



Institución Universitaria

Metodología de Navegación de robots móviles para detección de vanos y discontinuidades en su trayectoria

Luis Fernando Ortiz Arroyave

Instituto Tecnológico Metropolitano

Facultad de Ingeniería

Medellín, Colombia

2019

Metodología de Navegación de robots móviles para detección de vanos y discontinuidades en su trayectoria

Luis Fernando Ortiz Arroyave



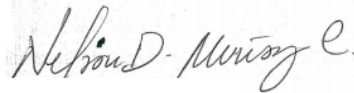
Tesis presentada como requisito parcial para optar al título de:

Magíster en Automatización y Control



Director:

Magíster Henry Mauricio Vásquez



Codirector:

Magíster Nelson David Muñoz

Línea de Investigación:

Sistemas de control y Robótica

Grupo de Investigación:

Automática, electrónica y ciencias computacionales

Instituto Tecnológico Metropolitano

Facultad de Ingeniería

Medellín, Colombia

2019

Dedicatoria

A mi madre Beatriz Arroyave y mi hermana Elizabeth Ortiz, quienes fueron grandes pilares para poder sacar este logro adelante

A mi padre Luis Fernando Ortiz Acevedo, aunque ya no este entre nosotros, agradezco que hayas hecho de mí una persona con buenos principios y valores

Agradecimientos

A mi asesor de tesis, el Msc Mauricio Vásquez, por el compromiso y apoyo en muchas de las etapas de desarrollo de la tesis

A mi asesor y codirector de tesis, el Msc Nelson David Muñoz por la confianza, compromiso y apoyo para la realización final de este trabajo, Además de sus consejos profesionales y sugerencias en el área de control en robótica móvil

A mis compañeros de generación Joe Alexander Jiménez y María Ruth Bonilla por su buena disposición y gran ayuda

RESUMEN

En la presente tesis de maestría, se pretende plantear el estudio e implementación de 2 estrategias de navegación autónoma que permitan a una plataforma robótica llegar al cumplimiento de la tarea asignada en entornos interiores estáticos, incluyendo trabajar con algunas discontinuidades, tales como puertas y/o pasillos. La primera estrategia que se evaluara será un sistema híbrido, teniendo en cuenta un sistema neuro-difuso, donde la parte de redes neuronales en combinación con comportamientos reactivos estimaran una decisión en cuanto a la orientación, y la parte difusa entregara mediante el uso de reglas, la velocidad a seguir por la plataforma, teniendo en cuenta las restricciones físicas y mecánicas del robot, el segundo método a analizar estará basado en estrategias estadísticas y/o probabilísticas, donde se enfocara en redes bayesianas. Se elaboraran pruebas en diversos entornos y cada una con diversos niveles de complejidad, donde datos representativos podrán ser comparados con métricas de desempeño y así poder establecer en alguna medida la mejor opción.

Palabras clave: Robótica móvil, Métricas de Seguridad, Navegación Autónoma, Estrategia Neuro-difusa, Estrategia Estadística, discontinuidades, pasillos y puertas

Abstract

In this master thesis, intends to present the study and implementation of 2 autonomous navigation strategies that lead to a robotic platform to fulfilling the task assigned in normal indoor environments, including work with some discontinuities, such as Corridor or hallways. The first strategy to be evaluated will be a hybrid system, taking into account a neuronal-fuzzy system, where the part of neuronal networks in combination with reactive behaviors estimate a decision on orientation, and the fuzzy logic part delivered through the use of rules , The speed to be followed by the platform, taking into account the physical and mechanical constraints of the robot, the second method to analyze will be based on statistical and / or probabilistic strategies, where it will focus on Bayesian networks. Tests will be developed in different environments and each with different levels of complexity, where representative data can be compared with performance metrics and thus be able to establish to some extent the best option.

Keywords: Mobile Robotics, Security Metrics, Autonomous Navigation, Neuronal-fuzzy Strategy, Statistical Strategy, Discontinuities, Corridor and Gates

Pág.

Contenido

RESUMEN	VII
LISTA DE FIGURAS	XI
LISTA DE TABLAS	XIII
INTRODUCCIÓN	15
Hipótesis.....	17
Objetivos.....	17
Alcances, Contribuciones y Limitaciones	18
Productos	19
Metodología.....	20
CAPITULO 1: MARCO TEORICO Y ESTADO DEL ARTE.....	23
1. NAVEGACIÓN AUTÓNOMA EN ROBÓTICA MÓVIL	23
1.1 TÉCNICAS DE NAVEGACIÓN.....	25
1.1.1 Auto-localización odometría.....	27
1.1.2 Dead-reckoning	28
1.1.3 Odometría-sensores inerciales	28
1.1.4 Odometría-IMU.....	28
1.1.5 Odometría-GPS.....	28
1.1.6 Odometría - Visión Artificial	29
1.2 ERRORES	29
1.3 HITOS Y DISCONTINUIDADES.....	31
1.4 MODELO CINEMÁTICO	34
1.5 SIMULADOR SRM.....	37
1.6 ROBOT GIRAA 02	40
1.7 MÉTRICAS O ÍNDICES DE DESEMPEÑO	41
1.7.1 Métricas de Seguridad.....	42
1.7.2 Métricas Dimensionales.....	42
1.7.3 Métricas de Suavidad	43
2. CAPITULO 2: ARQUITECTURA DEL SISTEMA NEURODIFUSO.....	45
2.1 INTRODUCCIÓN	45
2.4 COMPORTAMIENTOS REACTIVOS.....	49
a. Ir hacia la Meta	49
b. Repulsión de objetos	51
c. Búsqueda de espacios libres:	54
d. Seguimiento de pared:.....	56
2.5 SISTEMA DELIBERATIVO – SUPERVISOR DE FUSIÓN: RED NEURONAL (RNA)	61
2.5.1 Aprendizaje de la red.....	65
2.6 ANALISIS DE RELEVANCIA.....	68
2.7 SISTEMA CONTROL VELOCIDAD:.....	70
2.7.1 Lineal.....	70
2.7.2 Lógica difusa o borrosa	70
2.7.3 Obstáculos en zona de colisión	74

2.7.4	Obstáculos o Hitos muy cercanos	74
2.7.5	Obstáculos cercanos.....	75
2.7.6	Obstáculos alejados o no presentes.....	75
2.7.7	Respecto al Motor de Inferencia:.....	77
2.7.8	Bloque de Defuzificación o desborrador.....	78
3.	CAPÍTULO 3 ARQUITECTURA ESTADISTICO_PROBABILISTICO	81
3.1	INTRODUCCIÓN.....	81
3.2	METODOLOGIA PROPUESTA	81
3.2.1	Cuantización y agrupamiento de vectores.....	81
3.2.2	Método de Clasificación	83
3.2.3	Modelo Oculto de Markov (HMM).....	83
3.3	PROBLEMAS FUNDAMENTALES CON HMM	87
3.3.1	Solución al problema 1	88
3.3.2	Solución al problema 2.....	89
3.3.3	Solución al problema 3: Aprendizaje del Modelo	91
3.4	APRENDIZAJE	92
3.5	GENERACION DE RUTA	93
3.6	SOFTWARE UTILIZADO PARA EL DESARROLLO	93
3.6.1	Simulación.....	93
3.6.2	Robot Real	93
4.	CAPÍTULO 4 ANÁLISIS DE RESULTADOS.....	95
5.	CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES	105
5.1	CONCLUSIONES	105
5.2	RECOMENDACIONES.....	106
	ANEXO A: DATOS ENTRENAMIENTO	107
	ANEXO B: REDUCCIÓN DE CARACTERÍSTICAS	113
	Bibliografía.....	99

LISTA DE FIGURAS

FIG. I 1: Fotografía de largar exposición que muestra el patrón ineficiente de navegación que describe una aspiradora robot Roomba al limpiar	16
fig. I 2: Diagrama de la Metodología	20
Fig. 1. 1: Sistemas de coordenadas globales y local asociado al robot	34
Fig. 1. 2: Robot en corredor.....	35
Fig. 1. 3: Referencia de las distancias para el control por seguimiento de pared.....	36
Fig. 1. 4: Simulador SRM	38
Fig. 1. 5: Interfaz Usuario SRM	39
Fig. 1. 6: Robot GIRAA 02.....	40
Fig. 1. 7: Estructura Hardware Software GIRAA 02	41
Fig. 2. 1: Arquitectura de control.....	47
Fig. 2. 2: Hacia la Meta.....	50
Fig. 2. 3: Diagrama de Bloques evasión de Obstáculos.....	51
Fig. 2. 4: Fuerzas Repulsión de objetos	52
Fig. 2. 5: Repulsión de objetos	54
Fig. 2. 6: Búsqueda de espacio Libre	55
Fig. 2. 7: Convolucion Usada para calcular comportamiento Zonas Libres.....	56
Fig. 2. 8: Seguimiento de Pared	56
Fig. 2. 9: Distribución de los sensores en la plataforma.....	57
Fig. 2. 10: Diagrama flujo algoritmo seguimiento de pared izquierda.....	58
Fig. 2. 11: Ejemplo de un mínimo local (OLLERO, 1995)	60
Fig. 2. 12: Configuración de las capas en el Robot.....	63
Fig. 2. 13: Topología de la red Neuronal desarrollada	64
Fig. 2. 14: Función principal de la Red Neuronal en el Robot	64
Fig. 2. 15: Entornos para aprendizaje, entrenamiento y pruebas de la Red Neuronal....	65
Fig. 2. 16 Sectorización del Entorno	66
Fig. 2. 17: Esquema de Inferencia Difusa	71
Fig. 2. 18: Ejemplos de Funciones de Pertenencia.....	72
Fig. 2. 19: Variable Lingüística Velocidad Lineal del Robot	73
Fig. 2. 20: Zona de seguridad de la plataforma.....	74
Fig. 2. 21: Ejemplo de regla difusa implementada	78
Fig. 3. 1: Diagrama de bloques de un sistema de estimación.	81
Fig. 3. 2: Topología HMM Ergódica	85
Fig. 3. 3: Arquitectura final método estocástico	87
Fig. 3. 4: Esquema que simboliza una iteración en el cálculo recursivo de la variable hacia delante.....	88
Tabla 4. 1 Entorno de Simulación 1.....	97
Tabla 4. 2: Datos de inicio y llegada simulación 1	98
Tabla 4. 3 Entorno de Simulación 2.....	99

Tabla 4. 4 Datos de inicio y llegada simulación 2	100
Tabla 4. 3: Resultados métricas sobre entorno de trabajo 3.....	101
Fig. A.2. 1: Simplificación 2 de datos	113
Fig. A.2. 2: Simplificación por Rango en las Variables Lingüísticas.....	114
Fig. A.2. 3: Datos mayores a 19 Normalizados	115
Fig. A.2. 4: Simplificación 3	116
Fig. A.2. 5: Valores Cercanos dentro de un mismo	117

LISTA DE TABLAS

	Pág.
TABLA I. 1: Elementos de Cumplimiento de Objetivos	19
Tabla 1. 1: Algunos de los artículos referente a discontinuidades en la robótica móvil ..	31
Tabla 2. 1: Comportamientos predominantes en cada Sector	68
Tabla 2. 2: Almacenamiento de Datos para validación de entrenamiento de la RN	69
Tabla 4. 1 Entorno de Simulación 1	97
Tabla 4. 2: Datos de inicio y llegada simulación 1	98
Tabla 4. 3 Entorno de Simulación 2.....	99
Tabla 4. 4 Datos de inicio y llegada simulación 2	100
Tabla 4. 5: Resultados métricas sobre entorno de trabajo 3.....	101
Tabla 4. 6: Datos de inicio y llegada simulación 3	102
Tabla 4. 7: Datos me Métricas simulación 4	103
Tabla 4. 8: Datos de inicio y llegada simulación 4	104
Tabla A1 1: Datos para entrenamiento	107
Tabla A.2. 1 Variables Lingüísticas.....	114

Lista de Símbolos y abreviaturas

Abreviaturas

NOTACION	SIGNIFICADO
E_d	Error por Diámetros de Ruedas desiguales
RNA	Redes Neuronales
SRM	Multi-Robot Simulator
HMM	Modelo oculto de Markov
EM	Expectation Modificación
FIS	Fuzzy Inference Solve- Motor de Inferencia difusa
CGt	Centro de gravedad total

INTRODUCCIÓN¹

Planteamiento del problema

El desarrollo de un sistema de navegación autónomo es un área que se ha venido trabajando desde el inicio de la robótica móvil, sin embargo, sigue siendo un campo abierto de investigación donde todavía se necesita lograr técnicas generales y robustas que pueden resolver las incertidumbres y la dinámica de los entornos por donde debe transitar el robot; que, dicho sea de paso, se vuelven cada vez más exigentes en cuanto a capacidades de navegación. La planificación suele ser considerada como un acto fuera de línea, y cuando el requerimiento implica la interacción con el mundo físico y el movimiento en entornos humanos, más otros factores como la localización inexacta, el conocimiento incompleto del mundo, el modelo del robot, y obstáculos inesperados, todo esto da a entender que puede haber una diferencia entre la planificación de un movimiento y ejecutarlo, es decir, estos problemas pueden invalidar el plan antes de que el robot sea capaz de llegar a la meta. (Velásquez, 2015).

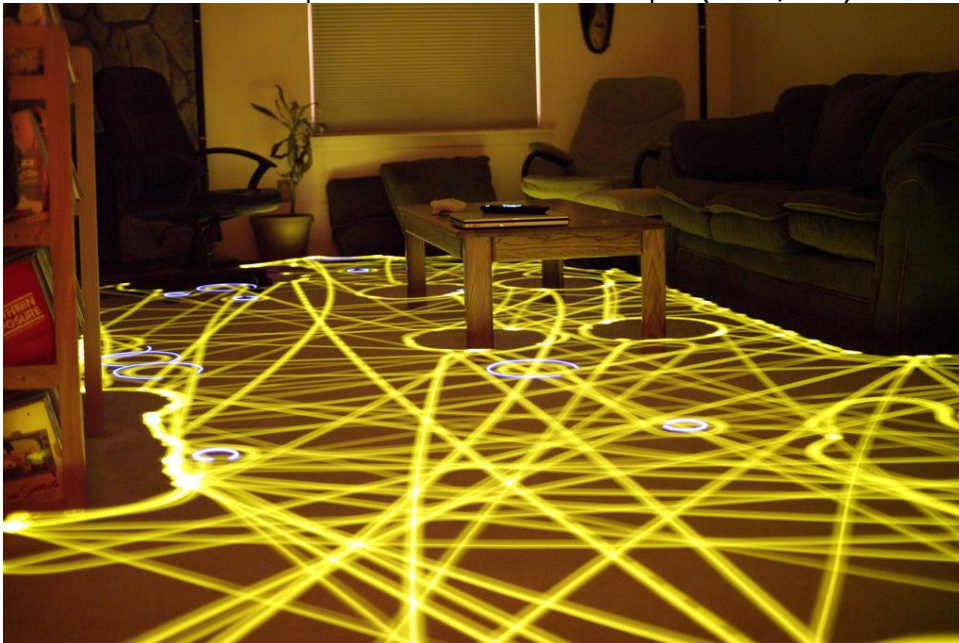
Para misiones de navegación autónoma donde se busca que un robot vaya desde un punto inicial hasta un punto final, se han desarrollado diversos métodos usando varias técnicas y nuevas tecnologías. La mayoría de esas soluciones hacen uso de sensores modernos y redundantes como GPS, IMU, lidar, laser, cámaras, etc., y algoritmos que demandan alto costo computacional y alto nivel deliberativo, que a su vez necesitan amplios recursos computacionales (RAM, tarjetas gráficas, procesadores de alto desempeño, coprocesadores matemáticos, etc.) tales como el mapeo, auto localización local o global, planificación de trayectorias, técnicas de SLAM, visión artificial, entre otros (Pandey, 2017). Este tipo de soluciones no son viables en aplicaciones de bajo costo,

¹ Un documento relacionado con este proyecto se han publicado en un artículo en revista indexada (Ortiz, 2018)

como los robots de consumo masivo (aspiradoras o trapeadoras robotizadas, robots académicos, etc.) donde no se cuenta con sensores de tecnología de punta, ni amplia capacidad computacional. Este tipo de robots generalmente solo se programan con estrategias de navegación reactivas, carecen de planificación global que le permita realizar tareas complejas y suelen ser ineficaces e ineficientes en consumo energético. Para un caso especial de estrategias reactivas como lo es campos potenciales, la presencia de vanos o discontinuidades en las paredes no causa factores de repulsión, por el contrario puede que sea interpretado como un área deseable para la navegación (Correa, 2013), y en muchas otras estrategias de navegación donde los movimientos o toma de decisiones para evitar obstáculos locales están basados en las medidas que se obtengan de los sensores, a la larga puede convertirse en un mínimo local, que se produce cuando un robot está navegando hacia un objetivo y este queda oscilante en presencia de un obstáculo que bloquea su misión.

Además de lo anterior, también se tiene la incertidumbre de cómo lograr combinar cada uno de los comportamientos individuales para producir comportamientos emergentes para poder llevar a cabo una orden u acción, lo que se conoce como coordinación de comportamientos (Herrero Reder, 2015).

FIG. I 1: Fotografía de larga exposición que muestra el patrón ineficiente de navegación que describe una aspiradora robot Roomba al limpiar (Bartle, 2009)



En consecuencia, se identifica el problema de abordar soluciones de navegación autónoma, in-door, con capacidad de evasión de obstáculos, vanos y discontinuidades en su trayectoria, útiles para aplicaciones con robots móviles de bajo costo, es decir, robots móviles que no cuentan con sofisticados sistemas de localización, sensado y planificación.

Este trabajo está dividido en cinco secciones: la primera tratará los temas de marco teórico y estado del arte; la segunda mostrará la estrategia de navegación neuro-difuso; la tercera presentará la estrategia estadístico-probabilístico; la cuarta expondrá los resultados y el análisis de las pruebas de navegación; y por último, en la quinta, se expondrán las conclusiones de este trabajo.

Hipótesis

Enmarcados en el problema identificado se formuló la siguiente hipótesis: Una plataforma robótica móvil que carece de sofisticados sistemas de localización, sensado y planificación, con estrategias de navegación convencionales o estadísticas, en entornos estructurados, estáticos, in-door, puede detectar la presencia de obstáculos, vanos o discontinuidades en las paredes, para el cumplimiento de tareas como ir desde un punto inicial hasta uno final, siendo capaz de evadir eficientemente los obstáculos que se le presenten en el camino, realizando la labor asignada en el menor tiempo y con un uso más eficiente de energía, es decir, alcanzando un mejor desempeño.

Objetivos

Objetivo General

Proponer una estrategia de navegación que le permita a una plataforma robótica móvil la detección de vanos o discontinuidades en las paredes y la evasión de obstáculos en entornos cercanos que impidan el cumplimiento de una tarea asignada, teniendo en cuenta que los ambientes serán estructurados y estáticos.

Objetivos Específicos

- Examinar las diversas estrategias y técnicas de navegación autónoma en entornos estructurados y estáticos que puedan ser aplicadas a la plataforma robótica a utilizar.

- Implementar dos estrategias de navegación autónoma, una basada en métodos convencionales y otra en métodos estadísticos, teniendo en cuenta las restricciones físicas de la plataforma robótica.
- Validar el desempeño del algoritmo estadístico con respecto a las estrategias convencionales de navegación (neurodifusas-reactivas, etc.) y evaluar con base a métricas de desempeño.

Alcances, Contribuciones y Limitaciones

Se proponen dos estrategias de navegación autónoma desde un punto inicial hasta uno final, trayecto en el cual se podrán encontrar algunos hitos tales como obstáculos, pasillos, vanos y puertas. Estas estrategias son útiles para aplicaciones con robots móviles como aspiradoras o limpiadoras robotizadas, robots académicos, etc. Este tipo de robots, por ser equipos para consumo masivo, no cuentan con costosos dispositivos hardware de tecnología de punta, por el contrario, con el fin de bajar costos y ser competitivos en el mercado se fabrican con procesadores de gama media o baja y con sensores o actuadores de desempeño muy limitado. Es en este tipo de robots donde las estrategias de navegación, como la que se propone en este proyecto, cobran mayor vigencia y contribuyen en la solución eficiente de la navegación autónoma. Para otros tipos de robots más sofisticados, con equipos hardware de tecnología de punta, muy alto desempeño, y muy costosos, se tiene solucionado en alto grado el problema de la navegación autónoma.

Para el desarrollo de este proyecto se hace uso de entornos de prueba mediante simulación por computador, técnica aceptada entre la comunidad científica y académica, que permite evaluar el funcionamiento de un robot móvil simulado con varias estrategias de navegación, además, se aplican métricas de desempeño con el fin de comparar el rendimiento de dichas estrategias, se evalúan la calidad del recorrido, la seguridad de la plataforma, el consumo de energía y el cumplimiento de la tarea asignada.

Se debe tener en cuenta que este proyecto está planteado para ambientes In-door, estructurados, planos y con obstáculos estáticos, sin embargo, los obstáculos que

interactúan en los escenarios se les puede cambiar su posición espacial, es decir, que aunque el ambiente es estático permite el intercambio espacial de los obstáculos, y el único que cambia su posición en el transcurso de las pruebas es el robot. Asimismo, se aclara que los obstáculos que se empleará para las pruebas serán de formas geométricas, como paredes, cubos y cilindros sólidos.

Productos

A continuación, se especifica con cuáles elementos se le dio cumplimiento a cada objetivo específico

TABLA I. 1: Elementos de Cumplimiento de Objetivos

Objetivo Especifico	Elemento de Cumplimiento
Examinar las diversas estrategias y técnicas de navegación autónoma en entornos estructurados y estáticos que puedan ser aplicadas a la plataforma robótica a utilizar.	Estudio de estrategias de navegación autónoma para un robot móvil que dispone de señales de 8 sensores infrarrojos de distancia, 1 sensor de orientación y 1 sensor de odometría en el plano x-y. Además, el robot tiene forma circular y locomoción diferencial.
Implementar dos estrategias de navegación autónoma, una basada en métodos convencionales y otra en métodos estadísticos, teniendo en cuenta las restricciones físicas de la plataforma robótica.	Implementación de los algoritmos de navegación autónoma. El primero neurodifuso, donde el sistema basado en redes neuronales pueda proveer un perfil de velocidad angular para un controlador de velocidad tipo fuzzy, el cual permite que los actuadores eléctricos alcancen los perfiles de velocidad requeridos. Mientras que el segundo, un control estadístico basado en el modelo oculto de Markov, el cual usa 4 comportamientos reactivos para ponderar la navegación siendo estos Ir hacia la meta, evasión de obstáculos y seguimiento de pared derecha e izquierda.
Validar el desempeño del algoritmo	Evaluación de trayectorias basándose en

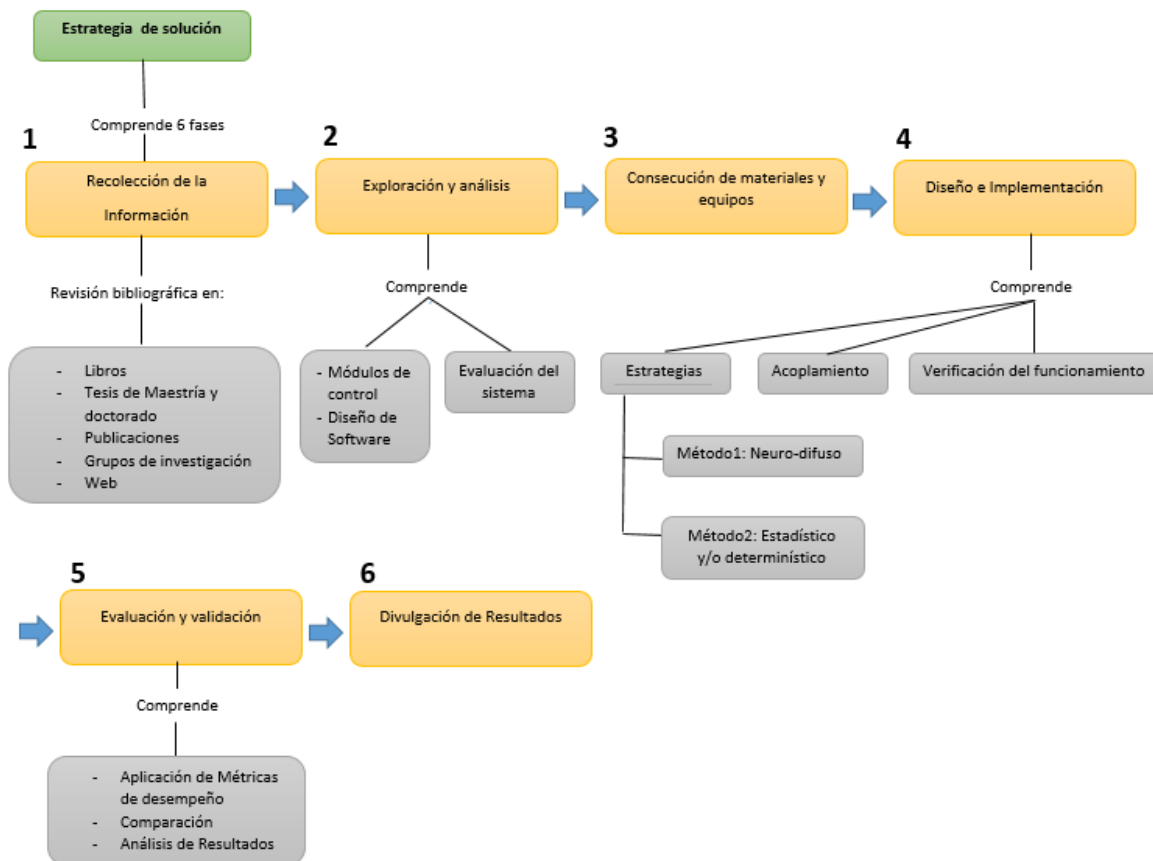
<p>estadístico con respecto a las estrategias convencionales de navegación (neurodifusas-reactivas, etc.) y evaluar con base a métricas de desempeño.</p>	<p>un entorno estructurado con criterios estándar, métricas de desempeño reportados en la literatura científica y pruebas de similitud en igualdad de condiciones mediante software de simulación por computador.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Metodología

En este trabajo se emplea una metodología basada en el diseño e implementación de algoritmos software los cuales son sometidos a múltiples pruebas mediante software de simulación para su validación.

El proyecto se desarrolla en 6 fases, de acuerdo con el diagrama presentado.

fig. I 2: Diagrama de la Metodología



Recolección de la información:

Se realiza un estudio de las características de la plataforma robótica móvil. Teniendo en cuenta que este proyecto se orienta en torno a la navegación de un robot móvil el cual posee unas características muy definidas, se analizan diferentes estrategias de navegación autónoma propuestos en la literatura especializada. Para la búsqueda de información se hace uso de los recursos de las bibliotecas locales universitarias, búsqueda en bases de datos especializadas e internet.

Exploración y análisis:

Esta etapa permite seleccionar dentro de la gama de estrategias de navegación, las dos que se implementan en el robot, además de los rediseños y ajustes necesarios para su adaptación a los objetivos del proyecto.

Consecución de materiales y equipos:

Para el desarrollo del proyecto se requiere un computador con dedicación exclusiva y unas características mínimas de hardware y software, ya que se hace uso del software de simulación, diseño y cálculos matemáticos MATLAB de MathWorks. Este software es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio basado en la sintaxis de lenguaje c, además, es un software ampliamente usado por ingenieros y científicos, especialmente porque posee varias librerías y software adicional con funciones especiales, denominados Toolbox, en este caso se hace uso de los Toolbox de lógica difusa, redes neuronales, estadística, robótica móvil.

Diseño e implementación:

Una frase célebre de la investigación científica es “estar subido a hombros de gigantes”, las estrategias de navegación autónoma que se proponen en este punto se basan en trabajos previos ya publicados y estudiados en la etapa previa de revisión bibliográfica, sin embargo, los algoritmos tienen componentes de diseño propio, debido a las particularidades de la plataforma robótica móvil, y a criterios de diseño propios que permiten integrar la funcionalidad del robot, su puesta en funcionamiento a nivel de simulación y mejorar los resultados del proyecto.

Evaluación y Validación:

En esta etapa se emplean las herramientas software que facilitan la simulación y depuración de algoritmos del sistema robotizado, se realiza diferentes pruebas a nivel de simulación para obtener datos de comparación, se contrastarán los resultados de la simulación de las dos estrategias de navegación, ante los mismos escenarios con el fin de establecer un marco de referencia y comparación único para las pruebas. Además, se hace uso de métricas de desempeño usadas en diferentes reportes de investigación sobre navegación de robots móviles.

Divulgación de resultados:

Se realiza al menos un artículo para revista indexada y el informe del trabajo de investigación, que estará disponible para su consulta en la biblioteca institucional.

CAPITULO 1: MARCO TEORICO Y ESTADO DEL ARTE

1. NAVEGACIÓN AUTÓNOMA EN ROBÓTICA MÓVIL

La robótica es una de las áreas que más ha estado en apogeo durante los últimos años, dado que aporta grandes avances y mejoras en lo referente a la realización de actividades que son desempeñadas por los humanos, pero que son repetitivas, tediosas, difíciles o peligrosas. Se define un robot como: “un sistema electromecánico, dotado de sensores y actuadores para realizar actividades similares a las realizadas por seres vivos, con un sistema de control y capacidad de decisión que le permite realizar tareas automáticas o autónomas, dependiendo de su grado de versatilidad y complejidad” (Londoño N. , 2009).

La robótica se ha separado en 2 grandes ramas, siendo estas la robótica manipuladora y la robótica móvil, donde cada una de ellas a su vez se puede subdividir de acuerdo a características de control, de construcción, físicas, entre otras

Dentro de la robótica móvil un elemento que es relevante es la autonomía, la cual se define para robots autónomos como “sistemas completos que operan eficientemente en entornos complejos sin necesidad de estar constantemente guiados y controlados por operadores humanos” (Reale, 2014).

Para que un robot pueda navegar, este debe tener la capacidad de obtener información del medio ambiente a través de sus sensores, procesar estos datos y tomar decisiones para poder moverse de forma segura en su entorno (Sales, 2011), si además se requiere que el sistema cuente con cierto grado de autonomía es necesario que el robot sea capaz de localizarse y planificar su tarea.

Para que una plataforma pueda navegar, se debe dar solución a 4 elementos básicos:

¿En dónde estoy? También llamado proceso de localización, este se puede entender como el mecanismo que usa un robot para extraer su ubicación dentro del lugar donde se

encuentra. En la actualidad hay básicamente 2 formas a través de la cual puede realizar esta tarea:

- Local: usando técnicas de odometría para brindarle la posición al robot y a partir de acá se inicia la misión a realizar
- Global: Usando sistemas como el GPS donde se le va mostrando al robot la ubicación del mismo.

¿Hacia dónde voy? el objetivo normalmente es determinado por un humano, un planificador es el encargado de realizar esta tarea.

¿Por dónde he estado? Para algunas aplicaciones tales como es SLAM, reconocimiento de espacios, exploración, etc., a medida que se exploran caminos es importante ir registrando el recorrido, el cual puede llegar a ser uno de los objetivos.

¿Cuál es el mejor camino? esta es el área de la navegación que ha recibido la mayor atención. La planificación de recorridos se divide en dos categorías: cualitativa y cuantitativa (o métrica). (Muñoz V. F., 2019).

De acuerdo al criterios de navegación surgen 4 subdivisiones en la robótica móvil, siendo estos: en primer lugar, el sistema deliberativo o jerárquico (Cerrada C. C., 2013) donde se requiere de una planificación previa para la realización de una acción, en un segundo escalón aparece el sistemas reactivo, donde solamente se usan la localización o posición donde se encuentren y a partir de ello, cuáles serán las acciones a tomar para elaborar una tarea (Calisi, 2009). A partir de la combinación de los 2 sistemas mencionados anteriormente aparecen los Métodos híbridos, donde el sistema deliberativo está por encima del sistema reactivo, el primero es quien planea y el segundo se encarga de la evasión de obstáculos (Oliva, 2011) y por ultimo están los sistemas con aprendizaje, donde aparece un nuevo ítem para tener en cuenta como es el aprendizaje.

1.1 TÉCNICAS DE NAVEGACIÓN

En lo referente a la navegación, el robot deberá ser capaz de reconocer algunos puntos significativos en el medio ambiente y decidir cuándo y cómo proceder con el fin de seguir recto, girar a la izquierda o a la derecha para lo que comúnmente se usan comportamientos reactivos, incluso cuando todas estas posibilidades se detectan simultáneamente (por ejemplo, intersecciones) (Oliva, 2011).

En cuanto al uso de inteligencia artificial, se pueden encontrar las redes neuronales las cuales tienen diversas aplicaciones en esta área, dentro de ellas la identificación de hitos como paredes y pasillos (Correa, 2013), dado que permiten la generalización de conocimiento y el ajuste de sus salidas a muchas entradas, incluso cuando algunas entradas no se les ha enseñado o entrenado explícitamente a la red; También se aplica en sistemas donde la parte deliberativa genera una serie de puntos de chequeo. Los sistemas difusos también resultan ser beneficiosos en lo referente a la navegación autónoma de los robots, dado que permiten variar la velocidad de navegación del robot móvil, en función de las entradas de control que logran la evasión de obstáculos (Correa, 2013). También con este sistema se puede trabajar bajo una teoría de agentes en donde se trabaja con elementos de control reactivo la parte de planificación, localización y control a bajo nivel (Acosta, 2011).

Cuando se trabajan con aplicaciones probabilísticas, este presenta 2 enfoques principalmente: el primero de ellos se presenta en la percepción, donde se considera la incertidumbre de la medida que obtienen los sensores, medio ambiente, entorno, etc. y la segunda se da en el propio control, que se encarga de la navegación autónoma donde se considera la situación actual del entorno y una o varias proyectadas (González, 2015), además de ser bastante precisos cuando se trabaja en entornos muy grandes y no se cuenta con un dispositivo de posicionamiento global, tales como GPS (Parrón, 2009), dentro de los métodos que se han usado están:

Procesos de decisión de Markov parcialmente observables (POMDPs) el cual funciona de manera óptima en condiciones de incertidumbre en tareas de decisión secuencial (Hortal, 2017).

Cuando se busca una construcción geométrica que permita construir una partición o un esquema del plano euclidiano o diagramar el contorno interior de una edificación, los diagramas de Voronoi (Rodríguez E. , 2010) se encargan de codificar los datos de cercanía entre componentes, para el área de enfoque de que se analiza, es observar la distancia entre el robot y los obstáculos y como estrategia se busca ampliar la distancia entre ambos (Ortega, 2010).

Los grafos de visibilidad, consiste en la escogencia de 2 puntos del escenario, y estos tienen que unirse mediante una línea recta sin que exista obstáculo alguno, escogiendo como nodo inicial el punto inicial del recorrido y nodo final la meta del robot (Guzmán, 2012). Los sistemas basados en nodos, requieren disponer de un conjunto de conductas individuales tales como:

- 1) conducta de seguimiento de línea.
- 2) conducta de detección de fin de línea.
- 3) conducta de cambio de línea para posicionarse correctamente en la cabecera de la siguiente línea.
- 4) conducta de detección y sorteo de obstáculos para evitar la colisión.
- 5) conducta de supervisión del nivel de las baterías, para poder tomar una decisión y dar una orientación a la plataforma (Cerrada C. C., 2013).

Las Máquinas de estado finito (FSM) permiten describir el sistema fácilmente como una secuencia de estados, teniendo en cuenta: entradas (sensores) que permite cambiar de un estado (situación) a otro, y también definir para cada estado una acción específica (acción motor). Así que para cada cambio de estado y el estado, el robot es capaz de reaccionar adecuadamente (Marino, 2009), cuando se habla de estado, se hace referencia a cada una de las posibles direcciones que pudiese tomar el robot.

Para aquellos casos en donde se trabaja con navegación basada en comportamientos, donde muchos de los investigadores tratan un comportamiento como un conjunto de trayectorias compuestas, cuyo fin es el de expresar el control del robot. Una vez que se especifica, se puede ejecutar tantas veces como sea necesario. Los comportamientos pueden ser expresados a nivel de motor, habilidad o tarea (Miranda Cruz, 2011). Aparecen aplicaciones a nivel de Comportamientos reactivos se pueden encontrar

algunos eventos como seguimiento de línea, seguimiento de pared derecha, seguimiento de pared izquierda, Bifurcaciones y repulsión de objetos, teniendo mínimamente algunas características, tales como: la percepción y la acción están estrechamente relacionadas, los comportamientos se definen a partir de los objetivos para los cuales se construye la plataforma y comportamientos complejos surgen de la combinación de comportamientos individuales más simples. Para sistemas híbridos, se puede resaltar que se destacan 2 métodos de control de comportamientos, siendo estos el competitivo y el cooperativo. Para este tipo de estrategia, se debe tener en cuenta que cuanto mejor sea el robot, mejor se ejecutaran los comportamientos, por ende estos comportamientos tienden a ser robustos, eficientes y adaptables a diferentes situaciones (Miranda Cruz, 2011).

La técnica de Campos potenciales, se basa netamente en comportamientos reactivos, la cual se enfoca en la planificación local en entornos desconocidos (Preciado, 2018), donde la plataforma es considerada como una partícula que está influenciada por un campo potencial artificial, donde esta se compone de un potencial que lo atrae hacia la meta y un potencial que hace que el robot se aleje de los obstáculos, donde el potencial resultante, se obtiene de la suma vectorial de ambos potenciales.

1.1.1 Auto-localización odometría

La auto-localización consiste en estimar la posición y la orientación del robot a partir de observaciones o mediciones que este esté realizando del entorno, para muchos de los casos se trabajan con técnicas odométricas sobre los movimientos propios del robot y con el uso de sensores de diversas tecnologías para obtener la información de posibles obstáculos cercanos.

Para lograr dar solución a los problemas de localización a un robot sobre su entorno, se han trabajado múltiples técnicas, en especial bajo los ítems mencionados en el párrafo anterior, entre ellas las que más han usado, son:

1.1.2 Dead-reckoning

Consiste en incorporar encoders en las llantas de la plataforma robótica y medir los pulsos entregados por dichos encoders y en base a ello calcular la distancia recorrida. Las principales desventajas de esta técnica son los errores acumulativos que se van adquiriendo en cada medida del encoder, y que ante un deslizamiento en las ruedas se va a presentar una falsa medida (Jiménez, 2015).

1.1.3 Odometría-sensores inerciales

Para tratar de corregir los errores acumulativos, se ha incluido en las plataformas robóticas el uso de sensores tales como giróscopos, acelerómetros, brújulas. Además de ello, esta tecnología ha permitido la corrección de errores no sistemáticos causados por irregularidades en el suelo (Boresntein, 1996).

1.1.4 Odometría-IMU

Sistema utilizado en plataformas que trabajan en 2D o 3D, dependiendo del entorno para el cual hayan sido diseñados (Lamon, 2003), aunque esta tecnología aun percibe algunos problemas con deslizamientos, presenta una gran ventaja a la hora de trabajar en entornos irregulares o desiguales.

1.1.5 Odometría-GPS

Se utiliza generalmente para sistemas de posicionamiento absolutos, con el fin de obtener mediciones más confiables y corrección de los errores acumulados. En algunos prototipos utilizan una mezcla de estos sistemas para compensar errores sistemáticos que puedan poseer ambas tecnologías (Parrón, 2009).

1.1.6 Odometría - Visión Artificial

Técnica empleada para generar sistemas de odometría visual (Yang, 1999), en donde se busca considerar posibles movimientos a partir de imágenes, que se puede presentar ya sea a nivel estéreo o monocular. En aplicaciones modernas y de gran impacto, como son los robots de la NASA Spirit y el Opportunity (Cheng, 2006), este tipo de tecnología ha sido trabajada para mantener la seguridad de las plataformas ante posibles obstáculos y en muchos casos para esquivar irregularidades en su ruta de exploración.

1.2 ERRORES

Dentro de los robots que manejan la odometría como sistema de auto localización se pueden encontrar varios tipos de errores, además de problemas causados por error en el contacto entre el robot y el suelo (Gómez, 2012), estos errores pueden ser clasificados así:

Errores sistemáticos: Causados por defectos en el diseño y la implementación mecánica propia del robot, en muchos casos suelen ser graves, dado que se acumulan constantemente, entre ellos se encuentran:

- Diámetros de Ruedas desiguales (E_d), influyen predominantemente sobre trayectos largos, dado que afecta directamente la estimación de la orientación del Robot, Esta falla se puede presentar por 2 fenómenos, el primero de ellos tiene que ver con el tipo de llanta y la forma en que fue manufacturada, el segundo de los casos forma parte de la integración de los parámetros de las ruedas al control de navegación del robot (Navarro García D. A., 2009), con el fin de evitar derrapes y deslizamientos por cambios de aceleración, Este error se intensifica si la distribución de carga del robot no es la correcta (Calderón García, 2016).

$$E_d = \frac{d_R}{d_L} \tag{1.1}$$

Donde d_R y d_L son los diámetros de las ruedas derecha e izquierda respectivamente, E_d debe tender a 1

- Diferencia entre la distancia entre ruedas teórica y la real (E_b): este elemento puede causar un error en la estimación del ángulo, en muchos de los casos, no se cumple la

condición ideal de un contacto puntual entre la rueda y el plano de desplazamiento, sino más bien que se trata de un área de contacto que generará en forma directa una incertidumbre en la base de giro real de la plataforma. ya que no es posible saber con exactitud el punto real de contacto, se considera que robots comerciales este error por lo general es del 1%. El factor E_b está dado por:

$$E_b = \frac{b_{real}}{b_{nominal}} \quad (1. 2)$$

Donde b_{real} es distancia actual de las base de las ruedas y $b_{nominal}$ es distancia nominal de la base de las ruedas.

- Ruedas mal alineadas.
- Resolución de los elementos como los encoders limitada: Presenta varios y variados inconvenientes principalmente en el derrape de las ruedas, las imperfecciones del acoplamiento de las ruedas a la superficie de rodamiento, y las no linealidades por desgaste de los engranajes.
- Desgaste no proporcional entre las llantas: Error que puede ser causado por diversos elementos, tales como: la no distribución adecuada de los elementos mecánicos y/o electrónicos en la plataforma, el no adecuado cálculo del centro de masa (Navarro García D. A., 2009) o en muchos casos, se puede asumir que la velocidad angular es igual, pero debido a los errores de diámetro de las llantas, ocasiona que el centro geométrico del vehículo no coincida con el centro instantáneo de rotación (CIR) al completar el giro. Esto ocasiona un desplazamiento lateral que puede ser ignorado ya que se anula en los giros restantes.

Errores no Sistemáticos: No son causados directamente por las propiedades cinemáticas de la plataforma, sino principalmente por la interacción en la navegación entre el robot y el suelo, No se provocan continuamente (depende del tiempo, tipo de uso y del entorno) por ende, son más complicados de medir y reparar. Entre estos errores se pueden encontrar.

- Navegación en suelos accidentales.
- Navegación sobre objetos inesperados.
- Deslizamiento de las ruedas (suelos resbaladizos, giros bruscos, interacción con objetos externos: Causa que las revoluciones de las ruedas no correspondan al desplazamiento real del móvil (Navarro García D. A., 2009).

- Dada la construcción de la plataforma, inclinaciones demasiado grandes.

1.3 HITOS Y DISCONTINUIDADES

Cuando los robots móviles autónomos navegan dentro de los ambientes interiores (por ejemplo, en edificios públicos o instalaciones industriales) deben estar dotados de la capacidad de moverse a lo largo de los pasillos, identificar vanos y puertas.

Dado el enfoque que tiene el presente trabajo, se mencionaran investigaciones que estarán enfocadas en la navegación a través de los pasillos y como tratar los movimientos a través de una puerta.

Tabla 1. 1: Algunos de los artículos referente a discontinuidades en la robótica móvil			
Articulo	Temática trabajada	Hito o discontinuidad analizada	Aspectos a mejorar
Guía de Personas con Robots Móviles Basado en la Interacción Humano-Robot. (Luna, 2018)	Se presenta el desarrollo e implementación de una nueva técnica de interacción humano-robot para la guía de personas a través de plataformas móviles con ruedas con el fin de facilitar la integración de los robots en la sociedad. La técnica de interacción propuesta consiste en un sistema de control que regule la velocidad del robot durante el recorrido tomando en cuenta la velocidad estimada del usuario en tiempo real	• Pasillos en laboratorios, museos, hospitales, otros	
Door Detection in 3D Colored Laser Scans for Autonomous Indoor Navigation (Quintana, 2016)	La técnica propuesta utiliza tanto la información geométrica (es decir, coordenadas XYZ) como el color (es decir, RGB / HSV), datos que son proporcionados por un escáner láser 3D y una cámara a color	•detección de puertas	
Vision-based adaptive assistance and haptic guidance for safe wheelchair corridor following	El objetivo de este proyecto es colocar un sistema de asistencia y guía adaptativo para evitar colisiones con la pared cuando un usuario maneja manualmente una silla	•control del usuario y del robot para el manejo de una silla de ruedas a	

(Vishnu, 2016)	de ruedas en un pasillo	través de pasillos	
Arquitectura y diseño de un sistema completo de navegación semántica. Descripción de su ontología y gestión de conocimiento (Jonathan, 2017)	Esta tesis aborda la idea subyacente de conseguir que un robot se comporte de manera semejante a como lo haría un ser humano, permitiéndole al robot una mayor comprensión del entorno y de los objetos que contiene	<ul style="list-style-type: none"> • Pasillos, los cuales se trabaja como figuras geométricas simples 	clasificación semántica del entorno
Topological Autonomous Navigation for Mobile Robots in Indoor environments using Ann and FSM (Oliva, 2011).	La aplicación del sistema propuesto es un robot de servicio de navegación en interiores, que puede moverse de forma autónoma, designado para la vigilancia y detección de situaciones anormales (por ejemplo, intrusiones, manchas de fuego).	<ul style="list-style-type: none"> • Seguir recto • Girar a la izquierda o a la derecha • Intersecciones 	Se tiene que usar un ANN muy robusto para tratar con entradas que posean un alto nivel de ruidos podrían conducir a una clasificación errónea y cambios de estado inadecuadas,
Control neuronal-difuso de navegación reactiva Continúa incluyendo la evasión de obstáculos Presentes y emergentes (Correa, 2013).	En este trabajo se le permite al robot que en ambientes estructurados y dinámicos, tenga la capacidad de variar su velocidad de desplazamiento, en virtud de la composición del entorno cercano y lejano,	<ul style="list-style-type: none"> • Obstáculos emergentes • Puertas 	Las soluciones planteadas requieren por obligación ejecutarse en el entorno para el cual fueron diseñados, en caso de necesitar implementar este tipo de sistemas en un entorno distinto se presenta la complejidad de tener que implementar todo el sistema guía y reprogramar al robot para el nuevo esquema.
A visual servoing approach for autonomous corridor following and doorway passing a wheelchair (Babel M. a., 2016).	Trabajo desarrollado para navegación autónoma en sillas de ruedas.	<ul style="list-style-type: none"> • Navegación por pasillos • Pasos de Puerta 	
Contribución a la auto localización de robots móviles basada en la fusión de información multisensorial (Navarro, 2009).	Trata sobre navegación en Hitos.	<ul style="list-style-type: none"> • Esquinas 	Los procesos altamente no lineales involucrados en la auto localización de un vehículo móvil son muy difíciles de modelar

Dentro de las áreas, donde se están aplicando las estrategias mencionadas y otras, están:

- a. Travesía, donde sobresalen los trabajos en navegación autónoma en sillas de ruedas, para aquellas personas que sufren de discapacidad severa, donde se le da gran relevancia a la seguridad, para evitar impactos o choques, algunos de estos proyectos son: (Quintana, 2016) para guiar personal en museos y hospitales, para manejo de sillas de ruedas a través de pasillos (Vishnu, 2016), proyecto europeo FP7 Radhar (Demeester, 2012) y el proyecto SYSSIASS (Kokosy, 2012), en algunos casos, estos proyectos también trabajan sobre el paso en puertas estrechas (Babel M. a., 2016).
- b. Labores de riego de insecticida en invernaderos, donde la salud de quienes manipulan estos químicos, podría verse afectada (Rodríguez F. S., 2016), aquí el recorrido óptimo es en zigzag, es decir, los pasos básicos del algoritmo de navegación serán avance en línea recta, giro a la derecha, avance, giro a la derecha, avance en línea recta y después se repetirá hacia la izquierda, ya que se optimiza la distancia recorrida y por lo tanto el tiempo de operación.
- c. En la agricultura de precisión donde se requiere disponer de un conjunto de conductas individuales tales como: 1) una conducta de seguimiento de línea, 2) una conducta de detección de fin de línea, y 3) una conducta de cambio de línea para posicionarse correctamente en la cabecera de la siguiente línea. Además debe estar dotado de 4) una conducta de detección y sorteo de obstáculos para evitar la colisión, y 5) una conducta de supervisión del nivel de las baterías (Cerrada C. C., 2013).
- d. Patrullaje: donde se busca identificar intrusos, posibles focos de fuego, etc. (Sales, 2011).
- e. Desactivación de artefactos explosivos (Cordero, 2011).
- f. Supervisión.
- g. Exploración de tuberías (Tobías, 2015).
- h. Exploración planetaria (Cheng, 2006), entre otros.

Con respecto al segundo hito que se analizó en el presente trabajo, esta los que se denomina el paso a través de puertas, para poder hacer un análisis con respecto a este tema, se deben vislumbrar varios aspectos, tales como:

- El estado de la puerta, es decir, si está abierta o cerrada.
- El proceso de toma de decisiones para pasar por la puerta, es decir, que puerta es la que se debe cruzar y cual no.

1.4 MODELO CINEMÁTICO

Se trabajó con una plataforma diferencial, que se describe mediante unas ecuaciones básicas para analizar su funcionamiento, para ello se considerara un $p(x, y)$ diferente al punto de inicio y al punto de llegada, teniendo como referencia un marco cartesiano fijo. El movimiento hacia la meta siempre estará dado bajo la combinación de 2 variables esencialmente, la velocidad angular ω y la velocidad lineal u (Dev, 1997), donde el conjunto de ecuaciones cinemáticas está dado por:

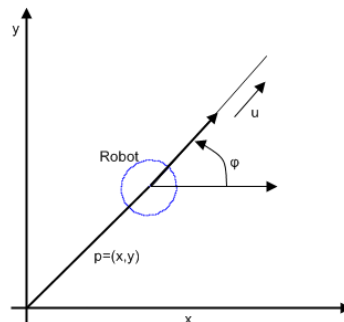
$$\dot{x} = u \cos \varphi \quad (1.3)$$

$$\dot{y} = u \sin \varphi \quad (1.4)$$

$$\dot{\varphi} = \omega \quad (1.5)$$

Donde u es la magnitud del vector de la velocidad lineal y φ se mide con respecto al eje x .

Fig. 1. 1: Sistemas de coordenadas globales y local asociado al robot

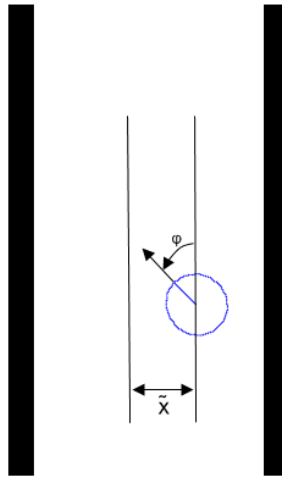


Es de anotar que ω siempre estará asociada a φ tal cual se ve en la ecuación 1.3 y que está siempre se mide en sentido contrario a las manecillas del reloj.

Para realizar una tarea de navegación en interiores, uno de los hitos o discontinuidades más comunes, son los pasillos o corredores, (este es uno de los ítems que se trabajó dentro de este proyecto), para poder analizar la forma en que se trabaja esta particularidad, se presentan las siguientes 2 formas:

1. Considerar el caso del robot que navega en un corredor, se definirán las siguientes variables, teniendo en cuenta que están referenciadas al entorno y principalmente al pasillo serán \tilde{x} y φ , donde \tilde{x} representa la desviación del robot respecto a la línea central del pasillo o a cualquier línea del corredor deseada, mientras tanto, φ representa la desviación angular relacionada con el eje del corredor tomado como referencia, tal cual se muestra en la Fig. 1.2.

Fig. 1. 2: Robot en corredor



$$\dot{\tilde{x}} = u \sin \varphi \tag{1. 6}$$

Para este tipo de análisis, lo que se busca es diseñar acciones de control $\omega(t)$ y $u(t)$ de manera que los errores de control $\tilde{x}(t)$ y $\varphi(t)$ tiendan asintóticamente a cero.

$$u = u(\varphi, \tilde{x}), u \geq 0 \tag{1. 7}$$

$$\omega = -k_{\varphi}(\varphi)\varphi - k_{\tilde{x}}(\tilde{x})\tilde{x}u \frac{\sin \varphi}{\varphi} \tag{1. 8}$$

Donde $k_\varphi(\varphi)$ y $k_{\tilde{x}}(\tilde{x})$ son funciones positivas adecuadamente seleccionadas para evitar la saturación de la orden de control de ω .

Para este caso, la velocidad lineal se puede tomar como constante, teniendo como referencia $u(\varphi, \tilde{x}) = u_d$, donde u_d representa el valor máximo deseable para la velocidad lineal u . Con ello se puede lograr un comportamiento equilibrado y suave para el robot en función de los errores de control, considerando la función.

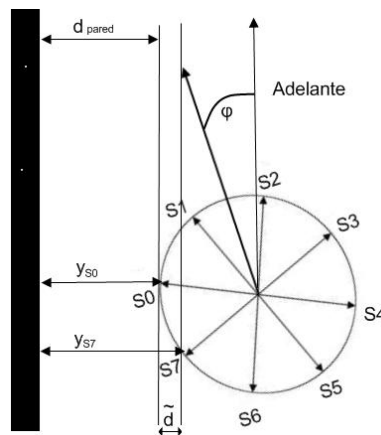
$$u(\varphi, \tilde{x}) = (u_d - k|\tilde{x}|) \cos \varphi, \quad k > 0, \quad (1.9)$$

k se toma tal que $u(\varphi, \tilde{x}) > 0$ para todo φ y \tilde{x} , dentro de un rango físicamente válido.

Para ver el desarrollo completo de la deducción de estas ecuaciones (Carelli R. a., 2003).

2. Otra de las estrategias utilizadas para la navegación de un robot en un corredor es permitir el seguimiento de pared, esto debido a que en un pasillo está compuesto generalmente por 2 paredes. En este caso, las variables se definen con respecto a la pared como φ y \tilde{d} , donde φ es la desviación angular relativa a la línea de pared, y \tilde{d} representa la distancia del robot desde una línea imaginaria a una distancia deseada d_{pared} , como se representa en la Fig. 1.3.

Fig. 1. 3: Referencia de las distancias para el control por seguimiento de pared



En la fig. 1.3 se representa la disposición de los sensores (S), cada uno separado por un ángulo de 45° (Carelli R. a., 2003) y Y_s como la distancia medida por cada sensor para la detección de obstáculos o para este caso específico la localización de una pared. En este caso, la variable \tilde{d} se puede calcular a través de la siguiente ecuación:

$$\tilde{d} = \pm \left\{ d_{pared} - \min \left(\frac{1}{2}(y_{s0} + y_{s7}) \right), \left(\frac{1}{2}(y_{s3} + y_{s4}) \right) \right\} \quad (1.10)$$

Donde el signo menos se considera si la pared que se sigue está en el lado derecho del robot.

En algunos casos, las mediciones entregadas por los sensores de ultrasonido pueden deteriorarse o ser imposibles de obtener en determinadas circunstancias, como por ejemplo cuando el robot se desplaza cerca de una puerta abierta en el pasillo o cuando el robot tiene un ángulo de desviación significativo con respecto al eje del pasillo. Esta última condición se origina en el hecho de que un sensor sonar recoge datos útiles sólo cuando su dirección ortogonal a la superficie reflectante se encuentra dentro de la anchura del haz del receptor, permitiendo de este modo la detección de pared sólo para una gama de rumbo restringida (Benporad, 1997). El rango para este ángulo es aproximadamente $\varphi = 15^\circ$ para los sensores usados en el robot. Por este motivo, es importante considerar también otras medidas, como los datos odométricos proporcionados por el robot. La fusión de estos datos utilizando filtros óptimos produce estimaciones óptimas de los estados del robot, minimizando así la incertidumbre en las mediciones del sensor, para la presente investigación, con el fin de minimizar los posibles errores mencionados anteriormente se hizo uso de un filtro de Kalman (Mutambara, 1998).

1.5 SIMULADOR SRM

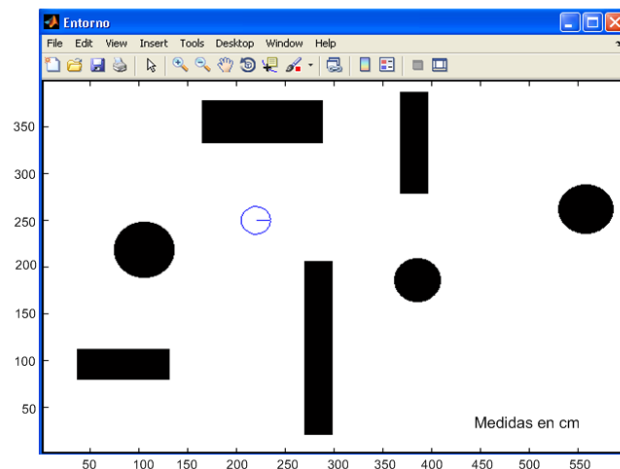
En general, los robots móviles son una tecnología costosa, así como su mantenimiento. El uso de entornos de simulación ofrece muchas ventajas para la investigación, el desarrollo y las pruebas. Por ejemplo, en robots reales cada vez que se realice un simple cambio en la plataforma o en el contexto, se necesita parar, reiniciar el robot y volver a instalar el nuevo programa. Con un simulador todas las pruebas se pueden hacer sin

dañar el robot real y es más fácil recuperar la información detallada acerca de la ejecución de la tarea asignada (Mejía, 2002).

El simulador es un Toolbox integrado a MATLAB, que permite al usuario simular el comportamiento de un robot móvil diferencial, como puede éste desplazarse a través de un entorno virtual. Donde el robot puede estar equipado con varios sensores virtuales, para nuestro caso son 8, que corresponden a los sensores ultrasónicos o infrarrojo además de poder navegar por su propio algoritmo de control.

El simulador incluye dos aplicaciones independientes. La primera de ellas es el editor, que permite al usuario crear y modificar la simulación. Esto significa la creación de un mapa de entorno virtual, la creación de robots, editar los algoritmos de control y la segunda sirve como un visor de simulación la cual permite ejecutar la simulación (el robot es manejado por el algoritmo de control de navegación).

Fig. 1. 4: Simulador SRM

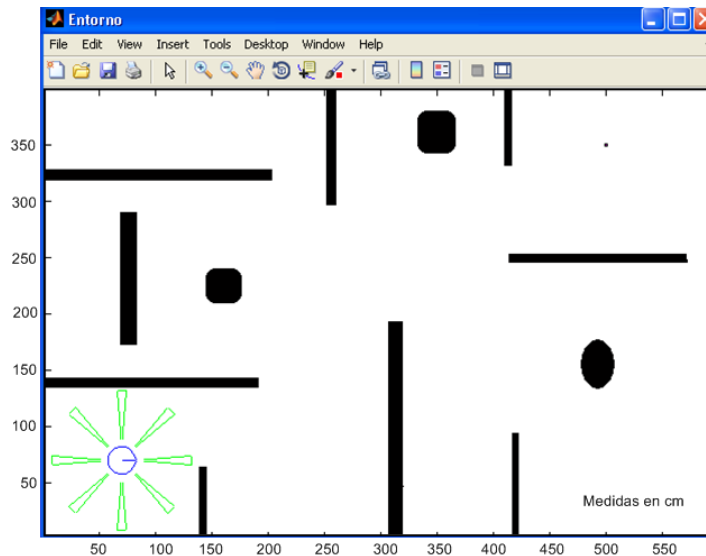


La simulación se ejecuta en pasos. En cada paso, los algoritmos de control de todos los robots se ejecutan y la posición de todos los robots se actualizan en consecuencia (la ventana de simulación se vuelve a dibujar). Los robots están representados visualmente por diversos tipos de polígonos, con la forma y color definido por el usuario.

El entorno de simulación virtual está representado por una matriz binaria (es decir de 1s y 0s). Para comodidad del usuario, la matriz puede ser creada a partir de un mapa de bits (* .bmp), y este puede ser editado fácilmente por cualquier editor de gráficos simples. En sí, el simulador tiene los siguientes componentes: el núcleo SRM, el escenario de simulación, la interfaz de usuario y el controlador (algoritmo de navegación).

La interfaz de usuario mostrada en la Fig. 1.5 proporciona la visualización y la interacción grafica que se está ejecutando al usuario de la simulación. Permite al usuario mover objetos alrededor de la escena para mostrar diferentes situaciones que pueden presentarse en un entorno real, da un control para la manipulación directa sobre los actuadores y tiene una visualización gráfica de la configuración y mediciones de los sensores.

Fig. 1. 5: Interfaz Usuario SRM

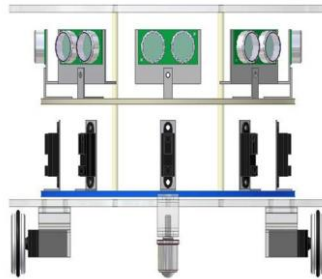


El controlador es el componente que implementa la acción pensar-actuar. En cada paso de la simulación, se leen los sensores, se procesa la información, se decide, es decir se planea la siguiente acción, y establece en los actuadores los estados a seguir. El controlador es proporcionado por el usuario y este contiene el algoritmo de control de navegación para los robots que por lo general es idéntico al algoritmo que se ejecuta en los robots reales.

1.6 ROBOT GIRAA 02

Es una plataforma robótica la cual ha sido creada a partir de varios proyectos de investigación, donde principalmente se han involucrado la Universidad de Antioquia y el Politécnico Jaime Isaza Cadavid, cuenta con 2 unidades de accionamiento y un pivote, configurado en modo diferencial, sistema que permiten movimientos en las direcciones: adelante, atrás y puede girar sobre un punto hasta alcanzar cualquier orientación que se necesite, siempre y cuando físicamente la estructura lo permita, tal como se observa en la Fig. 1.6.

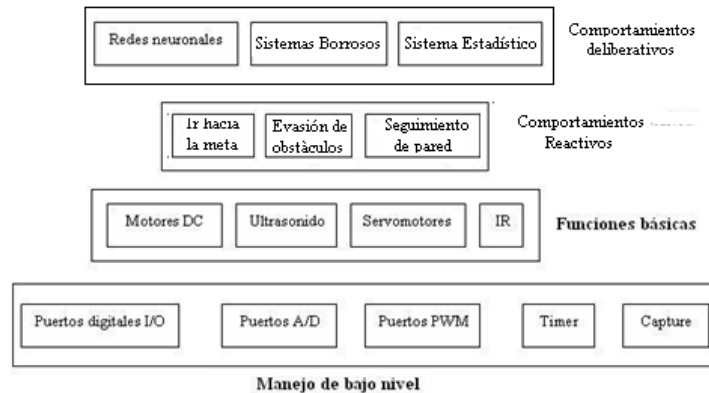
Fig. 1. 6: Robot GIRAA 02



Además el GIRA 02 está equipado con 8 sensores de ultrasonido e infrarrojo para mediciones de distancia para detectar posibles obstáculos, con un alcance de 26.5cm y un cono de detección de 25 grados, los cuales son usados para evitar colisiones, distribuidos equitativamente en la periferia del robot, y un sistema de odometría conformado por una IMU, brújula digital y un encoder óptico para detectar la velocidad real.

La estructura Hardware-Software con la que fue diseñada la plataforma es distribuida (Camargo, 2008) y escalable tal como se presenta en la Fig. 1.7, donde con las adecuadas funciones implementadas sobre los módulos de software, se pueden generar otras librerías derivadas y de un nivel superior de abstracción que son independientes del hardware, Por ejemplo las librerías de motores, esta librería llama funciones que están definidas en otros módulos de código (puertos, PWM, timers) y no maneja directamente el hardware de la plataforma

Fig. 1. 7: Estructura Hardware Software GIRAA 02



Es importante mencionar que se pueden desarrollar nuevas librerías a medida que se incorporen nuevos dispositivos hardware al robot. También está la posibilidad de optimizar las librerías actuales con la elaboración de algoritmos mejorados o comportamientos específicos como la búsqueda de objetos.

Los datos de a dónde debe llegar el Robot, obtenidos por la odometría propia del mismo, se le suministran a los algoritmos de control en coordenadas cartesianas (Vikas Kumar, 2004), pero todas las operaciones en la estructura de bajo nivel se realizan en coordenadas polares, todo ello basado en una referencia, si solo se trabaja con la brújula digital, se trabaja con respecto al norte magnético y con ángulos entre 0 y 360 grados, cuando se usa la IMU, se analizan las aceleraciones y las orientaciones, normalmente se trabaja solo con 2 de sus ejes, el X y el Y.

Los algoritmos en todas sus fases se desarrollaron para que gastaran la menor cantidad de recursos informáticos, tuvieran alto rendimiento y que permitiera manejar la mayor cantidad de hardware posible, esto con el fin de que cumplieren con las tareas que le fuesen asignadas, las cuales para este caso, están relacionadas con la navegación autónoma en interiores, teniendo caso especial con el manejo de pasillos y puertas.

1.7 MÉTRICAS O ÍNDICES DE DESEMPEÑO

Existen varias métricas que pueden ser empleadas para evaluar el desempeño de un sistema de navegación, no hay una sola que pueda indicar suficientemente la calidad del

sistema completo, por lo tanto, para una evaluación apropiada es importante combinar diferentes índices que cuantifiquen diferentes aspectos del sistema, entre estas métricas se pueden encontrar:

1.7.1 Métricas de Seguridad

Estas métricas expresan la relación de seguridad con la que transita un robot por una trayectoria determinada, teniendo en cuenta la distancia entre el vehículo (Álvarez, 1998) y los obstáculos que encuentra en el camino (Correa, 2013).

- **Medida de Seguridad-1 (SM1):** distancia media a los obstáculos durante toda la misión y para todos los sensores. Los valores máximos se obtienen cuando los entornos están libres de obstáculos. Si la desviación que se obtiene del índice de su valor máximo es baja, significa que la ruta elegida estaba libre de obstáculos.
- **Medida de Seguridad-2 (SM2):** distancia media mínima a los obstáculos. Para ello se toma el valor más bajo de los n sensores y luego se toma un promedio de dichas medidas. En un entorno libre de obstáculos $SM1 = SM2$.
- **Distancia Mínima (Min):** distancia mínima de cualquier sensor a cualquier obstáculo durante toda la trayectoria, mide el máximo riesgo que se corrió durante la misión.

1.7.2 Métricas Dimensionales

Se considera la trayectoria hacia la meta en sus dimensiones espaciales y temporales.

En general, se busca que la trayectoria óptima hacia el objetivo sea una línea con longitud mínima y curvatura cero entre el punto inicial (x_i, y_i) y el punto final (x_f, y_f) cubiertos en el tiempo mínimo.

- **Longitud de la trayectoria recorrida (PL):** es la distancia total recorrida por el vehículo desde el punto de inicio hasta la meta. Para una trayectoria en el plano x - y , compuesta de n puntos, y asumiendo el punto inicial como $(x_1, f(x_1))$ y la meta como $(x_n, f(x_n))$, PL se puede calcular como:

$$P_L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (f(x_{i+1}) - f(x_i))^2} \quad (1.11)$$

Donde $(x_i, f(x_i))$, $i = 1, 2, \dots, N$ son los n puntos de la trayectoria en coordenadas cartesianas.

- **Distancia media al objetivo (Mgd):** Esta métrica se puede aplicar en robots con capacidad de seguimiento de trayectorias de referencia. Un aspecto importante para determinar la calidad del sistema de navegación de un robot, es la habilidad de seguir una trayectoria que busca alcanzar una meta; entonces para evaluar la calidad en la ejecución de una trayectoria se analiza la distancia promedio entre el vehículo y la meta. (Rosenblatt, 1997) La diferencia es más significativa mientras más corta sea la distancia recorrida (Muñoz N. , 2014).
- **Periodos de Control (LeM):** cantidad de periodos de control. Esta medida es relativa al número de decisiones tomadas por el planificador para alcanzar el objetivo, si el robot se mueve a velocidad lineal (v) constante, da una idea del tiempo empleado en completar la misión (Álvarez, 1998).

1.7.3 Métricas de Suavidad

La suavidad de una trayectoria refleja la consistencia entre la relación decisión-acción tomada por el sistema de navegación (Rosenblatt, 1997), también, la habilidad de anticiparse y responder a eventos de un modo oportuno (Muñoz N. , 2014). La suavidad con que se genera una trayectoria, es una medida de los requerimientos de energía y tiempo para el movimiento, una trayectoria suave permite el ahorro de energía y tiempo; adicionalmente, es más conveniente para la estructura mecánica del vehículo.

- **Energía de Curvatura (Bending Energy BE):** es una función de la curvatura k , empleada para evaluar la suavidad del movimiento del robot. Para curvas en el plano x - y , la curvatura k en cualquier punto $(x_i, f(x_i))$ a lo largo de una trayectoria. Mientras la trayectoria sea más recta, tendrá menor BE y TBE, lo cual es deseable, la demanda de energía se incrementa de acuerdo al incremento de la curvatura en la trayectoria generada.

- **Suavidad de Curvatura (Smoo):** está definida por el cuadrado del cambio en la curvatura k de la trayectoria de un vehículo con respecto al tiempo, integrando a lo largo de la trayectoria y normalizada por el tiempo total t (Muñoz N. , 2014).

Suavidad de trayectoria (SmM_mod): es una medida del cambio de la dirección (w) dado en grados, por periodo de control. Esta medida está relacionada en proporción directa con el esfuerzo de control, cuanto menor sea, menor fue el esfuerzo de control (minimización de actuaciones) realizado en ese eje (Muñoz N. , 2014).

2. CAPITULO 2: ARQUITECTURA DEL SISTEMA NEURODIFUSO

En el presente capítulo se describe la estrategia del sistema desarrollado para la navegación de la plataforma robótica a través de entornos desconocidos, haciendo principal énfasis en el interior de edificaciones.

En este apartado en primera instancia se hablara sobre la arquitectura hibrida que se desarrolló para poder cumplir la tarea que se le asigna al robot, teniendo en cuenta que se trabajara sobre la velocidad angular, dejando la velocidad lineal de forma constante. Dentro de este primer bloque se explicaran los comportamientos reactivos usados, los cuales fueron Ir hacia la meta, búsqueda de espacios libres, repulsión de objetos y seguimiento de pared, luego el sistema deliberativo que se utilizó para controlar y ponderar la injerencia de cada uno de los comportamientos reactivos en la navegación. En el segundo bloque se le añadirá al robot la posibilidad de modificar su velocidad lineal resguardando la integridad propia de la plataforma.

2.1 INTRODUCCIÓN

Según Hebb (Hebb, 1958)], el comportamiento es la adaptación inicial al entorno considerando la percepción sensorial. Esto permite que el organismo se aleje de los eventos perjudiciales y se acerque hacia los favorables, o que realice cambios en su entorno local para su beneficio en cuanto a su supervivencia. En este contexto se indica que la manera más apropiada de obtener un comportamiento adaptable en la robótica surge de la interacción directa con el entorno, limitando la intervención del hombre a la definición de los objetivos a cumplir por el comportamiento requerido. De este modo, se hace necesaria una regla que determine cuantitativamente el grado que una determinada conducta se aproxima al comportamiento deseado (Fernández León, 2005).

También se puede expresar que un comportamiento es un conjunto de trayectorias compuestas, cuyo fin es el de expresar el control del robot. Una vez que se especifica, se puede ejecutar tantas veces como sea necesario. Los comportamientos pueden ser expresados a nivel de motor, habilidad o tarea. A nivel de motor, los comportamientos se

expresan mediante la prescripción de comandos directamente a los actuadores del sistema. A nivel de habilidad, los comportamientos expresan las capacidades del robot, por ejemplo, girar, parar, y así sucesivamente. Las habilidades son expresadas como modelos parametrizados, programas para controladores de nivel de motor, sin la capacidad de estrategias acerca de los objetivos o la semántica codificada sobre el mundo real. Una habilidad se puede definir con la trayectoria de cada articulación afectada. Por último, los comportamientos a nivel de tareas son realizadas por habilidades con el fin de lograr los objetivos especificados (Miranda Cruz, 2011).

En sí, cuando se entran a estudiar los comportamientos en un robot móvil se hace necesario implementar estrategias o modelos matemáticos que permitan precisar, inicialmente, el comportamiento teórico del móvil a partir de una relación de sus velocidades y posiciones, que implemente algoritmos de posicionamiento en 2D, planificación de rutas, planificación de tareas o simplemente desarrollar un sistema de localización que son necesarios para la implementación de un sistema más robusto que cumpla con los objetivos especificados por el equipo de investigación (Bermúdez, 2012).

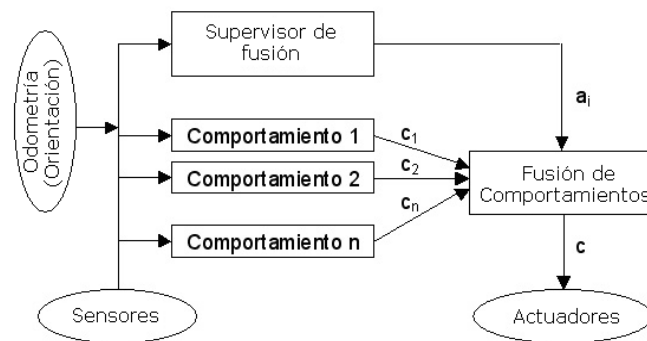
Es importante entender la cuestión de cómo emerge un comportamiento autónomo en el robot a partir de esta serie de condicionamientos, sin que exista conflictos e interferencias entre las diferentes conductas, y se produzca un comportamiento coordinado y correcto; por ejemplo, en los animales, cuando se produce una necesidad asociada con un motivo primario, toda la motivación del sujeto se dirige a realizar de forma inmediata la conducta que satisfaga esa necesidad. Para que el robot tenga un comportamiento adecuado a sus motivaciones y no existan conflictos entre unas y otras, se propone una estructura jerárquica, que maneje unos pesos estimados, para cada uno de los comportamientos reactivos, los cuales son generados por el bloque de Supervisión, (Fig. 2.1).

Para poder enfrentar el problema de la navegación autónoma a través de esta estrategia, se dividió el problema en 2 partes. En primera instancia se trabajó el control sobre la orientación del robot y la velocidad es constante, con eso es suficiente para que el robot llegue a la meta y se simplifica el problema. Para la segunda parte de este método, se realizó un trabajo sobre la velocidad lineal, la cual ya se puede variar siempre y cuando la seguridad de la plataforma no se vea afectada.

2.3 CONTROL DE ORIENTACIÓN

Se empezara este apartado apoyando una idea sobre la necesidad de las arquitecturas de control, para posteriormente analizar las soluciones que fueron adoptadas en su diseño, interesándonos especialmente en las arquitecturas que involucran en alguna medida el uso de la naturaleza reactiva, ya que se quiere adaptar en la medida de lo posible a la arquitectura de orden jerárquico; donde se entiende una arquitectura de control de robots móviles como un sistema software que establece las acciones o movimientos que debe realizar el robot a partir de la adquisición y tratamiento de la información sensorial y del objetivo u objetivos que le hayan sido indicados (De Lope Asiaín, 2001).

Fig. 2. 1: Arquitectura de control



El software de control puede ser relativamente simple como se puede encontrar la funcionalidad de un robot rastreador, pero puede también llegar a alcanzar una cota de complejidad muy alta como en los robots humanoides de última generación, como el Asimo de Honda o el QRIO de Sony (Garre del Olmo, 2006). Por ello cuando se llega a cierto nivel de complejidad en el diseño, se hace casi imprescindible dividir el problema en una serie de niveles de abstracción para que hagan el problema más manejable. Es aquí donde se ve la necesidad de definir una arquitectura de control para el robot, entendida normalmente como un sistema jerárquico de niveles de abstracción sobre los que se implementará el software de control del robot.

Para este ítem se analizaron algunos elementos en el área de robótica, entre ellos uno relevante como lo son las arquitecturas de control, las cuales se pueden definir como el

grupo de elementos Hardware-software involucrados en el control de un robot (Londoño N. y., 2003), donde se deben satisfacer elementos tales como la detección de un evento en el momento en que se produzca y poder reaccionar satisfactoriamente a él, capacidad de aprendizaje y de decisión, simplicidad de manejo, entre otras (Simo Ten, 1997). Estas arquitecturas de control pueden clasificarse en:

- **Arquitecturas Deliberativas o Jerárquicas:** En esta se presentan diversas escalas (Noreils, 1989), en el cual cada nivel trae incorporada una tarea (Ridao P. B., 1999).
- **Arquitecturas Reactivas:** Presenta una distribución horizontal, en donde cada una de las partes personifica un comportamiento simple (Brooks, Rodney, 1997), en donde cada uno de ellos recibe información de los sensores directamente e inmediatamente se genera una respuesta (Brooks, Rodney, 1987).
- **Arquitectura Híbrida:** Posee las ventajas de las 2 arquitecturas anteriores (Brooks, Rodney, 1987), la regularidad de las arquitecturas deliberativas y la capacidad de reacción de las arquitecturas reactivas (Ridao P. C., 2001).

El método que se propone para dar solución a esta parte del trabajo planteado es el presentado en la Fig.2.1 la cual estará compuesta por una serie de conductas que son tomadas de comportamientos existentes en la vida de los animales y un sistema denominado supervisor de fusión, el cual será el encargado de darle un peso a cada uno de los comportamientos, de acuerdo a la situación en que se encuentre el robot en cada periodo de muestreo, durante la misión asignada.

Dentro del trabajo se utilizaron 5 comportamientos reactivos, dentro de ellos 2 de gran importancia como seguimiento de pared derecha e izquierda, dado que es una de las herramientas usadas y con mejores resultados a la hora de realizar navegación autónoma a través de pasillos, el cual es uno de los objetivos relevantes que se propuso elaborar en este trabajo de grado.

La comunicación de información es unidireccional, los comportamientos pueden suministrar información a los controladores o a los actuadores, pero no en sentido contrario.

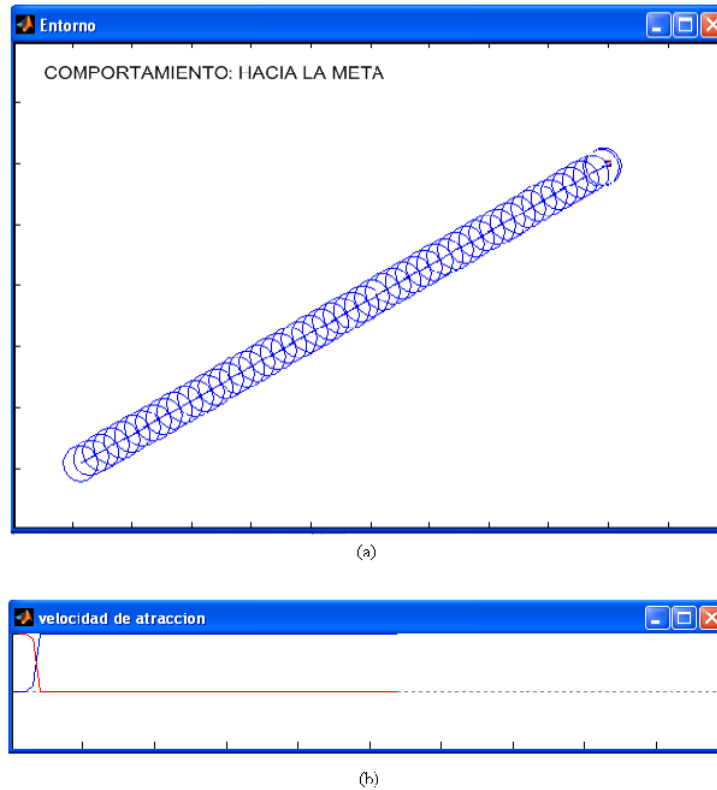
2.4 COMPORTAMIENTOS REACTIVOS

Dado que en un principio, estos comportamientos reactivos vienen dados por las conductas y/o experiencias propias de los animales, se ha de tener en cuenta que en un contexto real, los animales pueden adaptarse a los cambios ambientales y además de ello también pueden acumular adaptaciones. A partir de ellos se puede decir que los animales son capaces de acumular “conocimiento” sobre su ambiente y a partir de ello usarlo cuando se encuentren nuevamente en dicha situación. Este proceso puede ser denominado como aprendizaje cuando sucede en un tiempo corto y evolución cuando ocurre en una escala generacional (Fernandez, 2002). En otras palabras, un agente (animal o robot) se adapta mediante la evolución a escala generacional (etapa de desarrollo del controlador) de diferentes comportamientos útiles para distintos entornos, y mediante aprendizaje (etapa de evaluación del controlador).

Esta técnica permite que sea asequible añadir, sustituir unos comportamientos por otros, respondiendo de esta forma a posibles cambios en la dotación sensorial del robot o a la incorporación de otras estrategias de percepción (Cerrada C. C., 2013). Dentro de los comportamientos que se trabajan inicialmente para el cumplimiento de las tareas propuestas están:

- a. **Ir hacia la Meta:** Es una conducta plenamente deliberativa, es decir que no tienen en cuenta la percepción (Cerrada C. C., 2013), es decir que los sensores para detectar obstáculos no son tenidos en cuenta a la hora de realizar cualquier tarea.

Fig. 2. 2: Hacia la Meta



En la fig.2.2 (b) se pueden identificar 2 líneas, una de color azul y la otra de color roja, la primera indica la velocidad lineal que conlleva la plataforma, allí se puede observar que esta es constante, mientras que la segunda línea indica la velocidad angular de la misma y para este caso también indica el ángulo que lleva hacia la meta, como no hay presencia alguna de obstáculos en la trayectoria.

Las ecuaciones para determinar la velocidad lineal v_{atr} y la velocidad angular w_{atr} de la plataforma robótica, están dadas por:

$$v_{atr} = v_{max} \left(1 - \frac{|w_{atr}|}{w_{max}} \right) \quad (2.1)$$

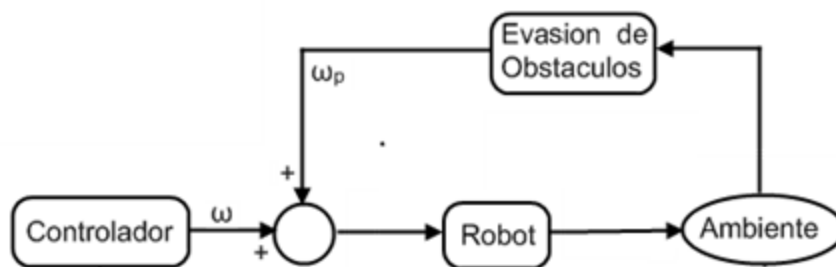
$$w_{atr} = k\theta_{meta} \quad (2.2)$$

Donde v_{max} y w_{max} son las máximas velocidades lineal y angular que puede alcanzar el robot sin arriesgar su seguridad, ni sufrir percance mecánico alguno, θ_{meta} es el ángulo

formado entre la plataforma robótica y la meta; es de anotar que la velocidad lineal depende de la angular para evitar que el robot no se desplace cuando éste no este guiado hacia la meta.

- b. **Repulsión de objetos:** La idea principal de este comportamiento es evitar los obstáculos que puedan aparecer eventualmente en el camino del robot a lo largo de su trayectoria, cuando este está cumpliendo una tarea asignada. Con este propósito se genera una señal de perturbación ω_p que se le suma a la orden proveniente del controlador, tal como se ve en la Fig. 2.3, para este caso la señal ω_p servirá para orientar el robot para que pueda evitar el obstáculo.

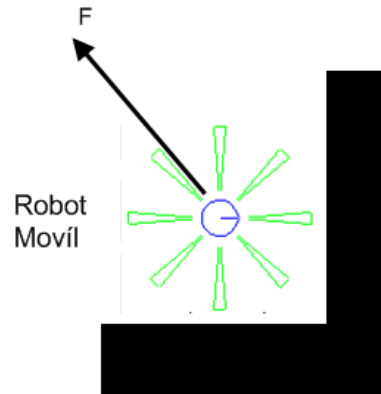
Fig. 2. 3: Diagrama de Bloques evasión de Obstáculos



$$\omega_p = k_\psi \psi + Y \sin \psi \cos \psi \quad (2.3)$$

Para el caso en particular, cuando la misión que se le asigna al robot es ir desde un punto inicial hasta un punto de destino o meta como, es el caso en el cual se realizara en este trabajo, se genera una fuerza de atracción virtual (f_a), que hará que la plataforma robótica trate de llevar siempre dicha dirección. Si se detecta un objeto en el entorno de trabajo del robot, aparece una fuerza (f) contraria, originada por dicho obstáculo, en ese instante el sistema cambia momentáneamente la posición de destino a un nuevo punto, esto producido por la suma vectorial producida por f y f_a , como resultado se produce una fuerza resultante f_r , la cual genera que el robot gire un ángulo ψ , la cual permitirá que el robot esquive el objeto y que en el próximo ciclo de control se analice la situación en que se encuentra y si éste ya no encuentra ningún riesgo, pueda retomar la tarea asignada.

Fig. 2. 4: Fuerzas Repulsión de objetos



En la ecuación 2.3 se toma de (Carelli R. y., 1999), donde k_ψ y Y son las constantes de diseño positivo.

Se puede lograr un comportamiento prudente y suave para el robot en función de los errores de control (ver Fig. 2.3), considerando la función:

$$u(\varphi, \tilde{x}) = (u_d - k|\tilde{x}|) \cos \varphi \quad (2.4)$$

El comando de velocidad lineal $u(\varphi, \tilde{x})$ de la ec.2.4 se vuelve a definir para generar un movimiento cauteloso cuando el robot se dirige hacia la meta para evitar un obstáculo, siendo este:

$$u_{new} = [u(\varphi, \tilde{x})] \cos \psi \quad (2.5)$$

Para calcular el ángulo ψ , se genera una fuerza ficticia de interacción f en función de la distancia d del robot-obstáculo, como se muestra en la Fig. 2.4. La fuerza f representa una interacción virtual entre el robot y el obstáculo.

$$f = a - b(d - d_{min})^2 \quad (2.6)$$

Donde a y b son las constantes positivas, tales que $f = a - b(d - d_{min})^2 = 0$ El valor d_{max} es la distancia máxima entre el robot y el obstáculo detectado que debe producir

una fuerza de repulsión f , d_{min} la distancia mínima que el sistema sensorial puede medir, y d la distancia medida entre el robot y el obstáculo. Para ver una ampliación del análisis matemático se pueden remitir a (Carelli R. a., 2003).

En resumen, a causa de la fuerza ficticia resultante, se desplaza el punto de destino del robot y, a continuación, se genera un comando de desviación conveniente ω_p , que permite al robot guiar y evitar el obstáculo detectado. La capacidad de reacción del robot para evitar el obstáculo se regula mediante una selección adecuada de constantes en las ecuaciones 2.3 y 2.4. De esta manera, el algoritmo puede aplicarse a la navegación normal o cuando incluye la navegación de pasillo o de pared con capacidad de evitar obstáculos. Debe tenerse en cuenta que cuando ningún obstáculo está presente dentro de una distancia máxima d_{max} del robot, ψ es cero y no se añaden señales de prevención de obstáculos a las órdenes de control. Por lo tanto, el robot evoluciona de acuerdo con el objetivo de control asignado.

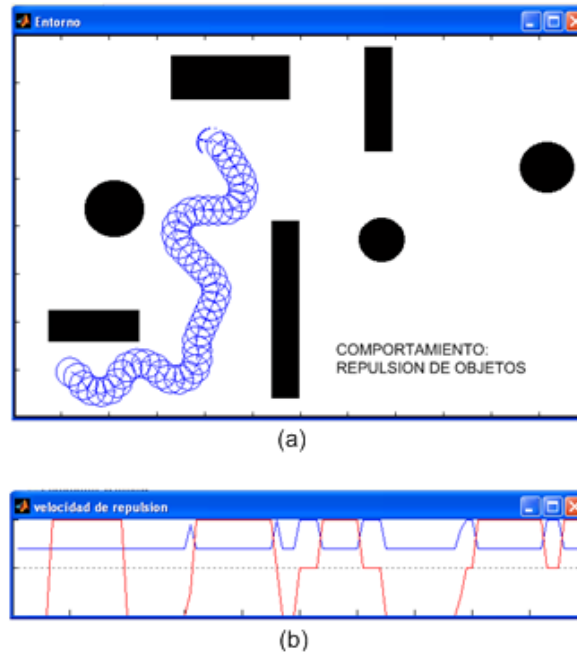
Para muchos casos las velocidades lineal y angular de repulsión se calculan así:

$$v_{rep} = v_{max} \left(1 - \frac{|w_{rep}|}{w_{max}} \right) + v_{min} \quad (2.7)$$

$$w_{rep} = k\theta_{rep} \quad (2.8)$$

Donde se incluye el parámetro de velocidad mínima v_{min} para evitar que el robot se detenga mientras se encuentra girando para evadir un obstáculo (Fernandez, 2002). Una velocidad lineal de cero en este comportamiento hace que en algunos casos el robot se quede atrapado en una zona.

Fig. 2. 5: Repulsión de objetos

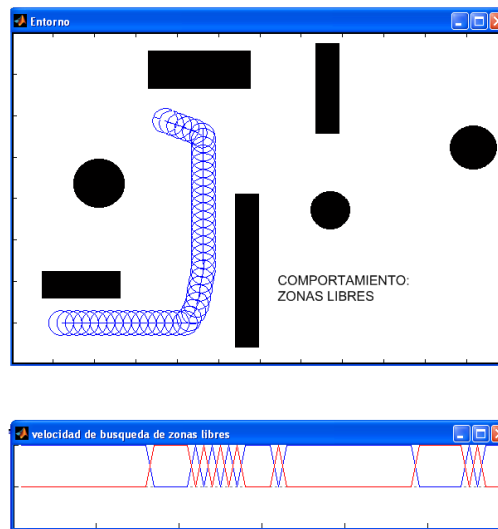


De la Fig. 2.5, en la parte a, se muestra el trayecto que sigue el robot de acuerdo a la tarea asignada, mientras que en la parte inferior hay 2 líneas, la primera indica la velocidad lineal que presentaría la plataforma, en ella se puede observar que la velocidad trata de ser la misma en casi todo el trayecto, solo presenta algunas alteraciones cuando presenta grandes variaciones en la velocidad angular, esta corresponde a la línea rojiza, allí también se puede ver, que el ángulo que debe tomar la plataforma en cada ciclo varia, elemento que debe saberse manejar adecuadamente para no afectar la estructura mecánica de la plataforma y la de los motores.

c. **Búsqueda de espacios libres:** Una zona libre es considerada como una dirección en la que tres sensores consecutivos no detectan ningún obstáculo. En caso de presentarse esta situación en varias direcciones, se escoge la que esté más cerca del frente del robot (Moreo, D., & J., 1996).

Para nuestro proyecto solo se tienen en cuenta para este comportamiento la medida captada por los sensores S0, S1, S2 y S3 los cuales son los sensores frontales del robot, tal cual se visualiza en la Fig. 2.6. Una vez se tiene esto, se aplica el método de la convolucion (Barragá Piña, 2009) para obtener la dirección hacia donde debería guiarse el robot de acuerdo a este proceso.

Fig. 2. 6: Búsqueda de espacio Libre



Lo primero que se tiene en cuenta son los ángulos que tienen los sensores con respecto a la frontal, los cuales siempre son iguales, estos ángulos son:

- $\text{ang_S0}=25^\circ$;
- $\text{ang_S1}=70^\circ$;
- $\text{ang_S6}=295^\circ$;
- $\text{ang_S7}=340^\circ$;

A continuación se muestra como queda el algoritmo usado en Matlab

Fig. 2. 7: Convolucion Usada para calcular comportamiento Zonas Libres

```

function [v,w] = zonaslibres(si)

masc = ones(1,3);
sith = si > 49;
sith = [sith(length(si)) sith sith(1)];
z1 = conv(sith,masc);
z1 = z1 > 2;
izl = find(z1);
zlizq = min(izl) - 2;      % Zona libre a la derecha
zlder = max(izl) - 2;     % Zona libre a la izquierda

v = 0;
w = 0;

if zlizq == 1
    v = 1;
    w = 0;
    return
end;

if zlizq <= (18 - zlder)
    v = 0;
    w = 1;
else
    v = 0;
    w = -1;
end

```

La variable *si* entrega la medida de los sensores en cada ciclo

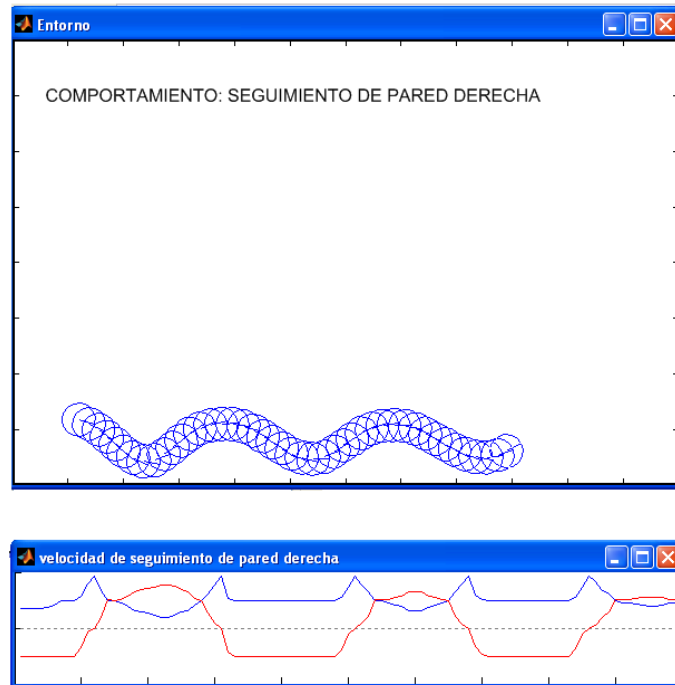
(a) Algoritmo simulador SRM. Matlab

d. Seguimiento de pared:

Normalmente se diferencian 2, siendo ellos seguimiento de pared Derecha y seguimiento de pared Izquierda, el funcionamiento básico de este comportamiento se explicó en el literal 1.5.

Para el caso que se evidencia en la Fig.2.8., se muestra como la plataforma robótica trata de seguir a una distancia segura la pared derecha de un entorno *x*, elemento de gran relevancia cuando se trata de seguir un corredor.

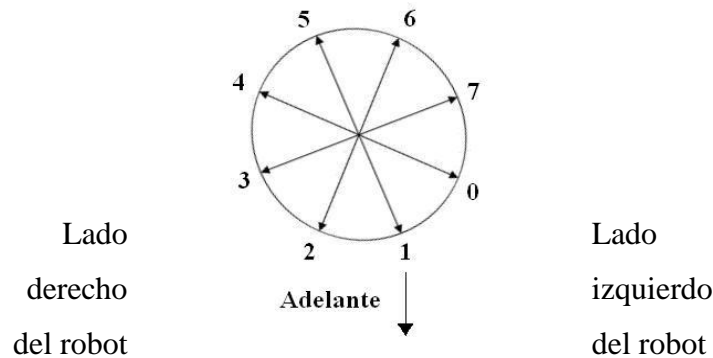
Fig. 2. 8: Seguimiento de Pared



Es de anotar que la diferencia entre ambos seguimientos de pared, radica esencialmente en 2 aspectos, uno puede ser en el signo, donde el signo menos se considera si la pared que se sigue está en el lado derecho del robot, o en el caso contrario en el peso que se le asigna a cada uno de ellos, donde mientras uno de ellos presenta un valor cercano al 1 el otro lo tendrá cercano al cero, esto es de aclarar que exista la presencia de un muro.

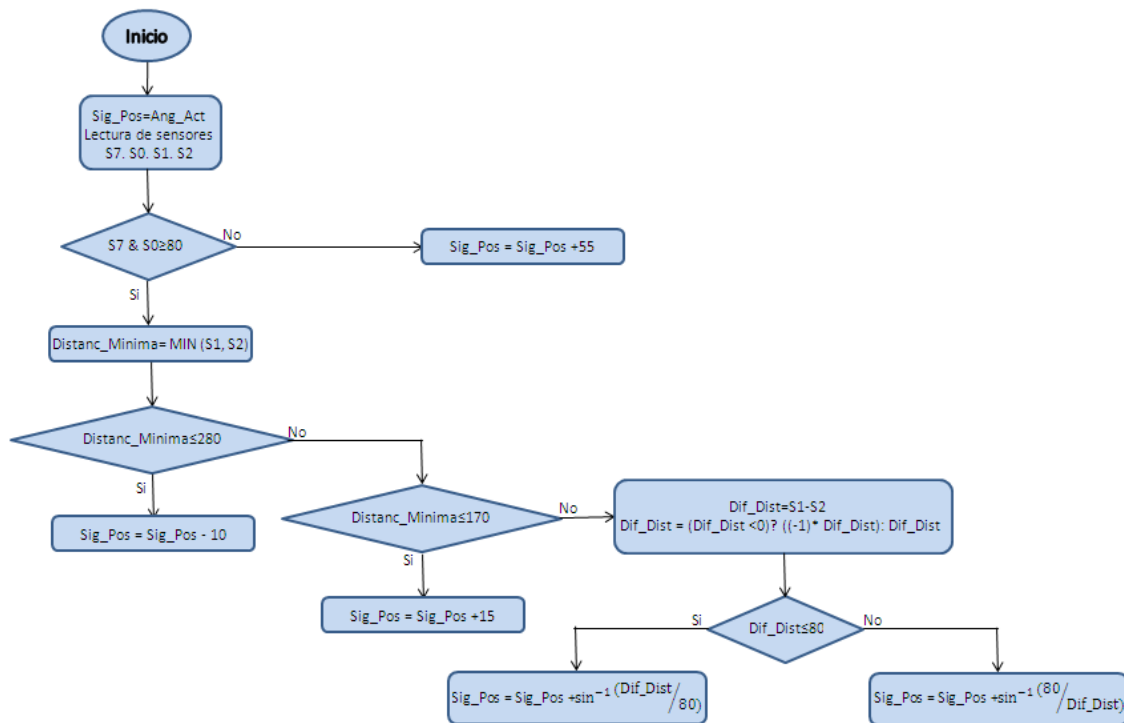
Para lograr realizar adecuadamente este comportamiento, lo primero que se debe tener en cuenta es la distribución de los sensores en la plataforma la cual se puede ver en la Fig. 2.9, donde se usan los sensores S1, S2, S3 y S4 para realizar el seguimiento de pared derecha, mientras que los sensores S0, S1, S2 y S7 para el seguimiento de pared izquierda.

Fig. 2. 9: Distribución de los sensores en la plataforma



Una vez ya se tuvieron en cuenta la distribución y la ubicación de los sensores, para el análisis del seguimiento de pared, se tuvo en cuenta el análisis mostrado en la Fig. 2.10.

Fig. 2. 10: Diagrama flujo algoritmo seguimiento de pared izquierda



Es de anotar que las constantes que se trabajan con la variable Distanc_Minima son en milímetros y que las constantes que se manipulan con la variable Sig_Pos son en grados. La variable Ang_Act entrega el ángulo con el cual se encuentra la plataforma robótica

con respecto a la meta y la variable Sig_Pos entrega la velocidad angular que entrega el comportamiento de seguimiento de pared derecho para el próximo ciclo de reloj.

Para este tipo de conducta es importante rescatar que los sensores utilizados por la plataforma robótica aprovechan al máximo el medio ambiente semiestructurado en interiores, esto significa que en el entorno pueden existir muchas paredes libres (por ejemplo, corredores) (Brooks, Rodney, 1987). Sin embargo, hay casos en los que esto no es cierto, ya sea porque hay muchos objetos en el medio donde se realiza la tarea asignada (sillas, mesas, personas) que ocultan la pared o porque las paredes están más allá de la escala de los sensores.

Dado todo lo anteriormente mencionado, para algunas plataformas robóticas, estas se podrían clasificar en:

- De exploración: robots que transitan sin rumbo fijo.
- Que preserve la dirección
- Ir a un objeto: ir a un objetivo puntual, sector o área.
- Seguir caminos, seguir una ruta o seguir una línea.
- Evitar un área: evitar o esquivar un obstáculo fijo o móvil.
- Caminar: andar, doblar o correr.

En una ejecución normal de la red existirán varios comportamientos activos ejecutándose en paralelo. Éstos pueden formar parte de la misma conducta, por ejemplo, elementos perceptivos que están captando información a la vez que se está deliberando sobre la información percibida anterior y al mismo tiempo que los actuadores estén actuando sobre el entorno (Cerrada C. C., 2013).

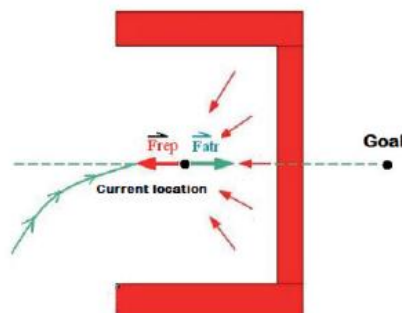
Una de las técnicas más representativas donde se mezclan los 2 primeros comportamientos mencionados anteriormente se denomina Campos Potenciales, la cual está enfocada en la planificación local en entornos desconocidos (Yandún, 2012), donde la plataforma es considerada como una partícula que puede verse influenciada por un campo potencial artificial, donde este se halla compuesto de un potencial que lo atrae hacia la meta (f_a) y un potencial que hace que el robot se aleje de los obstáculos (f), donde el potencial resultante (f_r) se obtiene de la suma vectorial de ambos potenciales. (Ollero, 1995), como se ve en la fig. 2.4.

Donde la navegación basada en campos potenciales lleva a cabo las siguientes acciones en el robot.

- Calcula el ángulo de orientación que debe tener la plataforma para ubicarse hacia su destino final, esto basado en la odometría.
- Obtener la medición de los sensores para detección de obstáculos, para poder saber cómo está su entorno, con respecto a su posición actual, para así calcular una orientación contraria con respecto a los obstáculos más cercanos.
- Cuando se tengan ambas orientaciones, cada una otorga un ángulo hacia el cual debería ir el robot, cada una de ellas es multiplicada por una constante, para las apreciaciones que se obtuvieron en el robot real, la constante del potencial de atracción es 3 veces mayor que el potencial de repulsión.
- Una vez se elabora la multiplicación de los pesos con la orientación obtenida, se elabora la suma vectorial de ambos valores, así se obtiene el ángulo o vector calculado, para construir las órdenes adecuadas para que los actuadores de la plataforma hagan que éste se mueva según el sentido y dirección obtenidas

Cabe resaltar que el potencial de atracción hacia la meta debe disminuir a medida que se acerque al destino final, para evitar que la convergencia sea lenta y el potencial de repulsión solo debe tenerse en cuenta cuando los obstáculos se encuentren demasiado cerca de la plataforma y pueda comprometer su seguridad. En el sitio de llegada, es preciso que la suma de ambos potenciales sea nula (Yandún, 2012).

Fig. 2. 11: Ejemplo de un mínimo local (OLLERO, 1995)



Otro de los problemas que se tienen con esta técnica, son los mínimos locales tal como se ve en la Fig. 2.11, son espacios donde el potencial final, resulta nulo pero no se trata de la posición final.

2.5 SISTEMA DELIBERATIVO – SUPERVISOR DE FUSIÓN: RED NEURONAL (RNA)

En este tipo de análisis es muy común el uso del término agente, el cual según S Russell y Norvig (Russell, 1995), “un agente es algo que puede percibir su entorno por medio de sensores y desenvolverse en él a través de actuadores”. Y que “un agente ideal” es “uno que siempre toma la acción que se espera para maximizar su desempeño o performance, dada una determinada secuencia de percepciones realizadas”; en muchos casos puede ser considerado que un agente haya sido capaz de acumular adaptaciones con pequeños cambios en la experiencia obtenida en una generación, es acá donde la experiencia obtenida por una aplicación de redes neuronales, para este caso un neuro-controlador se refiere a las características que pueden ser desarrolladas en una generación (por ej., los pesos sinápticos establecidos luego de un proceso evolutivo) (Fernández León, 2005).

Así que uno de los desafíos principales en RN es descubrir y modelar distintos mecanismos de adaptación, de acuerdo a lo cambiante que puede ser la navegación de los robots (que puede ser entendido como un agente) en entornos interiores y más aún en aspectos tan sensibles como pasillos y puertas. Por ello es que se propone que la RN es una metodología viable para desarrollar agentes autónomos que puedan presentar habilidades con algún grado de conciencia necesaria en la metodología acá desarrollada.

Un aspecto que también es importante entender es la multiplicidad de arquitecturas simples de RN, las cuales son capaces de realizar procesamiento temporal, dentro de estas se pueden encontrar diversos ejemplos (Floreano, 2001), tales como las Redes Neuronales Recurrentes de Tiempo Discreto (TRNN- Discrete Time Recurrent Neural Networks), con dos variantes se pueden encontrar las Redes Neuronales Plásticas (PNN - Plastic Neural Networks) utilizadas en (Nolfi, 1999), y las RN de una variante denominadas (Floreano, D. y Urzelai, J., 1999) como las redes tipo Feed-Forward (FFNN) descritas en (Togelius, 2003). Según Fernández (Fernández León, 2005) “Este

tipo de controladores son capaces de presentar un comportamiento apropiado, recordando las habilidades adquiridas y pasándolas a las generaciones venideras. Las mismas, permiten almacenar experiencia previa y utilizarla para modificar su comportamiento actual, así como el comportamiento de sus descendientes cuando se presente un entorno determinado (o situación). Esta adaptación es más rápida a medida que se incrementa el valor de la función de fitness”, donde el fitness de un controlador representa la cantidad de movimientos exitosos respecto del total de movimientos que son posibles de realizar.

Los comportamientos mencionados anteriormente no pueden resolver por sí solos un problema de navegación complejo. Por ejemplo, en el caso del comportamiento de atracción hacia la meta orienta al robot hacia ella, estrellándose con todos los obstáculos que hallan en su camino, caso contrario sucedería con el de repulsión de objetos, el cual mantendría la plataforma alejada de cualquier obstáculo, pero el robot no cumpliría con la tarea asignada.

La idea que se pretende es que el robot adquiriera una conducta más compleja a partir de una secuencia de conductas elementales, donde se hace necesario encontrar la forma de como fusionar los comportamientos, para ello es importante que todos ellos intervengan en el control final del robot, coordinados por el supervisor de fusión, el cual indicara el grado de participación de cada uno de ellos.

En el trabajo que se presenta se optó por utilizar una red del tipo Feed-Forward (FFNN), entrenada con la regla de aprendizaje backpropagation, esta red que se diseñó es no recurrente en las capas ocultas, ósea que no tiene pesos sobre sí mismo; También se permite la inicialización aleatoria de los pesos sinápticos con la función *InitializeRandoms()*, ya sea cuando reinicia la red o cuando se presenta un reset inesperado en la operación.

La RNA que se implementó presenta 4 capas (ver Fig. 2. 12), contando con la entrada de sensores y odometría del sistema como una capa.

Fig. 2. 12: Configuración de las capas en el Robot

```
#define NUM_LAYERS 4 //Número de capas
#define N 9 //Capa entrada
#define Capa1 20 //Capa oculta 1
#define Capa2 20 //Capa oculta 2
#define Capa3 5 //Capa de salida
```

1. Capa de Entrada: El número de neuronas de esta capa son 9, las primeras 8 neuronas de entrada corresponden a la cantidad de señales de sensores que posee el robot en toda su periferia, ya sea infrarrojos o ultrasonidos o la mezcla de ambos a través de un sistema de combinación como el filtro de Kalman, y la última señal corresponde al ángulo de orientación que tiene la plataforma en el momento en que se captura los datos, en cada periodo de control.

2. Dentro de la RNA diseñada, se encuentran 2 capas ocultas, siendo estas:

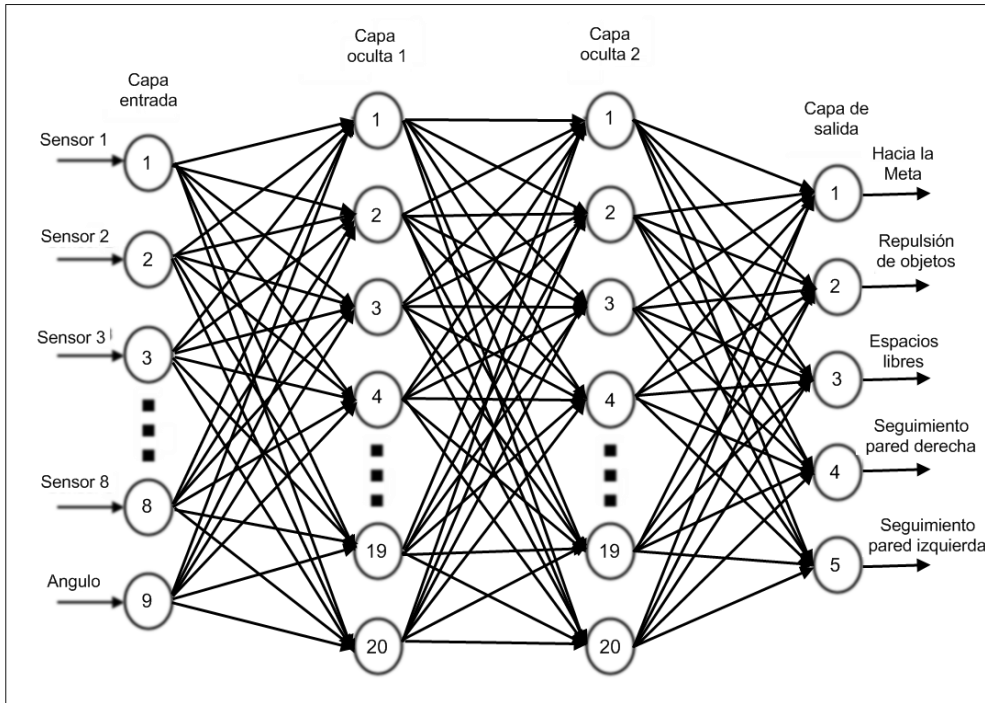
a. Capa Oculta 1: El número de neuronas de esta capa es de 20, se debe tener en cuenta ya hay ciertas recomendaciones que dicen que no se debe superar el doble de las neuronas de entrada (Flórez, 2008), pero se tuvo que hacer uso de un número mayor, dado que la red presentaba algunos problemas cuando se pretendía abordar algunos de los hitos que se están trabajando.

b. Capa Oculta 2: Para esta capa se trabajó bajo los mismos principios de la capa oculta 1.

3. Capa de Salida: Para esta capa, el número de salidas corresponden a la cifra de comportamientos que se tienen, que para este caso es de 5, este entregara un peso de cuánto debe aportar cada comportamiento, con respecto a la orientación del robot para el próximo movimiento.

La topología de la RN que se utilizó para desarrollar el elemento deliberativo puede observarse en la fig. 2.13. En ella se ve que neuronas con funciones de activación del tipo tansigmoial (llamadas tansig) fueron utilizadas en todas las capas en que fuese necesario, para este caso, en las 2 capas ocultas y la capa de salida. Esta función de activación normalmente viene para trabajar bajo normalización entre -1 y 1, pero dado la circunstancia a la cual se está aplicando, donde no se tienen velocidades negativas, el rango de trabajo que se aplicó esta entre 0 y 1.

Fig. 2. 13: Topología de la red Neuronal desarrollada



Una vez la red fue entrenada, se valoraron sus resultados en 1 entorno simulado el cual está basado en el SRM que se explicó en el capítulo anterior, el cual usa como software base el Matlab, ya que contiene un Toolbox de Redes Neuronales que facilitan el diseño, análisis, entrenamiento y propagación, acá la función principal de la FFNN quedo como se muestra en la fig. 2.14.

Fig. 2. 14: Función principal de la Red Neuronal en el Robot

```
void simulateNetAlt(NET* Net, REAL* Input, REAL* Output)
{
  setInput(Net, Input);          //Captura los elementos a evaluar en la red
  PropagateNet(Net);            //Propaga y ejecuta la red
  GetOutput(Net, output);       //se obtienen las salidas finales de la red
}
```

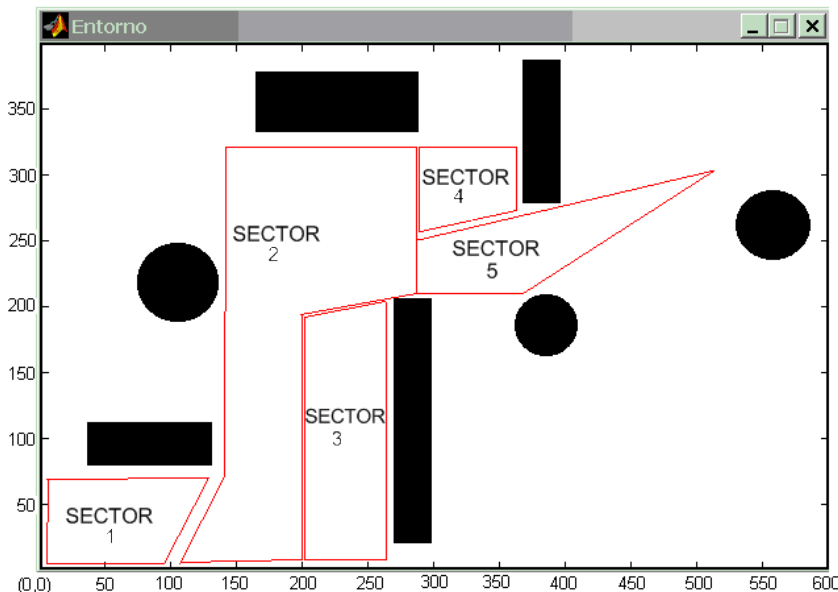
Para la salida del sistema el orden de las salidas para los comportamientos se organizó de la siguiente forma:

Hacia la meta (a1) -Repulsión de objetos (a2) -Búsqueda de espacios libres (a3)- Seguimiento de pared derecha (a4)- Seguimiento de pared Izquierda (a5).

El número de elementos y su forma en cada entorno era diferente, la posición dentro de cualquiera de los entornos para iniciar la meta fue aleatoria y en cada generación, los miembros de la población (diferentes instancias de neuro-controladores para el robot) también fueron inicializados en posiciones aleatorias dentro del ambiente, los ejemplos de algunos de estas simulaciones y/o pruebas se podrán ver en el capítulo de resultados. Como salvedad en este ambiente de simulación no se utiliza un modelo del robot que tenga en cuenta dinámicas ni no linealidades debido a los actuadores motores y llantas.

Para cada uno de los entornos en donde se realizó el aprendizaje y el entrenamiento se dividió por sectores tal cual se ve en la fig. 2.16, en cada uno de ellos inicialmente se analizó cada comportamiento individual, para observar cuales eran los ángulos que estos suministraban, una vez se tenían estos valores se asumían unos pesos que fuesen acordes al resultado que se esperaba, esto para obtener los 300 puntos que se mencionaron anteriormente. Es de anotar que para el caso que se está mostrando en la fig. 2.16 se tenía como misión ir desde un punto inicial, la cual era la posición 50, 50 en coordenadas cartesianas y como punto de llegada el punto 500, 300. Luego se elaboró un programa en Matlab que ayudo a determinar cuáles eran los pesos que se debían tener ante ciertas situaciones, esto ayudo a mejorar en gran medida la caracterización de las muestras que se necesitaban.

Fig. 2. 16 Sectorización del Entorno



Sector 1 se analizó el caso de pasillos el cual es uno de los Hitos que se analiza en la presente tesis, para ello se trabajó con seguimiento de pared a ambos lados, tratando de conservar una distancia igual a ambos lados tal cual se demostró en el capítulo 1,

Sector 2 se analizó el hito de puertas, aunque en la bibliografía no se encontró una técnica en particular que sirva para este elemento, siempre se tiene en cuenta la seguridad del robot, que pueda pasar a través de este espacio sin sufrir daño alguno, en el caso de puertas se estableció prioridades del comportamiento hacia la meta, donde se miraba si el espacio que se abre en la pared, si daba el peso adecuado para arriesgar el cumplimiento de la tarea asignada, para el caso específico que se está mostrando.

Sector 3 se dio prioridad a la navegación soportada en el comportamiento de seguimiento de pared derecha y evasión de obstáculos, siempre teniendo en cuenta la importancia del comportamiento de “Ir hacia la meta”.

Para el sector 4 se le brindo gran importancia al comportamiento de evasión de obstáculos, dado que si se le permite entrar a la plataforma a este espacio, fácilmente puede caer en un mínimo local (Fig. 2.11), lo cual sería un gran problema para el cumplimiento de la tarea asignada.

En el sector 5 se tomó como principales comportamientos el “ir hacia la meta”, y “búsqueda de espacios libres”, esto en su respectivo orden, con ello se buscaba que la plataforma convergiera lo más rápido posible a su último tramo de la misión y así poder completar la tarea que se le programo.

Se puede decir que los comportamientos que más predominan en cada uno de los sectores son los mostrados en la tabla 2.1, aunque de acuerdo a como llegue el robot en cada ciclo de control se pueden presentar diversas particularidades o mediciones de cada uno de los sensores. De acuerdo a la importancia que se vislumbre en cada caso, los pesos a_i se distribuyen como se muestra a continuación.

Tabla 2. 1: Comportamientos predominantes en cada Sector

Sector	Descripción	a1	a2	a3	a4	a5
1	Corredor horizontal y Robot orientado hacia la meta.	0.6	0.1	0.2	0.5	0.5
2	Sin obstáculos visibles hacia la meta, pero con el riesgo de encontrarse con puertas y posibilidad de búsqueda de espacios libres.	0.9	0.1	0.6	0.3	0.3
3	Muro a la derecha, evasión de obstáculos y robot en dirección hacia la meta y orientado hacia arriba	0.3	0.8	0.4	0.9	0.0
4	Posibilidad de caer en un mínimo local	0.3	0.9	0.0	0.0	0.0
	Seguimiento de pared Izquierda y hacia la Meta	0.3	0.2	0.0	0.0	0.8
5	Dirección de ir hacia la meta y posibilidad de búsqueda de espacios libres	0.9	0.0	0.6	0.0	0.0

2.6 ANALISIS DE RELEVANCIA

Lo que se busca con este paso es quitar características poco relevantes causadas por ruido, variables muy correlacionadas o distracciones, con el fin de para ahorrar tiempo de procesamiento y memoria.

Dentro del análisis de relevancia se identificaron zonas dentro del entorno que cumplen con unas características específicas y en las que el comportamiento del robot puede ser el mismo lo cual da para estos casos variables que estén estrechamente correlacionadas. De esta forma se facilita la obtención de datos para el operador humano y para el software implementado en Matlab, ya que utiliza los mismos coeficientes a_i para situaciones similares. Esta estrategia evita el problema de asignar diferentes coeficientes a entradas similares de los sensores, lo que dificultaría el aprendizaje de la red. Todo el análisis se elaboró con el programa Código R el cual está orientado a temas estadísticos.

Al usar una red neuronal se evita tener que crear un sistema de reglas que tenga en cuenta todas las configuraciones que se puedan presentar. El operador humano es capaz de discernir cómo deben actuar los comportamientos ante los casos usados durante el aprendizaje y la red neuronal se encargará de decidir qué hacer en aquellos casos en que se presenten.

Para Lo cual después de realizar un análisis de relevancia se concluyó que con tener solo 178 de esos puntos y/o características, la red neuronal podría ser entrenada, incluyendo los casos en donde aparecían los hitos tales como pasillos y puertas.

Una vez ya se tenían los datos con los cuales se deseaba entrenar la red, estos fueron llevados a Matlab, a un archivo .Mat, para que se tuvieran presentes y fuese más fácil la manipulación de estos datos.

Para el entrenamiento de la red neuronal fueron utilizados la totalidad de los puntos o características extraídos y obtenidos después del análisis de relevancia, puesto que se realizaban pruebas y si alguno de ellos no se incluía, el robot no lograba el cumplimiento de la tarea asignada, para poder probar el correcto funcionamiento del entrenamiento de la red, se colocaron puntos aleatorios de inicio y de llegada en los entornos mostrados en la Fig. 2.15 y se evaluaba el funcionamiento y cumplimiento de la misión que se le determino.

Tabla 2. 2: Almacenamiento de Datos para validación de entrenamiento de la RN

patrón	entradas									salidas				
	s1	s2	s3	s4	s5	s6	s7	s8	ang	a1	a2	a3	a4	a5
1	26,5	26,5	26,5	26,5	26,5	19	19	26,5	29,1	0,2	0,1	0,9	0	0
2	26,5	26,5	26,5	26,5	26,5	19	19	26,5	32	0,2	0,1	0,9	0	0
3	26,5	26,5	26,5	26,5	26,5	26,5	26,5	26,5	35,5	0,9	0,2	0	0	0
4	26,5	26,5	26,5	26,5	26,5	26,5	26,5	26,5	24,8	0,9	0	0	0	0
5	26,5	26,5	26,5	26,5	26,5	26,5	26,5	26,5	16,5	0,2	0,9	0,4	0	0
6	15	23	26,5	26,5	26,5	26,5	26,5	26,5	3,18	0,3	0,8	0,4	0,9	0
7	11	15	26,5	26,5	26,5	26,5	26,5	26,5	-11	0,3	0,8	0,4	0,9	0
8	15	15	26,5	26,5	26,5	26,5	26,5	26,5	-25	0,3	0,8	0,4	0,9	0
9	26,5	11	19	26,5	26,5	26,5	26,5	26,5	-49	0,3	0,8	0,4	0,9	0
10	26,5	26,5	19	26,5	26,5	26,5	26,5	26,5	-53	0,9	0,7	0,1	0,5	0
11	26,5	26,5	15	26,5	26,5	26,5	26,5	26,5	-44	0,9	0,4	0,1	0,3	0
12	26,5	26,5	23	26,5	26,5	26,5	26,5	26,5	-25	0,9	0,4	0,1	0,3	0
13	26,5	26,5	26,5	26,5	26,5	26,5	26,5	26,5	-16	0,9	0	0	0	0
14	26,5	26,5	26,5	26,5	26,5	26,5	26,5	26,5	-1,6	0,9	0,2	0	0	0

.		
.		
.		
176	26,5	26,5	26,5	26,5	26,5	19	26,5	26,5	10,5	0,9	0,2	0	0	0
177	26,5	26,5	26,5	26,5	26,5	11	26,5	26,5	2,83	0,9	0,2	0	0	0
178	26,5	26,5	26,5	26,5	26,5	15	26,5	26,5	5,2	0,9	0,1	0	0	0

Es de anotar que cada uno de los comportamientos por si solos entrega una velocidad lineal y una velocidad angular y que el módulo de control se encarga de ponderar cuanta importancia se le da a cada uno en un instante específico, la forma como podrían mezclarse se explican en el siguiente literal.

2.7 SISTEMA CONTROL VELOCIDAD:

2.7.1 Lineal

Tal cual se mencionó en el anterior apartado, la salida del módulo supervisor de fusión es un conjunto de pesos a_i que indican la importancia de cada comportamiento en esa iteración o periodo de control. A su vez, cada comportamiento entrega un vector c_i con los valores de velocidad lineal y angular. Todos estos valores pasan al módulo de fusión de comportamientos (Fig. 2.1) el cual para este primer caso, elabora un promedio ponderado, donde n corresponde al número de comportamientos con que se está trabajando, para el presente trabajo, esa n corresponde a 5.

$$c = \frac{\sum_{i=1}^n a_i \cdot c_i}{\sum_{i=1}^n a_i} \quad (2.9)$$

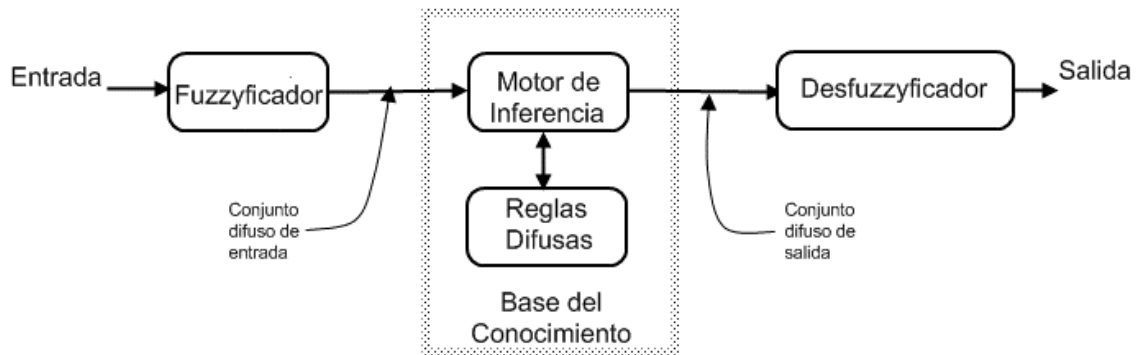
El resultado final es un vector c que posee la velocidad lineal y angular para guiar al robot durante el siguiente periodo de control, con el fin de acercarlo más al cumplimiento de la misión asignada.

2.7.2 Lógica difusa o borrosa

En el contexto del control de robot móvil, un sistema basado en lógica difusa tiene la ventaja de que permite que la naturaleza intuitiva de la navegación de la plataforma

robótica sea fácilmente modelada usando una terminología lingüística. Las cargas computacionales de los sistemas de inferencia fuzzy típicos son relativamente ligeras. Para este proyecto se hace necesario vincular el mundo real con la aplicación de inferencia difusa, se incorporaran los elementos de fuzzyficación y defuzzyficación que se adapte mejor (Barragá Piña, 2009), ver fig. 2.17.

Fig. 2. 17: Esquema de Inferencia Difusa



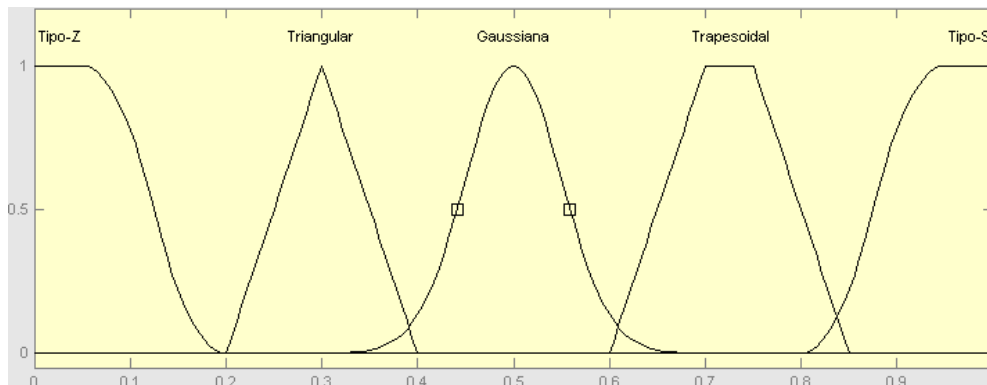
Como resultado de los 2 elementos mencionados anteriormente, los sistemas de control difusos permiten tomar decisiones inteligentes en tiempo real, permitiendo al robot un movimiento suave e ininterrumpido (Mora, 1998), esto se debe a que su idea no radica en establecer la pertenencia de un elemento a un conjunto, sino por el contrario, evalúa el grado de pertenencia que tiene el elemento a dicho conjunto (Barragá Piña, 2009).

Para poder saber si un elemento pertenece a un conjunto difuso se utiliza una **función de pertenencia** (Fig. 2.18). Ésta, denotada con la letra griega μ , determina el grado de pertenencia que posee un elemento x dicho conjunto A (Barragá Piña, 2009).

- x no pertenece a A si $\mu_A(x) \geq \alpha$
- x pertenece a A si $\mu_A(x) \leq \beta$
- x tiene un estatus indeterminado con respecto a A si $\beta < \mu_A(x) < \alpha$

Dado los temas que se tratan en el presente proyecto, es conveniente indicar que el grado de pertenencia de un elemento a un conjunto difuso, no es equivalente al significado de la probabilidad de que el mismo elemento pertenezca a un conjunto normal

Fig. 2. 18: Ejemplos de Funciones de Pertenencia



Para tratar de optimizar más el módulo de fusión de comportamientos el cual es el encargado de brindar la velocidad lineal y angular a la plataforma, se optó por elaborar y conectar al bloque de redes neuronales, un módulo basado en lógica difusa, que le permitiese al robot realizar movimientos más suaves y aprovechar en algunos periodos de control el aumento de la velocidad lineal sin que estos sean muy abruptos (Gonzalez Acevedo, 2007) (esto quiere decir que en diferentes periodos control la velocidad puede variar, pero en el mismo periodo de control, la velocidad es la misma), logrando así, llegar a la meta en menos ciclos, el ahorro de energía y que el sistema realice movimientos menos bruscos, esto con el propósito de ampliar al máximo la vida útil de los componentes del robot. Además de ello se debe lograr que el control de movimiento del robot móvil le permita moverse entre obstáculos en entornos estructurados con movimientos sin colisiones como prioridad y que la estrategia de control inteligente basada en lógica difusa para implementar computacionalmente el razonamiento aproximado permita manejar la incertidumbre inherente en el problema de evitar colisiones.

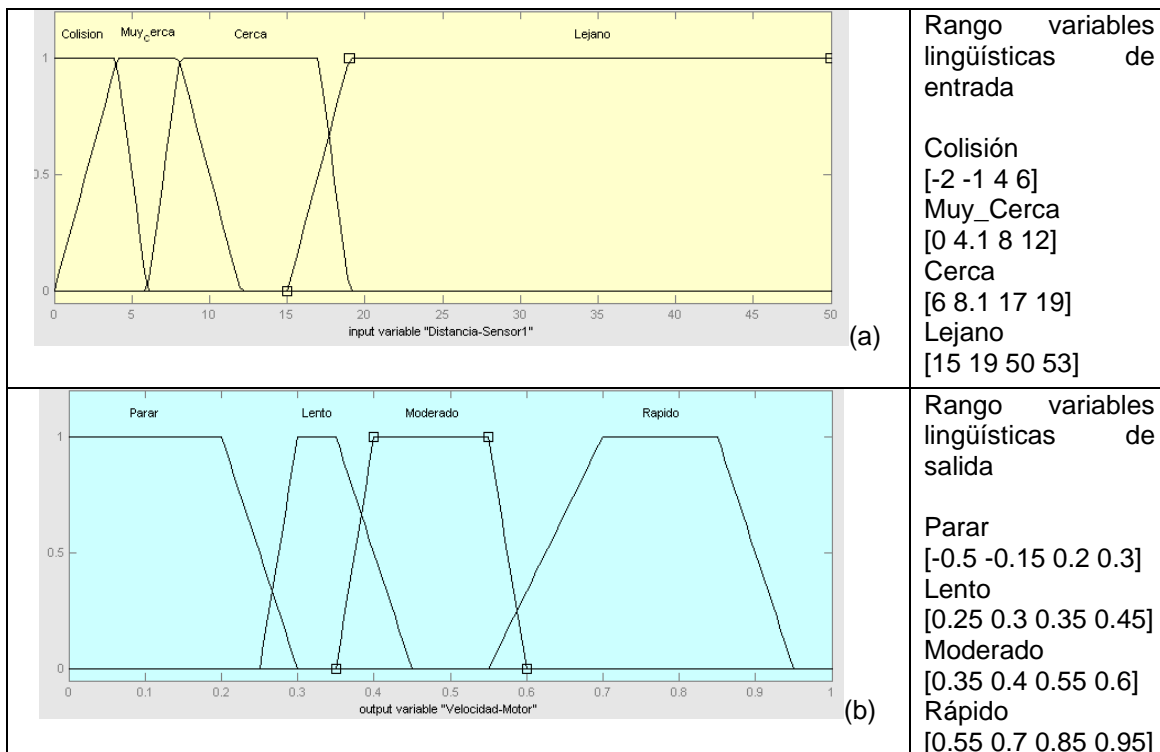
Con lo anteriormente mencionado las variables o términos lingüísticos que se trabajaron son la velocidad lineal para el movimiento de la plataforma, la cual podrá tomar etiquetas lingüísticas tales como *Parar*, *Lento*, *Moderado* y *Rápido*, como se ve en la fig. 2. 19b, donde cada una de ellas se ha representado con una función de pertenencia diferente, el intervalo de maniobra esta entre 0 y 1, lo cual correspondería al porcentaje de ciclo de dureza del PWM que se le entregara al drivers de los motores, teniendo en cuenta que las características del robot simulado son similares a las del real.

En lo referente a las entradas del módulo fuzzy (ver Fig. 2.19 a) están sujetas a las medidas entregada por los sensores de la plataforma, los cuales miden entre 0 y 50 cm.

La variable lingüística que se usó para las entradas fue Distancia, la cual se distribuyó en 4 variables lingüísticas, siendo estas *Colisión*, *Muy_Cerca*, *Cerca* y *Lejano*, las principales ideas que se tuvieron en cuenta para la elaboración de este módulo fueron:

- Si el robot está en la zona de riesgo para la plataforma, este debe *parar*
- Si hay un obstáculo *muy cerca* del robot, el robot debe reducir ostensiblemente la velocidad para evitar la colisión.
- Si hay un obstáculo *cerca* del robot, el robot debe reducir la velocidad para no comprometer la seguridad de la plataforma y así esquivar dicho obstáculo.
- Si hay un obstáculo a una distancia segura para el robot (en caso que no sea necesario evadirlo por el momento) este debe ir a una velocidad normal.
- Si no hay presencia de obstáculo o el obstáculo está *lejos* del robot, la plataforma debe incrementar la velocidad, esto sin afectar las consideraciones mecánicas del robot, siguiendo la trayectoria para alcanzar la posición objetivo.

Fig. 2. 19: Variable Lingüística Velocidad Lineal del Robot

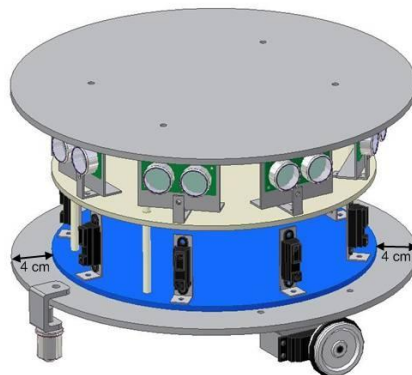


Se debieron analizar muy detenidamente los 4 casos mencionados anteriormente, para ello en este aparte se indicaran los aspectos que se tuvieron en cuenta.

2.7.3 Obstáculos en zona de colisión

Las características y parámetros del robot simulado son similares a las del robot real, es de anotar que la distancia mínima a la cual debe encontrarse cualquier tipo de obstáculo es de 4 cm en cualquier dirección, puesto que ya a esta medida se comenzaría a arriesgar la seguridad de la estructura mecánica de la plataforma, si se llega a esta medida, el robot deberá detenerse. Tal cual se muestra en la fig. 2.20 la plataforma se comenzó a diseñar, y se notó que en algunos casos el robot en la vida real podría parar muy abruptamente cuando llegara a este rango, por ello se optara en la función de pertenencia de salida “parar” llevarla hasta el 20% del PWM, para que la plataforma redujera su velocidad más suavemente y posteriormente si se lleva al 0%. Para que el robot pueda mantener una velocidad mínima constante, el porcentaje del PWM que se le entregaría debiera superar el 35 a 40% del PWM.

Fig. 2. 20: Zona de seguridad de la plataforma



2.7.4 Obstáculos o Hitos muy cercanos

Es de tener en cuenta que hay algunos obstáculos difíciles de evitar a la hora de estar realizando una tarea, más aun si es el caso por ejemplo de un corredor, para nuestro caso, la zona donde se pueden encontrar los objetos muy cercanos se encuentran entre 4 y 8 cm de toda la periferia de la plataforma, en esta zona el robot se movería al 40% de

su velocidad máxima. Es de anotar que el mínimo al cual arrancan los motores es el 30%, porque si no los motores no son capaces de romper la inercia propia de su construcción, por ende el robot no puede desplazarse.

2.7.5 Obstáculos cercanos

Esto involucra objetos que estén a una distancia superior a 8 cm e inferior a 17 cm esto teniendo en cuenta toda la curvatura del robot pero como prelación los delanteros, en este caso la velocidad que podría tomar la plataforma es del 50% al 60% de la velocidad máxima generada por los motores y donde se debe tener en cuenta el peso generado por la propia estructura del robot.

2.7.6 Obstáculos alejados o no presentes

Esto involucra obstáculos que se miden por distancias superiores a los 17 centímetros o para aquellos casos en donde no se detecta objeto alguno, para este caso al robot se le permitirán alcanzar velocidades entre el 70% y el 80% de la velocidad máxima dado que para velocidades superiores, la plataforma podría sufrir daños en su estructura, además de propagación de errores causados por problemas mecánicos o vibración, en este caso la plataforma debería moverse hacia la meta siguiendo el trayecto más corto.

También hay que tener en cuenta que los motores y servomotores que se usaran en la plataforma, solo poseen un comportamiento lineal hasta aproximadamente un 85% de la velocidad máxima, elemento que también se tiene en cuenta para el análisis (Angulo Usategui, 2000), porque de lo contrario se tendría que diseñar otro control adicional para brindarle seguridad a la plataforma.

Si se tiene en cuenta la fig. 2.16, donde el primer componente del sistema fuzzy es el fuzzificador (también denominado borrosificador) el cual es el encargado de crear un puente con el motor de inferencia borrosa que genere los conjuntos borrosos a partir de entradas no borrosas (Barragá Piña, 2009), para nuestro caso los datos provenientes de los sensores; existen varias formas y tipos de fuzzificar, de ellos para el proyecto se trabajara con la posibilidad de emplear la capacidad de la lógica borrosa para manejar la

incertidumbre como método de borrosificación, en donde se sustituye la variable numérica de entrada por un conjunto borroso que represente mediante una distribución de posibilidad (función de pertenencia) los posibles valores que ésta podría tomar realmente, esto porque este tipo de borrosificador resulta útil para casos donde hay que tratar con un grupo de medidas que se encuentran en un rango y además que permite incorporar la incertidumbre propia de toda medida causada por ruidos u otros factores.

La función de pertenencia que se escogió para establecer el grado de pertenencia de los valores (entrada y salida) a los términos lingüísticos fue la trapezoidal, dado que permite trabajar por rangos tal cual se necesita en este trabajo, además de poseer un funcionamiento lineal por tramos, lo cual involucra un menor costo computacional, elemento que no nos brindan algunas funciones como la gaussiana, S y Z. Otro de los elementos que indica por que se escogió una función que fuese lineal, es que en otras funciones existen múltiples divisiones y potencias, lo cual contribuiría en la propagación de errores en el caso que estos se presentaran.

El bloque 2 (Fig. 2.17) del módulo difuso que se implementó corresponde al módulo de inferencia borrosa o la base del conocimiento el cual consta del motor de Inferencia difusa (FIS) y el conjunto de las reglas difusas (Anexo 2), de ellos es importante resaltar:

El sistema difuso que se utilizó es el modelo Mamdani, el cual permite seleccionar los métodos de fusificación, inferencia y defusificación, además de tratar con facilidad las entradas y salidas del sistema.

Para Mamdani, el grado de verdad de $p \rightarrow q$ es idéntico al de la proposición A y B. Se Puede justificar esto diciendo que, para Mamdani, una condición tan sólo resulta cierta cuando el antecedente es cierto y el consecuente también, como se muestra en la ecuación 2.10.

$$p \rightarrow q \equiv p \wedge q \Rightarrow \mu_{p \rightarrow q}(u, v) = \min(\mu_A(u), \mu_B(v)) \quad (2.10)$$

Este metodo contiene 4 pasos, los cuales son explicados en este documento

1. Fuzificación de las variables de entrada.

2. Evaluación de las reglas.
3. Agregación de las salidas de las reglas.
4. Defuzificación.

2.7.7 Respecto al Motor de Inferencia:

Para entender el funcionamiento como se trabajó con el motor de inferencia se debe entender que una regla difusa o borrosa esta expresada bajo la estructura condicional: “si sucede x Entonces resultara y”, donde x es la condición y se conoce como antecedente, mientras y es el resultado y se conoce como consecuente, bajo esta acotación se explicara los procesos que se realizan en este bloque:

1. Como las entradas están representadas por una función basada en tramos se aplican operadores de intersección sobre los conjuntos.
2. Cuando ya se tengan todos los antecedentes de cada regla, se aplican los operadores Y u O, Esto con el fin de establecer su grado de activación.
3. Se aplica el operador de implicación, el cual para nuestro caso fue el “mínimo”, para el cual la función de pertenencia del consecuente es recortada a la altura definida por el grado de veracidad de la regla.
4. Se combina el resultado de todas las reglas y de allí se obtiene un único conjunto borroso, este proceso se realiza aplicando el operador de agregación, el cual en este proyecto fue el máximo.
5. En caso de que la aplicación necesite que resultado sea numérico, se debe hacer uso de algún método de defuzzificación para obtener dicho valor.

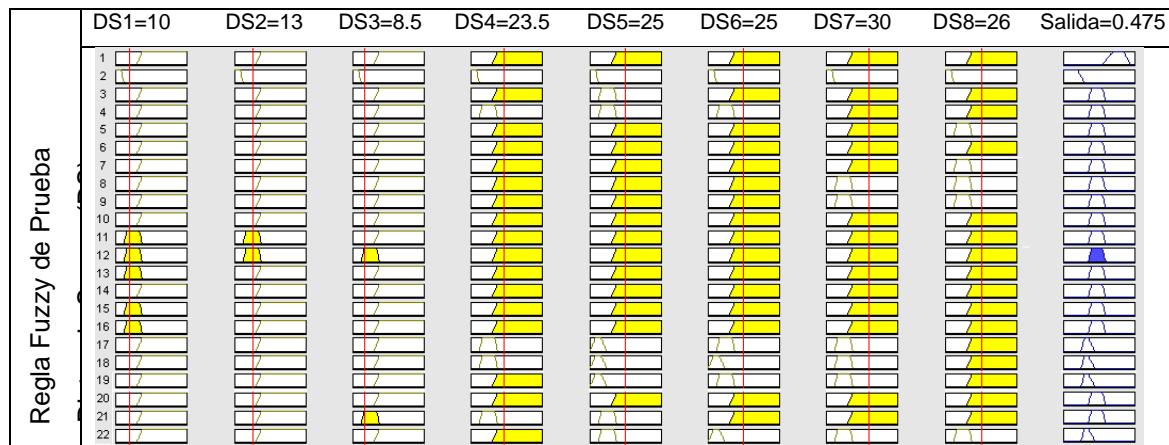
En relación al conjunto de reglas difusas

La base de conocimiento que se implementó, está conformada por las funciones de pertenencia señaladas anteriormente y por el conjunto de reglas difusas que se pueden ver en anexo 2, donde se detalla el comportamiento de la plataforma robótica. Esta base almacena las relaciones entre las entradas y salidas (Correa, 2013).

El total de las reglas para el sistema difuso implementado, estaría determinado por la cantidad de variables de entrada y la tipificación de las variables lingüísticas, el cual para

este proyecto, se cuenta con 8 variables de entrada tipificadas en 4 variables lingüísticas, las cuales generarían gran multiplicidad de reglas, pero para simplificar la cantidad de estas, se analizaron diversos escenarios y aspectos, dado los ambientes en donde se realizarían las pruebas, logrando que la cantidad de reglas total que se implementaron fuesen 35, un ejemplo de estas reglas y su funcionamiento se puede observar en la Fig. 2.21. Para observar todos los ítems que se tuvieron en cuenta para la simplificación de las reglas, remitirse al anexo 1.

Fig. 2. 21: Ejemplo de regla difusa implementada



Para la elaboración de las reglas se trabaja de forma semejante a la realización de una tabla lógica, donde se forman las combinaciones que se necesiten, teniendo en cuenta las entradas y para la obtención del valor de salida se evalúan las entradas y se busca la respuesta deseada u óptima para la condición que se presente.

2.7.8 Bloque de Defuzificación o desborrificador

El método utilizado para la defuzificación, fue el centro de gravedad, esta técnica posee la virtud de tener un costo computacional de cero (0), lo cual implica que este no depende de la complejidad de las funciones activadas o requiere de muchos ciclos para su cálculo, aspecto de gran relevancia dado que dentro de los aspectos importantes para este trabajo es el que se trabaja con procesadores de gama media y la optimización de recursos es uno de los elementos más importantes a tener en cuenta.

El centro de gravedad (CGt) se determina dividiendo la suma ponderada de los productos de cada uno centros de gravedad con su grado de activación (Barragá Piña, 2009), con la suma de todos los grados de activación, como se muestra en la Ecuación 2.11.

$$CGt = \frac{y_1 \cdot Cg_1 + y_2 \cdot Cg_2 + \dots + y_n \cdot Cg_n}{y_1 + y_2 + \dots + y_n} \quad (2.11)$$

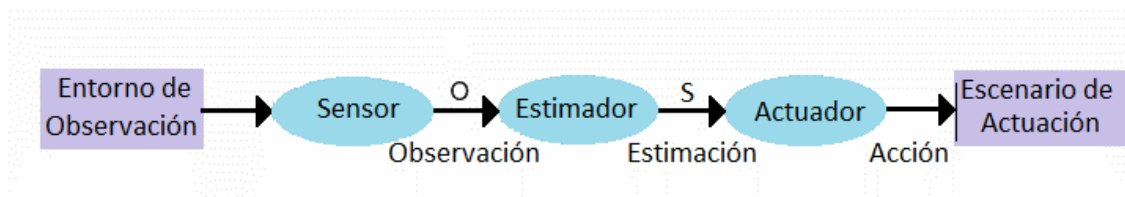
En este proyecto se utilizó este bloque dado que se necesita de un valor para poder brindarle a los drivers de los motores el porcentaje del PWM necesario para mover la plataforma y así cumplir con la tarea asignada, teniendo en cuenta que las características del robot simulado se asemejan a las del robot real.

3. CAPÍTULO 3 ARQUITECTURA ESTADISTICO_PROBABILISTICO

3.1 INTRODUCCIÓN

Los métodos para poder modelar las señales en los diferentes entornos normalmente se dividen en dos tipos: los determinísticos y los estocásticos. Los primeros aprovechan cualidades reconocidas de las señales, mientras que en los estocásticos se trata de modelar las propiedades estadísticas de la señal.

Fig. 3. 1: Diagrama de bloques de un sistema de estimación.



Dentro de esos métodos estocásticos, a su vez existen dos grandes familias predominantes de algoritmos, los probabilísticos y los no probabilísticos. Los primeros han demostrado, en general, ser los más apropiados para problemas de estimación debido a sus propiedades para modelar la incertidumbre en el estado de los objetos a estimar (Aguero, 2009). Otra de las divisiones que se presentan en esta área es si son o no estacionarias según modifiquen o no sus propiedades estadísticas en función del tiempo (Sarmiento Gutiérrez, 2017).

3.2 METODOLOGIA PROPUESTA

3.2.1 Cuantización y agrupamiento de vectores

El robot a medida que comience a recorrer su ambiente, empezara a muestrear su entorno a través de las mediciones que obtenga de sus sensores (infrarrojo, ultrasonido, etc estas son las Observaciones) y el sistema de odometría, a partir de ello cuantizara las medidas obtenidas y las llevara a una lista de vectores, cuando se tiene un numero de terminado de observaciones en la lista, el proceso estocástico, ya se el HMM o el bayesiano podrá determinar en tiempo real el estado más probable a seguir por la

plataforma. Es de anotar que esta lista se va actualizando constantemente con las últimas observaciones obtenidas, mientras que las mediciones más antiguas se van descartando.

Para analizar el sistema implementado, se indica que cada vector de observación \vec{o} que ingresa en la lista es medido cada periodo de muestreo P , el tamaño del vector es conocido, dado que está conformado por la medición de los 8 sensores periféricos para detectar obstáculos y por los datos provenientes de la odometría. Cuando ya se tiene la lista llena, esta es dividida por subconjuntos, estos vectores de un mismo agrupamiento presentan algún tipo de similitud, correlación o cercanía (Sarmiento Gutiérrez, 2017), mientras que vectores de diferentes grupos no presentan ninguna propiedad en común, aproximadamente el tamaño de los componentes de un vector, cada subconjunto es representado por el centroide del mismo (Casarrubias, 2018).

Para nuestro caso el método que se usó para la agrupación de los vectores es el llamado K medias, donde se pretende encontrar el problema de encontrar los K centroides que minimicen la distancia media que puede existir entre los vectores de muestra y los centroides más cercanos a estos. Es de anotar la existencia de otros métodos como son la distancia euclidiana, la distancia euclidiana normalizada, la distancia de Mahalanobis y redes neuronales auto organizables (Sarmiento Gutiérrez, 2017).

Dentro de la Literatura analizada se indicó que el método más usado para desarrollar la técnica K medias es el algoritmo de Lloyd (Casarrubias, 2018) (Sarmiento Gutiérrez, 2017) (Lloys, 2006), abriendo otras opciones como el algoritmo de Elkan (Elkan, 2003).

El algoritmo de Lloyd indica que optimizar la generación de los grupos de manera conjunta es difícil, pero optimizar un grupo dado los demás es relativamente fácil. Este algoritmo necesita para su funcionamiento algunos datos de entrada: los datos de entrenamiento X , la cantidad de centroides k , y la cantidad de iteraciones n o un umbral del error como condiciones de paro (Lloys, 2006).

Es de anotar que este algoritmo puede presentar algunos problemas cuando los puntos están muy dispersos dentro de un grupo o cuando son demasiados número de grupos a procesar, aspecto que no se presentó en el trabajo desarrollado.

Dada las investigaciones que se presentan en (Hamerly, 2015) y (Newling, 2016) donde indican que para $N > 32$ y $N > 50$ respectivamente para resolver el problema de K medias el algoritmo más eficiente es el de Elkan, dado los requerimientos de memoria aleatoria llegando al orden de gigabytes dependiendo de la información a procesar, aspecto con el cual no se cuenta en este proyecto. Como en nuestro caso el N a trabajar es inferior a 32 existen algoritmos más apropiados como el de Lloyd (Hamerly, 2015) que anteriormente fue indicado.

3.2.2 Método de Clasificación

En la navegación autónoma existen variadas técnicas para analizar, escoger y obtener la mejor información que se pueda de los datos y observaciones suministrados por los sensores, esta información debe ser organizada de tal forma que pueda ser posible cotejarla con otras posibles observaciones con el fin de que puedan ser usadas en variados sistemas automáticos que en muchos casos son denominados clasificadores.

En este apartado se mostrara un método de clasificación estadístico o estocástico, se trabajara con el HMM que está basado en secuencias de símbolos para darle un significado semántico a los datos.

3.2.3 Modelo Oculto de Markov (HMM)

Este es un modelo estocástico en cual se presenta un proceso estocástico doble, donde las actuaciones de un primer proceso llamado proceso oculto $S = \{S_t\}$, brindan origen a un segundo proceso denominado proceso observado $O = (O_t)$, en el cual las realizaciones en conjunto de los dos procesos permiten una caracterización medible u observable. Según (Sarmiento Gutiérrez, 2017) "El principal atributo de los modelos ocultos de Markov es que son modelos que asignan probabilidades de aparición a secuencias de símbolos, se dice que son procesos generativos ya que las probabilidades de ocurrencia para cada símbolo, se definen por una serie de pasos que incrementalmente producen la secuencia observada. Este tipo de proceso estocástico se le describe mediante un conjunto de unidades llamadas estados que son el mecanismo

de emisión de los símbolos. Los estados tienen probabilidades de transición asociadas a los cambios internos del sistema durante la generación de los símbolos”.

Los procesos de Markov también llamados cadenas de Markov por el número finito de estados, dependen del número de estados o sucesos previos que se necesitan para tomar una decisión, por ejemplo se denomina de primer orden cuando solo depende del estado anterior para tomar una decisión, este será el caso de nuestro proyecto, para aquellos casos donde se depende de más estados se denominan de orden superior (Por ejemplo el de segundo orden,, es porque necesita conocer que ha pasado en 2 estados anteriores para poder tomar una acción).

Como se mencionó en el párrafo anterior, nuestro proyecto se trata como un proceso de marco de primer orden, para ello se deben de cumplir con una serie de propiedades:

- Que la probabilidad condicional cumpla con el principio de Markov, la cual se muestra en la siguiente ecuación:

$$P[X(t+1) = j \mid X(0) = k_0, X(1) = k_1, \dots, X(t) = i] = P[X(t+1) = j \mid X(t) = i] = P_{ij} \quad (3.1)$$

- Presencia de un número finito de estados.
- El muestreo de las señales se realicen en periodos iguales.
- Las probabilidades P_{ij} , son constantes con respecto al tiempo o período de medición (proceso homogéneo respecto a la variable tiempo).

Para analizar un HMM de primer orden se debe entender este como $\lambda = (V, S, A, B, \Pi)$, donde:

1. Tamaño de estados N: $S = \{s_1, s_2, \dots, s_N\}$
2. notación finita de símbolos M: $V = \{v_1, v_2, \dots, v_m\}$
3. $A = [a_{ij}]_{N \times N}$ Es una matriz de probabilidades de transición donde el elemento a_{ij} es la probabilidad de transición del estado i al estado j , con.

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i) \text{ para todo } 1 \leq i, j \leq N. \tag{3. 2}$$

4. La probabilidad del símbolo emitido en el estado j: $b_j(k) = P(V_k \text{ en } t | q_t = S_i)$

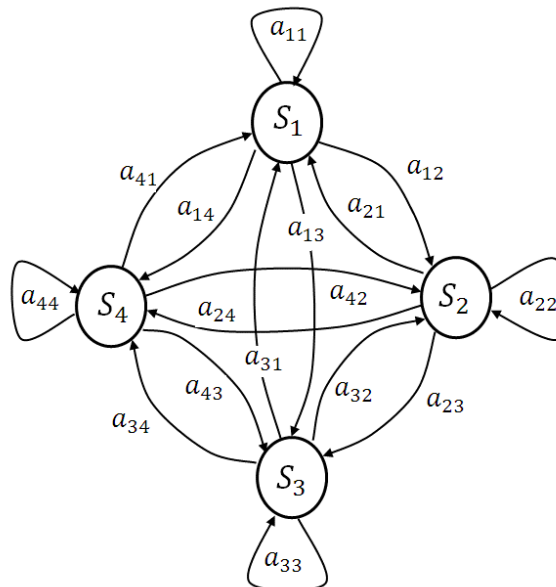
5. Matriz de probabilidades iniciales donde

$$\Pi_i = P(q_t = S_i) \text{ con } 1 \leq i \leq N \tag{3. 3}$$

Con lo anteriormente mostrado se puede indicar que un HMM como una secuencia observada $O = \{O_1, O_2, \dots, O_T\}$ es la única secuencia que se puede observar, posea un tamaño T y donde cada elemento O podrá ser un símbolo en los posibles V, Con lo anterior el modelo queda definido en función de A,B y Π , quedando $\lambda=(A,B, \Pi)$.

Dentro de los modelos ocultos de Markov existen diversas topologías, como la ergódica, lineal, Izquierda a derecha y Bakis, teniendo en cuenta que en aplicaciones automatizadas de reconocimiento de movimiento y reconocimiento de actividad, los datos de entrada a procesar tienen una estructura cronológica o lineal (Figuroa Angulo, 2015), con lo anterior se optó por usar la tipo ergódica, la cual se muestra a en la fig. 3.2.

Fig. 3. 2: Topología HMM Ergódica



Con lo anterior la plataforma robótica navegará a través del entorno donde podrá realizar la tarea, mientras captura un número O de observaciones. Este número O será definido

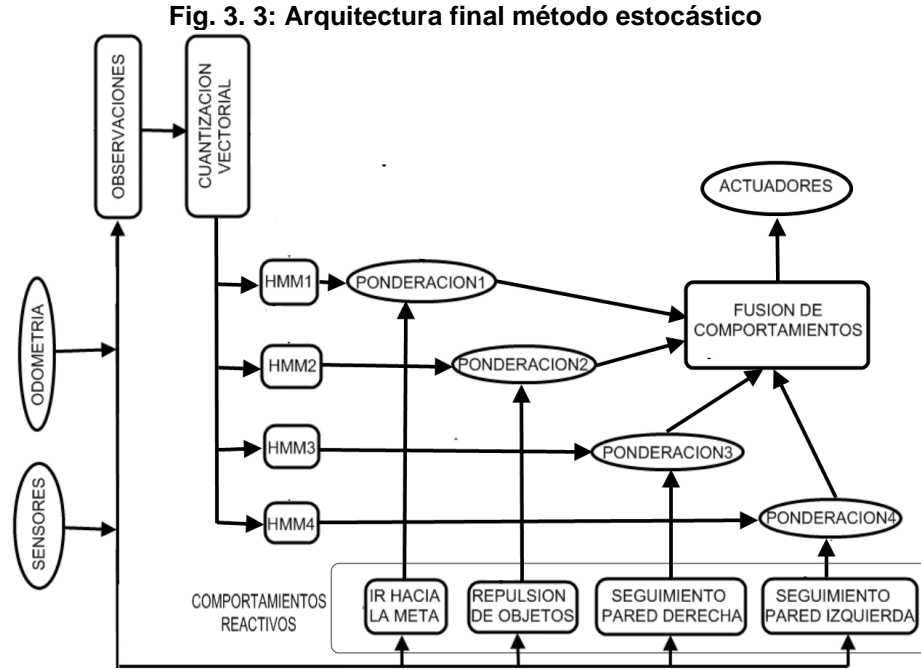
más adelante, dado que es un ítem importante para la realización de la misión, estas observaciones son almacenadas, cuantizadas y procesadas para ir navegando a través del ambiente (Casarrubias, 2018), esto con el fin de ir generando el Modelo oculto de Markov del mismo (HMM).

En si la estrategia para aplicar el HMM para aplicaciones de robótica móvil (Navarro García D. A., 2009) consiste en:

1. Un grupo de posibles estados S que la plataforma podrá ir tomando dentro de su entorno, a medida que va navegando a través de ella, estos estados serán los nodos que va tomando en cada periodo de control.
2. Un numero finito de movimientos V que el robot realizara mientras pasa de un estado a otro, estas acciones serán actividades tales como girar en una dirección, Ir hacia la meta, navegar a través de un pasillo, atravesar una puerta, entre otros, acá es donde se ve el trabajo de los comportamientos reactivos.
3. La matriz de transición A de estado en un momento t que modela la probabilidad para que el posible nuevo estado en el sistema S_j , ya que se está en un estado S_i dado que se tomó la acción provista por q_t , esto dada por la ecuación 3.2.
4. Una función de recompensa B , en donde dado un estado S y una acción V , determina una recompensa adquirida por la plataforma tratando de alcanzar el objetivo asignado.
5. La siguiente agrupación de observaciones O obtenidas por el robot a través de sus sensores y su sistema de odometría y así obtener una nueva matriz de probabilidades iniciales dadas por la ecuación 3.3.

Para la solución final mediante el método estocástico se usaron los comportamientos reactivos Ir hacia la meta, repulsión de objetos y seguimiento de pared derecha e izquierda, el comportamiento de búsqueda de espacios libres se omitió, dado que cuando se trataba de usar, la plataforma comenzaba a deambular, dada la naturaleza propia de

este comportamiento, la navegación de la plataforma no lograba obtener los resultados esperados. Para cada uno de los comportamientos se utiliza un HMM, quedando la arquitectura final tal como se ve en la siguiente fig. 3.3.



3.3 PROBLEMAS FUNDAMENTALES CON HMM

Como se busca solucionar un problema que tal vez cumpla con la propiedad de Markov, existen 3 problemas que interesan resolverse:

1. Calcular la probabilidad de una secuencia, ósea que como se conoce una secuencia observada O y un modelo λ , se debe determinar $P(O | \lambda)$.
2. Establecer la secuencia de estados más probable que genera las observaciones, ósea que como se tiene una secuencia observada O y un modelo λ , calcular Q óptimo para generar un nuevo O .
3. Hallar un nuevo modelo λ que permita maximizarse, es decir encontrar $P(O | \lambda)$ ajustando A, B y Π .

3.3.1 Solución al problema 1

Dentro de la bibliografía se ofrecen algunas soluciones como el algoritmo de adelanto (Casarrubias, 2018), procedimiento de hacia adelante hacia atrás (Sarmiento Gutiérrez, 2017), entre otros.

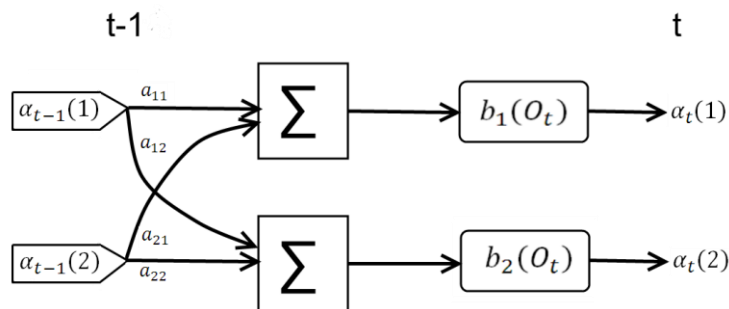
Dado lo analizado y teniendo en cuenta el trabajo que se debe realizar, se optó por trabajar con el procedimiento de hacia adelante hacia atrás llamado también en inglés Forward-Backward procedure.

Este proceso tiene 2 partes, la primera de ellas es hacia adelante definida como $\alpha_t(i)$, esta nos permitirá hallar la suma de las probabilidades de todos los posibles caminos. Se define como la probabilidad de que el modelo genere la salida hasta que en $t=t_0$ y que en t_0 se esté en el estado i (Salcedo, 2006), con:

$$\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda) \quad 1 \leq t \leq T \quad (3.4)$$

Si se desea hallar la variable hacia delante, en cada iteración se tiene que hacer $N*N$ productos (probabilidades de transición), $N*N$ sumas y N productos (probabilidad de generar la observación) luego el nuevo algoritmo requiere sólo $2N*(N1)*T$ operaciones, es lineal con el tiempo y cuadrático con el número de estados.

Fig. 3. 4: Esquema que simboliza una iteración en el cálculo recursivo de la variable hacia delante



Con lo anterior se obtiene

$$P(O|\lambda) = \sum_j P(O_1 O_2 \dots O_n, q_t = j|\lambda) = \sum_j \alpha_T(j) \quad (3.5)$$

Quedando un procedimiento iterativo y factible de la siguiente forma (Sarmiento Gutiérrez, 2017) (Salcedo, 2006):

1. Inicializacion:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (3.6)$$

2. Induccion:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1, 1 \leq j \leq N \quad (3.7)$$

3. Terminacion:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_t(i) \quad (3.8)$$

La variable hacia atrás no se usa para solucionar el problema de evaluación, pero es importante para resolver la cuestión del aprendizaje de un modelo sobre las observaciones obtenidas un caso de ser necesario.

3.3.2 Solución al problema 2

Se busca determinar la secuencia de estados más probable o en algunos casos la que sea óptima conociendo una secuencia de observaciones y el modelo λ , de acuerdo a la literatura, casi todos los autores concluyen que la mejor forma es a través del algoritmo de Viterbi (Casarrubias, 2018) (Sarmiento Gutiérrez, 2017) (Kanungo, 1999) (Figuroa Angulo, 2015).

El algoritmo de Viterbi usa la variable δ , la cual contiene la probabilidad más alta de una ruta única, en el momento t .

$$\delta_t(i) = \text{Max}_{q_1, q_2, \dots, q_{t+1}} [P(q_1 q_2, \dots, q_t = O_1, O_2, \dots, O_t | \lambda)] \quad (3.9)$$

La probabilidad más alta a lo largo de una sola ruta, en el tiempo $t + 1$, se calcula como:

$$\delta_{t+1}(i) = [\text{Max}_i(\delta_t(i)a_{ij})b_j(O_{t+1})] \quad (3.10)$$

La ecuación anterior busca la mejor probabilidad para toda observación O . Para deducirse se debe rastrear la mejor probabilidad en cada trayectoria posible en el tiempo t tomando en cuenta las trayectorias anteriores, Todo este procedimiento puede elaborarse siguiendo los siguientes pasos:

1. Inicialización:

$$\delta_1(i) = \pi_i b_i O_1, \quad 1 \leq i \leq N \quad (3.11)$$

$$\psi_1(i) = 0 \quad (3.12)$$

2. Recursión

$$\delta_t(j) = \text{Max}_{1 \leq i \leq N} [(\delta_{t-1}(i)a_{ij})b_{ij}], \quad 2 \leq t \leq N, \quad 1 \leq j \leq N \quad (3.13)$$

$$\psi_t(j) = \text{Arg Max}_{1 \leq i \leq N} [\delta_{t-1}(i)a_{ij}], \quad 2 \leq t \leq N, \quad 1 \leq j \leq N \quad (3.14)$$

3. Terminación

$$P^* = \text{Max}_{1 \leq i \leq N} (\delta_t(i)) \quad (3.15)$$

$$q_t^* = \text{Arg Max}_{1 \leq i \leq N} (\delta_t(i)) \quad (3.16)$$

4. Chequeo

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T - 1, T - 2, \dots, 1 \quad (3.17)$$

3.3.3 Solución al problema 3: Aprendizaje del Modelo

La mayor dificultad que se presenta en los modelos ocultos de Markov es poder encontrar un procedimiento para adecuar los parámetros A, B y Π de λ para cumplir con los criterios de optimización, para ello hay varias formas, pero aun no aun un método que sea el mejor (Rabiner, 1993). Dentro de estos métodos sobresalen los algoritmos basados en gradientes (Levinson, 2008), los Expectación Modification (EM) (Dempster, 1977), en (Rabiner, 1993) se usa el algoritmo de Viterbi el cual permite estimar los parámetros a partir de un conjunto de múltiples secuencias de observaciones, pero de acuerdo a la literatura (Casarrubias, 2018) (Sarmiento Gutiérrez, 2017) (Figuroa Angulo, 2015) y al trabajo realizado el método que más se acomoda es el de Baum-Welch que aunque que requiere todas las observaciones de entrenamiento en una sola secuencia, el método es interactivo.

Este método en primera instancia define:

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \quad (3.18)$$

Aprovechando los valores que se obtuvieron iterativamente con los algoritmos de adelanto y de atraso, esta ecuación se reescribe:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) (\beta_{t+1}(j))}{P(O | \lambda)} \quad (3.19)$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (3.20)$$

Ahora se define $\gamma_t(i)$, como la probabilidad de estar en el estado S_i dada la secuencia de observaciones O y el modelo λ y está definida por:

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (3.21)$$

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{Numero esperado de transiciones por } S_i \text{ en } O \quad (3.22)$$

$$\sum_{t=1}^{T-1} \xi_t(i) = \text{Numero esperado de transiciones de } S_i \text{ a } S_j \quad (3.23)$$

Posteriormente se determinan las matrices de transición y transferencia que crearan el nuevo modelo λ , quedando este así:

$$\hat{\pi}_i = \text{Numero de veces que el estado } S_i \text{ se encuentra en el tiempo } t = 1 = \gamma_1(i) \quad (3.24)$$

$$\hat{a}_{ij} = \frac{\text{Numero de transiciones de } S_i \text{ a } S_j}{\text{Numero esperado de transiciones por } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (3.25)$$

$$\hat{b}_j(k) = \frac{\text{Numero de veces que se observa el simbolo } V_k \text{ en } S_j}{\text{Numero de veces en } S_j} \quad (3.26)$$

$$= \frac{\sum_{t=1}^T 1_{O_t=v_k} \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)} \quad (3.27)$$

Como resumen de lo explicado hasta el momento se puede decir que la plataforma no conoce el entorno donde tiene que realizar su tarea, donde la navegación se realiza a través de la combinación lineal de los comportamientos reactivos ponderada por cada una de las probabilidades dependiendo de la situación. El entrenamiento se va realizando por cada una de las mediciones obtenidas por los sensores periféricos y el sistema de odometría en el periodo de control P, formándose la lista \vec{o}_p generándose una secuencia de observaciones v_k la cual se convertirá en la variable observable del HMM, mientras que la variable oculta será la orientación del robot.

3.4 APRENDIZAJE

Es de anotar que para obtener los datos con los cuales se entrenó el sistema estocástico HMM se utilizaron las mismas características usadas para obtener los datos con los cuales se entrenó la RN, la principal y única diferencia está en que el dato que suministraba el algoritmo se tomó como la probabilidad de que si se ejecutara el

comportamiento al cual hacía referencia, es de anotar que todos los datos que entrega en algoritmo están normalizados entre 0 y 1, lo cual facilita la aplicación.

3.5 GENERACION DE RUTA

Con este tipo de método, se busca la optimización de una ruta, la cual se consigue al tener una posición inicial desde donde inicia el movimiento del robot y un punto final donde se procura hallar algún recorrido en el que el robot pueda desplazarse de forma que no sufra colisiones con los obstáculos, y al mismo tiempo el trayecto escogido a seguir tenga la menor distancia en comparación al resto de las posibles soluciones (Correa, 2013).

3.6 SOFTWARE UTILIZADO PARA EL DESARROLLO

3.6.1 Simulación

Tal cual como se elaboró con el desarrollo del sistema Neuro-difuso se usó el Simulador SRM basado en la plataforma Matlab, se acondiciono al desarrollo el Toolbox Hidden Markov Model Toolbox (HMM), el cual Incluye Viterbi, filtro HMM, HMM más suave, algoritmo EM, entre otros.

3.6.2 Robot Real

Fuera de los programas provistos por el IDE Codewarrior se les adaptaron las librerías UMDHMM necesaria para el desarrollo de los algoritmos, esta librería fue presentada en (Kanungo, 1999), la cual es una librería escrita en C, la cual permite trabajar con modelos discretos de Markov ocultos, e implementa los algoritmos Forward-Backward, Viterbi y Baum Welch, entre otros. También se utilizó el paquete KMLOCAL versión 1.7.2, usado en (Figuroa Angulo, 2015), escrito también en C, este permite usar el algoritmo del método K means usado para la cuantización vectorial, De este último paquete cabe rescatar que posee una librería especial llamada Hibrido, la cual es la combinación de las técnicas de Lloyd's y Swap, que permite evitar quedar atrapado en los mínimos locales.

4. CAPÍTULO 4 ANÁLISIS DE RESULTADOS

En este trabajo, se proporcionó a una plataforma robótica la posibilidad de moverse a través de entornos estructurados y con obstáculos estáticos, y principalmente se le provee capacidad de poder moverse a través de hitos como pasillo y puertas en su ambiente de navegación, además dependiendo de las características del entorno puede variar la velocidad de desplazamiento en función de la presencia de obstáculos, es de anotar, que se usó un robot diferencial sobre ruedas el cual posee restricciones no holónomas, consistentes en que la orientación debe ser siempre tangente al trayecto y, el radio de curvatura de la estructura mecánica del robot está limitado a un valor mínimo.

Para el análisis de resultados y verificación de la hipótesis planteada, se compararon cuatro arquitecturas de control diferentes, Entre ellas: navegación reactiva pura-Campos potenciales, navegación neuro-reactiva, navegación neuro-difusa-Reactiva y navegación estocástica. Estos algoritmos inicialmente se probaron en el simulador SRM. Una vez observados los resultados, se corroboró el funcionamiento de las técnicas de control propuestas en la plataforma real, los cuales corresponden a los 2 últimos mencionados anteriormente. En estas pruebas se han utilizado los controladores y una estrategia de control descentralizada y escalable. En un futuro cercano, los experimentos de laboratorio se pretenden probar utilizando el robot GIRA02, este robot está equipado con un anillo de ocho sensores infrarrojos y 8 de ultrasonido que cubren toda su periferia los cuales permiten obtener la información del entorno y se escoge la medida a tomar en cuenta usando un filtro Kalman.

Los programas que se elaboraron como controladores, ósea los elementos de la estrategia deliberativa, suministran varias facultades a la plataforma robótica, tales como la capacidad de evadir obstáculos, transitar a través de pasillos, realizar seguimiento de pared, en la mayoría de los casos estimar si la existencia de una puerta permita la transitabilidad y seguridad de la plataforma robótica y la de elaborar una trayectoria hacia una meta específica, todo aquello también teniendo en cuenta cada uno de los comportamientos con que se trabajaron.

La captura de información en la plataforma robótica se realiza durante cada periodo de control, en ella, se obtiene la posición y la orientación actual del robot (x_i, y_i, θ_i) , también

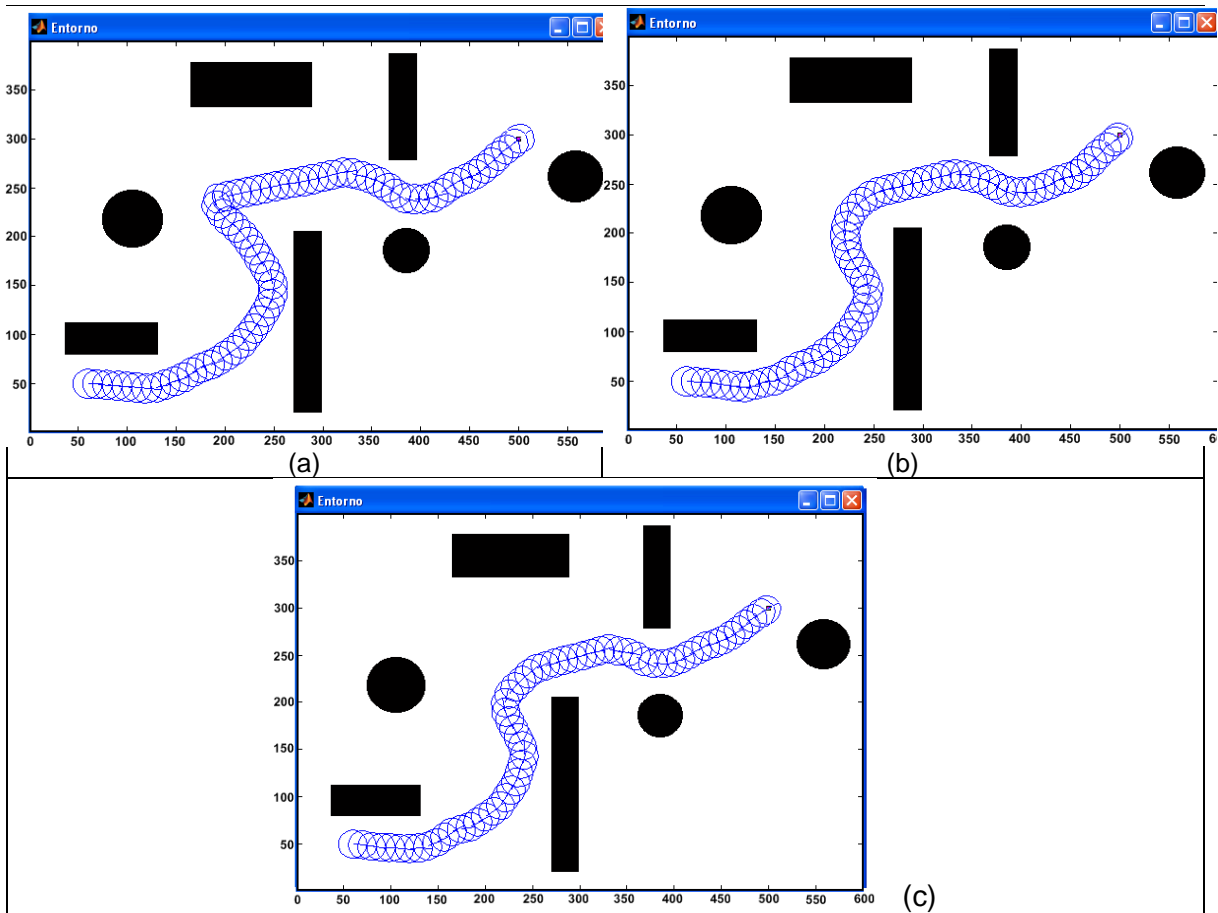
se obtiene la información de los ocho (8) sensores de proximidad, cuya lectura máxima en distancia es de 50 cm., así que cuando el robot utiliza n periodos de control en conseguir llegar a la meta, se obtiene una matriz de $n \times 11$, con n puntos de muestreo por 11 datos (8 sensores distancia y 3 coordenadas x , y , el ángulo de orientación respecto al norte geográfico), para ello la tarea que se le asigna al robot es ir desde un punto inicial hasta un punto de llegada.

En las siguientes figuras, se muestran los resultados que se alcanzan al someter al robot ante diversas misiones, y en las tablas se publican los valores que se obtienen al aplicar las métricas de desempeño a cada una de las tareas de navegación asignadas, las variables proporcionan la siguiente información: las tres primeras permiten analizar la seguridad de la trayectoria, midiendo el riesgo de colisión que corrió el robot durante su desplazamiento hacia la meta. La cuarta, quinta y sexta, miden aspectos de la planificación de la trayectoria.

Para las siguientes pruebas se compararan los 3 métodos que tienen cierto grado de "inteligencia", los cuales son el método neuronal (a), el Neuro-difuso (b) y el estadístico (c).

El presente es un entorno normal donde no se presentan muchas restricciones a la plataforma, pero se puede extraer muy buena información, se observa que el que menos tiempo requiere para llegar a la meta es el neurodifuso el cual posee control tanto en la velocidad angular como la lineal, que el que mejor seguridad le presta a la plataforma es el estadístico, mientras que el neuronal es el método que menos prestaciones le da a la plataforma, dado que es el que más se demora en llegar y el que menos seguridad le brinda a la plataforma. Para este caso el punto de inicio que se estipula fue 50,50 con ángulo 0° y el punto de llegada 500,300, para ninguna de las simulaciones se programa ángulo de llegada.

Fig. 4. 1 Entorno de Simulación 1



Donde los datos que entregaron de acuerdo a las métricas son mostrados en la tabla 4.1

Tabla 4. 1 Entorno de Simulación 1

TECNICA \ METRICA	Sistema híbrido (RNA-comportamientos reactivos)	Sistema híbrido (RNA-Difuso-comportamientos reactivos)	Sistema híbrido (HMM-comportamientos reactivos)
Distancia media a los obstáculos SM1 (cm)	45.7836	45.0697	46.0258
Distancia media mínima a los obstáculos SM2 (cm)	18.2500	20.9566	23.4566
Distancia mínima de cualquier sensor MIN (cm)	7.0000	11.0000	15.0000
Distancia total recorrida PL (cm)	513.2253	509.4373	509.4958
Distancia media al objetivo Mgd (cm)	2.88	5.3625	3.7463
Periodos de Control LeN	179	96	137

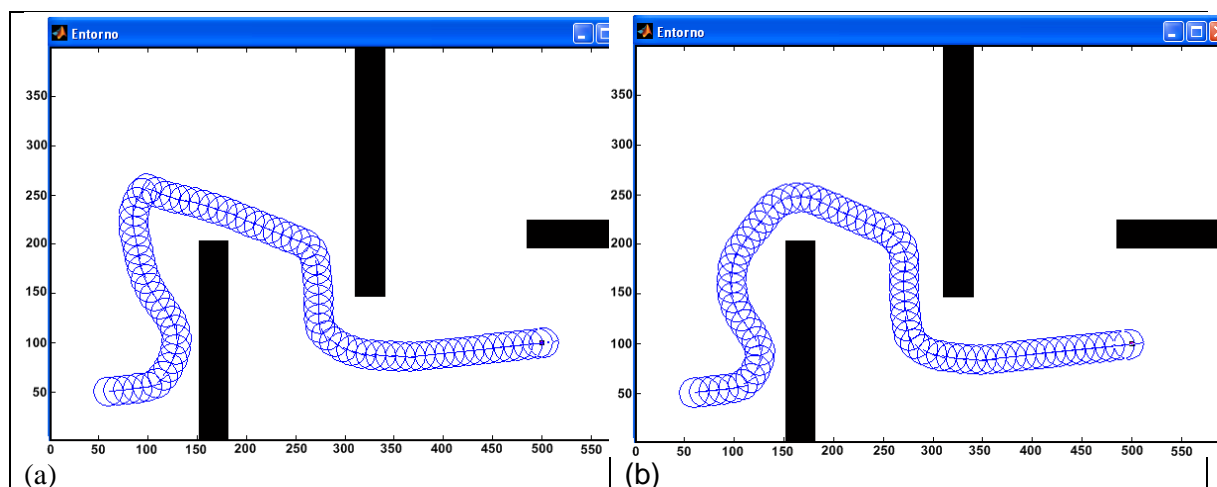
Es de anotar que cuanto se están corriendo los algoritmos para finalizar se le brinda la posibilidad de parar cuando la convergencia está cerca a los 3 centímetros del punto de llegada (esto aplica para todas las simulaciones) como máximo, teniendo esto en cuenta los datos de inicio y llegada que se obtienen se pueden ver en la tabla 4.2.

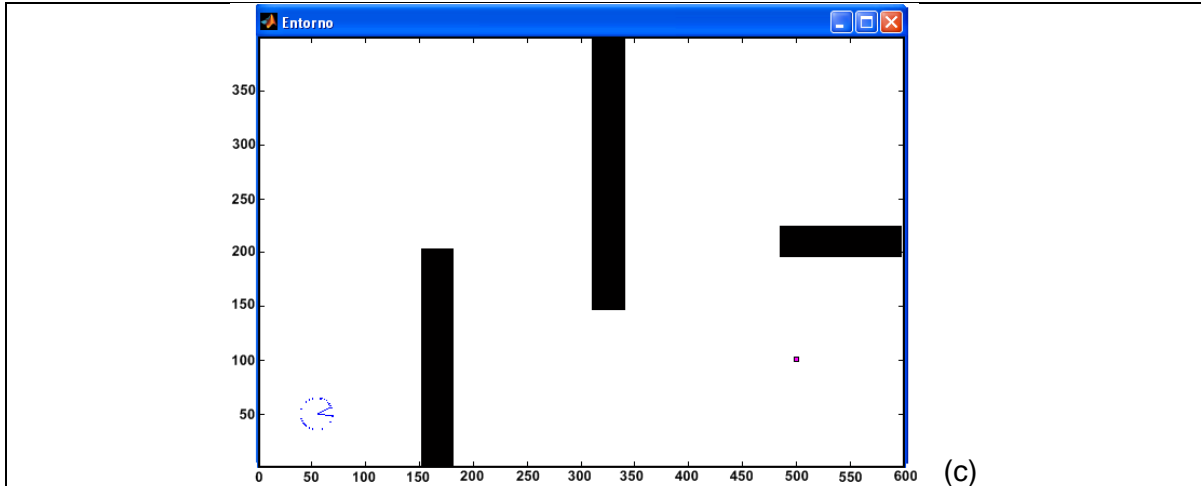
Tabla 4. 2: Datos de inicio y llegada simulación 1

TECNICA \ ITEM TAREA	Sistema híbrido (RNA-comportamientos reactivos)	Sistema híbrido (RNA-Difuso-comportamientos reactivos)	Sistema híbrido (HMM-comportamientos reactivos)
Punto de salida en X (cm)	50	50	50
Punto de salida en Y (cm)	50	50	50
Angulo inicial en grados	0	0	0
Punto de llegada en X (cm)	500,89	498.3154	497.7623
Punto de llegada en Y (cm)	300.14	297.0748	298.1146
Angulo de llegada en grados	25,7	48.2	22,4

Para el entorno de simulación 2 mostrado en la Fig. 4.2, se puede ver que cuando se trabaja solo con el sistema neuronal hay cambios de dirección más abruptos, lo cual en la vida real podría afectar la estructura de la plataforma.

Fig. 4. 2 Entorno de simulación 2





Nota: Del caso (c) no se ponen datos por que no son comparables, ya que es un intento fallido que no logra terminar la tarea asignada.

En este caso cuando el sistema estadístico elaborado necesita hacer curvas tan cerradas presenta problemas, este presenta problemas y el sistema Neuro-difuso sigue funcionando de forma adecuada y los datos obtenidos se presentan en las tablas 4.3 y 4.4

Tabla 4. 3 Entorno de Simulación 2

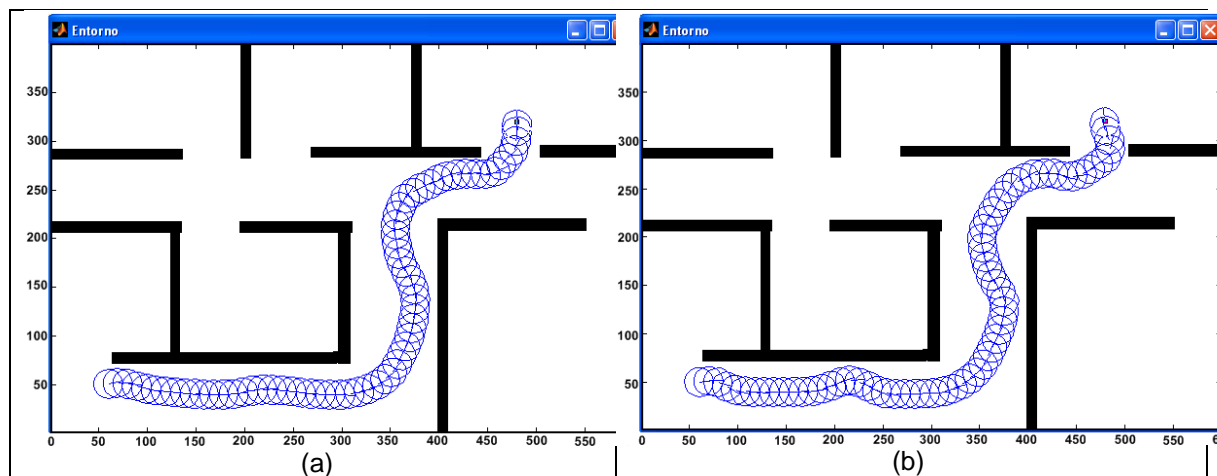
TECNICA METRICA	Sistema hibrido (RNA- comportamien tos reactivos)	Sistema hibrido (RNA- Difuso- comportamien tos reactivos)	Sistema hibrido (HMM- comportamien tos reactivos)
Distancia media a los obstáculos SM1 (cm)	47.7794	47.3549	---
Distancia media mínima a los obstáculos SM2 (cm)	24.4566	25.4566	---
Distancia mínima de cualquier sensor MIN (cm)	7.0000	11.0000	---
Distancia total recorrida PL (cm)	440.7124	435.8741	---
Distancia media al objetivo Mgd (cm)	2.2601	4.0736	---
Periodos de Control LeN	196	108	---

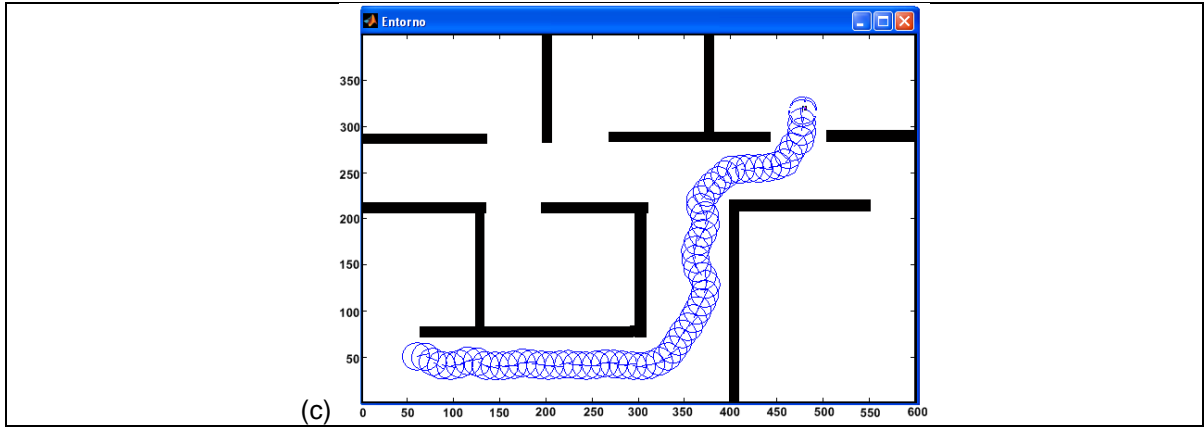
Los datos en cuanto a los puntos de arranque y llegada después de simulada la tarea asignada se muestran a continuación:

Tabla 4. 4 Datos de inicio y llegada simulación 2

TECNICA \ ITEM TAREA	Sistema híbrido (RNA-comportamientos reactivos)	Sistema híbrido (RNA-Difuso-comportamientos reactivos)	Sistema híbrido (HMM-comportamientos reactivos)
Punto de salida en X (cm)	500	500	500
Punto de salida en Y (cm)	100	100	100
Angulo inicial en grados	0	0	0
Punto de llegada en X (cm)	501.5132	497.6822	---
Punto de llegada en Y (cm)	100.1901	99.674	---
Angulo de llegada en grados	12.3	5.2	---

Sobre el tercer entorno de simulación se podrá ver parte del trabajo de esta investigación, el cual era la navegación a través de pasillos y puertas. Para este caso la tarea es arrancar en el punto (50, 50,0) donde los dos primeros corresponden a las coordenadas x e y o el ultimo al ángulo inicial, el punto de llegada es (480, 320), correspondientes a las coordenadas x e y respectivamente, el ángulo de llegada depende de cómo llegue el robot.

Fig. 4. 3: Entorno de simulación 3



En esta simulación podemos ver que las técnicas trabajadas manejan de forma adecuada la navegación a través de pasillos y puertas, también se puede observar como en los casos anteriores que el neurodifuso sacrifica seguridad por velocidad, mientras el estadístico hace una buena ruta sin arriesgar la integridad de la plataforma.

Tabla 4. 5: Resultados métricas sobre entorno de trabajo 3

TECNICA METRICA	Sistema híbrido (RNA-comportamientos reactivos)	Sistema híbrido (RNA-Difuso-comportamientos reactivos)	Sistema híbrido (HMM-comportamientos reactivos)
Distancia media a los obstáculos SM1 (cm)	39.3657	38.1667	39.3446
Distancia media mínima a los obstáculos SM2 (cm)	14.9566	11.2500	14.7500
Distancia mínima de cualquier sensor MIN (cm)	3.0000	3.0000	7.0000
Distancia total recorrida PL (cm)	504.6757	504.3464	503.3954
Distancia media al objetivo Mgd (cm)	2.2135	3.8796	2.3971
Periodos de Control LeM	229	131	211

Tabla 4. 6: Datos de inicio y llegada simulación 3

TECNICA ITEM TAREA	Sistema híbrido (RNA-comportamientos reactivos)	Sistema híbrido (RNA-Difuso-comportamientos reactivos)	Sistema híbrido (HMM-comportamientos reactivos)
Punto de salida en X (cm)	480	480	480
Punto de salida en Y (cm)	320	320	320
Angulo inicial en grados	0	0	0
Punto de llegada en X (cm)	470.0471	478.8466	478.47
Punto de llegada en Y (cm)	317.2131	318.4282	317,257
Angulo de llegada en grados	91	106	85

Para los casos en donde los sistemas no poseen la suficiente capacidad de procesamiento se puede ver que las respuestas que se obtuvieron no tienen la respuesta esperada, sobre todo para uno de los casos en los cuales se le da relevancia al presente trabajo como es el caso de una puerta, para el caso *a* se puede observar que el sistema se queda atrapado en un mínimo local, donde el algoritmo no es capaz de resolver la situación que se le presenta en el momento, para el caso *b* aunque el sistema es capaz de salir de la situación de aprieto, se nota que es demasiado el consumo en energía, esfuerzo y uso forzado de la plataforma, cuando pudiese responder de una mejor forma; mientras que cuando el sistema que se diseña esta adecuadamente entrenado, como en el caso *c*, se observa que en este caso en especial, cumple con la tarea asignada, no hay mayor consumo energético y se ve la suavidad del trabajo realizado por el robot.

Fig. 4. 4 Entornos mal simulados

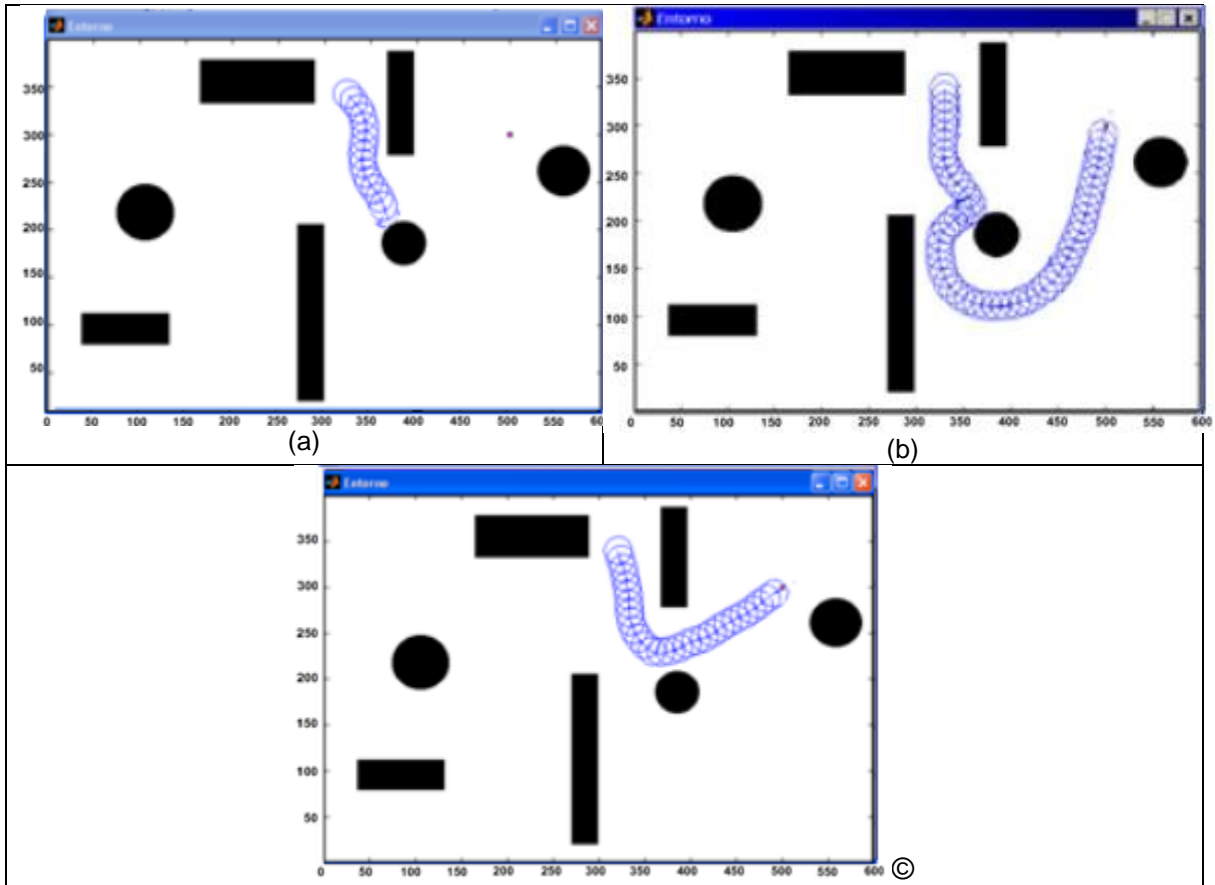


Tabla 4. 7: Datos me Métricas simulación 4

TECNICA ITEM TAREA	Sistema hibrido (RNA- comportamien tos reactivos)	Sistema hibrido (RNA- Difuso- comportamien tos reactivos)	Sistema hibrido (HMM- comportamien tos reactivos)
Punto de salida en X (cm)	320	320	320
Punto de salida en Y (cm)	340	340	340
Angulo inicial en grados	0	0	0
Punto de llegada en X (cm)	--	499.53	498.27
Punto de llegada en Y (cm)	--	297.12	298.42
Angulo de llegada en grados	--	95	75

Tabla 4. 8: Datos de inicio y llegada simulación 4

METRICA \ TECNICA	Sistema híbrido (RNA-comportamientos reactivos)	Sistema híbrido (RNA-Difuso-comportamientos reactivos)	Sistema híbrido (HMM-comportamientos reactivos)
Distancia media a los obstáculos SM1 (cm)	---	43.7895	40.21
Distancia media mínima a los obstáculos SM2 (cm)	---	9.2500	7.34
Distancia mínima de cualquier sensor MIN (cm)	---	3.0000	7
Distancia total recorrida PL (cm)	---	202	54
Distancia media al objetivo Mgd (cm)	---	0.5687	2
Periodos de Control LeN	---	114.3036	40

Es de anotar que en las simulaciones no en todas las pruebas los puntos de llegada son los mismos, dado que en varios de los ensayos se le incluyen funciones de ruido para simular posibles perturbaciones que permitiesen representar elementos de lo que sucede en el mundo real tales como ruido en la toma de datos por parte de los sensores, vibraciones en la parte de motores o llantas, entre otros.

5. CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

En este trabajo se desarrollaron algunas estrategias de navegación autónoma en ambientes desconocidos, entornos que son estructurados y con obstáculos estáticos, con el objetivo de que el robot posea una autonomía tal que este pudiese moverse a través de hitos como pasillo y puertas. El robot debía tomar la decisión sobre qué zonas podía explorar buscando equilibrio entre la información necesaria para auto localizarse y la cantidad de información nueva que adquiriría del entorno, esto con el fin de poder evaluar la distancia recorrida y lo faltante para llegar al objetivo, tratando que la trayectoria recorrida fuera óptima.

A través del uso de un conjunto de métricas de desempeño se consigue valorar y cotejar las características de navegación que originan el empleo de diferentes algoritmos de control, teniendo en cuenta que el uso de una sola métrica no brinda suficientemente información sobre la calidad de la trayectoria generada por él, la seguridad de la plataforma, el consumo de energía y el cumplimiento de la tarea asignada.

En este trabajo se presentó un análisis cuantitativo de 3 (tres) algoritmos que se usan para controlar la navegación de un robot móvil, se comparó el desempeño de los algoritmos en diferentes situaciones en donde se trabajaban algunos hitos tales como puertas y pasillos, donde los índices de calidad mostraron cada una de las cualidades que presentaban ante los retos propuestos.

Se pudo visualizar que si un sistema no se entrena apropiadamente, ya sea red neuronal, sistema estadístico y/o estocástico, puede que el resultado esperado para el cumplimiento de una tarea asignada no sea el esperado.

Es interesante que en robótica móvil, cuando se trabaja con comportamientos simples, como ir hacia la meta, repulsión de objetos, entre otros, es posible obtener comportamientos emergentes más complejos todo esto siempre y cuando se cuente con un elemento deliberativo que los coordine adecuadamente, además se tiene la

posibilidad que en un futuro cercano se incluyan otros de estos, permitiendo aun mejorar las posibles aplicaciones.

los métodos que están basados en inteligencia artificial (entre ellos el neuro-difuso) exhiben un buen nivel adaptativo excepcional, logrando cumplir las misiones que se han asignado en variados ambientes, teniendo en cuenta aspectos como las restricciones que sus sensores, la estructura de la plataforma robótica. etc., es de anotar que las rutas obtenidas no siempre son las más óptimas, pero se pueden obtener soluciones seguras

5.2 RECOMENDACIONES

Dentro de los elementos que se podrían analizar para un trabajo futuro, se podrían tener en cuenta:

- El desarrollo de simulaciones con mayor alcance en varios hitos o discontinuidades en entornos de Interiores
- Simulación de robots con otros sistemas de locomoción diferentes al actual.
- Inclusión de nuevas métricas, que permitan evaluar aspectos complementarios a los actuales.
- Realizar experimentos en un entorno real, en donde se pueden llegar a implementar los algoritmos en sistemas de desarrollo que tengan una mayor cantidad de procesamiento y de memoria como la tarjeta Raspberry-PI, PCs industriales, entre otros.
- En algún momento para el método estadístico se trabajó con Bayes, pero la convergencia que se obtenía cuando el robot se aproximaba a la meta era muy poca, por eso se omitió el trabajar con esta técnica.

ANEXO A: DATOS ENTRENAMIENTO

A continuación se muestran los 178 datos que se usaron para entrenar tanto la red neuronal como el sistema HMM, los datos no necesariamente están en orden, algunos de ellos se adjuntaban de acuerdo a como se iba observando el entrenamiento. Es de recordar que inicialmente los datos que se extrajeron pasaban los 320, pero después del análisis estadístico para mirar la relevancia quedaron los anteriormente dichos

Los datos acá mostrados se hacen con 2 cifras significativas, pero en el simulador y en el robot se trabajan con un número mayor de cifras, con el fin de que las operaciones y los resultados sean más exactos

Es de anotar que los datos para la orientación se manejan de cero a 180 grados

Tabla A1 1: Datos para entrenamiento

Angulo actual	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
29,05	50,00	50,00	27,00	19,00	15,00	19,00	47,00	39,00
30,68	50,00	50,00	47,00	35,00	35,00	50,00	47,00	39,00
30,62	50,00	50,00	19,00	15,00	11,00	15,00	19,00	39,00
32,71	50,00	50,00	47,00	31,00	31,00	31,00	43,00	50,00
32,28	50,00	50,00	50,00	15,00	15,00	15,00	23,00	43,00
35,32	43,00	50,00	50,00	50,00	30,11	23,00	19,00	21,00
40,43	50,00	50,00	50,00	50,00	30,11	23,00	19,00	21,00
34,29	50,00	50,00	47,00	50,00	50,00	50,00	50,00	50,00
24,29	50,00	50,00	50,00	50,00	31,00	35,00	43,00	50,00
12,84	50,00	50,00	43,00	47,00	50,00	50,00	50,00	50,00
0,19	50,00	50,00	31,00	23,00	19,00	23,00	50,00	50,00
7,04	50,00	50,00	50,00	27,00	23,00	39,00	50,00	50,00
14,04	50,00	15,00	11,00	15,00	50,00	50,00	50,00	50,00
22,11	35,00	50,00	50,00	50,00	47,00	39,00	50,00	50,00
29,49	50,00	50,00	11,00	15,00	19,00	50,00	50,00	50,00
15,08	50,00	50,00	50,00	50,00	50,00	50,00	23,00	23,00
25,57	50,00	50,00	50,00	50,00	50,00	50,00	31,00	19,00
-10,34	50,00	50,00	50,00	39,00	37,17	39,00	50,00	50,00
-24,33	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00
-42,82	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00
-22,19	50,00	50,00	50,00	50,00	50,00	19,00	19,00	50,00
86,13	50,00	50,00	43,00	50,00	50,00	50,00	31,00	23,00

76,97	50,00	50,00	50,00	50,00	50,00	47,00	50,00	50,00
56,15	50,00	47,00	43,00	47,00	50,00	50,00	50,00	50,00
70,23	50,00	50,00	50,00	50,00	50,00	47,00	27,00	50,00
-75,12	50,00	50,00	50,00	50,00	43,00	35,00	50,00	50,00
-72,76	50,00	50,00	19,00	11,00	15,00	50,00	50,00	50,00
-18,91	39,00	31,00	32,92	41,44	47,00	39,00	39,00	50,00
-23,37	50,00	50,00	50,00	50,00	47,00	39,00	39,00	50,00
-28,42	50,00	50,00	50,00	50,00	47,00	39,00	39,00	50,00
-34,43	50,00	50,00	50,00	50,00	47,00	39,00	39,00	50,00
-41,13	50,00	50,00	50,00	50,00	47,00	39,00	50,00	50,00
-40,13	50,00	50,00	50,00	43,00	31,00	23,00	50,00	50,00
-33,10	50,00	50,00	50,00	47,00	31,00	27,00	27,00	35,00
-26,66	50,00	50,00	50,00	47,00	31,00	27,00	27,00	35,00
-20,81	50,00	50,00	50,00	47,00	31,00	27,00	27,00	35,00
-16,02	23,00	19,00	19,00	26,10	43,00	27,00	27,00	35,00
-14,38	23,00	19,00	19,00	26,10	43,00	11,00	11,00	15,00
-18,81	50,00	50,00	47,00	23,00	15,00	11,00	11,00	15,00
-24,09	50,00	50,00	50,00	23,00	15,00	11,00	11,00	15,00
-29,86	50,00	50,00	50,00	23,00	15,00	11,00	11,00	15,00
-36,80	50,00	50,00	50,00	23,00	19,00	11,00	11,00	15,00
-39,77	50,00	50,00	50,00	27,00	19,00	11,00	50,00	50,00
-65,11	50,00	47,00	19,00	15,00	11,00	11,00	50,00	50,00
-57,84	50,00	47,00	19,00	15,00	11,00	11,00	19,00	39,00
-51,80	50,00	47,00	19,00	15,00	11,00	11,00	19,00	39,00
-46,10	50,00	50,00	19,00	15,00	11,00	11,00	23,00	39,00
-41,68	11,00	11,00	18,84	38,81	11,00	11,00	19,00	39,00
-43,42	11,00	13,48	19,00	39,00	27,00	27,00	45,00	50,00
-48,09	50,00	50,00	43,00	31,00	27,00	27,00	39,00	50,00
-53,36	50,00	50,00	43,00	31,00	27,00	27,00	35,00	50,00
-60,07	50,00	50,00	43,00	31,00	27,00	27,00	30,00	50,00
-66,83	50,00	50,00	43,00	31,00	27,00	27,00	50,00	50,00
-67,82	50,00	50,00	50,00	43,00	39,00	50,00	50,00	50,00
-60,79	50,00	50,00	50,00	43,00	39,00	43,00	50,00	50,00
-54,81	50,00	50,00	50,00	47,00	43,00	43,00	50,00	50,00
-49,49	50,00	50,00	50,00	47,00	43,00	43,00	50,00	50,00
-44,97	11,00	13,48	19,00	39,00	43,00	43,00	50,00	50,00
-74,52	14,76	23,00	50,00	43,00	47,00	50,00	50,00	50,00
-79,25	50,00	50,00	43,00	43,00	47,00	50,00	50,00	50,00
-84,26	50,00	50,00	47,00	43,00	47,00	50,00	50,00	50,00
-90,12	50,00	50,00	47,00	43,00	47,00	50,00	50,00	50,00
-95,06	50,00	39,00	31,00	31,00	35,00	50,00	50,00	50,00

-87,90	50,00	39,00	31,00	31,00	35,00	47,00	50,00	50,00
-82,33	50,00	39,00	31,00	31,00	35,00	47,00	50,00	50,00
-77,20	50,00	39,00	31,00	31,00	35,00	47,00	50,00	50,00
-73,18	11,00	22,09	47,00	31,00	35,00	47,00	50,00	50,00
-71,57	11,00	22,09	47,00	15,00	19,00	27,00	50,00	50,00
-76,31	50,00	23,00	19,00	15,00	19,00	31,00	50,00	50,00
-81,31	47,00	23,00	19,00	15,00	19,00	31,00	50,00	50,00
-87,38	47,00	23,00	19,00	15,00	19,00	31,00	50,00	50,00
-94,03	47,00	27,00	19,00	19,00	19,00	50,00	50,00	50,00
45,44	50,00	50,00	35,00	27,00	23,00	26,09	35,00	15,00
40,79	50,00	50,00	50,00	50,00	50,00	47,00	19,00	15,00
35,62	50,00	50,00	50,00	50,00	50,00	43,00	19,00	15,00
28,96	50,00	50,00	50,00	50,00	50,00	39,00	19,00	11,00
21,76	50,00	50,00	50,00	50,00	50,00	39,00	19,00	11,00
20,65	50,00	50,00	50,00	50,00	50,00	50,00	35,00	27,00
27,83	50,00	50,00	50,00	50,00	50,00	50,00	39,00	27,00
34,37	50,00	50,00	50,00	50,00	50,00	50,00	39,00	27,00
40,18	50,00	50,00	50,00	50,00	50,00	50,00	39,00	31,00
45,11	50,00	46,53	23,00	15,00	13,15	14,67	22,33	43,00
43,56	50,00	46,53	23,00	15,00	13,15	14,67	22,33	43,00
39,17	50,00	50,00	50,00	50,00	50,00	50,00	50,00	47,00
34,42	50,00	50,00	50,00	50,00	50,00	50,00	50,00	47,00
27,99	50,00	50,00	50,00	50,00	50,00	50,00	50,00	47,00
20,93	50,00	50,00	50,00	50,00	50,00	50,00	50,00	47,00
-6,83	50,00	50,00	50,00	50,00	50,00	50,00	43,00	43,00
0,31	50,00	50,00	50,00	50,00	50,00	50,00	43,00	43,00
6,25	50,00	50,00	50,00	50,00	50,00	50,00	43,00	43,00
11,51	50,00	50,00	50,00	50,00	50,00	50,00	43,00	43,00
16,11	31,00	19,00	11,00	10,07	11,00	19,00	47,00	43,00
17,77	31,00	19,00	11,00	10,07	11,00	19,00	47,00	27,00
12,91	50,00	50,00	50,00	50,00	50,00	35,00	27,00	27,00
7,09	50,00	50,00	50,00	50,00	50,00	35,00	27,00	27,00
1,13	50,00	50,00	50,00	50,00	50,00	35,00	27,00	27,00
-5,41	50,00	50,00	50,00	50,00	50,00	35,00	27,00	27,00
-4,05	50,00	50,00	50,00	50,00	31,00	19,00	15,00	11,00
3,37	50,00	50,00	50,00	50,00	31,00	19,00	15,00	11,00
9,65	50,00	50,00	50,00	50,00	31,00	19,00	15,00	11,00
14,90	50,00	50,00	50,00	50,00	31,00	19,00	15,00	11,00
19,19	31,00	18,14	13,25	11,00	13,75	22,65	50,00	11,00
67,75	50,00	35,00	19,00	11,00	7,00	11,00	15,00	26,79
73,03	50,00	35,00	19,00	11,00	7,00	11,00	15,00	43,00

78,62	50,00	35,00	19,00	11,00	7,00	11,00	15,00	35,00
84,44	50,00	50,00	19,00	11,00	7,00	11,00	15,00	35,00
89,63	50,00	50,00	50,00	11,00	7,00	7,00	15,00	31,00
95,19	50,00	50,00	50,00	50,00	7,00	11,00	15,00	35,00
94,42	50,00	50,00	50,00	50,00	23,00	23,00	35,00	50,00
87,95	50,00	50,00	50,00	27,00	23,00	23,00	35,00	50,00
83,21	50,00	50,00	35,00	27,00	23,00	23,00	35,00	50,00
78,89	50,00	50,00	35,00	27,00	23,00	23,00	35,00	50,00
73,65	50,00	50,00	31,00	23,00	23,00	23,00	31,00	39,00
74,72	50,00	50,00	50,00	39,00	39,00	39,00	50,00	39,00
80,08	50,00	50,00	50,00	39,00	39,00	39,00	50,00	50,00
85,63	50,00	50,00	50,00	39,00	39,00	39,00	50,00	50,00
90,63	50,00	50,00	50,00	39,00	39,00	39,00	50,00	50,00
62,87	50,00	50,00	39,00	39,00	43,00	50,00	50,00	50,00
57,56	50,00	47,00	39,00	39,00	43,00	50,00	50,00	50,00
52,98	50,00	50,00	39,00	39,00	43,00	50,00	50,00	50,00
47,91	50,00	50,00	39,00	39,00	43,00	50,00	39,00	38,09
46,70	50,00	31,00	23,00	23,00	27,00	39,00	39,00	38,09
51,19	50,00	31,00	23,00	23,00	27,00	39,00	50,00	50,00
56,16	50,00	31,00	23,00	23,00	27,00	39,00	50,00	50,00
60,89	50,00	50,00	23,00	23,00	27,00	39,00	50,00	50,00
65,63	50,00	50,00	23,00	23,00	27,00	39,00	50,00	50,00
65,93	50,00	50,00	11,00	11,00	11,00	19,00	47,00	50,00
60,39	50,00	19,00	11,00	11,00	11,00	19,00	47,00	50,00
54,89	31,00	19,00	11,00	11,00	11,00	19,00	47,00	50,00
49,52	31,00	19,00	11,00	11,00	11,00	19,00	50,00	50,00
44,03	31,00	19,00	11,00	11,00	11,00	19,00	35,00	34,09
118,87	50,00	50,00	50,00	50,00	11,00	7,00	11,00	15,00
112,93	50,00	50,00	50,00	19,00	11,00	7,00	11,00	15,00
107,81	50,00	50,00	50,00	19,00	11,00	7,00	11,00	15,00
100,79	50,00	50,00	47,00	19,00	11,00	7,00	11,00	15,00
102,53	50,00	50,00	50,00	39,00	27,00	23,00	23,00	31,00
106,71	50,00	50,00	50,00	39,00	27,00	23,00	23,00	31,00
111,71	50,00	50,00	50,00	50,00	31,00	23,00	27,00	31,00
116,75	50,00	50,00	50,00	50,00	31,00	23,00	27,00	31,00
113,12	50,00	50,00	50,00	50,00	47,00	39,00	39,00	50,00
108,90	50,00	50,00	50,00	50,00	47,00	39,00	43,00	50,00
104,09	50,00	50,00	50,00	50,00	47,00	39,00	39,00	50,00
23,59	47,00	43,00	43,00	50,00	50,00	50,00	50,00	23,00
28,49	50,00	39,00	43,00	50,00	50,00	50,00	50,00	31,00
33,40	50,00	39,00	43,00	50,00	50,00	50,00	50,00	47,00

38,00	50,00	50,00	35,00	43,00	50,00	50,00	50,00	50,00
33,43	50,00	27,00	31,00	39,00	50,00	50,00	50,00	47,00
26,52	31,00	27,00	27,00	35,00	50,00	50,00	50,00	43,00
26,73	19,00	11,00	15,00	19,00	35,00	50,00	50,00	50,00
31,44	50,00	11,00	15,00	19,00	35,00	50,00	50,00	50,00
38,64	50,00	50,00	19,00	19,00	31,00	50,00	50,00	50,00
10,47	50,00	15,00	19,00	39,00	50,00	50,00	50,00	50,00
5,13	11,00	11,00	19,00	39,00	50,00	50,00	50,00	50,00
-1,10	11,00	11,00	19,00	39,00	50,00	50,00	50,00	50,00
-6,22	11,00	11,00	19,00	39,00	50,00	50,00	50,00	47,00
-5,64	27,00	27,00	39,00	50,00	50,00	50,00	39,00	31,00
-1,33	27,00	27,00	39,00	50,00	50,00	50,00	39,00	47,00
3,33	27,00	27,00	39,00	50,00	50,00	50,00	47,00	50,00
9,46	50,00	31,00	39,00	50,00	50,00	50,00	50,00	50,00
9,07	50,00	47,00	50,00	50,00	50,00	50,00	50,00	50,00
4,56	43,00	47,00	50,00	50,00	50,00	43,00	47,00	50,00
0,00	43,00	47,00	50,00	50,00	50,00	43,00	31,00	50,00
-4,56	43,00	47,00	50,00	50,00	50,00	43,00	19,00	27,00
83,40	27,00	50,00	50,00	50,00	50,00	27,00	19,00	50,00
81,74	31,00	39,00	50,00	50,00	50,00	27,00	31,00	50,00
80,36	43,00	43,00	50,00	50,00	47,00	35,00	50,00	50,00
84,94	35,00	31,00	43,00	50,00	50,00	47,00	50,00	50,00
86,84	23,00	19,00	50,00	50,00	50,00	47,00	39,00	50,00
89,06	11,00	19,00	50,00	50,00	50,00	50,00	35,00	50,00
117,24	19,00	11,00	15,00	50,00	50,00	50,00	50,00	39,00
115,80	50,00	23,00	19,00	27,00	50,00	50,00	50,00	43,00
113,45	50,00	50,00	31,00	31,00	50,00	50,00	50,00	50,00
109,26	50,00	47,00	39,00	50,00	50,00	50,00	39,00	50,00
110,89	50,00	31,00	31,00	50,00	50,00	50,00	35,00	31,00
112,95	31,00	23,00	31,00	50,00	50,00	50,00	35,00	27,00
107,94	47,00	35,00	50,00	50,00	50,00	50,00	19,00	15,00
106,73	50,00	43,00	43,00	50,00	50,00	50,00	19,00	31,00
105,31	50,00	50,00	50,00	50,00	50,00	47,00	35,00	50,00

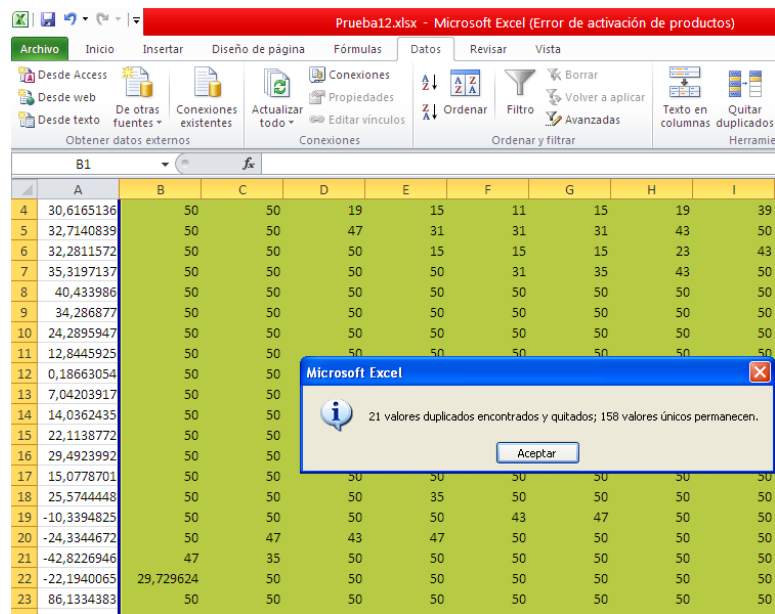
ANEXO B: REDUCCIÓN DE CARACTERÍSTICAS

El procedimiento que a continuación se muestra corresponde a la simplificación que se realizó para elaborar el módulo fuzzy

Para la simplificación de las últimas 178 características obtenidas, después de obtener el análisis estadístico. Se llevan los datos a EXCEL, allí se omite la variable ángulo, dado que de este ítem se encarga la Red Neuronal. En el primer renglón de la tabla están los nombres de las señales, por eso este renglón se obvia.

Para proseguir con la simplificación de datos, se tienen en cuenta solo los datos obtenidos por los sensores de la plataforma, en Excel se aplica la opción “Quitar duplicados”

Fig. A.2. 1: Simplificación 2 de datos



Es de anotar que en la figura x se muestran 158 valores porque tiene en cuenta el primer renglón que tiene el nombre de las señales

Como se muestra en la Fig. A.2.1. los datos que quedan para la diseñar el módulo Fuzzy se reduce a 157, estos se trataran de minimizar para lograr que el cómputo que tenga que elaborar el procesador sea menor y cumpla de manera óptima la tarea propuesta

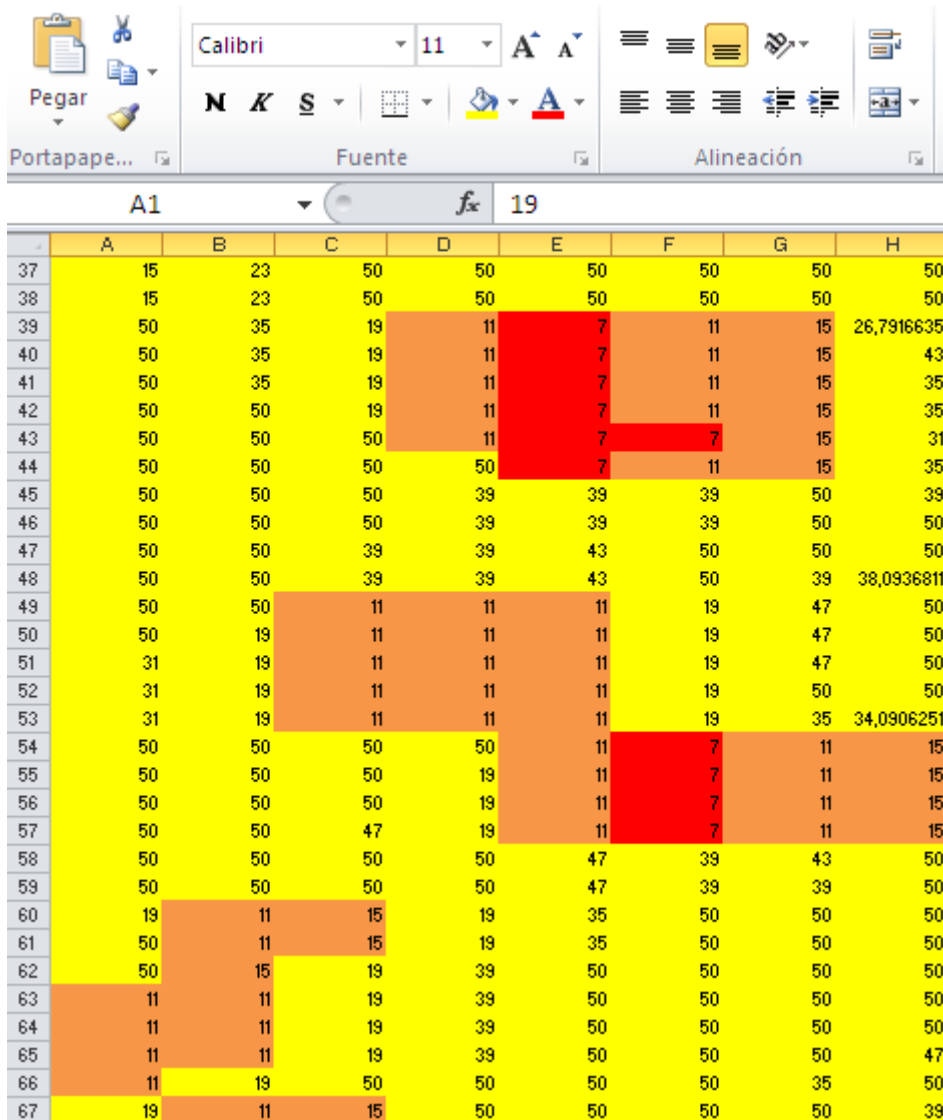
Luego se separaron por colores cada uno de los datos, teniendo en cuenta las variables lingüísticas, teniendo en cuenta lo mostrado en Tabla A.2.1

Tabla A.2. 1 Variables Lingüísticas

Variable Lingüística	Rango (cm)	Color asignado
Choque	≤ 4	Morado
Muy Cerca	$4.01 \leq \text{Obstáculo} \leq 8$	Rojo
Cerca	$8.01 \leq \text{Obstáculo} \leq 17$	Naranja
Alejados o No presentes	≥ 17	Amarillo

Una vez aplicado este concepto, los resultados que se obtienen, se ven en la fig X2

Fig. A.2. 2: Simplificación por Rango en las Variables Lingüísticas



Y para acabar de terminar de simplificar, se observó, que para valores superiores a 19 cm medidos por la totalidad los sensores, la velocidad tomada por el robot es la misma que cuando toma el valor de 19 cm, así que para valores donde todos los sensores de distancia midieran 19 a más, estos se acotaron a 19, aplicando estos 2 condicionantes se logró que la cantidad de reglas que se debiesen implementar en la Red fuesen 68, como lo se ve en la Fig., A.2.3

Fig. A.2. 3: Datos mayores a 19 Normalizados

	A	B	C	D	E	F	G	H
48	19	19	19	19	19	19	19	19
49	19	19	11	11	11	19	19	19
50	19	19	11	11	11	19	19	19
51	19	19	11	11	11	19	19	19
52	19	19	11	11	11	19	19	19
53	19	19	11	11	11	19	19	19
54	19	19	19	19	11	7	11	15
55	19	19	19	19	11	7	11	15
56	19	19	19	19	11	7	11	15
57	19	19	19	19	11	7	11	15
58	19	19	19	19	19	19	19	19
59	19	19	19	19	19	19	19	19
60	19	11	15	19	19	19	19	19
61	19	11	15	19	19	19	19	19
62	19	15	19	19	19	19	19	19
63	11	11	19	19	19	19	19	19
64	11	11	19	19	19	19	19	19
65	11	11	19	19	19	19	19	19
66	11	19	19	19	19	19	19	19
67	19	11	15	19	19	19	19	19
68	19	19	19	19	19	19	19	15
69								
70								

Luego, nuevamente se aplicó la herramienta de Excel “Quitar duplicados”, donde la cantidad de características o reglas para el sistema Fuzzy se redujo a 50, tal cual se ve en la Fig. A.2.4

Fig. A.2. 4: Simplificación 3

	A	B	C	D	E	F	G	H
31	19	19	19	11	7	11	15	19
32	19	19	19	11	7	11	15	19
33	19	19	19	11	7	11	15	19
34	19	19	19	11	7	7	15	19
35	19	19	19	19	7	11	15	19
36	19	19	19	19	19	19	19	19
37	19	19	19	19	19	19	19	19
38	19	19	19	19	19	19	19	19
39	19	19	19	19	19	19	19	19
40	19	19	11	11	11	19	19	19
41	19	19	11	11	11	19	19	19
42	19	19	19	19	11	7	11	15
43	19	19	19	19	19	19	19	19
44	19	11	15	19	19	19	19	19
45	19	15	19	19	19	19	19	19
46	11	11	19	19	19	19	19	19
47	11	11	19	19	19	19	19	19
48	11	19	19	19	19	19	19	19
49	19	11	15	19	19	19	19	19
50	19	19	19	19	19	19	19	15
51								

Para simplificar más el número de reglas, se aplicó la herramienta del vecino más cercano, donde valores que están muy cercanos entre si y que se encuentran dentro del mismo intervalo de las variables lingüísticas, permiten tener un único valor, se observa un ejemplo de estos valores en la Fig., A.2.5

Fig. A.2. 5: Valores Cercanos dentro de un mismo

11	11	18,8381863	19	11	11	19
11	13,4811779	19	19	19	19	19
14,7609184	19	19	19	19	19	19
11	19	19	19	19	19	19
11	19	19	15	19	19	19
19	19	19	15	19	19	19
19	19	19	19	19	19	19
19	19	19	19	19	47	19
19	19	19	19	19	43	19
19	19	19	19	19	39	19
19	19	19	19	19	39	19
19	19	19	15	13,1466096	14,6619212	19
19	19	11	10,072323	11	19	19
19	19	19	19	19	19	15
19	19	19	19	19	19	15
19	18,1408652	13,2549899	11	13,7452886	19	19
19	23	19	19	19	19	19
19	39	19	19	19	19	19
19	19	19	19	19	19	19

Se volvió a aplicar la herramienta “Quitar valores repetidos” de Excel, y el número total de reglas se redujo a 35 aplicadas en el Sistema Difuso

Por último se separaron por nombre de acuerdo a los rangos interpuestos por las variables lingüísticas usadas en el sistema difuso

BIBLIOGRAFÍA

- Acosta, A. y. (2011). *Análisis de un esquema de control de comportamientos basado en subespacios nulos para robots móviles*. Revista Avances en Sistemas e Informática: ISSN 1657-7663, vol. 8, págs. 165-172.
- Aguero, C. E. (2009). *Estimación de objetos con fusión bayesiana en equipos de robots móviles*. Universidad Rey Juan Carlos.
- Álvarez, J. (1998). *Planificación del movimiento de vehículos autónomos basada en sensores*. Tesis doctoral, . Oviedo, España: Universidad de Oviedo, 178 pp.,
- Angulo Usategui, J. M. (2000). *Microbótica*. 1a. Edición, Paraninfo.
- B. Quintana, S. A. (2016). Door Detection in 3D Colored Laser Scans for Autonomous Indoor Navigation. *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 4-7.
- Babel, M. a. (2016). A visual servoing approach for autonomous corridor following and doorway passing a wheelchair. *Robotics and Autonomous Systems*, págs. 28-40.
- Babel, M. y. (2014). *A visual servoing approach for autonomous corridor following and doorway passing a wheelchair*. Robotics and autonomous systems.
- Barragá Piña, A. (2009). *Síntesis de sistemas de control borroso estables por diseño*. Tesis Doctoral. Universidad de Huelva, España.
- Bartle, C. (14 de 05 de 2009). *Roomba Long Exposure*. Recuperado el 15 de 04 de 2019, de Roomba Long Exposure: <https://www.flickr.com/photos/13963375@N00/3533146556>
- Benporad, R. D. (1997). *Wall-following controllers for sonar-based mobile robots*. in: Proceedings of the 36th IEEE Conference on decision and control, San Diego: Diciembre.
- Bermúdez, G. R. (2012). *Modelo Cinemático de un robot móvil implementado con LEGO NXT para un sistema de localización indoor diseñado en Labview*. Universidad Distrital Francisco José de Caldas-Tecnura. Vol. 16. Edición Especial. págs. 23 - 34.
- Boresntein, J. (1996). *Gyrodometry: A new method for combininig data from gyros and odometry in mobile robots*. IEEE international conference on robotics and automation: Minneapolis.
- Brooks, Rodney. (1987). *Intelligence without representation*. Preprints of the Workshop in Foundations of Artificial Intelligence. Endicott House. Dedham. MA.

- Brooks, Rodney. (1997). *From earwings to humans. Robotic and autonomous systems*. Vol 20. No 2. Pags 291-304.
- Calderón García, J. E. (2016). *Evaluación de los métodos Umbmark y triangular path para La estimación y corrección de errores sistemáticos en odometría*. Fundación Universitaria Los Libertadores: Bogota, Colombia.
- Calisi, D. (2009). *Mobile robots and vehicles motion systems: a unifying framework*. Tesis Doctoral. Italia: Universidad de Roma.
- Camargo, R. a. (2008). *Hardware and software architecture for execution control of an autonomous mobile robot*. Industrial Electronics, control and Automation 1992 Power Electronics and Motion Control: Vol. 2, págs. 818-825.
- Carelli, R. a. (2003). *Corridor navigation and wall-following stable control for sonar-based mobile robots*. Robotics and Autonomous Systems 45: Págs. 235–247.
- Carelli, R. y. (1999). *Algorithms for stable control of mobile robots with obstacle avoidance*. Latin American Applied Research 29. Págs. 191-196.
- Casarrubias, O. F. (2018). *Localización de un robot móvil utilizando modelos ocultos de Markov*. Tesis de maestría. Universidad Nacional Autónoma de México: Facultad de Ingeniería.
- Cerrada, C. C. (2013). *Generación de comportamiento autónomo para un robot de inspección de cultivos*. Centro de Automática y Robotica (CSIC-UPM). Universidad Nacional de Educación a Distancia (UNED). ISBN: 978-84-695-8175-9: Madrid, España.
- Cerrada, C. C. (2013). *Generación de comportamiento autónomo para un robot de inspección de cultivos*. Centro de automática y robótica (CSIC-UPM). Universidad Nacional de educación a distancia (UNED): ISBN: 978-84-695-8175-9. Madrid, España.
- Cheng, Y. M. (2006). *Visual odometry on the Mars exploration rovers*. IEEE Robotics and Automation magazine: Vol. 8, págs. 54-62.
- Cordero, M. &. (2011). *Control de un robot móvil de ruedas mediante campos potenciales artificiales y procesamiento digital de imágenes en la evasión de obstáculos*. Instituto Politécnico Nacional-México.: Tesis de Maestría.
- Correa, F. (2013). *Control neuronal–difuso de navegación reactiva continua incluyendo la evasión de obstáculos presentes y emergentes*. Tesis de Maestría. Antofagasta, Chile: Universidad Católica del Norte.

- De Lope Asiaín, J. (2001). *Robots móviles. Evolución histórica y técnicas de programación*.
- Demeester, E. V. (2012). *Wheelchair navigation assistance in the fp7 project radhar: Objectives and current state*. IROS Workshop on progress, challenges and future perspectives in navigation and manipulation assistance for robotic wheelchairs.
- Dempster, A. P. (1977). *Maximum likelihood from incomplete data via the em algorithm*. Journal of the royal statistical society, Series B, 39(1): pags 1–38.
- Dev, A. K. (1997). *Navigation of a mobile robot on the temporal development of the optic flow*. in: Proceedings of the IEEE/RSJ/GI International conference on intelligent robots and systems IROS'97: Grenoble, pags. 558–563.
- Eduardo, C. R. (2003). Corridor navigation and wall-following stable control for sonar-based mobile robots. Robotics and Autonomous Systems. *Robotics and Autonomous Systems Vol 45*, 235–247.
- Elkan, C. (2003). *Using the triangle inequality to accelerate k-means*. in Proc. ICML.
- Fernández León, J. A. (2005). *Estudio experimental sobre comportamientos reactivos-evolutivos en navegación de robots móviles*. Facultad de Ingeniería: Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina.
- Fernandez, C. (2002). *Accumulation of adaptations in plastic neural networks*. MSc. Dissertation. COGS, University of Sussex. 2002.
- Figueroa Angulo, J. I. (2015). *Behaviour recognition system based on hidden Markov models*. Tesis Doctoral Universidad Nacional Autónoma de México. Ciudad de México.
- Floreano, D. y Urzelai, J. (1999). *Evolution of adaptive-synapse controllers*. . Proceedings of the 5th european conference on artificial life, Berlin: Springer Verlag. (ECAL'1999).
- Floreano, D. y. (2001). *Neural morphogenesis, synaptic plasticity, and evolution. Theory in biosciences*. 120 (3-4), pags. 223-238.
- Flórez, R. y. (2008). *Las redes neuronales artificiales*. Netbiblio. Universidad de León. México.
- Garre del Olmo, C. (2006). *Niveles reactivo y estructural para la arquitectura de control del robot sancho iii*. Tesis de Maestría. Universidad Politécnica de Madrid: España.

- Gómez, J. C. (2012). *Estudio e implementación de un sistema de auto-localización para la navegación de un robot en entornos irregulares*. Tesis de Maestría. Universidad de Antioquia: Medellín, Colombia.
- Gonzalez Acevedo, H. &. (2007). *Estudio comparativo de tres técnicas de navegación para robots móviles*. Bucaramanga, Colombia.
- González, J. y. (2015). Estimación de la posición de un robot móvil. <https://www.researchgate.net/publication/267222718>, pags. 1-25.
- Guzmán, J. A. (2012). Búsqueda de la ruta óptima mediante los algoritmos: genético y Dijkstra utilizando mapas de visibilidad. *Scientia et Technica Año XVII, No 51, Universidad Tecnológica de Pereira. ISSN 0122-1701*, pags. 107-112.
- Hamerly, G. &. (2015). *Accelerating Lloyd's algorithm for k-Means clustering*. Cham: Springer international publishing: pags. 41–78.
- Hebb, D. O. (1958). *A Textbook of psychology*. Philadelphia, PA., W. B. Saunders: págs. 44-45.
- Herrero Reder, I. (2015). *Arquitectura de comportamientos reactivos para agentes robóticos basada en CBR*. Malaga: Tesis Doctoral Universidad de Malaga.
- Hortal, A. R. (2017). Hacia un formalismo basado en sistemas multiagente para evaluar modelos de organización humanos. Caso de estudio (I): Holacracy. *XXIII Congreso argentino de ciencias de la Computación*, pags. 73-82.
- Jiménez, A. R. (2015). *Localización precisa usando radio UWB y Dead-Reckoning para cooperación eficiente humano-robot*. Actas de las XXXVI jornadas de Automática. ISBN 978-84-15914-12-9. Págs. 284-281: Bilbao España.
- Jonathan, C. (2017). *Arquitectura y diseño de un sistema completo de navegación semántica. Descripción de su ontología y gestión de conocimiento*. Madrid. España: Tesis Doctoral. Universidad Carlos III de Madrid.
- Kanungo, T. (1999). *Extended finite state models of language, chapter UMDHMM: Hidden Markov model Toolkit*. Cambridge University Press.
- Kokosy, A. F. (2012). *SYSIASS an intelligent powered wheelchair*. International conference on systems and computer science.
- Lamon, P. a. (2003). *3D odometry for rough terrain*. IEEE international conference on robotics and automation: Taipei.
- Levinson, S. E. (2008). *An introduction to the application of the theory of probabilistic functions of a markov process to automatic*.

- Lloys, S. (2006). *Least squares quantization in pcm*. IEEE Trans. Inf. Theor., vol. 28, pags. 129– 137.
- Londoño, N. (2009). Metodologías de desarrollo de software, un enfoque a Robots Móviles. *Revista Politécnica ISSN 1900-2351*, Número 8, pags. 74-83.
- Londoño, N. y. (2003). *Una concepción metodológica para el análisis y diseño de arquitecturas software para robots móviles*. Propuesta de investigación doctorado en ingeniería eléctrica y electrónica. Universidad del Valle.
- Luna, K. L. (2018). *Guía de personas con robots móviles basado en la interacción humano-robot*. San Luis Potosí: Tesis de doctorado. Universidad Autónoma de San Luis Potosí facultad de Ingeniería.
- Marino, A. a. (2009). *Behavioral control for multi-robot perimeter patrol: A finite state automata approach*. In: ICRA.
- Martín, J. (2017). *Metodos de Montecarlo aplicados a Inferencia*. Madrid: Universidad Carlos III.
- Mejía, J. C. (2002). *Arquitectura de control para la navegación de u robot movil simulado*. Medellin, Antioquia: Politecnico Jaime Isaza Cadavid.
- Miranda Cruz, L. (2011). *Humanoid robot NAO: Developing behaviours for soccer humanoid robots*. Master in Informatics and Computing Engineering: Facultad de ingenieria de Universidad de Porto.
- Mora, T. E. (1998). *Fuzzy logic-based real-time navigation controller for a mobile robot*. Proceedings of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems: Victoria, B.C., Canada.
- Moreo, L. M., D., G., & J., P. A. (1996). *Neural networks for mobile robot piloting control*. en: A. Zalzala y A. Morris, Neural Networks for Robotic Control, primera edición, Prentice Hall.
- MUÑOZ CEBALLOS, N. D. (2008). *Método de comparación de algoritmos de control de robots móviles*. Memorias Congreso CISCI: Miami-Estados Unidos.
- Muñoz, N. (2014). *Simulation and assessment educational framework for mobile robot algorithms*. Journal of Automation, Mobile Robotics & Intelligent Systems. Vol 8, N° 1.
- Muñoz, V. F. (2 de 1 de 2019). *Planificación de trayectorias para robots móviles*. Recuperado el 2 de 1 de 2019, de <http://webpersonal.uma.es/~VFMM/PDF/cap2.pdf>: <http://webpersonal.uma.es/~VFMM/PDF/cap2.pdf>

- Mutambara, A. (1998). *Decentralized estimation and control for multisensor systems*. CRC: Boca Raton.
- Navarro García, D. A. (2009). *Contribución a la autolocalización de robots móviles basada en la fusión de información multisensorial*. Tesis doctoral. Universidad Politécnica de Valencia: España.
- Navarro García, D. A. (2009). *Contribución a la autolocalización de robots móviles basada en la fusión de información multisensorial*. Tesis Doctoral. Universidad Politécnica de Valencia: DISCA.
- Navarro, D. A. (2009). *Contribución a la autolocalización de robots móviles basada en la fusión de información multisensorial*. Valencia, España: Tesis Doctoral. Universidad Politécnica de Valencia.
- Newling, J. &. (2016). *Fast k-means with accurate bounds*. in Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, pags 936–944.
- Nolfi, S. y. (1999). *Learning and evolution. Autonomous robots*. Vol 7: Pags 89-113.
- Noreils, F. &. (1989). *A general structure for mobile robot action control*. IEEE/RSJ International Workshop on Intelligent and Systems. Tsukuba Japan.
- Oliva, D. a. (2011). *Topological autonomous navigation for mobile robots in indoor environments using ANN and FSM*. Universidad de San Pablo: Brazil.
- Ollero, A. (1995). *Planificación de trayectorias para robots móviles*. Universidad de Málaga: España.
- Ortega, L. (2010). *El diagrama de Voronoi*. Universidad de Jaen: España.
- Pandey, A. y. (2017). Mobile Robot Navigation and obstacle avoidance. *International Robotics & Automation Journal*, Volume 2 Issue 3 - pags: 1-12.
- Parrón, F. G. (2009). *Sistema de localización para robots móviles basado en GPS. Aplicación a campos de golf*. CEA jornadas de Automática: Valladolid, España.
- Preciado, J. D. (2018). *Planificación de trayectorias de robots móviles de diferentes arquitecturas en entornos dinámicos*. Granada, España: Universidad militar Nueva Granada.
- Quintana, S. A. (2016). Door detection in 3D colored laser scans for autonomous indoor navigation. *International conference on indoor positioning and indoor navigation (IPIN)*, pags. 4-7.

- Rabiner, L. y. (1993). *Fundamentals of speech recognition*. United States ed edition. Prentice Hall.
- Reale, F. P. (2014). *Estudio del estado del arte en robots agrícolas*. Universidad de la Republica de Uruguay: Uruguay.
- Ridao, P. B. (1999). *Recent trends in control architectures for autonomous underwater vehicles*. International Journal of Systems Science. vol. 30. No 9. Pags 1033-1056.
- Ridao, P. C. (2001). *O2CA2: A new hibrid control architecture for a low cost AUV*. Control Application in Marine Systems CAMS'2001 Scotland.
- Rodríguez, E. (2010). *Diagramas de Voronoi*. Cinvestav: Tamaulipas, Mexico.
- Rodríguez, F. S. (2016). *Análisis de viabilidad de la utilización de robots en invernaderos*. Almería, España: Propuesta para el modelo Almería. II simposio nacional de ingeniería hortícola, Vol. 1.
- Rosenblatt, J. (1997). *DAMN: Distributed architecture for mobile navigation*. Ph.D. Thesis. Pittsburgh, PA: Carnegie Mellon University Robotics Institute.
- Russell, S. y. (1995). *Artificial intelligence: A modern approach*. Prentice-Hall, Inc.
- Salcedo, F. J. (2006). *Modelos ocultos de Markov*. España: Universidad de Granada.
- Sales, D. O. (2011). *Vision-based autonomous navigation system using ANN and FSM control*. In: Proceedings of the IEEE LARS/EnRI. San Bernardo de Campo, SP: IEEE S.
- Sarmiento Gutiérrez, C. A. (2017). *Sistema de localización en espacios interiores para un robot móvil utilizando nubes de puntos y modelos ocultos de Markov*. Tesis de maestría, Universidad nacional autónoma de México: programa de maestría y doctorado en ingeniería, ingeniería eléctrica – procesamiento digital de señales.
- Simo Ten, J. E. (1997). *Una arquitectura basada en motivaciones para el control de robots móviles*. Tesis doctoral Universidad Pontificia de Valencia.
- Tobías, M. y. (2015). *Construcción de un robot que preste servicios de gasfitería, en ductos planos, semi inclinados, y subterráneos*. Quevedo, Ecuador: Universidad Técnica Estatal De Quevedo.
- Togelius, J. (2003). *Evolution of the layers in a subsumption architecture robot controller*. Master of science in evolutionary and adaptive systems. University of Sussex, UK.
- Velásquez, C. A. (2015). *Implementación de sistema de navegación autónomo en robot móvil experimental para reconstrucción y exploración de entornos desconocidos*.

IV Congreso internacional de ingeniería mecatrónica y automatización - CIIMA 2015, 72-84.

Vikas Kumar, N. (2004). *Integration of Inertial Navigation System and Global Positioning System using Kalman Filtering*. Ph. D Thesis Indian Institute of Technology. Bombay.

Vishnu K, N. F. (2016). Vision-based adaptive assistance and haptic guidance for safe wheelchair corridor following. *Computer Vision and Image Understanding*, vol. 149, pags 171–185.

Vishnu, N. F. (2016). Vision-based adaptive assistance and haptic guidance for safe wheelchair corridor following. *Computer vision and image understanding*, vol. 149, pags. 171–185.

Yandún, A. &. (2012). *Planeación y seguimiento de trayectorias para un robot móvil*. Escuela Politécnica Nacional: Quito – Ecuador.

Yang, Z. a. (1999). *Viewing corridors as right parallelepipeds for vision-based vehicle localization*. IEEE Transactions on Industrial Electronics 46 (3).