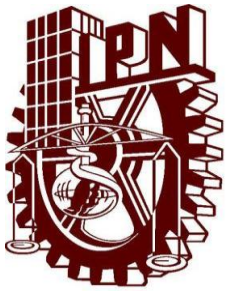


INSTITUTO POLITÉCNICO NACIONAL

Centro de Investigación en Computación
Laboratorio de Reconocimiento de Patrones

Localización y manipulación de objetos mediante un robot móvil Khepera II auxiliado de una cámara y un elemento prensil

T E S I S

Que para obtener el grado de

Maestro en Ciencias en Ingeniería de Computo

P R E S E N T A:

Gabriel Omar Huerta Moreno

Directores de tesis:

Dr. Juan Humberto Sossa Azuela

Dra. Elsa Rubio Espino

México D.F., Diciembre de 2009

1030

SIP-14

INSTITUTO POLITECNICO NACIONAL

SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 12:00 horas del día 26 del mes de Noviembre de 2009 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"LOCALIZACIÓN Y MANIPULACIÓN DE OBJETOS MEDIANTE UN ROBOT MÓVIL KHEPERA II AUXILIADO DE UNA CÁMARA Y UN ELEMENTO PRENSIL"

HUERTA	MORENO	GABRIEL OMAR
(Apellido Paterno)	(Apellido Materno)	(Nombre)

Con registro: B 0 7 1 2 8 0

aspirante al grado de: ***MAESTRÍA EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN 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 COMISIÓN REVISORA

Presidente

Dr. Sergio Suárez Guerra

Secretario

Dr. José Luis Oropeza Rodríguez

Primer vocal
(Director)

Dr. Juan Humberto Sossa Azuela

Segundo vocal

Dr. Domingo de Jesús Cortés Rodríguez

Tercer vocal

Dr. Ricardo Barrón Fernández

EL PRESIDENTE DEL COLEGIO

Dr. Jaime Álvarez Gallegos

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION
DIRECCION

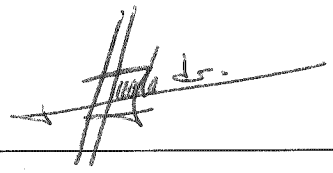
INSTITUTO POLITECNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESION DE DERECHOS

En la Ciudad de **México, D.F.** el día **07** del mes **DICIEMBRE** del año **2009**, el que suscribe **ING. GABRIEL OMAR HUERTA MORENO** alumno del Programa de **MAESTRÍA EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES** con número de registro **B071280**, adscrito al **CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN DEL INSTITUTO POLITÉCNICO NACIONAL**, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección del **DR. JUAN HUMBERTO SOSSA AZUELA** y la **DRA. ELSA RUBIO ESPINO,** cede los derechos del trabajo titulado ***"LOCALIZACIÓN Y MANIPULACIÓN DE OBJETOS MEDIANTE UN ROBOT MÓVIL KHEPERA II AUXILIADO DE UNA CÁMARA Y UN ELEMENTO PRENSIL "***, 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 **omar.bit@gmail.com.**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



GABRIEL OMAR HUERTA MORENO

Resumen

En esta tesis se presenta a detalle una metodología dirigida hacia el control difuso aplicada a robots móviles de manejo diferencial, mediante la constante retroalimentación de información del entorno a través de una cámara instalada sobre el móvil. El objetivo general de esta tesis, es que el robot sea capaz de alcanzar puntos de referencia median navegación, así como manipulación de objetos cuyo centro de masa es el punto de referencia perseguido. El sistema desarrollado localiza su objetivo mediante segmentación de imágenes de configuración RGB, y discriminación por forma a través de una clasificación hecha por un modelo red neuronal perceptrón multicapa previamente entrenado. Se analiza y describe una modificación al modelo de manejo diferencial de un robot móvil, para que sea útil bajo las condiciones de visión que se tiene, para lo cual se implementa y desarrolla una interfaz grafica que permite evaluar el desempeño del sistema de visión y control, ante la realización de una tarea que comprende la búsqueda, aproximación, toma y manipulación de un objeto.

Abstract

In this thesis a methodology directed towards the diffuse control applied robots movable of handling differential appears to detail, by means of the constant feedback of information of the surroundings through a camera installed on the moving body. The general mission of this thesis, is that the robot is able to reach datum points mediate navigation, as well as manipulation of objects whose center of mass is the persecuted datum point. The developed system locates its objective by means of segmentation of images of configuration RGB, and discrimination by form through a classification done by a model neuronal network multi-layer perceptrón previously trained. It analyzes and describes a modification to model of handling differential of robot movable, so that he is useful under the conditions of vision that is had, for which one implements and it develops a graphic interface that allows to evaluate the performance of the system of vision and control, before the accomplishment of a task that includes/understands the search, approach, takes and manipulation from an object.

Agradecimientos

Deseo expresar mi más profundo agradecimiento a mis asesores Dr. Juan Humberto Sossa Azuela y Dra. Elsa Rubio Espino por su apoyo durante todo este tiempo, atención y paciencia recibida.

A mis sinodales por sus valiosas observaciones, Dr. Sergio Suárez Guerra, Dr. José Luis Oropeza Rodríguez, Dr. Ricardo Barrón Fernández, Dr. Domingo de Jesús Cortés Rodríguez, por el tiempo dedicado a la revisión de este manuscrito; y a todas aquellas personas que directa o indirectamente han intervenido en el desarrollo de esta tesis.

Al IPN y al CIC por la oportunidad de continuar con mi formación académica y a la Universidad Autónoma del Estado de México mi alma mater.

A todos los compañeros CIC-IPN por sus aportaciones y desinteresada cooperación: Julio Fernando Jiménez Vielma, Laura Elena Gómez Sánchez, Cecilia Albortante Morato, Juan Villegas Cortes, Jaime Alfonso Pérez León, Liliana Sánchez Romero, Julio León Márquez, José Adriel, Miguel Ángel Zurita, Gastón Salazar, Christian Castro Morales, Guadalupe Sáenz.

A mis amigos de siempre: Carmen Hernández Carbajal, Gabriela Romina Hernández Carbajal, Cristina Gonzales Pérez, Carolina Gonzales Pérez.

A mis padres y hermanos por haberme apoyado en el transcurso de mi vida.

Este documento se ha realizado con la ayuda financiera de la Comunidad Europea, la Unión Europea, la Comisión Europea y el CONACYT bajo el proyecto FONCICYT 93829.

El contenido de este documento es responsabilidad exclusiva del Instituto Politécnico Nacional y en modo alguno debe considerarse refleja la posición de la Unión Europea.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

La fuerte competencia por el desarrollo de robots cuyos propósitos son definidos por sus movimientos, ha derivado en múltiples líneas de investigación, sobre las cuales la principal motivación de los investigadores es la posibilidad de que los robots puedan llegar a mostrar un comportamiento inteligente que imité al de los humanos o animales. Para tal fin la IA (Inteligencia Artificial), ha jugado un papel importante, ya que esta disciplina informática se ha dedicado al desarrollo de agentes racionales no vivos, los cuales basan su racionalidad principalmente en técnicas de entrenamiento y aprendizaje que requieren de sistemas de validación. La ejecución de tareas tales como, moverse, hacer funcionar un brazo mecánico, tocar y manipular su entorno, son posibles gracias al desarrollo de técnicas de control, sin embargo la eficiencia en la ejecución de estas tareas implica un ambiente controlado libre de factores que puedan alterar de cualquier forma la dinámica del ambiente, soluciones tecnológicas como el empleo de complejos sistemas sensoriales dan respuestas eficientes para dar solución a problemas de control. Lo anterior hace evidente que se ha dependido del avance tecnológico, el cual facilita el trabajo de los investigadores ante el diseño de algoritmos o programación, sin embargo la tecnología demanda altos costos, por lo que muchos trabajos no son probados en situaciones experimentales reales.

Los intentos de crearlos tienen una larga historia, las máquinas totalmente autónomas aparecieron hasta el siglo XX, desde entonces se busca que un dispositivo robótico sea capaz de operar por sí mismo, la robótica móvil ha destacado por dar mayor aporte a la autonomía de

las máquinas, ya que se enfrenta a problemas de navegación ante entornos dinámicos, se han propuesto estrategias y desarrollado técnicas de control y navegación con resultados cada vez más exitosos ante tareas específicas. Sistemas de robots autónomos dotados con capacidad de visión han atraído la atención de los investigadores, estos sistemas móviles con visión, prometen un amplio campo de aplicación en actividades comunes con humanos. La visión por computadora hace a los sistemas robóticos en general, adquirir robustez ante entornos dinámicos, ya que representa la capacidad visual sensorial para abstraer características que describen a su entorno.

Hoy en día la ciencia persigue llevar a la práctica los nuevos modelos de control que son diseñados para actividades de navegación y manipulación, de forma segura y accesible en cuestión de costos, mediante el empleo de robots miniatura a escala, de esta forma es posible manipular y proponer escenarios bajo condiciones controladas, cada vez menos estructuradas.

Khepera es un robot móvil de configuración diferencial, el cual consta de una plataforma cilíndrica sobre la que se alojan sistemas de sensores, sistemas de comunicación que garantizan comunicación con ordenadores y al cual pueden agregarse módulos de visión, y elementos de manipulación, su mecanismo de movimiento consta de dos ruedas, que giran sobre el mismo eje. Se han reportado muchos trabajos con este robot que intentan resolver problemas de navegación y control, mediante visión por computadora con resultados cada vez mejores, sin embargo, aún quedan puntos débiles por reforzar, como el caso en que la perspectiva de visión no incluye la exploración total de su entorno.

Con el fin de proponer una metodología útil para sistemas de navegación en robótica móvil con visión paralela al plano en que se mueve, se abordarán temas relacionados con control basado en lógica difusa, aplicados en un robot khepera II dotado de una cámara y un elemento prensil.

1.1. Planteamiento del problema

Es claro que para dar solución a problemas de navegación y control, específicamente en robótica móvil, son insuficientes los aspectos tecnológicos, y aunque se han resuelto tareas de

alto grado de dificultad y con alto grado de eficiencia, ha resultado muy costoso, y sólo se han resuelto tareas específicas del todo controladas. El control de éstas estructuras robotizadas, en espacios abiertos y en distancias no definidas, junto con la limitante de ambientes antes mencionados, ha dado pie a la implementación de estrategias y propuestas como las que se enlistan a continuación:

- Arquitecturas que describen e interpretan su entorno a través de información obtenida mediante sensores de proximidad;
- Cálculo de trayectorias y evasión de obstáculos mediante visión por computadora, algunos con perspectivas completas del área de trabajo, que dan una descripción completa y detallada de su entorno y otras con esa perspectiva de visión paralela a la superficie en que se mueven, éstas con menor eficiencia en sus resultados ;
- Metodologías que combinan técnicas como algoritmos genéticos o redes neuronales que resuelven problemas de navegación mediante sistemas clasificadores;
- Métodos de control clásico y difusos que se aplican a la navegación y manipulación automática de objetos de forma inteligente, que se basan en el uso y modificación del conocimiento, con lo que se busca imitar el uso de experiencia como lo hacen los seres vivos, a lo que se ha llamado aprendizaje, el cual sigue siendo un reto para mejorar el rendimiento de los sistemas.

Lo anterior hace evidente que los robots tienden a ser cada vez más inteligentes, flexibles y fáciles de usar, sin embargo estas técnicas avanzadas de control, aprendizaje y validación no han logrado reducir en gran medida los altos costos, por el contrario son adquiridos nuevos cotos en términos computacionales, los algoritmos de control y guiado de robots móviles, demanda complejos sistemas de programación, que consumen recursos como tiempo de ejecución y gran cantidad de memoria, estos factores afectan el rendimiento de los métodos de navegación por el tiempo que pueden tardar en responder. Queda claro que aún está distante el tiempo en que los robots puedan comportarse de manera similar a los seres vivos, técnicas de control inteligente basadas en lógica difusa, son implementadas para evolucionar los sistemas

robotizados, ya que ofrecen una forma flexible de medir y evaluar los parámetros de entrada para decidir sobre las salidas del sistema y dan la impresión de que actúan con cualidades de adaptación ante la incertidumbre del comportamiento de su entorno.

Uno de los retos que imperan en robótica sigue siendo la búsqueda de la autonomía de los robots móviles, términos más ambiciosos como el de tiempo real o parcialmente real, se integran también al mundo de la robótica móvil, al respecto se emplean robots de tipo diferencial dotados con dispositivos de visión, como el robot khepera II cuya breve descripción se hace en la introducción de este capítulo, para el que se buscan nuevas metodologías que ofrezcan más robustez ante condiciones de imprecisión e incertidumbre, mediante el uso de técnicas de control difuso, para la navegación y manipulación de objetos.

1.2. Justificación

Puesto que el mundo real representa un espacio dinámico e impredecible, al que cualquier robot que no cuente con la capacidad de procesar información incompleta, aproximada o difusa, será poco eficiente en la búsqueda de su objetivo, más aun si su controlador es clásico. La respuesta del controlador para un robot que se enfrenta a estos problemas debe ser lo suficientemente rápido, para actuar ante sucesos inesperados en la medida que van sucediendo. Es lo que nos motiva a proponer una metodología basada en técnicas de control difuso para el alcance de metas, apuntado hacia objetos de interés estáticos para definir la orientación del avance, mediante el censado del objetivo con visión, ya que hoy en día la visión parece ser uno de los medios más efectivos ante entornos complejos y aproximados.

Sin olvidar que los robots desde su idealización y puesta en práctica están diseñados para el servicio del hombre y de sustituirlo en tareas comunes y peligrosas tales como:

- La desactivación de bombas
- Realización de traslados de cargas pesadas que representan trabajo rudo
- Exploración de espacios hostiles para el hombre

- Sistemas de seguridad y transporte
- Expediciones espaciales, marinas o desconocidas

Por ahora los resultados de este trabajo no serán llevados a la práctica con robots que resuelvan las tareas antes mencionadas o de fin específico práctico. El esquema que ahora se plantea será implementado en un robot miniatura de tipo diferencial Khepera II, realizando tareas con fines de experimentación que reflejen robustez en aspectos de control y manipulación de forma eficiente, segura y menor costo computacional, reduciendo el uso del conjunto de sensores al sistema de visión como único sensor.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar una metodología basada en control difuso que permita a un robot móvil Khepera II, dotado de una cámara con ángulo de visión paralelo a la superficie en que se desplaza y un elemento prensil, navegar en un entorno poblado de objetos estáticos, para buscar a uno en particular, tomarlo y realizar una tarea específica con él.

1.3.2. Objetivos particulares

- Diseñar una interfaz de software, que permita validar la capacidad visual del robot en tiempo real ante la localización del objetivo para la implementación del controlador.
- Evaluar el rendimiento de los algoritmos de procesamiento y etiquetado de imágenes en tiempo real.
- Proponer un algoritmo para dar solución a las consignas de control difuso con la única entrada de visión con la que se cuenta.
- Aplicar el controlador y el modulo de visión para manipular un objeto, que demuestre la utilidad y eficiencia del sistema de control.

1.4. Alcances

En este trabajo de investigación se pretende desarrollar un modelo de control difuso acompañado de otras técnicas propias de la IA (Inteligencia Artificial), cuya principal base de conocimientos sea la visión por computadora, con el fin de crear un método o mecanismo inteligente capaz de dotar a una máquina con la capacidad de realizar tareas tal como lo harían los seres humanos o animales, con información imprecisa o incompleta. Para el cumplimiento de los objetivos de esta investigación se desarrolla un prototipo de interfaz de control con propiedades reconfigurables, sobre el posible cambio de objetivo o control manual del móvil a consideración del usuario, dotado además de un apartado que refleje el rendimiento de los módulos que componen al sistema de control y navegación mediante campos de información que proporcionen datos sobre mediciones de precisión y error a cada instante hasta terminar una tarea programada.

1.5. Contribuciones

Una metodología basada en control difuso para el control de robots diferenciales a partir de información puramente visual, capaz de realizar exitosamente tareas simples de alcance de objetivos estáticos y manipulación de objetos, con cierta adaptabilidad a persecución de objetos móviles. El desarrollo de una herramienta de software bajo la tecnología de programación java, que permite la evaluación del desempeño de navegación del móvil mediante la evaluación visual del experto, que permite su reutilización para trabajos posteriores, cuya plataforma programada no se encuentra atada, al sistema operativo Windows, aprovechando las ventajas de la máquina virtual de java.

1.6. Organización del trabajo

Esta tesis se compone de seis capítulos, el primero de ellos da una breve introducción que abarca generalidades de la robótica, en particular temas dirigidos al control de robots

móviles, de tal manera que la justificación de este trabajo hace especialmente énfasis en atacar problemas relacionados al respecto, así como la fijación de objetivos claros y posibles de alcanzar. El segundo capítulo profundiza en una colección de trabajos previos, que servirán como base para no recorrer camino ya andado y que fundamente la metodología propuesta como innovadora. En el tercer capítulo se documenta con material teórico relacionado con las áreas o técnicas de estudio que sirvieron de base para la propuesta de algoritmos y métodos experimentales, material que fue debidamente referenciado para hacer evidente su autenticidad. En el capítulo cuatro, se expone la metodología propuesta y el desarrollo del proyecto de tesis en sus diferentes etapas y módulos, así como una descripción de tallada de los algoritmos propuestos para cumplir con las sentencias de control difuso. El capítulo 5 corresponde a la fase experimental, realizando una seria de tareas sencillas que prueban la eficiencia del sistema de control difuso. Y por último el capítulo seis muestra una reflexión sobre las carencias o puntos que pudieron cubrirse de forma parcial o total y en base a ello hace una seria de recomendaciones dirigidas hacia trabajos futuros relacionados con el presente trabajo de tesis.

Capítulo 2

Estado del Arte

2.1. Introducción

En este capítulo se describen algunos de los trabajos que plantean esquemas de control clásico y difuso para robot móviles, preferentemente esquemas que dependen de la percepción de su entorno mediante sensores de proximidad o visión por computadora, para la ejecución de tareas dirigidas a la búsqueda, alcance de metas, acciones de reacción, planificación, etc. Es decir líneas de investigación de la robótica móvil que fundamentan la relevancia de esta investigación.

2.2. Una arquitectura distribuida para el control de robots autónomos móviles (2001).

Esta tesis doctoral [Martínez 02], el autor propone e implementa una arquitectura distribuida para robots móviles, el objetivo principal es que un robot bajo esta arquitectura, pueda realizar tareas de forma autónoma en un entorno desconocido y con incertidumbre, en un entorno web que permite la tele operación del robot móvil en tareas de exploración. Se hace uso de redes neuronales, algoritmos genéticos, y lógica difusa para realizar tareas tales como, aprendizaje, interpretación del entorno, realización de mapas, planificación de

trayectorias y localización. El resultado fue el robot Quaky-Ant, una plataforma móvil de conducción diferencial con un soporte de una rueda loca, también está dotado con un conjunto de sensores de proximidad infrarrojos y de ultrasonido, así como de una cama RGB para retroalimentación visual. Algo que separa a este robot del común de las plataformas móviles comerciales, es que está dotado de dos dispositivos muy útiles para fines de orientación como son:

- Un compas electrónico que mide la orientación del robot con respecto al polo magnético de la tierra para compensar errores de odometría en interiores.

- Un receptor GPS para posicionamiento absoluto, que se utiliza para compensar los errores de odometría en entornos exteriores.

Figura 2.1: Robot Quaky-Ant.

Al revisar este trabajo doctoral, en particular se presta especial atención en el uso de un conjunto de redes neuronales artificiales tipo backpropagation y un sistema de reglas difusas. No se realiza el entrenamiento de una única red, ni se usa un único sistema de reglas difusas, ya que el uso de algoritmos genéticos permite elegir a la mejor red y al mejor conjunto de reglas. En nuestro trabajo no haremos una selección de entre varias redes, ni tampoco de entre varios sistemas de reglas difusas, solo nos interesa la capacidad de generalización de la red backpropagation al comparar patrones.

2.3. Aprendizaje evolutivo de reglas fuzzy en un sistema clasificador modificado para control de agentes móviles (2004).

Esta tesis doctoral [Vallejo 01], es una extensión del algoritmo ELF(Evolucionary Learning of Fuzzy Rules) desarrollado por Bonarini en 1994, está adaptado para aprender controladores fuzzy en una plataforma móvil, como lo es el robot Khepera. Es probablemente uno de los trabajos más complejos, el autor propone una metodología basada en algoritmos genéticos, su estrategia de aprendizaje evolutivo de reglas difusas para la navegación, se hace dinámicamente al relacionar el antecedente con el consecuente de las reglas. Durante su entrenamiento generan un conjunto de reglas difusas dinámicas bajo la supervivencia del más apto, el número de reglas puede variar con el tiempo al ser probadas en el mismo entorno que realizó su entrenamiento o en otro incluso si es dinámico. El autor enfatiza que se conocen las curvas de los sensores que describen la posición o distancia de obstáculos y objetivos, y ya que la lectura de estos sensores puede ser ambigua o ruidosa es que se recurrió a la lógica difusa. El empleo de cuatro conjuntos difusos que atienden la distancia y posición con respecto a los obstáculos y objetivos, es tal vez lo más significativo que se revisara de este trabajo.

Distancia al obstáculo	Dirección del obstáculo	Dirección del objetivo	Dirección del robot
C	F	F	D

En este trabajo las reglas difusas pueden ser diferentes, cambiar e incluso desaparecer de acuerdo a su desempeño, pero la estructura nunca cambiará, de acuerdo al autor cada regla podría tener la forma:

Si (la distancia al objeto es <a>) y (la dirección al obstáculo es <b>) y (la dirección del objetivo es <c>) entonces (la dirección del robot es <d>).

2.4. Control visual de un robot móvil khepera II (2007).

En esta tesis de maestría [Petrilli 03], se presenta una metodología para el control de robots diferenciales para interceptar objetos en movimiento, la metodología es probada en un robot khepera II que intercepta una pelota en un espacio de color homogéneo y bajo condiciones de iluminación también homogénea. La acción del controlador depende de la retroalimentación visual del espacio de trabajo del robot, el autor presenta cinco etapas:

- Etapa de adquisición
- Etapa de detección
- Etapa de cálculo de trayectorias
- Etapa de cálculo de la intersección
- Etapa de cálculo de la velocidad

El método de cálculo de velocidad de cada uno de los motores, es lo que atrajo nuestra atención a este trabajo. El modo de operación que se requiere para llevar a un robot de tipo diferencial como el khepera de un punto a otro, es variar la velocidad de cada una de sus ruedas. El autor hace uso del error de orientación del robot con respecto a la pelota, para calcular la velocidad a la que ha de moverse el robot. Las funciones que emplea para calcular las velocidades para cada rueda del robot deben cumplir las siguientes características:

1. Para ángulos pequeños la velocidad de traslación debe ser grande.
2. La velocidad angular debe ser pequeña para ángulos pequeños, particularmente para un ángulo igual a cero, la velocidad angular debe ser cero.
3. La velocidad angular debe ser grande para ángulos grandes.
4. vr y vl no pueden ser ambas negativas.

Tomando en cuenta el siguiente sistema de ecuaciones el autor propone una función gaussiana para el cálculo de la velocidad de traslación y una función sigmoidal para el cálculo de la velocidad angular.

$$\left.\begin{array}{l} v_r = v + \frac{1}{2}w \\ v_l = v - \frac{1}{2}w \end{array}\right\} \text{ si } v_l < v_r \qquad \left.\begin{array}{l} v_r = v - \frac{1}{2}w \\ v_l = v + \frac{1}{2}w \end{array}\right\} \text{ si } v_l > v_r$$

Se propone la siguiente función gaussiana para el cálculo de velocidad, para la cual un $\theta = 0$ alcanza el valor máximo, cumpliendo así con el punto 1 y como una gaussiana siempre es positiva, se cumple también el punto 4.

$$v = v_{max} e^{-\frac{\theta^2}{2\sigma^2}}$$

Para la velocidad **w** se propone la siguiente función

$$w = w_{max}\left(\frac{2}{1+e^{-\frac{\theta}{\sigma_w}}} - 1\right)$$

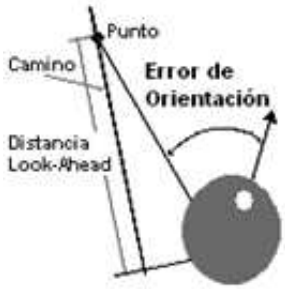
La cual vale cero para un θ igual a cero y conforme el θ crece hasta llegar a $\mathbf{w_{max}}$.

2.5. Generación de trayectorias para un robot móvil khepera II usando técnicas de aprendizaje automático (2007).

La metodología propuesta en esta tesis de maestría [Casarrubias 04], está diseñada para la generación de trayectorias en un espacio poblado de objetos, con la cual se busca evitar choques con esos objetos que son considerados obstáculos. La metodología incluye algoritmos que le permiten extraer características para representar al entorno, a través de una cámara que proporciona vista aérea del área de trabajo en donde navega el robot, en esta etapa de la metodología, se usan técnicas de morfología matemática para la segmentación y clasificación de obstáculos, robot y espacio libre. La planificación de la ruta consiste en encontrar un camino que lleve al móvil a su destino, es decir un punto inicial y el alcance de un punto final, con el empleo de un algoritmo que define un conjunto de celdas conectadas, marcadas como no ocupadas en el proceso de segmentación de la imagen. El mecanismo que el autor propone para la selección del camino es mediante una configuración de un vector que define la posición y orientación que el robot pueda tomar en cualquiera de las

posiciones que están dentro del espacio libre, de igual forma existe una configuración para definir a los obstáculos. Enseguida se adoptan algoritmos que garantizan la navegación del robot alejándose de los obstáculos. Por último se hace un suavizamiento de curvas en atención a que se usará un modelo de control de un robot diferencial. Finalmente lo relevante de este trabajo es el método con el que se hace el seguimiento de la trayectoria planeada, ya que el controlador hace un ajuste en las velocidades de traslación y rotación, mediante marcas obtenidas por el procesamiento de las imágenes que describen la posición y orientación del robot a cada instante que se hace la captura de una nueva imagen. El control reactivo visual proporciona acciones correctivas en las velocidades de traslación y rotación para llevar el robot de un punto a otro, mediante el siguiente sistema de ecuaciones.

$$(v,w) = \begin{cases} (\alpha.cos\theta, \beta.sen\theta) & \text{si } |\theta| < \frac{\pi}{4} \text{ o } |\theta| > \frac{3\pi}{4} \\ (0, sigx\beta_0) & \text{en otro caso} \end{cases}$$



2.6. Sistema de monitoreo autónomo basado en el robot móvil Khepera (2007).

En este trabajo se describe un sistema prototipo de navegación e identificación de puntos de prueba, el autor emplea algoritmos en cascada que perciben su entorno basados en lógica difusa y el uso de una red neuronal de Kohonen de tipo no supervisada para la identificación de puntos de prueba. Estos algoritmos son probados en un robot kherpa, el robot sigue una trayectoria descrita por una línea negra sobre la cual se encuentran obstáculos, los cuales son evadidos por el robot hasta llegar a un punto dentro de una zona de interés o puntos

de prueba según el autor. Para lograr esto el autor implementa una serie de pasos al que denomina algoritmo difuso en cascada. La primera etapa, se compone de una estructura difusa encargada de la función sensorial, compuesta por 8 sensores de de proximidad, con los cuales se realiza las mediciones de proximidad entre el robot y los objetos. Un par de sensores infrarrojos detectan una línea negra. En la segunda etapa se realiza la detección de los denominados puntos de prueba a lo largo de la trayectoria, dicha búsqueda se basa en la implementación de una red neuronal como mecanismo de detección de fuentes emisoras de luz infrarroja, además cuenta con un algoritmo de procesamiento de datos de visión lineal. Este grupo de sensores representan las entradas del sistema.

A continuación se enlistan las cuatro estructuras difusas del sistema que forman la fusión sensorial compuesta de 10 entradas y 4 salidas:

Capítulo 3

Marco teórico

3.1. Introducción

En este capítulo de hace una completa revisión de conceptos relativos a la robótica móvil, redes neuronales, control difuso y procesamiento digital de imágenes. Debido a que el sistema se prueba en un robot de conducción tipo diferencial se hace una completo análisis de conceptos de cuerpos rígidos móviles, cuyo significado puede puede extenderse a cualquier tipo de robot de este tipo, se tiene especial cuidado de presentar los modelos matemáticos clásicos asociados a este tipo de móvil.

3.2. Robot

La palabra robot viene de la traducción al inglés del checo robota que quiere decir (servidumbre, trabajo forzado). Se utilizó por primera vez en la obra del escritor dramaturgo checo Karel Capek en 1921, en su obra titulada R.U.R. (Rossumoví Univerzální Roboti), cuya traducción al español es: Robots Universales de Rossum. Isaac Asimov un escritor estadounidense de origen ruso se unió al género de la ciencia-ficción y propone la palabra robótica, definiéndola como la ciencia que estudia a los robots. Crea tres leyes que los robots de sus obras deben seguir:

- Un robot no debe dañar a un ser humano o, por su inacción, dejar que un ser humano sufra daño
- Un robot debe obedecer las órdenes que le son dadas por un ser humano, excepto si estas órdenes entran en conflicto con la Primera Ley
- Un robot debe proteger su propia existencia, hasta donde esta protección no entre en conflicto con la Primera o la Segunda Ley

3.3. Modelo odométrico diferencial del robot Khepera II

Antes de comenzar el diseño de un controlador para navegación de robots, es indispensable conocer las capacidades cinemáticas del móvil. La cinemática como rama de la mecánica clásica estudia las leyes del movimiento de los cuerpos sin tener en cuenta las causas que lo producen, por otra parte la odometría es usada para estimar la posición de vehículos con ruedas a lo largo del tiempo o la distancia que ha recorrido, o para describir al modelo matemático que genera las consignas que pueden llevar al móvil de un punto inicial a uno final a partir del desplazamiento de las ruedas.

El khepera II, es un robot móvil de manejo diferencial, el cual consta de dos llantas que rotan sobre el mismo eje, controladas de forma independiente por dos motores y dos soportes fijos, al frente y detrás del robot para garantizar que la plataforma circular del robot se mantenga paralela al plano en que se mueve. Este tipo de móvil de conducción diferencial, depende de la variación de las velocidades de los motores que controlan las llantas, para definir su desplazamiento y su orientación para el seguimiento de trayectorias.

Si consideramos al robot como un punto que se mueve en un plano coordenado $(\mathbf{x}, \mathbf{y})$, veremos que el robot puede describir los siguientes movimientos:

- **Movimiento circular:** En este tipo de movimiento el móvil sigue la trayectoria de una circunferencia, basta con que la diferencia que existe entre las velocidades de las dos ruedas sea diferente de cero y sea constante en todo momento hasta alcanzar de nuevo el punto de partida

- **Movimiento de traslación:** Para el movimiento de traslación todo vector relacionado con el sólido tienen la misma dirección, módulo y sentido en cada instante, es decir que la velocidad de ambas ruedas son iguales en magnitud y el móvil se desplaza en una trayectoria rectilínea

- **Movimiento de rotación:** Un sólido está animado de un movimiento de rotación cuando el punto central de su masa no cambia su posición en un plano coordenado y el resto de sus puntos se mueven en una trayectoria circular, entonces el robot khepera realiza movimientos de rotación cuando la velocidad de sus dos ruedas es igual en magnitud pero de sentido contrario

Es posible calcular la posición y orientación del robot como un punto que se mueve en una trayectoria mediante odometría diferencial.

La velocidad total de desplazamiento del robot está dada por:

$$V = \frac{v_l + v_r}{2} \tag{3.1}$$

En donde:

$\mathbf{v_l}$ Es la velocidad de la rueda izquierda.

$\mathbf{v_r}$ Es la velocidad de la rueda derecha.

Sea **X** (revisar figura 3.1), el eje coordenado horizontal en un plano cartesiano rectangular, en donde es posible proyectar el espacio recorrido por el móvil, representado por **s**. Siendo **R** el radio de la circunferencia y θ el ángulo que forman los catetos formados por **XCIR** y **CIRM**, se sabe que

$$s = R.\theta \tag{3.2}$$

La velocidad tangencial del móvil realizando un movimiento circular está relacionada con el radio vector y con la velocidad angular, y se describe con la siguiente expresión de producto escalar:

$$\vec{v} = [\vec{w}\vec{R}] \tag{3.3}$$

El valor algebraico del vector de velocidad situado en la tangente del centro geométrico del

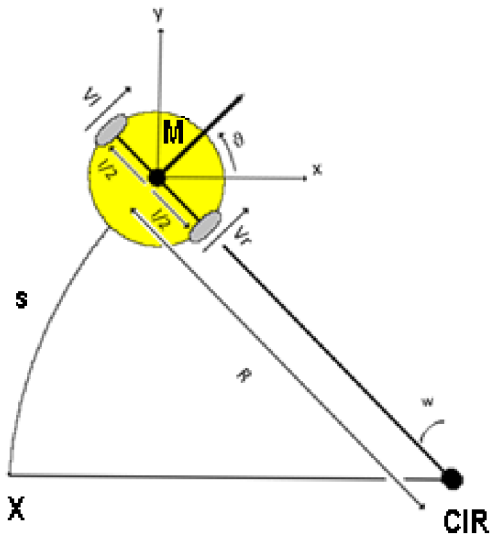


Figura 3.1: Modelo cinemático odométrico

robot esta dado por la siguiente expresión:

$$v = \frac{dh}{dt} = R\frac{d\theta}{dt} = Rw \tag{3.4}$$

Donde θ está en función del tiempo y **w** es la velocidad angular del móvil. Generalmente la velocidad angular se representa por un vector $\overrightarrow{w}$, que es perpendicular al plano de la circunferencia descrita por el móvil, la longitud de este vector está dado por **w**, es decir el módulo, cuyo sentido corresponde al avance de un tornillo realizando un giro de rotación. El módulo de la velocidad angular media se define por:

$$\vec{w} = \frac{\Delta\theta}{\Delta t} \tag{3.5}$$

Su valor algebraico en un instante dado esta dado por:

$$w = \frac{d\theta}{dt} \tag{3.6}$$

La velocidad angular se representa en radianes por segundo, siendo **v** una constante, $\mathbf{s_0}$ el punto correspondiente a la abscisa del avance curvo del móvil en un instante dado tomado como el origen. Si se considera que la velocidad **v** es constante, entonces la velocidad angular **w**, también lo será. En un instante dado al móvil situado en un punto $\mathbf{s_0}$ corresponde un ángulo

φ y tomando en cuenta la ecuación de movimiento circular uniforme dada por:

$$s = vt + s_0 \tag{3.7}$$

Y sustituyendo las ecuaciones [3.2] y [3.4] en [3.7] se obtiene que:

$$R\theta = Rwt + R\varphi \tag{3.8a}$$

$$\theta = wt + \varphi \tag{3.8b}$$

Una vez habiendo definido el concepto de vector tangencial de velocidad, se considera a $\vec{\mathbf{vl}}$ y $\vec{\mathbf{vr}}$ como los vectores tangenciales de velocidad de las ruedas izquierda y derecha que describen la trayectoria de las ruedas del móvil y están dadas por el siguiente sistema de ecuaciones:

$$\vec{v}l = \vec{w}.\left(\vec{R} + \frac{\vec{l}}{2}\right) \tag{3.9a}$$

$$\vec{v}r = \vec{w}.\left(\vec{R} - \frac{\vec{l}}{2}\right) \tag{3.9b}$$

Donde **R** es la distancia del centro geométrico del robot considerado el centro instantáneo de rotación (**CIR**), es decir el radio de la circunferencia que describe la trayectoria del robot, **l** es la distancia que existe entre las dos ruedas del robot. La aceleración es una magnitud vectorial que existe en todo móvil que sigue una trayectoria circular, el módulo de la velocidad permanecerá contante, lo que no sucede con la dirección. Si en el instante **t** la velocidad está dada por el vector $\vec{\mathbf{v}}$, entonces en el instante $t + \Delta t$ la velocidad estará dada por $\vec{v} + \Delta\vec{v}$, se observa que al incremento de tiempo Δt corresponde un incremento de la velocidad $\Delta\vec{v}$. El sentido y dirección de $\Delta\vec{v}$ es el mismo que se define para un vector dado por la razón $\frac{\Delta\vec{v}}{\Delta t}$, para el cual se supone que Δt disminuye hasta alcanzar un límite, de tal forma que la aceleración del móvil se define mediante el vector:

$$\vec{a} = lim_{\Delta t \to 0}\frac{\Delta\vec{v}}{\Delta t} = \frac{d\vec{v}}{dt} \tag{3.10}$$

Se concluye que la aceleración es la derivada de la velocidad con respecto al tiempo. Los correspondientes componentes de la velocidad **v** sobre los ejes $(\mathbf{x}, \mathbf{y})$, en un momento

determinado, se obtienen de la siguiente forma:

$$v_x = -vsen\theta \tag{3.11a}$$

$$v_y = vcos\theta \tag{3.11b}$$

Suponiendo que $\varphi = \mathbf{0}$ y tomando en cuenta a [3.10], se puede obtener las componentes de la aceleración de la siguiente forma:

$$a_x = \frac{dv_x}{dt} = -vwcoswt = -wv_y \tag{3.12a}$$

$$a_y = \frac{dv_y}{dt} = -vwsenwt = -wv_x \tag{3.12b}$$

3.4. Redes Neuronales Artificiales

Durante siglos la pregunta que ha inquietado a los científicos y pensadores es si: ¿Podemos construir máquinas que emulen el funcionamiento del cerebro?, es decir máquinas inteligentes. La disciplina que ha buscado dar respuesta a esta pregunta es la IA (Inteligencia artificial), bajo la firme creencia de que el cerebro humano funciona como un sistema computacional, por lo que para muchos investigadores el conocimiento acerca del funcionamiento del cerebro ha resultado útil para el desarrollo de sistemas u ordenadores cada vez más inteligentes, sin confundir el término máquina con el de ordenador digital y las restricciones que esté último representa, a excepción de Roger Penrose (1989), físico matemático nacido en Inglaterra y Profesor Emérito de Matemáticas en la Universidad de Oxford, quien se volvió famoso entre muchos de sus trabajos, además de ser duramente criticado por su teoría sobre la mente, en la que sostiene que debe de haber algo en la naturaleza de la mente que no es computable, dicho en términos computacionales que no existe un algoritmo que siga las leyes físicas que describen la actividad cerebral, explicable en términos de mecánica cuántica.

A ciencia cierta no se sabe con exactitud cuál es el funcionamiento del cerebro o de los sistemas neuronales biológicos, siguiendo la creencia de que el cerebro funciona como un sistema computacional, se han propuesto metáforas acerca de su funcionamiento. Entre

un grupo selecto de estas metáforas se han considerado cinco como las más aptas para los propósitos de la IA:

- Simbólica
- Conexionista
- Moboticista
- Sociedad de mente
- Auto organizativa

De estas metáforas, la primera y más desarrollada dentro de la IA, es la simbólica. Se basa en la idea de que el ser humano usa símbolos para clasificar el mundo que le rodea y razonar acerca de él. Sin embargo debe quedar claro que aunque la característica que distingue a la IA de los métodos numéricos es el uso de símbolos no matemáticos, no necesariamente implica que cualquier sistema que use simbología no matemática es inteligente, como es el caso de compiladores y sistemas de bases de datos que por lo regular manejan símbolos matemáticos y no matemáticos.
"En este caso, se definen el problema a resolver y se diseña el sistema capaz de resolverlo siguiendo esquemas prefijados por la disciplina. Los sistemas expertos siguen este esquema: se introducen una serie de reglas lógicas que recogen el conocimiento de un experto sobre una materia y mediante mecanismos de inferencia parecidos a los que empleamos al razonar, se sacan conclusiones". [Isasi , Galván 06].

Sin embargo el modelo simbólico no es del todo efectivo, presenta algunos problemas a la hora de construir algoritmos cuyo comportamiento inteligente no esté sólo dirigido a aéreas muy especificas, el problema radica en la dificultad de generalizar la enorme cantidad de información de distinta naturaleza que caracteriza al conocimiento que un humano pueda adquirir a lo largo de su vida, incluyendo al complejo sentido común. Los seres vivos poseen un enorme número de habilidades que dependen de complejos sistema sensoriales, sin meditar o considerar las opciones para tomar una decisión, ósea de carácter no deliberativo,

para ejemplificar esto tomemos a la visión como una de esas habilidades, ya que aún no podemos explicar cómo se logra el reconocimiento instantáneo de un rostro, lo cual para una máquina resulta imposible de lograr bajo la metáfora simbólica.

Es por eso que se recurre al empleo de modelos de redes neuronales artificiales que tienen cierta capacidad para generalizar, que sean capaces de generar respuestas adecuadas para actuar frente a estímulos desconocidos durante algún proceso de entrenamiento o aprendizaje, si y solo si estos estímulos guardan un grado de similitud pequeño a los que se vieron durante el entrenamiento. Bajo el concepto de metáfora conexionista, es que las redes neuronales artificiales modelan el funcionamiento del cerebro, a través de una simulación del aprendizaje al asociar estímulos con respuestas, mediante la aproximación de valores con un peso asignado a las conexiones entre sus nodos, o neuronas si se les compara con los sistemas bilógicos que se describen más adelante. La metáfora conexionista se ha mostrado más apta que la simbólica para manejar capacidades sensoriales, por ejemplo en visión por computadora, reconocimiento de patrones y el control de robots.

"El mecanismo fundamental que capacita a los seres vivos para la realización de tareas sofisticadas no programadas directamente es el sistema nervioso. Desde este punto de vista la perspectiva subsimbólica trata de estudiar los mecanismos de los sistemas nerviosos, del cerebro, así como su estructura, funcionamiento y características lógicas, con la intención de diseñar programas basados en dichas características que se adapten y generen sistemas capaces de resolver problemas. Esto hace imprescindible el estudio profundo de los mecanismos que rigen el comportamiento de los sistemas neuronales, como lo son los fundamentos biológicos de los sistemas neuronales". [Isasi , Galván 06].

3.4.1. Fundamentos biológicos del sistema nervioso neuronal

Santiago Ramón y Cajal, médico español especializado en anátomo-patología microscópica, descubrió en 1906, el mecanismo que gobierna la morfología y procesos conectivos del sistema nervioso, basado en considerar a las neuronas como elementos individuales funcio-

nales del sistema nervioso, como entidades que se comunican unas con otras, establecían una especie de red mediante conexiones especializadas. Lo cual le valdría el premio nobel.

El cerebro humano contiene alrededor cien mil millones 10^{11} de neuronas, una neurona se conecta con alrededor de 10,000 de neuronas formando así una red de neuronas gigantesca en el sistema nervioso humano. Aproximadamente 10 billones de neuronas conectadas consideradas unidades de procesamiento lento que trabajan en paralelo para alcanzar un potente procesamiento.

Son cuatro los elementos más importantes del sistema biológico neuronal que forman parte del estudio de la IA para el desarrollo de algoritmos inteligentes que emulen el funcionamiento del cerebro humano. Véase figura 3.2

- Sinapsis: La conexión que existe entre neuronas
- Dendritas: Son las extremidades de la neurona que recibe señales eléctricas
- Axón: Envía señales de salida a otras neuronas
- Núcleo: Podría considerarse al elemento que se encarga del procesamiento, por lo que se considera la función principal

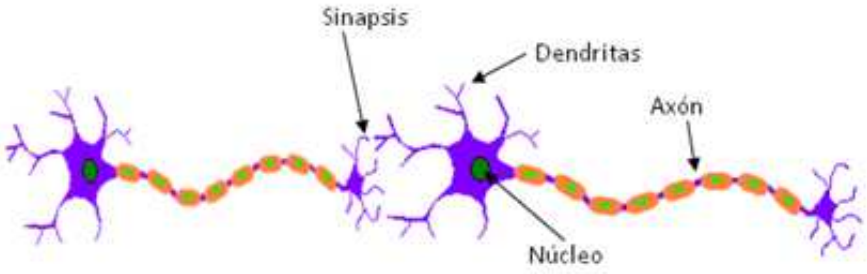


Figura 3.2: Sistema nervioso neuronal bilógico

Las neuronas pueden comunicarse con precisión, rapidez y a distancia con células nerviosas, musculares o glandulares, a través de señales eléctricas denominadas impulsos nerviosos, Estos impulsos nerviosos viajan por la neurona comenzando por las dendritas, y pasa por toda la neurona hasta llegar a los extremos terminales, que pueden conectarse con otras neurona,

fibras musculares o glándulas. Algunas de las células que forman parte del sistema nervioso, consideradas receptores de estímulos, reciben información del exterior de forma visual, auditiva, táctil, etc. La cual es procesada por el sistema nervioso y convertida a impulsos electro-químicos, que se propagan asíncronamente por todo el sistema biológico hasta llegar a órganos efectores, que se encargan de actividades motoras, glandulares, secreciones, etc.

"Uno de los aspectos más interesantes consiste en que las conexiones sinápticas poseen cierta plasticidad (el grado de influencia de las sinapsis cambia con el tiempo, en incluso se crean nuevas sinapsis). Aunque no se conocen los mecanismos exactos, se sabe que las capacidades de aprendizaje y memorización que poseen los seres vivos se basan en la plasticidad del cerebro." [Vallejo 01].

3.4.2. Estructura básica de las redes neuronales artificiales

El modelo computacional de una red neuronal artificial, consiste en encontrar un modelo de estructura de datos con capacidad de abstracción, es decir capaces de encontrar relaciones (patrones) de forma inductiva, sin salir del espacio de trabajo de la neurociencia.

Estos modelos neuronales también se caracterizan por que de alguna forma elaboran la información de entrada para generar su salida. Las RNA(Redes Neuronales Artificiales) pueden ser consideradas estructuras distribuidas, de procesamiento paralelo, formada de neuronas artificiales (elementos de procesamiento), interconectados por un gran número de conexiones (sinapsis), los cuales son usados para almacenar conocimiento que está disponible para poder ser usado. Estas estructuras pueden ser consideradas como un sistema de procesamiento de información con características como aprendizaje con capacidad de generalización y cierta tolerancia a errores bajo un umbral.

El recorrido que sigue la información dentro de la red es una de las características más comunes que diferencian el rendimiento de estos modelos de RNA son redes alimentadas

hacia adelante y redes con retro alimentación.

3.4.3. Modelo de una neurona artificial

"La neurona artificial, célula o autómata, es un elemento que posee un estado interno, llamado nivel de activación y recibe señales que le permiten, en su caso, cambiar de estado. Si se denomina **S** *al conjunto de estados posibles de una neurona,* **S** *podrá ser, por ejemplo* $\mathbf{S} = \mathbf{0}, \mathbf{1}$*, siendo* **0** *el estado inactivo y* **1** *el activo.* **S** *también podrá tomar un mayor número de valores,* $\mathbf{S} = \mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}$ *para representar por ejemplo, una imagen con* $\mathbf{n}+\mathbf{1}$ *niveles de gris, o incluso un intervalo continúo de valores, por ejemplo* $\mathbf{S} = [\mathbf{0}, \mathbf{1}]$*."* [Isasi , Galván 06]. En la

Figura 3.3: Esquema de una unidad neuronal típica

figura 3.3 se muestra el modelo básico de red neuronal artificial, en la que puede observarse que puede existir **n** número de entradas, las cuales para este caso están representadas por las variables $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$, las cuales se encuentran asociadas a pesos que son representados por las variables $\mathbf{w_{ij}}$ los cuales determinan el nivel de fuerza de una conexión sináptica, es decir la influencia de la neurona **j** a la neurona **i**.

Existen dos etapas de proceso para cada una de las neuronas, suma y activación.

Primera etapa: las señales de entrada $\mathbf{x_j}$ y los pesos $\mathbf{w_{ij}}$ son relacionadas con la siguiente sumatoria.

$$y_i = \sum_{j=1}^{n} w_{ij} x_j \tag{3.13}$$

Donde $\mathbf{y_i}$ es el estado interno de la neurona **i**.

Segunda etapa: La salida de la neurona es generada a través de una función de activación.

$$x_i = f(y_i) \tag{3.14}$$

Donde la salida de la neurona es representada por $\mathbf{x_i}$ y **f** es la función de activación aplicada al estado interno de la neurona, que tiene como objetivo producir la señal de salida de la neurona, el nivel de activación se situará entre $[\mathbf{0}, \mathbf{1}]$ o $[-\mathbf{1}, \mathbf{1}]$, para el caso de que xi sea un valor continuo, y entre $\mathbf{0}, \mathbf{1}$ o $-\mathbf{1}, \mathbf{1}$ en el caso de que $\mathbf{x_i}$ sea un valor discreto. Existen varios tipos de función de activación, la función de grado, la tangente hiperbólica y la sigmoidea, como se muestra en la siguiente figura 3.4.

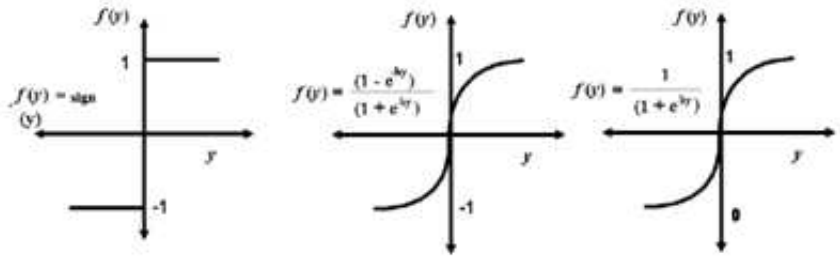


Figura 3.4: Funciones de activación. A)Función de activación de grado. B)Función de activación tangente hiperbólica. C)Función de activación sigmoidea.

3.4.4. Arquitectura de una RNA

La arquitectura de una RNA puede estar compuesta de una capa o de múltiples capas, las que pueden ser consideradas dentro de tres grupos:

- Capa de entrada
- Capas intermedia o capas a ocultas
- Capas de salida

Las redes neuronales artificiales se clasifican en dos clases: FeedForward (propagación hacia delante) y Redes Recurrentes, la diferencia entre ellas, es la forma en que las salidas de las neuronas puedan convertirse en entradas de otras neuronas. La señal de salida de una neurona puede ser una entrada de otro elemento de proceso, o incluso de sí mismo (auto-recurrente).

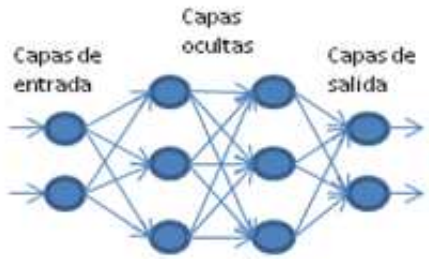


Figura 3.5: Esquema de RNA de múltiples capas

- **Redes feedforwar**

  La estructura de una red feedforward consiste en que ninguna de las señales de salida de una neurona de una capa, sea la señal de entrada de una neurona del mismo nivel o de niveles anteriores, es decir que la salida de una neurona solo alimente a todas las neuronas de la capa siguiente y por consiguiente no existen las conexiones de retroalimentación.

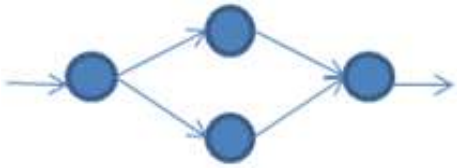


Figura 3.6: Esquema de una RNA Feedforward

- **Redes recurrentes**

  Este tipo de redes son conocidas como, de propagación hacia atrás, las salidas de cada neurona pueden estar conectadas como entradas de otras neuronas del mismo nivel, de niveles anteriores o de ellas mismas.

  De esta forma el tipo de conexión de las redes neuronales artificiales puede agruparse en tres:

  1. Conexiones hacia delante
  2. Conexiones laterales

3. Conexiones hacia atrás (o recurrentes)

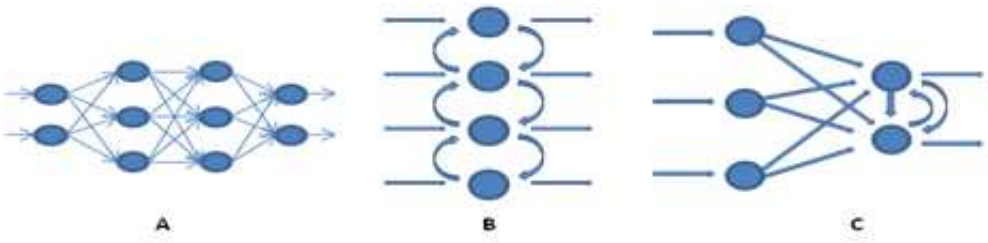


Figura 3.7: A) Conexión hacia delante. B) Conexión lateral. C)Conexión hacia atrás(recurrente).

En la figura 3.7 se ejemplifican los tipos de conexiones, a partir de las cuales se pueden clasificar las redes en dos tipologías:

- **Redes mono-capa** Este tipo de redes son comúnmente utilizadas para propósitos de auto asociación, como en la reconstrucción de señales incompletas, la red de [Hopfield, 1982], una red recurrente, la cual funciona como un modelo de memoria asociativa de patrones ya que es capaz de recuperar patrones a través de información incompleta o con ruido.
- **Redes multicapa** Este tipo de redes como su nombre lo indica están formadas por más de una capa, una característica básica de esta topología es que todas las neuronas de una capa en común reciben señales de entrada de una capa anterior y de la misma forma envían señales a capas posteriores es decir de redes feedforward. En otro grupo de esta topología existe la posibilidad de conectar salidas de las neuronas de capas posteriores a entradas de capas anteriores, es decir conexiones feedback. A su vez estos tipos de arquitecturas dan pie a una red más del tipo multicapa en las que se dispone de conexiones hacia adelante y hacia atrás (feedforward-feedback).

El mecanismo de aprendizaje de las redes neuronales artificiales, consiste en la modificación de los pesos como consecuencia de la excitación que representa la información de entrada

con valores que pueden ser positivos o negativos, los criterios bajo los cuales puede cambiar el valor de los pesos asignados a las conexiones, cuando se quiere que la red aprenda a partir e nueva información, suelen ser considerados de dos tipos:

- Aprendizaje supervisado
- Aprendizaje no supervisado

La diferencia entre estos dos tipos de aprendizaje parece ser tan simple como que una cuenta con un agente externo que determina la salida que debería generar la red a partir de una entrada definida, en donde el supervisor comprueba que la salida de la red coincida con la esperado y en caso de que ésta no, se modifican los pesos de las conexiones, con el fin de que la salida obtenida se aproxime a la esperada ya sea por corrección de error o refuerzo. Por el contrario las redes con aprendizaje no supervisado no requieren de un supervisor para ajustar los pesos asociados a las conexiones entre sus neuronas, esta red no recibe ninguna información que le permita saber si su salida es correcta, se considera que son modelos que tiene la capacidad de auto organizarse.

3.4.5. Perceptrón

El modelo de perceptrón desarrollado por Frank Rosenblatt, fue el primer sistema de red neuronal capaz de realizar tareas de clasificación automática. Este modelo intenta modelar el comportamiento de una neurona biológica, derivado del modelo de McCulloch y Pitts (Looney 1997), puede considerarse un discriminador lineal. El perceptrón trabaja con funciones de activación.

La estructura del cuerpo de la neurona asemeja al de una neurona biológica y puede verse como un sumador de tipo lineal de estímulos previo a una función de activación lineal, la cual genera la salida del sistema a partir de la suma de los estímulos. En este modelo cada entrada se multiplica por un factor de peso correspondiente $\mathbf{w}$, cuyos resultados son sumados y comparados contra el valor de umbral representado por θ, si el valor es mayor que el máximo entonces se activa. Figura 3.8. La fase de entrenamiento del perceptron se hace

mediante una función de aprendizaje, la cual consiste en modificar el valor de los pesos de las conexiones, siempre que la respuesta dada por el perceptrón sea incorrecta.

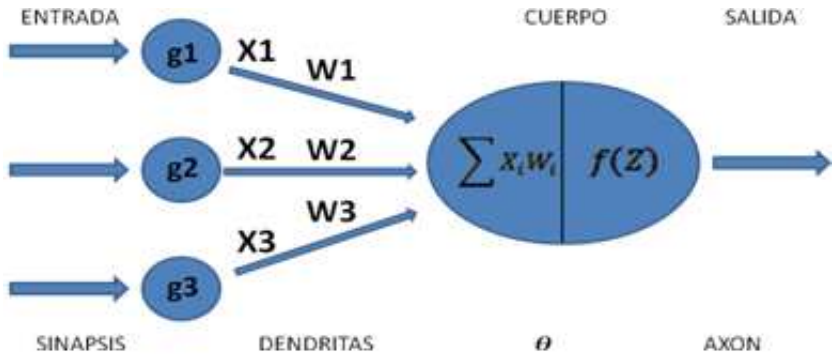


Figura 3.8: Modelo del perceptrón

3.4.6. Red Adaline

La red ADALINE (Adaptative Linear Element), desarrollada por Bernie Windrow en 1960, es un tipo de red que consta de una sola neurona de salida, cuya función de transferencia es de tipo escalón, este modelo usa la regla delta, desarrollada por Windrow-Hoff, mejor conocida como regla del mínimo error cuadrado medio, la cual busca el mínimo error entre la salida lineal obtenida y la salida deseada, antes de usar una función de activación, ya sea lineal o sigmoidea. Ya que las salidas de esta red son binarias solo pueden codificarse un conjunto discreto de estados.

"ADALINE. Es un elemento combinador adaptativo, que recibe un conjunto de entradas y las combina para producir una salida. Estas salidas puede tranformarse en binaria mediante un conmutador bipolar que produce un 1 si la salida es positiva y un -1 si es negativa." [Isasi , Galván 06].

$$\bar{y} = \sum_{i=1}^{n} w_i . x_i + \theta \tag{3.15}$$

En donde θ es un umbral.

Existen más criterios que sobre el proceso de aprendizaje, ya que también es posible que algunos modelos puedan aprender durante el funcionamiento de la red es decir (en línea), o si

el proceso de aprendizaje requiere desconectar el sistema hasta que el proceso de aprendizaje termine, en cuyo caso se considera un sistema de aprendizaje fuera de línea.

3.4.7. Perceptron Multicapa

El Perceotron Multicapa o red multicapa con conexiones hacia delante, este modelo surge a partir de las limitaciones de separabilidad no lineal que tiene el modelo de perceptron simple, en el cual Minsky y Papert en 1969 mostraron que la combinación de varios perceptrones simples, incluso en las neuronas ocultas, podía dar solución a algunos problemas no lineales, aun cuando el modelo que propusieron incluía el uso de perceptrones en capas ocultas no dieron solución al problema de adaptar los pesos de las capas ocultas. Rumelhart, Hinton y Willians en 1986, propusieron una forma de retro propagar los errores en salida de la red hacia las neuronas ocultas, basándose en la idea de Minsky y Papert de combinar varios perceptrones, dando así lugar a la regla delta generalizada. [Isasi , Galván 06].

Este modelo ha sido usado ampliamente con éxito en tareas de reconocimiento de patrones, en una amplia variedad de aéreas, como en el reconocimiento de formas, habla, señales, etc. Debido a su gran capacidad como aproximador universal.

La arquitectura del Perceptrón Multicapa, se caracteriza porque tiene tres tipos diferentes de capas: la capa de entrada, las capas ocultas y la capa de salida. Véase figura 3.9 Esquema de un perceptrón multicapa, otra característica es que las neuronas de la capa de entrada tienen la única función de recibir y propagar los datos de entrada a todas las neuronas de la capa posterior. Por otra parte las neuronas de la capa de salida se encargan de proporcionar al exterior la respuesta para cada uno de los patrones de entrada. Como podemos ver las capas de entrada y salida desempeñan las funciones comunes que describen su nombre, entrada y salida. Las neuronas de las capas ocultas, son quienes realizan el procesamiento de los patrones recibidos de forma no lineal.

El perceptrón multicapa es una red de tipo feedForward ya que sus conexiones están dirigidas hacia delante, los pesos asociados a las conexiones están representados por un número real. De la misma forma todas la neuronas de la red tiene asociado un umbral, que se consi-

dera una conexión mas de cada neurona cuyo de valor del umbral es contante igual a 1.

El perceptrón multicapa puede ser totalmente o localmente conectado ya que pueden darse casos en los que neuronas de una determinada capa pueden estar conectadas a capas inmediatas y a capas no necesariamente inmediatas o redes en las que neuronas de cierta capa no están conectadas a la siguiente capa, por lo que se considera que el valor de esos pesos es contante e igual a cero. En este trabajo se hará uso de aquellas totalmente conectadas.

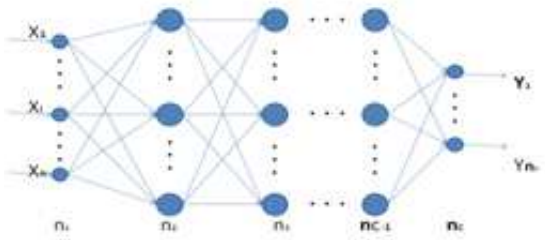


Figura 3.9: Esquema del perceptrón multicapa

A continuación se explica a detalle la relación que existe entre las variables de entrada y las de salida, así como las expresiones de activaciones de las neuronas durante la propagación hacia delante y por último el algoritmo de retro propagación para el perceptrón multicapa.

"Sea un perceotrón multicapa como el de la figura 3.9 en donde $\mathbf{C}$ *representa el número de capas de la red incluyendo a la capa de las neuronas de entrada y la capa de salida, por lo tanto el número de capas ocultas esta dado por* $\mathbf{C-2}$*, con* $\mathbf{n_q}$ *neuronas en la capa* $\mathbf{q}$*, donde* $\mathbf{q = 1, 2, \ldots, C}$*. Existe también una matriz de pesos la cual está representada por la siguiente expresión:*

$$w^q = (w^q_{ij}) \tag{3.16}$$

"Pesos que se encuentran asociados a las conexiones de la capa $\mathbf{q}$ *a la capa* $\mathbf{q+1}$ *para* $\mathbf{q = 1, 2, \ldots, C-1}$*, donde* $\mathbf{w^q_{ij}}$ *representa el peso de la conexión de la neurona* $\mathbf{i}$ *de la capa* $\mathbf{c}$ *a la neurona* $\mathbf{j}$ *de la capa* $\mathbf{q+1}$*. Sea* $\mathbf{U^q} = (\mathbf{u^q_i})$ *representa al vector de umbrales de las neuronas que van de la capa* $\mathbf{q}$ *para* $\mathbf{q = 2, \ldots, C}$*. La activación de la neurona* $\mathbf{i}$ *a la neurona de la capa* $\mathbf{q}$ *se denota como* $\mathbf{a^q_i}$*".* [Isasi , Galván 06].

Con la descripción de los parámetros que acompañan a la red perceptrón multicapa se procede a describir cómo se comportan estos parámetros dentro de la red.

- Activación de las neuronas de la capa de entrada. Las neuronas de la capa de entrada adquieren el valor del vector de características en turno ya que la función de estas es transmitir hacia delante las señales de entrada:

$$a_j^1 = x_i \tag{3.17}$$

  En donde $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_{n1}})$ es el vector de patrones de entrada e $\mathbf{i} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{n_1}$

- Activación de las neuronas de la capa oculta. La activación de las neuronas de la capa oculta está dada por la suma del producto de las activaciones de entrada por los pesos correspondientes:

$$a_i^q = f(\sum_{j=1}^{n_{q-1}} w_{ji}^{q-1} a_j^{q-1} + u_i^q) \tag{3.18}$$

  Donde $\mathbf{i} = \mathbf{1}, \mathbf{2}, \ldots, \mathbf{n_q}$ y $\mathbf{q} = \mathbf{2}, \mathbf{3}, \ldots, \mathbf{C-1}$ y $\mathbf{a_j^{q-1}}$ corresponde a las neuronas de la capa $\mathbf{q-1}$.

- Activación de las neuronas de la capa de salida. Al igual que en los dos casos anteriores la activación de estas neuronas de salida está dada por la suma de los productos de las entradas por sus pesos correspondientes.

$$y_i = a_i^C = f(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C) \tag{3.19}$$

  Donde $\mathbf{Y} = (\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_{n_C}})$ es el vector de salida.

Las funciones de activación más usadas para el perceptrón multicapa son la función sigmoidal y la función hiperbólica, cuyos valores se encuentran en un rango continuo de valores en los intervalos [0,1] y [-1,1]. Revisar Figura 3.4 B) Sigmoidal y 3.4 C) hiperbólica. En las que se puede observar que las dos funciones son crecientes con dos niveles de saturación, uno máximo con valor a 1 para ambas y un mínimo 0 para la función sigmoidal y -1 para la hiperbólica.

- Características del modelo:
  - Las funciones de activación en las capas ocultas deben ser funciones continuas y diferenciables;
  - Dependiendo de la naturaleza del problema las neuronas que se encuentran en la capa de salida se distinguen del resto de las neuronas de la red utilizando otro tipo de función de activación. Las más usadas son la función de identidad $\mathbf{f}(\mathbf{x}) = \mathbf{x}$ ó la función línea y la función escalón.

- Algoritmo de retro propagación

  La salida de la red debe de ser lo más próxima posible a la salida deseada $(\mathbf{y})$, el aprendizaje de la red se trata como un problema de minimización del siguiente modo:

$$Min_w E \tag{3.20}$$

  Donde **w** es el conjunto de pesos y umbrales de la red, **E** es una función de error que evalúa la diferencia entre las salidas de la red y las salidas deseadas. Comúnmente la función de error más usado se define como:

$$E = \frac{1}{N}\sum_{n=1}^{N} e(n) \tag{3.21}$$

  Donde:

$$e(n) = \frac{1}{2}\sum_{i=1}^{n_c} (y_i(n) - \hat{y}_i(n))^2 \tag{3.22}$$

  $\mathbf{y_i}(\mathbf{n})$: Salidas deseadas

  $\hat{\mathbf{y}}_\mathbf{i}(\mathbf{n})$: Salidas de la red

  N : Numero de patrones de entrada

  $\mathbf{e}(\mathbf{n})$: El error cometido por la red para el patron **n**

  $\mathbf{n_c}$: Numero de neuronas en la capa de salida

  De tal forma que si $\mathbf{W}^*$ es un mínimo de la función del error **E** en dicho punto el error es próximo a cero, implica que la salida de la red sea próxima a la salida deseada, alcanzando así la meta de la regla de aprensizaje.

Función no lineal

$$y = f(x) \approx \hat{y} = \phi(W^* x + U^*) + \varepsilon \tag{3.23}$$

$$\phi = Funcindeactivacin$$

$$W^* = (W^*, U^*) \tag{3.24}$$

$$\varepsilon = Errordemodelado$$

- Regla de aprendizaje

  El aprendizaje de una red neuronal artificial se resume estrictamente al acto de reducir el error total $(\mathbf{E})$. El procedimiento más común está basado en el método del gradiente estocástico, que consiste en una sucesiva minimización del error para cada patrón $\mathbf{e}(\mathbf{n})$. Aplicando el método de descenso de gradiente estocástico, cada parámetro de la red se modifica para cada patrón de entrada **n** de acuerdo a la siguiente ley de aprendizaje:

$$W \begin{cases} w(n) = w(n-1) - \delta \frac{\partial e(n)}{\partial w} \\ u(n) = u(n-1) \delta \frac{\partial e(n)}{\partial u} \end{cases} \tag{3.25}$$

- Regla delta generalizada(aprendizaje) Backpropagation

  Su propósito es generalizar la regla delta sobre Redes Neuronales Artificiales de múltiples capas y funciones de transferencia no lineales y diferenciables.

  Su principal objetivo consiste en ajustar los pesos, tratando de minimizar la suma de error cuadrático, de manera continua modificando dichos valores en dirección contraria a la pendiente del error, este método es conocido como la técnica del gradiente descendente y se le considera como aprendizaje supervisado. Las redes neuronales artificiales entrenadas bajo estas condiciones presentan razonables respuestas ante entradas nunca antes vistas. A cada entrada le hará corresponder una salida similar a la salida obtenida para un patrón de entrenamiento siendo similar al patrón sometido a la red. Este modelo presenta una potente capacidad de generalización.

  Como función de activación se usara la función sigmoidal

$$f(x) = \frac{1}{1+e^{-x}}; f(x) = f(x)(1-f(x))$$

Caso a) Pesos de la capa oculta $C-1$ a la capa de salida $0 < \delta < 1$.

$$w_{ij}^{C-1}(n) = w_{ij}^{C-1}(n-1) + \delta S_i^C(n) a_j^{C-1}(n) \tag{3.26a}$$

$$\hat{y}(n) = f'(\Sigma wx + U) = \hat{y}_i'(n) = (1-\hat{y}_i'(n)) \tag{3.26b}$$

$$U_i^C(n) = U_i^C(n-1) + \delta S_i^C(n) \tag{3.26c}$$

$$S_i^C(n) = (y_i(n) - \hat{y}_i(n))\hat{y}_i(n)(1-\hat{y}_i(n)) \tag{3.26d}$$

$$para\ j = 1,2,...,n_{C-1}$$
$$i = 1,2,...,n_C$$

Caso b) Pesos de la capa q a la capa $q+1$ y umbrales de las neuronas de capa $q+1$ para $q = 1,2,\ldots,C-2$

$$w_{kj}^q = w_{kj}^q(n-1) + \delta S_j^{q+1}(n) a_k^q(n) \tag{3.27a}$$

$$U_j^{q+1}(n) = U_j^{q+1}(n-1) + \delta S_j^{q+1}(n) \tag{3.27b}$$

$$S_j^{q+1}(n) = a_j^{q+1}(n)(1-a_j^{q+1}(n)) \sum_{i=1}^{n_{q+2}} S_i^{q+2}(n) w_{ji}^{q+1} \tag{3.27c}$$

$$para\ k = 1,2,...,n_q$$
$$j = 1,2,...,n_{q+1}$$

Limitaciones El Perceptrón Multicapa no extrapola bien, es decir, si la red se entrena mal o insuficientemente las salidas pueden ser imprecisas. La existencia de mínimos locales en la función de error dificulta considerablemente el entrenamiento pues una vez alcanzado un mínimo el entrenamiento se para aunque no hayamos alcanzado la tasa de convergencia fijada.

3.5. Lógica Difusa

La teoría de conjuntos fue introducida a principios de la década de los sesenta, por el Dr. Lofti A. Zadeh, profesor emérito del Departamento de Ingeniería Eléctrica y Ciencias de la

Computación perteneciente a la Universidad de California en Berkeley. Según Zadeh la idea nace de la necesidad de dar solución a problemas de clasificación de categorías imprecisas, ya que en sus análisis de sistemas se daba cuenta que las técnicas convencionales eran demasiado precisas o poco flexibles ante problemas del mundo real. Luego de cancelar una cena con amistades, una noche en el apartamento de sus padres en Nueva York se le ocurre el concepto de grado de membresía. En 1965 Zadeh publicó el articulo "Fuzzy Sets" con lo que se funda la teoría moderna de lógica difusa, los demás conceptos fueron introducidos a finales de los sesenta y principios de los 70. En 1972 escribe un artículo titulado "A rationale for fuzzy Control" publicado por ASME Journal of Dynamic Systems, Measurement and Control. En 1973 publicó un artículo titulado "Outline of a new approach to the analysis of complex systems and decisión processes", en el cual define conceptos como variable lingüística, gráfica difusa y reglas difusas. Dos o tres años más tarde Mandani y Assilian publicaron un artículo, en el que demuestran el alcance de la teoría de lógica difusa como parte de un controlador, implementado en un control para un motor de vapor, con el uso de conjuntos difusos, variables lingüísticas y reglas difusas del tipo "Si (condición) then (realiza)", las cuales desde entonces se usan con éxito para dar solución a sistemas de control del mundo real.

3.5.1. Conceptos de conjuntos difusos

Es un conjunto que puede contener elementos con grados parciales de membresía es decir, que sus elementos no le pertenecen de una manera absoluta, sino de forma graduada. El nivel de membresía de cada elemento a un conjunto difuso, indica lo que se denomina función de membresía, a diferencia de los conjuntos clásicos en donde un elemento pertenece o no pertenece, pues los métodos que usa la lógica matemática clásica son muy restrictivos.

"Un conjunto difuso puede representarse en forma continua utilizando una gráfica o en forma discreta mediante un conjunto de pares ordenados $(x, \mu(x))$*, donde x es el elemento y* $\mu(x)$ *es su grado membresía en el conjunto difuso"* [García Cabrera 07].

Soporte

El soporte de un conjunto difuso **A** es el conjunto de todos los puntos **x** para los cuales la función de membresía $\mu_{\mathbf{A}}\mathbf{x} > \mathbf{0}$.

Centro

El centro de un conjunto difuso **A**, es el valor promedio de todos los elementos del conjunto con grado de membresía igual a la altura. Si el valor promedio es positivo infinito o negativo infinito, entonces el centro se define como el más pequeño o el más grande de todos los elementos con el valor de grado de membresía máximo.

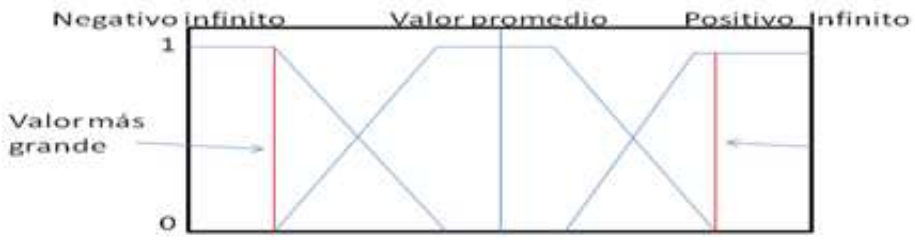


Figura 3.10: Ejemplo de centro de conjuntos difusos

Altura

La altura de un conjunto difuso es el valor máximo que asume su función de membresía. Si la altura es 1, el conjunto difuso es un conjunto difuso normal, es decir que un conjunto es normal solo si existe un punto para el cual la función de membresía es 1, dicho de otra manera que el centro del conjunto no es vacío vease figura 3.11.

Punto de cruce

Son los puntos del conjunto difuso para los cuales $\mu_{\mathbf{A}}(\mathbf{x}) = \mathbf{0.5}$, si el soporte de un conjunto difuso se le conoce como conjunto difuso vacío. Si el soporte de un conjunto difuso es el conjunto nulo, entonces el conjunto difuso se le conoce como conjunto difuso vacio.

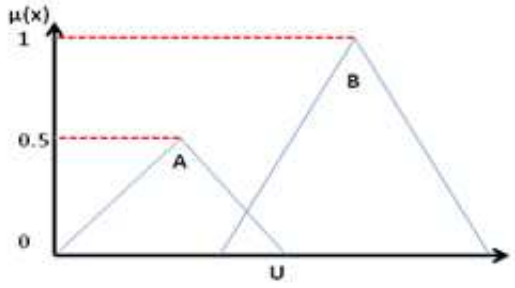


Figura 3.11: Ejemplo de altura y normalidad de un conjunto difuso

Corte α

El corte de un conjunto es el sub-conjunto de los elementos que del conjunto universal que pertenecen al conjunto difuso con un grado de membresía mayor o igual que α, es decir la parte superior que no está dentro del área del conjunto difuso que cubre su grado de membresía.

$$A_\alpha = \{x \varepsilon U | \mu_A(x) \geq \alpha\} \tag{3.28}$$

Variable lingüística

Una variable lingüística, es una variable cuyos valores son palabras o sentencias en un lenguaje natural. Por ejemplo, la distancia que marca un espacio entre dos puntos, "Distancia.[es] una variable lingüística y sus valores son, "muy grande", "grande", "pequeña", "muy pequeña", "nula", es decir que se escogerán palabras de preferencia cortas que sean descriptivas con relación al significado de la variable lingüística elegida.

3.5.2. Función de membresía

Esta función está en el intervalo **0** y **1**, para ser mas explícito esta función da para cada punto en el eje **X** (Universo de Discurso), un grado de membresía, siendo **1** el valor de membresía máximo. A continuación se enlistan las diferentes funciones de membresía utilizadas en la lógica difusa comúnmente más utilizadas descritas en el trabaja de Luis I. García Cabrera **[García Cabrera 07]**.

Singleton

Es el conjunto difuso para el cual el soporte es solo un punto en **U**, en el cual el valor de la función de membresía es **1**.

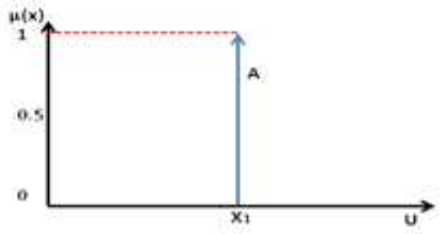


Figura 3.12: Ejemplo de una conjunto singleton

Triangular

Transforma un valor numérico real $\mathbf{x}^* \varepsilon \mathbf{U}$ en un conjunto difuso **A** en **U**, el cual posee la siguiente función de membresía triangular.

$$\mu_A(A) = \begin{cases} 1 & \text{si } x^* = x_c \\ \left(1 - \frac{|x_c - x^*|}{x_c - a}\right) & \text{,si } a < x^* < x_c \\ \left(1 - \frac{|x^* - c|}{d - c}\right) & \text{,si } x_c < x^* < b \\ 0 & \text{,en cualquier otro caso} \end{cases} \tag{3.29}$$

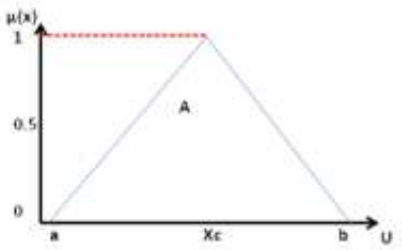


Figura 3.13: Ejemplo de conjunto triangular

Trapezoidal

Transforma un valor numérico real $\mathbf{x}^*\varepsilon\mathbf{U}$ en un conjunto difuso **A** en **U**, el cual posee la siguiente función de membresía trapezoidal.

$$\mu_A(A) = \begin{cases} 1 & \text{,si } b \leq x^* \leq c \\ \left(1 - \frac{|b-x^*|}{b-a}\right) & \text{,si } a < x^* < b \\ \left(1 - \frac{|x^*-c|}{d-c}\right) & \text{,si } c < x^* < d \\ 0 & \text{,en cualquier otro caso} \end{cases} \tag{3.30}$$

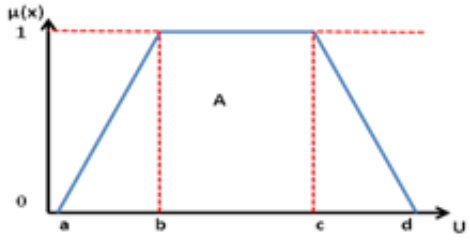


Figura 3.14: Ejemplo de conjunto trapazoidal

Gaussiano

Transforma un valor numérico real $\mathbf{x} * \varepsilon\mathbf{U}$ en un conjunto difuso **A** en **U**, el cual posee la siguiente función de membresía gaussiana.

$$\mu_A(x) = e^{\frac{-(x^*-x_c)^2}{2\sigma^2}} \tag{3.31}$$

Donde $\mathbf{x_c}$ es el valor de **x** donde la función gaussiana es máxima y σ es la desviación estándar.

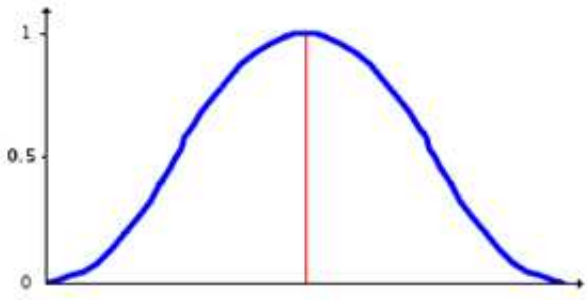


Figura 3.15: Ejemplo de un conjunto gaussiano

3.5.3. Forma típica de un controlador difuso

Un controlador de lógica difusa típico está formado de cuatro componentes:

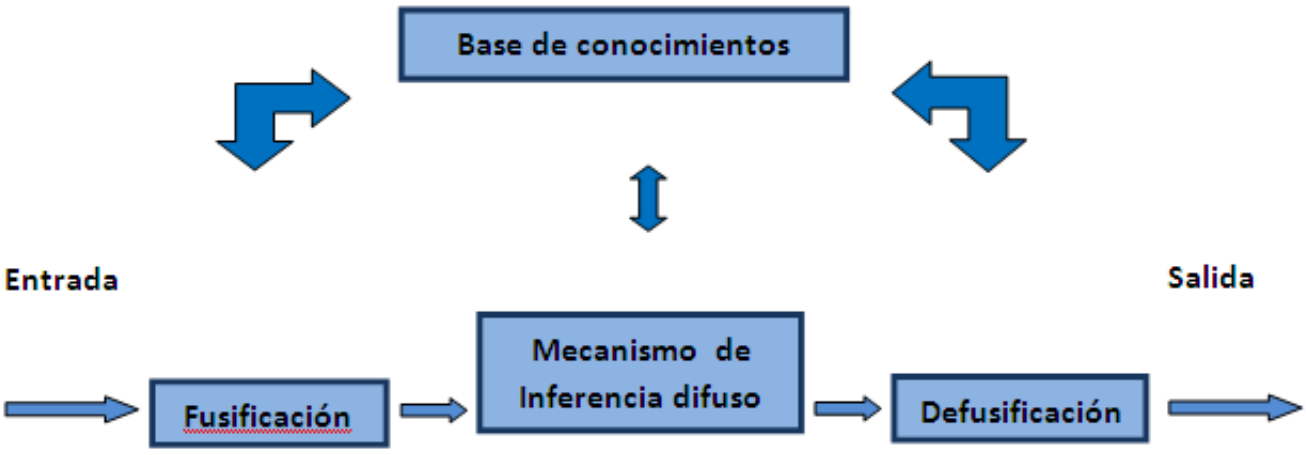


Figura 3.16: Estructura de un controlador difuso típico

Fusificador

Éste es el módulo del sistema difuso mediante el cual los valores numéricos de entrada son convertidos en formas difusas como se muestra en el punto 3.5.2 de este capítulo, lo cual es necesario para que el sistema de inferencia pueda interpretar y procesar las entradas.

Base de conocimiento

La base de conocimiento, almacena un conjunto de reglas de tipo IF-THEN que miden el grado de verdad del conocimiento que es proporcionado por el experto en el área para el cual está siendo diseñado el controlador difuso.

Mecanismo de inferencia difuso

Este mecanismo está encargado de seleccionar cual de las reglas difusas serán el recurso para tomar una serie de decisiones que se tomarán como salidas, la selección de ellas se hace mediante la evaluación de los valores numéricos de entrada que satisfagan la exigencia de cada regla para funcionar.

Defuzificador

Combina las conclusiones numéricas a las que llega el mecanismo de inferencia y las convierte en un valor numérico real que podrá ser utilizado por el tipo de actuador para el cual fue implementado el controlador difuso.

3.5.4. Tipos de defuzificadores y métodos de inferencia

Los problemas basados en el uso de conjuntos difusos, requieren de la evaluación de variables lingüísticas mediante calificaciones, para dar una respuesta difusa a los conjuntos difusos correspondientes a un determinado problema y que después debe convertirse en una respuesta no difusa, es decir una respuesta de salida clara. Para esto existe el concepto de "defuzificación". Se han propuesto algunos métodos para transformar una respuesta difusa en una respuesta no difusa, el cual debe ser escogido de acuerdo a la aplicación que se desea hacer. A continuación se presenta algunos de ellos:

Defuzificador de criterio máximo

La salida de este defuzificador es el valor numérico que está asociado al punto más alto de la función de membresía del conjunto difuso de salida.

$$s^* cualquier punto en la altura(A)$$

Donde $\mathbf{s}^*$ es la variable numérica de salída y **A** en conjunto difuso de salida. De acuerdo al número de puntos de la altura, se pueden especificar tres defuzificadores, a menos que la altura contenga un solo punto $\mathbf{s}^*$:

- Defuzificador del valor más pequeño del máximo

$$s^* = min x \varepsilon altura(A)$$

- Defuzificador del valor más grande del máximo

$$s^* = max x \varepsilon altura(A)$$

- Defuzificador del promedio de los centros

$$s^* = \frac{\int_{altura(A)} s ds}{\int_{altura(A)} ds} \tag{3.32}$$

  El valor numérico real de este defuzificador, está dado por el promedio de los centros de los n conjuntos difusos de salida con los pesos **w**, dando como resultado la altura de los conjuntos difusos correspondientes, es decir $\mathbf{s}^*$ será la altura de todos.

$$s^* = \frac{\sum_{i=1}^{n} s^{-1} w_i}{\sum_{i=1}^{n} w_i} \tag{3.33}$$

- Defuzificador de centro de gravedad

  Para este defuzificador la variable de salida $\mathbf{s}^*$ será el valor numérico real producto de calcular el centro del área cubierta por la función de membresía $\mu(\mathbf{s_A})$ en el conjunto de difuso de salida **A**.

$$s^* = \frac{\int_{altura(A)} \mu_A(s) s ds}{\int_{altura(A)} \mu_A(s) ds} \tag{3.34}$$

3.5.5. Base de reglas difusas

La base de reglas son la forma en que el sistema difuso almacena el conocimiento linguístico que le permiten resolver el problema para el cual ha sido diseñado. La estructura de estas reglas es del tipo IF-THEN, como se muestra en el siguiente diagrama.

$$\text{IF } \underbrace{(\text{la entrada es pequeña})}_{Antecedentes} \text{ THEN } \underbrace{(\text{la salida es grande})}_{Consecuente}$$

En el sistema difuso tipo Mamdani del que se hablara más adelante tanto el antecedente como el consecuente de las reglas están dados por expresiones lingüísticas.

El modelo formal de las reglas difusas esta dado por la siguiente expresión:

$$\text{IF } a_1 \text{ es } A_1 \text{ and } \ldots \text{ and } x_n \text{ es } A_n \text{ THEN z=} B_1 \tag{3.35}$$

En donde:

- **a** es el consecuente, o variables lingüísticas de entrada $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n} \varepsilon \mathbf{U})$.
- $\mathbf{A_1} \ldots \mathbf{A_n}$ y $\mathbf{B_1}$ son conjuntos difusos en $\mathbf{U} \varepsilon \mathbf{R}$ y $\mathbf{V} \varepsilon \mathbf{R}$, siendo **R** una relación discreta que puede definirse por una función típica que asigna **1** a cada combinación de elementos de universo que pertenecen a la relación y **0** a los que no pertenecen.
- $\mathbf{z} \varepsilon \mathbf{V}$ son las variables lingüísticas de salida
- **U** es el conjunto universal de entrada
- **V** es el conjunto universal de salida

$$R(A_1, A_2, \ldots, A_n) = \begin{cases} 1 & \text{si } (A_1, A_2, \ldots, A_n) \varepsilon R \\ 0 & \text{si } (A_1, A_2, \ldots, A_n) \not\varepsilon R \end{cases} \tag{3.36}$$

"En una maquina de inferencia difusa los principios de lógica difusa se utilizan para combinar las reglas difusas IF-THEN de la base de reglas difusas en una relación de un conjunto difuso **A** *en* **U** *a un conjunto* **B** *en* **V**. *Una regla difusa IF-THEN es interpretada como una relación difusa en el espacio entrada-salida* **UxV**." [García Cabrera 07].

3.5.6. Operaciones básicas de conjuntos difusos

Sean **A** y **B** conjuntos difusos:

Unión

De la unión de dos conjuntos **A** y **B**, genera un tercer conjunto difuso **C**, el cual está formado por los máximos valores que resultan de comparar, el grado de membresía de cada uno de los elementos del conjunto difuso **A** con los del conjunto difuso **B**. La unión está dada por:

$$C = A \cup B$$

Donde:

$$\mu_C(x) = \mu_A(x) \vee \mu_B(x) = max[\mu_A(x), \mu_B(x)]; x \in U \tag{3.37}$$

Intersección

De la intersección de dos conjuntos difusos **A** y **B**, genera un tercer conjunto difuso **C**, el cual está formado por los mínimos valores que resultan de comparar, el grado de membresía de cada uno de los elementos del conjunto difuso **A** con los del conjunto difuso **B**. La intersección está dada por:

$$C = A \cap B$$

Donde:

$$\mu_C(x) = \mu_A(x) \wedge \mu_B(x) = min[\mu_A(x), \mu_B(x)]; x \in U \tag{3.38}$$

Complemento

Dado que el valor máximo de grado de membresía que se puede asociar a una función de membresía es 1, el completo esta dado por la diferencia del grado de membresía con 1 y se define de la siguiente forma:

$$A(\neg A)$$

Donde:

$$\mu_{\neg A}(x) = 1 - \mu_A(x) \quad \forall x \varepsilon U \tag{3.39}$$

Complemento relativo

El complemento relativo de dos conjuntos difusos **A** y **B**, esta dada por los elementos que están en **B** pero que no están en **A** y se expresa de la siguiente forma:

$$\mu_{B-A}(x) = \mu_B(x) - \mu_A(x); \quad (\mu_B(x) \geq \mu_A(x)) \tag{3.40}$$

3.5.7. Módelo de inferencia de Mamdani

El método de Mamdani o también conocido como método de inferencia min-max, es uno de los métodos más utilizados por su eficiencia como función de implicación.

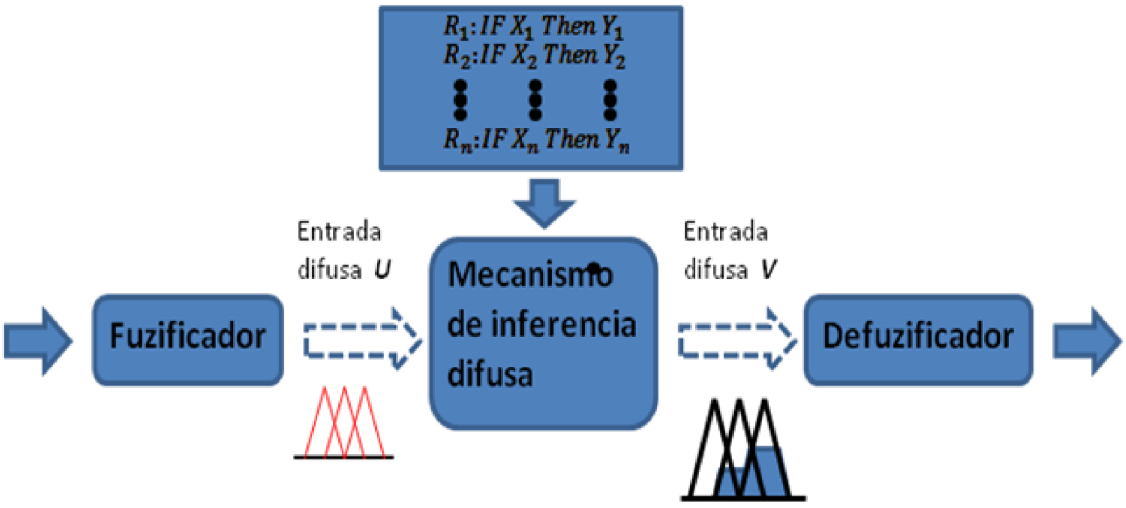


Figura 3.17: Configuración básica de Mamdani

A continuación se muestra un esquema que describe el funcionamiento del método de Mamdani.

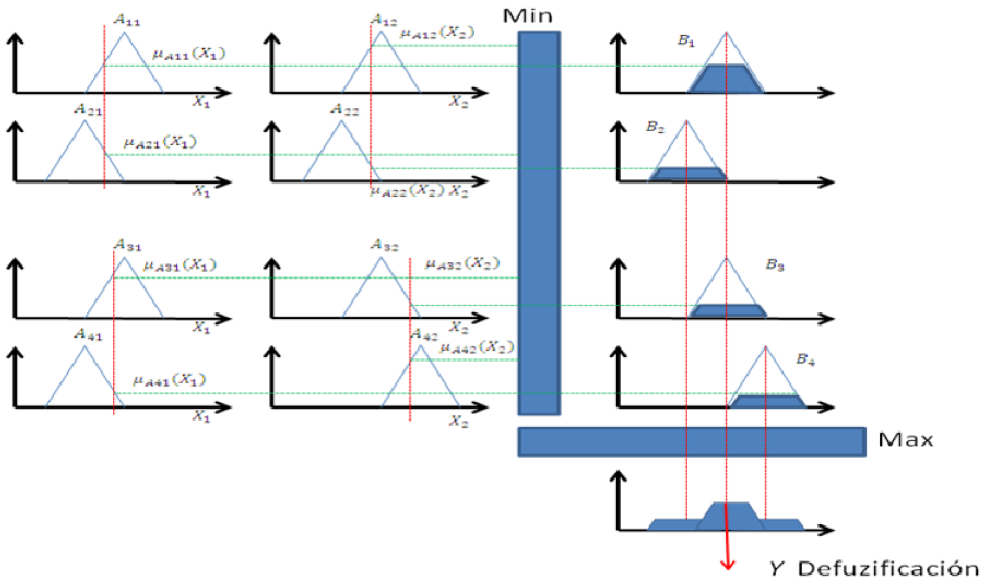


Figura 3.18: Esquema del proceso de defusificación de Mamdani

3.5.8. Distancia de Mahalanobis

Prasanta Chandra Mahalanobis (29 de junio de 1893 – 28 de junio de 1972), [Mahalanobis, 2008], científico de origen indio en el área de estadística aplicada. Introduce una medida de distancia a la estadística en 1936, conocida como la distancia de Mahalanobis, la cual es su más importante contribución a la estadística. Es útil para determinar la similitud entre dos variables aleatorias multidimensionales. Su principal característica es que tiene en cuenta la correlación que existe entre dos o más variables aleatorias, lo cual la diferencia de la distancia euclidiana. La distancia de Mahalanobis entre dos variables aleatorias con la misma distribución de probabilidad $\tilde{\mathbf{x}}$ y $\tilde{\mathbf{y}}$ con matriz de covarianza $\sum$ se define como:

$$d_m(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T \sum^{-1} (\vec{x}-\vec{y})} \tag{3.41}$$

Haciendo énfasis en la matriz de covarianza que para este caso de distancia es la principal característica, se presenta un análisis de la covarianza y sus propiedades.

La covarianza es una medida descriptiva que sirve para medir o cuantificar la similitud

que existe entre dos variables aleatorias, y se define como:

$$x_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{3.42}$$

- Si $\mathbf{S_{xy} > 0}$ existe dependencia directa, a grandes valores de **x** corresponden grandes valores de **y**.
- Si $\mathbf{S_{xy} = 0}$ se interpreta como la no existencia de una relación lineal entre las dos variables aleatorias.
- Si $\mathbf{S_{xy} < 0}$ hay dependencia inversa o negativa, a grandes valores de **x** corresponden pequeños valores de **y**.

3.6. Visión por computadora

Los sentidos que acompañan al sistema humano reportados en la literatura son cinco; olfato, oído, gusto, tacto y visión, los cuales interactúan entre sí para dotar de información sensorial sobre el mundo a los seres humanos. La ausencia de cualquiera de ellos repercute de forma drástica en la información enviada al cerebro donde es interpretada. La visión es al parecer el sentido que más información provee al cerebro acerca del entorno, los ojos son un complejo sistema bilógico dotado de millones de sensores, como los foto-receptores que permiten detectar luz, útil para detectar movimiento y diferenciar entre color y forma, además de dotar al cerebro con la enorme habilidad para interpretar la conducta de los demás cuerpos de alrededor, animados o inanimados. Otros parámetros como el cálculo aproximado de distancias y tamaño son cualidades que el cerebro humano es capaz de interpretar de forma ambigua pero altamente aproximada a lo real, dado que sin conocer la distancia real a un objeto en metros, centímetros o aun más exacto en milímetros, tareas como manipular objetos, moverlos o evitarlos se realizan de forma perfecta, claro está, después de un factor clave como lo es el entrenamiento que adquirimos desde que somos niños. Sin embargo el sistema de visión tiene una serie de limitaciones ante ciertas situaciones, como lo confuso que puede ser interpretar información visual incompleta que pueda hacernos confundir objetos, formas, colores, etc.

Dado que el sentido de la visión en seres humanos proporciona aproximadamente el 50 % de la información necesaria para resolver problemas de orientación, equilibrio y el reconocimiento de objetos y formas, es entonces que se estudia su estructura biológica para tratar de emular sus funciones a través de mecanismos tecnológicos como cámaras para la adquisición de imágenes de color y equipos de computo para su descomposición y análisis. El reconocimiento de patrones y visión por computadora son las propuestas para atender esta problemática.

La habilidad de seguir objetos dentro de imágenes es posible si se realiza un análisis de los rasgos geométricos, *"Si se desea determinar la identidad de tales objetos es mejor utilizar rasgos mesurables que permitan aplicar vectores descriptivos."*[H. Sossa, 08].

3.6.1. Rasgos descriptores

Los rangos son mediciones que se obtienen de los pixeles que componen a un objeto dentro de una imagen y a su frontera con respecto al fondo o demás objetos, vértices, esquinas y puntos característicos del esqueleto de un objeto como terminales y triadas también son parte de las medicines. [H. Sossa, 08]. Los rasgos pueden clasificarse en dos tipo:

Geométricos:

Describen propiedades geométricas que pueden ser obtenidas a partir de la región y contorno del objeto como el área y el perímetro.

Topológicos:

Describen propiedades que tiene que ver con la estructura del objeto, el número de hoyos y el número de Euler, pueden obtenerse de la región de un objeto o del esqueleto de su región.

3.6.2. Rasgos basados en momentos geométricos

Siguiendo con el análisis de rasgos basados en momentos según el Dr. Juan Humberto Sossa, los momentos geométricos han sido utilizados para el análisis de formas en el

reconocimiento de patrones con éxito, “para el caso de una función continua bidimensional $\mathbf{f}(\mathbf{x},\mathbf{y})$, la cual se supone ser continua por partes y con soporte compacto, el momento de orden $(\mathbf{p}+\mathbf{q})$ se define como:” [H. Sossa, 08].

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p x^q f(x,y) dxdy \tag{3.43}$$

para $\mathbf{p},\mathbf{q} = \mathbf{0},\mathbf{1},\mathbf{2},\ldots$ Para el caso de una imagen digital, la integral se sustituye por la sumatoria, de tal forma que:

$$m_{pq} = \sum_x \sum_y x^p x^q f(x,y) dxdy \tag{3.44}$$

Para el caso de una imagen binaria 0,1, la ecuación está dada por:

$$m_{pq} = \sum_{xy} \sum_{\varepsilon R} x^p x^q \tag{3.45}$$

Donde **R** denota la región del objeto. Normalmente solo se usan los momentos de orden tres, es decir los momentos para los cuales $(\mathbf{p}+\mathbf{q}) \geq \mathbf{3}$, ya que representan varias propiedades fundamentales. Su aplicabilidad en imágenes consiste en considerar los valores de los momentos de una distribución binaria contigua, es decir la silueta de un objeto segmentado.

Momentos de orden cero

El momento de orden cero $\mathbf{m_{00}}$ representa la masa total de una función de distribución, ya que **m00** coincide con el área geométrica de un objeto segmentado, es decir el número de pixeles $(\mathbf{x},\mathbf{y})\varepsilon\mathbf{R}$.

Momentos de orden uno

$\mathbf{m_{10}}$ y $\mathbf{m_{01}}$ son usados para localizar el centro de masa del objeto representado por el sistema de coordenadas $(\overline{\mathbf{x}},\overline{\mathbf{y}})$, donde $\mathbf{x} = \overline{\mathbf{x}}$ y $\mathbf{y} = \overline{\mathbf{y}}$ representan las líneas en donde masa total del objeto puede ser concentrada sin cambios en los momentos de orden uno a lo largo de los ejes **x** y **y**.

$$\bar{x} = \frac{m_{10}}{m_{00}} \tag{3.46a}$$

$$\bar{y} = \frac{m_{01}}{m_{00}} \tag{3.46b}$$

Momentos de orden dos

Conocidos como momentos de inercia $\{\mathbf{m_{02}}, \mathbf{m_{20}}, \mathbf{m_{11}}\}$, mediante los cuales se pueden calcular otras características útiles. Los momentos de orden dos son usados para determinar los ejes alrededor de los cuales se localizan los momentos segundos, mínimo y máximo, eje mayor y menos, cuya orientación ϕ viene dada por:

$$\phi = \frac{1}{2} tan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{3.47}$$

En donde: ϕ s el ángulo del eje principal más cercano al eje x cuyo valor se encuentra en el rango $-\pi 4 \leq \phi 4 \leq \mu$ Se refiere a los momentos centrales, cuyos valores son calculados a partir de considerar que el centro de masa de un objeto coincida con el campo de vista, es decir $(\overline{\mathbf{x}} = \mathbf{0})$ y $(\overline{\mathbf{y}} = \mathbf{0})$. El ángulo de cualquiera de los dos ejes puede ser determinado a partir de los momentos centrales $\mu\mathbf{11}$ y ($\mu_{\mathbf{20}}$- $\mu_{\mathbf{02}}$). Otra propiedad que se determina a partir de los momentos de orden dos en el radio de giro de un objeto. El radio de giro alrededor de los ejes **x** y **y** están dados por:

$$GR_x = \sqrt{\frac{m_{20}}{m_{00}}} \tag{3.48a}$$

$$GR_y = \sqrt{\frac{m_{02}}{m_{00}}} \tag{3.48b}$$

El radio de giro de un objeto en términos de momentos centrales de orden dos posee la propiedad de ser invariante a rotaciones y está dado por:

$$RG = \sqrt{\frac{\mu_{20} + \mu_{02}}{\mu_{00}}} \tag{3.49}$$

Momentos de orden tres

Los momentos de orden tres $\{\mu_{\mathbf{30}}, \mu_{\mathbf{03}}\}$ se refieren al sesgo proyectivo del objeto, el cual es una medida estadística del grado de distribución de la desviación de la simetría alrededor de la medida. Las proyecciones de un objeto sobre los ejes **x** y **y** cuyo signo nos indican hacia qué lado se encuentra sesgada con respecto a un eje, este se basa en el hecho de que una rotación de $\mathbf{180}\circ$ cambia el signo del sesgo de la proyección con respecto a cualquiera de los

ejes y vienen dadas como:

$$Sk_x = \frac{\mu_{30}}{\mu_{20}^{3/2}} \tag{3.50a}$$

$$Sk_y = \frac{\mu_{03}}{\mu_{02}^{3/2}} \tag{3.50b}$$

Invariantes a traslaciones: Momentos centrales

Los momentos centrales $\mu_{\mathbf{pq}}$ se expresan como:

$$\mu_{00} = m_{00} \,.\, \mu_{11} = m_{11} - ym_{10} \tag{3.51a}$$

$$\mu_{10} = 0 \,.\, \mu_{03} = m_{30} - 3xm_{20} + 2x^2 m_{10} \tag{3.51b}$$

$$\mu_{01} = 0 \,.\, \mu_{12} = m_{12} - 2ym_{11} - xm_{02} + 2y^2 m_{10} \tag{3.51c}$$

$$\mu_{20} = m_{20} - xm_{10} \,.\, \mu_{21} = m_{21} - 2xm_{11} - ym_{20} + 2x^2 m_{01} \tag{3.51d}$$

$$\mu_{20} = m_{02} - ym_{01} \,.\, \mu_{03} = m_{03} - 3ym_{02} + 2y^2 m_{01} \tag{3.51e}$$

Invariantes a rotaciones: Los invariantes de Hu

Los invariantes a transformaciones de este tipo se obtienen al sustituir la ecuación (3.52) en la ecuación (3.43).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{pmatrix} cos\theta & -sen\theta \\ sen\theta & cos\theta \end{pmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.52}$$

Los primeros cuatro invariantes a traslaciones y rotaciones de Hu [Hu 1962] están dados por las siguientes ecuaciones:

$$\phi_1 = \mu_{20} + \mu_{02} \tag{3.53a}$$

$$\phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \tag{3.53b}$$

$$\phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \tag{3.53c}$$

$$\phi_4 = (\mu_{30} - \mu_{12})^2 + (\mu_{21} - \mu_{03})^2 \tag{3.53d}$$

Invariantes a cambios de escala: Momentos centrales normalizados

Los invariantes a cambios de escala se obtienen al dividir cada momento por un factor de normalización que cancele el efecto de escalamiento.

Considerando que los cambios de escala son causados por transformaciones de coordenadas de la forma:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.54}$$

Sea $\mathbf{f}'(\mathbf{x}',\mathbf{y}')$ la imagen $\mathbf{f}(\mathbf{x},\mathbf{y})$ después de un escalamiento en cada eje por un factor α a través de la aplicación dada por la ecuación (3.54), se observa que $\mathbf{f}'(\mathbf{x}',\mathbf{y}') = \mathbf{f}(\alpha\mathbf{x},\alpha\mathbf{y}) = \alpha\mathbf{f}(\mathbf{x},\mathbf{y})$ y $\mathbf{x}' = \alpha\mathbf{x}, \mathbf{y}' = \alpha\mathbf{y}$ en consecuencia:

$$m'_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x'^p y'^q f'(x',y')dx'dy' \tag{3.55a}$$

$$m'_{pq} = \alpha^{p+q+2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy \tag{3.55b}$$

De tal forma que $\mathbf{m'_{pq}} = \alpha^{\mathbf{p+q+2}}\mathbf{m_{pq}}$, de igual manera $\mu'_{\mathbf{pq}} = \alpha^{\mathbf{p+q+2}}\mu_{\mathbf{pq}}$. Para el momento de orden cero en particular se puede ver que $\mu'_{\mathbf{00}} = \alpha^{\mathbf{2}}\mu_{\mathbf{00}}$, puede verse que:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{3.56a}$$

$$\gamma = \frac{p+q}{2} + 1, \text{ p+q = 2,3,...} \tag{3.56b}$$

Invariantes a traslaciones, rotaciones y cambios de escala

Reemplazando los momentos centrales en las expresiones para invariantes a rotaciones por los invariantes a escala $\eta_{\mathbf{pq}}$ resultan los siete invariantes a traslaciones y cambios de

escala de Hu.

$$\phi_1 = \eta_{20} + \eta_{02} \tag{3.57a}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{3.57b}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{3.57c}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{3.57d}$$

$$\begin{aligned}\phi_5 =& (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ &+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \tag{3.57e}$$

$$\begin{aligned}\phi_6 =& (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})\right] \\ &+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})\end{aligned} \tag{3.57f}$$

$$\begin{aligned}\phi_7 =& (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ &+ (3\eta_{21} - \eta_{30})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \tag{3.57g}$$

Capítulo 4

Metodología propuesta

En este capítulo se presenta a detalle la metodología desarrollada para la navegación y manipulación de objetos, los procedimientos y conceptos más relevantes bajo los que se construyeron los algoritmos que combinan las técnicas del área de visión por computadora, reconocimiento de patrones y control moderno, para la ejecución de las acciones dirigidas a cumplir con los objetivos de esta tesis.

4.1. Descripción general del sistema

La metodología propuesta, incluye el desarrollo de una plataforma de software bajo el lenguaje de programación java, que permite adoptar conceptos como el de multiprocesamiento, concepto conveniente en la implementación de los algoritmos de visión y control en robótica móvil. Mediante el uso de técnicas de aprendizaje y de clasificación como redes neuronales, específicamente, perceptrón multicapa en combinación con métodos de tratamiento digital de imágenes para el reconocimiento de formas como lo es el cálculo de los invariantes de Hu, se dota al software de una herramienta visual para evaluar la efectividad del seguimiento de una trayectoria, apuntado hacia objetos de interés estáticos, usando filtros como el reconocimiento de formas y discriminación por color a partir del aislamiento de una muestra de color en el plano RGB, que servirá como punto central de un matiz de color al cual se calcula la distancia propuesta por Mahalanobis, para determinar si existe una relación

entre esa muestra (píxel) con todos los demás pixeles de la imagen y de esta forma se reduzca el campo de búsqueda de formas y por consiguiente el tiempo de reconocimiento, así como del uso de técnicas de control difuso para dotar al controlador con la capacidad para tratar situaciones de incertidumbre comunes en la robótica móvil y haciendo caso a los trabajos previos, se intenta marcar algunas diferencias significativas en el rendimiento o la efectividad de las tareas escogidas como métodos de experimentación.

El uso de herramientas como el modulo “fuzzy” del toolbox de Matlab, se usa para modelar el conjunto de reglas difusas de la forma IF-THEN, para ser aplicadas en una configuración de controlador difuso del tipo Mamdani, con el fin de que sean probadas antes de ser llevadas a la programación. Cabe mencionar que se hizo un esfuerzo por diseñar algoritmos de tratamiento digital de imágenes en forma de vector hasta donde fue posible.

Aunque en este trabajo se menciona el modelo diferencial del robot para entender la naturaleza de sus movimientos, que son básicamente tres. Estos tres movimientos describen trayectorias en línea recta, circular y de rotación sobre el centro de masa del robot, el controlador fue diseñado con una estrategia que ignora algunos parámetros de ese modelo, pues no es posible obtenerlos bajo las condiciones de abstracción del entorno, de las cuales la orientación y distancia hacia el objetivo están puramente relacionados con la imagen que se procesa y poco tienen que ver con las medidas reales que guarda el cuerpo rígido del móvil con respecto al objetivo. Para satisfacer la capacidad del robot de realizar esos tres tipos de movimientos, se propusieron 5 variables lingüísticas, de las cuales 2 se ocupan de la orientación y distancia, la distancia es propuesta en la fase de entrenamiento y esta basada en el tamaño del objeto segmentado despues de aplicar el filtro de color y forma, y un error de orientación ver figura (4.3), que se corrige llevando el centro de masa del objetivo, al eje medio vertical de la componente ancho de la imagen, sin importar la altura. Las tres restantes se ocupan de las salidas del sistema, las cuales se encargaran de variar las velocidades del robot para alcanzar los parámetros de distancia y orientación, que se requieren para la ejecución.

Descripción del diagrama de flujo del sistema

Antes de continuar con la descripción de cada uno de los módulos del sistema, se hará una descripción del diagrama de flujo del sistema en lazo cerrado, ver figura (4.21).

- Al inicio del sistema se observan dos módulos que garantizan la comunicación bidireccional con el robot, el primer módulo se refiere a la comunicación con el puerto (com) asignado a la tarjeta de vídeo que recibe una señal de vídeo proveniente de un cable RCA conectado a un dispositivo inalámbrico conectado a la cámara del robot khepera II vía wireless, el segundo medio de comunicación se refiere al dispositivo bluetooth mediante el cual se mandan los comandos de control con los servomotores y control del elemento prensil. La comunicación con el puerto com se realiza mediante la versión del Comm API (Java(tm) Communications API) para Windows, vease figura (4.1), está formada por tres archivos:

  - win32com.dll en
  - comm.jar
  - javax.comm.properties

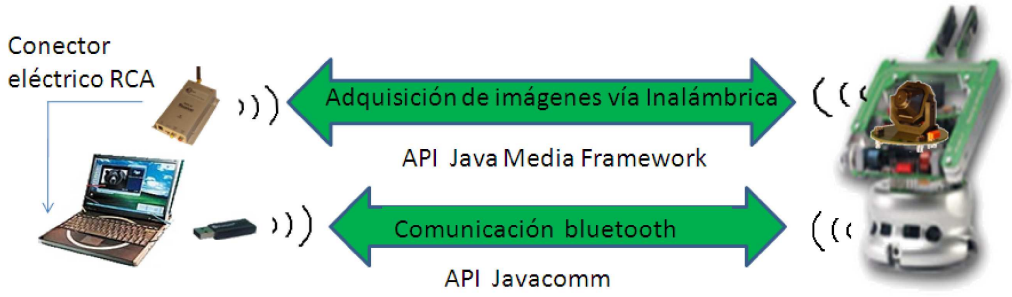


Figura 4.1: Esquema de comunicación inalámbrica

- El siguiente módulo implementa la API (JMF)Java Frame Work, con la cual se despliega una pantalla de vídeo continuo.

- El módulo de captura multi-hilo de imágenes realiza una captura infinita de frames de imagen que toma de la secuencia de vídeo y actualiza una variable única que contiene a la imagen actual en forma vectorizada.
- El módulo de giro continuo hace girar al robot sobre la rueda derecha en busca del objeto buscado.
- El módulo del filtro de color, binariza la imagen vectorizada tomando como umbral una muestra de color que se obtiene durante un proceso de entrenamiento.
- EL módulo de etiquetado de componentes conectadas, etiqueta a cada región conectada con etiquetas numéricas incrementables a cada nueva región conectada encontrada ver figura (4.2).

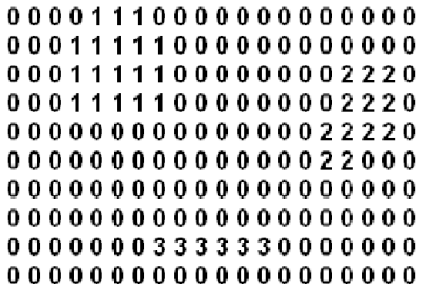


Figura 4.2: Ejemplo de imagen con regiones etiquetadas

- Se verifica que por lo menos exista una etiqueta, es decir al menos un objeto del color buscado en la imagen, si esto sucede se pasa al siguiente módulo, si no es así si se regresa al modulo de giro continuo para realizar un nuevo análisis de imagen.
- El módulo de extracción del vector de características de Hu, calcula un vector de características por cada etiqueta encontrada, es decir por cada objeto, obteniendo así un vector de vectores de Hu.
- El siguiente módulo se trata de un ciclo que somete los vectores antes calculados a una red neuronal perceptrón multicapa y obtener así un vector de aproximación de clases compuesto por números reales.

- El siguiente módulo toma el vector de aproximación y elige al elemento **x** de menor distancia al valor **y** buscado como clase, usando el algoritmo de mínimas distancias.

- El siguiente módulo calcula una diferencia entre el valor **x** antes calculado y el valor **y** buscado, si el valor resultante es menor o igual a **0.001**, entonces se considera que el objeto se encontró con éxito, si no es así entonces se realiza el procesamiento de una nueva imagen, siempre y cuando un contador que se encarga de llevar el control de imágenes en las que no se encontró el objeto buscado sea mayor a **5**, de esta forma si el robot pierde al objeto por un par de segundos no cambiara su ruta a menos que la no localización del objeto rebase un tiempo prolongado.

- El módulo de controlador difuso, calcula un error de orientación (**O**) y tamaño del objeto segmentado en la imagen, para el sistema el tamaño del objeto en píxeles se considera la distancia de operación optima del objeto (**D**). Conociendo la distancia y el error de orientación se realizara el cálculo de las velocidades de cada rueda haciendo uso del módelo difuso diseñado.

- El último módulo inicializa el contador que determina cuántas veces se ha fallado en la búsqueda del objeto ya que se supone que estamos saliendo de un proceso exitoso de búsqueda y control.

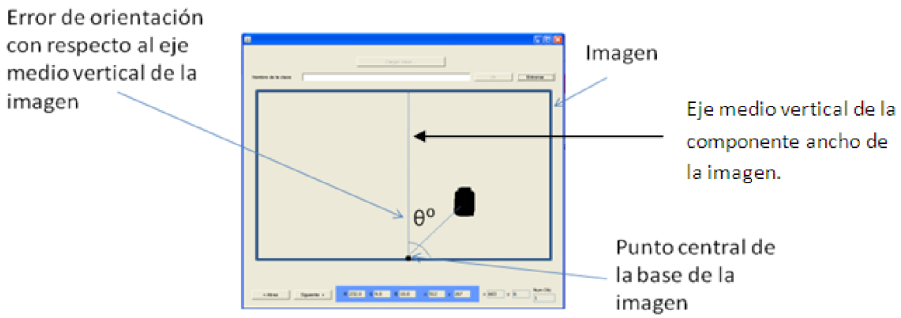


Figura 4.3: Error de orientación con respecto al eje medio vertical de la imagen

4.1.1. Fase de entrenamiento

En este apartado se plantea el mecanismo de entrenamiento utilizado para el seguimiento de puntos de referencia, en este caso uitlizamos el centro geométrico de formas seguidas a partir de una imagen, a través de un módelo de clasificación previamente entrenado, como lo es una red neuronal tipo Backpropagation. Esta red es un perceptrón multicapa con diferentes configuraciones de números de capas y neuronas por capa; cabe mencionar que basta con que la red sea capaz de clasificar objetos a partir de un vector de dos elementos. Estos elementos describen rasgos geométricos del objeto cuyos parámetros corresponden a los dos primeros invariantes a traslaciones, rotaciones y cambios de escala. Estos rasgos están descritos en el sistema de ecuaciones ((3.57a)) y ((3.57b)) del capítulo 3.

Un aspecto importante para solucionar el problema de proximidad al objeto, consiste en calcular de la primera imagen de entrenamiento el momento de orden cero $\mathbf{m_0 0}$ descrito en el punto 3.6.2 del capítulo 3, cuyo valor representa la masa total del objeto segmentado, dado en pixeles. El problema de proximidad al objeto es equivalente a dar al sistema un dato que le permita definir cuando el robot se encuentro cerca o muy cerca del objeto. Por lo anterior se hace necesario que la primera fotografía del conjunto de entrenamiento de cada clase de objeto de forma geométrica sea tomada con el objeto colocado frente al robot, de forma tal que sea a esa distancia manipulable y se encuentre dentro del rengo de visión de la cámara montada sobre el robot véase figura(4.4).

Fase de extracción del vector de características

Esta es la primera fase del proceso de entrenamiento, para la cual se requiere del análisis de un conjunto de imágenes de un objeto común con características geométricas idénticas y color homogéneo, las cuales pasan por un proceso de binarización y etiquetado antes de ser expuestas a la red neuronal que realizará la clasificación de sus formas.

Figura 4.4: Adquisición de la imagen en rango de operación

Proceso de binarización

Dada una imagen **y** tomada de un conjunto de imágenes, representada por un vector de tamaño **n** = **hxw**, en donde **h** se refiere al alto y **w** al ancho de la imagen, se considera que existen **n** píxeles, cada uno con un valor entero asociado a cada uno de los colores primarios, rojo, verde y azul. Es decir una configuración RGB. A partir de un pixel **x** = **RGB**, tomado como muestra de color del área del objeto de interés en una imagen que denominamos **X**$_{(}$**entrenamiento**$)$, se calcula la distancia de Mahalanobis, con cada uno de los **n** píxeles de la imagen analizada, buscando con eso un valor real que mide la similitud de los tres niveles de color. Aquellos cuya distancia al píxel de muestra que no rebasen un umbral establecido bajo un criterio experimental fijado en **0**.**12**, por Pablo Roncagliolo [Roncagliolo,01], bajo condiciones de iluminación controladas, serán neutralizados o puestos a **0**, por el contrario aquellos que si lo rebasen o igualen serán conservados con el valor numérico**1**para el siguiente nivel de reconocimiento. A continuación se muestra un esquema de este proceso.

1. Se toma una muestra de color que pertenece al objeto buscado, vease figura (4.5).

2. Se realiza la binarización de imágenes de color mediante el filtro de color de distancia de Mahalanobis.

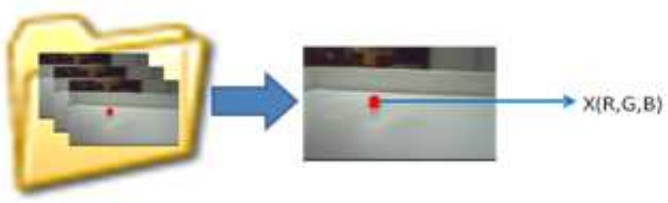


Figura 4.5: Extracción de un píxel de muestra en un plano RGB

Retomando lo expuesto en el punto 3.5.8 del capítulo anterior en donde se establece que la covarianza **cov**$(\mathbf{x},\mathbf{y})$ en la ecuación (4.1), corresponde a una medición estadística que indica la fuerza con la que se relacionan 2 o más variables, en caso en particular la covarianza entre dos o más píxeles con **N** número de muestras esta dado por:

$$cov(x,y) = \sum_{i=1}^{N} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N} \tag{4.1}$$

"La distancia de Mahalanobis permite calcular distancias entre colores ponderadas por la importancia del Matiz (color principal) y la varianza en cada componente (R, G y B). Incluyendo la inversa de la matriz de covarianza de un conjunto de píxeles representativos, se puede lograr esta "ponderación". La distancia de Mahalanobis es igual a la distancia Euclidiana si la matriz de covarianza C corresponde a la matriz identidad". [Roncagliolo,01].

	R	*G*	*B*
R	1	0	0
G	0	1	0
B	0	0	1

Cuadro 4.1: Matriz identidad

$$d_m(\vec{x}_i,\vec{y}_i) = \sqrt{(\vec{x}-\vec{y}_i)^T C^{-1}(\vec{x}-\vec{y}_i)} \tag{4.2}$$

Si se analiza con cuidado la ecuación (4.2), podrá comprobarse que para el caso de tomar solo una muestra de color se obtiene el mismo resultado si obviamos la matriz identidad

y manejamos al primer vector definido por la diferencia entre los vectores $\tilde{\mathbf{x}}$ y $\tilde{\mathbf{y}}_\mathbf{i}$ como un vector no transpuesto. Ya que el resultado de multiplicar $(\vec{x}-\vec{y})^T C^{-1}$ nos da como resultado un vector idéntico a $(\vec{x}-\vec{y})$. Por lo que obtendríamos el mismo resultado aplicando para este caso la distancia euclidiana, como lo indica [Pablo Roncagliolo B].

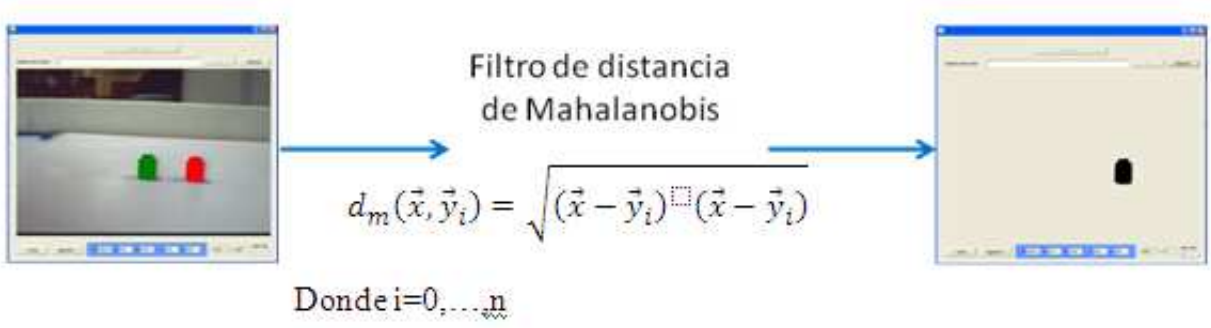


Figura 4.6: Binarización de imágenes usando la distancia de Mahalanobis

4.1.2. Proceso de etiquetado de componentes conectadas

El proceso de etiquetado se realiza mediante el recorrido de una matriz compuesta de **0** y **1** que representa a la imagen resultante del filtro de color, con una máscara en forma de escuadra de elementos como se muestra en la figura (4.7). El recorrido se hace comenzando

Figura 4.7: Mascara de recorrido descendente

en la parte superior izquierda, avanzado primero hacia la derecha hasta encontrar el fin de la línea y después nuevamente con la segunda línea hasta haber recorrido por completo la imagen, el segundo recorrido se hace de la misma forma pero de manera invertida, es decir, comenzando de la esquina inferior derecha avanzando a la izquierda y una vez alcanzando el inicio de la línea se continua con la línea anterior hasta terminar en el primer pixel de la imagen, con una máscara similar a la anterior pero invertida vease figura(4.8). A continuación

Figura 4.8: Mascara de recorrido ascendente

se describen los algoritmos que hacen uso de las mascaras mostradas en las figuras (4.7) y (4.8) antes mostradas. La descripción del algoritmo se hará considerando que se implementa a partir de la segunda línea de la imagen binarizada para que sea más claro el proceder de la máscara, el píxel analizado es el que se encuentra marcado en rojo y sus vecinos en amarillo vease figura () y figura ().

```
Si (Pixel analizado) > 0
{
    Se analizan los vecinos
     {
        Si (el mínimo de los vecinos > 0) entonces
            Pixel analizado es igual al mínimo
        Si (Todos los vecinos son cero) entonces
            Pixel analizado toma el valor de una nueva etiqueta
     }

}
```

Figura 4.9: Algoritmo de rrecorrido descendente

```
Si (Pixel analizado) > 0
{
    Se analizan los vecinos
     {
        Si (el máximo de los vecinos > 0) entonces
            Pixel analizado es igual al máximo
        Si (Todos los vecinos son cero) entonces
            Pixel analizado toma el valor de una nueva etiqueta
     }

}
```

Figura 4.10: Algoritmo de rrecorrido ascendente

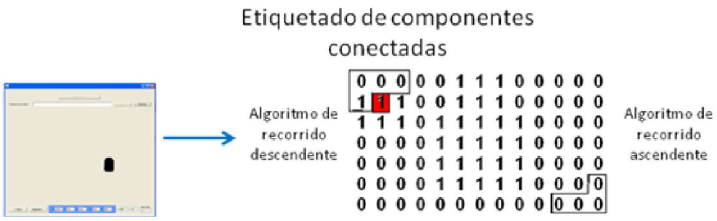


Figura 4.11: Proceso de etiquetado de componentes conectadas con doble recorrido en diagonal, descendente y ascendente,[Jimenez,2008]

Extracción del vector de características de los invariantes de Hu

Debido a que los invariantes de Hu son invariantes a traslaciones, rotaciones, cambio de escala y desplazamiento, se eligieron como método de segmentación de imágenes, ya que aunque la navegación del robot puede hacer que la perspectiva de los objetos cambie en alguno de esos movimientos. Se calcularon los 7 invariantes para contar con la mayor información posible del los objetos que se buscan, sin embargo sólo se utilizaron los dos primeros para la clasificación.

En esta fase de entrenamiento es necesario contar con todo el conjunto de vectores descriptivos cuyo número es igual al número de imágenes utilizadas para el entrenamiento.

- Matriz de momentos normales

1. El momento de orden cero $\mathbf{m_{00}}$, representa la masa total de la imagen o también conocida como área geométrica. Específicamente eN la primera imagen se usa como marcador de distancia óptima en rango de operación, para lo que es necesario que la primera imagen del conjunto de imágenes de entrenamiento, sea tomada cuando el objeto al que se extraen los invariantes este dentro del rango de visión y dentro del rango de operación de las pinzas.

2. Los momentos de orden uno $\mathbf{m_{10}}$ y $\mathbf{m_{01}}$, los cuales son usados para localizar el centro de masa del objeto. Este centro de masa será usado en este trabajo para

identificar el grado de error de dirección hacia el objeto en rango de visión.

$$\bar{x} = \frac{m_{10}}{m_{00}} \tag{4.3a}$$

$$\bar{y} = \frac{m_{01}}{m_{00}} \tag{4.3b}$$

- Matriz de momentos centrales

  Los momentos centrales denotados como μ_{pq}, son equivalentes a los momentos normales de un objeto que se ha desplazado, por lo que se consideran invariantes a traslaciones de la imagen. Los cuales serán necesarios para calcular los dos primeros invariantes de Hu, que necesitamos para concluir con la extracción de rasgos invariantes.

- Matriz de momentos invariantes de Hu

  Los cuatro primeros invariantes de Hu, los cuales son invariantes ante traslaciones y rotaciones pero sólo los dos primeros son usados para formar el vector de patrones para la clasificación de imágenes de este trabajo.

Implementación de un modelo de red perceptron multicapa (Backpropagation) con función de activación sigmoidal

La configuración de la red debe cumplir con tres características:

Primera

Se refiere a que el número de entradas deben ser dos, ya que son los dos primeros invariantes de Hu, como se vio en el punto anterior. Estos invarinates serán los que conforman el vector de características, y su elección se hizo a prueba y error incluyendo a los siete invariantes.

Segunda

Se refiere a que la salida de la red solo debe ser una neurona, para que el rango que existe entre 0 y 1 sea perfectamente divisible entre el número de objetos que se quieran clasificar

atendiendo así al uso de funciones de activación de tipo sigmoidal, de tal forma que si solo se entrena para un objeto, todo valor cercano a 1 con un grado de error de 0.001 será considerado un objeto similar, si fueran dos, el primer objeto será cercano a 1 y el segundo cercano 0, si fueran 3 objetos, el primer objeto estará cercano a 1 el segundo cercano a 0.5 y el tercero cercano a 0, y así sucesivamente, lo que implica también que entre mayo sea el numero de objetos con el que se entrena se incrementa la posibilidad de confundirlos.

Tercera

El número de capas internas, así como el número de neuronas por capa será elegido de acuerdo al tipo de objetos que se quiera clasificar, algunos necesitaran de más capas, otros con diferencias muy notables necesitarán de menos capas para ser correctamente clasificados, sin embargo es conveniente un análisis de las formas de los objetos que son usados, mediante la graficación de los dos invariantes en un sistema coordenado que refleje que existe suficiente grado de separabilidad entre ellos. Una vez que la red converge, se almacena la matriz de

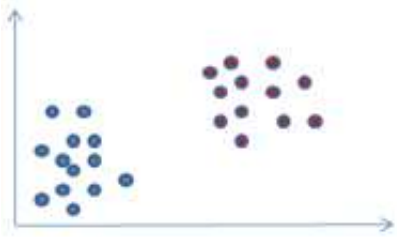


Figura 4.12: Conjunto perfectamente separable

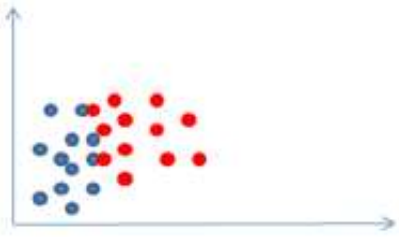


Figura 4.13: Conjunto dificilmente separables

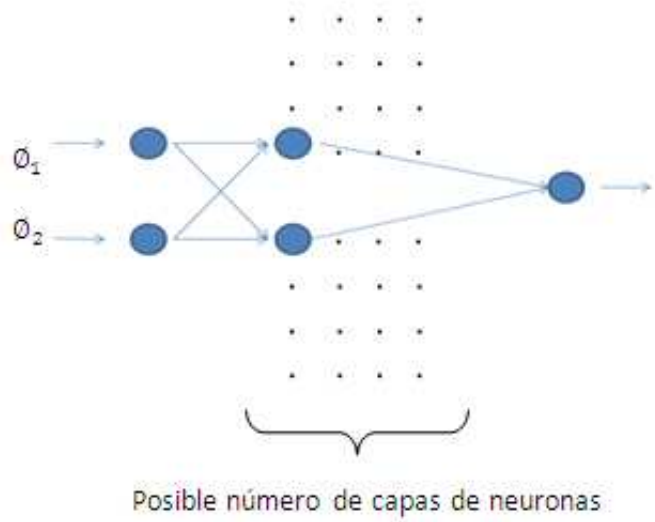


Figura 4.14: Esquema de configuración de capas del perceptrón propuesto

pesos y la matriz de las funciones de activación para la posterior clasificación de los objetos segmentados mediante el uso de los filtros de color y forma antes descritos.

4.1.3. Diseño del controlador difuso tipo Mamdani

Análisis del modelo diferencial del robot para alimentar la base de reglas difusas

Después de hacer un profundo análisis de las capacidades cinemáticas del robot khepera II y del modelo diferencial que lo acompaña, se llegó a la conclusión de que algunos parámetros estaban fuera de nuestras posibilidad de calcular o adquirir, como lo es la distancia real que existe entre el centro geométrico del robot y el punto al que se quiere llevar al robot. Este punto resulta indispensable si se requiere usar el sistema de ecuaciones (3.9a) y (3.9b) del modelo diferencial descrito en el punto 3.3 del capítulo 3. Como se menciono antes, el mecanismo para llevar el robot de un punto a otro es variar en magnitud las velocidades de cada rueda. Para lo que resulta necesario conocer el radio de la circunferencia que debe describir el móvil para llegar al punto deseado, lo que dificulta conocer la velocidad angular que también es requerida por el sistema de ecuaciones del modelo diferencial. La única información que se tienes, el tamaño en píxeles del objeto segmentado y un error de

orientación del centro geométrico del objeto con respecto a un centro imaginario, ubicado en el centro de la última línea de pixeles de la imagen, ver figura (4.3), este error de orientación que poco tiene que ver con el error de orientación real que existe entre el robot y el objeto. A continuación se describe la estrategia para dar solución a esta problemática. Algo que nunca se perdió de vista es que el robot es un cuerpo rígido móvil de tipo diferencial y que debe ser capaz de describir tres tipos de trayectorias que son:

- Trayectoria lineal
- Trayectoria circular
- Rotación sobre su centro de masa

Para cumplir con estos tres tipos de movimiento resulta indispensable asociar al robot una velocidad de traslación y algún factor que altere la velocidad de desplazamiento de alguna de las ruedas o de ambas. El caso más fácil al que realmente podemos asociar un valor de velocidad a cada rueda y al robot en general como un móvil rígido, es cuando el objeto segmentado bajo las condiciones de color y forma se encuentra en el centro del ancho de la imagen, es decir, con un error de orientación de **0**°, ya que éste es el único punto en la imagen que realmente coincide con el error de orientación real con respecto al objeto, cuyo valor será de **90**°, dado en radianes con respecto al eje que une a las dos ruedas y al eje horizontal inferior de la imagen, véase figura (4.15). Para este caso en especifico, las velocidades de

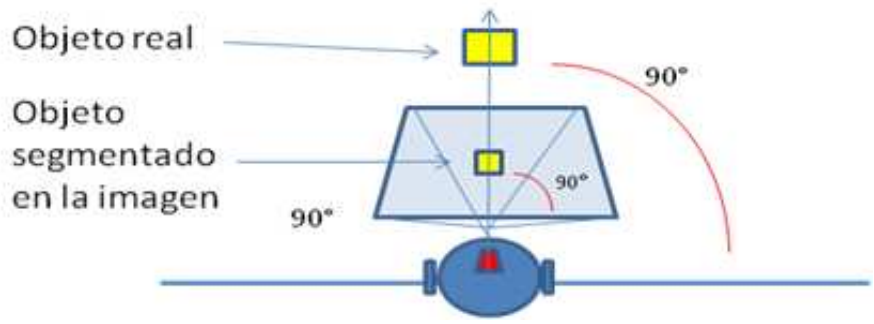


Figura 4.15: Esquema del único punto de orientación real obtenido mediante visión

traslación de cada rueda podrían considerarse de igual magnitud, con lo que se garantizaría

una trayectoria en línea recta. Sin considerar la distancia, las velocidades además de ser igual en magnitud se consideran máximas. Puede verse que con esto se cumple con la capacidad del robot de describir una trayectoria recta.

$$VT_{robot} = V_{max} \tag{4.4a}$$

$$VT_{Ruedaderecha} = V_{max} \tag{4.4b}$$

$$VT_{Ruedaizquierda} = V_{max} \tag{4.4c}$$

donde:

- VT_robot se refiere a la velocidad de traslación total del móvil
- V_{max} es un entero con valor 10, ya que es posible dar diez niveles de velocidad al robot khepera II
- $VT_{Ruedaderecha}$ y $VT_{Ruedaizquierda}$ son las velocidades de traslación de las ruedas, derecha e izquierda

Lo cual es válido pues no se viola la ley de la ecuación (3.1) en la que se afirma que el promedio de las velocidades es igual al valor total de desplazamiento del móvil. La siguiente suposición es que el objeto se encuentre a 0° con respecto al eje que une a las dos llantas del robot, para tal caso la velocidad de traslación debería de ser cero, ya que para corregir el error de orientación no es necesario trasladarse, sino rotar sobre su centro de masa. Para lograr esta rotación las velocidades de las ruedas deben ser de igual magnitud, pero de sentido opuesto. Entonces el valor máximo también puede asignarse a cada rueda pero de sentido contrario como se mencionó anteriormente y suponiendo que el objetivo se encuentra del lado derecho, se estable que:

$$VT_robot = 0 \tag{4.5a}$$

$$VGIRO_{Ruedaderecha} = -V_{max} \tag{4.5b}$$

$$VGIRO_{Ruedaizquierda} = V_{max} \tag{4.5c}$$

Done el prefijo VGIRO se refiere a una magnitud de velocidad que no es la velocidad de traslación pero que puede ser igual en magnitud. Lo cual sigue siendo válido pues nuevamente el valor promedio de las velocidades de las ruedas es igual a la velocidad total de traslación del móvil. Puede verse que con esto se cumple la capacidad del robot de rotar sobre su centro geométrico.

La última suposición es cuando el error de orientación es mayor que 0° y menor que 90°, cabe aclara que las mediciones de error de orientación las tomaremos del ángulo que forma la línea que une al centro geométrico del obejeto al punto central de la base inferior de la imagen, es decir el resto de los casos, si para 00° la velocidad de traslación es máxima y para 0° es nula. Podemos normalizar el valor asignado al valor máximo 1 y al valor mínimo 0. Para los valores intermedios habrá un valor real asociado entre 0 y 1; lo más fácil y por sentido común es suponer que si el grado de orientación del objeto en la imagen con respecto a la base de la imagen es 45°, entonces la velocidad de traslación del móvil no será máxima ni mínima sino la mitad, es decir 0.5, pues el error de orientación se encuentra a la mitad de los dos casos anteriores. Es decir, que la velocidad de traslación total será:

$$VT_{Robot} = V_{max} * \frac{90}{e_0} \tag{4.6}$$

donde e_0 se refiere al error de orientación dado entre 0 y 90 grados.

Sin embargo esa no puede ser la velocidad de las dos ruedas pues el robot seguiría una trayectoria recta y el robot debe seguir una trayectoria circular pues existe un error de orientación mayor que 0° y menor que 90°. Siguiendo con la ley de que el promedio de las velocidades es igual a la velocidad total de desplazamiento del móvil y de que para este caso una velocidad debe ser mayor que la otra, se sigue el siguiente razonamiento: La velocidad de traslación se ha establecido como un valor real proporcional de la velocidad máxima del móvil, calculada como la normalización del error de orientación entre 0 y 1, de esta forma se alcanza una velocidad de traslación menor a la máxima ya que se considera que el robot esta describiendo una trayectoria circular. Para lograr esto, se cálcula el complemento de la V_{max} que cada rueda del robot tiene capacidad de adoptar y cuyo valor entero es 10.

$$VT_{com} = V_{max} - VT_{Robot} \tag{4.7}$$

donde: V_{com} se refiere a una velocidad complementaria. Esta velocidad complementaria se usara para variar las velocidades de las ruedas ya que aunque se conoce la velocidad de traslación, no puede aplicarse a las dos ruedas pues describiría un desplazamiento rectilíneo. Con el fin de variar las velocidades de las ruedas, se divide la velocidad complementaria a la mitad y se resta una mitad a la rueda que deba ir más lento y se suma el resto a la rueda que deba ir más rápido, de esta forma no se viola la ley de un veiculo de conducción diferencial, que obliga a que el promedio de las ruedas es la velocidad de traslación y ya que aqui cálculamos el complemento de esa velocidad entonces debe darse la mitad a cada rueda para realizar el ajuste de velocidad, como se muestra a continuación.

$$VGIRO_{Robot} = \frac{V_{com}}{2} \tag{4.8a}$$

$$VGIRO_{Ruedaizquierda} = VT_{Robot} + VGIRO_{Robot} \tag{4.8b}$$

$$VGIRO_{Ruedaderecha} = VT_{Robot} - VGIRO_{Robot} \tag{4.8c}$$

Lo cual parece ser valido pues de igual forma que las anteriores el promedio de las velocidades de las dos ruedas da como resultado la VT_{Robot}. Bajo este nuevo módelo de sistema de ecuaciones, se diseñan las reglas difusas del controlador difuso.

4.2.1Propuesta de conjuntos difusos para el robot khepera II

Para el diseño del controlador difuso se proponen cinco variables lingüísticas, dos para la entrada y tres para la salida:

Variables lingüísticas de entrada

Para tratar el problema de orientación se propone la variable lingüística ı**Orientacin**(**O**)▮ y sus valores son: muy izquierda, izquierda, centrado, derecha y muy derecha, bajo la forma de funciones de membresía triangulares, cuyo universo del discurso está comprendido entre 0° y 180°, vease figura(4.16). Para tratar el problema de distancia al objetivo se propone la variable lingüística ı**Distancia**(**D**)▮ y sus valores son: lejos, cerca y muy cerca, bajo la forma de funciones de membresía triangulares, cuyo universo del discurso está comprendido entre **0**

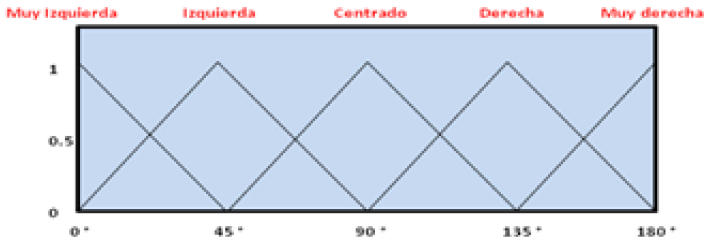


Figura 4.16: Funciones de membresía para la variable orientación (O)

y el número máximo de pixeles obtenido en la etapa de entrenamiento definido como el área geométrica del objeto con el momento de orden cero $\mathbf{m_{00}}$, vease figura (4.17).

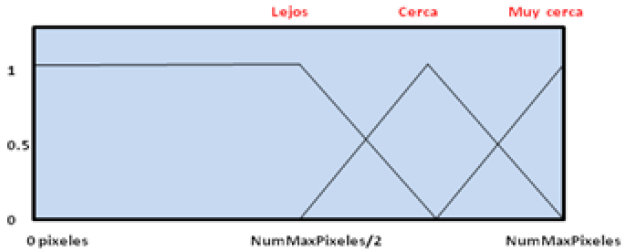


Figura 4.17: Función de membresía para la variable distancia (D)

Variables lingüísticas de Salida

Como se mencionó en el análisis del modelo diferencial del robot khepera II, se llegó a la conclusión de que deben definirse dos magnitudes de velocidad, la velocidad de traslación y la una velocidad de giro, cabe mencionar que el valor máximo de la velocidad en cada rueda será diez, pues es posible dar diez niveles de velocidad a cada motor del robot khepera II, según el fabricante, [Manual khepera]. Con respecto a ese análisis, se proponen tres variables lingüísticas de salida cuyos valores se expresan en funciones triangulares.

1. Los valores de la variable lingüística **"velocidad de traslación de salida (VTS)"** son: VTReposoSalida, VTDespacioSalida, VTMediaSalida, VTRapidoSalida, VTMuyrapidoSalida. Estas funciones de membresía se muestran en la figura (4.18).

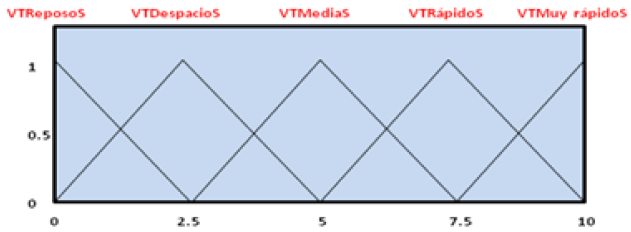


Figura 4.18: Funciones de membresía para la variable Velocidad de Traslación de Salida (VTS)

2. Los valores de la variable lingüísticas de giro izquierdo son: **"Velocidad de giro izquierda (VGI)"** GNegMaxIzq, GNegMaxIzq,GNuloIzq, GPosMedIzq, GPosMaxIzq, vease figura (4.19).

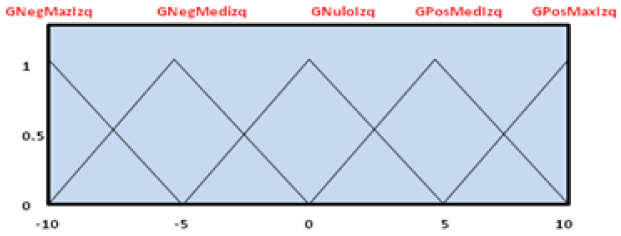


Figura 4.19: Funciones de membresía para la variable de Velocidad de Giro Izquierda (VGI)

3. Los valores de la variable lingüísticas de giro derecho son: **"Velocidad de giro derecha (VGD)"** GNegMaxDer, GNegMaxDer,GNuloDer, GPosMedDer, GPosMaxDer, vease figura (4.20).

Una vez que se tiene la descripción difusa de las variables de control, se procede a definir las reglas de control difuso. Siempre el número de reglas total, está dado por las posibles combinaciones de los conjuntos difusos de entrada. Para este caso se definen dos entradas, la primera se define como orientación con 5 valores lingüísticos y la segunda se define como distancia con 3 valores lingüísticos, entonces el número de reglas resultantes debería ser 5x3 = 15, sin embargo como se puede apreciar, en la tabla matricial de mapeo en donde

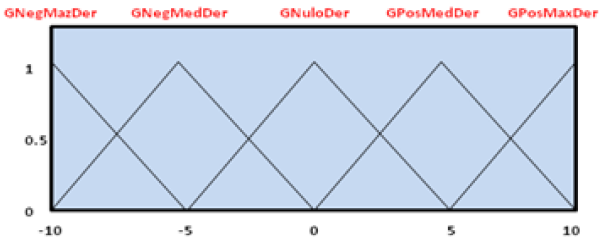


Figura 4.20: Funciones de membresía para la variable de Velocidad de Giro Derecha (VGD)

será mas fácil visualizar la relación de las entradas y las salidas, vease tabla(), las variables lingüísticas se reducen a 11 reglas, ya que si analizamos los extremos de la tabla (los cuales se refieren al grado de orientación muy derecha y muy izquierda), ante la variable de distancia en cualquiera de los tres casos, el controlador actúa de la misma forma, por lo que de las seis reglas involucradas se reducen a dos reglas, una para cada extremo de la tabla. Los valores de los conjuntos difusos de salida, se refieren a la velocidad de traslación y giro, se asignaron de acuerdo al análisis previo que se hizo al modelo diferencial del robot, tal y como lo establece la teoría de la lógica difusa en la que la experiencia del experto es la base de las reglas.

A continuación se describe cada una de las variables lingüísticas utilizadas.

VTReposoS: Representa al robot en “estado de reposo”, **S** para todas las variables lingüísticas denota un valor de salida.

VTMediaS: Representa un nivel de velocidad cuyo máximo valor es “la mitad del nivel máximo”.

VTRapidoS: Representa un nivel de velocidad cuyo máximo valor es exactamente el 75 % del máximo nivel de velocidad, es decir “Rápido”.

VTMuyrpido: Como su nombre lo dice, es “muy rápido” puesto que es el máximo nivel de velocidad.

GPosMaxIzq, **GPosMaxDer**: Estos dos valores asociados a la variable lingüística Velocidad de Giro de la rueda Izquierda y Derecha, se refieren al “Máximo valor positivo”, es decir a la velocidad máxima necesariamente de signo positivo.

GNegMaxIzq, **GNegMxDer**: Estos dos valores asociados a la variable lingüística Velocidad de Giro de la rueda Izquierda y Derecha, se refieren al "Máximo valor negativo", es decir a la velocidad máxima necesariamente de signo negativo.

GPosMedIzq, **GPosMedDer**: Estos dos valores asociados a la variable lingüística Velocidad de Giro de la rueda Izquierda y Derecha, se refieren al "valor medio positivo", es decir a la mitad del nivel máximo de velocidad, necesariamente de signo positivo.

GNegMedIzq, **GNegMedDer**: Estos dos valores asociados a la variable lingüística Velocidad de Giro de la rueda Izquierda y Derecha, se refieren al "valor medio negativo", es decir a la mitad del nivel máximo de velocidad, necesariamente de signo negativo.

GNuloIzq, **GNuloDer**: Estos dos valores asociados a la variable lingüística Velocidad de Giro de la rueda Izquierda y Derecha, se refieren al "valor nulo", es decir cero.

	Muy derecha			Derecha			Centrado			Izquierda			Muy izquierda		
Lejos	*GPosMaxIzq*	*VTReposoS*	*GNegMaxDer*	*GPosMedIzq*	*VTMediaS*	*GNegMedDer*	*GNuloIzq*	*VTMuyrapidoS*	*GNulDer*	*GNegMedIzq*	*VTMediaS*	*GPosMedDer*	*GNegMaxIzq*	*VTReposoS*	*GPosMaxDer*
Cerca	*GPosMaxIzq*	*VTReposoS*	*GNegMaxDer*	*GPosMedIzq*	*VTMediaS*	*GNegMedDer*	*GNuloIzq*	*VTRapidoS*	*GNulDer*	*GNegMedIzq*	*VTMediaS*	*GPosMedDer*	*GNegMaxIzq*	*VTReposoS*	*GPosMaxDer*
Muy cerca	*GPosMaxIzq*	*VTReposoS*	*GNegMaxDer*	*GPosMedIzq*	*VTReposoS*	*GNegMedDer*	*GNuloIzq*	*VTReposoS*	*GNulDer*	*GNegMedIzq*	*VTReposoS*	*GPosMedDer*	*GNegMaxIzq*	*VTReposoS*	*GPosMaxDer*
	VGiroIzq	*VTReposoS*	*GNegMaxDer*	*GPosMedIzq*	*VTReposoS*	*GNegMedDer*	*GNuloIzq*	*VTReposoS*	*GNulDer*	*GNegMedIzq*	*VTReposoS*	*GPosMedDer*	*GNegMaxIzq*	*VTReposoS*	*GPosMaxDer*

El conjunto de reglas razonable es el siguiente:

- Si (Posición es Centrado) y (Distancia es Lejos) Entonces
  (VTS es Muy rápido) y (GRI es Nulo) y (GRD es Nulo)

- Si (Posición es Centrado) y (Distancia es Cerca) Entonces
  (VTS es Rápido) y (GRI es Nulo) y (GRD es Nulo)

- Si (Posición es Centrado) y (Distancia es Muy cerca) Entonces
  (VTS es en Reposo) y (GRI es Nulo) y (GRD es Nulo)

- Si (Posición es Derecha) y (Distancia es Lejos) Entonces
  (VTS es Media) y (GRI es Media Positiva) y (GRD es Media Negativa).

- Si (Posición es Derecha) y (Distancia es Cerca) Entonces
  (VTS es Media) y (GRI es Media Positiva) y (GRD es Media Negativa).

- Si (Posición es Derecha) y (Distancia es Muy cerca) Entonces
  (VTS es en Reposo) y (GRI es Media Positiva) y (GRD es Media Negativa).

- Si (Posición es Izquierda) y (Distancia es Lejos) Entonces
  (VTS es Media) y (GRI es Media Negativa) y (GRD es Media Positiva).

- Si (Posición es Izquierda) y (Distancia es Cerca) Entonces
  (VTS es Media) y (GRI es Media Negativa) y (GRD es Media Positiva).

- Si (Posición es Izquierda) y (Distancia es Muy cerca) Entonces
  (VTS es en Reposo) y (GRI es Media Negativa) y (GRD es Media Positiva).

- Si (Posición es Muy derecha) Entonces
  (VTS es en Reposo) y (GRI es Máxima Positiva) y (GRD es Máxima Negativa).

- Si (Posición es Muy Izquierda) Entonces
  (VTS es en Reposo) y (GRI es Máxima Negativa) y (GRD es Máxima Positiva).

Una vez definidas el conjunto de reglas y los conjuntos difusos, se implementan en un modelo Mamdani descrito en la sección 3.5.7 del capítulo 3, siguiendo el siguiente esquema de lazo cerrado.

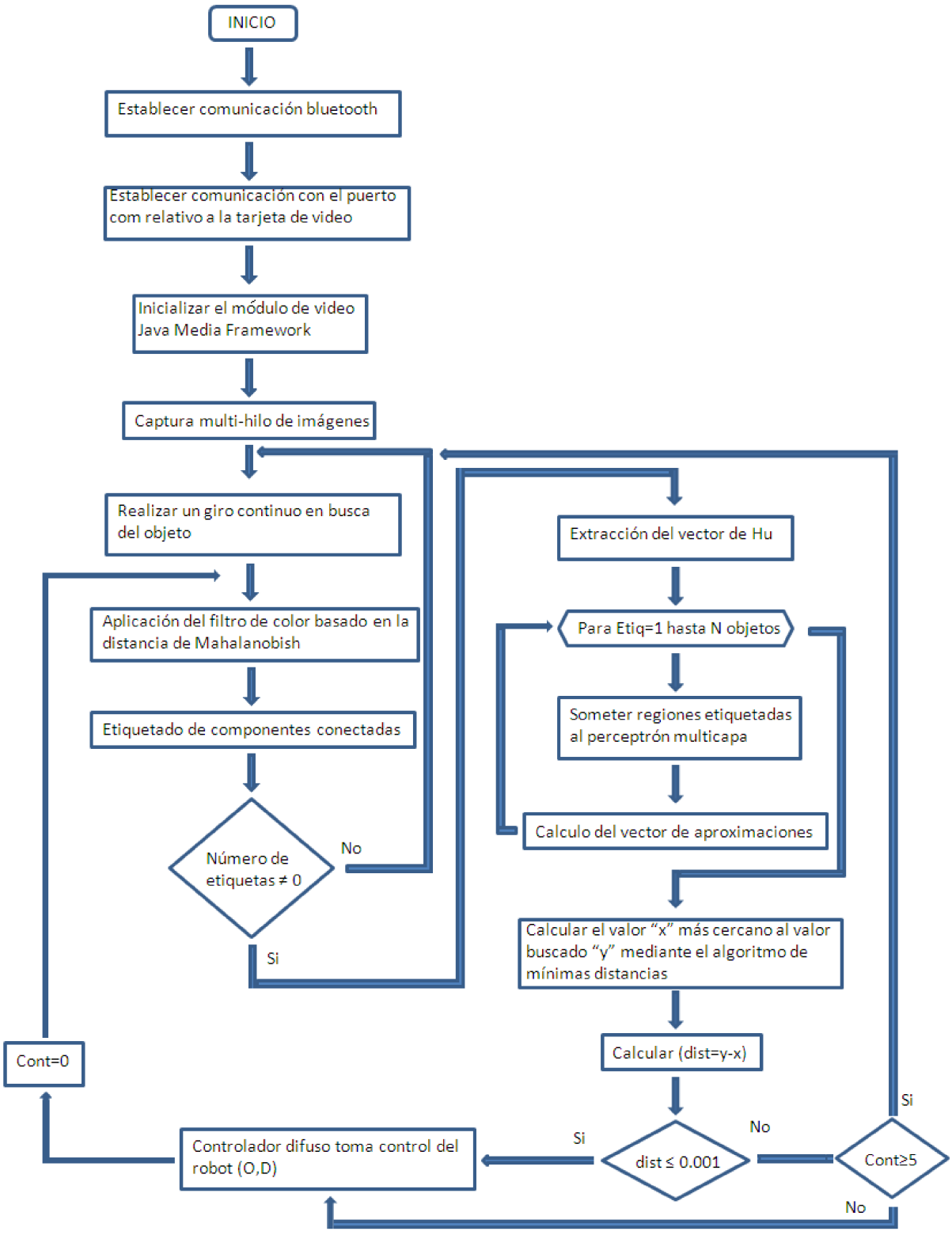


Figura 4.21: Diagrama de flujo del sistema en lazo cerrado

Capítulo 5

Experimentación y resultados

En este capítulo, se muestra una serie experimentos con los que se mide el desempeño de la metodología de control, para la navegación y manipulación de objetos a través del seguimiento de puntos de referencia. Para este caso en particular, el punto de referencia es el centro de masa de un objeto segmentado de una imagen mediante el reconocimiento visual de formas y color. Una de las medidas que se tomaron antes de llevar el algoritmo de control difuso propuesto a la práctica, fue modelar las reglas de control generadas, con la interfaz gráfica del toolbox de Matlab. Con el propósito de mostrar la utilidad de esta herramienta de simulación, se mostraran los resultados obtenidos con la herramienta de simulación de metlab en comparación con los obtenidos por el sistema. Cabe mencionar que para llevar a cabo la experimentación es necesario un entrenamiento del sistema de visión y reconocimiento, con el fin de obtener rasgos descriptivos de acuerdo al objetivo que el robot persigue (entiéndase navegar hacia el), una vez que el sistema converge en su fase de entrenamiento el comportamiento del robot será similar para todos los casos, ya que el algoritmo repite de forma infinita el reconocimiento de su objetivo y avanza hacia el, corrigiendo su error de orientación haciéndolo nulo lo más pronto posible hasta que tal objeto este en rango visual de manipulación. En especifico la tarea con la que se experimentó, fue la busqueda de un objeto para tomarlo y trasladarlo con el fin de colocarlo sobre otro objeto de forma similar al objeto manipulado. Para lograrlo fue necesario partir la tarea en tres sub-tareas: Buscar y aproximarse, tomar y por ultimo realizar una tarea con él, en este caso

desplazarlo hacia una ubicación. Cada tarea por simple que parezca requirió el empleo de varios módulos que a su vez dividen a la tarea en sub-tareas. Tal es el caso de la tarea de reducción del espacio de búsqueda mediante discriminación por color, la de reconocimiento y clasificación de formas, la de evaluar distancias dadas por la masa total del objetivo durante el entrenamiento y errores de orientación, y por último ejecución del control difuso, el cual se reduce a proporcionar un nivel de velocidad con la mayor precisión posible a cada motor que forma parte del sistema mecánico, para convertirlo en un móvil útil.

5.0.4. Configuración de la red neuronal usada

La configuración de la red neuronal artificial perceptrón multicapa, se eligió después de una serie de pruebas que describieran un desempeño satisfactorio, ante la clasificación de dos objetos de formas geométricas sencillas, tales como un cubo y otro objeto de forma cilíndrica. Estos dos objetos tienen características geométricas diferentes que permiten extraer de ellos rasgos descriptivos que facilitan su clasificación con alto éxito. A continuación se muestra un conjunto de 60 vectores compuestos por los dos primeros invariantes de Hu, ver tabla (5.1) y tabla (5.2), estos invariantes son descritos en el capitulo 3, estos vectores son graficados en un plano coordenado bidimensional en donde el primer invariante corresponderá a las abscisas y el segundo a las ordenadas, estos valores son normalizados entre 0 y 1. Puede verse a simple vista que los dos objetos considerados clases son perfectamente separables, véase figura (5.1). Como se menciono en el capítulo anterior la red neuronal artificial utilizada tiene tres características fundamentales, que son, dos neuronas en la capa de entra, las cuales corresponden al vector descriptivo de imágenes, compuesto por los dos primeros invariantes de Hu, la característica número dos es que por lo menos exista una capa interna de n número de neuronas, tal como lo exige el modelo de red neuronal artificial perceptrón multicapa, y por último la tercera característica es que en la capa de salida solo exista una neurona. Atendiendo a lo anterior se probo con una arquitectura que a la entrada tienes 2 neuronas, tres capas ocultas, de las cuales la primera cuenta con tres neuronas, la segunda 7 neuronas y la tercera 3 neuronas y como se menciono anteriormente una neurona en la capa de salida,

véase la figura (5.2). Sometiendo los 60 vectores descriptores obtenidos de las imágenes al

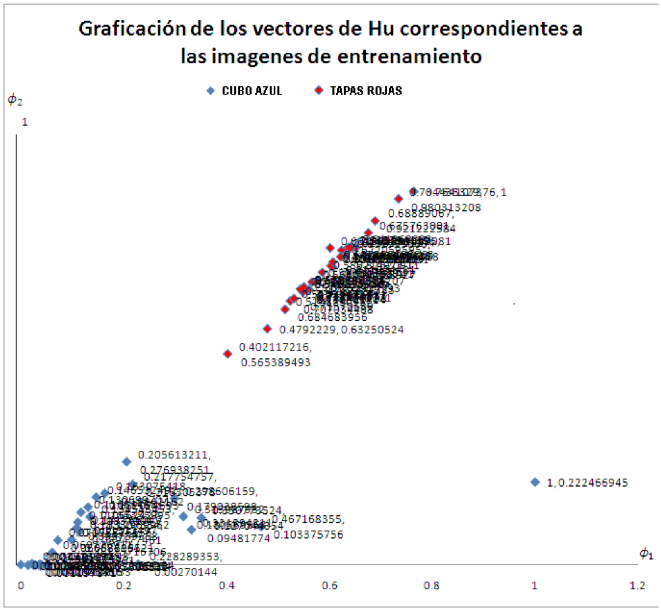


Figura 5.1: Graficación de 60 vectores de imégenes de los dos primeros invariantes de Hu

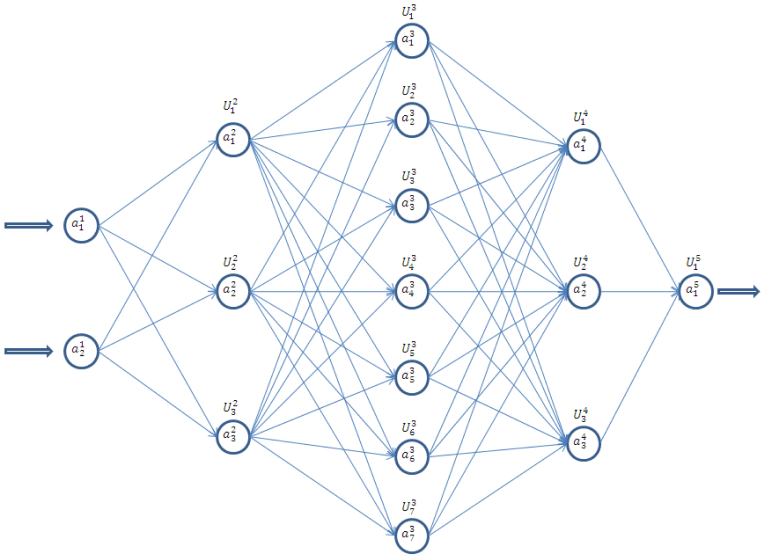


Figura 5.2: Configuración del perceptrón multicapa utilizado

proceso de entrenamiento, se obtiene una tabla de pesos, véase tabla (5.4) y umbrales en la tabla (5.5). Los parámetros de inicialización del sistema y finalización se muestran en la tabla

	ϕ_1	ϕ_2		ϕ_1	ϕ_2		ϕ_1	ϕ_2
1	0.21	0.28	11	0.55	0.74	21	0.58	0.76
2	0.55	0.74	12	0.62	0.83	22	0.55	0.73
3	0.68	0.89	13	0.73	0.98	23	0.64	0.85
4	0.64	0.85	14	0.76	1.0	24	0.61	0.80
5	0.52	0.71	15	0.57	0.76	25	0.51	0.68
6	0.63	0.84	16	0.48	0.63	26	0.53	0.71
7	0.59	0.78	17	0.64	0.85	27	0.56	0.74
8	0.60	0.80	18	0.40	0.57	28	0.69	0.92
9	0.62	0.84	19	0.55	0.75	29	0.60	0.85
10	0.54	0.74	20	0.65	0.85	30	0.61	0.81

Cuadro 5.1: Primeros dos invariantes de Hu, normalizados entre 0 y 1, de 30 imágenes de un objeto cilíndrico

(5.3), a excepción de los pesos y umbrales que inicialmente obtienen valores aleatorios entre 0 y 1.

5.0.5. Implementación del sistema de lógica difusa propuesto para su simulación en la interfaz grafica del toolbox de Matlab

Primera etapa

Uso del “FIS Editor”, en el que se definen las funciones de membresía de tipo triangular asociadas a las variables lingüísticas de entrada y salida con sus respectivos valores. El sistema de inferencia seleccionado es el de Mamdani o max-min, figura (5.3).

Segunda etapa

A continuación se muestra las variables de entrada y salida, con su universo del discurso correspondiente y sus respectivos valores representados con funciones triangulares,

	ϕ_1	ϕ_2		ϕ_1	ϕ_2		ϕ_1	ϕ_2
1	0.06	0.03	11	1.0	0.22	21	0.04	0.0
2	0.12	0.14	12	0.0	0.0	22	0.03	2.68E-05
3	0.16	0.19	13	0.04	2.60E-4	23	0.11	0.12
4	0.15	0.18	14	0.05	0.0	24	0.02	3.72E-05
5	0.13	0.16	15	0.13	0.13	25	0.30	0.18
6	0.10	0.07	16	0.03	2.08E-04	26	0.23	0.0
7	0.32	0.13	17	0.47	0.10	27	0.01	0.0
8	0.35	0.13	18	0.11	0.09	28	0.07	0.07
9	0.02	0.0	19	0.22	0.22	29	0.11	0.10
10	0.02	0.0	20	0.33	0.09	30	0.13	0.02

Cuadro 5.2: Primeros dos invariantes de Hu, normalizados entre 0 y 1, de 30 imágenes de un objeto cúbico

$Regladeaprendizaje$	$\delta = 0.5$
$Intervalodevalidacin$	100
$Nmerodeiteracionesalcanzadas$	3602
$Errortotal$	0.00009817788
$Errortotaldevalidacin$	0.00001577874

Cuadro 5.3: Parámetros de inicialización y finalización del entrenamiento del perceptrón multicapa

figura(5.4). En donde para la variable lingüística orientación el universo del discurso se define en el rango 0° a 90° grados, que representa el error que el objeto guarda con respecto al centro de la imagen. Véase figura (5.4). Para la variable lingüística distancia, depende del análisis que se realice durante el aprendizaje de la primera imagen del conjunto de imágenes de entrenamiento, ya que el universo del discurso va de 0 pixeles al tamaño del objeto extraído de la primera imagen de entrenamiento, por lo que es necesario que esa primera imagen sea tomada con la cámara del robot estando el objeto en rango de operación. Véase figura (4.4)

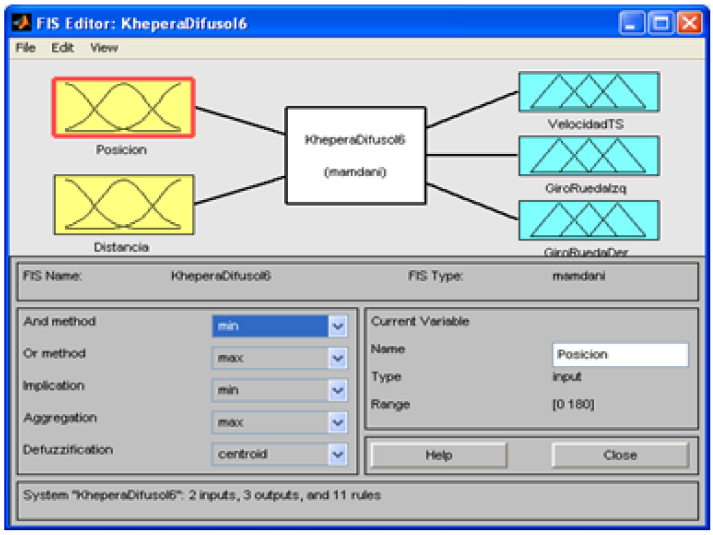


Figura 5.3: Edición del sistema difuso khepera II para navegación y manipulación

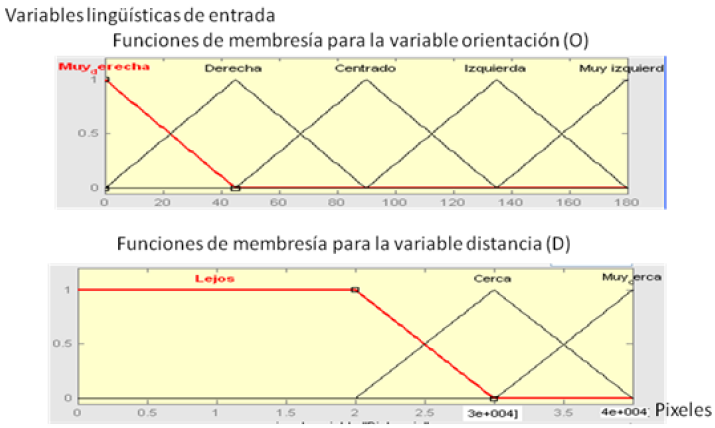


Figura 5.4: Definición de las variables lingüísticas de entrada

y figura (5.4) . Para la variable lingüística Velocidad de Transferencia de Salida (VTS), se observa que el universo del discurso se encuentra entre -12 y 12, véase figura(5.5), el máximo nivel de velocidad debería ser -10 para la máxima velocidad negativa y 10 para la máxima velocidad positiva; sin embargo es necesario completar el triangulo de la función para los valores de los extremos. La razón es que se está utilizando el método de defuzificación del centroide o promedio de los centros y si está incompleto el conjunto, el centroide se encontrará a la mitad de la pendiente y no en el punto máximo del conjunto. Es el mismo

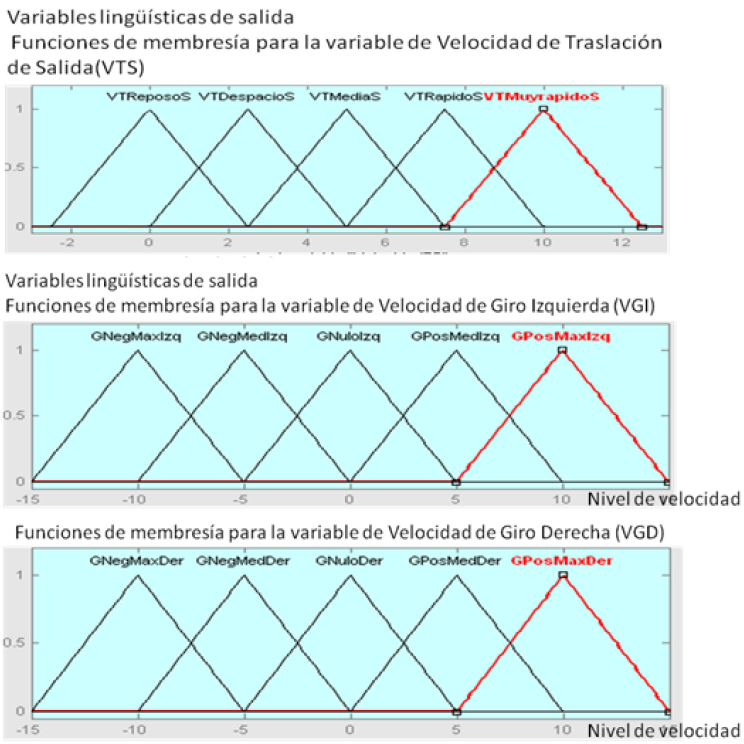


Figura 5.5: Definición de las variables lingüísticas de salida

caso en la Variable de Giro Izquierdo y Variable de Giro Derecho, en donde la velocidad máxima tanto negativa como positiva debería ser 10.

Tercera etapa

Se definen las reglas en el editor de reglas difusas, las cuales se ven en la figura (5.6).

Cuarta fase

Una vez cargado el sistema difuso, es posible hacer uso de las reglas para modelar los posibles casos tomando en cuenta la orientación y distancia. A continuación se muestra un caso que muestran cual será la acción de corrección del controlador sobre las velocidades de traslación y de gira para cada rueda. Se elige un caso en particular para probar el sistema simulado con el toolbox de matlab el cual da solución al siguiente cuestiona miento:

¿Cuál será la acción de corrección del controlador cuando la orientación es 56° y la

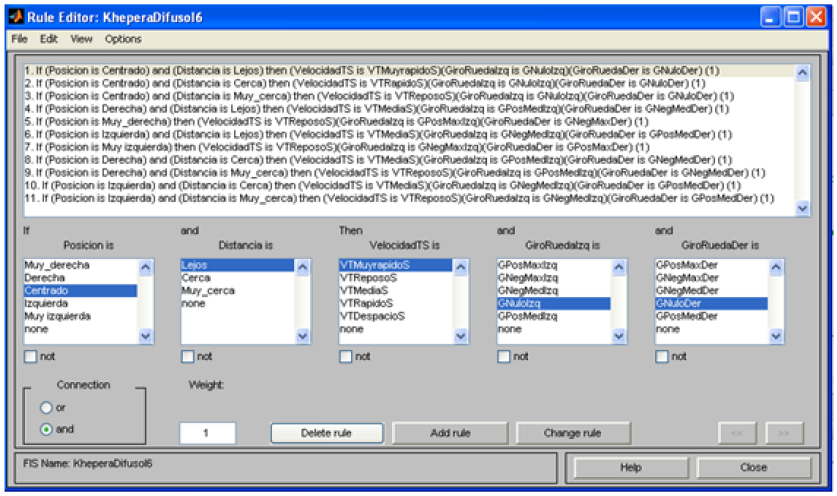


Figura 5.6: Definición de las reglas difusas en el editor de reglas difusas

distancia del objeto dado en función de su tamaño en pixeles es de 10,000 pixeles? Cuando el tamaño del objeto en rango de operación es de 40,000 píxeles. En la figura (5.7) puede observarse que la salida de la velocidad de traslación es de 6.59 y para la velocidad de giro de la rueda izquierda es de 3.45 y para la velocidad de la rueda derecha -3.45. Como se definió en el análisis del punto (4.1.3) del capítulo 4, la velocidad de traslación de salida y la velocidad de giro para los casos en que el error de orientación está entre 1 y 89 o 91 7 179, tanto la velocidad de traslación como la velocidad de giro de cada rueda podrán alcanzar un nivel máximo de 5. Aplicando el sistema de ecuaciones (4.8b) y (4.8c), se obtiene que:

$$VGIRO_{Ruedaizquierda} = 6.59 + \frac{(3.45)(2)}{=} 8.315$$

$$VGIRO_{Ruedaderecha} = 6.59 + \frac{(-3.45)(2)}{=} 4.865$$

5.0.6. Probando el sistema difuso con el modelo Mamdani de forma manual

Probando el sistema

Antes de llevar a la programación el algoritmo propuesto se realiza una prueba de comparación con el ejercicio que se hizo con el simulador descrito en el punto anterior, con

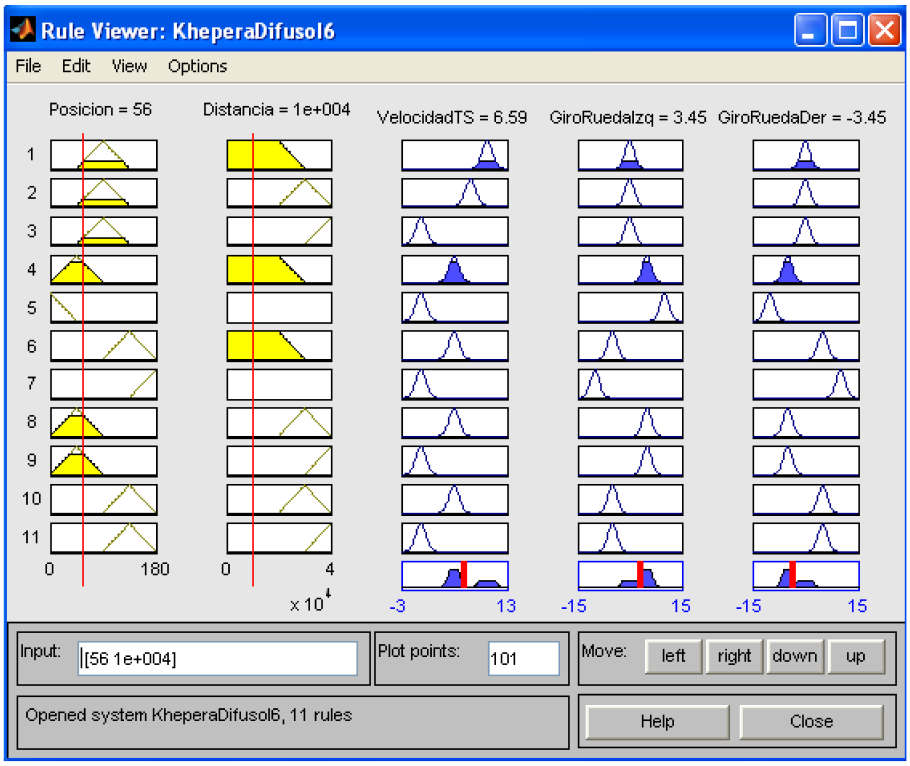


Figura 5.7: Modelado de acción de corrección de las reglas difusas representado en la grafica de reglas (Rule viewer)

el fin de tener claro el funcionamiento del algoritmo y garantizar que los resultados que se obtengan del sistema programado sean muy similares o igual si es posible a la simulación. Se tomara el mismo cuestiona miento con el que se probo la simulación.

¿Cuál será la acción de corrección del controlador cuando la orientación es 56° y la distancia del objeto dado en función de su tamaño en píxeles es de 10,000 píxeles?

Según el análisis para una O = 56° le corresponde un grado de membresía “Derecha” considerando que es una función triangular , de 0.7555 y “Centrado” 0.2444, del mismo modo para D=10,000 píxeles, le corresponde un grado de membresía “lejos” de 1, véase figura (5.8).

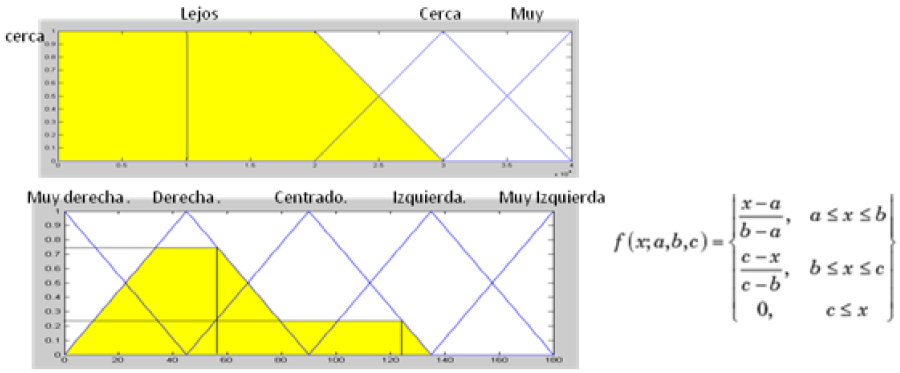


Figura 5.8: Fuzzificando variables de entrada (O,D), mediante funciones triangulares

Según el análisis de inferencia para la VTS (Velocidad de Transferencia de Salida)

A continuación se usan las reglas difusas que cumplen con las variables de entrada y se calcula su grado de membresía.

$if P(centrado) and D(lejos) then VTS(Muyrpido) VGI(esnulo) VGD(esnulo)$

$$\mu_{Muyrapido}(O,D) = min[\mu_{Centrado}, \mu_{Lejos}] = min[0.2444, 1] = 0.2444$$

$if P(derecha) and D(lejos) then VTS(Media) VGI(PMedIzq) VGD(NMedDer)$

$$\mu_{Media}(O,D) = min[\mu_{Derecha}, \mu_{Lejos}] = min[0.7555, 1] = 0.7555$$

Defuzzificación: Método del Centroide

$$VP = \frac{(\text{Centro de } \mu_{Media}) * (\mu_{Media}(O,D)) + (\text{Centro de } \mu_{Muyrpido}) * (\mu_{Muyrpido(O,D)})}{\mu_{Media}(O,D) + (\mu_{Muyrpido(O,D)})} \quad (5.1)$$

En donde VP se considera la velocidad promedio, es decir no es aun salida del sistema.

Sustituyendo los grados de menbresía correspondientes obtenidos en el análisis de inferencia, en el modelo de defuzificación del centroide se obtiene en valor del promedio de los centros, por lo que VP puede ser considerado ya la VTS (Velocidad de Transferencia de salida), véase figura(5.9):

$$VP = \frac{(5)*(0.7555)+(10)*(0.2444)}{(0.7555)+(0.2444)} = 6.2214$$

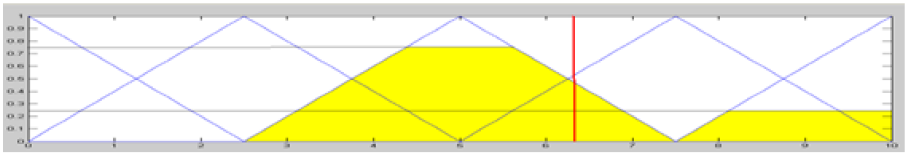


Figura 5.9: Proyección de los valores de membresía de salida para VTS

Según el análisis de inferencia para las variables VGI (Velocidad de Giro Derecha) Y VGD (Velocidad de Giro Izquierda)

$$if P(centrado) and D(lejos) then VTS(Muyrpido) VGI(esnulo) VGD(esnulo)$$

$$\mu_{GRInulo} = min[\mu_{Centrado}, \mu_{Lejos}] = min[0.2444, 1] = 0.2444$$

$$if P(derecha) and D(lejos) then VTS(Media) VGI(PMedIzq) VGD(NMedDer)$$

$$\mu_{GRIPosMedIzq}(O,D) = min[\mu_{Derecha}, \mu_{Lejos}] = min[0.7555, 1] = 0.7555$$

Sustituyendo el grado de membresía antes obtenido para la VGI enla ecuación (5.1) se obtiene que:

$$VP = \frac{(0)*(0.2444)+5*(0.7555)}{(0.2444)+(0.7555)} = 3.780$$

El valor promedio obtenido será la salida del defuzificador para la varia velocidad de giro izquierda, ver figura (5.10).

$$VGI = 3.780$$

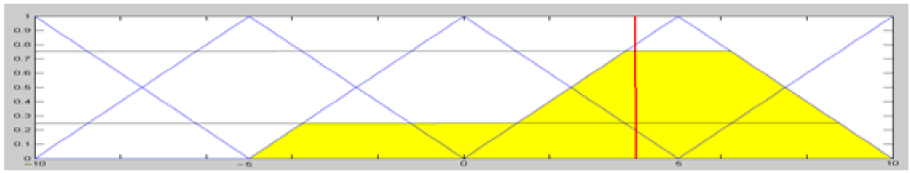


Figura 5.10: Proyección de los valores de membresía de salida para la VGI

Según el análisis de inferencia para la VGD (Velocidad de Giro Derecha)

$$if P(centrado) and D(lejos) then VTS(Muyrpido) VGI(esnulo) VGD(esnulo)$$

$$\mu_{GRDnulo}(O,D) = min[\mu_{Centrado}, \mu_{Lejos}] = min[0.2444, 1] = 0.2444$$

$$if P(derecha) and D(lejos) then VTS(Media) VGI(PMedIzq) VGD(NMedDer)$$

$$\mu_{GRDNegMerDer}(O,D) = min[\mu_{Derecha}, \mu_{Lejos}] = min[0.7555, 1] = 0.7555$$

Sustituyendo el grado de membresía antes obtenido para la VGD en la ecuación (5.1) se obtiene que:

$$VP = \frac{(0)*(0.2444)+(-5)*(0.7555)}{(0.2444)+(0.7555)} = -3.780$$

El valor promedio obtenido será la salida del defizificador para la varia velocidad de giro izquierda, véase figura (5.11).

$$VGD = -3.780$$

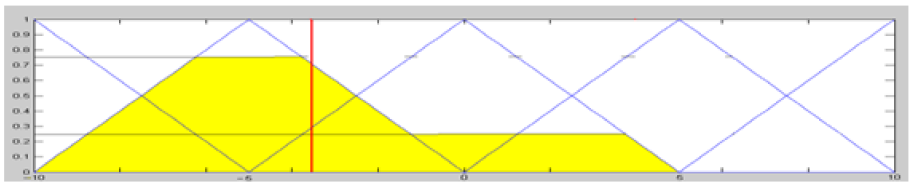


Figura 5.11: Proyección de los valores de membresía de salida para la variable VGD

Como puede notarse, los valores que se obtuvieron son muy similares a los de la simulación con el Toolbox de Matlab que se reportan en la figura 5.5 de este capítulo. Una vez que se ha ensayado lo suficiente con el sistema de reglas difusas y el método de fuzificación y defuzificación elegidos.

5.0.7. Implementación del conjunto de reglas en un ambiente experimental de mesa con el robot khepera II

Las condiciones de experimentación se definen como sigue:

- El móvil se encuentra en algún punto de una mesa de dimensiones 1.0m de ancho por 1.3 m de largo, cuya superficie es laminada de color blanco.
- Existe un nivel de iluminación de luz blanca, proporcionada por dos lámparas circulares de gas neón.
- Sobre la mesa de trabajo están colocadas figuras de forma geométrica para las cuales se cuenta con una base de imágenes para el previo entrenamiento.
- La perspectiva de visión es paralela al plano en que se mueve el móvil, pues la cámara se encuentra colocada sobre el móvil.

Para comenzar, partimos del hecho de que una sesión de análisis del espacio de trabajo del robot comprende la segmentación por color, etiquetado de componentes conectadas, extracción de caracteristicas y clasificación de formas. En este sentido el sistema de visión toma como estado inicial, el procesamiento de su ambiente de trabajo, el cual después de 5 sesiones de análisis fallidas sin encontrar al objetivo pasa el siguiente estado, el cual comprende un movimiento que describe una trayectoria circular cuyo radio al centro instantáneo de rotación es de 5cm. Con este movimiento se pretende dar un barrido completo del área de trabajo para localizar al objeto de interés, una vez ubicado el objetivo, se pasa a un siguiente estado, el cual comprende la navegación del móvil hasta alcanzar al objetivo. Es aquí en donde actúa el controlador difuso, ya que es quien de aquí en adelante toma el control de las ruedas del robot para llevar el móvil rígido con mayor grado de precisión mediante el seguimiento de un punto de referencia situado en el centro de masa del objeto en rango de visión. Cuando el objetivo es alcanzado, el controlador entra en pausa momentáneamente, mientras se ejecuta la toma del objeto, enseguida se pasa al siguiente estado, y prácticamente se vuelve a la misma dinámica, hasta encontrar la marca u objeto destino. Cuando lo encuentra se aproxima hasta alcanzar al objetivo denominado destino y una vez que es alcanzado, el controlador vuelve a entrar en pausa mientras se realiza la tarea de bajar al objeto. De esta forma, se cumple un ciclo completo y exitoso de búsqueda y manipulación de un objeto. El sistema volverá a su estado inicial en una posición cualquiera de la mesa,

esperando una nueva instrucción de búsqueda y manipulación. Se considera una sesión de fallo cuando el objetivo o el destino no es alcanzado gracias a un movimiento torpe por parte del móvil controlado por el controlador difuso, o que provoque que el móvil pierda de vista al objetivo o no complete su tarea con éxito.

Experimento 1

Busqueda y agarre de un cubo de color azúl con un tamaño de muy lejos al 80 %, véase tabla(5.6).
donde * denota el fracaso del intento.

Experimento 2

Busqueda yagarre de un cubo de color azul con un tamaño de muy lejos al 70 %, véase tabla (5.7).

Experimento 3

Busqueda de un cubo azul y colocación sobre otro cubo azul con un porcentaje de muy lejos del 80 %, véase tabla (5.8).

Experimento 4

Busqueda de un cubo azul y colocación sobre otro cubo azul con un porcentaje de muy lejos del 70 %, véase tabla (5.9).

5.0.8. Interpretación de resultados

Experimento 1

De acuerdo a lo observado en la Tabla (5.6),donde se muestran diez experimentos de búsqueda, que donde comprende desde el estado inicial de barrido visual del entorno y termina cuando el móvil ha navegado hasta el objeto de interés. La última medición reportada,

arroja con que precisión el móvil completo la tarea de búsqueda con un porcentaje del 80 % de espacio considerado muy lejos. lo anterior implica que el móvil pasará la mayor parte del tiempo a velocidad máxima posible mientras se encuentre fuera del 20 % de espacio de cercanía.

- 4 de los intentos terminan con velocidades en cada rueda de 0, y con un área mayor de 36036 píxeles. La cual según el entrenamiento, es el tamaño del objeto en rango de operación, por lo que el promedio del error de orientación para la tarea de búsqueda con éxito es de 4.72 grados, dados en radianes. Con un promedio de grado de error de reconocimiento de 0.00299625 %.

- Uno de los intentos termina con velocidades de nivel 2 en cada rueda, y con un área menor de 36036 píxeles, lo que indica que se detuvo antes de llegar al objetivo, el error de orientación es de 0.00° y el error de reconocimiento es de 0.001854. Aunque estos dos intentos pueden ser considerados buenos, debido a que cumplieron con la tarea de navegación hacia el objeto de interés sin atropellarlos. Se esperaba que la última lectura fuera de nivel de velocidad 0 y no de nivel de velocidad 2, lo que implica que el robot pudo haber seguido caminando por la inercia.

- 4 de los intentos terminan con velocidades máximas positivas del orden de nivel de velocidad 8 y nivel de velocidad 9, con un error de orientación promedio de 5.27125°, con un error de reconocimiento del 0.5024, lo que implica que se perdió de vista al objetivo o no hubo capacidad de reconocimiento, aunque estos intentos pudieron ser afectados por factores externos como un desnivel de la superficie donde navega el robot o un brinco producto de un movimiento brusco, aun estos 4 intentos fallidos representan el 40 % de los intentos.

Experimento 2

En el experimento 2 también se realiza una búsqueda y toma de un objeto. En este caso se cambió el porcentaje de distancia muy lejos al 70 %, lo que da un 30 % de distancia de

cercanía al robot para que el controlador haga uso de la disminución de velocidad y detener el robot justo antes de atropellar al objeto. Se observa claramente un mejor desempeño en la ejecución de una tarea de aproximación y manipulación. El error de orientación promedio es de 6.4° y el error de reconocimiento es 0.0012, aunque el error promedio de orientación parece más grande que en el experimento ,1 la toma del objeto se hace con mas precisión.

Experimento 3

En el experimento 3 se realiza la búsqueda de un cubo azul y colocación sobre otro cubo azul con un porcentaje de muy lejos del 80 %. Sólo fue posible realizar dos experimentos completos, se concluye que uno de ellos es exitoso y el otro no. El error medio de orientación para esta pequeña muestra de intentos es de 3.9° y el error de reconocimiento promedio es de 0.001, el 50 % del experimento resulto con éxito y el 50 %, por obvia deducción ya que la muestra de intentos es muy pequeña.

Experimento 4

Búsqueda de un cubo azul y colocación sobre otro cubo azul con un porcentaje de muy lejos del 70 %. En este experimento la muestra es de 4 intentos. En los cuatro intentos la operación se realiza con éxito, con un error medio de orientación de 5.6° y con un error medio de reconocimiento de 0.000855.

$w_{11}^1 = 0.0228$	$w_{11}^2 = 5.7815$	$w_{11}^3 = -1.8505$	$w_{11}^4 = 2.0070$
$w_{12}^1 = -0.07657$	$w_{12}^2 = 2.2912$	$w_{12}^3 = -4.9044$	$w_{21}^4 = 4.8661$
$w_{13}^1 = 0.3639$	$w_{13}^2 = -0.0202$	$w_{13}^3 = -4.8895$	$w_{31}^4 = 5.1705$
	$w_{14}^2 = 4.8257$		
$w_{11}^2 = 11.4998$	$w_{15}^2 = -0.2677$	$w_{21}^3 = -1.1758$	
$w_{12}^2 = 11.8608$	$w_{16}^2 = 2.3859$	$w_{22}^3 = -0.8260$	
$w_{13}^2 = -0.0808$	$w_{17}^2 = -0.0940$	$w_{23}^3 = -1.3206$	
	$w_{21}^2 = 5.8295$	$w_{31}^3 = 0.4578$	
	$w_{22}^2 = 2.4671$	$w_{32}^3 = 1.4882$	
	$w_{23}^2 = -0.0824$	$w_{33}^3 = 1.8798$	
	$w_{24}^2 = 5.0225$		
	$w_{25}^2 = -0.2146$	$w_{41}^3 = -1.6969$	
	$w_{26}^2 = 2.9826$	$w_{42}^3 = -3.8933$	
	$w_{27}^2 = 0.1897$	$w_{43}^3 = -3.5554$	
	$w_{31}^2 = -1.3651$	$w_{51}^3 = 8.9420$	
	$w_{32}^2 = -0.1899$	$w_{52}^3 = 1.4248$	
	$w_{33}^2 = 0.9649$	$w_{53}^3 = 1.3981$	
	$w_{34}^2 = -1.3212$		
	$w_{35}^2 = 0.2076$	$w_{61}^3 = -1.2599$	
	$w_{36}^2 = -0.8045$	$w_{62}^3 = -1.2794$	
	$w_{37}^2 = 0.3418$	$w_{63}^3 = -1.2413$	
		$w_{71}^3 = 0.1628$	
		$w_{72}^3 = 0.8306$	
		$w_{73}^3 = 0.3677$	

Cuadro 5.4: Tabla de pesos para la configuración de red perceptrón multicapa, 2,3,7,3,1

$U_1^2 = -3.0029$	$U_1^3 = -4.0349$	$U_1^4 = 0.0718$	$U_1^5 = -5.9655$
$U_2^2 = -3.0680$	$U_2^3 = -1.9959$	$U_2^4 = 2.1247$	
$U_3^3 = 0.9325$	$U_3^3 = 1.0459$	$U_3^4 = 2.3826$	
	$U_4^3 = -3.1770$		
	$U_5^3 = 1.0424$		
	$U_6^3 = -1.6682$		
	$U_7^3 = 0.0870$		

Cuadro 5.5: Tabla de umbrales de la capa 2 a la capa 5 de salida

No. Intento	Error de θ	Velocidad de Rueda Izquierda	Velocidad de Rueda Derecha	Error de reconocimiento del objeto	Tamaño objeto	Porcentaje de distancia
1	5.064	0	0	0.006808	36649	80 %
2	0.455	0	0	0.001132	42016	80 %
3	8.351	8	9	0.003742	33876	80 %
4	7.99	9	8	0.998025	29666	80 %
5	44.49	6	-1	0.996159	14360	80 % *
6	0.0	2	2	0.001854	34928	80 %
7	5.394	0	0	0.002032	41496	80 %
8	7.998	0	0	0.002013	38059	80 %
9	1.625	9	9	0.169137	13660	80 %
10	2.563	9	9	0.995411	27786	80 %

Cuadro 5.6: Experimentación de resultados de procedimiento de búsqueda con un porcentaje de distancia muy lejos del 80 %

No. Intento	Error de θ	Velocidad de Rueda Izquierda	Velocidad de Rueda Derecha	Error de reconocimiento del objeto	Tamaño objeto	Porcentaje de distancia
1	12.24	0	0	0.001361	99654	70 %
2	5.132	0	0	0.001188	33145	70 %
3	2.121	0	0	0.001310	41164	70 %

Cuadro 5.7: Experimentación de resultados de procedimiento de búsqueda con un porcentaje de distancia muy lejos del 70 %

No. Intento	Error de θ	Velocidad de Rueda Izquierda	Velocidad de Rueda Derecha	Error de reconocimiento del objeto	Tamaño objeto	Porcentaje de distancia
1	4.35	0	0	0.001273	48882	80 %
2	3.773	0	0	0.001909	48134	80 % *

Cuadro 5.8: Experimentación de resultados de procedimiento de colocación con un porcentaje de distancia muy lejos del 80 %

No. Intento	Error de θ	Velocidad de Rueda Izquierda	Velocidad de Rueda Derecha	Error de reconocimiento del objeto	Tamaño objeto	Porcentaje de distancia
1	1.279	0	0	0.0000094	35581	70 %
2	5.429	0	0	0.0011306	44536	70 %
3	10.421	0	0	0.0011751	49666	70 %
4	5.482	0	0	0.0011071	40215	70 %

Cuadro 5.9: Experimentación de resultados de procedimiento de colocación con un porcentaje de distancia muy lejos del 70 %

Capítulo 6

Experimentación y resultados

6.0.9. Conclusiones y trabajos futuros

Durante el desarrollo del controlador difuso fue necesario tener una visión amplia sobre el área de control clásico de un robot diferencial para darse cuenta que no siempre es viable el uso de métodos matemáticos clásicos, ya que si por una parte son modelos probados y sujetos a rigurosos procesos de comprobación que los hacen validos, también es cierto que no siempre se cuenta con los medios para tener acceso a tan formales parámetros, por lo que es necesario en algunos casos implementar métodos diferentes algo más flexibles o especulativos, tal es el caso del conjunto de ecuaciones que usamos para implementar las salidas del sistema difuso, cuya dificultad matemática consiste en partir a la mitad el máximo grado de velocidad, dar la mitad al vector de velocidad tangencial asociado al centro de masa del robot y la otra mitad destinarlo para marcar una diferencia entre las ruedas, sumando o restando de acuerdo a la orientación que se desea tomar, ponderada por un 0.5 para garantizar que a cada rueda le toque a su vez la mitad y no se rebase la velocidad máxima permitida, dejando claro que un modelo clásico bajo las condiciones adecuadas podría ser más eficiente.

Es importante elegir con cuidado la plataforma y el hardware con el que se va a trabajar, el desarrollo de modelos de control para robots móviles requiere tener presente el concepto de tiempo real, ya que la velocidad de respuesta que se obtenga definitivamente es el 75 % de éxito del trabajo realizado, ya que para nuestro caso hemos desarrollada una plataforma

de programación que permite al usuario evaluar a simple vista si las acciones de control son viables o si simplemente fallan.

Uno de los anhelos al comienzo de este trabajo fue buscar robustez ante perturbaciones del medio, que tienen que ver con accidentes físicos, tales como el rose excesivo de las llantas del móvil, problemas de desnivelación de una llanta o brincos que desajustaran la ruta ya descrita por el móvil para llegar a su meta. Puede entonces concluirse que tal vez no podemos anticiparnos a hechos como esos para hacer que parezca imperceptible pero si podemos ser lo suficientemente rápidos para corregir el error generado de forma eficiente.

6.0.10. Trabajos futuros

Probar la técnica de modelado difuso Takagi-sugeno (TS), así como el diseño de un controlador robusto para éste modelo. El obtener un modelo difuso del tipo TS del robot khepera. Es casi inmediato , debido a que ya se cuenta con los ragos de operación de cada parámetro, gracias a que ya tenemos un modelo del tipo Mamdani.

El modelo difuso TS también se describe por reglas del tipo SI-ENTONCES, las cuales representan relaciones lineales y locales de entrada-salida de modelos de sistemas no lineales. Dada la descripción del movimiento del robot khepera, podemos considerar restricciones y hacer adecuaciones en los parametros de la ecuación que describe este movimiento. Así, podemos obtener una aproximación a estas relaciones lineales de entrada y salida. La principal caracteristica de un modelo difuso TS, es su posibilidad de expresar las dinámicas locales de cada implicación difusa (regla), a través de un modelo lineal.

Bibliografía

[Vallejo 01] "Aprendizaje evolutivo de reglas fuzzy en un sistema clasificador modificado para control de agentes móviles" Universidad Politécnica de Valencia. Departamento de Informática de Sistemas y Computadores. Diciembre de 2004. Eric Vallejo Rodríguez

[Martínez 02] "Una arquitectura distribuida para el control de robots autónomos móviles" Universidad de Murcia España. Depto. Ingeniería de Información y las Comunicaciones. Enero de 2001. Humberto Martínez Barberá.

[Petrilli 03] "Control visual de un robot móvil khepera II". Centro de Investigación en computación, Instituto Politécnico Nacional. Laboratorio de Reconocimiento de Patrones. Alberto Elías Petrilli Barccló. Julio de 2007

[Casarrubias 04] "Generación de trayectorias para un robot móvil Khepera II usando técnicas de aprendizaje automático". Centro de Investigación en computación, Instituto Politécnico Nacional. Laboratorio de Reconocimiento de Patrones. Heriberto Casarrubias Vergas. Julio de 2007

[5] Enciclopedia metódica Larousse, tomo 5, 1991

[Isasi , Galván 06] Redes Neuronales Artificiales Un enfoque práctico, Pedro Isasi Viñuela, Inés Galván León. Departamento de Informática. Universidad Calos III de Madrid. Pearson Educación, S.A., Madrid (España) 2004.

[García Cabrera 07] "Introducción a la teoría de conjuntos y su aplicación a sistemas de control". Luis I. García Cabrera. Departamento de Ingeniería Eléctrica y Computadoras. Universidad de Puerto Rico.

[H. Sossa,08] Rasgos descriptores para el reconocimiento de objetos, Dr. Juan Humberto Sossa Azuela.

[Zadeh, 09] "Logical difusa ", 1965 por Lotfi A. Zadeh, profesor de la universidad de Berkeley California.

[Asimov,10] Isaac Asimov, I, Robot, Panther Science Fiction, 1968.

[Hu, 1962] M. K. Hu, "Visual Pattern Recognition by Moment Invariants" 1962.

[Jiménez,2008] Feature Selection based on Genetic Algorithm applied to Invariant Object Recognition. Computación Evolutiva: Congreso Mexicano de Computación Evolutiva, COMCEV08. Pag. 55-60. ISBN. Centro de Investigación en Computación. IPN México D.F. 2008, MC. Julio Fernando Jiménez Vielma

[Mahalanobis,2008] Wikipedia. La enciclopedia libre. Prasanta Chandra Mahalanobis [En línea]. Disponible en <http://es.wikipedia.org/wiki/Prasanta_nobis> [Consultado 20 de Junio de 2008].

[Roncagliolo,01] Proceso Digital de Imágenes [En línea]. Disponible en http://prontus.uv.cl/pubacademica/pubprofesores/r/pubroncagliolopablo/site/artic [consultado el 1 Agosto de 2008] Página 9. por Pablo Roncagliolo B.

[Roncagliolo,02] Ejemplo práctico de función de segmentación de color utilizando la distancia de Mahalanobis [En línea]. Disponible en <http://www.elo.utfsm.cl/~elo328/PDI14_EjemploMahalanobis.pdf>[consultado el 1 Agosto de 2008] por Pablo Roncagliolo B.