.33: Experimento 6. Error de posición y orientación del robot móvil de ruedas.

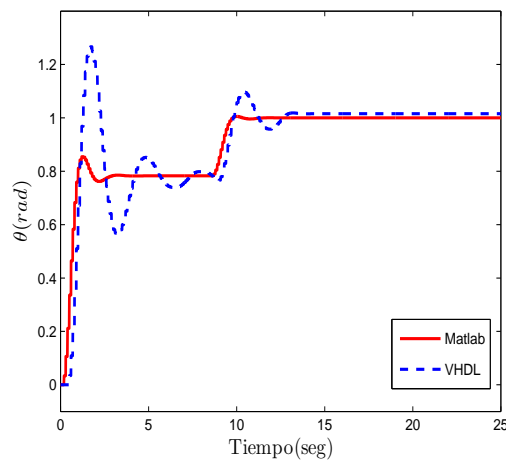
Experimento 7.- El robot móvil de ruedas debe alcanzar la posición y orientación deseada $q_d = (2, 2, 1)^T$. Se considera el sistema difuso de control de posición descrito en la sección 4.1.2 y el de control de orientación descrito en la sección 4.2.2 para resolver el problema de regulación de postura del robot móvil de ruedas, los cuales fueron desarrollados en VHDL y Matlab-Simulink para realizar comparaciones entre ambos. En la Figura 4.34 se muestra la evolución en el tiempo de la posición y orientación del robot móvil de

ruedas utilizando los sistemas difusos desarrollados en VHDL y Matlab-Simulink.



(a) Posición final en x .

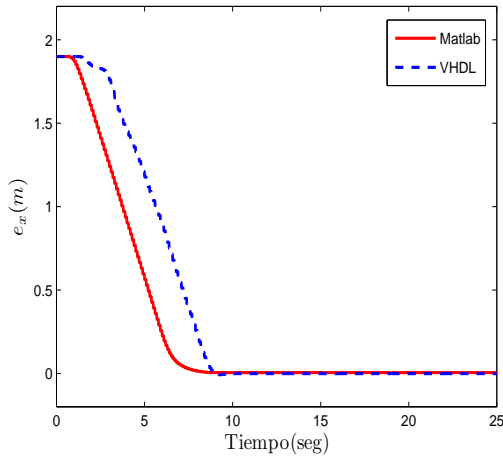
(b) Posición final en y



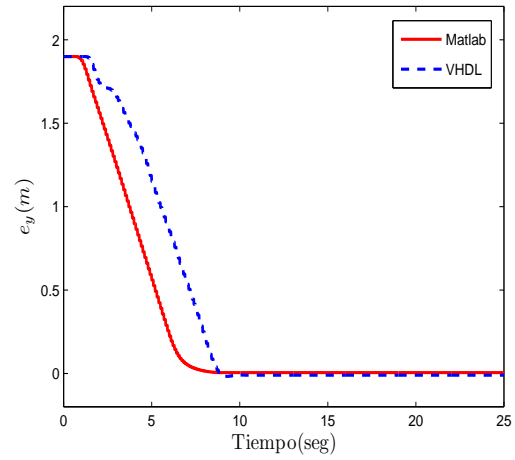
(c) Orientación final en θ

Figura 4.34: Experimento 7. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.

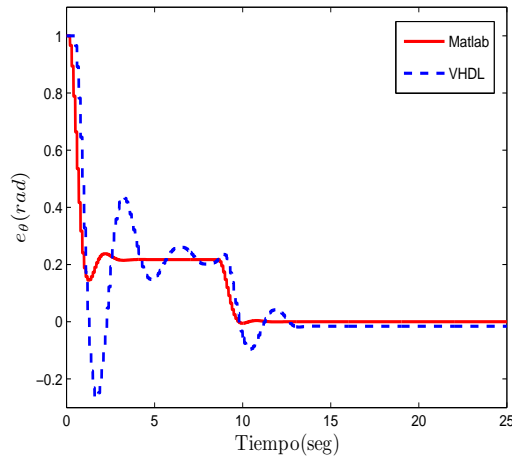
En la Figura 4.35 se muestra el error de posición en y y y , y de orientación en θ al realizar la simulación del sistema del control de movimiento del robot móvil de ruedas en el experimento 7.



(a) Error de posición en x .



(b) Error de posición en y

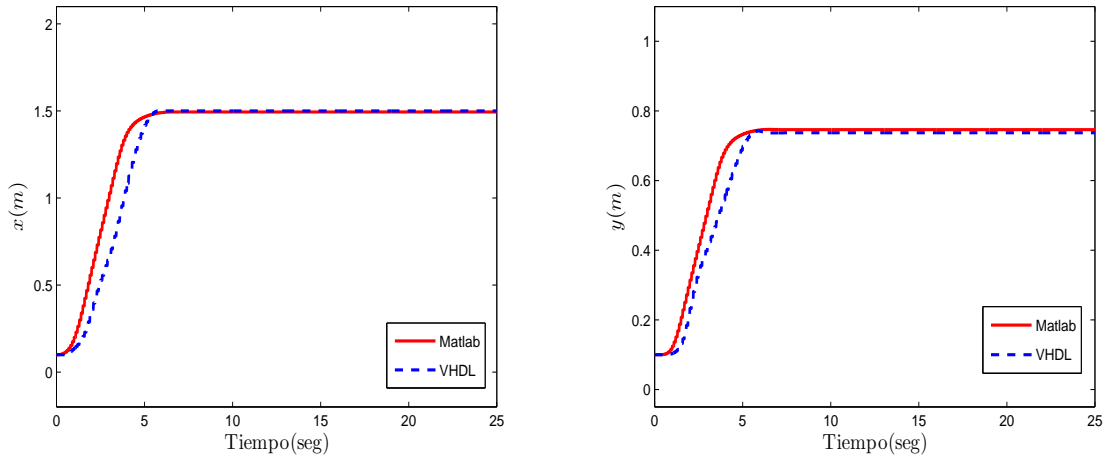


(c) Error de orientación en θ

Figura 4.35: Experimento 7. Error de posición y orientación del robot móvil de ruedas.

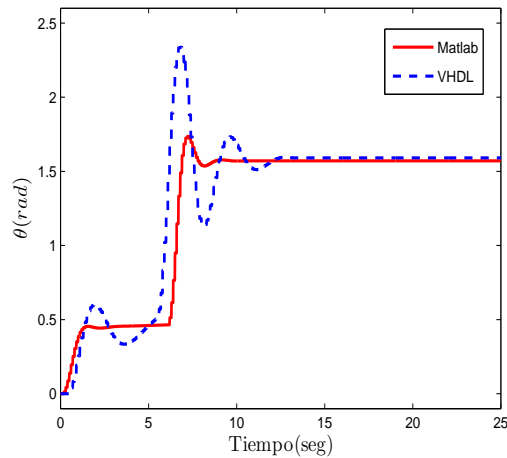
Experimento 8.- El robot móvil de ruedas debe alcanzar la posición y orientación deseada $q_d = (1.5, 0.75, \frac{\pi}{4})^T$. Al igual que en los experimentos anteriores se sigue considerando el sistema difuso de control de posición descrito en la sección 4.1.2 y el de control de orientación descrito en la sección 4.2.2 para resolver el problema de regulación de postura del robot móvil de ruedas, los cuales fueron desarrollados en VHDL y Matlab-Simulink para realizar comparaciones entre ambos. En la Figura 4.36 se muestra

la evolución en el tiempo de la posición y orientación del robot móvil de ruedas utilizando los sistemas difusos desarrollados en VHDL y Matlab-Simulink.



(a) Posición final en x .

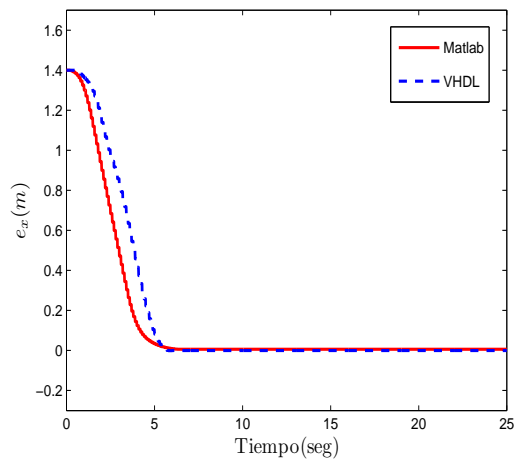
(b) Posición final en y



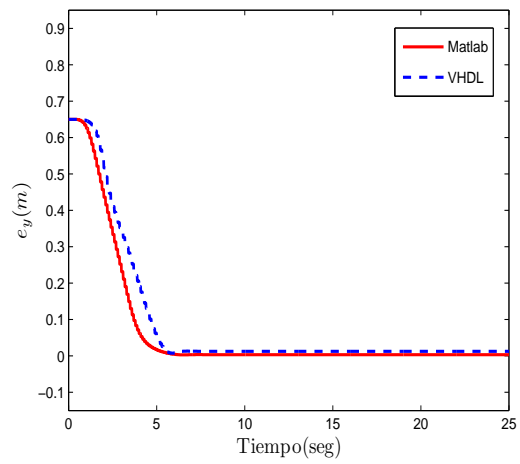
(c) Orientación final en θ

Figura 4.36: Experimento 8. Evolución en el tiempo de la posición y orientación del robot móvil de ruedas.

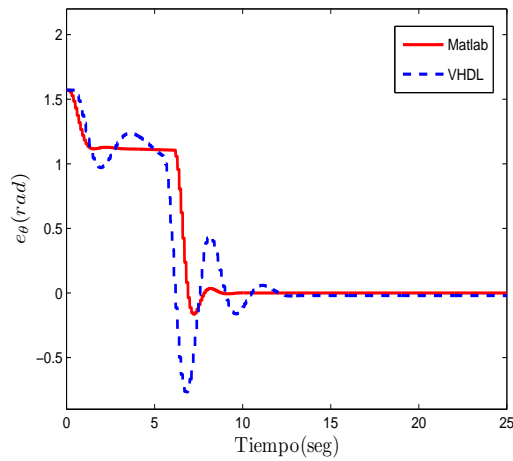
En la Figura 4.37 se observan los errores de posición en x y y , y el error de orientación θ obtenidos al realizar la simulación del sistema de control del robot móvil de ruedas en el experimento 8.



(a) Error de posición en x .



(b) Error de posición en y



(c) Error de orientación en θ

Figura 4.37: Experimento 8. Error de posición y orientación de robot móvil de ruedas.

Capítulo 5

Conclusiones

5.1. Conclusiones

En este trabajo de tesis se desarrolló un sistema difuso para resolver el problema de regulación de postura de un robot móvil con ruedas. Para resolver este problema fue necesario dividirlo en dos subproblemas debido a que se requería proponer un sistema difuso de tres entradas lo cual no es muy conveniente debido a que ocasiona que el número de reglas crezca dificultando su sintonización, por ello se propuso un sistema difuso que controla la posición y otro la orientación del robot móvil de ruedas para satisfacer el objetivo de control propuesto.

La segunda parte de este trabajo, consistió en desarrollar la codificación de los sistemas difusos en VHDL. Ya codificados se realizaron una serie de pruebas, las cuales consistieron en obtener las superficies de control de cada uno de los sistemas para ser comparadas con las que se obtuvieron al realizar las simulaciones de los sistemas desarrollados en Matlab. Al observar los resultados se refleja claramente que el error que se obtuvo al realizar las comparaciones entre las superficies de control es aceptable. De manera que si se requiere disminuir los errores de superficies obtenidos, solamente se tiene que aumentar el número de bits con los van a trabajar las entradas y salidas de los sistemas difusos.

Por último, se desarrolló una serie de simulaciones computacionales del sistema

de control de movimiento del robot móvil con ruedas, integrando a este los sistemas difusos de control de posición y orientación desarrollados en VHDL y Matlab propuestos en este trabajo de tesis. Al analizar los resultados que se obtuvieron al realizar cada uno de los experimentos, se observa claramente que el objetivo de control propuesto se satisface con un error de posición y orientación del robot móvil con ruedas mínimo. Comparando el desempeño de los sistemas difusos desarrollados en Matlab y VHDL, se observa que el robot móvil con ruedas tarda más tiempo en alcanzar su posición y orientación deseada cuando está trabajando con los sistemas difusos codificados en VHDL.

5.2. Trabajos Futuros

- Implementar el sistema de control de movimiento del robot móvil de ruedas en un FPGA. Para desarrollar la implementación es necesario diseñar controladores digitales para el control de velocidad de las ruedas del robot móvil e integrarlos con los sistemas difusos desarrollados para el control de posición y orientación.
- Integrar el sistema difuso de posición y orientación del robot móvil de ruedas en un mismo sistema con la ayuda de algún método como redes neuronales, ANFIS o algoritmos genéticos que nos permita realizar la sintonización de las reglas más fácilmente.
- Experimentar con los objetivos de control de seguimiento de trayectorias y seguimiento de ruta, para validar que los sistemas difusos desarrollados en este trabajo pueden dar solución a dichos problemas y no solo se limita al control de regulación de postura.

Referencias

- [1] R. Siegwart y I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. The MIT Press, 2004.
- [2] A. O. Baturone, *Robótica: Manipuladores y Robots Móviles*. MARCOMBO, 2007.
- [3] R. Silva, M. A. Molina, V. M. Hernández, y G. Silva, “Control de un robot móvil de ruedas mediante linealización de entrada-salida,” *III Congreso Internacional Tendencias Tecnológicas en Computación*, 2007.
- [4] R. Fierro y F. L. Lewis, “Control of a nonholonomic mobile robot using neural networks,” *IEEE Transactions on Neural Networks*, vol. 9, no. 4, Julio 1998.
- [5] O. Castillo, L. T. A. Aguilar, y S. Cárdenas, “Fuzzy logic tracking control for unicycle mobile robots,” *Engineering Letters (ENGL)*, vol. 13, no. 2, pp. 73–77, 2006.
- [6] S. Cárdenas, “Controlador de seguimiento para un robot móvil empleando lógica difusa,” Tesis de Maestría, Instituto Tecnológico de Tijuana, 2005.
- [7] L. Astudillo, O. Castillo, y L. T. Aguilar, “Intelligent control for a perturbed autonomous wheeled mobile robot: a type-2 fuzzy logic approach,” *Journal of Nonlinear Studies*, vol. 14, no. 1, pp. 37–48, Marzo 2007.
- [8] R. Martínez, O. Castillo., y L. T. Aguilar., “Intelligent control for a perturbed autonomous wheeled mobile robot using type-2 fuzzy logic and genetic algorithms,”

- Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 2, no. 1, pp. 12–22, Enero 2008.
- [9] L. Astudillo, O. Castillo, P. Melin, A. Alanis, J. Soria, y L. T. Aguilar, “Intelligent control of an autonomous mobile robot using tipe-2 fuzzy logic,” *Engineering Letters (ENGL)*, vol. 13, no. 2, pp. 93–97, 2006.
- [10] B. Lacevic, J. Velagic, y N. Osmic, “Desing of fuzzy logic based mobile robot position controller using genetic algorithm,” *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1–6, 2007.
- [11] J. Velagic, N. Osmic, y B. Lacevic, “Neural network controller for mobile robot motion control,” *World Academy of Science, Engineering and Technology*, vol. 47, pp. 193–198, 2008.
- [12] L.-C. Hung y H.-Y. Chung, “Design of hierarchical fuzzy logic control for mobile robot system,” *IEEE Conference on Robotics, Automation and Mechatronics*, pp. 1–6, 2006.
- [13] N. Ouadh, L. Ourak, M. Hamerlain, y F. Boudjema, “Implementation of an oriented positioning on a car-like mobile robot by fuzzy control,” *IEEE 32nd Annual Conference of IEEE on Industrial Electronics*, pp. 4076–4081, 2006.
- [14] S. Islam, A. Azim, S. Jahan, y M. Othman, “Desing and sinthesis of mobile robot controller using fuzzy,” *IEEE International Conference on Semiconductor Electronics*, pp. 825–829, 2006.
- [15] S. Islam, M. Zaman, B. Madon, y M. Othman, “Designing fuzzy based mobile robot controller using VHDL,” *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 2, pp. 138–142, 2008.

-
- [16] C. J. Jiménez, S. S. Solano, y Barriga, “Hardware implementation of general purpose fuzzy controller,” *Sixth International Fuzzy Systems Fuzzy Systems Association World Congress (IFSA '95)*, vol. 2, pp. 185–188, 1995.
- [17] D. Kim, “An implementation of fuzzy logic controller on the reconfigurable FPGA system,” *IEEE Transactions on Industrial Electronics*, vol. 47, pp. 703–715, Junio 2000.
- [18] S. S. Solano, A. Barriga, C. J. Jiménez, y J. L. Huertas, “Design and application of digital fuzzy controllers,” *Sixth IEEE International Conference on Fuzzy Systems (Fuzz-IEEE'97)*, vol. 2, pp. 869–874, 1997.
- [19] O. Montiel, Y. Maldonado, R. Sepúlveda, y O. Castillo, “Simple tuned fuzzy controller embedded into an FPGA,” *NAFIPS Conference Proceedings*, pp. 1–6, 2008.
- [20] O. Montiel, R. Sepúlveda, J. Olivas, Y. Maldonado, O. Castillo, y P. Melin, “Hardware implementation of a fuzzy controller,” en *International Journal of Factory Automation, Robotics and Soft Computing*. International society for Advanced Research, 2008, no. 4, pp. 5–10.
- [21] O. Montiel, J. Olivas, R. Sepúlveda, y O. Castillo, “Development of an embedded simple tuned fuzzy controller,” *IEEE International Conference on Fuzzy Systems*, pp. 555–561, 2008.
- [22] S. Singh y K. S. Rattan, “Implementation of a fuzzy logic controller on an FPGA for a dc motor,” *Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Technology Conference*, pp. 387–394, 2003.
- [23] J. J.-S. Roger, C. T. Sun, y E. Mizutani, *Neuro - Fuzzy and Soft Computing*. Prentice Hall, 1997.
- [24] R. Sepúlveda, O. Montiel, OscarCastillo, y P. Melin, *Fundamentos de Lógica Difusa*. Ediciones ILCSA, 2002.

-
- [25] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Information Science*, 1975.
- [26] P. P. Chu, *FPGA Prototyping by VHDL Examples*. John Wiley & Sons, Inc., 2008.
- [27] F. Pardo y J. A. Boluda, *VHDL. Lenguaje para Síntesis y Modelado de Circuitos*. RA-MA, 1999.
- [28] F. J. T. Valle, "Síntesis y descripción de circuitos digitales utilizando VHDL," Universidad Autónoma de Guadalajara, 2001.
- [29] M. Melgarejo, *Desarrollo de un Sistema de Inferencia Difusa sobre FPGA*. Universidad Distrital Francisco José de Caldas, 2003.
- [30] T. S., *Field-Programmable Gate Arrays Technology*. Boston: Kluwer Academic Publishers, 1995.
- [31] Pittman DC Servo Motor, hoja de especificaciones disponible en: <http://www.clickautomation.com/PDF/items/GM9236S025.pdf>.
- [32] National Semiconductor, hoja de especificaciones disponible en: <http://www.national.com/ds/LM/LMD18200.pdf>.
- [33] Y. Maldonado, O. Montiel, R. Sepúlveda, y O. Castillo, "Desing and simulation of the fuzzification stage through the xilinx system generator," en *Soft Computing for Hybrid Intelligent Systems*. Springer, 2008, pp. 297–305.
- [34] J. A. Olivas, R. Sepúlveda, O. Montiel, y O. Castillo, "Methodology to test and validate a VHDL inference engine through the xilinx system generator," en *Soft Computing for Hybrid Intelligent Systems*. Springer, 2008, pp. 325–331.
- [35] G. Lizárraga, R. Sepúlveda, O. Montiel, y O. Castillo, "Modeling and simulation of the defuzzification stage using xilinx system generator and simulink," en *Soft Computing for Hybrid Intelligent Systems*. Springer, 2008, pp. 333–343.

-
- [36] E. Aranda, T. Salgado, y M. Velasco, "Control no lineal discontinuo de un robot móvil," *Computación y Sistemas*, pp. 42–49, 2002.
- [37] J. Jantzen, *Foundations of Fuzzy Control*. John Wiley & Sons Ltd., 2007.
- [38] Y. Maldonado, "Diseño, implementación y prueba de la etapa de fuzzificación para sistemas difusos tipo 1 y tipo 2 en FPGA's," Tesis de Maestría, Centro de Investigación y Desarrollo de Tecnología Digital del Instituto Politécnico Nacional, 2008.
- [39] J. A. Olivas, "Diseño, implementación y prueba de maquinas de inferencia difusas tipo 1 y tipo 2 en FPGA's," Tesis de Maestría, Centro de Investigación y Desarrollo de Tecnología Digital del Instituto Politécnico Nacional, 2008.
- [40] I. N. da Silva, F. A. C. Gomide, y W. C. do Amaral, "Navigation of mobile robots using fuzzy logic controllers," *5th International Workshop on Advanced Motion Control*, pp. 346–349, 1998.
- [41] O. Soto, "Identificación de sistemas mediante cómputo evolutivo," Tesis de Maestría, Centro de Investigación y Desarrollo de Tecnología Digital del Instituto Politécnico Nacional, 2009.