

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS CIUDAD DE MÉXICO**

DIVISIÓN DE GRADUADOS E INVESTIGACIÓN DIRECCIÓN DE MAESTRÍA EN INGENIERÍA



**NAVEGACIÓN AUTÓNOMA DE ROBOTS MÓVILES MEDIANTE
LÓGICA DIFUSA**

**TESIS QUE PARA OPTAR EL GRADO DE
MAESTRO EN CIENCIAS COMPUTACIONALES
PRESENTA**

ALEJANDRO RAMÍREZ SERRANO

Asesor: Dr. MARC BOUMEDINE

**Comité de tesis: Dr. CARLOS RODRÍGUEZ LUCATERO
Dr. JESÚS SÁNCHEZ
Dr. MARC BOUMEDINE**

**Jurado: Dr. JESÚS SÁNCHEZ,
Dr. MARC BOUMEDINE,
Dr. CARLOS RODRÍGUEZ LUCATERO,**

**Presidente
Secretario
Vocal**

Tlalpan D.F., México, Mayo de 1996

CONTENIDO

	Pág.
AGRADECIMIENTOS	2
LISTA DE FIGURAS	5
LISTA DE TABLAS	7
CAPITULO	
1. INTRODUCCIÓN	8
1.1 Antecedentes del estudio de tesis	10
1.2 Objetivos del trabajo	10
1.3 La importancia y necesidad de desarrollar robots móviles	11
1.4 Problemas aún sin resolver de los robots móviles	13
1.4.1 Navegación	14
1.4.2 Reconocimiento (Recognition)	14
1.4.3 Aprendizaje	15
1.4.4 Cooperación	15
1.5 Como lograr los objetivos planteados	16
1.6 Resumen del trabajo a realizar	17
1.7 Organización del trabajo	17
2. NAVEGACIÓN Y PLANEACIÓN DE TRAYECTORIAS	20
2.1 Navegación	20
2.2 Técnicas de navegación y planeación de trayectorias	23
2.2.1 Técnicas de navegación que utilizan conceptos mat.	24
2.2.2 Técnicas de navegación que utilizan métodos gráficos	29
2.2.3 Otras técnicas de navegación	34
2.3 Comentarios sobre las técnicas estudiadas	39
2.4 Metodología a usar	40
3. LÓGICA DIFUSA (FUZZY LOGIC)	42
3.1 Introducción	42
3.2 Generalidades de la lógica difusa	43
3.3 Variables lingüísticas	44
3.4 Conjuntos difusos	45
3.5 Creación de conjuntos difusos	47
3.6 Representación de conjuntos difusos	48
3.7 Hedges	50

CAPITULO	Pág.	
3.8	Operaciones sobre conjuntos difusos	51
3.9	Inferencia difusa	52
3.10	Métodos de defuzificación	54
3.11	Comentarios finales	56
4.	ELABORACIÓN DEL CONTROLADOR DIFUSO	58
4.1	ETAPA I: Definición del problema	59
4.2	ETAPA 2: Definición de las variables lingüísticas	64
4.3	ETAPA 3: Definición de los conjuntos difusos	68
4.4	ETAPA 4: Definición de las reglas difusas	71
4.5	ETAPAS 5, 6 Y 7	75
5.	CONSTRUCCIÓN DEL SISTEMA	76
5.1	Organización del controlador difuso	76
5.2	Controlador difuso (Software)	79
5.2.1	primera fase (Simulación)	79
5.2.2	Ajuste del sistema	82
5.2.3	Segunda fase (Generación del código en C++)	83
5.3	Prototipo del robot móvil (Hardware)	86
6.	PRUEBAS	95
6.1	Ambiente y condiciones de navegación	95
6.2	Gráficas	97
7.	CONCLUSIONES	105
7.1	Resultados	105
7.2	Conclusiones	106
7.3	Trabajos futuros	107
ANEXO A	Reglas	109
ANEXO B	Inferencia Tipo Mamdani	115
ANEXO C	Controlador Difuso, Funciones en C++	118
ANEXO D	Especificaciones	127
REFERENCIAS		131

LISTA DE FIGURAS

FIGURA		Pág.
2.1	Los tres niveles dentro de la planeación de movimientos.	22
2.2	Regiones pesadas conforme la técnica de Mitchel.	26
2.3	Movimiento de un brazo mecánico de cuatro grados de libertad.	28
2.4	Diagramas de Voronoi y triangulación de Delaunay.	30
2.5	Representación geométrica el algoritmo de gráficas de visibilidad.	32
2.6	Navegación por marcas.	33
2.7	Sistema controlado por medio de redes neuronales.	35
2.8	Sistema de navegación basado en lógica difusa:	37
3.1	Conjuntos difusos sobre la variable lingüística estatura.	48
3.2	Conjuntos difusos sobre la variable lingüística estatura (muy).	50
3.3	Inferencia Max-Product.	53
3.4	Inferencia Max-Min.	53
3.5	Método de defuzificación centroidal.	54
3.6	Método de defuzificación Max-membership principle.	55
3.7	Método de defuzificación weighted average.	55
3.8	Ejemplos del método de defuzificación Mean-max.	56
3.9	Método de defuzificación center of sums.	57
4.1	Geometría de la navegación para un robot móvil.	60
4.2	Localización de los sensores de ultrasonido.	62
4.3	Robot móvil diseñado y construido durante el proyecto.	63
4.4	Conjuntos difusos sobre la variable "Ángulo del obstáculo".	69
4.5	Conjuntos difusos sobre la variable "Ángulo error".	69
4.6	Conjuntos difusos sobre la variable "Ángulo de dirección".	69
4.7	Conjuntos difusos sobre la variable "Velocidad".	70
4.8	Conjuntos difusos sobre la variable "Distancia del robot al punto meta".	70
4.9	Conjuntos difusos sobre la variable "Dist. del obstáculo".	70
4.10	Representación general de las reglas en forma de tablas.	72
4.11	Situaciones típicas en la navegación de robots móviles.	72
5.1	Sistema controlado mediante la teoría de control clásico.	76
5.2	Controlador difuso para el caso de la navegación de robots.	77
5.3	Sistema para controlar la navegación de un robot móvil.	78

FIGURA		Pág.
5.4	Proceso de inferencia y defuzificación.	81
5.5	Comunicación entre el controlador difuso y el robot móvil.	83
5.6	Componentes funcionales para controlar las acciones de un robot.	84
5.7.	Interfase Computadora-Robot Móvil.	87
5.8	Señales de entrada y salida del amplificador operacional.	87
5.9	Servomotores utilizados en el robot prototipo.	88
5.10	Configuración del motor que controla la velocidad del robot.	89
5.11	Principio de funcionamiento de un sensor de ultrasonido.	90
5.12	Diagrama esquemático del sensor de ultrasonido.	91
5.13	Interfase Robot móvil-Computadora.	93
6.1	Representación gráfica de los coeficientes de las ecuaciones 6.x.	96
6.2	Vector de referencia para localizar el punto meta.	97
6.3	Posición inicial del robot móvil.	97
6.4	Comportamiento del controlador difuso sobre el terreno No. 1.	98
6.5	Comportamiento del controlador difuso sobre el terreno No. 2.	99
6.6	Comportamiento del controlador difuso sobre el terreno No. 3.	100
6.7	Comportamiento del controlador difuso sobre el terreno No. 4.	100
6.8	Comportamiento del controlador difuso sobre el terreno No. 5.	101
6.9	Comportamiento del controlador difuso sobre el terreno No. 6.	102
6.10	Comportamiento del controlador difuso sobre el terreno No. 7.	102
6.11	Comportamiento experimental del robot móvil sobre el terreno No. 8.	103
6.12	Comportamiento experimental del robot móvil sobre el terreno No. 9.	103
6.13	Comportamiento experimental del robot móvil sobre el terreno No. 10.	103
6.14	Comportamiento experimental del robot móvil sobre el terreno No. 11.	104
6.15	Comportamiento experimental del robot móvil sobre el terreno No. 12.	104
6.16	Comportamiento experimental del robot móvil sobre el terreno No. 13.	104
B.1	Ejemplo de funciones de pertenencia.	116
B.2	Procedimiento de inferencia tipo Mamdani.	117
C.1	Diagrama de flujo del controlador difuso.	126

LISTA DE TABLAS

TABLA		Pág.
2.1	Ventajas y desventajas del método de proyección.	25
2.2	Ventajas y desventajas del método basado en regiones pesadas.	27
2.3	Ventajas y desventajas del método de campos potenciales artificiales.	28
2.4	Ventajas y desventajas del método basado en mapas de espacio libre.	31
2.5	Ventajas y desventajas del método de gráficas de visibilidad.	33
2.6	Ventajas y desventajas de la navegación por marcas.	34
2.7	Ventajas y desventajas del uso de redes neuronales.	37
2.8	Ventajas y desventajas del uso de lógica difusa.	38
2.9	Ventajas y desventajas del empleo de técnicas híbridas.	39
4.1	Variables lingüísticas con sus respectivos universos de cobertura.	65
4.2	Conjuntos difusos con sus respectivos adjetivos.	68
4.3	Reglas para cambiar la dirección del robot y evadir obstáculos.	73
4.4	Reglas para cambiar la dirección del robot y evadir obstáculos.	73
4.5	Reglas para cambiar la dirección del robot y evadir obstáculos.	74
4.6	Reglas para controlar la velocidad del robot.	74

1 INTRODUCCIÓN

Al estar frente a las puertas de un nuevo milenio, milenio en el cual el ser humano pretende detener el constante deterioro de la capa de ozono, conquistar nuevos planetas, construir y habitar estaciones espaciales, y finalmente, vivir en paz y armonía, es necesario para lograr dichos objetivos, la construcción de más y mejores máquinas “inteligentes” que le permitan al hombre trabajar de manera segura y eficiente en ambientes hostiles y peligrosos como lo serán el espacio exterior (en donde se construirán estaciones espaciales), la superficie de nuevos planetas y en las profundidades del océano, por mencionar solo algunos ejemplos. El presente trabajo de tesis, realizado para obtener el grado de maestría en ciencias computacionales, pretende contribuir con nuevas y mejores técnicas necesarias para la navegación autónoma de robots móviles, dispositivos necesarios e indispensables que serán vitales para poder lograr las metas antes mencionadas dentro del nuevo milenio.

Hace cuarenta años, cuando las primeras computadoras ocupaban cuartos enteros (i.e. UNIVAC), nadie hubiera pensado que una computadora podría ser usada como un controlador “fuel-injection” en todo tipo de automóviles, o incluso utilizadas en todo tipo de operaciones bancarias y comerciales, las cuales en la actualidad son manejadas por muchas personas de manera cotidiana. En épocas posteriores, con el uso de lenguajes de programación llamados de alto nivel (i.e. FORTRAN, PASCAL, C) y con el surgimiento de los primeros microprocesadores (Intel 8004 y 8008), comenzó el boom de la computación. Cuando comenzó el uso de estos componentes, la gente decía que con un poco de CPU, con

una poca memoria y con un poco de razonamiento lógico no se lograba mucho, pero eventualmente al transcurrir el tiempo, el uso de miles y miles de estos componentes comenzó a cambiar la manera en la cual el mundo moderno piensa y trabaja. En la actualidad miles de estos microprocesadores definen de manera radical la forma de resolver los problemas, tales como son el diseño y operación de robots, los cuales utilizan estos componentes que han cambiado la forma de pensar de como dar solución a los problemas que actualmente afectan a la humanidad.

Cabe recordar algunas de las áreas en las cuales la computación ha adquirido gran importancia como son la medicina (diagnósticos médicos), las finanzas (pronósticos y análisis financieros), la ingeniería (diseño de nuevos dispositivos y herramientas) y la robótica (i.e., MORIA [AR-69], Shakey [TX-2], Navlab [AR-3]). El área que el presente trabajo abarca y que esta íntimamente relacionada con la evolución de la computación, y en especial con las nuevas técnicas desarrolladas dentro de la inteligencia artificial (IA), es referente a la navegación de robots y en especial a la navegación autónoma de robots móviles.

Los robots son una herramienta que permite a la humanidad vivir y trabajar mejor, logrando un mejor desempeño en las tareas cotidianas. Es de especial interés, el desarrollo de robots móviles que ayuden al ser humano en la ejecución de tareas peligrosas, que son realizadas por obreros y empleados en diversas áreas. Algunas de estas tareas, incluyen la exploración e investigación vulcanológica, la limpieza de desechos nucleares, la explotación de pozos petroleros y el manejo de productos químicos, por mencionar solo algunos ejemplos.

La construcción y diseño de sistemas robóticos "inteligentes" que sean capaces de "razonar" al mismo tiempo que trabajan en ambientes altamente impredecibles y dinámicos, es una tarea altamente complicada. Estos sistemas deben además, trabajar sin la constante supervisión e intervención humana. Por estas razones tales máquinas autónomas requieren poseer requerimientos computacionales de vanguardia con los que puedan trabajar de manera continua en ambientes desconocidos, hostiles y peligrosos tomando decisiones a partir de su conocimiento en tiempo real. Para ello, estos sistemas deben poseer la habilidad de monitorear y reaccionar a eventos (dinámicos o estáticos) totalmente inesperados que ocurran a sus alrededores con la ayuda de sensores (i.e. de ultrasonido, ópticos, cámaras, magnéticos, infrarrojos, etc.) los cuales les permitan percibir su entorno o con los cuales les sea posible crear modelos (bidimensionales o tridimensionales) del ambiente en el que se desempeñen.

1.1 ANTECEDENTES DEL ESTUDIO DE TESIS

El desarrollo de nuevas y mejores tecnologías, es una necesidad básica dentro de todo país con aspiraciones de crecimiento. México, un país con grandes cambios y necesidades, requiere del desarrollo de propias tecnologías que impulsen, apoyen y alienten su desarrollo. Un área de especial interés que requiere impulso y desarrollo es la robótica.

A partir de la década de los setenta, la robótica (i.e. manipuladores y vehículos autónomos) se ha destacado como una herramienta sumamente rentable y versátil, necesaria para la realización de trabajos peligrosos y/o repetitivos [TX-20]. Desafortunadamente, el desarrollo de robots móviles en México es poco y casi desconocido. Es por esto, y dado que se desea apoyar al sector industrial con mejores dispositivos de trabajo, es que se quiere impulsar esta área tecnológica, en especial dentro del Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM). En este esfuerzo de desarrollar mejores robots móviles, la aportación que surgirá a partir del presente trabajo, es el de desarrollar, encontrar y/o mejorar alguna de las técnicas ya existentes para la planeación de trayectorias en 2-D (de robots móviles) en tiempo real. Las técnicas a estudiar y/o desarrollar serán técnicas basadas en la inteligencia artificial, tales como búsquedas heurísticas, redes neuronales y lógica difusa que determinen y planifiquen las trayectorias de movimiento de dichos sistemas.

1.2 OBJETIVOS DEL TRABAJO

Durante la última década la robótica ha sufrido una profunda transformación. Muchos investigadores en robótica que comenzaron sus trabajos en manipuladores industriales han emigrado hacia la investigación de más avanzados sistemas robóticos, particularmente hacia los robots móviles, los que generalmente son destinados a trabajar fuera de las áreas de manufactura (i.e. procesos de maquinado, ensamble, etc.).

Mientras que los métodos desarrollados para los manipuladores industriales están trasladándose hacia más sofisticados robots, también se reconoce que éstos métodos no siempre satisfacen las necesidades de máquinas diseñadas a desempeñar trabajos útiles y complejos en ambientes poco conocidos. Para enfrentar éstas necesidades, enlaces con otras disciplinas (computacionales y no computacionales) tales como computación en tiempo

real (real time computing), inteligencia artificial (IA), psicología y neuropsicología están siendo establecidas. Estos esfuerzos han originado muchos interesantes resultados expuestos en gran cantidad de conferencias internacionales alrededor de todo el mundo, en donde se ha demostrado la importancia de éste nuevo campo de la ciencia. Sin embargo, a pesar de los avances, es necesario realizar más estudios e investigación para lograr mejores sistemas robóticos, en especial en el área de planeación y navegación.

El presente trabajo de tesis desea proporcionar una solución adecuada al problema de navegación (que se describirá en los siguientes capítulos) el cual es todavía en la actualidad un problema sin resolver de manera satisfactoria [TX-2], [TX-3], [TX-4] dentro de la robótica móvil. Con esto, un robot móvil será capaz de navegar de manera segura (sin sufrir colisiones) a través de un ambiente totalmente desconocido a priori por el robot, hasta alcanzar un punto final llamado meta a partir de su posición inicial sobre superficies o ambientes de dos dimensiones (2-D) (superficies planas). Además de poder lograr esto, el presente trabajo pretende optimizar el desempeño del robot (o por lo menos incrementar la eficiencia en la navegación). Esto se logrará al minimizar tanto la distancia recorrida, como el tiempo de navegación (es decir, lograr que el robot se mueva a través de una trayectoria óptima empleando el menor tiempo posible en llegar desde su posición inicial al punto meta). Para alcanzar estos objetivos, se analizarán varias de las técnicas dentro de la IA empleadas hasta el momento por varios investigadores que han tratado de resolver problemas similares al del presente trabajo de tesis. Observado los pros y contras de dichas metodologías se decidirá la técnica que se usará para lograr los objetivos antes mencionados, ya sea modificando la técnica elegida o implementando un nuevo procedimiento para nuestro caso en particular.

1.3 LA IMPORTANCIA Y NECESIDAD DE DESARROLLAR ROBOTS MÓVILES

Por muchos años, la tecnología robótica se ha desarrollado principalmente alrededor de aplicaciones industriales, como el ensamble, la manufactura, la soldadura y el manejo de materiales. Recientemente, los robots móviles se han convertido en otra generación de máquinas, principalmente utilizadas como dispositivos de servicio, dedicados a desempeñar tareas sencillas que actualmente son realizadas por personas poco calificadas. Entre estas

tareas, se incluyen trabajos de transporte de material, limpieza de pisos y trabajos de servicio en hospitales, por mencionar solo algunos.

Dejando a un lado la excitación y el entusiasmo de crear robots que en realidad corran y se desplacen a través de terrenos hostiles, existen varias razones importantes para investigar y diseñar robots móviles impulsados con ruedas o con patas. El desarrollo de robots móviles autónomos capaces de tomar sus propias decisiones de movimiento en ambientes totalmente nuevos e inesperados, ayudará a diversas áreas de investigación. La vulcanología, una de estas áreas, donde la necesidad de obtener muestras directamente de fumarolas y cráteres activos, sin poner en peligro la integridad física de los investigadores, es de suma importancia. En la exploración espacial, donde la realización de tareas como reparación y recuperación de satélites, recolección de muestras y realización de mapas de cuerpos celestes es vital para la búsqueda de nuevas fuentes de energía y en el combate de la contaminación. Estas, y otras muchas áreas se verán beneficiadas con el desarrollo de mejores y sofisticados robots móviles autónomos. La navegación y planeación de trayectorias es de especial interés para los robots móviles diseñados a trabajar en ambientes totalmente desconocidos, y para los cuales algoritmos basados en los principios de la inteligencia artificial son sumamente deseados. Por esto, es que se desea desarrollar y/o mejorar las técnicas de planeación de trayectorias y/o navegación que actualmente existen para estos nuevos dispositivos (robots móviles).

Los "robots autónomos móviles" o "autonomous mobile robots (AMR's)" son sistemas en movimiento capaces de interpretar, planear, ejecutar tareas (cualquiera que esta sea) y navegar de manera autónoma a través de un ambiente desconocido o parcialmente conocido, sin ayuda externa por parte de un ser humano [AR-5], [AR-40]. Para lograr estas habilidades, el robot debe ser capaz de construir modelos internos del terreno, que con cierto detalle representen las características topológicas y geométricas del ambiente que le rodea. En consecuencia, dichos modelos del mundo son las bases que influenciarán su comportamiento con base en diversas estrategias como son el uso de técnicas de IA (i.e. redes neuronales, lógica difusa, búsquedas heurísticas, algoritmos genéticos, etc.).

Durante los últimos 50 años (aproximadamente) el interés y contribuciones hacia máquinas móviles autónomas inteligentes (robots móviles) dentro de muchas universidades tales como Stanford (Stanford Cart [AR-1] y Shakey [TX-2]), Carnegie Mellon University (Neptune mobile robot [AR-2], The Ambler and The Navlab [AR-3]) y MIT (Attila y Genghis [TX-3]) además del interés de varias compañías como NASA (JPL Robotic Vehicle [AR-4]), Naval Ocean Systems (Robart II [TX-3]), International Robotic Systems (RM3 legged robot

[TX-2]) entre muchas otras, ha crecido significativamente dado que este tipo de máquinas ayudará al ser humano en la realización de sus presentes y futuras tareas con un mayor grado de eficiencia, tales como la conquista y exploración de nuevos mundos, la extracción y explotación de minerales y el manejo de materiales radiactivos. Con el diseño y construcción de mejores vehículos autónomos, se minimizará el riesgo que sufren las personas durante la realización de su trabajo, al mismo tiempo que se tenga un desempeño seguro y eficiente del mismo. Por otra parte, la investigación en éste campo ha recibido bastante atención dado sus amplios usos potenciales que van desde la vigilancia o patrullaje hasta la exploración planetaria, además de que provee oportunidades para realizar investigación en virtualmente todo el campo de la robótica (desde el control inteligente hasta la planeación y ejecución de tareas).

Obviamente, la elaboración de dichos sistemas autónomos inteligentes requiere de un trabajo multidisciplinario en donde se conjunten diversas áreas tales como la mecánica, la electrónica y claro esta, las ciencias computacionales, con la cual pueden atacarse problemas de adquisición de datos (a partir de sensores), manejo de bases de conocimientos (almacenadas en la memoria del robot), visión computacional, interpretación de información (proveniente de sensores), modelado del mundo, control de actuadores, planeación de trayectorias y la navegación (aspecto que será cubierto en el presente trabajo).

1.4 PROBLEMAS AUN SIN RESOLVER DENTRO DE LOS ROBOTS MÓVILES

A pesar de los grandes esfuerzos y la gran cantidad de trabajos realizados para tratar de hacer que los robots móviles se comporten de manera parecida y con los niveles de eficiencia presente de los sistemas naturales (i.e. insectos, aves), en la actualidad aún existen muchos problemas que requieren de mucho más trabajo para lograr que dichos sistemas alcancen tales niveles de eficiencia. La navegación, la cooperación, el reconocimiento y el aprendizaje son algunas de las áreas dentro de las ciencias computacionales en donde aún se presentan problemas a pesar de la gran cantidad de trabajo y esfuerzo en torno a ellos que al resolverlos se lograra hacer que los robots móviles alcancen los niveles de comportamiento esperado. Principalmente esto es debido a que estos problemas son sumamente complejos y por ende difíciles de controlar, de predecir y de modelar matemáticamente. A continuación se presenta un breve panorama de estos problemas.

1.4.1 NAVEGACIÓN

Como ya se menciona, en la actualidad existen muchos problemas aun sin resolver dentro del área de los robots móviles. Una de las preguntas abiertas aun sin responder de manera definitiva y satisfactoria, tiene que ver con la determinación de los factores que están involucrados y al mismo tiempo son necesarios para dotar a un robot móvil con la habilidad y/o capacidad de navegar a través de su medio ambiente [TX-2].

Dentro del mundo real es posible observar como es que la naturaleza a resuelto el problema de la navegación eficientemente y de diversas maneras [TX-3]. Los salmones por ejemplo, son capaces de encontrar sus lugares de desove desde cientos de kilómetros de distancia, las palomas por otra parte, son capaces de encontrar sus lugares de destino en días nublados, soleados y aún lluviosos. Las abejas son capaces de establecer rutas con las cuales pueden encontrar la fuente de comida. Los murciélagos aún estando casi ciegos, son capaces de evitar obstáculos. Por el contrario, muy pocos robots móviles [AR-69], [AR-48], [AR-22] son capaces de navegar a través de un simple pasillo (corredor) sin la constante supervisión y corrección de su trayectoria por parte de un ser humano. Es por esto, y motivados por la alta eficiencia encontrada en la naturaleza, por lo que el presente trabajo intenta contribuir en menor o mayor grado a la solución de este problema, "*La navegación de vehículos autónomos*".

Otros de los problemas aun sin resolver dentro de la robótica móvil que no se estudiara dentro del presente trabajo, pero no por ello menos importantes, son los problemas de reconocimiento, que en cierto grado va ligado a la navegación de robots, el problema de aprendizaje y el problema de cooperación de robots (móviles o no) entre otros muchos que no se mencionarán aquí por no pertenecer al área de las ciencias computacionales, área para la cual el presente trabajo esta enfocado. A continuación un pequeño resumen de estas dificultades.

1.4.2 RECONOCIMIENTO (Recognition)

Uno de los problemas aún totalmente sin resolver y que de cierta manera va de la mano con la navegación, es el de reconocimiento [TX-3] problema que consiste en identificar de manera adecuada el ambiente de trabajo para poder desarrollar eficientemente las tareas dentro del mismo. Dentro de los robots móviles, el reconocimiento de marcas en el terreno dentro de

ambientes no estructurados es un problema muy difícil de resolver [TX-2]. Ya sea utilizando cámaras, sensores piezoeléctricos, sensores de fuerza e incluso micrófonos, el reconocimiento de patrones en el medio ambiente es un problema no trivial. El problema del reconocimiento es un problema que requiere de intensos esfuerzos computacionales el cual esta sujeto a problemas y dificultades presentes en el ambiente tales como la luz, oclusión e incluso a datos o información ruidosa.

1.4.3 APRENDIZAJE

Otro de los problemas dentro de la robótica móvil todavía sin resolver a los niveles esperados, es el problema del aprendizaje maquina (machine learning) [TX-3] en el cual se investiga como, cuando y donde, algoritmos de aprendizaje pueden incorporarse en el sistema inteligente (robots) de tal forma que se disminuya la carga del programador haciendo que el sistema aprenda y se autoprograme (por decir así). Conforme más y mejores robots móviles son diseñados, estos se vuelven cada vez más complejos. Este incremento de complejidad se debe al hecho de que cada día que transcurre se incorporan más sensores y actuadores al sistema, esto automáticamente incrementa el grado de sofisticación del software necesario que controla dichos sistemas (manejar más entradas, posiblemente proveniente de sensores, y salidas (acciones del robot)).

Dentro del área del aprendizaje, se trata básicamente de responder a preguntas tales como, Qué es lo que los sistemas robóticos deben aprender ?, Podrán los robots calibrar sus propios sensores ? Podrá un robot móvil aprender nuevos y mejores comportamientos ?, etc.

1.4.4 COOPERACIÓN

Finalmente, otro de los problemas aún sin resolver dentro de la robótica, es el problema de cooperación entre robots [TX-3], [AR-86]. En la actualidad existen gran cantidad de robots industriales que ayudan al ser humano en sus tareas cotidianas (i.e. Robots Mitsubishi, Robots Unimation). Conforme pasa el tiempo, el hombre requiere de mejores máquinas que le ayuden en tareas cada vez más complejas y en las cuales se requiere cooperación hombre-hombre, hombre-máquina y máquina-máquina. La cooperación hombre-hombre y hombre-máquina esta hasta cierto punto resuelto, sin embargo, la cooperación máquina-máquina es un problema aún sin resolver, en donde los robots tienen que trabajar inteligentemente de manera conjunta y cooperando ente ellos (ayudarse) sin interferir unos

con otros para lograr un objetivo común. Dentro de la robótica móvil este problema se enfoca principalmente en lograr que una serie de robots móviles trabajen conjuntamente (como lo hacen las colonias de hormigas) de manera inteligente en ambientes hostiles.

En la actualidad este problema esta siendo estudiado intensamente por Rodney Brooks en el Instituto Tecnológico de Massachusetts (MIT) con sus microrobots (mayor información acerca de dicho proyecto puede consultarse la siguiente dirección dentro del Word Wide Web (WWW) dentro de la página principal del MIT: <http://www.ai.mit.edu/projects/microrobotics/>).

1.5 COMO LOGRAR LOS OBJETIVOS PLANTEADOS

El presente trabajo tiene como finalidad mejorar y/o desarrollar alguna técnica de inteligencia artificial idónea para controlar y planear de manera eficiente las trayectorias de un robot móvil. Para lograr esto, los principales puntos realizados durante el estudio los cuales conforman los siguientes capítulos fueron los siguientes:

- Investigación bibliográfica en torno a las técnicas utilizadas actualmente para el control, planificación de trayectorias y navegación de robots.
- Estudio de las posibles alternativas para controlar, navegar y planear los movimientos y trayectorias de un robot móvil, identificando sus ventajas, desventajas y las características de las técnicas que actualmente existen.
- Selección de la(s) técnica(s) a implementar y/o mejorar para el control y planeación de los movimientos y trayectorias de un robot móvil, con base en los resultados obtenidos por otros investigadores y observaciones realizadas en los puntos anteriores.
- Desarrollar la(s) técnica(s) para controlar y planear los movimientos y trayectorias (navegar) de un robot móvil seleccionada(s) en los puntos anteriores. En esta parte del trabajo se empleo la simulación y la experimentación (para esto se construyo un robot prototipo).
- Analizar el desempeño (performance) de la(s) técnica(s) desarrolladas o mejoradas por medio de la simulación y experimentación.
- Presentar conclusiones y resultados obtenidos.

1.6 RESUMEN DEL TRABAJO A REALIZAR

El presente trabajo de tesis consiste principalmente en implementar y/o mejorar una técnica o técnicas de IA para la navegación y planeación de trayectorias en tiempo real de un robot móvil, el cual estará diseñado para moverse únicamente sobre superficies planas con obstáculos sin movimiento. El sistema de navegación y planeación de trayectorias a desarrollar, consistirá de algoritmos inteligentes mediante los cuales el robot decidirá la velocidad y la orientación de sus ruedas (dirección) para así poder llegar a su objetivo (punto meta) sin sufrir colisiones, evitando los obstáculos presentes en el terreno. Se estudiarán las técnicas más importantes de planeación y navegación desarrolladas por varios investigadores con el fin de determinar la(s) técnica(s) más conveniente(s) para llevar a cabo dicho propósito. La o las técnicas a emplear, serán estudiadas para seleccionar la mejor, o en su defecto, una combinación de ellas (método híbrido). Entre las técnicas a estudiar se encuentran las redes neuronales, algoritmos de búsqueda heurística, algoritmos genéticos y lógica difusa (Fuzzy Logic).

Durante el estudio, además de la utilización de lenguajes de programación para la implementación, se utilizarán diversos paquetes computacionales que ayudarán a determinar y/o a diseñar él o los algoritmos de manera rápida mediante simulaciones y análisis realizados con dichos apoyos

1.7 ORGANIZACIÓN DEL TRABAJO

El presente trabajo se divide en siete capítulos. El capítulo No. 1 (este) presenta los objetivos del trabajo así como los motivos que llevaron a su elaboración. En el capítulo No. 2 se presentan las técnicas de planeación y navegación más utilizadas hoy en día dentro de la robótica. A partir de esta recopilación de información se elaboran una serie de tablas en donde se resumen las ventajas y desventajas de cada método estudiado con base en las características que se desean contenga el sistema de navegación a desarrollar. A partir de este estudio se determinará la(s) metodología(s) a usar. El capítulo No. 3 presenta los conocimientos básicos sobre la técnica de navegación a emplear (seleccionada en el capítulo No. 2). El propósito de este capítulo es el de presentar un panorama general al lector que no

este familiarizado con la metodología seleccionada y con la cual se elaborara el sistema de navegación. Los siguientes tres capítulos presentan la secuencia de pasos mediante los cuales se elaboro el sistema de navegación cuyos resultados son expuestos en el último capítulo del presente trabajo. En el capítulo No. 4 se diseña y elabora la técnica de navegación a utilizar. Aquí también se describe el problema que se desea resolver así como las restricciones en el mismo. Dentro del capítulo No. 5 se describe la construcción del sistema previamente diseñado en el capítulo No. 4. La construcción consiste de un programa (software) y de un robot prototipo (hardware) con los que se realizaron pruebas de simulación y experimentación. El capítulo No. 6 presenta las pruebas (simulación y experimentación) realizadas sobre el sistema de navegación diseñado y finalmente en el capítulo No. 7 se presentan las conclusiones del trabajo y se proporcionan ideas para posibles trabajos futuros.

- -
- -

2 NAVEGACIÓN Y PLANEACIÓN DE TRAYECTORIAS

Como se menciona en la parte introductoria, el presente trabajo se enfoca principalmente en proporcionar una solución al problema de la navegación autónoma de robots móviles. El presente capítulo proporciona los conceptos básicos sobre navegación incluyendo su definición. También contiene una breve descripción de las técnicas de navegación más populares y más utilizadas dentro de la robótica móvil. Después de cada metodología se presenta una tabla en donde se muestra un resumen de las ventajas y desventajas de la técnica de navegación previamente vista con base en el problema que se desea resolver (navegación autónoma de robots móviles en ambientes desconocidos). La elaboración de dichas tablas ayudara a decidir la(s) metodología(s) a emplear para resolver el problema que nos concierne. Esta elección es realizada en la parte final del capítulo.

Es difícil realizar una división clara de las técnicas de navegación en grupos o en categorías debido a que existen una gran diversidad de metodologías. Existen técnicas de navegación basadas en conceptos matemáticos, otras basadas en gráficas y diagramas, otras que utilizan y las que no utilizan mapas. También existen metodologías de navegación que son empleadas en ambientes dinámicos y las que son empleadas en ambientes estáticos. Además existen técnicas que únicamente reconocen cierta clase de obstáculos, técnicas empleadas en brazos manipuladores, técnicas que están enfocadas hacia vehículos autónomos, etc., y claro esta también existen combinaciones de las técnicas anteriores. Por estas razones y dado que nuestro problema se enfoca únicamente en la navegación autónoma de robots móviles en ambientes estáticos, dentro del presente capítulo únicamente se describirán metodologías relacionadas o que puedan ser aplicadas a nuestro problema.

Aún cuando es difícil categorizar las técnicas de navegación, las técnicas aquí estudiadas se han dividido en tres grupos. En el primer grupo se encuentran las técnicas que

utilizan conceptos matemáticos (i.e. modelos matemáticos del terreno) mientras que en el segundo grupo se encuentran técnicas que utilizan métodos gráficos (i.e. mapas del terreno). El tercer grupo contiene técnicas que no pertenecen a ninguno de los dos grupos anteriores (i.e. lógica difusa, técnicas híbridas).

2.1 NAVEGACIÓN

La investigación en torno a los robots móviles abarca varias áreas dentro de la robótica tradicional, incluyendo diversas disciplinas tales como visión de máquina [AR-70], [AR-71], interpretación y acoplamiento de sensores [AR-2], [AR-3], [AR-5] modelado del medio ambiente [TX-4], control de actuadores [TX-17], planeación y ejecución de tareas, navegación y planeación de trayectorias [TX-4], [TX-5], [AR-4], [AR-6], [AR-8] entre otras. Dentro de éstas, la investigación en torno a la navegación [TX-16], [AR-10], [AR-72] y al control de actuadores [TX-2], [TX-4], [TX-11] ha recibido mucha atención por parte de la comunidad científica, donde la navegación y planeación de trayectorias es un área de vital importancia en el desempeño general del robot. Mediante la planeación de trayectorias el robot es capaz de determinar el mejor camino a seguir para llevar a cabo su tarea de la manera más eficiente dentro de sus posibilidades.

En décadas recientes, gran interés se ha generado en el diseño de algoritmos para aplicaciones en robótica tales como planeación de movimientos y trayectorias [AR-72], estimaciones de posición, adquisición de información y control inteligente [TX-4], [TX-7], [TX-17]. En particular, algoritmos desarrollados para la planeación de movimientos [AR-8], [AR-10], [AR-16], ha originado un campo de intensa investigación. Típicamente, esta nueva área investiga algoritmos para la planeación de trayectorias a través de las cuales los sistemas robóticos, tanto robots móviles como manipuladores, deberán seguir desde una configuración inicial hasta una posición final, pasando y cumpliendo con restricciones de movimiento tales como obstáculos y condiciones particulares del entorno. La navegación y planeación de trayectorias es de especial interés para los robots móviles destinados a trabajar en ambientes totalmente desconocidos, y para los cuales, algoritmos basados en los principios de la IA son sumamente deseados. En la actualidad existen varias técnicas que resuelven el problema de la planeación de trayectorias [AR-17], [AR-20], [AR-28] pero únicamente de manera parcial, sin que exista una verdadera autonomía por parte del robot y/o tengan la eficiencia

encontrada en los sistemas de navegación presentes en la naturaleza (i.e. el vuelo de los murciélagos, el sistema de navegación de los delfines).

Las investigaciones realizadas en torno a la planeación de trayectorias, incluyen técnicas de búsquedas heurísticas [AR-7], planeación de movimientos usando diagramas generalizados de Voronoi [TX-2], [TX-4], [TX-5], [AR-16] y técnicas de control utilizando redes neuronales [TX-2], [TX-17]. La mayoría de estos algoritmos se han desarrollado para robots diseñados a moverse en ambientes totalmente conocidos con anticipación (se cuenta con mapas), como fabricas, bodegas y hospitales. Algoritmos inteligentes también han sido desarrollados para evitar obstáculos [AR-72], [AR-73]. Tales algoritmos incluyen técnicas de búsquedas en bases de datos, de donde se obtienen trayectorias de costo mínimo y en donde la base de datos contiene todas las especificaciones acerca del entorno en donde se moverá el robot. Estas técnicas, son adecuadas para robots móviles destinados a moverse en ambientes conocidos. Sin embargo, para un robot móvil destinado a moverse a través de ambientes o terrenos totalmente desconocidos, estas técnicas son inadecuadas. Es por esto que deben diseñarse métodos que ayuden a este tipo de robots a tomar sus propias decisiones, basadas en cierto aprendizaje y/o en búsquedas basadas en conocimiento.

Cuando el medio ambiente en el cual el robot se moverá no es conocido de antemano (no se cuenta con mapas del terreno), es necesario desarrollar otras técnicas alternas de navegación a las elaboradas para robots que se desempeñan en ambientes conocidos y/o controlados.

En robótica "navegación" se define como "*la ciencia de dirigir el curso de un robot (que puede ser móvil) a través de un espacio (tierra, aire, agua o espacio) conforme el robot se mueve*" (parcialmente tomada de [TX-2]). Dentro de la robótica móvil, y en general dentro de todos los robots autónomos, algunos autores [AR-6] identifican tres niveles necesarios para que dichos sistemas puedan planear y navegar adecuadamente. Los tres niveles mencionados son el nivel de misión, el nivel de navegación y el nivel de pilotaje o nivel piloto, como puede apreciarse en la figura 2.1.

Algunos otros autores [TX-8] consideran un cuarto nivel llamado "nivel controlador" el cual no es considerado por muchos investigadores, debido a que este nivel es común a todo tipo de robots (sistemas) autónomos, pues dicho nivel es el encargado de controlar directamente los actuadores que impulsan al robot.

El planeador de "misión", mencionado anteriormente, es el encargado de tomar un conjunto de objetivos como entrada y proporcionar un plan como salida con el cual se

alcanzarán los objetivos. El nivel de misión es considerado el nivel más alto y el único nivel que esta en contacto con un posible supervisor humano del que recibe la información necesaria sobre los objetivos que tiene que cumplir. El nivel de misión es el encargado de dividir el problema en sub-problemas los cuales formarán sub-objetivos a ser alcanzados.

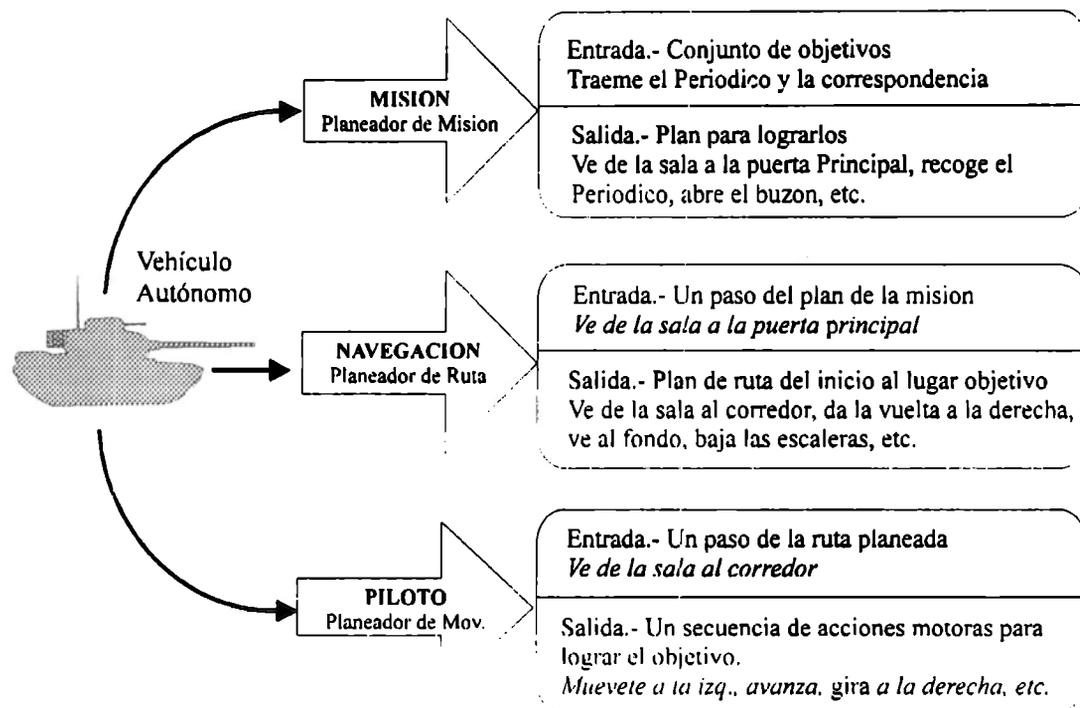


Fig. 2.1 Los tres niveles dentro de la planeación de movimientos y trayectorias para robots autónomos (misión, navegación y piloto).

El nivel de "navegación" es el encargado de eludir los obstáculos con base en la información proveniente de los sensores que el robot posea y con los cuales el robot debe ser capaz de elaborar un mapa o un modelo del ambiente que le rodea. Con el modelo o mapa, el robot tendrá la capacidad de planear y elaborar trayectorias seguras (sin colisiones) por las cuales pasará.

El nivel de "piloto" es el encargado de evaluar el terreno próximo al robot para que esté pueda determinar el orden o secuencia de movimientos que debe realizar para avanzar en la dirección deseada o alcanzar el objetivo esperado. Generalmente el nivel de piloto recibe como entradas uno de los pasos dentro de la ruta planeada (sub-objetivo) y proporciona como salida una secuencia de acciones hacia los actuadores con las cuales logrará la meta intermedia. Para lograr la planeación y navegación deseada, los tres niveles (misión, navegación, piloto) tienen que poder comunicarse con el fin de transmitirse la información. Esta comunicación generalmente se logra mediante conexiones físicas

(hardware) o mediante la creación de archivos codificados (software) que son elaborados por cada nivel según la información que desee transmitir. Estos archivos pueden ser leídos por cualquier nivel que requiera de dicha información (i.e. bases de datos creadas con información del terreno por donde se navegara y el objetivo a cumplir).

Un sistema completo de planeación para robots autónomos móviles (que es el tema que nos atañe) tiene que tener la habilidad de planear a diferentes niveles de resolución espacial (2D ó 3D), así como también manejar las interacciones entre la planeación de tareas y el medio ambiente. Dentro de la navegación, el conocimiento espacial, y la planeación de trayectorias, problemas clásicos de robots móviles que han recibido considerable atención por parte de investigadores dentro de la IA [AR-6 a AR-19], son de suma importancia dado que dichos sistemas trabajan y se desempeñan en ambientes tridimensionales. Para atacar éste problema se han desarrollado diversos modelos, estructuras de datos y algoritmos de planeación con los cuales se han logrado construir diferentes robots que trabajan de manera satisfactoria. Recientemente, se ha generado gran interés en el diseño de nuevos algoritmos con aplicaciones a la robótica y en especial a los robots móviles, que permitan ser usados en diversas aplicaciones como en la planeación de movimientos, estimación de la posición con base en información proveniente de sensores y en la adquisición de información referente al terreno. Por lo general el área de la planeación de movimientos tiene que ver con el desarrollo de algoritmos. Estos algoritmos deben ser capaces de planear trayectorias a través de las cuales él o los robots (móviles o no) naveguen desde una configuración inicial hasta una configuración final, sujetos a ciertas restricciones de movimiento.

2.2 TÉCNICAS DE NAVEGACIÓN Y PLANEACIÓN DE TRAYECTORIAS

A continuación se describirán las técnicas y algoritmos más utilizados por varios investigadores, los cuales tratan de resolver el problema de navegación de robots móviles tanto en ambientes conocidos como desconocidos. Dichas metodologías se estudiarán y analizarán observando sus ventajas y desventajas con base en nuestro problema a resolver con lo que se podrá tomar una decisión sobre la técnica o técnicas a emplear dentro del presente trabajo. Las técnicas presentadas se encuentran clasificadas en los tres grupos descritos al inicio del capítulo. En el primer grupo se encuentran las técnicas que utilizan

conceptos matemáticos. En el segundo se agrupan las técnicas que utilizan métodos gráficos y el tercer grupo contiene técnicas que no pertenecen a ninguno de los dos grupos anteriores. No es el propósito del capítulo hacer un estudio de todas las técnicas de navegación existentes hoy en día puesto que el número es tan grande que difícilmente se terminaría. Aquí únicamente se expondrán las técnicas más comunes y las más utilizadas para la navegación de robots móviles.

Después de describir brevemente cada una de las técnicas seleccionadas de navegación se presenta una tabla en donde se resumen las ventajas y desventajas del método con base en las siguientes dos características deseadas principalmente en el sistema de navegación para robots móviles a diseñar:

- El sistema de navegación a diseñar debe ser capaz de trabajar satisfactoriamente bajo ambientes desconocidos a priori por el robot.
- El sistema de navegación a diseñar debe ser capaz de trabajar en tiempo real.

2.2.1 TÉCNICAS DE NAVEGACIÓN QUE UTILIZAN CONCEPTOS MATEMÁTICOS

Las técnicas de navegación que emplean conceptos matemáticos se apoyan fuertemente en ecuaciones algebraicas, ecuaciones diferenciales, etc. para representar de alguna manera ya sea el terreno por el cual el robot navegara (modelo del terreno) o bien representar "aproximadamente" el sistema (modelo matemático del robot móvil). Estos modelos son la base sobre los cuales se planean los movimientos del robot.

En cuanto a este tipo de metodologías de navegación, Schwartz y Sharir [AR-20] realizaron en 1988 un examen de algunos algoritmos desarrollados para la planeación de movimientos concentrándose en soluciones exactas proporcionadas por tales procedimientos. Dentro de los métodos examinados en [AR-20] se incluyen el método de proyección (projection method) y el método de retracción (retraction method) los cuales se describirán a continuación.

El método de proyección es un método de planeación de movimientos el cual consiste en planear los movimientos de un sistema de "n" grados de libertad (i.e. un robot manipulador) en ambientes de 2 ó 3 dimensiones. El ambiente o terreno es representado por un conjunto de espacios FP (Free positions) libres de colisiones por el cual el sistema puede moverse de manera segura. La clave de este método esta en la manera de representar el espacio FP, el cual es representado mediante un conjunto de desigualdades algebraicas que

expresan el hecho de que en la posición "Z" (que cumple con las desigualdades con las cuales es representado el espacio libre de colisiones), el sistema evita sufrir colisiones con cualquiera de los obstáculos presentes en el ambiente. Dentro de este método, el número "k" de desigualdades necesario para representar el espacio libre de colisiones es llamado la complejidad geométrica (geometric complexity) la cual cambia (de forma dinámica) conforme el sistema se mueve a través del ambiente. Esta técnica ha sido utilizada en varios sistemas (i.e. el problema del "piano mover", brazos manipuladores) en donde se han obtenido soluciones en tiempos polinomiales. Sin embargo, recientemente este método ha sido mejorado (para algunos casos particulares) en los cuales se han obtenido algoritmos del orden de $O(n^2 \log n)$ [AR-28], el cual se pudo mostrar que era casi una solución óptima.

TABLA 2.1 Ventajas y desventajas del método de proyección.

MÉTODO DE PROYECCIÓN (projection method)	
Ventajas	Desventajas
<ul style="list-style-type: none"> • El método es aplicable a espacios en 2 y 3 dimensiones. • Es aplicable a sistemas de "n" grados de libertad. • Con este método es posible representar obstáculos de diversas formas y tamaños. • El método maneja translaciones y rotaciones del sistema a mover (i.e. robot móvil). • Debido a las características del método, el problema de planeación se reduce a una búsqueda discreta de trayectorias sobre un grafo obtenido a partir del espacio FP. • Es posible emplear el método para coordinar sistemas independientes trabajando en la misma área de trabajo [AR-30] (i.e. varios robots móviles o brazos mecánicos cooperando entre sí). 	<ul style="list-style-type: none"> • Es difícil representar de manera adecuada mediante desigualdades algebraicas (requerido por el algoritmo) el espacio FP del ambiente por el cual el robot navegará. • Como el espacio FP (free positions) cambia a cada instante conforme el robot se mueve, es necesario definir el nuevo espacio FP a cada instante mediante nuevas desigualdades. • Es difícil utilizar el método en tiempo real. • Es difícil generalizar el espacio FP para navegar a través de una gran gama y configuraciones de terrenos o ambientes. • El método es capaz de resolver una gran cantidad de problemas, pero cada algoritmo desarrollado resuelve únicamente problemas específicos [AR-28 y 29]. • Para poder emplear este método, es necesario conocer el ambiente (mapa) en donde se desempeñara el sistema. • La complejidad del método aumenta drásticamente si se emplea en ambientes dinámicos.

Por otra parte el método de retracción (retraction method) utilizado para la navegación de sistemas, es un método simplificado del método de proyección, donde el espacio libre FP se reduce a dimensiones menores (usualmente en una sola dimensión) con lo que se reduce el problema de navegación. La ventaja de este método sobre el método de proyección es que es más fácil obtener el conjunto de desigualdades algebraicas para representar el terreno por donde navegara el robot

Otra de las técnicas desarrolladas para la navegación de vehículos autónomos que se basa en conceptos matemáticos es la técnica de navegación mediante regiones pesadas (weighted regions) desarrollada y utilizada por Mitchell [AR-10], la cual emplea el mapa del terreno por el cual el robot navegará. Aquí se requiere que el mapa que representa el terreno este formado por un conjunto de regiones, cada una de las cuales tiene asociado un peso (o un costo). El peso de cada región representa el "costo por unidad de distancia" que el vehículo paga al atravesar dicha región. El objetivo del algoritmo de "weighted regions" es el de encontrar una trayectoria desde el punto inicial hasta el punto meta que minimice el costo total que el vehículo tiene que pagar, tal como se observa en la figura 2.2. Regiones difíciles de atravesar presentan costos superiores a las regiones por donde el robot no encuentra mucha dificultad al navegar.

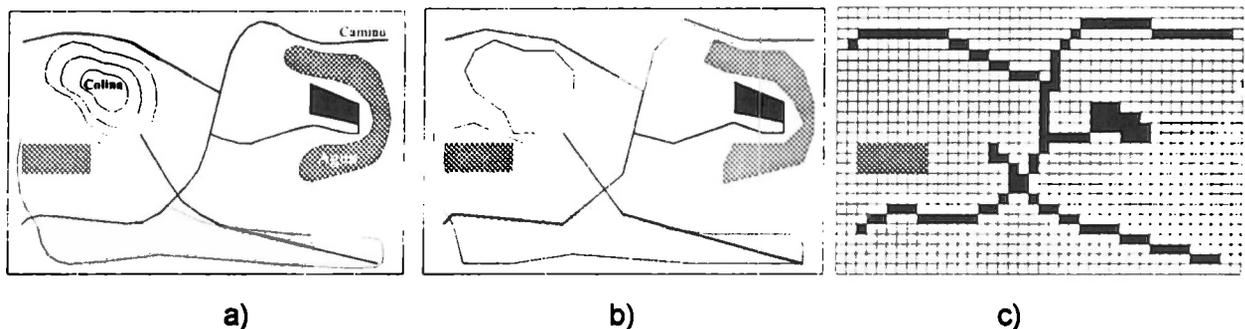


Fig. 2.2 Determinación de las regiones pesadas conforme la técnica de Mitchell. a) Mapa del terreno, b) Asignación de pesos a las diferentes partes del terreno, según su facilidad o dificultad de ser recorrido por un vehículo, c) Discretización de las regiones pesadas de la parte "b", las regiones negras representan las partes del terreno de menor costo.

La última de las metodologías basadas en conceptos matemáticos estudiada en el presente trabajo la cual es utilizada para la navegación de robots móviles esta cimentada en los conceptos de campos potenciales artificiales (artificial potencial fields). Esta técnica fue desarrollada por Khatib [AR-11], con la cual es posible evadir obstáculos en tiempo real. En

dicha metodología el sistema (i.e. brazo manipulador, robot móvil) se mueve en un campo de fuerzas donde el punto meta es un polo de atracción (i.e. para el efector final) mientras que los obstáculos son superficies repulsoras que rechazan las partes que componen el sistema. El campo de fuerzas por donde navega el sistema es construido con base en ecuaciones que representan las fuerzas dentro del ambiente.

TABLA 2.2 Ventajas y desventajas del método basado en regiones pesadas.

REGIONES PESADAS (Weighted Regions)	
Ventajas	Desventajas
<ul style="list-style-type: none"> • Se encuentran trayectorias óptimas de costos mínimos. • Se aplica a cualquier tipo de terreno con o sin obstáculos. • Es posible encontrar trayectorias óptimas para cualquier tipo de sistema (animal, robot, silla de ruedas, etc.). • Si el terreno no es muy complicado el método puede aplicarse para encontrar trayectorias óptimas en tiempo real. 	<ul style="list-style-type: none"> • Las regiones pesadas deben especificarse para cada tipo de vehículo (robot móvil con patas o con ruedas) donde cada tipo de terreno presenta diferentes dificultades (diferentes costos) para cada tipo de sistema (i.e. Automóvil, Tanque). • Dependiendo de la topología del terreno, la búsqueda de la trayectoria que minimiza el costo puede requerir de bastante tiempo. Puede no ser aplicable en tiempo real. • Es necesario conocer con detalle el mapa del terreno en donde el robot navegará. • No es aplicable en ambientes dinámicos. • Es difícil determinar los costos para cada tipo de terreno y para cada tipo de sistema.

Dichas ecuaciones son obtenidas principalmente mediante las ecuaciones dinámicas del sistema y la posición y forma de los obstáculos dentro del terreno. Los detalles para la elaboración de las ecuaciones del campo potencial se describe en [AR-11]. Con esta técnica es posible lograr que todas las partes del sistema evadan los obstáculos adecuadamente, tal como puede observarse en la figura 2.3, donde se muestra la simulación de un brazo mecánico redundante de 4 grados de libertad operando en el plano. Además de campos potenciales estáticos, con esta técnica es posible definir campos potenciales artificiales variantes en el tiempo con lo que pueden evitarse obstáculos (u otros sistemas) en movimiento. Las ventajas y desventajas observadas para éste método (con base en el problema que se desea resolver) son expuestas en la tabla 2.3.

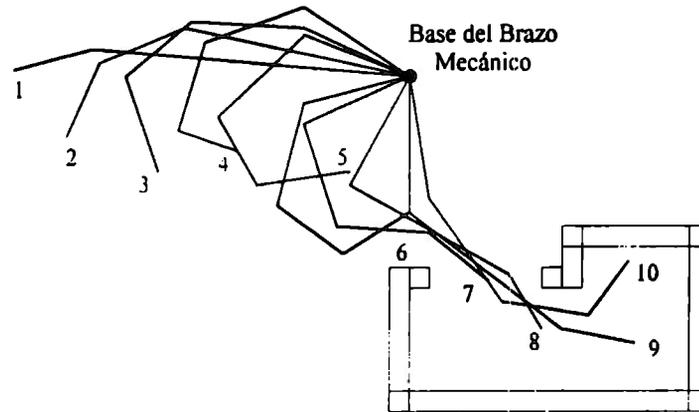


Fig. 2.3 Movimiento en un plano de un brazo mecánico de cuatro grados de libertad utilizando campos potenciales artificiales.

TABLA 2.3 Ventajas y desventajas del método de campos potenciales artificiales.

CAMPOS POTENCIALES ARTIFICIALES (artificial potencial fields)	
Ventajas	Desventajas
<ul style="list-style-type: none"> • Puede ser usado para la navegación en ambientes estáticos y dinámicos, en donde los obstáculos pueden presentar cualquier forma. • Puede ser usado en tiempo real. • Puede ser usado tanto para brazos manipuladores como para robots móviles en tiempo real • Conforme el robot se mueve el campo potencial ($U_0(x)$) puede obtenerse fácilmente en tiempo real. • Pueden generarse varios campos potenciales sobre un mismo ambiente y aplicar el principio de superposición a diversos puntos del sistema, lo que reduce la complejidad de la metodología enormemente. 	<ul style="list-style-type: none"> • Dependiendo del sistema, puede ser difícil obtener las ecuaciones dinámicas que representen de manera adecuada el comportamiento del sistema. • El campo potencial ($U_0(x)$) debe ser diseñado para cumplir las condiciones de estabilidad del sistema. • Para crear el campo potencial ($U_0(x)$) es necesario considerar las fuerzas disipativas del sistema, lo que en ocasiones es sumamente difícil. • El campo potencial ($U_0(x)$) debe ser una función no negativa continua y diferenciable, donde sus derivadas tiendan a ∞ conforme el sistema se aproxime a los obstáculos. Esto es complicado de lograr para robots móviles destinados a trabajar en ambientes desconocidos debido a que no existe un punto de referencia para el sistema de coordenadas a usar. • Se requieren de ecuaciones analíticas para describir los obstáculos. • La complejidad en las tareas que pueden realizarse es limitada, especialmente si el terreno presenta una gran cantidad de obstáculos de formas asimétricas.

2.2.2 TÉCNICAS DE NAVEGACIÓN QUE UTILIZAN MÉTODOS GRÁFICOS

En esta sección se presentan las técnicas de navegación que emplean métodos basados en gráficas y/o diagramas. Este tipo de técnicas obtienen información de mapas (gráficas y/o diagramas) a partir de la cual planean los movimientos del robot. A continuación se exponen tres de los mejores y más populares métodos basados en gráficas.

Algunos investigadores como O'Dunlaing y Yap [AR-31] han utilizado el método de proyección (descrito en la sección 2.2.1) en casos sencillos como en el movimiento de discos moviéndose en un espacio poligonal bidimensional (2-D), donde el espacio reducido de FP (free positions) es tomado como el "Diagrama de Voronoi" asociado con el conjunto de obstáculos presentes en el terreno. Con esto se obtienen algoritmos de movimiento de orden $O(n \log n)$ [AR-31]. Esta técnica de utilizar Diagramas de Voronoi también llamada "mapas de espacio libre" (Free-Space Maps), consiste en crear un diagrama de Voronoi, como el que se muestra en la figura 2.4b mediante el cual un robot es capaz de recorrer ambientes con la ayuda de sensores. A medida que el robot se desplaza, la ruta por la que viaja es registrada en una gráfica de espacio, cuyos nodos son puntos llamados de "stop", puntos en los cuales el robot se detuvo a sensar el ambiente y cuyos arcos representan las trayectorias de un punto de stop a otro entre los cuales no existen obstáculos. A partir de este grafo (ver figura 2.4b) se construye el diagrama de Voronoi asociado al terreno en el cual el robot se desempeñará. Obviamente entre más puntos de stop genere el robot conforme realiza una exploración previa del terreno, más complicado pero a la vez será más exacto será el mapa generado por el robot. Esta técnica fue utilizada exitosamente en el robot HILARE [AR-32], el cual únicamente navegaba a través de ambientes de oficinas (cuartos, pasillos, etc.).

Otra de las técnicas de navegación basada en grafos es la técnica desarrollada por Lozano-Pérez y Wesley [AR-8]. En [AR-8] Lozano-Pérez y Wesley describen un algoritmo para evadir obstáculos, basado en gráficas de visibilidad (visibility graphs) conocido por algunos autores [AR-20] como un algoritmo de movimiento óptimo. Dicho algoritmo planea trayectorias libres de colisiones de cuerpos poliedricos (i.e. robot) a través de otros objetos también de formas poliedricas (figura 2.5). El algoritmo desarrollado consiste en aumentar el volumen de los obstáculos al tiempo que disminuye el volumen del robot (hasta llegar a ser un robot puntual) para después encontrar la trayectoria óptima que llevará al robot desde su posición inicial a una posición final mediante la búsqueda, ya sea utilizando el algoritmo de

Dijkstra o el algoritmo A*, sobre una malla (o grafo) llamada gráfica de visibilidad (visibility graph) generada a partir de los vértices de los objetos poliedricos (3-D) o poligonales (2-D) que forman el ambiente por el cual el robot navegara, tal como puede observarse en la figura 2.5a.

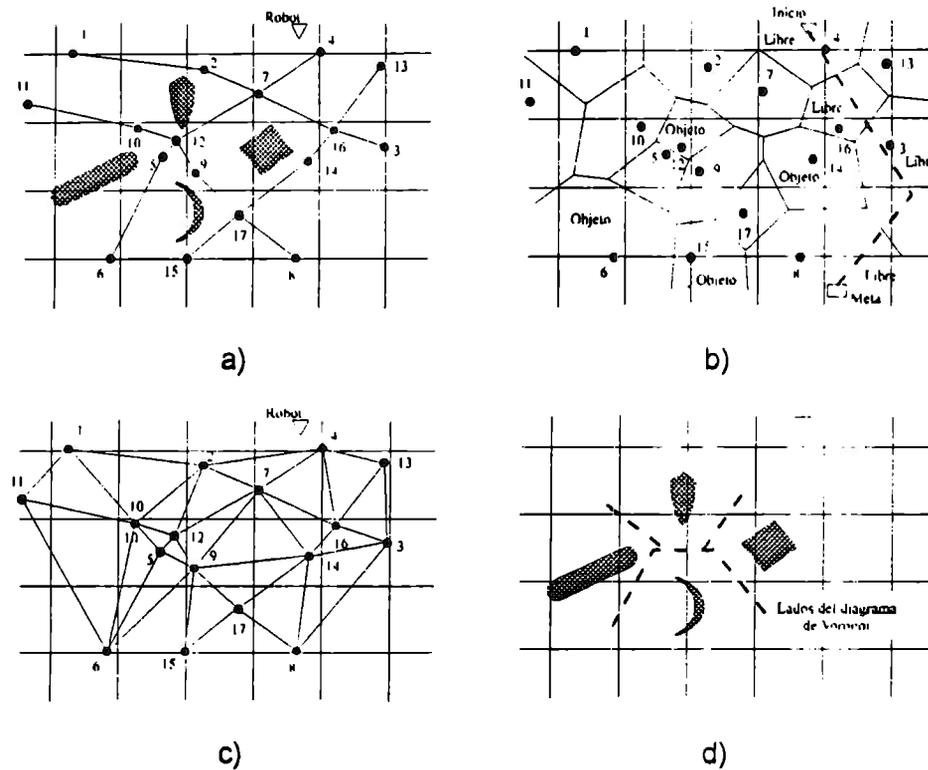


Fig. 2.4 a) Gráfica espacial en la que se muestran 4 trayectorias recorridas por el robot al estudiar el entorno, los nodos son puntos en los cuales el robot se detuvo a sentir el ambiente, b) Diagrama de Voronoi obtenido a partir del terreno de la figura " a" en donde se muestra una trayectoria óptima, c) Triangulación de Delaunay obtenido a partir del diagrama de Voronoi de la figura " b" , d) Detección de objetos mediante el uso de los diagramas de Voronoi y triangulación de Delaunay.

Las figuras 2.5a, 2.5b y 2.5c muestran como un objeto R (robot) es llevado desde su posición inicial (I) hasta una posición final o meta (M) evadiendo los obstáculos (objetos sombreados). Esta técnica fue utilizada en Shakey [AR-26], un robot móvil desarrollado en la Universidad de Stanford, con el cual se ha estudiado experimentalmente el uso de esta metodología la cual también es descrita con un cierto nivel de detalle en [AR-27].

TABLA 2.4 Ventajas y desventajas del método basado en mapas de espacio libre.

MAPAS DE ESPACIO LIBRE (uso de diagramas de Voronoi)	
Ventajas	Desventajas
<ul style="list-style-type: none"> • Los robots móviles pueden navegar a través de un terreno desconocido mediante el uso de sensores sin la necesidad de un mapa detallado del terreno. • El mapa creado a partir de una exploración previa del terreno es representado en forma de grafo o malla. • Para el caso 2-D el algoritmo es bastante eficiente. • Después de que robot ha generado un mapa preliminar del terreno el método puede ser usado en tiempo real (únicamente para ambientes en 2-D). 	<ul style="list-style-type: none"> • Requiere de mapas (aunque no muy detallados) del terreno. • La técnica asume que todos los obstáculos son de formas convexas. • El robot tiene que generar un mapa (no detallado) del terreno mediante una exploración previa. • Por lo general el mapa generado por el robot presenta redundancias, lo que requiere de la implementación de otros algoritmos para eliminarlas. • Es sumamente complicado aplicarlo a casos en 3-D. • No es fácil aplicarlo en ambientes dinámicos puesto que el robot tiene que generar mapas preliminares del terreno mediante exploraciones previas del mismo. • Es casi imposible aplicarlo en tiempo real en ambientes dinámicos.

Mediante el uso de esta técnica se han podido desarrollar algoritmos para aplicaciones tanto en 2-D como en 3-D, donde para el caso de 2-D se ha conseguido implementar algoritmos en donde la trayectoria más corta es obtenida en tiempo $O(n^2 \log n)$ [AR-34] e incluso en tiempo $O(n^2)$ [AR-34], [AR-35]. Para el caso de 3-D la situación se complica por lo que hasta la fecha para el caso general de ambientes con objetos poliedricos, únicamente algoritmos de tiempo exponencial han sido obtenidos, y aún no se sabe con exactitud si este caso es realmente un problema intratable (según [AR-20]). Finalmente en cuanto a esta técnica, algunos investigadores como Baker [AR-37], Chew [AR-38] y Mitchell [AR-39], han aplicado técnicas similares para desarrollar algoritmos que encuentran las trayectorias más cortas en ambientes con obstáculos circulares y elípticos.

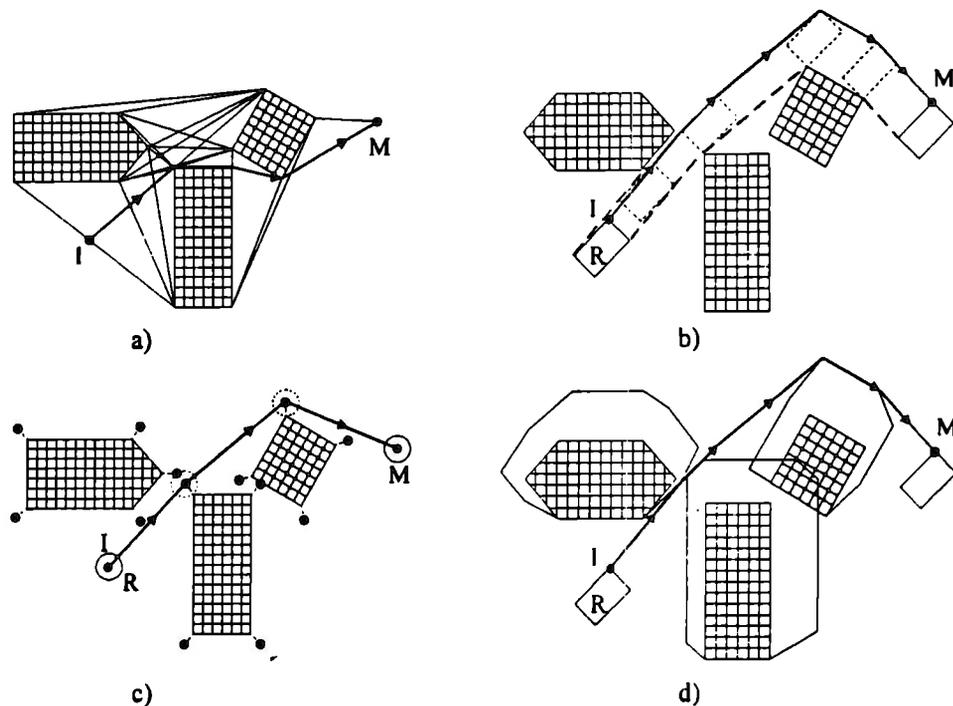


Fig. 2.5 Representación geométrica del algoritmo de gráficas de visibilidad desarrollado por Lozano-Perez y M. A. Wesley en el laboratorio de inteligencia artificial del Instituto Tecnológico de Massachusetts en 1979 para la navegación de robots móviles [AR-8]. a) Robot puntual, b) c) y d) Robot con volumen en donde se observa el incremento en volumen de los obstáculos y la disminución del robot a un punto. En c) y d) se presentan dos formas diferentes en las que se puede aumentar el volumen de los obstáculos.

Finalmente otra técnica que emplea mapas del terreno es la navegación por marcas (landmark navigation) [TX-2], donde la navegación se lleva a cabo con base en marcas presentes en el terreno (i.e. sensores infrarrojos montados sobre las paredes y techos), las cuales son fácilmente reconocidas por el robot. Con las marcas presentes en el terreno, el sistema móvil es capaz de definir trayectorias (en ocasiones almacenadas en su memoria) por las cuales se moverá sin sufrir daño alguno. Esta técnica usada en robots como Sencar AGV [AR-23] y MELDOG [AR-24] consiste en almacenar en la memoria del robot rutas o trayectorias en forma de listas ligadas de segmentos de marcas sobre el terreno o como árboles de planeación tal como puede verse gráficamente en la figura 2.6. Obviamente para poder utilizar éste enfoque, el terreno o ambiente de trabajo debe ser conocido a priori por el robot. Esta técnica es en la actualidad una de las más utilizadas en la industria por sus altos niveles de eficiencia y confiabilidad.

TABLA 2.5 Ventajas y desventajas del método de gráficas de visibilidad.

GRÁFICAS DE VISIBILIDAD (Visibility Graphs)	
Ventajas	Desventajas
<ul style="list-style-type: none"> • Este algoritmo es capaz de encontrar trayectorias óptimas tanto en dos como en tres dimensiones. • La trayectoria generada esta compuesta de segmentos de líneas rectas. • El objeto a mover es considerado como un punto lo cual simplifica la técnica, sobre todo cuando el objeto a mover es mucho mayor (en dimensiones) con respecto a los obstáculos presentes en el terreno. • El algoritmo es capaz de considerar las dimensiones (volumen) del robot. • El algoritmo es capaz de considerar la rotación del objeto a mover (robot). 	<ul style="list-style-type: none"> • No puede utilizarse en ambientes dinámicos. • Se aplica únicamente en terrenos donde los obstáculos presentan formas poliedricas conexas (3-D) o formas poligonales convexas (2-D) y por lo tanto el robot esta restringido a moverse en ambientes controlados. • Es necesario conocer con anticipación el terreno donde se desempeñara el robot (mapas). • El tiempo de búsqueda sobre la malla generada por este método crece significativamente conforme aumenta el número de obstáculos presentes en el terreno. • No puede emplearse para cuando el terreno es desconocido. • Cuando las dimensiones del objeto a mover (robot) son relativamente mayores a las dimensiones de los objetos presentes en el terreno, el algoritmo presenta problemas. • Se requiere de bastante tiempo para construir la gráfica de visibilidad necesaria para buscar la trayectoria óptima.

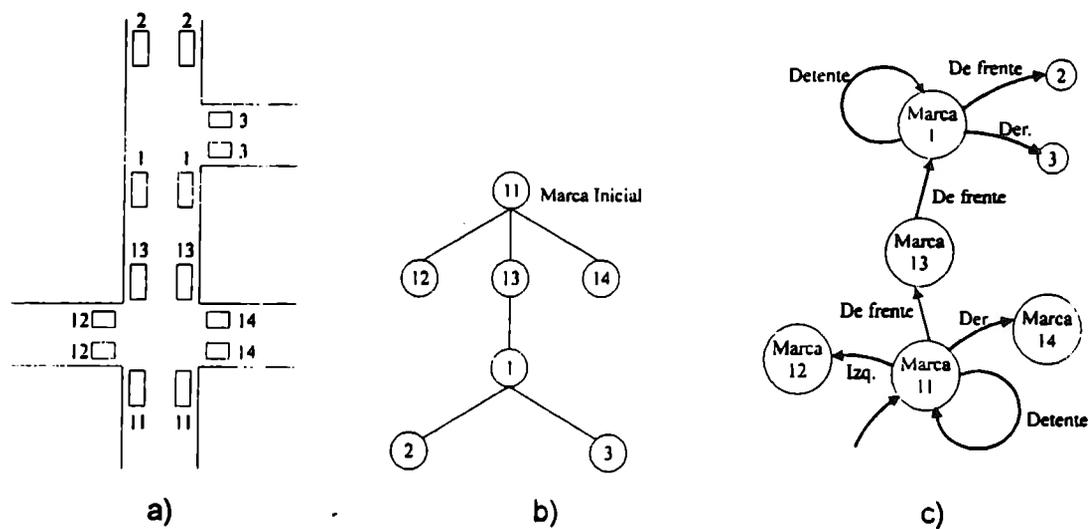


Fig. 2.6 Navegación por marcas. a) Terreno marcado por sensores, b) representación de rutas en forma de listas, c) Representación de trayectorias en forma de árbol de planeación.

TABLA 2.6 Ventajas y desventajas de la navegación por marcas.

NAVEGACIÓN POR MARCAS (Landmak Navigation)	
Ventajas	Desventajas
<ul style="list-style-type: none"> • Es fácil representar el ambiente por el cual se moverá el robot mediante árboles de planeación y ser usados como máquinas de estados con lo que se puede controlar fácilmente la ejecución de la ruta a seguir por el robot. • Es un método eficiente para cuando el robot tiene un número pequeño de rutas claramente definidas. • Por medio de esta técnica es posible simplificar el mapa del terreno eliminando información geográfica innecesaria. • Puede ser utilizado en tiempo real. 	<ul style="list-style-type: none"> • Utiliza mapas del terreno por el cual navegará el robot, por lo que el terreno debe ser conocido con anticipación. No puede emplearse para cuando el terreno es desconocido • La información almacenada en la memoria del robot (mapa del terreno) tiene tamaños fijos, por lo que el robot esta restringido a moverse dentro de dichos limites. • La forma de almacenar la información dentro de la memoria del robot hace difícil realizar cambios de dicha información. • El robot esta restringido a moverse en ambientes restringidos (fabricas, bodegas, etc.). • deben emplearse sensores tanto en el robot como sobre el terreno, lo cual incrementa el costo del sistema. • No puede ser utilizado en ambientes dinámicos

2.2.3 OTRAS TÉCNICAS DE NAVEGACIÓN

El tercer y último grupo de técnicas de navegación dentro del presente trabajo contiene técnicas que no pertenecen a ninguno de los dos grupos descritos anteriormente. Aquí se presentan técnicas como son las redes neuronales, la lógica difusa y las técnicas híbridas. Otro conjunto de metodologías para la navegación de robots móviles utilizadas para cuando el ambiente en el cual se desempeñara el robot no es conocido a priori (que es precisamente el caso que se pretende resolver) son técnicas conocidas como "planeación exploratoria de movimientos y trayectorias" las cuales varían según el tipo y características de los sensores con los que cuenta el robot, pudiendo ser sensores táctiles, de proximidad o de visión. El uso de estas metodologías requiere que el robot cuente con los sensores necesarios para explorar su entorno.

Las redes neuronales, una de estas técnicas exploratorias, ha sido utilizada exitosamente para la navegación de vehículos autónomos, y en espacial empleada con magníficos resultados por la Universidad de Carnegie Mellon en su muy conocido vehículo autónomo Navlab [AR-3]. Esta técnica está basada en el uso de redes neuronales para el reconocimiento del terreno (determinación de obstáculos y posibles trayectorias) el cual es sentido por medio de dispositivos sensoriales (i.e. cámaras de vídeo) colocados en el robot. La principal función de las redes neuronales dentro de este tipo de metodologías, es el de reconocer ciertos patrones (i.e. obstáculos) dentro del terreno y pasar dicha información al sistema de control del robot, para que éste cambie su dirección, velocidad, aceleración etc. a fin de evitar los percances del terreno en tiempo real. En general las redes neuronales emulan el comportamiento biológico de las neuronas además que son capaces de aprender a controlar de la mejor manera cualquier sistema sin la necesidad de conocer su modelo analítico. Esto es realizado mediante el ajuste continuo de los parámetros que definen la red neuronal (i.e. valor de las conexiones entre neuronas) con base en las respuestas generadas del sistema. Las ventajas y desventajas observadas para éste esquema de control son expuestas en la tabla 2.7.

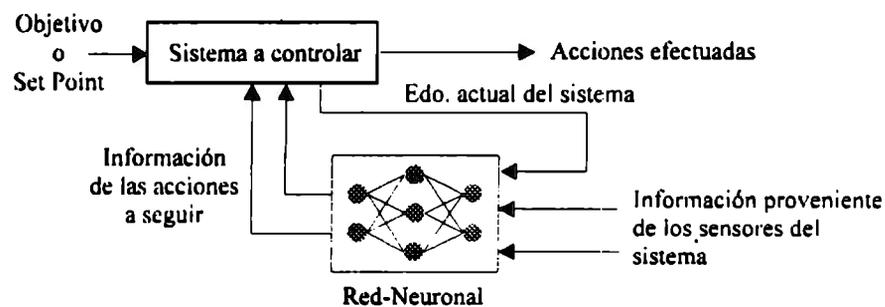


Fig. 2.7 Esquema de un sistema controlado por medio de redes neuronales.

Otro de los nuevos y más recientes métodos desarrollados para la navegación en ambientes desconocidos en tiempo real, es la utilización de lógica difusa. Esta técnica ha sido propuesta por Beaufrère y Zeghloul [AR-21], [AR-22], pero también ha sido desarrollada y utilizada por Hartmut, Jörg y Peters [AR-68], [AR-69]. Esta metodología se basa esencialmente en la utilización de sensores (i.e. de ultrasonido, de luz infrarroja, de visión, etc.) colocados en el robot, en la definición de conjuntos difusos y en la definición de una serie de reglas (utilizando conjuntos difusos). Estos tres elementos son usados por el sistema (i.e. robot móvil) para generar la nueva trayectoria a seguir conforme se desplaza por el terreno a fin de evitar los obstáculos localizados, al mismo tiempo que cumple su objetivo

(punto meta). La figura 2.8 muestra de manera esquemática el sistema de navegación basado en lógica difusa desarrollado en Alemania por Hartmut, Jörg y Peters [AR-68] para el robot MORIA [AR-69]. La razón principal de emplear lógica difusa para la navegación y planeación de trayectorias de sistemas autónomos, es la de generar soluciones adecuadas a partir de información vaga, imprecisa y ruidosa proveniente de los sensores. Las ventajas y desventajas de la lógica difusa empleada para la navegación de robots móviles se resumen en la tabla 2.8.

TABLA 2.7 Ventajas y desventajas del uso de redes neuronales en la navegación de robots.

REDES NEURONALES	
Ventajas	Desventajas
<ul style="list-style-type: none"> • Dado la naturaleza paralela de las redes neuronales, esta técnica puede utilizarse en tiempo real (una vez entrenada adecuadamente la red neuronal). • Existen gran cantidad de paquetes computacionales con los cuales es posible entrenar, modificar y analizar la arquitectura de la red neuronal construida para resolver el problema planteado. • Una vez entrenada la red neuronal es posible la construcción de la red en hardware con lo cual el sistema de control y navegación estaría totalmente integrado en el robot sin la necesidad del uso de un sistema complicado de computo. • Las redes neuronales son capaces de aprender gran cantidad de situaciones al que se puede enfrentar el sistema de control. • No es necesario conocer el modelo matemático que represente el sistema a controlar. Basta tener información experimental. 	<ul style="list-style-type: none"> • Es difícil y consume mucho tiempo entrenar y definir adecuadamente la topología y arquitectura de la red neuronal. • La red neuronal entrenada únicamente es útil para ambientes para los cuales fue entrenada (oficinas, bodegas, etc.). • La red neuronal es una pequeña parte de todo el sistema de navegación y control del robot.

Finalmente, también existen otro tipo de técnicas de navegación llamadas adaptivas las cuales consisten en adaptar y modificar el algoritmo de navegación conforme el espacio por el que se mueve el robot se modifica o incluso el mismo sistema robótico cambia (i.e. cuando el robot toma un objeto para manipularlo y cambiarlo de posición). Estos procedimientos son capaces de adaptarse a situaciones en donde el ambiente por el que se moverá el robot presenta objetos dinámicos (i.e. otros robots o personas en movimiento cuya

trayectoria puede o no ser conocida). Este tipo de técnicas de navegación son técnicas hasta ahora poco estudiadas [AR-21], [AR-36] pero que día con día se utilizan más debido a que los sistemas a controlar son cada vez más complejos y requieren de mejores sistemas de control capaces de actuar satisfactoriamente conforme a cambios externos al sistema.

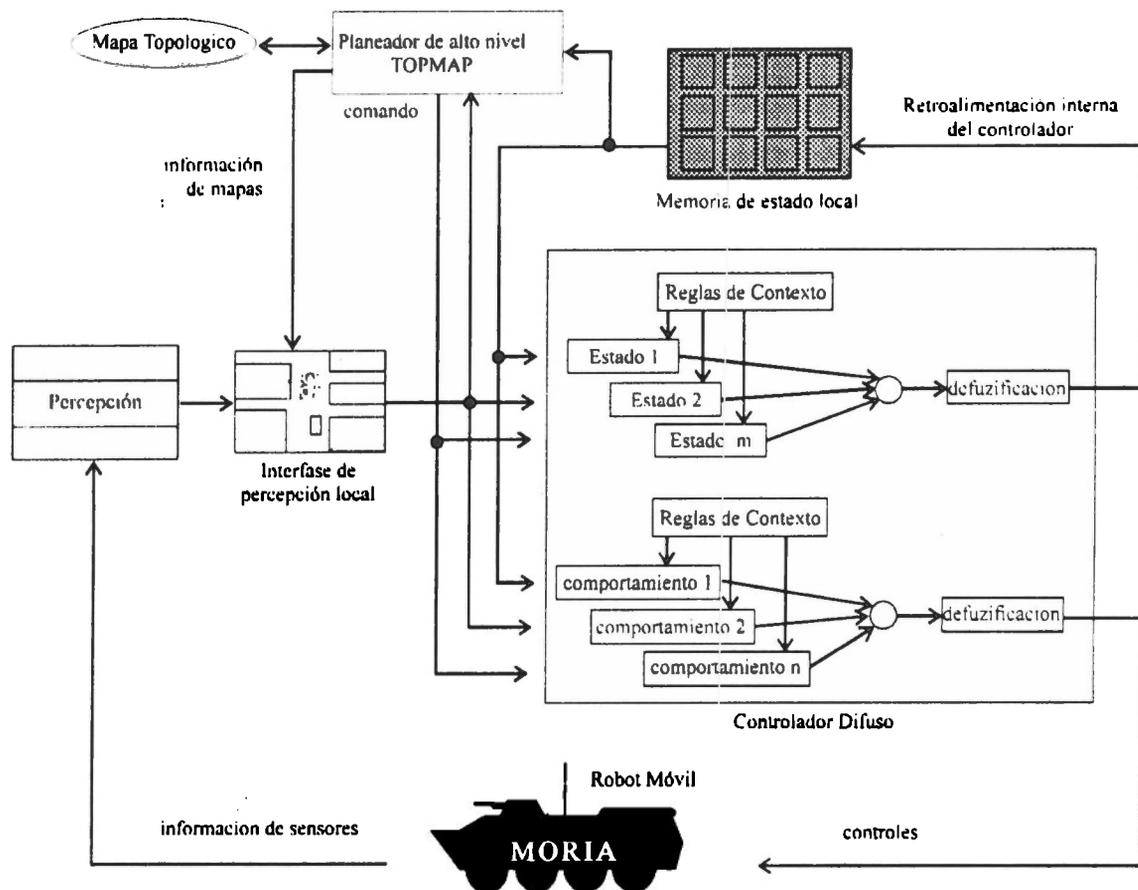


Fig. 2.8 Sistema de navegación basado en lógica difusa para el robot MORIA. Este esquema en particular emplea un mapa del terreno por el cual MORIA navegará.

Además de los métodos para la navegación de robots móviles autónomos descritos anteriormente, en la actualidad se siguen mejorando y desarrollando nuevos métodos y dispositivos (i.e. nuevos sensores) para resolver el complejo y difícil problema de navegación. Tan solo basta observar la gran cantidad de artículos publicados recientemente ([AR-40] - [AR-69], [AR-78] - [AR-81]) dentro de los cuales se presentan una gran diversidad de enfoques que tratan de dar solución, no solo a la navegación de robots móviles, sino a la navegación y planeación de trayectorias de todo tipo de sistemas autónomos (i.e. brazos manipuladores). A partir de estos nuevos enfoques se han desarrollado métodos híbridos [TX-23] y se han utilizado nuevas metodologías como los planeadores basados en algoritmos

genéticos [AR-65], [AR-66], el reconocimiento de patrones mediante redes neuronales difusas, por mencionar solo algunos ejemplos.

TABLA 2.8 Ventajas y desventajas del uso de lógica difusa en el problema de navegación.

LÓGICA DIFUSA (Fuzzy Logic)	
Ventajas	Desventajas
<ul style="list-style-type: none"> • Es capaz de manejar información vaga, imprecisa e incompleta sobre el ambiente que rodea al robot. • Puede utilizarse en tiempo real. • Puede ser usado en ambientes estáticos y dinámicos. • Existen varios paquetes computacionales con los cuales es posible simular y ajustar el sistema diseñado. • Es fácil añadir nuevas reglas a la base de conocimiento con la que el sistema realiza sus inferencias. • Es posible cambiar la configuración y enfoque del problema a resolver. • Es aplicable a una gran gama de situaciones. • El sistema puede utilizar gran variedad de dispositivos para sensor el ambiente que le rodea. • Pueden utilizarse gran variedad de conjuntos difusos y variables lingüísticas con las cuales el sistema controlador puede tomar varios enfoques y características propias. • No se requiere conocer el modelo analítico del sistema a controlar. Basta con saber controlar el sistema manualmente. 	<ul style="list-style-type: none"> • Es difícil establecer la base de conocimientos con las reglas necesarias y suficientes para que el sistema funcione adecuadamente. • Ajustar el sistema es un proceso lento y tardado. • Al finalizar el diseño del controlador difuso no se sabe si la solución proporcionada por el controlador es óptima. • Hasta el momento únicamente se ha demostrado que esta técnica genera resultados satisfactorios para cuando el número de reglas necesarias para controlar el sistema es pequeño (menor a 100).

Las técnicas híbridas (i.e. redes neuronales difusas) constan de dos o varios métodos trabajando conjuntamente para solucionar problemas específicos. Los métodos que componen el sistema híbrido se complementan entre sí para resolver problemas más complejos que los que pueden ser resueltos por las técnicas individuales que los integran.

TABLA 2.9 Ventajas y desventajas del empleo de técnicas híbridas para la navegación.

TÉCNICAS HÍBRIDAS	
Ventajas	Desventajas
<ul style="list-style-type: none">• Resuelven problemas de mayor complejidad que los problemas que pueden ser resueltos por las técnicas que los integran.• Se sabe que pueden generar mejores soluciones que las obtenidas por técnicas individuales.	<ul style="list-style-type: none">• Es difícil saber cuales son los métodos que deben cooperar entre sí para resolver algún problema de manera óptima.• No se conoce mucho sobre estas técnicas.• Estas técnicas únicamente han sido empleadas para casos específicos.

2.3 COMENTARIOS SOBRE LAS TÉCNICAS ESTUDIADAS

En el presente capítulo se estudiaron las técnicas de navegación más importantes y utilizadas hoy en día dentro de la robótica. Se generaron tablas en donde se aprecia las ventajas y desventajas de cada método con base en las dos características deseadas en el sistema de navegación a desarrollar mencionadas al principio de la sección 2.2.

- El sistema de navegación a diseñar debe ser capaz de trabajar satisfactoriamente bajo ambientes desconocidos a priori por el robot.
- El sistema de navegación a diseñar debe ser capaz de trabajar en tiempo real.

A partir de dichas tablas se observa que los métodos que hemos clasificado dentro de los dos primeros grupos (métodos que emplean conceptos matemáticos y métodos que utilizan gráficas) no son adecuados para resolver nuestro problema. Esto se debe principalmente a que dichas técnicas requieren información previa sobre el terreno en donde trabajara el robot. El sistema de navegación que se desea debe tener la capacidad de navegar en ambientes totalmente desconocidos a priori por el robot, es decir, el robot debe ser capaz de navegar satisfactoriamente sin ninguna información previa del terreno. Esta característica es satisfecha por las metodologías clasificadas dentro del tercer grupo (técnicas que no emplean conceptos matemáticos ni gráficas para caracterizar el terreno). En especial la lógica difusa una de las técnicas clasificadas dentro del tercer grupo además de no requerir de mapas, también puede ser utilizada en tiempo real y no requiere de modelos exactos (i.e. matemáticos, geométricos) del sistema a controlar. Esto es de especial

interés para nuestra aplicación dado que un robot móvil es un sistema muy complejo por los que es muy difícil obtener una representación exacta de su funcionamiento.

Por otra parte las técnicas híbridas son técnicas que requieren ser estudiadas por más tiempo para poder determinar de mejor manera que combinación de metodologías es la mejor de tal forma que produzcan los mejores resultados para el problema que se desee resolver.

2.4 METODOLOGÍA A USAR

Dadas las características (ventajas y desventajas) de cada uno de los métodos estudiados en el presente capítulo, y con base en el problema de lograr que un robot móvil navegue de manera autónoma a través de ambientes desconocidos, se ha decidido emplear lógica difusa (Fuzzy Logic) de manera similar a la metodología empleada recientemente por Beaufrère y Zegloul [AR-21], [AR-22].

Además de emplear lógica difusa, también se seguirá la metodología de McKrrow [TX-2] el cual ha descompuesto la navegación de robots móviles en tres sub-tareas a saber: mapeo y modelado del ambiente, planeación y selección de la ruta y finalmente atravesar la ruta y evadir obstáculos. En las dos últimas etapas es en donde se empleara lógica difusa, como se describirá en los siguientes capítulos, mientras que la primera etapa será realizada mediante sensores de ultrasonido (cuya justificación del porque de su empleo será descrita en los siguientes capítulos) localizados en la parte frontal del robot con los cuales el sistema obtendrá la información necesaria de su entorno para poder navegar de manera segura sin sufrir colisiones.

Una de las razones por la cual se decidió utilizar lógica difusa para resolver nuestro problema, es que para que el robot pueda planear acciones con el objetivo de navegar a través de un ambiente desconocido, éste debe crear un modelo interno del ambiente que le rodea con base en información incompleta, vaga e imprecisa suministrada por dispositivos sensoriales. El ambiente que rodea al robot debe ser percibido por una serie de sensores (i.e. ultrasonido, infrarrojos, de contacto, etc.) que inherentemente por diversos factores (de fabricación, de diseño, de calibración, desajustes, etc.) proporcionan este tipo de información (i.e. información incompleta y ruidosa). La lógica difusa es capaz de manipular este tipo de información con lo que el robot podrá percibir de manera muy próxima a la realidad el mundo que le rodea y así generar acciones adecuadas.

Otra de las razones por las que se decidió emplear lógica difusa, es de que no es necesario conocer el modelo matemático del sistema a controlar. Para el caso de robots móviles dicho modelo es muy complejo y es muy difícil de obtener de forma exacta.

En el siguiente capítulo se proporcionarán los conceptos básicos referentes a la lógica difusa (Fuzzy Logic) necesarios para poder diseñar un controlador de navegación difuso y se proporcionarán otras razones por las que se decidió emplear esta técnica.

3 LÓGICA DIFUSA (FUZZY LOGIC)

El presente capítulo tiene como finalidad introducir al lector (que no este familiarizado) en la teoría de la lógica difusa. Los conceptos básicos aquí presentados son la base necesaria para el diseño de cualquier controlador difuso. En particular para un controlador difuso que controle la navegación de robots móviles en ambientes desconocidos. En este capítulo únicamente se expondrá un panorama general de la teoría de la lógica difusa con la cual el lector comprenderá sin ningún problema los siguientes capítulos en donde se elabora el sistema de navegación antes mencionado.

A continuación se explorara y describirán los aspectos más importantes referentes a la lógica difusa, metodología que esta cobrando popularidad para la manipulación e interpretación de conceptos ambiguos mediante el uso de las computadoras [TX-1], [TX-15]. Recordar que esta técnica fue seleccionada en el capítulo anterior para solucionar el problema de lograr que un robot móvil autónomo navegue en ambientes totalmente desconocidos mediante la manipulación de información imprecisa, incompleta y ruidosa (noisy data) proveniente de sus sensores.

3.1 INTRODUCCIÓN

Los expertos, en la mayoría de las áreas confían en su sentido común para dar solución a muchos de los problemas encontrados en su rama o campo de trabajo. Este tipo de razonamientos basado en el sentido común y en el nivel de conocimiento que posee un experto acerca de algún problema en particular, es generalmente expresado en términos

vagos o ambiguos. Por ejemplo, un experto en robótica podría establecer la siguiente frase difusa:

Cuando la velocidad de navegación de un robot móvil es rápida la cantidad de energía que consume es mucha (Cuanto es rápido ?, Cuanto es mucha ?).

Desde que el ser humano nace, se acostumbra a escuchar oraciones expresadas en esta forma (ambigua) y por lo común no tiene ninguna dificultad en interpretar el uso de términos ambiguos. Sin embargo, las computadoras una de las herramientas de mayor utilidad hoy en día, no poseen el nivel de razonamiento encontrado en el ser humano, por lo que hacer que las computadoras lo posean es hoy en día un enorme reto. En la actualidad metodologías como la lógica difusa son utilizadas para resolver este problema (hacer que las computadoras razonen o infieran cosas), que de alguna u otra forma son capaces de representar y manipular términos expresados en forma vaga mediante una computadora.

3.2 GENERALIDADES DE LA LÓGICA DIFUSA

Los sistemas difusos han comenzado a estudiarse desde la década de 1920, cuando fueron propuestos por primera vez por Lukasiewicz [TX-25]. Lukasiewicz estudio la representación matemática de algunos términos difusos tales como *alto*, *joven*, *caliente*, *grande*, etc., (solo por mencionar algunos) los cuales desafiaban la lógica de Aristóteles en la cual todo se representaba mediante dos únicos valores de cierto-o-falso $[0, 1]$. Lukasiewicz desarrollo un sistema lógico que extendía el rango en los valores de verdad hacia todos los números reales entre 0 y 1. Dentro de esta nueva lógica se utiliza algún número real entre 0 y 1 para representar la *posibilidad* de que alguna proposición fuera falsa o verdadera. Uno de los ejemplos típicos con los que se demuestra esta lógica es el siguiente:

La posibilidad de que una persona de 1.80 mts. de altura sea realmente alta, puede establecerse mediante una posibilidad de 0.9, lo cual es bastante probable (el valor de 0.9 deja la posibilidad de que dicha persona no sea realmente alta).

Las investigaciones realizadas en tono a esta nueva teoría llevo a una técnica formal de razonamiento inexacto llamada "**teoría de la posibilidad**" (possibility theory). Recientemente este tipo de razonamiento a resurgido con nuevos auges debido a los trabajos del profesor Zadeh (profesor de la Universidad de California-Berkeley), considerado actualmente por muchos investigadores como el padre de la lógica difusa. Zadeh extendió los trabajos de la teoría de la posibilidad desarrollada por Lukasiewicz en una nueva forma de sistemas tratados mediante la lógica matemática. Este nuevo enfoque permite manipular y trabajar con los términos difusos encontrados en el vocabulario humano. A esta nueva herramienta lógica con la cual es posible representar y manipular términos difusos, se le dio el nombre de "**lógica difusa**" (definición 3.1) y la cual a comenzado a adquirir gran popularidad como herramienta para solucionar una gran gama de problemas que difícilmente son resueltos mediante técnicas clásicas (i.e. navegación de robots móviles en ambientes desconocidos).

Definición 3.1: *La "Lógica Difusa" es una rama de la lógica la cual utiliza grados de pertenencia para las variables dentro de los conjuntos en vez de grados de pertenencia únicamente de falso / verdadero.*

Definición propuesta por John Durkin, profesor de la Universidad de Akron [TX-22].

3.3 VARIABLES LINGÜÍSTICAS

La lógica difusa esta principalmente enfocada en cuantificar y razonar acerca de términos vagos, imprecisos o difusos que son usados de manera frecuente en el lenguaje natural, los cuales dentro de esta nueva lógica son llamados "**variables lingüísticas**" (Definición 3.2) o variables difusas. El poder de la lógica difusa recae principalmente en que utiliza variables lingüísticas en vez de variables cuantitativas para representar conceptos imprecisos.

Definición 3.2 : *Las variables lingüísticas son términos usados dentro del lenguaje natural del ser humano para describir algún concepto que usualmente presenta valores vagos o difusos.*

Definición propuesta por Timothy J. Ross [TX-1].

Dentro de los sistemas expertos difusos, las variables lingüísticas son utilizadas para generar reglas difusas. Una regla difusa infiere información acerca de una variable lingüística contenida en su conclusión a partir de la información de otra variable contenida en su premisa. Al rango de posibles valores que puede adoptar una variable lingüística se le conoce como el "universo de cobertura" de dicha variable. Para el caso de la navegación de un robot móvil autónomo estos conceptos podrían presentarse en la siguiente regla:

Regla 1 IF La distancia del obstáculo es cerca *Variable lingüística*
 AND El ángulo del obstáculo relativo al robot es casi cero *valor lingüístico "0"*
 AND El ángulo del obstáculo relativo al robot es hacia derecha del robot
 THEN Gira las llantas del robot ligeramente hacia la izquierda

En este ejemplo, dentro de la condición "IF" la variable lingüística utilizada es "distancia" mientras que su valor lingüístico es "cerca". Por otra parte, supóngase que dentro de la misma condición "IF" a la variable lingüística "distancia" se le ha asignado un rango o universo de cobertura de 0 a 500 [cm]. En estas condiciones la frase "La distancia del obstáculo es cerca" ocupa una sección dentro del universo de cobertura de la variable y por lo tanto es un "**conjunto difuso**". Finalmente, la oración contenida dentro del comando "THEN" es la conclusión y las oraciones contenidas dentro de los comandos "IF" y "AND" son las premisas (las premisas junto con la conclusión forman la regla).

3.4 CONJUNTOS DIFUSOS

Tradicionalmente la teoría de conjuntos estudia y observa al universo como si fuera blanco o negro, verdadero o falso, frío o caliente, es decir, la teoría clásica de conjuntos clasifica a los objetos de tal forma que el objeto está o no dentro de un cierto conjunto (falso o verdadero). Por su parte, la lógica difusa provee una interpretación más razonable de la pertenencia o no de un objeto mediante "**conjuntos difusos**" (Definición 3.3). Los conjuntos difusos son una extensión de la lógica tradicional que sirven para asignar un valor de pertenencia entre 0 y 1 a un objeto, el cual refleja de manera más adecuada el grado de pertenencia de dicho objeto o variable dentro de un conjunto dado. Para entender estos conceptos, observe la "Regla 1" propuesta anteriormente. Para el caso de la variable lingüística "distancia", Qué valores se

podrían considerar como cerca ?. Obviamente se podrían proporcionar varios valores tales como 0.5 mts., 0.47 mts. ó 0.78 mts., pero para alguna otra persona puede darse el caso de que una distancia de 0.78mts. sea considerada como lejos en vez de cerca. También se puede observar que la clasificación de la distancia (cerca o lejos) puede depender de otras variables como es la velocidad del robot. Si el robot se mueve a una gran velocidad, probablemente una distancia de 1.25mts. sea considerada como cerca para una persona y lejos para otra. Es por estas razones que la lógica difusa utiliza conjuntos difusos para clasificar a una variable dentro de uno u otro conjunto, lo que realiza mediante la asignación de grados de pertenencia. El grado de pertenencia de una variable indica el grado con el cual el valor de dicha variable es clasificada dentro de uno u otro conjunto, pudiendo pertenecer a uno, dos o más conjuntos al mismo tiempo con diferente grado de pertenencia para cada uno de los diferentes conjuntos.

Definición 3.3: Para definir un "conjunto difuso" considere lo siguiente: Suponga que "X" es el universo de cobertura, donde sus elementos están denotados por "x". Un conjunto difuso "A" de "X" es caracterizado por una función de pertenencia la cual asocia cada elemento "x" con un grado de pertenencia en "A".

conjuntos de A
 $\mu_A(x)$
 grado de pertenencia de elemento "x"

Definición propuesta por John Durkin, profesor de la Universidad de Akron [TX-22].

A partir de este razonamiento es que la lógica difusa puede decir (por ejemplo) que una distancia de 1.0mts. es considerada una distancia "lejos" con un grado de pertenencia de 0.8 y al mismo tiempo, considerada una distancia "cerca" con un grado de pertenencia de 0.4. Para el caso que se acaba de ver la "distancia" es la variable lingüística y "cerca" o "lejos" son algunos de sus conjuntos difusos (pudiendo existir más conjuntos difusos dentro de la variable lingüística "distancia" tales como "muy cerca", "muy lejos", "super lejos", "regular", etc.).

A diferencia de la teoría de la probabilidad (la cual se apoya en la asignación de probabilidades a un determinado evento bajo las bases de su frecuencia) la lógica difusa asigna valores al evento o elemento "x" bajo las bases de una "función de pertenencia" μ definida como:

conjunto de A
 $\mu_A(x)$
 universo de cobertura "x"

$$\mu_A(x) : X \rightarrow [0, 1]$$

(3.1)

la cual representa el grado en que una variable o elemento "x" pertenece al conjunto difuso "A" (ecuación 3.2).

$$\mu_A(x) = \text{Grado de Pertenencia } (x \in A) \quad (3.2)$$

Donde el grado de pertenencia esta acotado por la relación: $0 \leq \mu_A(x) \leq 1$

3.5 CREACIÓN DE CONJUNTOS DIFUSOS

Para poder representar los conjuntos difusos en una computadora es necesario definir su función de pertenencia, para lo cual existen varias metodologías basadas en diferentes aproximaciones. Existen técnicas novedosas como la propuesta por Kosko en 1991 [TX-21] la cual esta basada en la utilización de redes neuronales las cuales son entrenadas para formar conjuntos difusos con base en las entradas al sistema. Otra de las técnicas más utilizadas hoy en día para poder representar los conjuntos difusos en una computadora y la cual se empleará durante el presente estudio (para crear los conjuntos difusos los cuales servirán para controlar la navegación de un robot móvil) es la creación de conjuntos difusos mediante la realización de una serie de encuestas aplicadas a un experto en el área que se desee estudiar. Una vez obtenida la información necesaria, los conjuntos difusos son creados mediante la identificación de alguna función de pertenencia que describa su comportamiento, tomando en consideración la variable lingüística a la cual pertenezca.

Para comprender como se crean los conjuntos difusos con base en la descripción anterior, a continuación se mostrara un ejemplo tomado del libro "Expert Systems, Design and Development" del autor John Durkin (1994). Considere el concepto de "personas altas" (para el cual se desea crear un conjunto difuso) concepto con el cual es interrogada una persona (experto) para que proporcione su punto de vista en cuanto a que grado de pertenencia considera a una persona alta dada una cierta estatura (i.e. 1.71 mts). Después de adquirir una serie de respuestas por parte del experto, es posible realizar un conjunto difuso referente a personas altas. Esta función podrá ser utilizada para representar los valores de pertenencia hacia algún individuo en particular de tal forma que esté pertenezca a dicho conjunto. La misma técnica puede ser usada para formar conjuntos difusos referentes a diferentes sub-conjuntos difusos pertenecientes a la misma variable lingüística (en este

caso "estatura") tales como estatura "corta" o "mediana". Con las funciones así obtenidas para una misma variable lingüística, es posible elaborar un diagrama en el cual se representen estas funciones de pertenencia como el esquema mostrado en la figura 3.1.

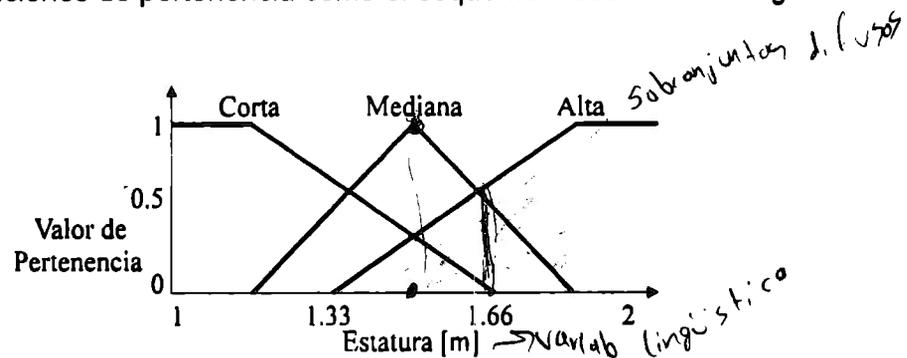


Fig. 3.1 Conjuntos difusos Corta, Mediana y Alta sobre la variable lingüística estatura.

La figura 3.1 muestra los conjuntos difusos representados por medio de funciones lineales para tres diferentes categorías en la altura de algún individuo (pudiendo ser más el número de categorías). Cuando se definen varios conjuntos difusos sobre el mismo universo de cobertura, como en el caso anterior, se dice que existen " *sub-conjuntos difusos*". De la figura 3.1 se observa que un individuo de 1.5 mts. de altura es un miembro de las personas de mediana estatura con un valor de pertenencia de 1, y al mismo tiempo es un miembro de las personas de corta estatura con un valor o grado de pertenencia de 0.25. Con estas observaciones se concluye que un objeto puede ser considerado un miembro parcial de diferentes sub-conjuntos difusos al mismo tiempo (básicamente es aquí donde se encuentra el poder de la lógica difusa, al ser capaz de colocar a un determinado objeto dentro de varios conjuntos al mismo tiempo con lo que captura la ambigüedad de algunos de los términos usados dentro del lenguaje natural).

3.6 REPRESENTACIÓN DE CONJUNTOS DIFUSOS

En esta sección se proporcionará una representación formal de conjuntos difusos, conceptos básicos dentro de lógica difusa. Para lograr esto suponga que se tiene un conjunto difuso "A" definido sobre un cierto universo de cobertura "X". También asuma que se tiene un conjunto discreto de elementos $\{x_1, x_2, \dots, x_n\}$ de "X" sobre el conjunto difuso "A" el cual define la función de pertenencia $\mu_A(x)$ que mapea los elementos x_i de "X" a un grado de pertenencia en $[0,1]$ (ecuación 3.1). Los grados de pertenencia indican el grado en que un

elemento x_i pertenece a "A" (ecuación 3.2). Para un conjunto discreto de elementos, una de las maneras más convenientes de representar un conjunto difuso, es a través del uso de un vector como el que se muestra en la ecuación 3.3.

$$A = (a_1, a_2, \dots, a_n) \quad \text{grado en el que pertenece conj. difuso} \quad (3.3)$$

donde:

$$a_i = \mu_A(x_i) \quad (3.4)$$

Por lo común, y para una clara representación del vector "A", éste incluye el símbolo "/" el cual asocia el valor de pertenencia a_i con su coordenada x_i , tal como se muestra en la ecuación 3.5.

$$A = (a_1/x_1, a_2/x_2, \dots, a_n/x_n) \quad \text{elemento de "X"} \quad \text{el elemento} \quad \text{valor de pertenencia} \quad (3.5)$$

Si retomamos el ejemplo anterior sobre las personas altas, dicho conjunto difuso de personas altas (figura 3.1) estaría representado como Alta = $(0/1), (0.25/1.15), (0.7/1.33), (1/1.66), (1/2)$. La unión de las dimensiones de un vector mediante la notación estándar se efectúa de la siguiente forma:

$$A = \mu_1 / x_1 + \mu_2 / x_2 + \dots + \mu_n / x_n \quad (3.6)$$

o de la forma:

$$A = \sum_{i=1}^n \mu_i / x_i$$

pero si "X" es una función continua, entonces el conjunto "A" puede representarse como:

$$A = \int_x \mu_A(x_i) / x_i \quad (3.7)$$

en cuyo caso es necesario emplear alguna función como la Sigmoid, Gaussiana o Pi para mapear los elementos de "A" a su correspondiente valor de pertenencia. Estas funciones proporcionan una representación adecuada de los sub-conjuntos difusos, sin embargo, al utilizar estas funciones la carga computacional aumenta considerablemente, razón por la cual en la práctica se prefiere utilizar funciones lineales (o segmentos de estas) para representar los conjuntos difusos (tal como aparece en la figura 3.1). Para obtener un ajuste lineal (de

cada conjunto difuso deseado) es posible codificar cada conjunto en un vector-de-ajuste. Por ejemplo, es posible codificar el conjunto difuso "Alto" de la figura 3.1, en el vector (0/1.33, 0.5/1.5, 0.7/1.66, 1/2) mientras que los conjuntos intermedios tales como el conjunto "Mediano" de la misma figura son generalmente representados mediante funciones triangulares (vector-de-ajuste triangular). Para el caso de la figura 3.1 el vector-de-ajuste para el conjunto "Mediano" estaría expresado como (0/1.15, 0.5/1.33, 1/1.5, 0.5/1.66, 0/1.8).

3.7 HEDGES

En las secciones anteriores se describieron las metodologías tanto para capturar como para representar términos lingüísticos vagos e imprecisos de forma cuantitativa mediante el empleo de conjuntos difusos. En conversaciones típicas entre humanos, las personas usualmente añaden nuevos términos vagos en oraciones comunes mediante el empleo de adverbios tales como "muy", "casi", "ligeramente" o "demasiado". Un adverbio es una palabra que modifica los verbos, los adjetivos, otros adverbios e incluso afecta oraciones completas (i.e. Juan es una persona *demasiada* alta).

Los "HEDGES" son una especie de adverbios dentro de la lógica difusa con los cuales es posible formar nuevos conjuntos difusos a partir de conjuntos ya existentes sin que estos sean modificados drásticamente. Los hedges son usados comúnmente para ajustar el sistema de conjuntos difusos. Los hedges modifican matemáticamente los conjuntos difusos tomando en consideración la utilización de adverbios, tal como puede observarse en la figura 3.2, en la cual los conjuntos sobre la variable lingüística estatura son modificados.

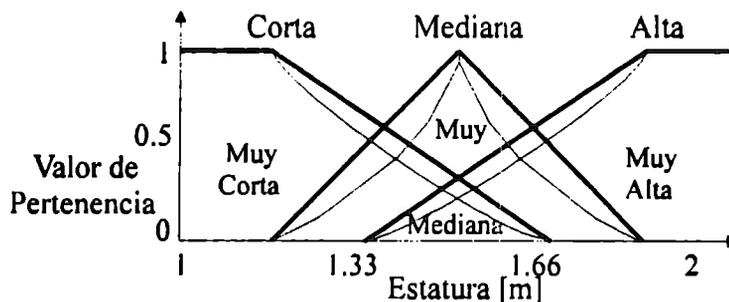


Fig. 3.2 Conjuntos difusos sobre la variable lingüística estatura con el HEDGE "Muy".

Los hedges más utilizados en el diseño de controladores difusos son los de concentración (adverbio muy), de dilatación (adverbio casi) y el de poder (adverbio muy-muy), los cuales se definen como se describe a continuación.

Concentración (Muy): La operación de concentración tiene como principal efecto (sobre los conjuntos difusos) el de reducir los valores de pertenencia para aquellos elementos que poseen menores valores de pertenencia. Esta operación es dada por la expresión 3.8.

$$\mu_{\text{con (A)}} (x) = (\mu_A (x))^2 \quad (3.8)$$

Dilatación (Casi): La operación de dilatación dilata los conjuntos difusos mediante el incremento de los valores de pertenencia lo cual se representa mediante la expresión 3.9.

$$\mu_{\text{dil (A)}} (x) = (\mu_A (x))^{0.5} \quad (3.9)$$

Poder (Muy-Muy): La operación de poder es simplemente una extensión de la operación de concentración la cual se representa mediante la ecuación 3.10.

$$\mu_{\text{pod (A)}} (x) = (\mu_A (x))^n \quad (3.10)$$

3.8 OPERACIONES SOBRE CONJUNTOS DIFUSOS

Al igual que en la teoría clásica de conjuntos, dentro de la lógica difusa existen las operaciones básicas de intersección, unión y complemento, operaciones que son sumamente utilizadas y que a continuación se muestran. Aquí únicamente se mostrara la expresión matemática que define cada operación. Para mayor detalle y formalidad sobre cada operación es conveniente consultar alguno de los textos [TX-1], [TX-15], [TX-19] y [TX-21] a partir de los cuales las expresiones 3.11, 3.12 y 3.13 fueron obtenidas.

Intersección:

$$\mu_{A \cap B}(X) = \min (\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) = \mu_A(x) \cap \mu_B(x) ; \quad \forall x \in X \quad (3.11)$$

Unión:

$$\mu_{A \cup B}(X) = \max(\mu_A(X), \mu_B(X)) = \mu_A(X) \vee \mu_B(X) = \mu_A(X) \cup \mu_B(X) ; \quad \forall x \in X \quad (3.12)$$

Complemento:

$$\mu_{\neg A}(X) = 1 - \mu_A(X) \quad (3.13)$$

3.9 INFERENCIA DIFUSA

Dentro de la lógica difusa los conjuntos difusos son manipulados como proposiciones difusas donde una proposición difusa es una oración que se sostiene para alguna variable lingüística (i.e. velocidad es rápida). En general las proposiciones difusas se representan como:

$$\text{Proposición: } X \text{ is } A \quad (3.14)$$

donde: "A" es un conjunto difuso dentro del universo de cobertura "X". Así las reglas difusas relacionan a dos o más proposiciones difusas en la forma de sentencias IF-THEN (ecuación 3.15).

$$\text{IF } X \text{ is } A \text{ THEN } Y \text{ is } B \quad (3.15)$$

(conj. difuso dentro de A)

La expresión o regla 3.15 establece una relación entre las dos proposiciones. De esta forma los controladores y sistemas expertos difusos almacenan reglas como asociaciones difusas. Dichas asociaciones mapean conjuntos difusos hacia otros conjuntos difusos lo que comúnmente es llamado proceso de inferencia difusa. El proceso de inferencia difusa trata de establecer una creencia (belief) de la conclusión de una regla dado que se posea suficiente evidencia referente a su premisa. Pero debido a que las proposiciones contenidas dentro de la regla son conjuntos difusos, la lógica difusa mapea información de conjuntos de premisas (difusas) a información de conjuntos de conclusiones (también difusas).

Actualmente las dos metodologías de inferencia difusa más utilizadas debido a su sencillez y su confiabilidad [TX-22] son los procesos de "Max-Min" y de "Max-Product" los cuales pueden ser utilizados bajo múltiples reglas (empleando Forward o Backward Chaining) como se muestra gráficamente en las figuras 3.3 y 3.4.

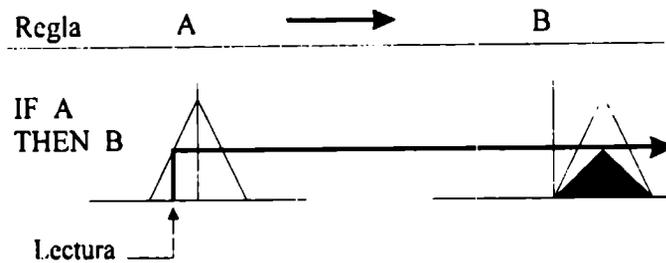


Fig. 3.3 Inferencia Max-Product para una sola premisa. Este tipo de inferencia puede utilizarse bajo múltiples reglas similar a proceso de la fig. 3.4b, con la diferencia de que en vez de trapecios se obtiene un conjunto de triángulos sobre los cuales se infiere un resultado.

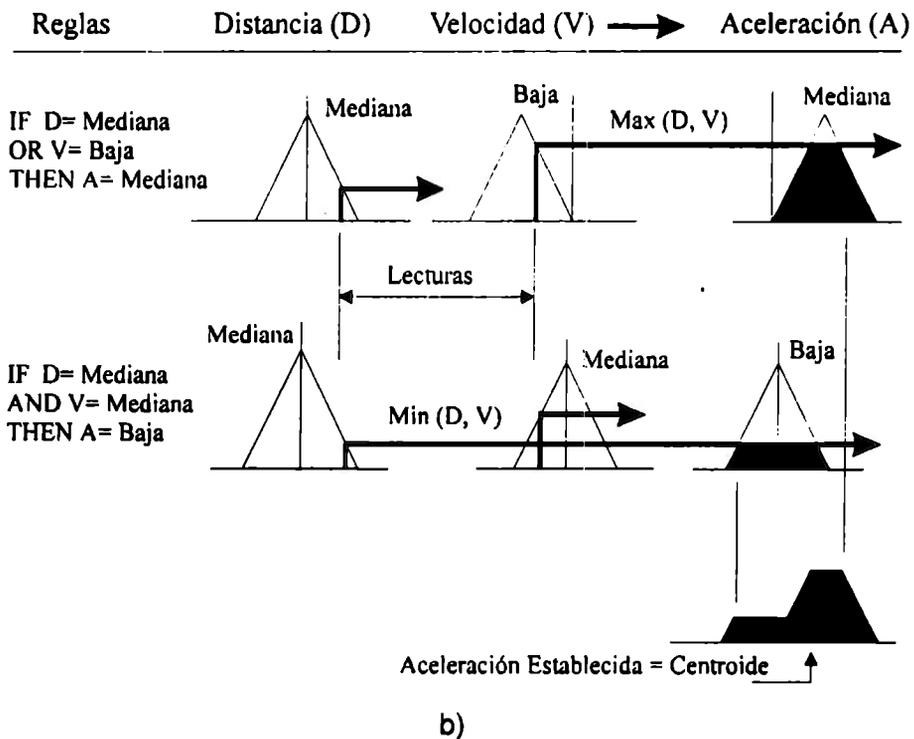
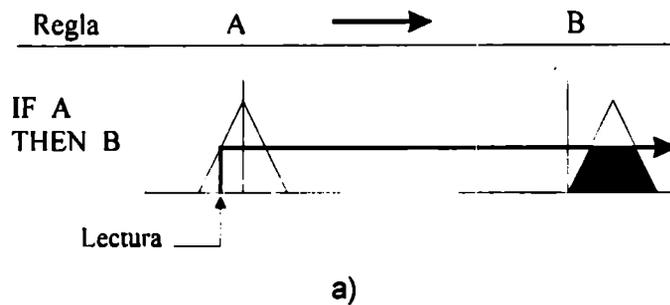


Fig. 3.4 Inferencia Max-Min a) para una sola premisa b) Inferencia Max-Min para un conjunto de reglas que se pueden generar para el caso de la navegación de robots móviles. En la parte "b" se muestra el proceso de defuzificación centroidal que será descrito en la sección 3.9.

3.10 MÉTODOS DE DEFUZIFICACION

Por lo general dentro de los sistemas de control (como el problema de navegación) existen situaciones donde la salida de un proceso difuso requiere ser un valor escalar en vez de un conjunto difuso. La conversión de cantidades difusas a cantidades escalares es llamada defuzificación (mientras que la conversión de valores escalares a conjuntos difusos es llamada fuzificación). En la actualidad existen muchos métodos de defuzificación pero dentro de la literatura [TX-1], [AR-25], se mencionan cinco métodos de uso general que han demostrado buenos resultados en el control de sistemas, y los cuales serán estudiados dentro del presente trabajo (esto con el fin de poder estudiar cual de ellos es el más recomendable para ser usado dentro de la navegación de robots móviles). Los cinco métodos a los que se hace referencia son los métodos de "Centroid method", "Max-membership principle", "Weighted average method", "Mean-max membership", y finalmente el método de "Center of sums" los cuales se describen a continuación.

1.- *Centroid method*: Este procedimiento de defuzificación es uno de los más utilizados el cual consiste esencialmente en localizar el centroide (o centro de gravedad) del área obtenida a partir del proceso de inferencia tal como puede observarse en la figura 3.5.

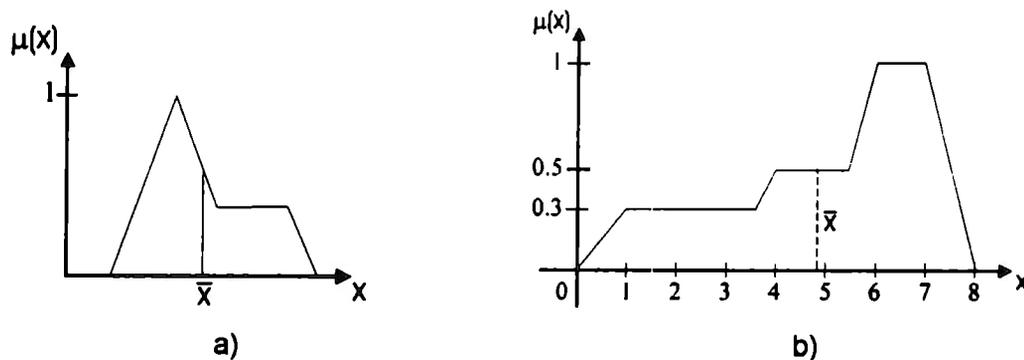


Fig. 3.5 a) Método de defuzificación centroidal, b) Ejemplo.

La ecuación para obtener el centroide de cualquier figura esta dada por la expresión 3.16.

$$x^* = \frac{\int \mu(x) \cdot x \, dx}{\int \mu(x) \, dx} \quad (3.16)$$

2.- *Max-membership principle*: Este método también conocido como "height method" consiste simplemente en seleccionar el valor máximo de las funciones de salida. Este método esta dado por la siguiente expresión algebraica:

$$\mu(x^*) \geq (\mu(x)) \quad (3.17)$$

Este método se representa gráficamente como se muestra en la figura 3.6.

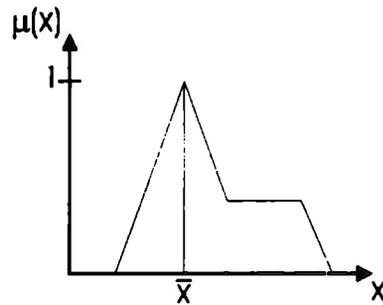


Fig. 3.6 Método de defuzificación Max-membership principle.

3.- *Weighted average method*: Este método es únicamente valido para cuando se tienen funciones de pertenencia simétricas. El método es descrito por la ecuación 3.18 y el cual puede observarse de manera gráfica en la figura 3.7.

$$x^* = \frac{\sum \mu(\bar{x}) \cdot \bar{x}}{\sum \mu(\bar{x})} \quad (3.18)$$

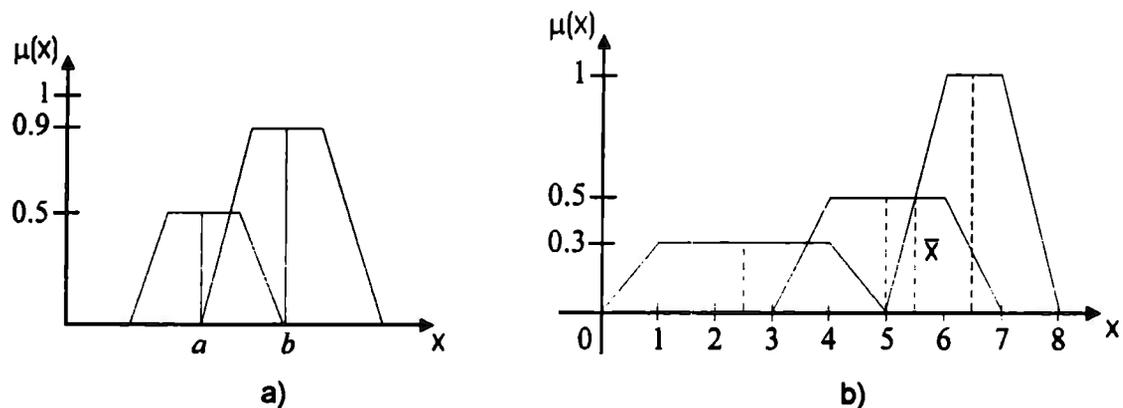


Fig. 3.7 a) Método de defuzificación Weighted average, b) Ejemplo.

4.- *Mean-max membership*: Este método es parecido al método de Max-membership principle, con la diferencia de que el máximo valor de pertenencia puede no ser único, por lo que la expresión para determinar la salida esta dada por la siguiente expresión:

$$x^* = (a + b) / 2 \quad (3.19)$$

donde los valores de "a" y de "b" pueden verse representados en la siguiente figura 3.8.

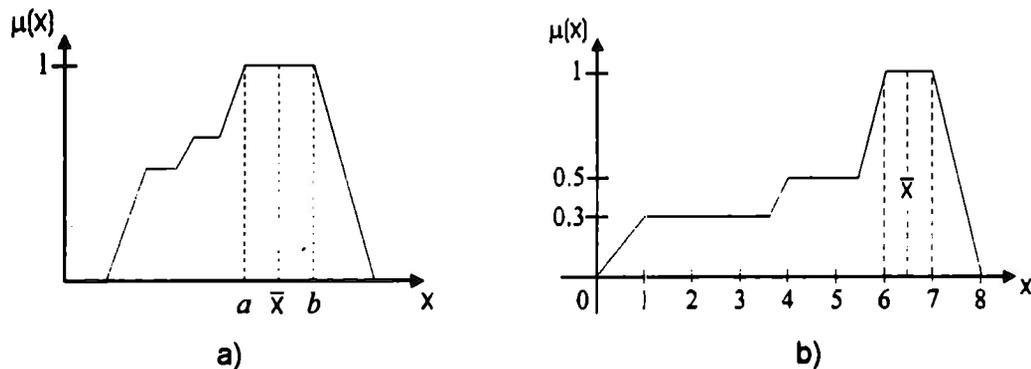


Fig. 3.8 Dos ejemplos que demuestran el método de defuzificación Mean-max.

5.- *Center of Sums*: Este método es uno de los más rápidos el cual consiste en realizar una suma algebraica de cada conjunto difuso obtenido de las salidas individuales. El único valor que se obtiene a partir de este método se genera a partir de la ecuación 3.19 cuya representación gráfica se muestra en la figura 3.9.

$$x^* = \frac{\int_x x \sum_{k=1}^n \mu_k(x) dx}{\int_x \sum_{k=1}^n \mu_k(x) dx} \quad (3.20)$$

3.11 COMENTARIOS FINALES

Ya conocidos los principios básicos referentes a la lógica difusa (descritos en el presente capítulo) con los cuales es posible manejar información imprecisa y/o vaga, se procederá a diseñar un controlador difuso que permita a un robot móvil navegar de manera autónoma en

ambientes desconocidos donde los obstáculos se encuentren en reposo (ambiente estático) y donde la superficie de navegación (terreno) es completamente plano. Este será el propósito del siguiente capítulo.

Cada una de las técnicas de inferencia y de defuzificación aquí descritas presentan ventajas y desventajas. De los dos métodos de inferencia estudiados (Max-Min y Max-Product) la técnica a emplear en el presente trabajo es el método de inferencia Max-Min. A pesar de que en generar es más fácil programar el método Max-Product se decidió utilizar para la técnica de inferencia Max-Min debido a que es una técnica más conservadora y que en general (según varios autores [TX-1], [TX-21]) es ideal para cuando se estudia el empleo de lógica difusa en sistemas por primera vez (en donde se desconoce totalmente el comportamiento del sistema bajo un controlador difuso). Por otra parte de entre los muchos métodos de defuzificación existentes hoy en día, en el presente trabajo se decidió emplear el método de defuzificación centroidal. Este método fue seleccionado debido a que es uno de los pocos métodos de defuzificación en el cual existen cambios moderados entre salidas o resultados adyacentes de dos inferencias (existe continuidad). Esto no siempre sucede en otros métodos donde pequeñas modificaciones en las señales de entrada al controlador difuso pueden causar grandes cambios en las señales de control. Esto es de suma importancia para la navegación de sistemas autónomos donde un cambio brusco en una señal de control puede causar que el vehículo no logre su objetivo (i.e. si el robot móvil se encuentra cerca de un obstáculo un cambio brusco en su velocidad o en su dirección puede causar que el robot choque con el obstáculo). Estas decisiones se describirán con mayor detalle en el capítulo No. 5 donde se describe la construcción del sistema de navegación.

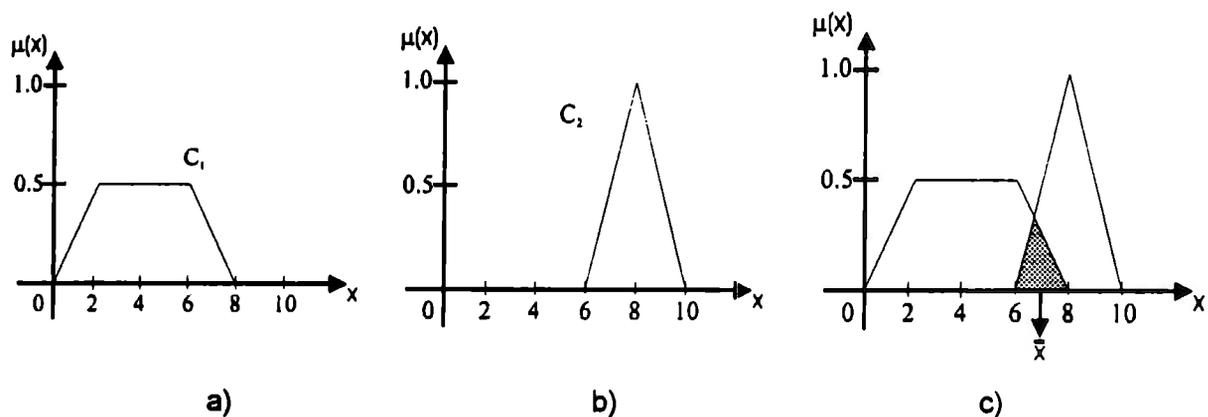


Fig. 3.9 Método de defuzificación Center of sums, a) primera función de pertenencia, b) segunda función de pertenencia y c) método de defuzificación realizado sobre los conjuntos difusos de las figuras 3.9a y 3.9b.

4 ELABORACIÓN DEL CONTROLADOR DIFUSO

Realizada la selección de la metodología a emplear basada en el estudio de las técnicas de navegación utilizadas con mayor frecuencia en sistemas autónomos (capítulo No. 2) y conociendo los principios y conceptos básicos referentes a la lógica difusa descritos en el capítulo No. 3, se procederá al diseño e implantación de la técnica seleccionada, un controlador de navegación difuso (Fuzzy Controller) el cual permita a un robot móvil navegar de manera autónoma y segura evitando obstáculos a través de un ambiente estático (obstáculos sin movimiento) totalmente desconocido a priori por el robot.

Para llevar a cabo esto, se seguirán siete pasos o etapas fundamentales en el desarrollo de un controlador difuso, etapas que son recomendadas por varios expertos en el área como Maeda, E.H.Mamdani, L.P.Holmblad, J.J.Ostergaard, C.Yasunobu, por mencionar solo algunos. Las etapas recomendadas y las cuales se seguirán durante el presente estudio, son las siguientes:

Etapas 1 : Definición del problema.

Etapas 2 : Definición de las variables lingüísticas.

Etapas 3 : Definición de los conjuntos difusos.

Etapas 4 : Definición de las reglas difusas.

Etapas 5 : Construcción del sistema (robot móvil).

Etapas 6 : Prueba del sistema.

Etapas 7 : Ajuste y afinación del sistema.

A continuación se presenta el desarrollo de cada una de estas siete etapas.

4.1 ETAPA 1: DEFINICIÓN DEL PROBLEMA

La primera etapa en la creación de un controlador de tipo difuso consiste en la definición y planteamiento del problema a resolver. Dentro de esta etapa se definirán las restricciones así como las condiciones sobre las cuales se elaborará el controlador difuso.

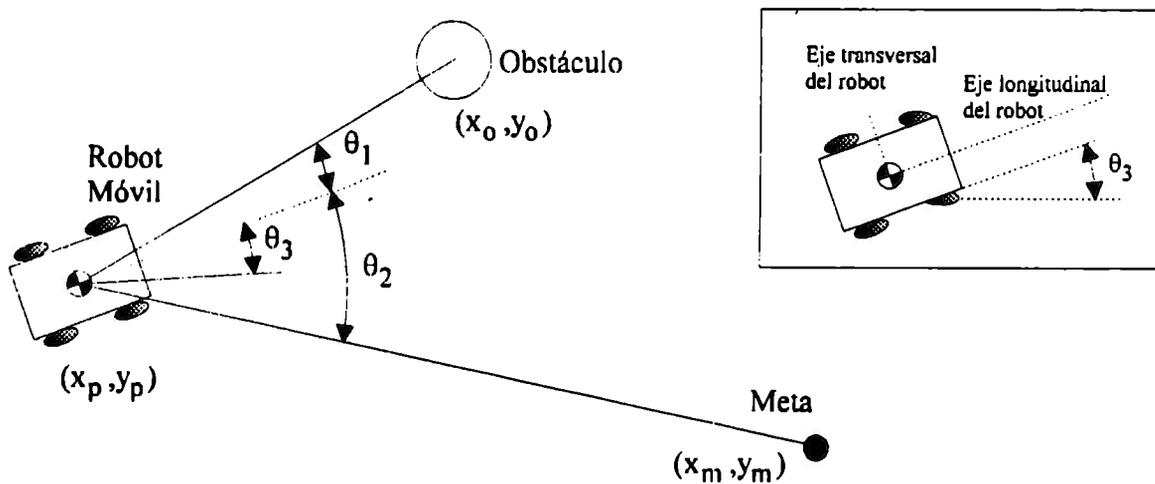
El problema a resolver consiste en hacer que un robot móvil navegue de manera autónoma dentro de un ambiente totalmente desconocido, es decir, que el robot no cuenta con ninguna información previa (mapas) sobre el terreno en el cual se desempeñará. Además de poderse mover y maniobrar sobre cierto espacio, el robot deberá alcanzar, desde su posición inicial un objetivo, el cual consiste en llegar a un determinado punto (fijado por el usuario) que llamaremos punto meta. Esto se deberá llevar a cabo sin que el robot sufra colisiones con los posibles obstáculos presentes en el terreno en el menor tiempo y recorriendo la menor distancia posible (optimizar la distancia y el tiempo de navegación).

Al elaborar un controlador difuso, lo primero que hay que obtener es la fuente de conocimiento, la cual usualmente proviene de un experto en el área. Para el presente problema, se ha obtenido el conocimiento de varios artículos presentados a lo largo de varios años en conferencias y simposiums internacionales en donde se ha tratado de resolver el mismo problema [AR-4, -, AR-69]. De dichos artículos se sabe que el problema básico a resolver es lograr que un robot móvil navegue de forma autónoma, eficiente y segura desde una posición inicial, a una posición final llamada meta. La máxima eficiencia de la navegación se obtiene al minimizar tanto la distancia recorrida como el tiempo que tarda el robot en llegar a la meta. La seguridad del robot también es un factor importante dentro de la navegación, la cual se logra evadiendo cualquier obstáculo, ya sea estático o dinámico encontrado en el trayecto. La figura 4.1 muestra de manera gráfica los parámetros y la configuración que describe el problema de navegación que se desea resolver.

Para llevar a cabo estas tareas (minimizar la distancia y el tiempo de recorrido evitando obstáculos en el trayecto, hasta que el robot alcance la meta) se ha decidido, analizando las ventajas y desventajas de las diferentes técnicas utilizadas hasta ahora por diferentes investigadores para solucionar el mismo problema (ver capítulo 2), desarrollar un sistema difuso para controlar tanto la dirección como la velocidad del robot móvil

Este problema es básicamente un problema en el cual el error, o diferencia entre el punto inicial del robot y el punto meta a alcanzar por el vehículo debe nulificarse (hacerse cero). Inicialmente el robot debe dirigirse hacia el punto meta mediante la minimización del

error existente entre la dirección angular del robot y la dirección hacia la meta dado por el ángulo θ_2 (ver figura 4.1). El robot también debe acelerar a una cierta velocidad máxima, y a continuación disminuir la velocidad hasta que eventualmente se detenga cuando llegue a la posición meta. Con esto, el error entre la localización del robot (x_p, y_p) y la localización de la meta (x_m, y_m) es minimizada de la manera más rápida posible. Además, cuando un obstáculo sea localizado en el trayecto del robot ($\theta_1 = 0$), el robot debe cambiar su dirección (incrementando θ_1 y modificando θ_3) de tal manera que logre rodear (evadir) el obstáculo de manera cuidadosa (con velocidad baja) para que a continuación vuelva a incrementar su velocidad dirigiéndose nuevamente hacia el punto Meta.



θ_1 - Ángulo relativo al robot de la posición del obstáculo.

θ_2 - Ángulo del error existente entre la dirección del robot y el punto meta.

θ_3 - Dirección de las ruedas del robot referente a su eje longitudinal.

(x_m, y_m) - Coordenadas de la posición del punto meta.

(x_o, y_o) - Coordenadas del punto a la que se detecto el obstáculo.

(x_p, y_p) - Coordenadas de la posición del centro de gravedad del robot.

Fig. 4.1 Geometría de la navegación para un robot móvil (en el presente trabajo los ángulos serán medidos en la dirección de movimiento de las manecillas del reloj respecto al eje longitudinal del robot).

Con estas acciones el robot móvil deberá lograr minimizar la distancia entre su posición actual y el punto meta, es decir, debe lograr que la ecuación 4.1 se cumpla, la cual se satisface cuando $x_p = x_m$ y $y_p = y_m$.

$$\sqrt{(x_p - x_m)^2 + (y_p - y_m)^2} = 0 \quad (4.1)$$

Por otra parte, para que el robot no choque contra los obstáculos presentes en el terreno, siempre debe satisfacerse la relación 4.2.

$$\sqrt{(x_p - x_o)^2 + (y_p - y_o)^2} > \left\{ \frac{\text{Longitud total del robot}}{2} + \Delta \right\} \quad (4.2)$$

donde Δ es una distancia de seguridad entre el robot y el obstáculo definida por el usuario conforme las restricciones de movimiento del robot de tal manera que esté pueda evadir el obstáculo.

Dada la alta complejidad del problema a resolver (ver capítulo 2), dentro de la primer etapa, además de definir el problema, se han impuesto varias restricciones sobre el sistema, de tal forma que el modelo que lo represente, sea lo más sencillo posible a fin de poderlo estudiar y resolver de la mejor manera. Las restricciones impuestas son las siguientes:

- Se considerará que el robot únicamente navega sobre superficies planas.
- Se considera que el ambiente en el cual navegara el robot es estático, es decir, todos los obstáculos están fijos y sin movimiento.
- Se consideraran únicamente poliedros como únicas formas de los obstáculos.
- Se considera que el robot no puede retroceder (no cuenta con reversa).

Además de las restricciones anteriores, se consideraran los siguientes aspectos dentro del proyecto, que de cierta manera plantean los problemas a resolver:

- Se considerará que el robot navegara en ambientes totalmente desconocidos (no se cuenta con información preliminar del terreno ni de los obstáculos, es decir, no se cuenta con mapas del terreno).
- El robot decidirá de manera autónoma la trayectoria a seguir para alcanzar el punto meta por medio de información proveniente de sensores de ultrasonido.

- Se considerará al robot como un objeto con volumen.

Para obtener la información referente a la posición y orientación relativa al robot de los obstáculos, el robot móvil cuenta con siete sensores de ultrasonido localizados en su parte frontal, tal como puede observarse en las figuras 4.2 y 4.3 (en el siguiente capítulo se expondrá la razón del porque se decidió emplear este tipo de sensores).

Con estos sensores el robot puede percibir su entorno, tal como puede observarse en la figura 4.2. La figura 4.2a muestra el rango de activación para cada uno de los siete sensores antes mencionados. Con los siete sensores, el robot tiene a su disposición siete valores de distancia y siete valores de dirección dentro de los cuales los obstáculos pueden ser localizados. Mediante la información proveniente de los sensores, es posible definir una posición y una dirección relativa de los obstáculos. En la figura 4.2c se muestra un obstáculo detectado por el sensor "S2", con lo cual la distancia del obstáculo estará dada por este sensor, mientras que la dirección relativa del mismo obstáculo estará entre -70° y -42° (cabe destacar que la posición, distancia y dirección relativa del obstáculo detectado, está dada en las coordenadas polares del robot).

Por otra parte, el robot puede adquirir una cierta velocidad máxima y sus ruedas pueden girar como máximo, un determinado ángulo (dado por θ_3 - Dirección de las ruedas del robot referente a su eje longitudinal, ver figura 4.1). Estas y otras variables serán descritas con mayor detalle dentro de este capítulo, así como también se definirán sus rangos de acción o de cobertura (conforme a restricciones de movimiento). Este será el tema de la siguiente etapa (etapa número 2) en el desarrollo del controlador difuso.

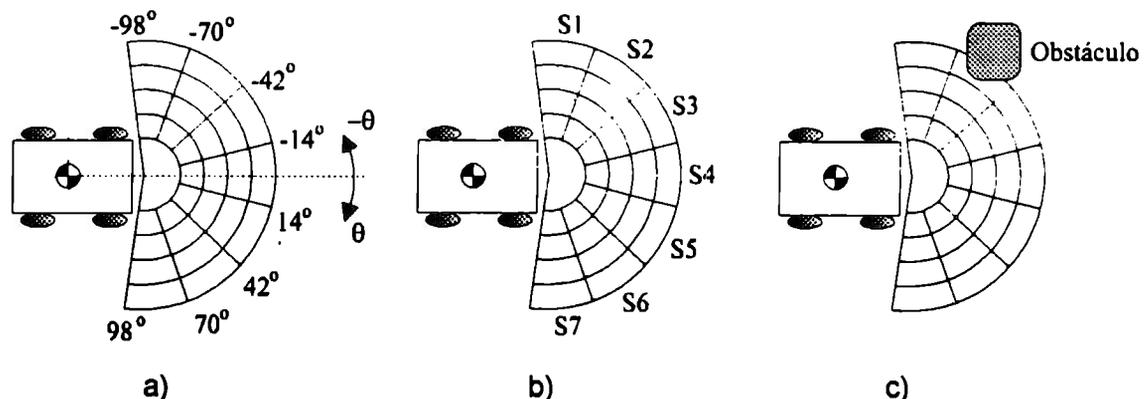
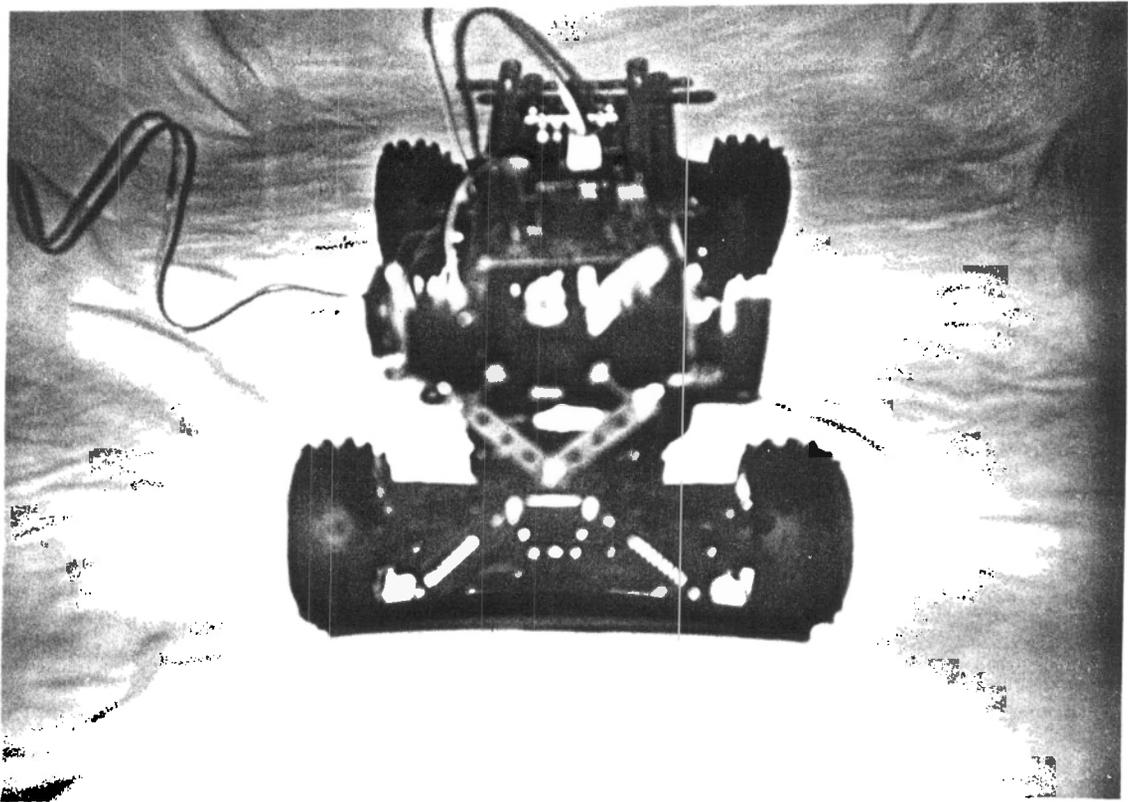
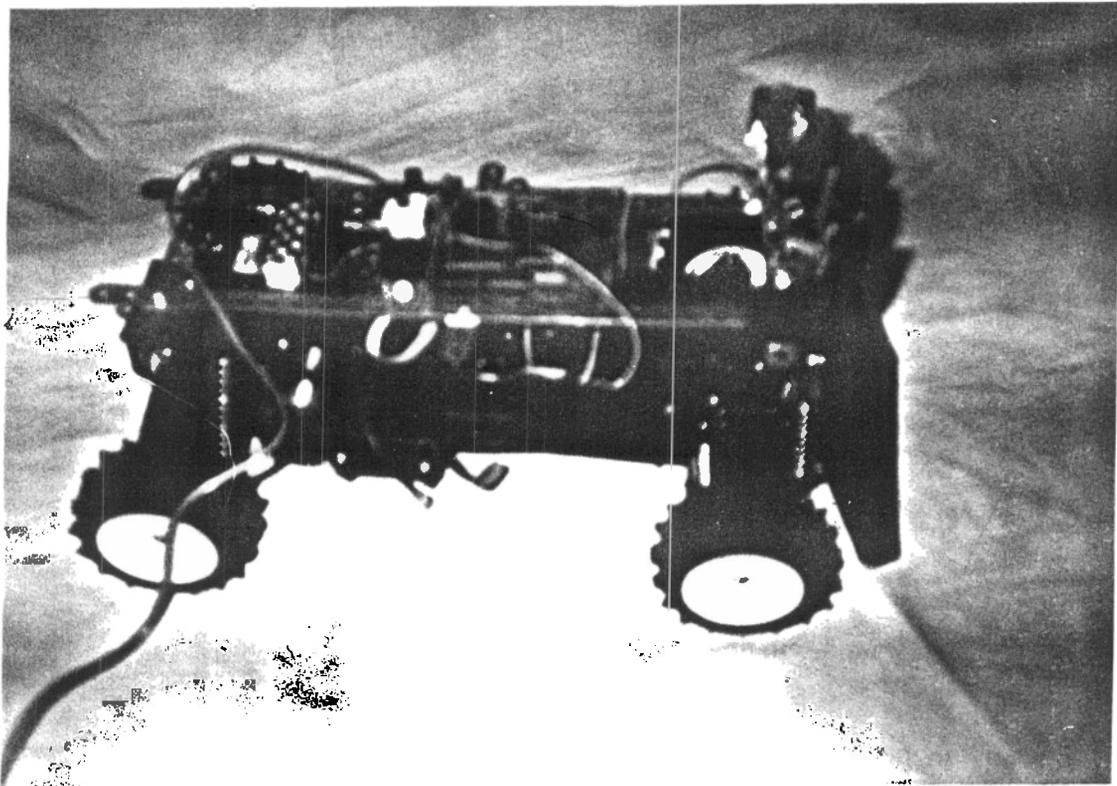


Fig. 4.2 Localización de los sensores de ultrasonido sobre el robot móvil a) Dirección y rango de cobertura de cada sensor, b) Número de sensor (S1 - S7) y c) Obstáculo detectado por el sensor S2.



a)



b)

Fig. 4.3 Robot móvil diseñado y construido durante el proyecto, a) Vista frontal, b) Vista lateral (fotografías originales a color).

4.2 ETAPA 2: DEFINICIÓN DE LAS VARIABLES LINGÜÍSTICAS

El siguiente paso en el diseño y creación del controlador difuso que controle la navegación autónoma de un robot móvil, es definir las variables lingüísticas para el problema a resolver. Para lograr esto, es necesario conocer la manera de resolver el problema. Esto se alcanzará mediante la conversación con un experto en el área quien indique como lograrlo. Pero debido a que no existe un robot experto con el que se pueda conversar para estudiar como es que un robot móvil resuelve el problema de navegación en ambientes totalmente desconocidos, en esta ocasión el experto será cualquier persona (nosotros mismos) quien con base en su propia experiencia y sentido común, nos indique como es que el ser humano resuelve el problema de navegación durante acciones cotidianas, como son la conducción de un automóvil, manejar una bicicleta o simplemente al caminar a través del medio que nos rodea, acciones similares a las que tiene que resolver un vehículo autónomo.

Mediante este razonamiento, el principal objetivo que se desea resolver al consultar a una persona común y corriente, es el de descubrir las variables que representaran los universos de cobertura (velocidad, dirección, distancia, etc.) y los conjuntos difusos (lejos, cerca, medio lejos, lento, medio lento, rápido, derecha, izquierda, etc.) utilizados durante la navegación que serán definidos para cada uno de estos universos.

De la etapa 1, definición del problema, se sabe que el sistema difuso a diseñar debe poder resolver de forma satisfactoria los siguientes tres problemas básicos, no importando el orden ni prioridades:

- Controlar la dirección del robot de tal forma de dirigirlo hacia el punto meta.
- Controlar la dirección del robot de tal forma que esté pueda evadir cualquier obstáculo.
- Controlar la velocidad del robot.

A partir de la experiencia del ser humano (consulta al experto) en la navegación (al conducir un automóvil o simplemente al caminar) se ha creado la siguiente estrategia lógica para controlar la dirección del robot de tal forma de dirigirlo hacia el punto meta:

“Cuando la dirección del robot no es hacia el punto meta, hacer que su dirección gire hacia el punto meta (hacer $\theta_2 = 0$, ver figura 4.1)”

De manera similar, con base en el sentido común, la siguiente estrategia para controlar la dirección del robot para que esté pueda evitar obstáculos ha sido elaborada:

“ Cuando el robot esta cerca de algún obstáculo y al mismo tiempo su dirección es hacia el obstáculo ($\theta_1 \approx 0$), entonces disminuir la velocidad del robot y hacer que la dirección del robot se aleje del obstáculo (incrementar θ_1 , ver figura 4.1)”

Finalmente, siguiendo la misma metodología (con base en el sentido común del ser humano), las siguientes estrategias para controlar la velocidad del robot han sido obtenidas:

“Cuando el robot este cerca de algún obstáculo, disminuir su velocidad”

“Cuando el robot este lejos del punto meta, incrementar su velocidad”

“Cuando el robot este cerca del punto meta, disminuir su velocidad”

Con base en la discusión y los resultados anteriores, a continuación se definirán las variables lingüísticas a utilizarse en el presente trabajo, así como también se definirán los universos de cobertura para cada una de ellas, y finalmente se especificaran sus rangos de cobertura con base en las restricciones de movimiento presentes en el robot (prototipo) que se ha venido construyendo paralelo al presente análisis, y cuyas características físicas se describirán más adelante dentro de la etapa número 5 (construcción del sistema). Las variables lingüísticas seleccionadas para atacar el problema de la navegación en ambientes desconocidos por robots móviles y que tienen como base el diagrama mostrado en la figura 4.1, son presentadas en la tabla 4.1.

Tabla 4.1 Variables lingüísticas con sus respectivos universos o rangos de cobertura utilizadas para la elaboración del controlador de navegación difuso.

VARIABLE LINGÜÍSTICA	RANGO
Ángulo del obstáculo (θ_1).	-98 a 98 grados
Ángulo error (θ_2).	-180 a 180 grados
Ángulo de dirección (θ_3).	-33 a 33 grados
Velocidad (del robot).	0 a 100 [%]
Distancia del robot al punto meta.	0 a 10 mts.
Distancia del obstáculo relativo al robot.	0 a 1 mts.

El rango para cada una de las variables lingüísticas presentadas en la tabla 4.1 fueron determinados con base en los siguientes factores (basadas en las características del robot prototipo):

1. **Ángulo del obstáculo** (θ_1).- El rango de -98 a 98 grados dentro del cual el vehículo autónomo es capaz de percibir su entorno y de detectar los obstáculos, esta determinado por la cobertura total que tienen los siete sensores de ultrasonido (colocados en la parte frontal del robot) dada su distribución sobre el robot móvil (ver figuras 4.2 y 4.3). Ángulos negativos, indicados por el símbolo menos (-), indican que el obstáculo (detectado) se encuentra a la izquierda del robot, mientras que ángulos positivos determinan que los obstáculos detectados se encuentran a su derecha (por simplicidad y costo, el robot únicamente percibe el entorno que se encuentra en su parte frontal, parecido al rango de visión que posee el sistema ocular del ser humano).
2. **Ángulo error** (θ_2).- El rango de -180 a 180 grados fue fijado dado que el punto meta puede estar localizado en cualquier dirección respecto a la dirección del robot. Un ángulo negativo le indica al robot que esté tendrá que girar hacia su izquierda para poder disminuir el error (dirigirse hacia su destino) entre su dirección actual y la localización del punto meta, mientras que un ángulo positivo indica lo contrario (el robot tendrá que girar hacia su derecha para poder disminuir el error entre la dirección actual y la localización del punto meta).
3. **Ángulo de dirección** (θ_3).- El rango de -33 a 33 grados en el ángulo de dirección, esta definido a partir del movimiento restringido de las ruedas delanteras (ver figura 4.1) con las que el robot cuenta para establecer la dirección de movimiento al desplazarse a través del terreno. Ángulos negativos indican que las ruedas están giradas hacia la izquierda respecto al eje longitudinal del robot (el robot esta girando a la izquierda) mientras que ángulos positivos indican lo contrario, es decir, que el robot esta girando hacia su derecha. La magnitud del ángulo indica que tan grande es el radio de giro del robot. Si el ángulo tiene una magnitud de cero, esto indica que el robot esta desplazándose en línea recta.

4. **Velocidad (del robot).**- El rango en la velocidad del robot de 0 a 100 [%] esta determinada por el sistema de engranes y tipo de motor con el que cuenta el robot (un único motor es el utilizado para este propósito). Se decidió emplear % de velocidad con el fin de poder utilizar el mismo controlador bajo diferentes velocidades máximas que puedan ser alcanzadas por el robot (este aspecto será descrito con mayor detalle cuando se explique la construcción del robot experimental). Para solucionar el problema de navegación de manera sencilla, únicamente se ha considerado que el robot se desplaza hacia adelante (el robot no cuenta con reversa). Esta decisión fue tomada con el motivo de evitar que el robot oscile (como se ha visto que suele suceder en algunos estudios y resultados obtenidos por varios investigadores) para ciertas configuraciones del entorno, como suelen ser pasillos sin salida, paredes encontradas y lugares cerrados en general, configuraciones que pueden ser evitadas si el robot, al ir navegando, va construyendo un mapa interno del entorno (i.e. en la memoria), problema que por el momento no se desea atacar y el cual se plantea como un posible tema para futuras investigaciones.

5. **Distancia del robot al punto meta.**- El rango de la distancia entre la posición actual del robot al punto meta, es simplemente una aproximación de la distancia dentro de la cual el robot debe navegar, pudiéndose incrementar o disminuir según las necesidades particulares y posibles aplicaciones para las que sea utilizado el robot móvil. En el presente caso, distancias mayores a 10 mts. son consideradas muy lejanas, lo que significa que el robot debe adquirir la mayor velocidad posible para alcanzar su meta en el menor tiempo posible.

6. **Distancia del obstáculo relativo al robot.**- El rango de 0 a 1 mts. esta determinado por la máxima distancia de percepción a la que se han ajustado los sensores de ultrasonido diseñados para ser colocados al robot (máxima distancia a la que pueden detectar un obstáculo). Esta distancia puede incrementarse si los sensores son reajustados para que tengan una mayor zona de sensibilidad, pero aún cuando esto se realizará, la distancia entre 0 y 1 mts. es una distancia razonable (conforme la máxima velocidad que puede ser alcanzada por el sistema) dentro de la cual el robot puede decidir si continuar por el mismo trayecto o modificarlo. El diseño de los sensores de ultrasonido será descrito en la etapa 5, "Construcción del sistema".

4.3 ETAPA 3 : DEFINICIÓN DE LOS CONJUNTOS DIFUSOS

La siguiente etapa que se llevara a cabo en el diseño y creación del controlador de navegación difuso, es la definición de los conjuntos difusos para cada uno de los universos de cobertura de cada una de las variables lingüísticas obtenidas anteriormente (ver tabla 4.1). Para lograr esto, el primer paso a realizar, es el de definir los adjetivos que se emplearán dentro de cada variable lingüística definida previamente en la etapa número 2 " Definición de las variables lingüísticas" . Para esto, nuevamente se ha recurrido al sentido común y a los adjetivos utilizados frecuentemente por el ser humano al navegar en el mundo que le rodea. Los adjetivos seleccionados para cada una de las variables lingüísticas de la tabla 4.1 están expuestos en la tabla 4.2, los cuales representan de cierta manera los adjetivos típicos utilizados dentro de la teoría de la lógica difusa.

Tabla 4.2 Conjuntos difusos con sus respectivos adjetivos para el problema de navegación.

Ángulo del obstáculo (θ_1)	Ángulo error (θ_2)	Ángulo de dirección (θ_3)	Velocidad (del robot)	Distancia del robot al punto meta	Distancia del obstáculo relativo al robot
IG	IG	IG	C	C	C
IP	IP	IP	RL	CE	CE
C	C	C	L	RE	M
DP	DP	DP	M	M	LE
DG	DG	DG	RA	LE	---

donde:

C = Cero

CE = Cerca

M = Mediana

DP = Derecha Pequeña

IG = Izquierda Grande

IP = Izquierda Pequeña

DG = Derecha Grande

L = Lento

LE = Lejos

RA = Rápido

RE = Regular

RL = Realmente Lento

A continuación se formarán los conjuntos difusos y sus respectivas funciones de pertenencia para cada uno de los adjetivos de cada una de las variables lingüísticas presentadas en la tabla 4.2. Las funciones de pertenencia para los conjuntos difusos pueden adquirir cualquier forma, dependiendo como se relacionen los diferentes valores del dominio a sus correspondientes valores de pertenencia o creencia. Para resolver nuestro problema se utilizarán segmentos de funciones lineales, con las cuales se formarán funciones

triangulares (triangular fuzzy sets). Se ha seleccionado este tipo de funciones de pertenencia para representar los conjuntos difusos, ya que capturan de manera adecuada la creencia basada en el sentido común, además de que reducen de manera significativa el esfuerzo computacional.

Las siguientes 6 figuras (figuras 4.4 a 4.9) muestran los conjuntos difusos para cada uno de los términos y variables lingüísticas presentados en la tabla 4.2. Además cada figura también presenta las correspondientes funciones triangulares (expresadas por sus dos valores extremos y su valor intermedio) en forma analítica para cada uno de los conjuntos difusos dentro de cada variable lingüística.

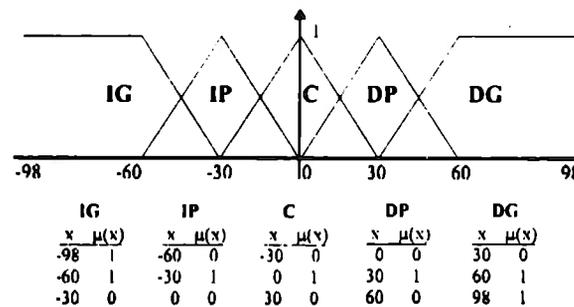


Fig. 4.4 Conjuntos difusos sobre la variable lingüística "Ángulo del obstáculo (θ_1)".

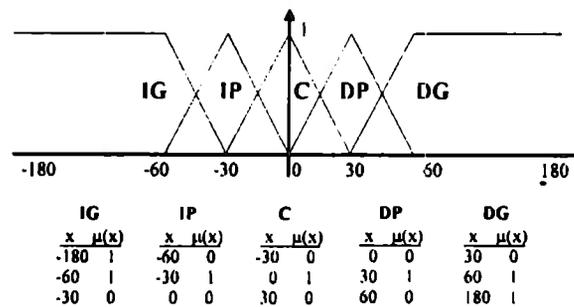


Fig. 4.5 Conjuntos difusos sobre la variable lingüística "Ángulo error (θ_2)".

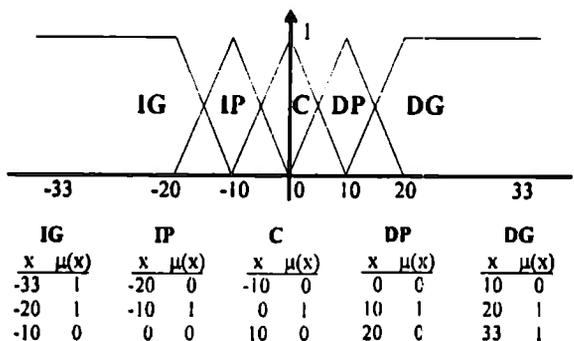


Fig. 4.6 Conjuntos difusos sobre la variable lingüística "Ángulo de dirección (θ_3)".

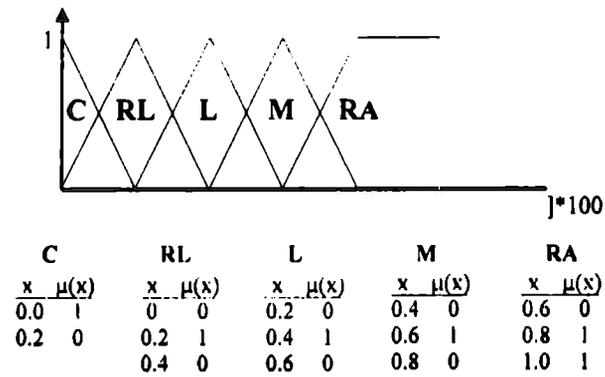


Fig. 4.7 Conjuntos difusos sobre la variable lingüística " Velocidad (del robot) ".

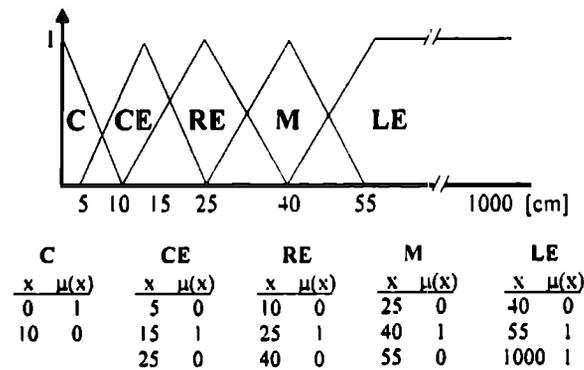


Fig. 4.8 Conjuntos difusos sobre la variable lingüística " Distancia del robot al punto meta ".

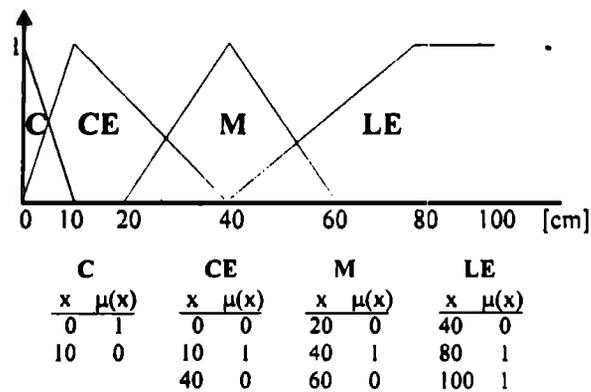


Fig. 4.9 Conjuntos difusos sobre la variable lingüística " Distancia del obstáculo relativo al robot ".

4.4 ETAPA 4: DEFINICIÓN DE LAS REGLAS DIFUSAS

La cuarta etapa en la elaboración de un controlador difuso, es la definición de las reglas difusas necesarias por la máquina de inferencia, para que ésta tome las decisiones pertinentes (acciones) del sistema a controlar. En nuestro caso en particular, las decisiones que interesan son las de la velocidad y dirección que el robot debe adoptar, de forma tal que alcance el punto meta sin sufrir colisiones con los posibles obstáculos presentes en el ambiente. Para llevar a cabo ésta tarea de forma adecuada, se tienen que generar un conjunto de reglas, tales que cubran los tres principales problemas a resolver mencionados anteriormente:

1. Controlar la dirección del robot de tal forma que se dirija hacia el punto meta.
2. Controlar la dirección del robot de forma tal que le sea posible evadir cualquier obstáculo.
3. Controlar la velocidad del robot.

La elaboración de las reglas, al igual que la elaboración de los conjuntos difusos, fue realizada con base en la experiencia y sentido común del ser humano cuando éste navega por nuevos e inesperados terrenos . Para esto, un pequeño conjunto de cinco individuos (alumnos del ITESM-CCM, incluyendo al autor del presente trabajo) fueron consultados para captar sus creencias y acciones que siguen al enfrentar situaciones comunes durante su andar por el medio que les rodea. Con las observaciones obtenidas a partir de las encuestas realizadas, fue posible generar las primeras reglas a partir de las cuales surgieron más. De todas las reglas generadas, algunas sufrieron modificaciones, otras fueron descartadas y finalmente nuevas reglas fueron añadidas durante las etapas 6 y 7, " Pruebas y ajuste del sistema", etapas que serán descritas en capítulos posteriores.

Otro de los puntos importantes, considerados al elaborar las reglas para controlar la navegación de un robot móvil, es que para su generación únicamente se consideraron los conjuntos difusos con sus respectivos adjetivos elaborados en secciones anteriores (tabla 4.2). Las reglas generadas son presentadas en las siguientes páginas en dos diferentes formatos. En forma de tablas y mediante sentencias IF-THEN. Para poder interpretar la información contenida en las tablas 4.3 a 4.6, es necesario observar la figura 4.10 donde se

muestra la manera de elaborar las reglas necesarias, utilizadas por la máquina de inferencia para que un robot móvil pueda navegar y alcanzar su objetivo de manera segura y autónoma.

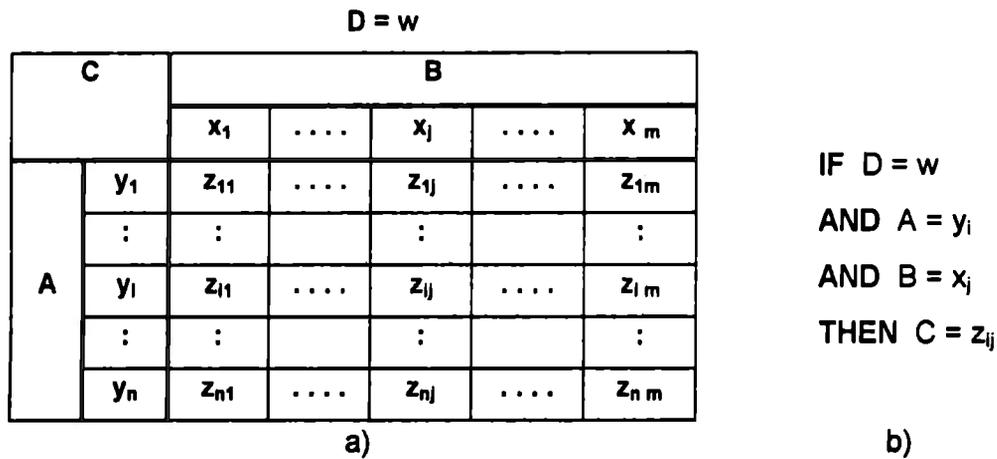


Fig. 4.10 a) Representación general de las reglas en forma de tablas y b) Forma general de las reglas generadas a partir de la información contenida en la tabla de la fig. 4.10a.

En la generación de las reglas que ha continuación se presentan, se considero para su elaboración que el robot debe ser capaz de maniobrar eficientemente a través de situaciones típicas como las que se presentan en la figura 4.11.

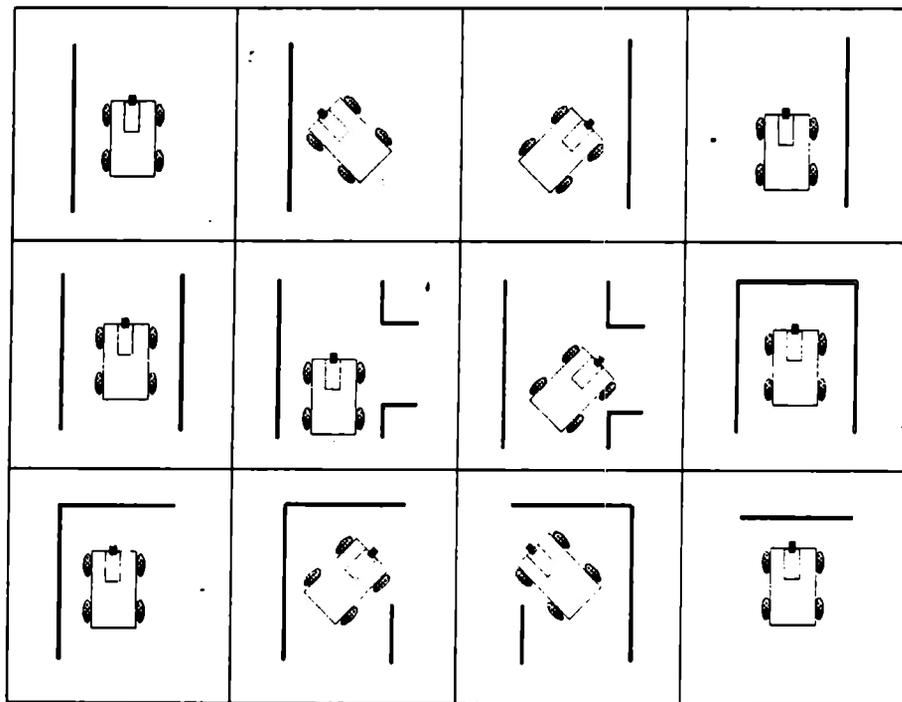


Fig. 4.11 Situaciones típicas que se le presentan a un robot durante la navegación y que éste debe ser capaz de resolver eficientemente.

A continuación se presenta el conjunto de reglas (en sus dos formatos, tablas y sentencias IF-THEN) que se ha decidido son necesarias para controlar la navegación de robots móviles mediante lógica difusa, y las cuales fueron elaboradas como anteriormente se describió. El número total de reglas generadas con las cuales es posible indicar al sistema (robot móvil) la acción a seguir, es de 47 (cuarenta y siete), 35 (treinta y cinco) para controlar la dirección del robot y 12 (doce) para controlar la velocidad.

Tabla 4.3 Reglas para cambiar la dirección del robot y evadir obstáculos tomando la distancia y el ángulo al que se encuentra el obstáculo (θ_1) y el ángulo error (θ_2).

Distancia del Obstáculo = Cerca (CE)

Ángulo de Dirección (θ_3)		Ángulo Error (θ_2)				
		IG	IP	C	DP	DG
Ángulo del Obstáculo (θ_1)	IG	DG				
	IP					
	C	IG				
	DP					
	DG					

Tabla 4.4 Reglas para cambiar la dirección del robot y evadir obstáculos tomando la distancia y el ángulo al que se encuentra el obstáculo (θ_1) y el ángulo error (θ_2).

Distancia del Obstáculo = Mediana (M)

Ángulo de Dirección (θ_3)		Ángulo Error (θ_2)				
		IG	IP	C	DP	DG
Ángulo del Obstáculo (θ_1)	IG	IP		DP		DG
	IP	IG	C			
	C		IP	C		
	DP					
	DG	DP				

Tabla 4.5 Reglas para cambiar la dirección del robot y evadir obstáculos tomando la distancia y el ángulo al que se encuentra el obstáculo (θ_1) y el ángulo error (θ_2).

Distancia del Obstáculo = Lejos (LE)

Ángulo de Dirección (θ_3)		Ángulo Error (θ_2)				
		IG	IP	C	DP	DG
Ángulo del Obstáculo (θ_1)	IG	IP		C	DP	DG
	IP	IG	IP			
	C					
	DP				DP	
	DG	DP				

Tabla 4.6 Reglas para controlar la velocidad del robot tomando en consideración la distancia del robot al punto meta y la distancia a la que se encuentran los obstáculos.

Velocidad (del robot)		Distancia del Obstáculo			
		C	CE	M	LE
Distancia del robot al punto meta	C	C			
	CE	C	RL		
	RE		L		
	M		L	M	
	LE		RA		

A continuación se presenta una serie de reglas en su formato de sentencias IF-THEN que sirven como ejemplo de las reglas que se generan a partir de las tablas anteriores. Las 47 reglas necesarias para controlar de manera adecuada la navegación de un robot móvil son presentadas en el anexo A.

Algunas de las reglas que se obtienen a partir de las tablas 4.3, 4.4 y 4.5 son las siguientes:

IF Distancia del Obstáculo es CERCA
 AND Ángulo del Obstáculo es IZQUIERDA GRANDE
 THEN Ángulo de Dirección es DERECHA GRANDE
 IF Distancia del Obstáculo es CERCA
 AND Ángulo del Obstáculo es CERO

IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA GRANDE
AND Ángulo error es IZQUIERDA GRANDE
THEN Ángulo de Dirección es IZQUIERDA PEQUEÑA

IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA GRANDE
AND Ángulo error es DERECHA PEQUEÑA
THEN Ángulo de Dirección es DERECHA PEQUEÑA

IF Distancia del Obstáculo es LEJOS
AND Ángulo error es CERO
THEN Ángulo de Dirección es CERO

Algunas de las reglas que se obtienen a partir de la tabla 4.6 son las siguientes:

IF Distancia del robot al punto meta es CERO
THEN Velocidad (del robot) es CERO

IF Distancia del robot al punto meta es MEDIANA
AND Distancia del Obstáculo es LEJOS
THEN Velocidad (del robot) es MEDIANA

IF Distancia del robot al punto meta es LEJOS
AND Distancia del Obstáculo es LEJOS
THEN Velocidad (del robot) es RÁPIDO

4.5 ETAPAS 5, 6 y 7: CONSTRUCCIÓN, PRUEBA Y AJUSTE DEL SISTEMA (robot móvil)

Las tres últimas etapas en el diseño del controlador de navegación difuso el cual permita a un robot móvil navegar de manera autónoma son de suma importancia en el desarrollo del presente trabajo. Dada la importancia, la gran cantidad de información y el análisis riguroso y detallado que se requiere realizar y mostrar para cada una de estas tres etapas, cada una de estas 3 etapas se describirá en los siguientes capítulos.

5 CONSTRUCCIÓN DEL SISTEMA

En el presente capítulo se describirá la quinta parte en la elaboración del controlador difuso (Fuzzy - controller): la construcción del sistema. Este se divide en dos fases o etapas. La primera es la simulación y elaboración del software, mientras que la segunda es la construcción del prototipo (robot experimental) con el cual se realizarán pruebas para observar el comportamiento del controlador difuso en ambientes reales.

5.1 ORGANIZACIÓN DEL CONTROLADOR DIFUSO

El principal objetivo de un sistema de control es el de modificar el comportamiento o estado futuro del sistema sobre el que actúa, modificando las entradas al sistema conforme a una serie de reglas [AR-59]. Este conjunto de reglas conforman el modelo que describe la manera en que el sistema opera. La teoría de control clásico hace uso de modelos matemáticos los cuales definen la relación que transforma el estado actual del sistema en una serie de entradas que afectarán su futuro, tal como puede apreciarse en la figura 5.1.

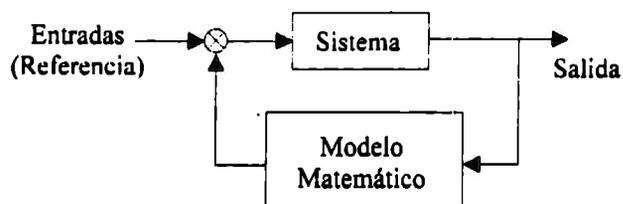


Fig. 5.1 Diagrama de un sistema controlado mediante la teoría de control clásico.

Sin embargo, el mayor problema con la teoría clásica de control, es de que usualmente ésta asume que los sistemas son o se comportan de manera lineal (comportamiento monotónico) [TX-18]. Pero al aumentar la complejidad de los sistemas, como es el caso especial de los robots móviles, se torna sumamente difícil obtener el modelo matemático que lo represente, por lo que estos sistemas son sumamente difíciles de controlar por estos medios. Esto se torna aún más difícil al tratar el problema de la navegación de vehículos autónomos en ambientes desconocidos, los que además deben ser capaces de manejar información imprecisa (i.e. con ruido) proveniente de los sensores (que pueden sufrir desajustes). Debido a esto deben emplearse nuevas técnicas de control, sobre todo, si el sistema de control debe emplearse en tiempo real. La necesidad de manipular información vaga e imprecisa proveniente de los sensores es otra de las razones, además de las descritas en capítulos anteriores, por las cuales se decidió elaborar un controlador difuso para el problema que se desea resolver (capítulo 1).

Los controladores difusos basados en sistemas de conocimientos son sistemas que emplean una base de conocimientos expresada en términos de reglas de inferencia difusas (fuzzy inference rules) y de una máquina de inferencia que utiliza la base de conocimientos para resolver un problema específico de control, que en el caso del presente trabajo es el de la navegación autónoma de robots móviles. Este tipo de controladores reciben información difusa y generan señales adecuadas de control. Dicho enfoque está representado de forma esquemática en la figura 5.2, enfoque que se seguirá en el presente estudio.

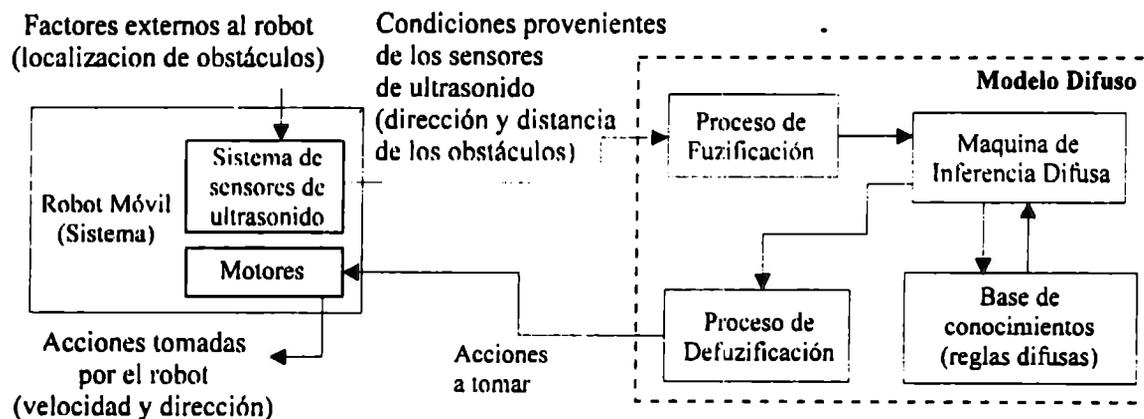


Fig. 5.2 Diagrama del controlador difuso para el caso específico de la navegación autónoma de robots móviles.

Con base en los lineamientos anteriores se definió el esquema del sistema de navegación a diseñar el cual se muestra en la figura 5.3.

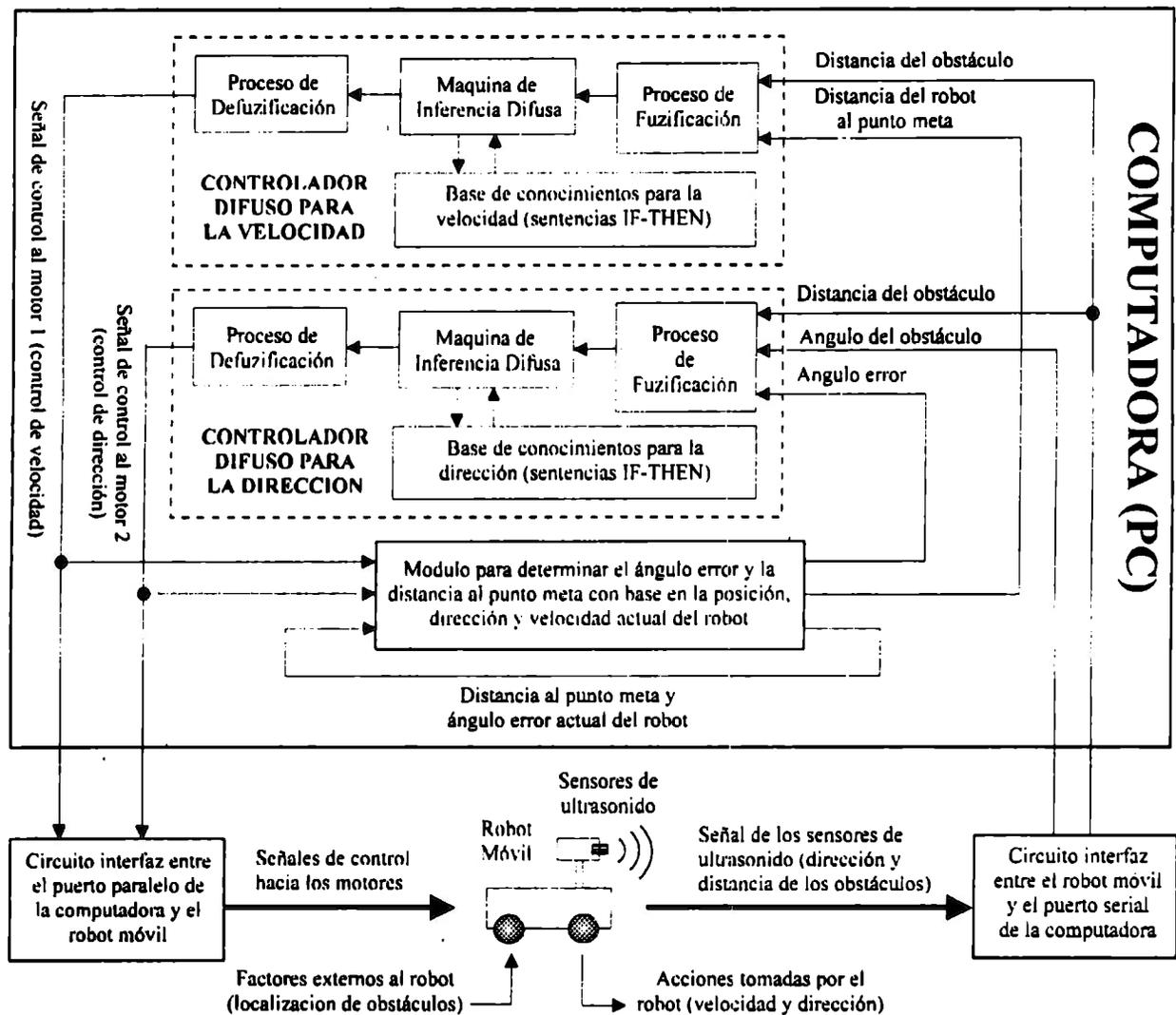


Fig. 5.3 Diagrama esquemático del sistema para controlar la navegación de un robot móvil mediante lógica difusa. El diagrama muestra la parte de software (computadora PC) y la parte de hardware (robot móvil experimental).

Como puede apreciarse en la figura 5.3 el sistema de navegación difuso esta dividido en dos controladores. Un controlador difuso para la velocidad y otro para la dirección del robot. El controlador difuso para la velocidad recibe como entradas la distancia del obstáculo y la distancia del robot al punto meta. Por su parte el controlador difuso para la dirección recibe como entradas la distancia a la que se encuentran los obstáculos detectados, el ángulo relativo al robot de cada uno de los obstáculos detectados y el ángulo error (descrito en el capítulo número 2) entre la dirección actual del robot y el punto meta. Ambos controladores (el de velocidad y el de dirección) al recibir sus respectivas entradas las fuzifican y transmiten la información fuzificada a su correspondiente máquina de inferencia en donde se lleva a cabo una inferencia tipo forward-chaining. La máquina de inferencia

interactúa con la base de conocimientos (reglas difusas) con lo que esta genera una respuesta a las entradas recibidas. La respuesta generada en forma difusa por cada una de las máquinas de inferencia es defuzificada mediante una defuzificación tipo centroidal y transmitida al robot el cual lleva a cabo las acciones recibidas. El controlador difuso almacenado en una computadora personal se comunica con el robot mediante interfaces (circuitos) las cuales son explicadas más adelante dentro del presente capítulo.

Una vez definido el esquema del controlador de navegación difuso, el siguiente paso es elaborar el software y hardware (prototipo del robot móvil) necesario para llevar a cabo simulaciones y pruebas experimentales del controlador de navegación diseñado. Se comenzará describiendo la elaboración del sistema en su parte de software con el cual se llevaron a cabo las simulaciones y control del robot prototipo.

5.2 CONTROLADOR DIFUSO (Software)

La elaboración del software necesario para llevar a cabo las simulaciones y pruebas experimentales del controlador de navegación difuso diseñado en el capítulo 4, se dividió en dos partes. La primera parte se dedicó a la simulación del sistema, vía el empleo de herramientas comerciales tales como MATLAB © y TILShell © entre otros (con las cuales también fue posible ajustar el sistema). La segunda parte se destinó a la programación en lenguaje C++ del controlador previamente simulado de tal forma que éste es capaz de controlar un robot móvil experimental como se describirá más adelante. Tal programa posee la capacidad de controlar la navegación de un robot móvil experimental mediante la comunicación del sistema controlador (controlador difuso) alojado en la memoria de una computadora personal (PC) con el robot vía los puertos serial y paralelo de la computadora.

5.2.1 PRIMERA FASE (Simulación)

Como ya se menciona, la primera parte en el desarrollo del software es la simulación del controlador de navegación difuso desarrollado en el capítulo 4 "Elaboración del Controlador Difuso". Con el fin de agilizar dicho proceso y obtener resultados de manera rápida y eficiente, la simulación se llevó a cabo con ayuda del paquete computacional "Fuzzy

Systems Toolbox" de MATLAB © para PC [TX-19], con el cual es posible definir variables y conjuntos difusos, manipular reglas, utilizar operadores difusos y realizar varios tipos de defuzificaciones entre otras cosas como son estudios de sistemas de control y reconocimiento de patrones. Los resultados obtenidos a partir de las simulaciones realizadas con la mencionada herramienta son descritos y presentados en el siguiente capítulo, por lo que se dejará hasta entonces la discusión de los resultados obtenidos en el desempeño del controlador. En esta sección simplemente se mencionará que a pesar de que existen varios otros paquetes computacionales en el mercado con los cuales es posible diseñar sistemas difusos (Tilshell de Togai Infra Logic ©, FuzzyCalc ©, etc.) el Toolbox de MATLAB © fue seleccionado entre todos ellos por varias razones, entre las cuales destacan su bajo costo, presenta capacidades gráficas, se pueden generar funciones de pertenencia, es posible utilizar diversos tipos de operadores así como definir propios y finalmente, es posible representar reglas mediante tablas de manera sencilla y amigable para el usuario. Con la ayuda de este paquete también fue posible realizar ajustes al controlador de navegación difuso, de tal forma que cumpliera con los objetivos planteados inicialmente.

De los dos métodos de inferencia estudiados (Max-Min y Max-Product) la técnica empleada durante la simulación fue el método de inferencia Max-Min. A pesar de que en general es más fácil programar el método Max-Product y de que durante los primeros resultados en el estudio de simulación se observó que no existía diferencia apreciable en los resultados generados entre ambas metodologías, se decidió utilizar para el resto del estudio la técnica de inferencia Max-Min debido a dos razones principales. La primera es que es una técnica conservadora. También en general (según varios autores [TX-1], [TX-21]) es ideal para cuando se estudia el empleo de lógica difusa en sistemas por primera vez en donde se desconoce totalmente el comportamiento del sistema bajo un controlador difuso. Además fue necesario utilizar una sola técnica de inferencia dado que durante los inicios del estudio no fue posible definir cual técnica de inferencia es la más apropiada puesto que aún falta ajustar y probablemente redefinir los conjuntos difusos así como también eliminar o agregar reglas a la base de conocimiento.

Una vez determinado el tipo de inferencia a utilizar (Max-Min Forward-Chaining), el sistema de simulación evalúa todas las reglas contenidas dentro de la base de conocimiento con base en la información proporcionada por los sensores, cuyo comportamiento por el momento también es simulado y cuyos datos son obtenidos mediante la lectura constante de un archivo (o matriz) de ceros y unos (presencia y ausencia de obstáculos respectivamente)

que representa el terreno por el cual el robot navegará. Una vez evaluadas todas las reglas con base en la información suministrada en un instante dado por los sensores de ultrasonido y obtenidos los resultados por la máquina de inferencia de cada regla que fue satisfecha, el sistema suma todos los resultados parciales y realiza una defuzificación sobre el resultado para obtener la señal de control correspondiente (velocidad o dirección del robot), similar al proceso de la figura 3.4b y que es representado esquemáticamente en la figura 5.4.

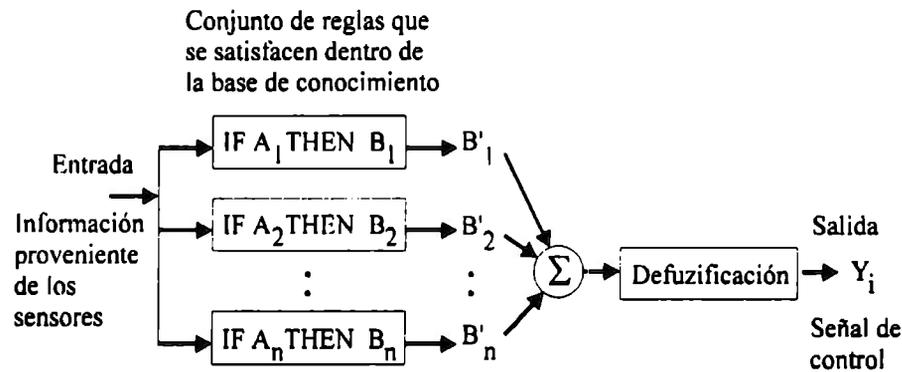


Fig. 5.4 Proceso de inferencia y defuzificación sobre todas las reglas que se satisfacen dentro de la base de conocimiento.

Para obtener el único valor de salida que corresponde a la señal de control, después de haber realizado el proceso de inferencia, se efectúa un proceso de defuzificación. De los tantos métodos de defuzificación existentes hoy en día y de los cuales algunos fueron descritos en la capítulo número 3, en el presente trabajo se utilizó el método de defuzificación centroidal. Este método fue seleccionado debido a que es uno de los pocos métodos de defuzificación en el cual existen cambios moderados entre salidas o resultados adyacentes de dos inferencias (existe continuidad). Esto no siempre sucede para otros métodos donde pequeñas modificaciones en las señales de entrada al controlador difuso pueden causar grandes cambios en las señales de control. Esto es de suma importancia para la navegación de sistemas autónomos donde un cambio brusco en una señal de control puede causar que el vehículo no logre su objetivo (i.e. si el robot móvil se encuentra cerca de un obstáculo un cambio brusco en su velocidad o en su dirección puede causar que el robot choque con el obstáculo). El diagrama de flujo que representa la secuencia de pasos que sigue el controlador difuso simulado se muestra en la figura C.1 dentro del anexo C.

5.2.2 AJUSTE DEL SISTEMA

Una vez definido los principales componentes del controlador difuso (conjuntos difusos, reglas, tipo de inferencia y método de defuzificación) es necesario probar y ajustar el controlador para que trabaje de la manera en que se desea. Básicamente existen dos formas de ajustar un controlador difuso después de ver su funcionamiento. Puede ajustarse modificando las reglas de su base de conocimiento o puede ajustarse mediante la modificación de los conjuntos difusos que lo componen.

Ambas formas de ajuste son procesos lentos y tardados. La primer forma (modificar las reglas) es peligrosa dado que al modificar las reglas se corre el riesgo de modificar las creencias y conocimientos del experto con los que fueron elaboradas las reglas, por lo que este método no fue empleado dentro del presente trabajo. Para la segunda forma (modificar los conjuntos difusos) existen varias formas de realizar el ajuste. Un ajuste puede lograrse mediante la modificación en la forma de representar los conjuntos difusos (i.e. cambiar un conjunto difuso triangular a trapezoidal), otra forma sería desplazar los conjuntos difusos a la derecha o a la izquierda sin modificar su forma hasta lograr los ajustes deseados. Otra manera de realizar un ajuste del sistema mediante la modificación de los conjuntos difusos que componen el controlador, es utilizar "Hedges" (descritos en el capítulo 3) con los cuales se puede incrementar o decrementar el efecto de un conjunto difuso sobre el sistema. Finalmente el ajuste puede realizarse mediante una combinación de varias de éstas metodologías e incluso mediante el empleo de otras técnicas computacionales como son la utilización de algoritmos genéticos para la optimización tanto de las reglas como de los conjuntos difusos.

La metodología empleada aquí, fue la de desplazar (a la derecha o a la izquierda) y la de modificar la forma de los conjuntos difusos (siempre conservando formas triangulares con las cuales se reducen los cálculos) hasta lograr el ajuste deseado. Como se menciono anteriormente, la modificación de las reglas fue descartado como medio de ajuste por temor a modificar las creencias y conocimientos con las cuales fueron elaboradas. Por otra parte, el uso de algoritmos genéticos como medio de optimización tanto de reglas como de conjuntos difusos fue descartada como medio de ajuste del sistema, no por su poco empleo en este tipo de sistemas y a la dificultad de definir de manera correcta el conjunto de parámetros necesarios para llevar acabo la optimización (además de definir de manera correcta como y donde llevar acabo los operadores de "crossover" y de "mutación"), sino porque éste podría ser otro trabajo de tesis debido al gran trabajo de pruebas que hay que realizar. Sin embargo

sería interesante comparar los resultados que pudieran generarse mediante el empleo de estos algoritmos como medios de ajuste del sistema con los resultados aquí obtenidos.

5.2.3 SEGUNDA FASE (Generación del código en lenguaje C++ para controlar un robot experimental utilizando el controlador difuso diseñado)

La segunda parte o etapa en la elaboración del software consistió en realizar un programa en lenguaje C++ (con base en el controlador difuso simulado y ajustado anteriormente) bajo plataforma PC con el cual es posible controlar un robot móvil experimental (sección 5.3) mediante la comunicación del robot con la computadora (la cual almacena el controlador), a través de los puertos serial y paralelo de la misma como se muestra en las figuras 5.3 y 5.5.

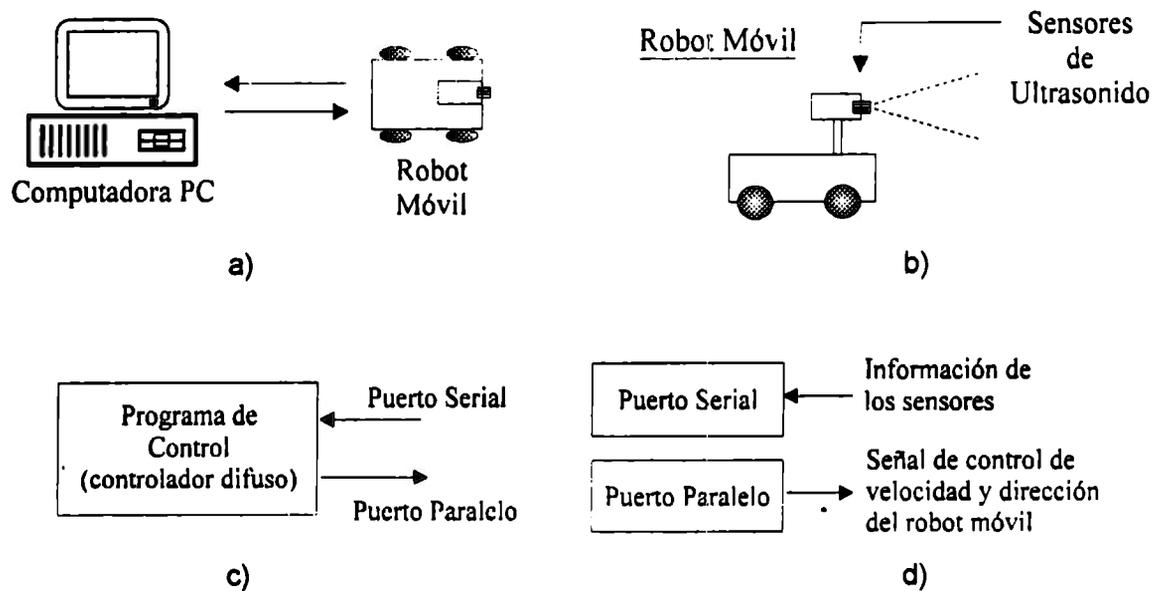


Fig. 5.5 Comunicación entre el controlador difuso almacenado en la memoria de una computadora y el robot móvil a través de los puertos serial y paralelo.

Para la elaboración de éste programa se siguió el paradigma de modelado/planeación basado en las ideas de llevar a acabo una planeación tomando como base el modelo del mundo generado a partir de información sensorial [TX-3], paradigma empleado desde los inicios de la robótica y el cual actualmente es un área de intensa investigación (i.e. sistema HANDEY, MIT AI Lab.). Dicho enfoque divide el programa de control en un conjunto ordenado de componentes funcionales como se ilustra en la figura 5.6. Como puede observarse en dicha figura los datos con los cuales se crea un modelo del mundo son obtenidos a través de sensores. Dicho modelo puede incluir detalles geométricos de los

objetos (formas, dimensiones, etc.) o simplemente incluir la posición y orientación de cada objeto. Dentro del presente trabajo se seguirá el enfoque de modelar el mundo (desconocido a priori por el robot) utilizando únicamente la dirección y distancia de los obstáculos respecto al robot. Esto debido al tipo de controlador y sensores empleados, con los cuales es difícil (no imposible) captar las formas geométricas de los objetos. Realizado el modelo del mundo que rodea al robot y dado un objetivo, generalmente proporcionado por un ser humano (operario), el robot utiliza "su modelo" del mundo para planear una serie de acciones con las cuales podrá lograr la meta, que en nuestro caso consiste en llegar a un punto final desde un punto inicial sin sufrir colisiones. Este conjunto de acciones serán proporcionados por el controlador difuso. Finalmente, el plan formulado por el controlador es ejecutado por el robot mediante el envío de las acciones apropiadas a seguir (plan) a los actuadores del robot.

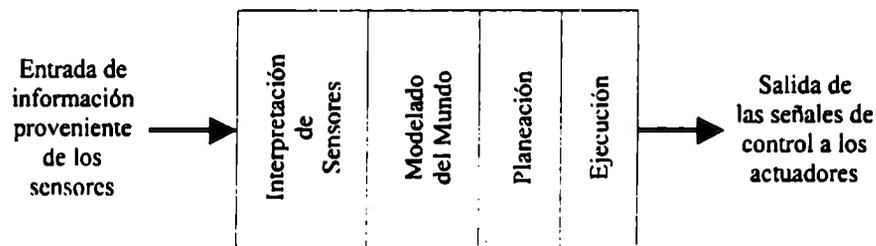


Fig. 5.6 Componentes funcionales dentro de un programa necesarios para controlar las acciones de un robot siguiendo el paradigma de modelado/planeación. Estas unidades funcionales transforman un instante de datos sensoriales en una serie de acciones destinadas a cumplir una meta específica.

Como se menciona anteriormente la obtención de la información del mundo y el envío de ésta (acciones a seguir) se realiza a través de los puertos de una computadora PC. Por sencillez y para evitar conflictos de sincronización (entre otros posibles problemas) con la comunicación a través de un solo puerto, se decidió utilizar ambos puertos de la computadora como medio de comunicación entre el controlador difuso y el robot. El puerto paralelo de la computadora es utilizado exclusivamente para transmitir las señales de control generadas por el controlador difuso a los motores (descritos en la sección 5.3) del robot mediante los cuales se controla la velocidad y dirección del mismo. El puerto serial es utilizado únicamente como puerto de entrada. Por medio de éste puerto el controlador difuso es capaz de recibir las señales provenientes de los sensores de ultrasonido (dirección y distancia de los obstáculos relativos al robot) mediante los cuales el vehículo autónomo percibe su entorno.

El programa desarrollado está integrado por varios módulos (o funciones) los cuales componen el controlador de navegación difuso como puede apreciarse en las figuras 5.2 y 5.3. Los principales módulos con los que cuenta el programa y de los cuales algunos de ellos están descritos en el anexo C (código) son los siguientes:

- Un módulo para llevar a cabo la fuzificación de la información proveniente de los sensores de ultrasonido obtenida a través del puerto serial de la computadora.
- Un módulo que contiene la máquina de inferencia (método Max-Min) con la cual el controlador difuso realiza la decisión sobre la nueva velocidad y dirección del robot con base en información proveniente de los sensores en forma difusa (generada en el módulo de fuzificación) y la base de conocimiento.
- Un módulo que contiene (o almacena) la base de conocimiento en forma de sentencias IF-THEN la cual puede ser modificada por el usuario.
- Un módulo de defuzificación (defuzificación centroidal) el cual transforma las decisiones tomadas por el módulo de inferencia a un solo valor de control. Los valores generados por el módulo de defuzificación son transmitidos al robot a través del puerto paralelo.
- Un módulo de comunicación entre la computadora y el robot móvil. La comunicación se lleva a cabo a través de los puertos serial y paralelo de la computadora, por lo que existen dos módulos de comunicación, uno para el envío de información a través del puerto paralelo y otro módulo para la lectura de información a través del puerto serial.

Finalmente, el programa realizado es capaz de recibir información proveniente de los sensores del robot, generar acciones con base en dicha información (inferencia) y transmitir dichas acciones al robot en tiempo real (lo suficientemente rápido para que el sistema capte información, la procese y actúe casi instantáneamente antes de captar nueva información, procesarla y actuar). El diagrama de flujo del programa desarrollado se muestra en la figura C.1 (anexo C). Obviamente aquí el concepto de "rápido" y de "tiempo real" depende enormemente de la velocidad máxima que puede ser alcanzada por el robot móvil. Si el robot se desplaza a grandes velocidades pudiera ser que la velocidad del controlador no sea lo suficiente rápido para poder controlar el sistema en tiempo real, sin embargo, considerando las velocidades a las que se mueve nuestro robot experimental (descrito en la siguiente sección) el controlador es capaz de actuar en tiempo real de manera satisfactoria.

5.3 PROTOTIPO DEL ROBOT MÓVIL (Hardware)

Una vez realizadas las pruebas de simulación utilizando el Fuzzy Logic Toolbox para PC de MATLAB ©, donde se observó que el problema de navegación de robots móviles en ambientes estáticos desconocidos es resuelto de manera satisfactoria utilizando lógica difusa (lo que se mostrará en el siguiente capítulo), se construyó un prototipo (robot móvil) el cual es controlado mediante el controlador de navegación difuso desarrollado en lenguaje "C++". La figura 5.3 muestra de forma esquemática la manera en que está organizado el sistema con el cual se realizaron las pruebas experimentales para comparar estas con los resultados obtenidos de la simulación.

No es nuestro propósito describir paso a paso y de manera detallada como se construyó el robot prototipo ya que no es el propósito del presente trabajo, sin embargo, se proporcionará un panorama general con el cual sea posible conocer su funcionamiento (y probablemente poder construir su propio robot móvil).

En los siguientes párrafos se muestran y se describen los diagramas electrónicos así como también se mencionan los dispositivos con los cuales se construyó el robot experimental y mediante los cuales el sistema es controlado (enviar y recibir información de la computadora).

INTERFASE PC-ROBOT

Una interfase es un dispositivo o medio por el cual existe comunicación entre dos o más dispositivos. El diagrama de la figura 5.7 es el circuito encargado de proporcionar las señales de control provenientes del puerto paralelo de la computadora a cada uno de los motores que posee el robot móvil, es decir, es el circuito interfase entre la computadora y el robot. El robot posee dos circuitos similares, uno para controlar el motor de dirección y otro para el motor de velocidad.

Las señales de control (velocidad y dirección) obtenidas por el controlador difuso son suministradas al robot por medio de una señal digital de 8 bits. Seis de estos ocho bits (los menos significativos) son introducidos a los dos convertidores digital-analógico (D/A), uno para cada motor, los cuales transforman el código digital a una señal analógica la cual es suministrada a un amplificador operacional (OpAmp) configurado como comparador. Los restantes 2 bits (provenientes del puerto paralelo) son empleados para seleccionar y habilitar el convertidor D/A que recibirá la información dependiendo si es una señal de velocidad o de dirección. Junto con la señal analógica (salida del convertidor D/A), el OpAmp recibe una

onda triangular. Ambas señales son comparadas tal como se muestra en la figura 5.8 donde también se muestra la señal de salida proporcionada por el OpAmp (UIA) después de haber realizado la comparación de las señales correspondientes.

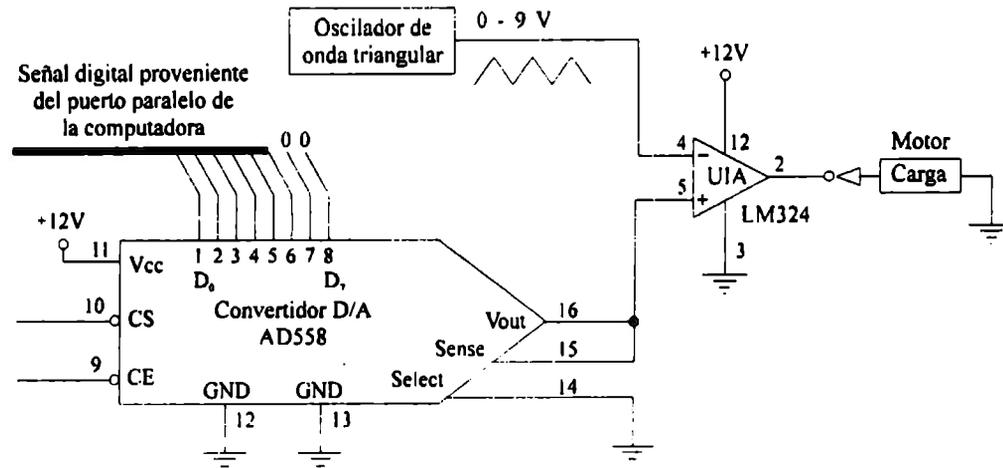


Fig. 5.7 Interfase Computadora-Robot Móvil vía el puerto paralelo. Cada uno de los dos motores a ser controlado por la computadora utiliza un circuito similar.

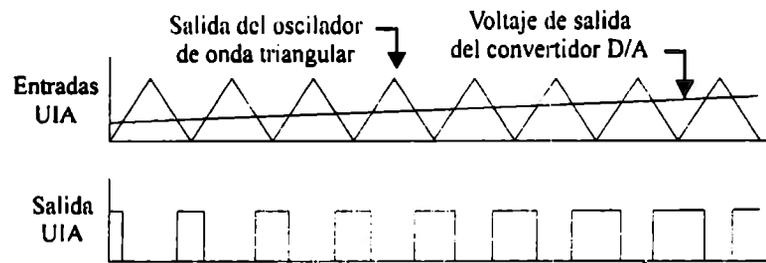


Fig. 5.8 Señales de entrada y salida del amplificador operacional (UIA) mediante las cuales se determina la posición que deben adoptar los motores.

El ancho del tren de pulsos que genera el amplificador operacional depende directamente de la señal analógica proporcionada por el convertidor D/A (que a su vez depende de la señal digital proporcionada por el controlador difuso a través del puerto paralelo de la computadora). Este tren de pulsos de ancho variable (pulse width) es la señal que indica a los motores la posición que deben adoptar.

MOTORES

El tipo de motores utilizados en el robot prototipo son servomotores que funcionan a través de un ancho de pulso modulado (pulse width modulation) los cuales adoptan ciertas

posiciones dependiendo del ancho del pulso que reciben. Básicamente este tipo de servo espera recibir un tren de pulsos de ancho variable (proporcionado por el OpAmp) a determinada frecuencia (el valor típico para este tipo de servos es de 1/20 Hz). El ancho del pulso es el código que indica al motor la posición que su flecha (o eje) debe adoptar. Los servos empleados operan entre anchos de pulsos de 0.2 ms. y 1.0 ms. Si el servo recibe un ancho de pulso de 0.2 ms. o menor, éste se moverá a su posición extrema derecha mientras que anchos de pulso mayores o iguales a 1.0 ms. provocarán que el servo se mueva a su posición extrema izquierda, tal como se muestra en la figura 5.9.

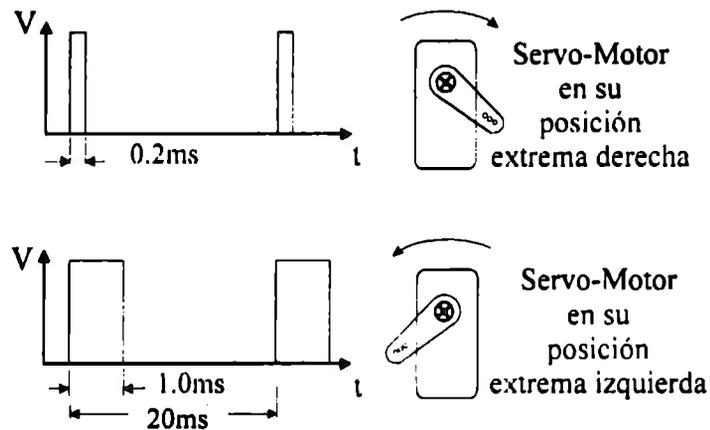


Fig. 5.9 Servomotores utilizados en el robot prototipo. Cada motor consta de un sistema de engranes y de circuitos apropiadas para su control a través de un ancho de pulso modulado. Además los motores poseen tres cables, uno para la alimentación, otro para la conexión a tierra y el tercer cable es la entrada del pulso cuyo ancho determina la posición designada del motor. Pulsos anchos o delgados determinan la posición a la cual se deberá posicionar el servo, ya sea en el sentido o en contra del movimiento de las manecillas del reloj referentes a su posición central.

Anchos de pulsos entre 0.2 ms. y 1.0 ms. provocarán que el servo se mueva a una cierta posición definida por el ancho del pulso (i.e. si el servo recibe un ancho de pulso de 0.7 ms. éste se moverá a su posición central sin importar en que posición se encuentre). Uno de los motores controla la dirección del robot (motor delantero) mientras que el segundo motor controla la velocidad del vehículo mediante el ajuste de una resistencia variable que controla la corriente suministrada a un tercer motor que impulsa al robot (éste tercer motor es independiente del sistema de control). La corriente máxima suministrada al tercer motor es manipulada por el usuario mediante una resistencia variable como puede apreciarse en la figura 5.10.

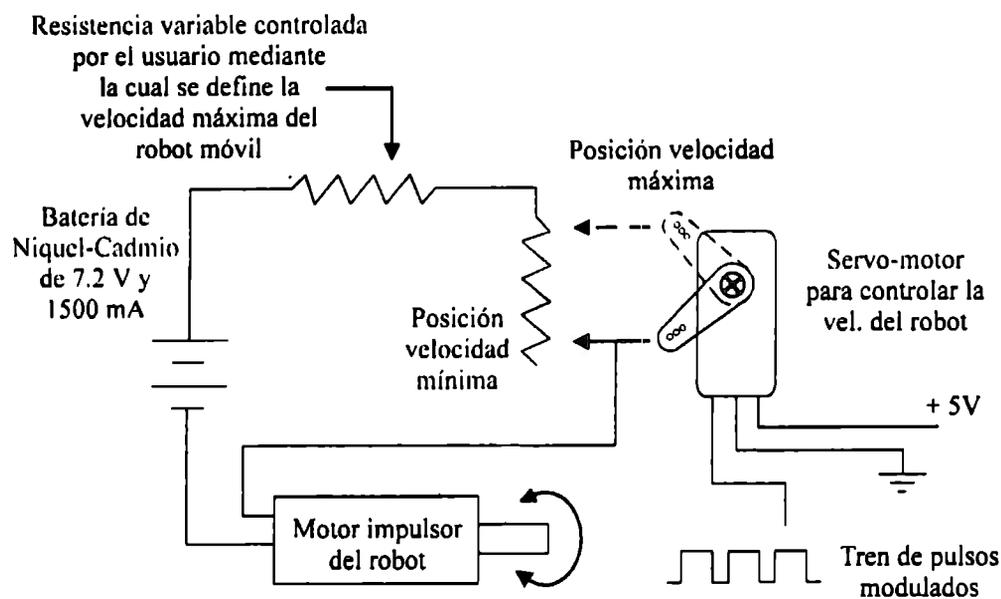


Fig. 5.10 Configuración del motor requerido para controlar la velocidad del robot.

SENSORES DE ULTRASONIDO

Como se ha expuesto en capítulos anteriores, el robot posee un conjunto de siete sensores de ultrasonido con los cuales es capaz de percibir su entorno. Para percibir su entorno, el robot móvil requiere saber la localización de los obstáculos presentes en el ambiente, lo cual se logra utilizando sensores de proximidad. En la actualidad existen varios tipos y modelos de sensores de proximidad (i.e. inductivos, capacitivos, infrarrojos, de ultrasonido, lasers, magnéticos, etc.) [CA-1], [Ca-2], [Co-2], [Co-3] de entre los cuales se decidió utilizar sensores de ultrasonido debido a que poseen un mayor rango de cobertura, mayor sensibilidad y no están restringidos a sensar únicamente ciertos tipo de material como es el caso de los inductivos y los magnéticos. Además, este tipo de sensores pueden utilizarse para percibir objetos a grandes distancias, característica que no poseen otros dispositivos.

El funcionamiento general de un sensor de ultrasonido es el de simplemente generar ondas mecánicas a frecuencias ultrasónicas (mayores a 20 KHz). Una vez generadas las ondas ultrasónicas, estas viajan por el medio (sólido, líquido o gaseoso) hasta chocar con algún objeto. Si las ondas ultrasónicas encuentran algún objeto en su trayecto, éstas rebotarán en dicho objeto generando un eco. Este eco es captado por el sensor de ultrasonido el cual medirá la diferencia entre el tiempo que existe entre el momento en que se generó la señal (onda ultrasónica) y el tiempo en el que se recibió su eco. Esta diferencia de tiempo permite conocer la distancia "d" a la que se encuentra el objeto que produjo el eco tal como se puede apreciarse en la figura 5.11.

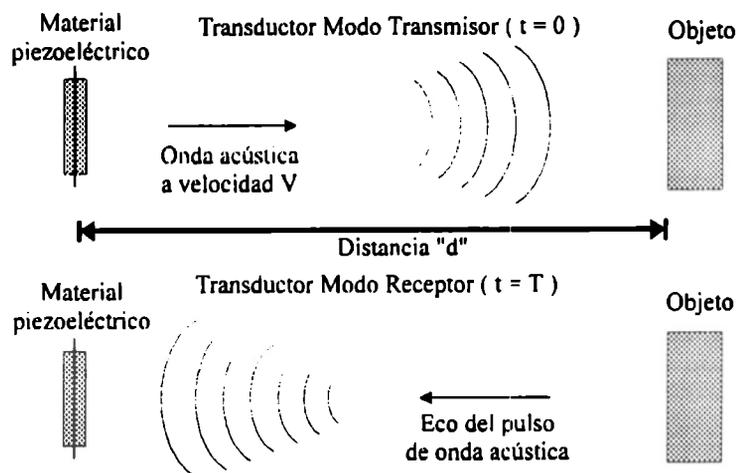


Fig. 5.11 Principio de funcionamiento de un sensor de ultrasonido.

Para determinar con exactitud la distancia del objeto sentido, es necesario conocer la rapidez con la cual viaja el sonido dentro del medio que se este utilizando. En el caso del aire el sonido viaja a una rapidez de 331.3 m/s (a una temperatura del medio de 0 °C).

Para generar ondas ultrasónicas se emplean comúnmente materiales piezoeléctricos como el cuarzo excitados a frecuencias de resonancia (la resonancia se logra cuando la frecuencia natural del sistema es igual a la frecuencia de excitación). En el presente trabajo las ondas ultrasónicas se generaron utilizando transductores electrostáticos de la serie 7000 de Polaroid los cuales pueden ajustarse para captar objetos a una distancia máxima de 10.7 mts. con una resolución de entre ± 3 mm. a 3 mts. y cuyas condiciones de operación se encuentran en el anexo D. La figura 5.12 muestra de forma esquemática el circuito completo el cual implementa un sensor de ultrasonido (se construyeron siete sensores de ultrasonido c/u de la misma forma como se muestra en la figura 5.12). Con este circuito se es capaz de captar objetos a una distancia máxima de 6 mts., sin embargo, los sensores se calibraron para detectar objetos a una distancia máxima de 1.0 mts. con lo que se logra mayor precisión en la distancia sensada.

El transductor de ultrasonido seleccionado (Polaroid 7000 Series Electrostatic Transducer) es controlado por el circuito integrado (CI) LM1812 el cual es un "transceiver" de ultrasonido. El CI LM1812 es capaz de excitar al transductor de ultrasonido a frecuencias de ultrasonido lo cual se logra mediante un simple pulso enviado al pin número 8 de dicho circuito. El LM1812 es capaz de captar el eco recibido por el transductor generando una señal digital (pulso) el cual indica que el eco se ha recibido. Esta señal es generada sobre el pin número 16 como puede verse en la figura 5.12. La ventaja de utilizar éste CI es el de que

se requiere de un solo transductor de ultrasonido con el cual se es capaz de emitir y de recibir las señales (onda ultrasónica y el eco) a diferencia de otros circuitos que requieren de dos transductores, uno como emisor y otro como receptor.

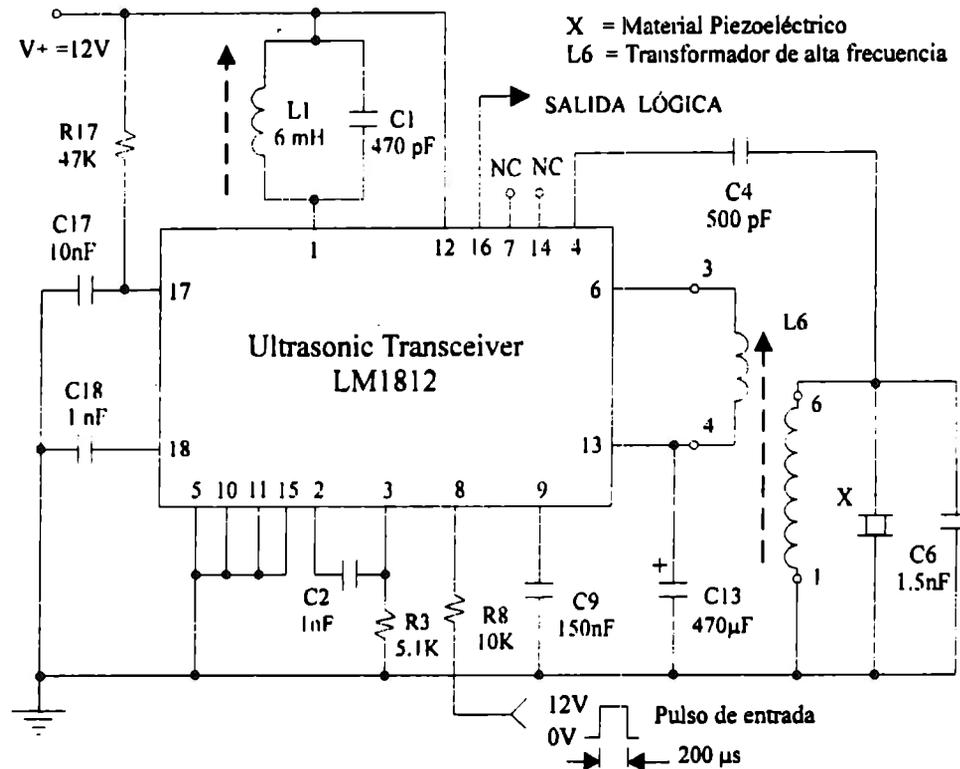


Fig. 5.12 Diagrama esquemático del sensor de ultrasonido.

INTERFASE ROBOT-PC

Una vez que los siete sensores de ultrasonido han captado la presencia (o ausencia) de objetos y generado sus correspondientes señales digitales a través de sus respectivos pines (pin 16), es necesario procesar dichas señales con el fin de determinar de manera exacta la diferencia de tiempo (Δt) entre la señal de excitación hacia los sensores de ultrasonido y cada una de las señales digitales proporcionada por cada sensor. Con esto se conocerá la distancia a la que se encuentran los obstáculos detectados y se podrá transmitir dicha información a la computadora para que sea procesada por el controlador de navegación difuso.

Para lograr este objetivo se decidió emplear un microcontrolador (dispositivo que puede ser programado) el cual es capaz de recibir, procesar y transmitir información a través de sus propios dispositivos inmersos dentro del mismo chip (i.e. CPU, registros, puertos de E/S, etc.). El microcontrolador (MCU) seleccionado es el 8052 de Intel (miembro de la familia

8051 de microcontroladores) el cual posee 32K bytes de memoria RAM, 8K bytes de memoria ROM, capacidades de realizar operaciones de punto flotante, un reloj interno de tiempo real, puertos serial y paralelo, entradas y salidas para controlar diversos dispositivos (i.e. motores, sensores de temperatura, etc.) entre otras muchas más características como es la facilidad de ser programado mediante comandos del lenguaje BASIC (a diferencia de otros MCU que deben ser programados mediante lenguaje ensamblador). Con éste MCU y otros dispositivos, se diseñó y construyó el circuito que se muestra en la figura 5.13. El circuito de la figura 5.13 es una interfase para computadora de uso general con la cual el robot móvil envía la información captada por los sensores de ultrasonido (distancia y dirección de los obstáculos presentes en el ambiente) hacia la computadora a través de su puerto serial para que esta sea procesada por el controlador.

El circuito U1 (MAX232) dentro de la figura 5.13 es un CI de comunicación serial el cual recibe y transmite información desde y hacia el MCU (circuito U2). Los cuatro capacitores C1-C4 ayudan a generar los niveles de voltaje requeridos por el protocolo RS-232 de comunicación serial sin la necesidad de un multiplicador o elevador de voltaje. La memoria RAM (circuito U3) provee almacenamiento externo de 32K bytes para datos, información e instrucciones que no puedan ser almacenados internamente en el MCU. El MCU 8052 selecciona localidades individuales de memoria a través del bus de direcciones (líneas A0-A15), y lee y escribe hacia la memoria (RAM externa) a través del bus de datos (líneas D0-D7). El CI 74LS373 (circuito U4) contiene ocho latches que le permiten al MCU acceder la memoria externa (circuito U3).

Los pines del 1 al 8 constituyen uno de los puertos de uso general (E/S) del MCU. De este puerto uno de los ocho bits (P1.0) es utilizado para enviar la señal de excitación a los sensores (una sola señal de excitación para los siete sensores) mientras que los restantes 7 bits del puerto (P1.1-P1.7) pines 2,3,...,8 están configurados para recibir las señales digitales provenientes de cada uno de los sensores de ultrasonido (pin número 16 de cada uno de los siete CI LM1812). Al recibir las señales de los sensores, el MCU procesa dichas señales y obtiene la distancia a la que los objetos fueron localizados por cada sensor. Obtenidos estos datos, éstos son enviados hacia la computadora a través del circuito MAX232 utilizando el protocolo RS-232 de comunicación serial.

La unión de todos los dispositivos descritos anteriormente forman la parte medular del sistema experimental (robot móvil) el cual puede observarse en la figura 4.2. Las dimensiones físicas del robot son 37 cm. de largo, 24 cm. de ancho y 18 cm. de altura.

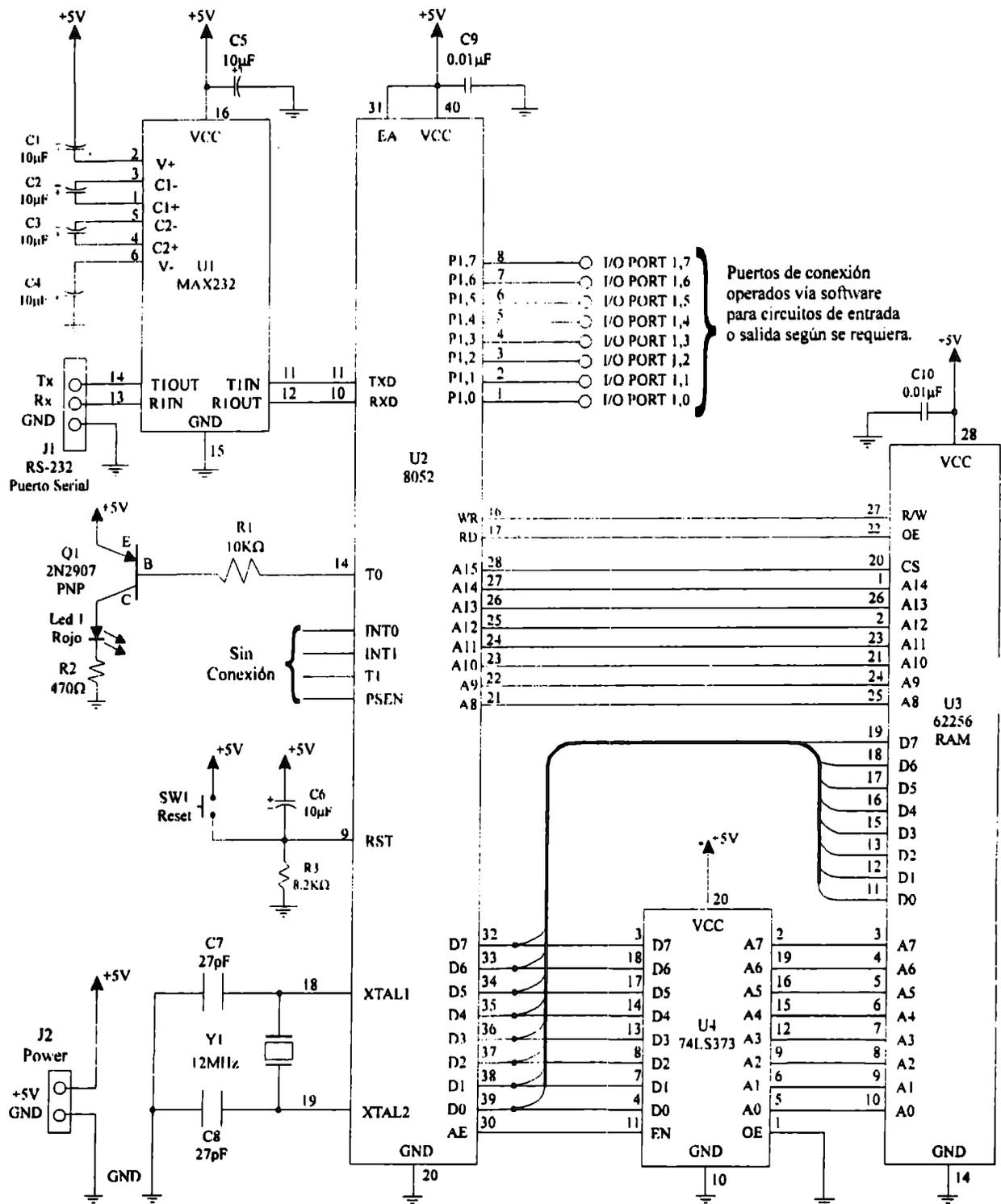


Fig. 5.13 Diagrama de la interfase entre el robot móvil y la computadora.

El robot experimental construido con todas las características antes descritas incluyendo la parte mecánica (i.e. chasis, ruedas, engranes, etc.) y la parte eléctrica y electrónica (i.e. motores, circuitos integrados, etc.) tuvo un costo total de \$2596.60 pesos repartidos conforme se muestra en el siguiente desglose.

<u>DESCRIPCIÓN</u>	<u>CANTIDAD</u>	<u>PRECIO UNITARIO</u>	<u>PRECIO TOTAL</u>
Sistema mecánico (ruedas, soporte, sistema de engranes, suspensión, etc.)	1	\$785.00	\$785.00
Batería de Niquel-Cadmio Recargable	1	\$261.60	\$261.60
Transductores de Ultrasonido	7	\$80.00	\$560.00
Servo Motores	2	\$120.00	\$240.00
Circuito para los sensores de Ultrasonido (CI LM1812, capacitores, resistencias, bobinas, etc.)	7	\$70.00	\$490.00
Interfaz PC-Robot (converidores D/A, OpAmps, etc.)	1	\$110.00	\$110.00
Interfaz Robot-PC (microcontrolador, memoria RAM, capacitores, CI MAX232, resistencias, etc.)	1	\$150.00	\$150.00
TOTAL			\$2596.60

6 PRUEBAS (Comportamiento del controlador)

Algunas de las pruebas experimentales realizadas sobre el controlador de navegación difuso diseñado, empleando el programa elaborado en lenguaje C++ conjuntamente con el robot experimental y los resultados de las simulaciones realizadas después de haber efectuado el ajuste del controlador son presentados en el presente capítulo.

6.1 AMBIENTE Y CONDICIONES DE NAVEGACIÓN

Las simulaciones y pruebas experimentales se realizaron empleando el paquete computacional " Fuzzy Toolbox de MATLAB ©" (versión para PC) y el programa elaborado en lenguaje C++ junto con el robot experimental respectivamente. Todas las pruebas (simulación y experimentación) se llevaron a cabo en un área de 5 x 10 mts. en la cual el robot tenía que navegar desde un punto inicial " A" hasta un punto final " B" (figuras 6.4 a 6.10) evitando obstáculos. En la simulación, el ambiente por el cual el robot navegaría fue representado por una matriz de ceros (0) y unos (1), con los cuales se indicaba la ausencia o presencia de obstáculos respectivamente. Esta matriz era constantemente leída por MATLAB © simulando de éste modo el funcionamiento de los sensores de ultrasonido en el sistema real. En el caso de la experimentación, el ambiente desconocido de navegación fue una superficie plana de 5 x 10 mts. en la cual se colocaron cajas de cartón como obstáculos.

Dado que el robot no cuenta con sensores de velocidad y dirección (de sus ruedas) estas únicamente se conocen por las señales de control transmitidas por el controlador hacia el robot. Por tal motivo la localización del robot dentro del ambiente de trabajo (área de 5 x

10 mts.) conforme éste se desplaza a través del terreno se basa en cálculos analíticos vía software. Dado que para ciertos períodos de tiempo los cuales están determinados por el tiempo entre las señales que envía el controlador, es posible conocer la velocidad y la dirección del robot, es decir su posición dentro del terreno después de dichos tiempos, la localización del robot dentro del terreno puede ser determinada con facilidad, utilizando para ello las ecuaciones 6.1, 6.2 y 6.3 que caracterizan el sistema.

$$p = x + jy \quad (6.1)$$

$$\frac{dp}{dt} = V (\cos \theta_3) e^{j\left(\phi - \frac{\pi}{2}\right)} \quad (6.2)$$

$$\frac{d\phi}{dt} = \frac{V \operatorname{sen} \theta_3}{L} \quad (6.3)$$

donde: "x" y "y" son las coordenadas del centro de gravedad del robot respecto al origen de los ejes coordenados del terreno por donde navega. Con estas coordenadas (x, y) es posible determinar la posición "p" del robot, "V" y "L" son la velocidad y la longitud del robot respectivamente, "φ" es el ángulo del eje longitudinal del robot respecto al eje "y", "t" es el tiempo y "θ₃" es el ángulo de las ruedas del robot con respecto a su eje longitudinal como puede verse en la figura 6.1.

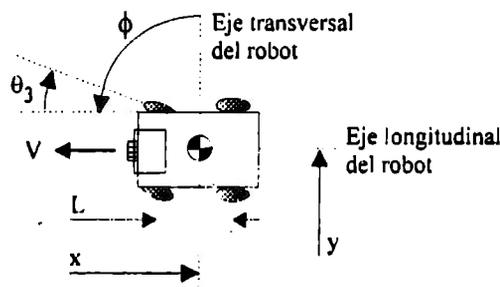


Fig. 6.1 Representación gráfica de los coeficientes de las ecuaciones 6.1, 6.2 y 6.3.

La localización o posición del robot (con respecto al punto meta "B") se realiza por medio de un vector V_m que parte del centro de gravedad del robot y que apunta hacia el punto meta tal como se muestra en la figura 6.2. Mediante la actualización contante de éste vector el robot mantiene su localización con respecto a su objetivo para así poderlo alcanzar. La magnitud

del vector V_m determina la distancia entre el robot y el punto meta. Cada vez que se envían las señales de control al robot dicho vector (V_m) es actualizado.

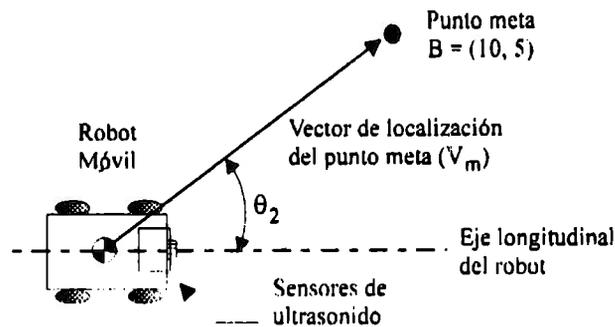


Fig. 6.2 Vector que utiliza el robot como referencia para localizar el punto meta "B".

6.2 GRÁFICAS

Las gráficas que se muestran en las siguientes páginas presentan los resultados que caracterizan de mejor manera (según el autor de esta tesis) el comportamiento general del controlador difuso diseñado para resolver el problema de la navegación de robots móviles autónomos en un ambiente estático (obstáculos sin movimientos) desconocido a priori por el robot. Las gráficas muestran los resultados obtenidos mediante la simulación (líneas continuas) conjuntamente con los resultados experimentales (líneas discontinuas). En cada uno de los ambientes presentados, el robot (simulado o experimental) llega al punto meta (punto "B") a partir del punto inicial "A" a través de la trayectoria generada representada por medio de líneas continuas (simulación) o por líneas discontinuas (experimentales). En todos los casos, el robot móvil parte del punto "A" con una dirección apuntando hacia arriba (dirección positiva del eje "y") tal como se muestra en la figura 6.3.

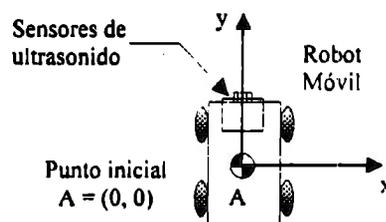
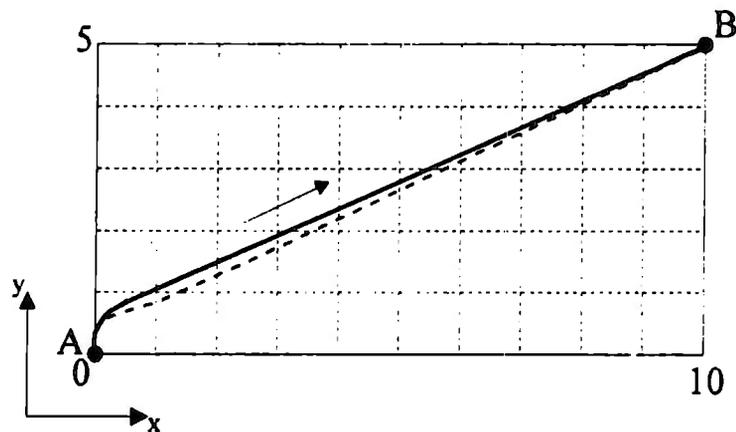
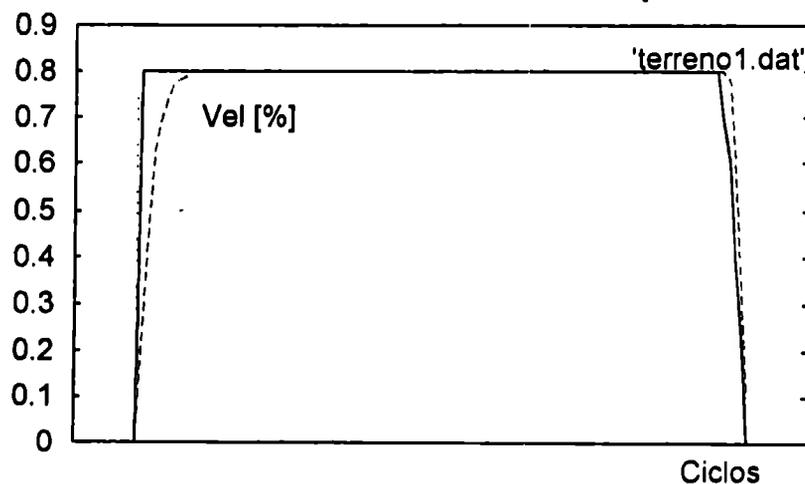


Fig. 6.3 Posición inicial del robot móvil.

Conjuntamente con las gráficas representativas del ambiente (terreno) a través del cual el robot navegará, se presentan las gráficas (Vel vs Ciclos) correspondientes a cada ambiente en donde se muestra la velocidad a la que viajaba el robot conforme navegaba. Cada ciclo representa una decisión tomada por el controlador diseñado con base en la información suministrada por los sensores de ultrasonido del robot (lectura de información proveniente de los sensores-fuzificación-inferencia-defuzificación-envío de las señales de control al robot). Dichas gráficas muestran como es que el robot disminuye la posibilidad de sufrir daños graves en caso de chocar con algún objeto al disminuir su velocidad cuando esta cerca de los obstáculos. Este comportamiento es precisamente el comportamiento que se deseaba. Nuevamente, líneas continuas representan la velocidad adoptada durante las pruebas de simulación mientras que las líneas quebradas corresponden a los resultados experimentales.

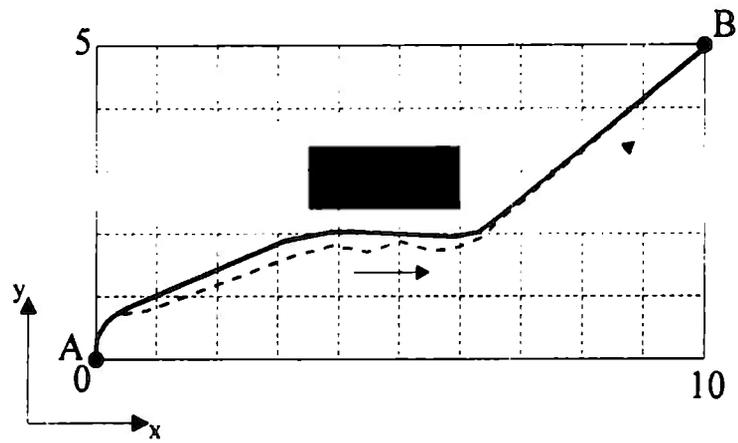


a)

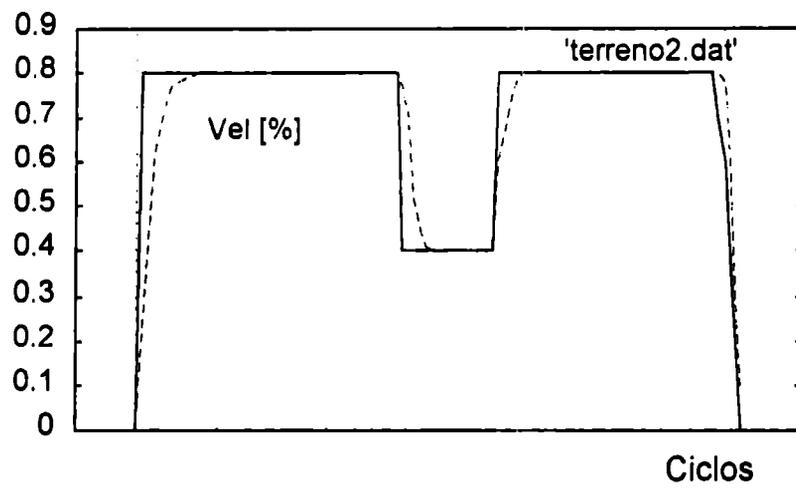


b)

Fig. 6.4 Comportamiento del controlador difuso sobre el terreno No. 1. Simulación (línea continua) y experimentación (línea quebrada). a) Trayectoria y b) Velocidad de navegación.

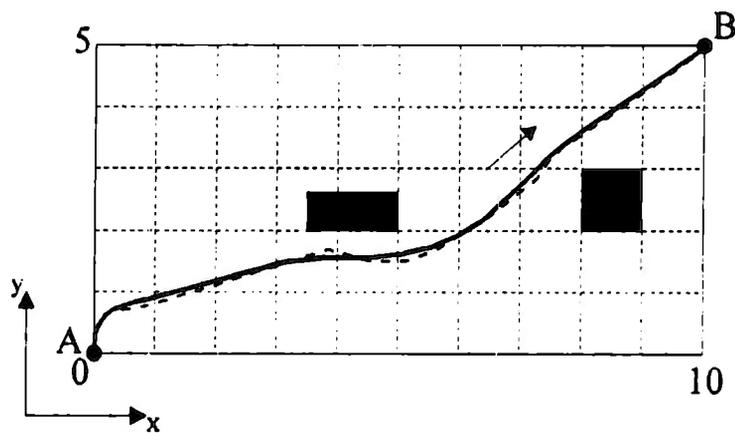


a)



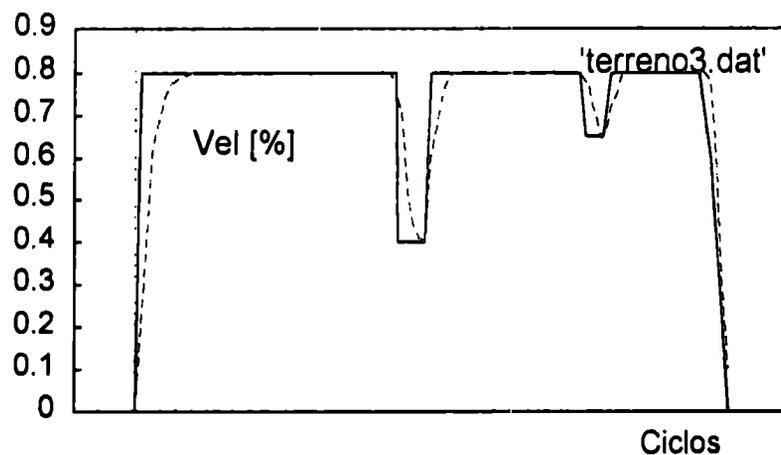
b)

Fig. 6.5 Comportamiento del controlador difuso sobre el terreno No. 2. Simulación (línea continua) y experimentación (línea quebrada). a) Trayectoria y b) Velocidad de navegación.



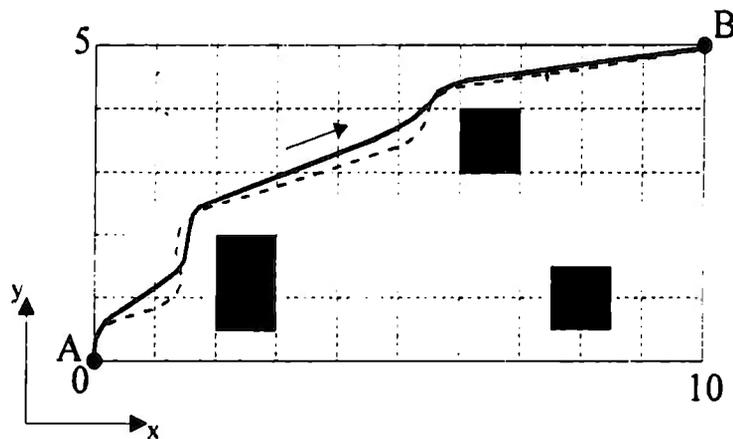
a)

Fig. 6.6 (ver siguiente página)

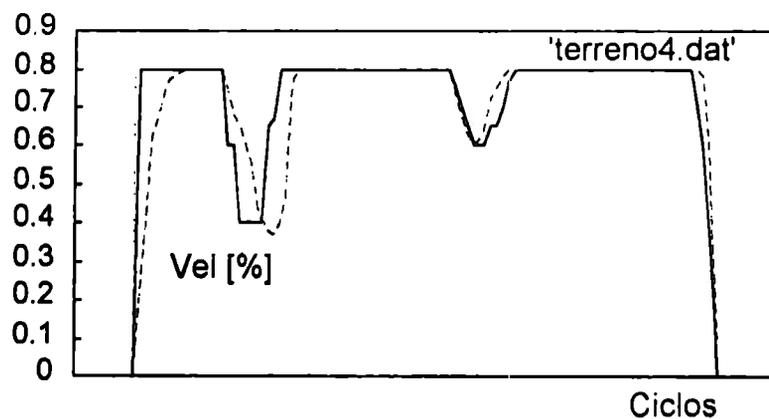


b)

Fig. 6.6 Comportamiento del controlador difuso sobre el terreno No. 3. Simulación (línea continua) y experimentación (línea quebrada). a) Trayectoria y b) Velocidad de navegación.

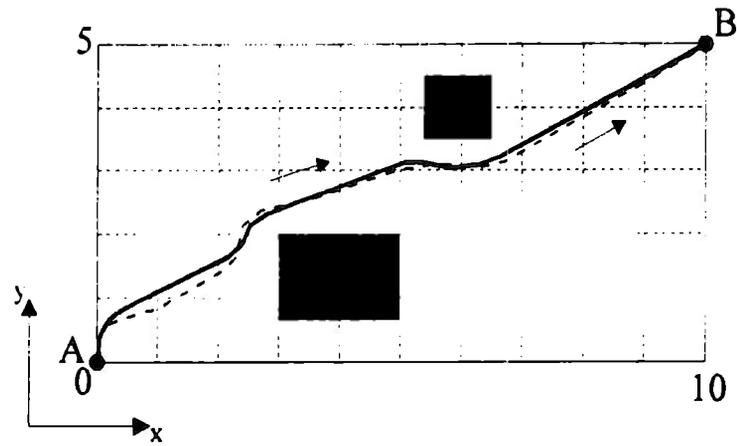


a)

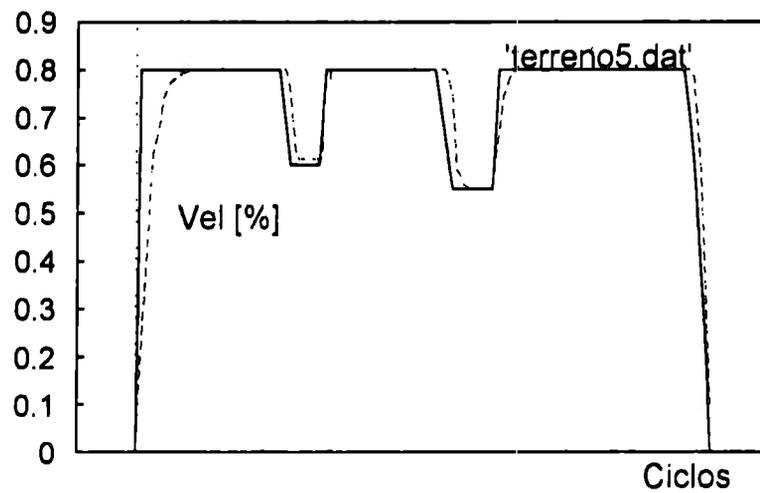


b)

Fig. 6.7 Comportamiento del controlador difuso sobre el terreno No. 4. Simulación (línea continua) y experimentación (línea quebrada). a) Trayectoria y b) Velocidad de navegación.

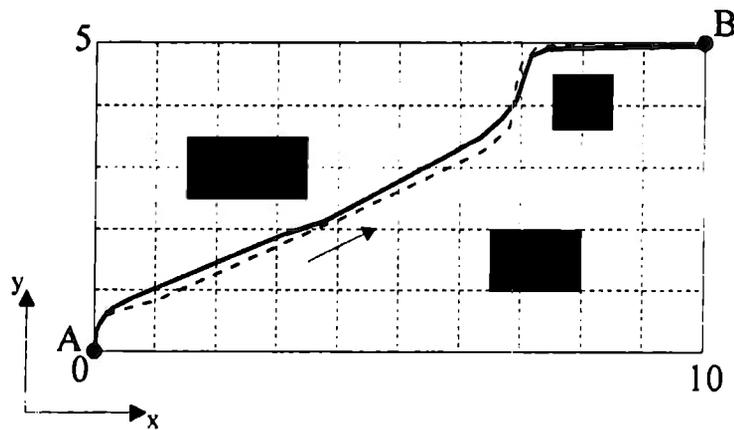


a)



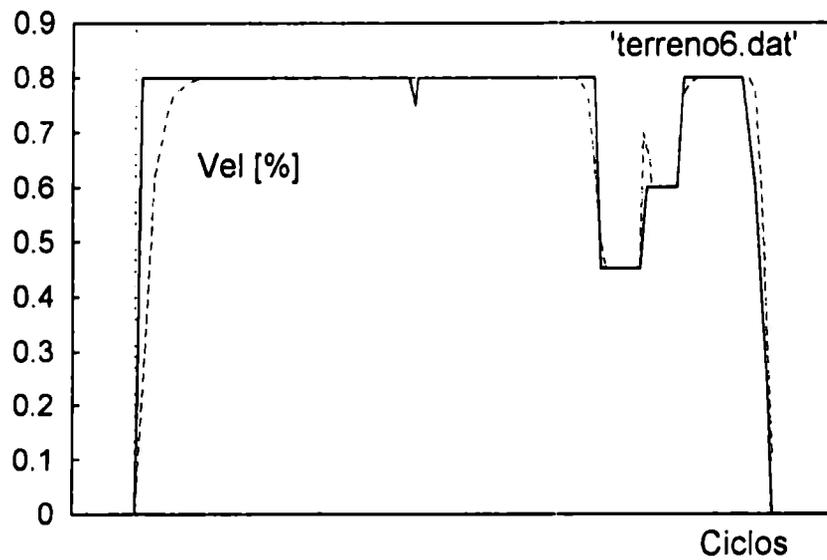
b)

Fig. 6.8 Comportamiento del controlador difuso sobre el terreno No. 5. Simulación (línea continua) y experimentación (línea quebrada). a) Trayectoria y b) Velocidad de navegación.



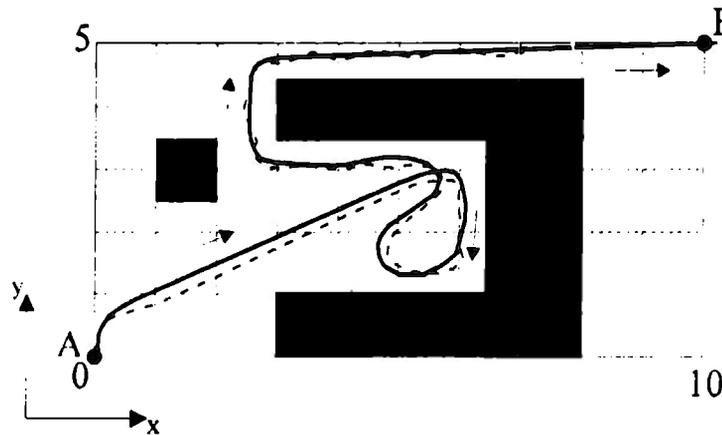
a)

Fig. 6.9 (ver siguiente página)

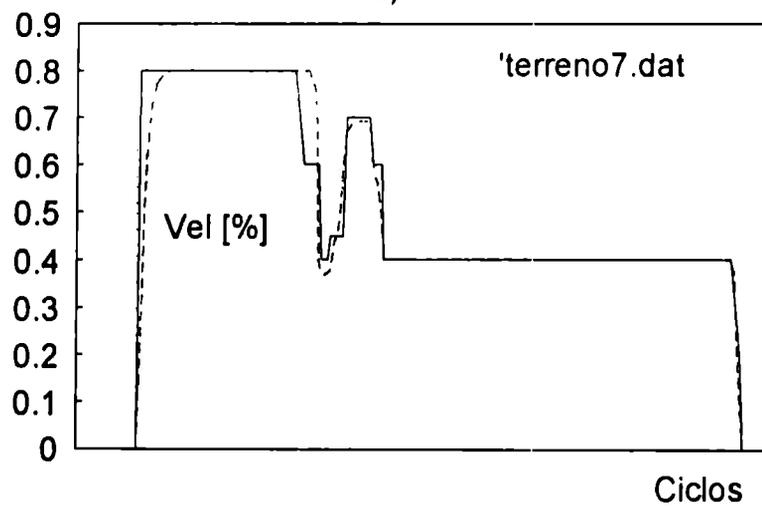


b)

Fig. 6.9 Comportamiento del controlador difuso sobre el terreno No. 6. Simulación (línea continua) y experimentación (línea quebrada). a) Trayectoria y b) Velocidad de navegación.



a)



b)

Fig. 6.10 Comportamiento del controlador difuso sobre el terreno No. 7. Simulación (línea continua) y experimentación (línea quebrada). a) Trayectoria y b) Velocidad de navegación.

Las siguientes 6 figuras (Fig. 6.11 a 6.16) muestran algunos resultados experimentales que se llevaron a cabo sobre diversos terrenos en los cuales se trato de observar el comportamiento del robot para las situaciones típicas descritas en el capítulo número 4 (Fig. 4.11, sección 4.4).

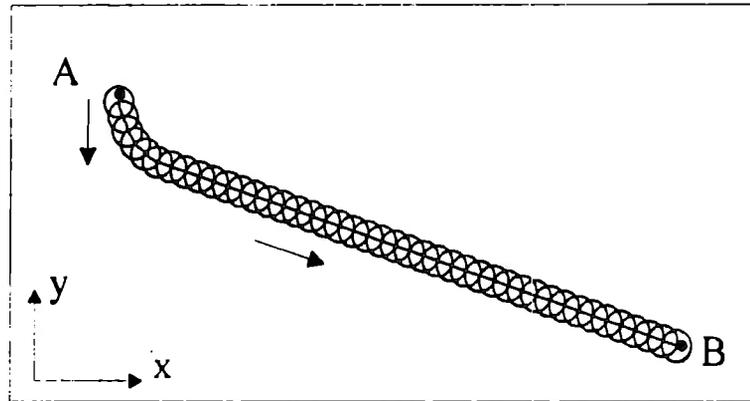


Fig. 6.11 Comportamiento experimental del robot móvil sobre el terreno No. 8.

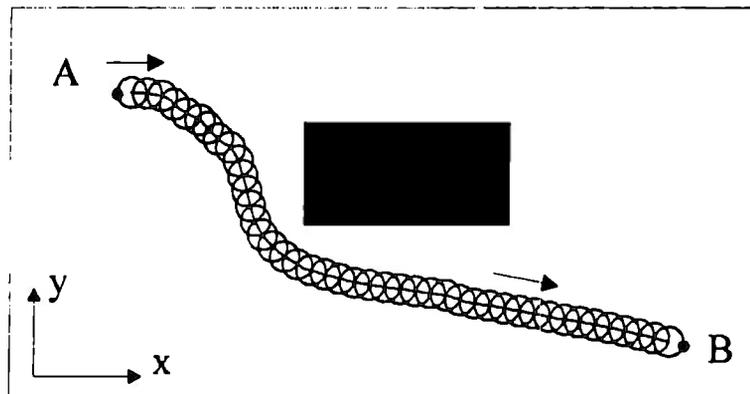


Fig. 6.12 Comportamiento experimental del robot móvil sobre el terreno No. 9.

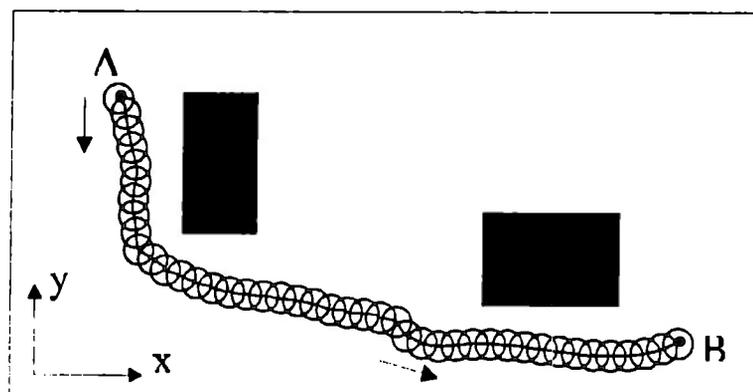


Fig. 6.13 Comportamiento experimental del robot móvil sobre el terreno No. 10.

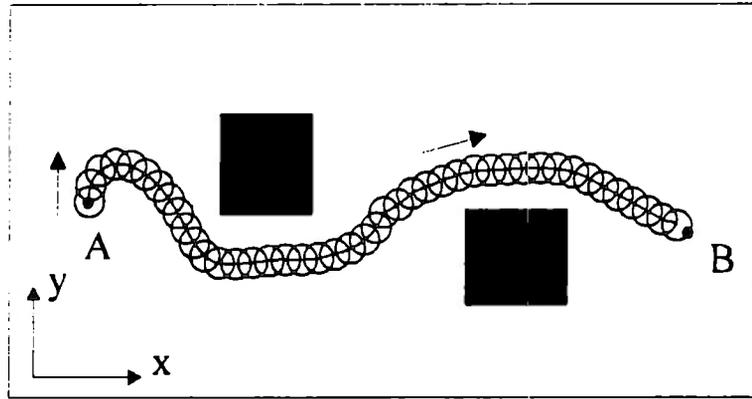


Fig. 6.14 Comportamiento experimental del robot móvil sobre el terreno No. 11.

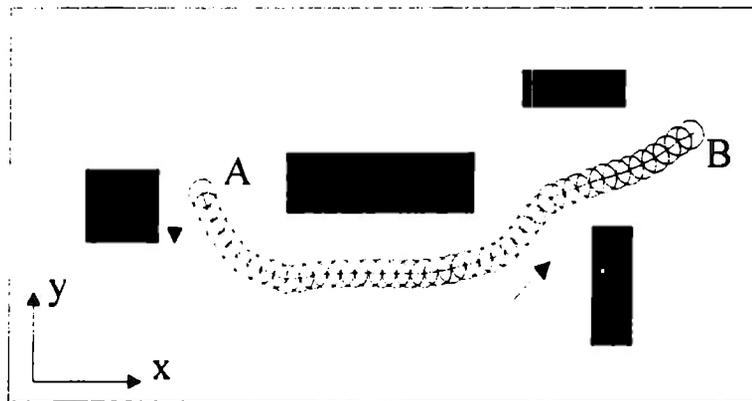


Fig. 6.15 Comportamiento experimental del robot móvil sobre el terreno No. 12.

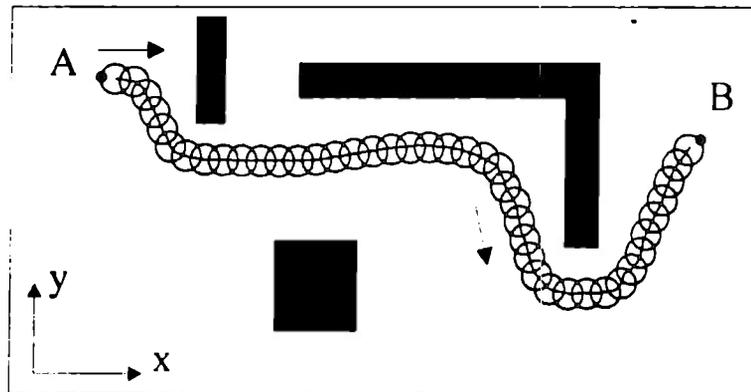


Fig. 6.16 Comportamiento experimental del robot móvil sobre el terreno No. 13.

7 CONCLUSIONES

En éste, el último capítulo de que consta el presente trabajo, se presentan los resultados y conclusiones a que se llegó las cuales están basadas en absolutamente todas las observaciones, comentarios y resultados de los capítulos anteriores. También se proporciona una guía para posibles trabajos a realizar en fechas futuras como una continuación de lo aquí elaborado.

7.1 RESULTADOS

La navegación de robots móviles en ambientes desconocidos mediante lógica difusa fue el principal tema del presente trabajo. Se estudiaron varias otras técnicas de navegación descritas en el capítulo 2, pero se decidió emplear lógica difusa para solucionar el difícil problema de la navegación de vehículos autónomos debido a que se requería manipular información vaga e imprecisa. Este tipo de información es inherente a todo tipo de sensores, dispositivos requeridos por el robot para percibir su entorno, los cuales sufren constantes desajustes y proporcionan frecuentemente datos erróneos e imprecisos al sistema. El empleo de lógica difusa facilita y reduce al mismo tiempo el proceso de calibración del sistema, aminorando así el tiempo y gastos de mantenimiento.

La lógica difusa permitió obtener resultados que van de acuerdo al conocimiento adquirido previamente por un experto el cual fue almacenado en forma de reglas dentro de una base de conocimiento mediante sentencias IF-THEN. Además permitió el manejo

adecuado de variables lingüísticas tales como Cerca, Lejos, Rápido, etc. empleadas con bastante frecuencia durante la navegación y con las cuales fueron elaboradas las reglas.

Pruebas de simulación y experimentación sobre el controlador de navegación difuso (control inteligente) fueron necesarias para validar su nivel de desempeño. En el capítulo seis se muestra como es que tanto los resultados simulados como los experimentales se comportan de la manera deseada. Las pequeñas diferencias entre ambos resultados (simulación y experimentación) se debe al hecho de que durante simulación el robot móvil automáticamente y de manera inmediata adquiere la nueva configuración proporcionada por el controlador, lo cual no sucede así en el sistema real. Es decir, el sistema simulado es discreto (su velocidad y orientación cambian de forma discreta) mientras que el sistema experimental sus cambios son continuos (su velocidad y orientación cambian de forma continua). Por ejemplo, los motores utilizados en el robot experimental no cambian discretamente de una posición a otra conforme a un determinado cambio de señal, sino que el cambio se da de manera gradual y continua. Por otra parte se logró obtener un conjunto de reglas con las cuales el controlador diseñado es capaz de controlar de manera efectiva la navegación de un robot móvil en ambientes desconocidos.

Para validar de manera experimental los resultados obtenidos mediante la simulación, se construyó un robot móvil de cuatro ruedas con un sistema de sensores de ultrasonido, con el cual se realizaron pruebas experimentales. El robot experimental tuvo un bajo costo como puede apreciarse en el desglose proporcionado al final del capítulo número 5 y con lo cual se muestra que es posible construir robots con relativamente poco dinero.

Finalmente como resultado del presente trabajo se escribieron 5 artículos [AR-59], [AR-82], [AR-83], [AR-84], [AR-85] para ser presentados en congresos nacionales e internacionales (algunos de estos artículos aún se encuentran en revisión).

7.2 CONCLUSIONES

A diferencia de las metodologías tradicionales de control, el diseño de controladores difusos no requiere de modelos (analíticos o numéricos) exactos acerca del sistema a controlar. Sin embargo, si se requiere poseer alguna intuición de como controlar el sistema. Dicha intuición

(sentido común) puede haberse obtenido a través del tiempo o a través del aprendizaje al manipular manualmente sistemas iguales o similares. En pocas palabras, si se tiene suficiente información referente al sistema a controlar obtenida mediante la experiencia o el sentido común, se está en posición de poder diseñar un controlador difuso (i.e. para controlar o manejar un automóvil no se requiere conocer el modelo matemático que lo representa).

El diseño de un controlador difuso consiste esencialmente en convertir información (intuición o conocimiento) acerca de cómo controlar un sistema en un conjunto de reglas las cuales son aplicadas al sistema o al modelo que lo representa (según se desee realizar un experimento o una simulación). Si las reglas controlan adecuadamente el sistema, el diseño se ha concluido, de otra manera la forma en que el sistema falla provee información que se debe tomar en cuenta en la realización de cambios al controlador (i.e. modificar las reglas, cambiar la forma de los conjuntos difusos, etc.) para que éste actúe de la manera en que se espera.

Finalmente, aun cuando el proceso de diseño de un controlador difuso es un proceso bastante conocido, todavía se tiene que emplear la experimentación junto con una intensa labor de prueba y error para lograr que el controlador funcione de la manera en que se desea. Esta actividad puede a su vez reducirse usando plataformas de diseño (software) como MATLAB © y FuzzyCalc © que pueden encontrarse fácilmente en el mercado. En cuanto a la parte experimental se tiene que recurrir a la compra o al diseño y construcción para llevar a cabo tal propósito.

7.3 TRABAJOS FUTUROS

La navegación de robots móviles es un problema difícil de resolver. En la actualidad se siguen realizando innumerables esfuerzos para lograr que sistemas artificiales (i.e. robots, vehículos, etc.) logren los niveles de navegación desarrollados por los sistemas naturales (i.e. aves, peces, insectos, etc.). Basta observar los más recientes artículos a ser publicados en las memorias de la conferencia internacional en robótica y automatización organizada por el Institute of Electrical and Electronics Engineers (IEEE) que se llevo a cabo los días 22 al 28 de Abril de 1996 en la ciudad de Minneapolis Minnesota, U.S.A. para darse cuenta de lo difícil

y complicado del problema y de los esfuerzos realizados por diversos investigadores alrededor de mundo para resolver este problema.

Es necesario y se debe seguir trabajando en controladores inteligentes para lograr los niveles de sofisticación presentes en la naturaleza. El presente trabajo deja abierto un panorama para investigar a futuro la navegación de robots móviles en ambientes dinámicos (obstáculos en movimiento) utilizando lógica difusa. Otro interesante problema a ser estudiado en un futuro, es la elaboración de mapas por parte del sistema conforme navega por el terreno, aspectos que no fueron cubiertos. También sería interesante estudiar la posibilidad de reducir el número de reglas necesarias para que el sistema diseñado navegue eficientemente, cosa que es muy probable que así sea puesto que aquí no se estudio dicha posibilidad por temor a modificar el conocimiento extraído del experto. Esto puede realizarse de varias maneras, incluso empleando otras metodologías como son por ejemplo, el uso de algoritmos genéticos para optimizar ya sea las reglas, los conjuntos difusos o ambos [TX-23]. También pudieran emplearse redes neuronales difusas [TX-21] con las cuales sería posible hacer que el sistema aprenda (en tiempo real) a enfrentar y resolver una gran gama de situaciones posiblemente no consideradas en el presente trabajo al formar el conjunto de reglas necesarias para la navegación de sistemas autónomos con base en la información extraída del experto.

Por último es conveniente realizar una interfaz gráfica con la cual sea posible estudiar con mayor facilidad y detalle el comportamiento del controlador de navegación diseñado y/o utilizar otro tipo de sensores de proximidad de los aquí empleados (i.e. cámaras de vídeo). De esta manera se podrá mejorar dicho controlador considerablemente.

ANEXO A REGLAS NECESARIAS PARA EL CONTROL DE LA NAVEGACION DE UN ROBOT MOVIL MEDIANTE LOGICA DIFUSA

A continuación se presentan las 47 reglas en su formato de sentencias IF-THEN generadas a partir de las tablas 4.3, 4.3, 4.5 y 4.6 necesarias para controlar de manera adecuada la navegación de un robot móvil. Cada regla esta identificada por un número precedido de una serie de letras que hacen referencia a si la regla sirve para controlar la dirección y evadir obstáculos (DyE) o es una reglas que sirve para controlar le velocidad (V) del robot.

Las reglas que se obtienen a partir de la tabla 4.3 son las siguientes:

- Regla DyE1 : IF Distancia del Obstáculo es CERCA
 AND Ángulo del Obstáculo es IZQUIERDA GRANDE
 THEN Ángulo de Dirección es DERECHA GRANDE
- Regla DyE2 : IF Distancia del Obstáculo es CERCA
 AND Ángulo del Obstáculo es IZQUIERDA PEQUEÑA
 THEN Ángulo de Dirección es DERECHA GRANDE
- Regla DyE3 : IF Distancia del Obstáculo es CERCA
 AND Ángulo del Obstáculo es CERO
 AND Ángulo error es IZQUIERDA GRANDE
 THEN Ángulo de Dirección es IZQUIERDA GRANDE
- Regla DyE4 : IF Distancia del Obstáculo es CERCA
 AND Ángulo del Obstáculo es CERO
 AND Ángulo error es IZQUIERDA PEQUEÑA
 THEN Ángulo de Dirección es IZQUIERDA GRANDE

Regla DyE5 : IF Distancia del Obstáculo es CERCA
AND Ángulo del Obstáculo es CERO
AND Ángulo error es CERO
THEN Ángulo de Dirección es DERECHA GRANDE

Regla DyE6 : IF Distancia del Obstáculo es CERCA
AND Ángulo del Obstáculo es CERO
AND Ángulo error es DERECHA PEQUEÑA
THEN Ángulo de Dirección es DERECHA GRANDE

Regla DyE7 : IF Distancia del Obstáculo es CERCA
AND Ángulo del Obstáculo es CERO
AND Ángulo error es DERECHA GRANDE
THEN Ángulo de Dirección es DERECHA GRANDE

Regla DyE8 : IF Distancia del Obstáculo es CERCA
AND Ángulo del Obstáculo es DERECHA PEQUEÑA
THEN Ángulo de Dirección es IZQUIERDA GRANDE

Regla DyE9 : IF Distancia del Obstáculo es CERCA
AND Ángulo del Obstáculo es DERECHA GRANDE
THEN Ángulo de Dirección es IZQUIERDA GRANDE

Las reglas que se obtienen a partir de las tablas 4.4 y 4.5 son las siguientes:

Regla DyE10 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA GRANDE
AND Ángulo error es IZQUIERDA GRANDE
THEN Ángulo de Dirección es IZQUIERDA PEQUEÑA

Regla DyE11 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA GRANDE
AND Ángulo error es IZQUIERDA PEQUEÑA
THEN Ángulo de Dirección es IZQUIERDA PEQUEÑA

Regla DyE12 : IF Distancia del Obstáculo es MEDIANA
AND Ángulo del Obstáculo es IZQUIERDA GRANDE
AND Ángulo error es CERO
THEN Ángulo de Dirección es DERECHA PEQUEÑA

Regla DyE13 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA GRANDE
AND Ángulo error es DERECHA PEQUEÑA
THEN Ángulo de Dirección es DERECHA PEQUEÑA

- Regla DyE14 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA GRANDE
AND Ángulo error es DERECHA GRANDE
THEN Ángulo de Dirección es DERECHA GRANDE
- Regla DyE15 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA PEQUEÑA
AND Ángulo error es IZQUIERDA GRANDE
THEN Ángulo de Dirección es IZQUIERDA GRANDE
- Regla DyE16 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA PEQUEÑA
AND Ángulo error es IZQUIERDA PEQUEÑA
THEN Ángulo de Dirección es CERO
- Regla DyE17 : IF Distancia del Obstáculo es MEDIANA
AND Ángulo del Obstáculo es IZQUIERDA PEQUEÑA
AND Ángulo error es CERO
THEN Ángulo de Dirección es DERECHA PEQUEÑA
- Regla DyE18 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA PEQUEÑA
AND Ángulo error es DERECHA PEQUEÑA
THEN Ángulo de Dirección es DERECHA PEQUEÑA
- Regla DyE19 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es IZQUIERDA PEQUEÑA
AND Ángulo error es DERECHA GRANDE
THEN Ángulo de Dirección es DERECHA GRANDE
- Regla DyE20 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es CERO
AND Ángulo error es IZQUIERDA GRANDE
THEN Ángulo de Dirección es IZQUIERDA GRANDE
- Regla DyE21 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es CERO
AND Ángulo error es IZQUIERDA PEQUEÑA
THEN Ángulo de Dirección es IZQUIERDA PEQUEÑA
- Regla DyE22 : IF Distancia del Obstáculo es MEDIANA
AND Ángulo del Obstáculo es CERO
AND Ángulo error es CERO
THEN Ángulo de Dirección es DERECHA PEQUEÑA
- Regla DyE23 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es CERO
AND Ángulo error es DERECHA PEQUEÑA
THEN Ángulo de Dirección es DERECHA PEQUEÑA

- Regla DyE24 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es CERO
AND Ángulo error es DERECHA GRANDE
THEN Ángulo de Dirección es DERECHA GRANDE
- Regla DyE25 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es DERECHA PEQUEÑA
AND Ángulo error es IZQUIERDA GRANDE
THEN Ángulo de Dirección es IZQUIERDA GRANDE
- Regla DyE26 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es DERECHA PEQUEÑA
AND Ángulo error es IZQUIERDA PEQUEÑA
THEN Ángulo de Dirección es IZQUIERDA PEQUEÑA
- Regla DyE27 : IF Distancia del Obstáculo es MEDIANA
AND Ángulo del Obstáculo es DERECHA PEQUEÑA
AND Ángulo error es CERO
THEN Ángulo de Dirección es IZQUIERDA PEQUEÑA
- Regla DyE28 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es DERECHA PEQUEÑA
AND Ángulo error es DERECHA PEQUEÑA
THEN Ángulo de Dirección es CERO
- Regla DyE29 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es DERECHA PEQUEÑA
AND Ángulo error es DERECHA GRANDE
THEN Ángulo de Dirección es DERECHA GRANDE
- Regla DyE30 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es DERECHA GRANDE
AND Ángulo error es IZQUIERDA GRANDE
THEN Ángulo de Dirección es IZQUIERDA GRANDE
- Regla DyE31 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es DERECHA GRANDE
AND Ángulo error es IZQUIERDA PEQUEÑA
THEN Ángulo de Dirección es IZQUIERDA PEQUEÑA
- Regla DyE32 : IF Distancia del Obstáculo es MEDIANA
AND Ángulo del Obstáculo es DERECHA GRANDE
AND Ángulo error es CERO
THEN Ángulo de Dirección es IZQUIERDA PEQUEÑA
- Regla DyE33 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es DERECHA GRANDE
AND Ángulo error es DERECHA PEQUEÑA
THEN Ángulo de Dirección es DERECHA PEQUEÑA

Regla DyE34 : IF Distancia del Obstáculo es MEDIANA OR LEJOS
AND Ángulo del Obstáculo es DERECHA GRANDE
AND Ángulo error es DERECHA GRANDE
THEN Ángulo de Dirección es DERECHA PEQUEÑA

Regla DyE35 : IF Distancia del Obstáculo es LEJOS
AND Ángulo error es CERO
THEN Ángulo de Dirección es CERO

Las reglas que se obtienen a partir de la tabla 4.6 son las siguientes:

Regla V1 : IF Distancia del robot al punto meta es CERO
THEN Velocidad (del robot) es CERO

Regla V2 : IF Distancia del Obstáculo es CERO
THEN Velocidad (del robot) es CERO

Regla V3 : IF Distancia del robot al punto meta es CERCA
AND Distancia del Obstáculo NO es CERO
THEN Velocidad (del robot) es REALMENTE LENTO

Regla V4 : IF Distancia del robot al punto meta es REGULAR
AND Distancia del Obstáculo es CERCA
THEN Velocidad (del robot) es REALMENTE LENTO

Regla V5 : IF Distancia del robot al punto meta es REGULAR
AND Distancia del Obstáculo es MEDIANA
THEN Velocidad (del robot) es LENTO

Regla V6 : IF Distancia del robot al punto meta es REGULAR
AND Distancia del Obstáculo es LEJOS
THEN Velocidad (del robot) es LENTO

Regla V7 : IF Distancia del robot al punto meta es MEDIANA
AND Distancia del Obstáculo es CERCA
THEN Velocidad (del robot) es LENTO

Regla V8 : IF Distancia del robot al punto meta es MEDIANA
AND Distancia del Obstáculo es MEDIANA
THEN Velocidad (del robot) es MEDIANA

Regla V9 : IF Distancia del robot al punto meta es MEDIANA
AND Distancia del Obstáculo es LEJOS
THEN Velocidad (del robot) es MEDIANA

Regla V10 : IF Distancia del robot al punto meta es LEJOS
AND Distancia del Obstáculo es CERCA
THEN Velocidad (del robot) es LENTO

Regla V11 : IF Distancia del robot al punto meta es LEJOS
AND Distancia del Obstáculo es MEDIANA
THEN Velocidad (del robot) es RÁPIDO

Regla V12 : IF Distancia del robot al punto meta es LEJOS
AND Distancia del Obstáculo es LEJOS
THEN Velocidad (del robot) es RÁPIDO

ANEXO B INFERENCIA TIPO MAMDANI

Con base en la teoría de conjuntos difusos la cual es capaz de manejar incertidumbre surgida a partir de conocimiento lingüístico, Mamdani propuso una técnica de inferencia conocida como "inferencia tipo Mamdani" la cual ha sido adoptada por muchos sistemas expertos.

Para ilustrar este tipo de inferencia, considere una expresión lingüística $E(A)$ sobre un espacio X a partir de la cual es posible definir una función de mapeo $\mu(A, x)$ ($x \in X, \mu \in [0, 1]$) llamada "función de pertenencia" con la cual es posible caracterizar al conjunto difuso "A", identificado por $E(A)$. La función de pertenencia $\mu(A, x)$ indica el grado en el que la variable "x" pertenece al conjunto "A" el cual es una extensión de una función de densidad probabilística. Por ejemplo, si $\mu(A, x) = 1$ se implica que "x" satisface completamente a $E(A)$, mientras que $\mu(A, x) = 0$ implica que "x" no satisface definitivamente de ninguna forma a $E(A)$. Dentro de la teoría de la lógica difusa el conjunto difuso "A" caracterizado por $\mu(A, x)$ es usualmente expresado como:

$$A = \int \mu(A, x) / x \quad (B-1)$$

donde \int denota un conjunto definido sobre un espacio continuo "X" (y no implica integración). Si "x" es una variable discreta, en vez de la integración se debe de calcular una sumatoria (Σ). Por ejemplo, sea "X" un intervalo de temperatura definido por $[0, 40]$ (grados), y sea $E(A)$ el conjunto difuso "Tibio". Conforme a la expresión lingüística $E(A)$, es posible definir una función de pertenencia $\mu(A, x)$ tal como la que aparece en la figura B-1, con la cual es posible expresar el grado en el que "x" pertenece a $E(A)$ (en algunas ocasiones expresado como "x IS A").

Tomando como base la definición de un conjunto difuso, la inferencia tipo Mamdani se formula mediante el uso de una relación difusa. La relación antes citada es definida como un conjunto difuso "Q" sobre un conjunto producto $X \times Y$ el cual es caracterizado por una función de pertenencia $\mu(Q, x, y): X \times Y \rightarrow [0, 1]$.

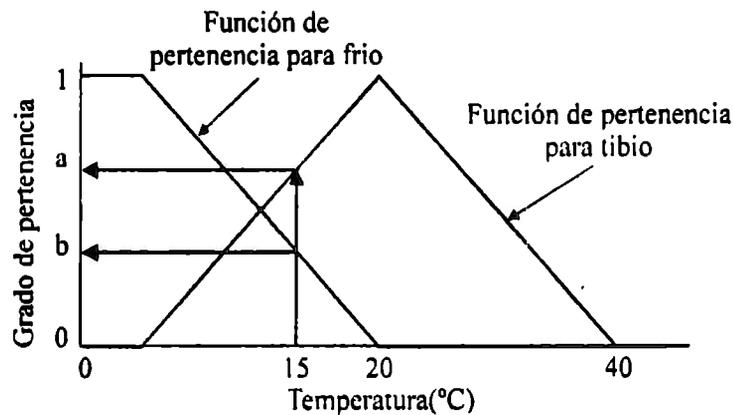


Fig. B-1 Ejemplo de funciones de pertenencia.

Asumiendo que los conjuntos difusos A y B están sobre los conjuntos X y Y respectivamente, y que $A \times B$ expresa el producto de los conjuntos difusos, la función de pertenencia del conjunto producto $A \times B$ es definido por:

$$\mu(AB, x, y) = \min \{ \mu(A, x), \mu(B, y) \mid (x, y) \in XY \} \quad (B-2)$$

donde $\mu(A, x)$ y $\mu(B, y)$ son las funciones de pertenencia para los conjuntos difusos A y B respectivamente.

Si existe una regla tal que "IF x IS A, THEN y IS B" y algún hecho tal que "x IS A'" el procedimiento tipo Mamdani infiere:

$$B' = A' \bullet (A \times B) \quad (B-3)$$

donde \bullet implica la composición de las relaciones difusas A' y $(A \times B)$ la cual es definida por:

$$\mu(B', y) = \max \{ \mu(A', x), \mu(A \times B, x, y) \mid x \in X \} \quad (B-4)$$

El resultante de la ecuación B-4 es generalmente un valor distribuido sobre "Y". Sin embargo, en ciertos casos (como en el del presente trabajo) un solo punto obtenido a partir

de $\mu(B', y)$ es lo que se requiere como salida de control, lo cual se logra mediante el empleo de algún método de defuzificación sobre $\mu(B', y)$, el cual convierte el conjunto de valores distribuidos sobre "Y" ($\mu(B', y)$) en un único valor. De los diferentes métodos de defuzificación, el empleado con mayor frecuencia en sistemas de control es el método "centroidal" también llamado "centro-de-área" el cual está definido de la siguiente forma:

$$\bar{y} = \int \mu(B', y) y dy / \int \mu(B', y) dy \quad (B-5)$$

Dicho método puede ser extendido fácilmente para abarcar casos generales basados en múltiples reglas con múltiples premisas. La figura B-2 ejemplifica el procedimiento de inferencia tipo Mamdani para dos reglas con diversas premisas, donde los operadores OR y AND entre premisas son interpretadas como las funciones "max" y "min" respectivamente entre las funciones de pertenencias sobre las que se actúa (éste tipo de inferencia es también conocido como inferencia "Max-Min").

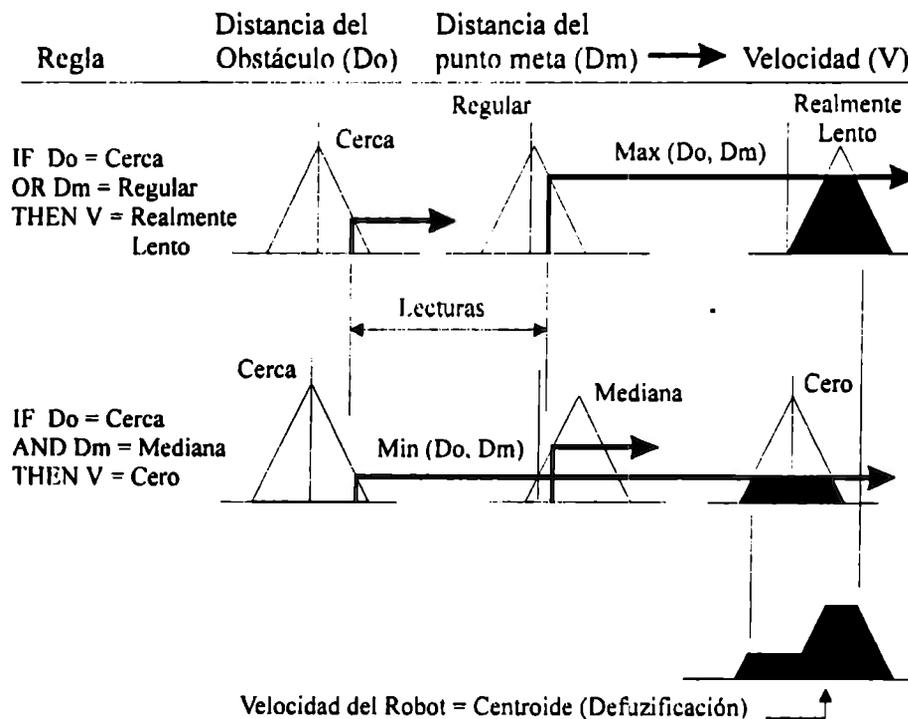


Fig. B-2 Procedimiento de inferencia tipo Mamdani y defuzificación centroidal para algunas reglas.

ANEXO C PRINCIPALES FUNCIONES DEL CONTROLADOR DIFUSO ELABORADO EN C++

En los siguientes párrafos se describen las principales funciones utilizadas en el controlador de navegación difuso escrito en lenguaje C++ con el que se controla el robot experimental.

```
float intrap (float, float, float, float); // Calcula la integral (area bajo la curva) por medio de la
// regla del trapecio de los dos punto (x,y) que se le
// envian 2 coordenadas (x0, y0), (x1, y1) que definen la
// recta bajo la cual el área será calculada.

float centroid (float, float, float, float); // Calcula el centroide de un poliedro dados los 2 puntos
// (x1,y1) (x2,y2) que lo definen. Requiere de la funcion
// "intrap".

float defuzify (float x[8], float y[8], int n); // Realiza el metodo de defuzificacion centroidal.
// Requiere de las funciones "intrap" y "centroid".

float ftriangulo(float val, // *(x2,y2) Funcion que determina el valor
float x1, // ** de pertenencia de un valor en un
float x2, // * * conjunto difuso triangular.
float x3); // (x1,y1) ***** (x3,y3)

float sensores(int); // Funcion que capta la informacion de los sensores de ultrasonido.
// Regresa valores de distancia.

float senal_control(float x[8],float y[8],int n); // Funcion que regresa la senal de control (vel o
// direccion). Requiere de la funcion "defuzify".

void obtain_trapecios(float val_pert, // Funcion para obtener las coordenadas de los
float x1, // trapecios dado un valor de perttencia y un
```

```

float x2, // conjunto difuso triangular.
float x3, int i);

void reglas_DyE(float dist_obst, // Funcion que evalua las reglas según valores crisp
int num_sensor, // (dist_obst y ang_error) y regresa las coordenadas
float ang_error); // de los trapecios obtenidos de la inferencia.
// DyE = Direccion y Evasion. Requiere de las funciones
// "ftriangulo" y "obtain_trapecios".

void reglas_V(float dist_obst, // Funcion que evalua las reglas según valores crisp
float dist_meta); // (dist_obst y dist_meta) y regresa las coordenadas de
// los 2 trapecios obtenidos de la inferencia. V = Vel.
// Requiere de las funciones "ftriangulo" y
// "obtain_trapecios".

//*****
// Funcion que obtiene la senal de control hacia el robot (vel. o direccion)
float senal_control (float x[ ], float y[ ], int n)
{int i; // contador
float x_nueva=0, y_nueva=0; // coordenada de la interseccion de los 2 trapecios.
float senal; // senal de control a MOBIRO (vel. o direccion).
float temporal, // Variables temporales que almacenan las
pend_1, pend_2, // pendientes y las ordenadas al origen de
ord_1, ord_2; // varias rectas.

// A continuacion se determina que rectas son las que se intersectan y se determina los
//nuevos valores de las variables "x" y "y" segun la siguiente figura:
//
// 5 ----- 6
// / \
// 1 ----- 2 \
// / / \ \
// _0 / _____ 4/ _____ \3 _____ \7 _____
//
if (y[5]>y[2] && x[2]>x[4]) // Intersección de las rectas 1-2 y 4-5
{temporal=(y[5]-y[4])/(x[5]-x[4]); // Pendiente de la recta 4-5
x_nueva=(y[2]-y[4]+temporal*x[4])/temporal; // Coordenada x del punto de intersección.
if (x_nueva<x[2])
{

```

```

        y_nueva=y[2]; // Coordenada y del punto de intersección.
        x[2]=x_nueva; y[2]=y_nueva;
        x[3]=x[5]; y[3]=y[5];
        x[4]=x[6]; y[4]=y[6];
        x[5]=x[7]; y[5]=y[7];
        n=6;
    }
}
if (y[2]>y[5] && x[5]<x[3]) // Intersección de las rectas 2-3 y 5-6
{
    temporal=(y[3]-y[2])/(x[3]-x[2]); // Pendiente de la recta 2-3
    x_nueva=(y[5]-y[2]+temporal*x[2])/temporal; // Coordenada x del punto de intersección.
    if (x_nueva>x[5])
    {
        y_nueva=y[5]; // Coordenada y del punto de intersección.
        x[3]=x_nueva; y[3]=y_nueva;
        x[4]=x[6]; y[4]=y[6];
        x[5]=x[7]; y[5]=y[7];
        n=6;
    }
}
if (n==8) // Intersección de las rectas 2-3 y 4-5
{
    pend_1=(y[3]-y[2])/(x[3]-x[2]); // pendiente de la recta 2-3
    pend_2=(y[5]-y[4])/(x[5]-x[4]); // pendiente de la recta 5-4
    ord_1 =y[2]-pend_1*x[2]; // ordenada al origen de la recta 2-3
    ord_2 =y[4]-pend_2*x[4]; // ordenada al origen de la recta 5-4
    x_nueva=(ord_2-ord_1)/(pend_1-pend_2); // Coord. "x" de intersección.
    y_nueva=ord_1+pend_1*x_nueva; // Coord. "y" de intersección.
    x[3]=x_nueva; y[3]=y_nueva;
    x[4]=x[5]; y[4]=y[5];
    x[5]=x[6]; y[5]=y[6];
    x[6]=x[7]; y[6]=y[7];
    n=7;
}

```

```

    }
    senal = defuzify (x, y, n);
return senal;
} // Fin de la función senal_control
//*****
// Función que realiza una defuzificación tipo centroidal.
float defuzify (float x[], float y[], int n)
{int i; // contador
float area[15]; // Areas parciales de que esta compuesta la fig.
float Qx=0; // Qx = Sumatoria ( cen_x[i]*area[i] )
float area_t=0; // Area total resultado de la integración
float CENTROIDE=0; // Valor del centroide sobre el eje x
// A continuación se determinan las áreas parciales así como también el área total bajo los
//puntos introducidos.
for (i=0;i<(n-1);i++)
{
area[i] = inttrap (x[i], y[i], x[i+1], y[i+1]);
area_t += area[i];
}
// A continuación se determina el valor de Qx para después determinar el centroide de la
// figura dada por los puntos.
Qx = 0;
for (i=0;i<(n-1);i++)
Qx += area[i] * centroid (x[i], y[i], x[i+1], y[i+1]);
if (area_t!=0)
CENTROIDE = Qx/area_t;
return CENTROIDE;
} // Fin de la función defuzify
//*****
// función inttrap. Realiza la integración dados dos puntos utilizando la regla del trapecio.
float inttrap (float x1, float y1, float x2, float y2)
{int i; // contador
int N_int=1; // Número de intervalos
float t=0.0; // valor de la integral

```

```

for (i=1; i<=N_int;i++)
{
t = (x2-x1)*(y1+y2)/2.0;          // fórmula de integración del trapecio.
}
return t;                          // Regresa el valor de la integral (área bajo la curva).
}                                    // Fin de la función inttrap
//*****

// La función centroid determina el centroide dados dos puntos (x1, y1), (x2, y2) los cuales
// determinan la figura:
float centroid (float x1, float y1, float x2, float y2)
{float area;                          // área bajo la recta que une los dos puntos.
int i;                                // contador
int N_int=1;                          // Número de intervalos.
int d;                                // valor característico de la pendiente. si d=2 implica pendiente
// es + si d=1 implica pendiente es -
float X=0.0;                          // valor del centroide sobre el eje x
if (y2>=y1 || y2< y1) d=2;
for (i=1;i<=N_int;i++)
{area = inttrap(x1,y1,x2,y2);
if (area!=0)
X=(((x2-x1)*(x2-x1)/2)*(y1+((y2-y1)/3)*d))/area+x1; }
return X;                              // Regresa el valor del.centroide.
}                                        // Fin de la función centroid.
//*****

// Función que regresa el valor de pertenencia de una función dado un crisp value y un
// conjunto difuso triangular.
float ftriangulo(float val,           //          * (x2,y2)
float x1,           //          * *
float x2,           //          * *
float x3)           // (x1,y1) ***** (x3,y3)
{float V_de_Pert;           // valor de pertenencia.
if (x1<val && val<x2)
V_de_Pert = (val-x1)/(x2-x1);
if (x2<val && val<x3)

```

```

    V_de_Pert = (x3-val)/(x3-x2);
if (val<=x1 || val>=x3)
    V_de_Pert = 0.0;
if (val==x2)
    V_de_Pert = 1.0;
return V_de_Pert;                // Regresa el valor de pertenencia.
}                                // Fin de la función ftriangulo.
//*****
// Función que regresa las cordenadas del trapecio que resulta dado un crisp value y un
// conjunto difuso triangular.
float ftrapecio(float val,        //      * (x2,y2)      2  3
               float x1,        //      **      -->      *****
               float x2,        //      *  *          *  *
               float x3,        // (x1,y1) ***** (x3,y3) 1 ***** 4
               int i)           // si i=0 1er trapecio, si i=4 2do trapecio
{
if (x1<val && val<x2)
{
    varios_y[i] = 0;
    varios_y[i+1] = (val-x1)/(x2-x1);           // Se determina el valor
    varios_y[i+2] = varios_y[i+1];             // de pertenencia
    varios_y[i+3] = 0;
    varios_x[i] = x1;                          // Se determinan los
    varios_x[i+1] = val;                       // nuevos valores de
    varios_x[i+2] = x3 - (varios_y[i+1]*(x3-x2)); // las coordenadas "x"
    varios_x[i+3] = x3;
}
if (x2<val && val<x3)
{
    varios_y[i] = 0;
    varios_y[i+2] = (x3-val)/(x3-x2);           // Se detremina el valor
    varios_y[i+1] = varios_y[i+2];             // de pertenencia
    varios_y[i+3] = 0;
    varios_x[i] = x1;                          // Se determinan los
}
}

```

```

    varios_x[i+1] = (varios_y[i+1]*(x2-x1)) + x1;    // nuevos valores de
    varios_x[i+2] = val;                            // las coordenadas "x"
    varios_x[i+3] = x3;
}
if (val<=x1 || val>=x3)                            // todos los valores de "y" son cero
{
    varios_y[i] = 0;
    varios_y[i+1] = 0;
    varios_y[i+2] = 0;
    varios_y[i+3] = 0;
    varios_x[i] = x1;                               // Se determinan los
    varios_x[i+1] = x2;                             // nuevos valores de
    varios_x[i+2] = x2;                             // las coordenadas "x"
    varios_x[i+3] = x3;
}
if (val==x2)
{
    varios_y[i]=0;
    varios_y[i+1] = 1.0;                            // Se determina el valor
    varios_y[i+2] = 1.0;                            // de pertenencia
    varios_y[i+3] = 0;
    varios_x[i] = x1;                               // Se determinan los
    varios_x[i+1] = val;                             // nuevos valores de
    varios_x[i+2] = val;                             // las coordenadas "x"
    varios_x[i+3] = x3;
}
return 0;
}                                                    // Fin de la función ftrapecio
//*****
// Función para obtener las coordenadas de los trapecios dado un valor de pertenencia y un
// conjunto difuso triangular
void obtain_trapecios(float val_pert,              // Función para obtener las coordenadas
                    float x1,                     // de los trapecios dado un valor de
                    float x2,                     // pertenencia y un conjunto difuso

```

```

        float x3,                // triangular.
        int i)                // si i=0 1er trapecio si i=4 2do trapecio
{
    if (val_pert == 0)
    {
        varios_y[i] = 0;        // No existe trapecio. Todas las
        varios_y[i+1] = 0;      // coordenadas y's tienen valor de
        varios_y[i+2] = 0;      // 0 (cero) y las coordenadas x's
        varios_y[i+3] = 0;      // toman valores cualquiera
        varios_x[i] = x1;
        varios_x[i+1] = x2;
        varios_x[i+2] = x2;
        varios_x[i+3] = x3;
    }
    else
    {
        varios_y[i] = 0;        // Si el valor de pertenencia es
        varios_y[i+1] = val_pert; // distinto de cero (val_pert>0)
        varios_y[i+2] = val_pert; // entonces existe el trapecio cuyas
        varios_y[i+3] = 0;      // 4 coordenadas aquí se obtienen.
        varios_x[i] = x1;
        varios_x[i+1] = (val_pert*(x2-x1)) + x1;
        varios_x[i+2] = x3 - (val_pert*(x3-x2));
        varios_x[i+3] = x3;
    }
}
// Termina Función obtain_trapecios

```

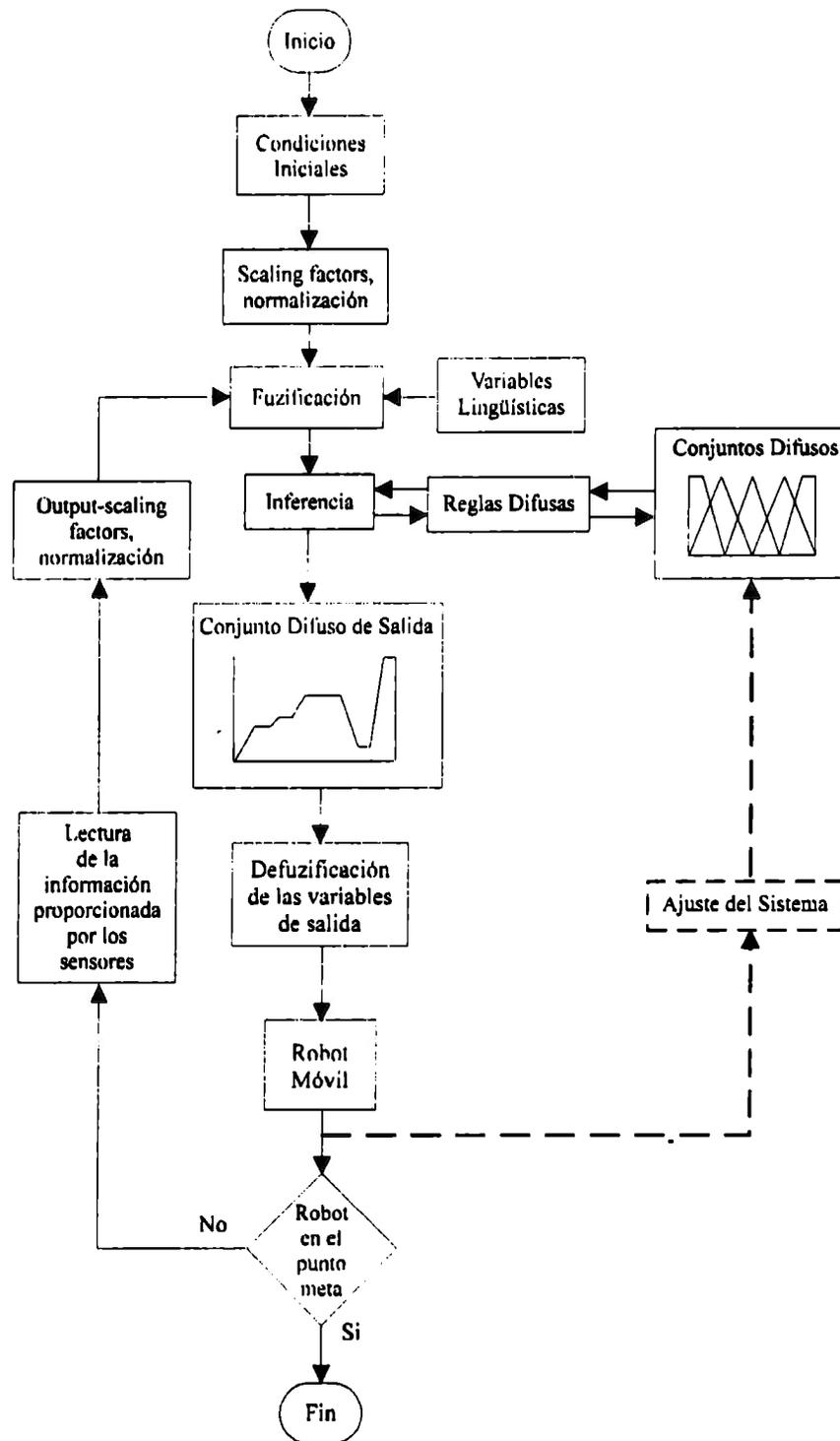


Fig. C.1 Diagrama de flujo que muestra el proceso que lleva a cabo el controlador difuso diseñado. El mismo esquema se utilizó para la simulación (MATLAB ©) y para la experimentación (programa desarrollado en C++).

ANEXO D ESPECIFICACIONES DE ALGUNOS DISPOSITIVOS UTILIZADOS PARA LA CONSTRUCCIÓN DEL SISTEMA EXPERIMENTAL

En las siguientes páginas se presentan las especificaciones de algunos de los componentes con los que se construyo el robot experimental.

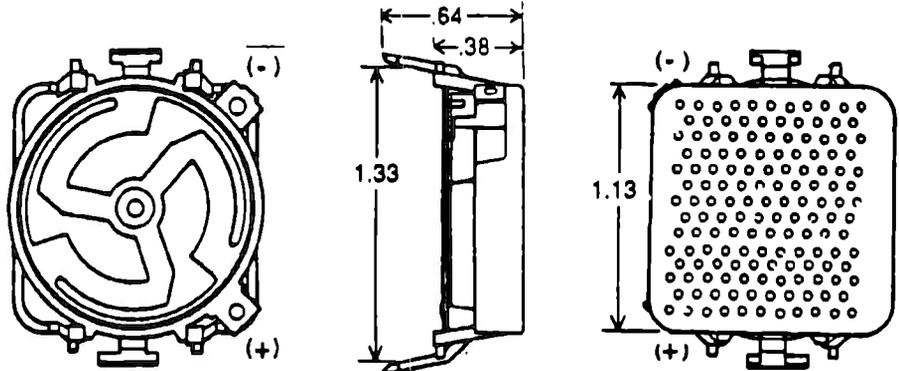
Technical Specifications for

7000 Series Electrostatic Transducer

Part# 616736-Minimum order 10, increments of 10

Part# 612366-Minimum order 100, increments of 100

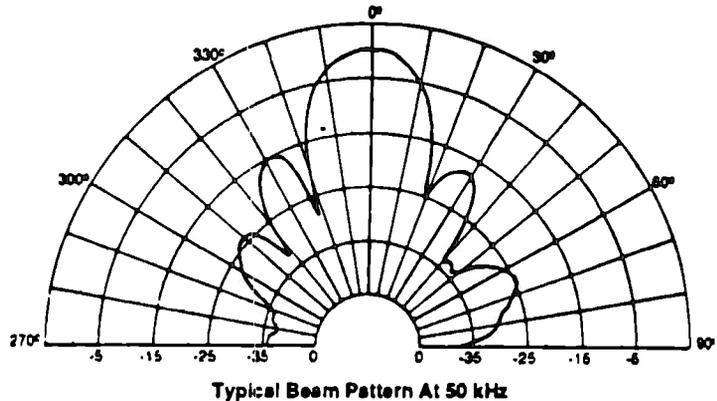
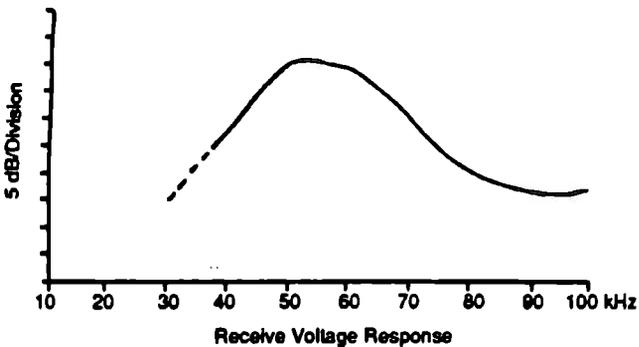
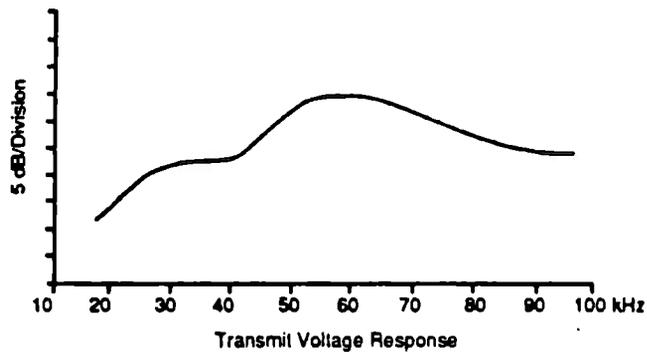
The 7000 Series electrostatic transducer is specifically intended for operation in air at ultrasonic frequencies. The assembly comes complete with a perforated protective cover.



Specifications:

Usable Transmitting Frequency Range	See Graph	Maximum Combined Voltage	400V
Usable Receiving Frequency Range	See Graph	Capacitance at 1 kHz (Typical)	600-700 pf
Beam Pattern	See Graph	150 vdc bias	
Minimum Transmitting Sensitivity at 50 kHz 300 vac pk-pk, 150 vdc bias (dB re 20 μPa at 1 meter)	106.9 dB	Operating Conditions	
Minimum Receiving Sensitivity at 50 kHz 150 vdc bias (dB re 1v/Pa)	-43.4dB	Temperature	32°-140°F
		Relative Humidity	5%-95%
		Standard Finish	
Suggested DC Bias Voltage	200V	Foil	Gold
Suggested AC Driving Voltage (peak)	200V	Housing	Black

Specifications subject to change without notice.



Note: dB normalized to on-axis response.
Note: Curves are representative only. Individual responses may differ.

For additional information, technical assistance or prices and delivery:
Polaroid OEM Components Group
Tel: (617) 386-3965 Fax: (617) 386-3966

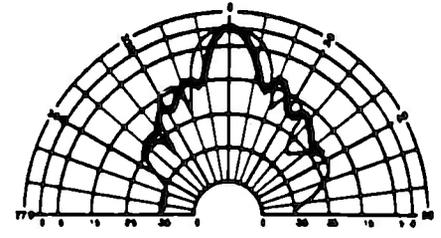
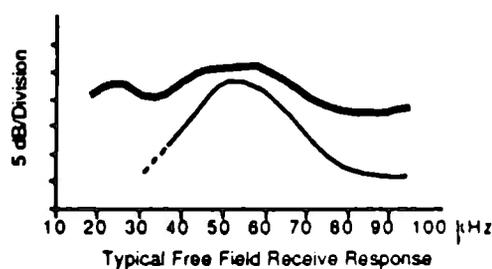
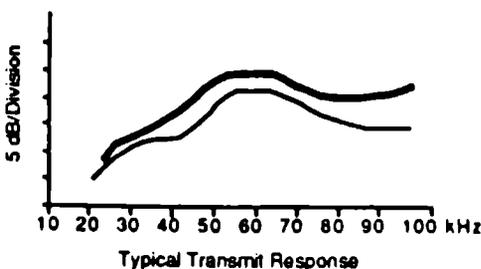
Comparison Chart Electrostatic Transducers

System Specifications	Instrument Grade Part# 604142/616341	*Environmental Grade Part# 607281/616342	7000 Series Part# 612366/616736
Distance Range	0.15 to 10.7m (0.5 to 35 ft.)	same	'same
Resolution ± 1% over entire range	±3mm to 3m (±.12 to 10 ft.)	same	same
Operating Conditions			
• Temperature	-30° to 70°C (-20° to 160°F)	same	0° -60°C (32°-140°F)
• Relative Humidity	5% to 95%	same	same
Beam Angle Typical, at 3dB down	12°	12°	17°
Transducer Drive Signal			
• Gated Sine Wave	50 kHz	same	same
• Duration	1.1 ms	same	same
• Suggested AC Drive Voltage (peak)	150v	same	200 V
• Bias Level	150 vdc	same	200 V
• Maximum Combined Voltage	400v	same	same
Min. Transmitting Sensitivity at 50 kHz • 300 vac pk-pk, 150 vdc bias • (dB re 20 µPa at 1 meter)	110dB	same	106.9dB
Min. Receiving Sensitivity at 50 kHz, • 50 kHz, 150 vdc bias (dB re 1v/Pa)	-42dB	same	-43.4dB
Capacitance at 1 kHz (typical)	380-410 pf	same	650 pf
Power Requirements Ranging Module			
• Voltage	6 vdc (4.5 – 6.8 vdc)	same	same
• Current	2.0 amps (1 ms pulse) 100 ma quiescent	same same	same same
Standard Finish			
• Foil	gold	same	same
• Housing	flat black cold roll steel	304 stainless steel	flat black plastic
Weight			
• Transducer	8.2 gm (0.29oz)	same	4.6 gm (0.150 oz)
• Modules	18.4 gm (0.94oz)	same	same
Dimensions			
• Thickness	0.46 in.	same	0.41 in.
• Diameter	1.69 in.	same	1.13 in. sq.

* Based on flat target 1 ft. sq. For irregular shape targets far range detection will be less.

* Environmental system is able to withstand salt spray, shock and vibration, water immersion, chemical exposure and sand bombardment when housed in "The Polaroid Test Enclosure."

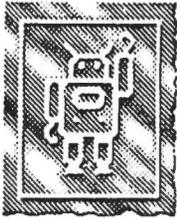
Specifications subject to change without notice.



For additional information, technical assistance or prices and delivery:

Polaroid OEM Components Group

Tel: (617) 386-3965 Fax: (617) 386-3966



Servo Modification For Continuous Rotation

How to convert a standard R/C servo into a continuously rotating drive motor.

This procedure results in a servo motor having continuous rotation in both forward and reverse, and with some degree of speed control from slow to full in both directions. BUT, the servo loses all of its position control abilities. *Study these notes carefully before beginning.*

Notice: This procedure is provided for information purposes only, and is recommended only for experienced hobbyists. Caution: This procedure involves making permanent, irreversible changes (that are not at all undoable, even), and that will certainly and undoubtably void the product's warranty. Warning: Product details and designs may vary greatly between manufacturers and even within the same product over time. Inspect your servo's actual design and features carefully before proceeding with any modifications. Mondo-tronics makes no claims to the accuracy or fitness of this information for any particular use, or appropriateness for any particular device. Proceed at your own risk.

Parts Required

- 1 each Hitec HS-300 servo (or similar servo)
- 2 each Resistor, 2.2K ohm 1/4 Watt 5%
- 1 each Shrink tubing 3/8" or black electrical tape

Tools Required

- Small "+" screwdriver
- Small "-" screwdriver
- Hobby knife
- Side cutters
- Needle nose pliers
- Soldering iron
- Solder

Modification Procedure

- 1) Open the servo housing by removing the 4 bottom screws.
- 2) Carefully remove the top cover, and remove all the gears and bearing ring – *making sure to note their positions for proper reassembly.* Use care to not loose any of the parts.
- 3) Locate the last and largest gear in the drive train, the one with the shaft that extends outside the case.
- 4) Locate the stop molded into the top surface of the gear. Carefully trim it off with a sharp hobby knife.
- 5) Use a needle nose pliers to unscrew retaining nut holding the potentiometer shaft in the servo housing.
- 6) Remove the bottom cover from the servo and gently lift out the small printed circuit board (PCB).
- 7) Press on the shaft of the potentiometer and remove it from the servo housing.

- 8) Cut the three wires on the potentiometer in half. Then trim and tin the ends of the wires still attached to the PCB with solder.
- 9) Twist the two 2.2K resistors closely together and solder them. When cool, trim all 3 ends to about 3 mm long.
- 10) Solder the red wire from the PCB to one end of the resistor pair, the green to the other end of the pair, and the yellow to the center where the two resistors join.
- 11) Slide the resistors into shrink tubing and shrink it (or wrap in tape) to prevent electrical contact.
- 12) Unsolder the wires remaining on the potentiometer .
- 13) Pry up on the retaining tabs on the bottom of the potentiometer and remove the bottom cover.
- 14) Clip off the wiper fingers inside the potentiometer .
- 15) Locate the stop notch on the top side of the potentiometer housing. Use a side cutter to make two sharp cuts into the body of the potentiometer on either side of the stop notch.
- 16) Bend and break away the potentiometer wall, and flatten the stop notch with pliers, using caution to not damage any other part of the potentiometer. Check that the potentiometer can now rotate freely. If not trim or flatten any remaining obstruction.
- 17) Return the potentiometer into the servo body, and secure it with the washer and nut.
- 18) Carefully pack the new resistor pair and shrink tube into the servo body, and replace the PCB. Make sure the resistors do not touch or interfere with the rotation of the potentiometer .
- 19) Carefully reassemble the gears and bearing ring, returning them to their original positions.
- 20) Replace the top and bottom covers. Secure with 4 screws.

Test the Modified Servo

- 1) Connect the servo to an appropriate R/C receiver or electronic driver circuit.
- 2) Apply power.
- 3) Send a "high" or "forward" signal. The servo should rotate in one direction continuously at full speed.
- 4) Send a "low" or "reverse" signal. The servo should rotate in the opposite direction at full speed.
- 5) If you are using a computerized driver circuit (like the Mini Serial Servo Controller, #3-205) you should be able to find an exact value (usually in the range of 150 to 220) that will cause the servo to stop rotating entirely. Values slightly to either side of this number will produce slowly increasing rotation in both directions, permitting some degree of speed control.
- 6) Double check that all screws are secure.
- 7) Label the servo as "modified for continuous rotation."

Advanced Tip

You may use 1/8 Watt 1% precision resistors for greater consistency when modifying more than one servo.

© MCMXCV Mondo-tronics Inc

REFERENCIAS

TEXTOS ([TX-xx])

- [TX-1] Timothy J. Ross, Fuzzy Logic with Engineering Applications, McGraw Hill 1995, ISBN : 0-07-053917-0
- [TX-2] Phillip J. McKerrow, Introduction to Robotics, Addison Wesley 1991, ISBN : 0-201-18240-8
- [TX-3] Anita M. Flynn & Joseph L. Jones, Mobile Robots, Inspiration to Implementation, A.K. Peters 1993, ISBN : 1-56881-011-3.
- [TX-4] S.S. Iyengar & Alberto Elfes, Autonomous Mobile Robots : perception, mapping and navigation, IEEE Computer Society Press 1991, Vol I, ISBN : 0-8186-2018-8.
- [TX-5] S.S. Iyengar & Alberto Elfes, Autonomous Mobile Robots : perception, mapping and navigation, IEEE Computer Society Press 1991, Vol II, ISBN : 0-8186-2116-8.
- [TX-6] Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems, Prentice Hall 1990, ISBN : 0-13-209768-0
- [TX-7] Rich E. and Knight K., Artificial Intelligence, International Edition, McGraw Hill 1991, ISBN : 0-07-100894-2
- [TX-8] Jimeno Garcia E., Alberdi Primicia J., Cabrera Jiménez J.A., Qué es nuevas tecnologías, Ediciones Granada, 1989.
- [TX-9] Eugene I. Rivin, Mechanical Design of Robots, McGraw Hill 1988, ISBN : 0-07-052992-2
- [TX-10] F.L. Chernousko, N.N. Bolotnik, V.G. Gradetsky, Manipulation Robots : Dynamics, Control and Optimization, CRC Press 1994, ISBN : 0-8493-4457-3
- [TX-11] M. Vukobratovic, D. Stokic, Applied Control of Manipulation Robots : Analysis, Synthesis and Exercises, Springer-Verlag 1989, ISBN : 0-387-514669-4
- [TX-12] M.W. Spong, M. Vidyasagar, Robot Dynamics and Control, John Wiley & Sons Inc. 1989, ISBN : 0-471-50352-5
- [TX-13] Ben-Zion Sandler, Robotics: Designing the Mechanisms for Automated Machinery, Prentice Hall 1991, ISBN : 0-13-781600-6
- [TX-14] K. S. Fu, R. C. Gonzalez, C. S. G. Lee, Robótica : Control, Detección, Visión e Inteligencia, McGraw Hill 1987, ISBN : 84-7615-214-0
- [TX-15] G. J. Klir, Bo Yuan, Fuzzy Sets and Fuzzy Logic : Theory and Applications, Prentice Hall 1995, ISBN : 0-13-1011711-5
- [TX-16] Fu Limin, Neural Networks in Computer Intelligence, McGraw Hill 1994, ISBN : 0-07-113319-4.

- [TX-17] Rao Vemuri, V. Artificial Neural Networks, Concepts and Control Applications, IEEE Computer Society Press Tutorial, 1992, pp. 437-506
- [TX-18] Tomas Kailath, Linear Systems, Prentice Hall 1980, ISBN : 0-13-536961-4.
- [TX-19] Mark Beale, Howard Demuth, Fuzzy Systems Toolbox, For use with MATLAB MS/DOS Version", PWS Productivity Tools 1994, ISBN : 053494578-3.
- [TX-20] Barrera Torres, De La Merced Sanchez. Proyecto de Robot Manipulador Actuado Hidraulicamente de 3 Grados de Libertad PROMAHI-3GL, Tesis de licenciatura Universidad Autónoma Metropolitana - Unidad Azcapotzalco, Depto. de Energía, Area Mecánica, 1994.
- [TX-21] Kosko B., Neural Networks and Fuzzy Systems, Prentice Hall Englewood NJ, 1991.
- [TX-22] John Durkin , Expert Systems, " design and development", Prentice Hall 1995, ISBN: 0-02-330970-0.
- [TX-23] Suran Goonatilake, S. Khebbal, Intelligent Hybrid Systems, Wiley & Sons Inc. 1995, ISBN: 0-471-94242-1.
- [TX-24] Witold P., Fuzzy Control & Fuzzy Systems, Wiley & Sons, ISBN: 0-471-93475-5.
- [TX-25] Eshragh F. and Mamdani E. H., A General Approach to Linguistic Approximation, in Fuzzy Reasoning and Its Applications, Mamdani E. H. and Gaines B. R. Eds, Academic Press, New York 1981.

MEMORIAS, PROCEEDINGS Y JOURNALS ([MPJ-xx])

- [MPJ-1] IASTED International Journal of Robotics and Automation, ISSN : 0826-8185, Volume 10, Issue 1, 1995.
- [MPJ-2] IASTED International Journal of Robotics and Automation, ISSN : 0826-8185, Volume 10, Issue 2, 1995.
- [MPJ-3] IASTED International Journal of Robotics and Automation, ISSN : 0826-8185, Volume 10, Issue 3, 1995.
- [MPJ-4] IEEE Transactions on Robotics and Automation, ISSN : 1042-296X, Volume 11, Number 3 (June 1995).
- [MPJ-5] IEEE Transactions on Robotics and Automation, ISSN : 1042-296X, Volume 11, Number 4 (August 1995).
- [MPJ-6] IEEE Transactions on Robotics and Automation, ISSN : 1042-296X, Volume 11, Number 5 (October 1995).
- [MPJ-7] Proceedings of the Third International Conference in Robotics and Manufacturing, June 1995, Cancun, Mexico, Editor : R.V. Mayorga, ISBN : 0-88986-220-6
- [MPJ-8] Memorias Simposium Internacional de Computación : tendencias de la computación hacia el nuevo milenio, Unidad profesional Jaime Torres Bodet, IPN, Noviembre 1995.
- [MPJ-9] IEEE Transactions on Robotics and Automation, ISSN : 1042-296X, Volume 11, Number 6 (December 1995).
- [MPJ-10] IEEE Transactions on Robotics and Automation, ISSN : 1042-296X, Volume 12, Number 1 (February 1996).
- [MPJ-11] IASTED International Journal of Robotics and Automation, ISSN : 0826-8185, Volume 10, Issue 4, 1995.

ARTÍCULOS ([AR-xx])

- [AR-1] Hans P. Moravec, The Stanford Cart and the CMU Rover, Proceedings of the IEEE, Vol. 71, No. 7, July 1983, pp. 872-884.
- [AR-2] Matthies, L., Elfes, A., Integration of Sonar and Stereo Range Data Using a Grid-Based Representation, IEEE Int. Conference on Robotics and Automation, 1988, pp. 727-733.
- [AR-3] Hebert M., Kanade T., Kweon I., 3-D Vision Techniques for Autonomous Vehicles, NSF Range Image Understanding Workshop, 1988, pp. 273-337.
- [AR-4] Rajkumar S. D., Raymond L., James E. W., Region Based Route Planning : multi-
abstraction route planning based on intermediate level vision processing, SPIE Vol. 1003 Sensor Fusion : Spatial Reasoning and Scene Interpretation, 1988.
- [AR-5] Buchberger M., Jörg Klaus-Werner, Von Puttkamer E., Laser Radar and Sonar Based World Modeling and Motion Control for Fast Obstacle Avoidance of the Autonomous Mobile Robot MOBOT-IV.
- [AR-6] R. Akin, Navigation Path Planning for Visual-Based Mobile Robot, Robotica, 7:49-63, 1989.
- [AR-7] Hart P.E., Nilson N.J., Raphael B., A Formal Basis for Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetics, Vol. SSC-4, No. 2, July 1968, pp. 100-107.
- [AR-8] T. Lozano-Pérez, M. A. Wesley, An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles, Communications of the ACM, Vol. 22, No. 10, 1979, pp. 560-570.
- [AR-9] T. lozano-Pérez, Spatial Planning: A Configuration Space Approach, IEEE Transactions on Computers, Vol. C-32, No. 2, February 1983.
- [AR-10] J.S.B. Mitchel, An Algorithmic Approach to Some Problems in Terrain Navigation", Artificial Intelligence, 37, 1988, pp. 171-201.
- [AR-11] O. Khatib, " Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, The Int. Journal of Robotics Research, Vol. 5, No. 1, Spring 1986, pp. 90-98.
- [AR-12] E. Freund, H. Hoyer, Real-Time Pathfinding in Multirobot Systems Including Obstacle Avoidance, The Int. Journal of Robotics Research, Vol. 7, No. 1, February 1988, pp. 42-70.
- [AR-13] B.H. Krogh, D. Feng, Dynamic Generation of Subgoals for Autonomous Mobile Robots Using Local Feedback Information, IEEE Transactions on Automatic Control, Vol. 34, No. 5, May 1989, pp. 483-493.
- [AR-14] D. Feng, D. Krogh, Satisficing Feedback Strategies for Local Navigation of Autonomous Mobile Robots, IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, No. 6, Novembre / December 1990.
- [AR-15] R.S. Doshi, R. Lam, J.E. White, Region-Based Route Planning : Multi-Abstraction Route Planning Based on Intermediate Level Vision Processing, SPIE Vol. 1003, Sensor Fusion : Spatial Reasoning and Scene Interpretation, 1988.
- [AR-16] S.S. Iyengar, C.C. Jorgensen, S.V.N.Rao, C.R. Weisbin, Robot Navigation Algorithms Using Learned Spatial Graphs, Robotica, Vol. 4, Part 2, April-June 1986, pp. 93-100.
- [AR-17] B.J. Oommen, S.S. Iyengar, N.S.V. Rao, R.L. Kashyap, Robot Navigation in Unknown Terrains Using Learned Visibility Graphs. Part I: The Disjoint Conex Obstacle Case, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 6, December 1987, pp. 672-681.
- [AR-18] N.S.V. Rao, S.S. Iyengar, Autonomous Robot Navigation in Unknown Terrains :
Incidental Learning and Environmental Exploration, IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, No. 6, November / December 1990.

- [AR-19] V.J. Lumelsky, S. Mukhopadhyay, K. Sun, Dynamic Path Planning in Sensor-Based Terrain Acquisition, IEEE Transactions on Robotics and Automation, Vol. 6, No. 4, August 1990.
- [AR-20] J.T. Schwartz, M. Sharir, A Survey of Motion Planning and Related Geometric Algorithms, Artificial Intelligence Journal, Vol. 37, 1988, pp. 157-169.
- [AR-21] B. Beaufrère, S. Zegloul, Avoidance of Moving Obstacles by Mobile Robot, Proceedings of the Third Int. Conference on Robotics and Manufacturing, Cancún, México 1995, pp. 237-240.
- [AR-22] B. Beaufrère, S. Zegloul, Navigation Method for a Mobile Robot Using a Fuzzy Based Method : Simulation and Experimental Aspects, International Journal of Robotics and Automation, Vol. 10, Issue 3, 1995, pp. 106-112.
- [AR-23] Rathbone R. R., Valley R. A., Kindlmann P. J., Beacon-Referenced Dead Reckoning : A Versatile Guidance System, Robotics Engineering, Vol. 8, pp. 11-16.
- [AR-24] Tachi S., Komoriya K., Guide Dog Robot, Int. Symposium on Robotics Research, Vol. 2, The Second International Symposium, MIT Press. pp. 333-340.
- [AR-25] H. Hellendoorn, C. Thomas, Defuzzification in Fuzzy Controllers, Intelligent and Fuzzy Systems, Vol. 1, pp. 109-123.
- [AR-26] Nilsson N. J., A Mobile Automaton: An application of artificial intelligence techniques, Proc. Int. Conference Artificial Intelligence 1969, pp. 509-520.
- [AR-27] Ignat'yev M. B., Kulakov F. M., Pokrovskiy A. M., Robot Manipulator Control Algorithms, Rep. No. JPRS 59717, NTIS, Springfield, VA, August 1973.
- [AR-28] Leven D., Sharir M., An Efficient and Simple Motion Planning Algorithm for a Ladder Moving in Two-Dimensional Space Amidst Polygonal Barriers, J. Algorithms, 1987, pp. 192-215.
- [AR-29] Sharir M., Ariel-Sheffi E., On the Piano Mover's problem IV: Various Decomposable Two-Dimensional Motion planning Problems, Comm. Pure Appl. Math. 1984, pp. 479-493.
- [AR-30] Schwartz J. T., Sharir M., On the Piano Mover's Problem III: Coordinating the Motion of Several Independent Bodies : The special case of circular bodies moving amidst polygnal barriers, Rob. Res. 2, 1983, pp. 46-75
- [AR-31] O'Dunlaing C., Yap C., A Retraction Method for Planning the Motion of a Disc, J. Algorithms 1985, pp. 104-111.
- [AR-32] Chatila R., Mobile Robot Navigation: Space Modelling and Decisional Processes, 3rd ISRR 1985, Gouvieux, France.
- [AR-33] Asano T., Guibas L., Hershberger J., Imai H., Visibility Polygon Search and Euclidean Shortest Paths, Proceedings 26th Symposium on Foundations of Computer Science 1985, pp. 155-164.
- [AR-34] Welzl E., Constructing the Visibility Graph for n Line Segments in $O(n^2)$ Time, Information Proceedings Lett. 1985, pp. 167-172.
- [AR-35] Reif J. H., Storer J. A., Shortest Paths in Euclidean Space with polyedral Obstacles, Tech. Report CS-85-121, Computer Science Department, Brandeis University, Waltham, MA 1985.
- [AR-36] Reif J. H., Sharir M., Motion Planning in the Presence of Moving Obstacles, Tech. Rept. 39 / 85, The Eskenasy Institute of Computer Science, Tel-Aviv University, 1985.
- [AR-37] B. Baker, Shortest Paths with Unit Clearance Among Polygonal Obstacles, SIAM Conference on Geometric Modeling and Robotics, Albany, NY, July 15-19, 1985.
- [AR-38] L. P. Chew, Planning the Shortest Path for a Disk in $O(n^2 \log n)$ Time, Proc. First Annual ACM Conference on Computational Geometry, Baltimore, MD, 1985, pp. 214-220.
- [AR-39] J. S. B. Mitchell, Planning Shortest Paths, Ph.D. Thesis, Department of Operations Research, Stanford University, August 1986.

- [AR-40] Stentz A., Optimal and Efficient Path Planning for Unknown and Dynamic Environments, IASTED International Journal of Robotics and Automation, Vol. 10, Issue 3, 1995, pp. 89-100.
- [AR-41] D. Tilbury, et. al., A Multisteering Trailer System: Conversion into Chained Form Using Dynamic Feedback, IEEE Transactions on Robotics and Automantion, Vol. 11, Number 6, December 1995, pp. 807-818.
- [AR-42] G. Desaulniers, F. Soumis, An Efficient Algorithm to Find a Shortest Path for a Car-Like Robot, IEEE Transactions on Robotics and Automantion, Vol. 11, Number 6, December 1995, pp. 819- 828.
- [AR-43] S. Negahdaripour, B.Y. Hayashi, Y. Aloimonos, Direct Motion Stereo for Pasive Navigation, IEEE Transactions on Robotics and Automantion, Vol. 11, Number 6, Dec. 1995, pp. 829-843.
- [AR-44] K.K. Gupta, Z. Guo, Motion Planning for Many Degrees of Freedom: Sequential Search with Backtracking, IEEE Transactions on Robotics and Automantion, Vol. 11, Number 6, December 1995, pp. 897-905.
- [AR-45] Y. Yagi, Y. Nishizawa, M. Yachida, Map-Based Navigation for a Mobile Robot with Omnidirectional Image Sensor COPIS, IEEE Transactions on Robotics and Automantion, Vol. 11, Number 6, October 1995, pp. 634-648.
- [AR-46] Y.-H. Liu, S. Arimoto, Finding the Shortest Path of a Disc Among Polygonal Obstacles Using a Radius-Independent Graph, IEEE Transactions on Robotics and Automantion, Vol. 11, Number 6, October 1995, pp. 682-691.
- [AR-47] F. Xi, R.G. Fenton, Point-to-Point Quasi-Static Motion Planning for Flexible-Link Manipulators, IEEE Transactions on Robotics and Automantion, Vol. 11, Number 6, October 1995, pp. 770-776.
- [AR-48] R. Andrew Russell, Laying and Sensing Odor Markings as a Strategy for Assisting Mobile Robot Navigation Tasks, IEEE Robotics & Automation Magazine, Vol. 2 No. 3, September 1995, pp. 3-9.
- [AR-49] O'Connor, Leo, High-tech Security Devices, ASME Mechanical Engineering Magazine, Vol. 117 / No. 11, November 1995.
- [AR-50] Martinez Alfaro H., Ulloa Pérez A., Planeación de Trayectorias en Espacio-C Mediante B-Splines y Recocido Simulado, Memorias XII Reunion Nacional de Inteligencia Artificial, Septiembre 1995, pp. 63-69.
- [AR-51] Michel J. Dreyfus Leon, Clasificación de Patrones para la Selección de Movimiento en Animales Artificiales, Memorias XII Reunion Nacional de Inteligencia Artificial, Septiembre 1995, pp. 136-139.
- [AR-52] C. Touzet, N. Giambiasi, S. Sehad, Improving Reinforcement Learning of an Obstacle Avoidance Behavior with forbidden Sequences of Actions, Proceedings of the Third IASTED International Conference on Robotics and Manufacturing, Cancun Mexico, June 1995, pp. 43-45.
- [AR-53] R. Hartley, Performance Evaluation of Navigation Algorithms Using Percolation Theory, Proceedings of the Third IASTED International Conference on Robotics and Manufacturing, Cancun Mexico, June 1995, pp. 46-50.
- [AR-54] G. Bessonnet, A.D. Jutard-Malinge, Optimal Free Path Planning of Robot Arms Submitted to Phase Constraints, Proceedings of the Third IASTED International Conference on Robotics and Manufacturing, Cancun Mexico, June 1995, pp. 51-56.
- [AR-55] J.M. Ahuactzin, K. Gupta, Using Manhattan Paths for Manipulator Planning, Proceedings of the Third IASTED International Conference on Robotics and Manufacturing, Cancun Mexico, June 1995, pp. 57-60.
- [AR-56] R.V. Mayorga, A Framework for the Path Planning of Robot Manipulators, Proceedings of the Third IASTED International Conference on Robotics and Manufacturing, Cancun Mexico, June 1995, pp. 61-66.

- [AR-57] S.R. Kundur, D. Raviv, Visual Motion Cue for Autonomous Navigation, Proceedings of the Third IASTED International Conference on Robotics and Manufacturing, Cancun Mexico, June 1995, pp. 67-69.
- [AR-58] -J.F. Gardner, K.A. Hrnciar, Trajectory Control Using Neural Networks to Improve Desired Trayectory Input, Proceedings of the Third IASTED International Conference on Robotics and Manufacturing, Cancun Mexico, June 1995, pp. 163-166.
- [AR-59] Ramirez A., Boumedine M., Development of a Fuzzy Expert System for Autonomous Navigation of Mobile Robots, Memorias Simposium Internacional de Computación, IPN, México D.F, Noviembre 1995 (sección redes neuronales).
- [AR-60] B. Barsham, H.F. Durrant-Whyte, Inertial Navigation Systems for Mobile Robots, IEEE Transactions on Robotics and Automation, June 1995, Vol. 11, No. 3, pp. 328-342.
- [AR-61] K. Fujimura, Time-Minimum Routes in Time-Dependent Networks, IEEE Transactions on Robotics and Automation, June 1995, Vol. 11, No. 3, pp. 343-351.
- [AR-62] S. Fleury, P. Souères, J.-P. Laumond, R. Chatila, Primitives for Smoothing Mobile Robot Trajectories, IEEE Transactions on Robotics and Automation, June 1995, Vol. 11, No. 3, pp. 441-447.
- [AR-63] K. Glass, R. Colbaugh, D. Lim, H. Seraji, Real-Time Collision Avoidance for Redundante Manipulators, IEEE Transactions on Robotics and Automation, June 1995, Vol. 11, No. 3, pp. 448-456.
- [AR-64] Brooks, Rodney A., Intelligence Without Reason, Prepared for Computers and Thoughts, IJCAI-91.
- [AR-65] Dorigo M., Genetic and Nongenetic Operators in ALECSYS, International Computer Science Institute, Berkeley, CA 94704, November 1992.
- [AR-66] Dorigo M., Colombetti M., Robot Shaping : Devepolping Situated Agents through Learning, International Computer Science Institute, Berkeley, CA 94704, April 1993.
- [AR-67] Dorigo M., Colombetti M., Training Agents to Perform Sequential Behavior, International Computer Science Institute, Berkeley, CA 94704, September 1993.
- [AR-68] Hartmut Surman, Jörg Huser, Liliane Peters, Fuzzy System for Indoor Mobile Robot Navigation, Proc. of the Fourth IEEE Int. Conf. on Fuzzy Systems, Yokohama, Japan, 1995, pp. 83-86.
- [AR-69] Jörg Huser, Hartmut Surman, Liliane Peters, A Fuzzy System for Realtime Navigation of Mobile Robots, 19th Annual German Conf. on AI, Bielfeld, 1995, pp. 170-172.
- [AR-70] R.A. Jarvis, A Perspective on Range Finding Techniques for computer Vision, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No.2, March 1983, pp. 122-139.
- [AR-71] R.Y. Tsai, Synopsis of Recent Progress on Camera calibration for 3-D Machine Vision, Robotics Review 1, 1989, pp. 147-159.
- [AR-72] Inigo, Rafael M, Alley D., Path Planning Algorithms for Mobile Robots, H0333-5/90/0000/1032, IEEE 1990, pp. 1032-1036.
- [AR-73] Warren Charles W., Fast Path Planning Using A* Method, 1050-4729/93. IEEE 1993, pp. 662-667.
- [AR-74] R.L. Madarasz, et al, The Design of an Autonomous Vehicle for the Disabled, IEEE Journal of Robotics and Automation, Vol. RA-2, No. 3, Sep. 1986, pp. 117-126.
- [AR-75] M. Fujie, et al, Mobile Robot System with transformable Crawler, Intelligent Guidance, and Flexible manipulator, Intenational Symposium on Robotics Research, Vol. 3, pp. 341-347.
- [AR-76] J. Bares, M. Hebert, T. Kanade, E. Krotkov, Amber-An Autonomous Rover for Planetary Exploration, IEEE Computer Magazine, June 1989, pp. 18-26.
- [AR-77] Jarvis, R.A., Collision-free Trayectory Planning Using Distance Transforms, Mech. Eng. Trans. of the I E Aust. ME10, 3, pp. 187-191.

- [AR-78] Raj Talluri, J.K. Aggarwal, Mobile Robot Self-Location Using Model-Image Feature Correspondence, IEEE Transactions on Robotics and Automation, February 1996, pp. 63-77.
- [AR-79] Kikuo Fujimura, Path Planning with Multiple Objectives, IEEE Robotics & Automation Magazine, March 1996, pp. 33-38.
- [AR-80] S. Du, J. Tang, G. Yang, A New Method of Discrete Trajectory Planning for Robot Manipulators, IASTED International Journal of Robotics & Automation, Vol. 10 Issue 4 1995, pp. 123-129.
- [AR-81] A. Basu, A. Elnagar, Safety Optimizing Strategies for Local Path Planning in Dynamic Environments, IASTED International Journal of Robotics & Automation, Vol. 10 Issue 4 1995, pp. 130-142.
- [AR-82] Ramírez Serrano A., Boumedine M., Navigation in Unknown Static Environments Using Fuzzy Logic and Ultrasonic Sensing, IEEE Int. Symposium on Intelligent Control (ISIC 96), September 15-18, 1996 (en revisión).
- [AR-83] Ramírez Serrano A., Boumedine M., Navigation in Unknown Environments Using Fuzzy Logic, Congreso Iberoamericano de Inteligencia Artificial, IBERAMIA 96 (en revisión).
- [AR-84] Ramírez Serrano A., Boumedine M., Real-Time Fuzzy Logic Navigation of Mobile Robots, IASTED Int. Conference on Artificial Intelligence, Expert Systems and Neural Networks, August 1996, Honolulu Hawaii, U.S.A., (en revisión).
- [AR-85] Ramírez Serrano A., Boumedine M., Ultrasonic Sensing and Fuzzy Logic Control for Navigation in Unknown Static Environments, 1st Euromicro Workshop on Advanced Mobile Robots, Kaiserslautern Germany, October 1996, (en revisión).
- [AR-86] Kraus, Wilkenfeld and Zlotkin, Multiagent Negotiation Under Time Constraints, Artificial Intelligence 1995.

REVISTAS ([RE-xx])

- [RE-1] IEEE Robotics & Automation Magazine, ISSN 1070-9932, IEEE Robotics & Automation Society.
- [RE-2] American Society of Mechanical Engineers (ASME) Mechanical Engineering Magazine.
- [RE-3] IEEE EXPERT Intelligent Systems & Their Applications, Magazine from the IEEE Computer Society

DIRECCIONES DEL Word-Wide-Web (www)

- Polaroid Corporation: <http://www.polaroid.com/>
- Motorola Corporation: <http://www.motorola.com/>
- Honeywell Corporation: <http://www.ssec.honeywell.com>
- Robix RCS-6 Robot construction set: <http://www.robix.com/>
Educational / Research /Kit. Makers of Robix (tm) Robotic Construction Set
- Fuzzy Logic Inf.: http://www.yahoo.com/Science/Computer_Science/Artificial_Intelligence/Fuzzy_Logic/
FuzzyCLIPS, Fuzzy inf., Fuzzy Logic, Fuzzy Logic and Probability, Fuzzy Logic. Archieve, Fuzzy Logic Explanation, Fuzzy Systems Tutorial, Int. journal of uncertainty, fuzziness and knowledge-based systems and Neurofuzzy resources page.

- IEEE Robotics & Automation Society: <http://www.acim.usl.edu/RAS/>
- AutomationNET Sensores & Measuring Devices: <http://www.automationnet.com/-Azure/>
Proximity sensors, Resolvers, Rotary pos. sensors, Tactile sensors, Encoders, etc.
- JRC/ESBA Robotics Group: <http://esba-www.jrc.it/mrgdocsPage.html>
Mobile robotics group , electronics & sensor based applications/ISEI, JRC
Euro-Robots : Mobile robots archive (front page).
- Directory of /pub/robots/RobotGallery/ : <ftp://ftp.essex.ac.uk/pub/robots/RobotGallery/>
- Microrobotics Reseach at MIT AI Lab.: <http://www.ai.mit.edu/projects/microrobotics/>
- Robotics Internet Resources Page: <http://piglet.cs.umass.edu:4321/robotics.html>
Varias ligas hacia sitios de interes referentes a robótica: Universidades, Productos comerciales, FTP, Software, Demos, Mondo-Tronics, Laboratorios, Concursos, etc.
- Problemas con sensores de ultrasonido: <http://esba-www.jrc.it/mrgdoes/robotPage.html>
Specular reflections of the ultrasonic beam & Crosstalk in a mutli-sensor environment.

CATÁLOGOS ([CA-xx])

- [CA-1] NAMCO, EE-ER-ET General purpose proximity sensors, General Catalog B
- [CA-2] Solid State Electronics Center, Honeywell Inc., 12001 State Hwy 55, Plymouth MN 55441, Permalloy Magnetic Sensors Booklet.
- [CA-3] 1996 IEEE Int Conference on Robotics & Automation, Advanced Program.

COMPAÑÍAS ([Co-xx])

- [Co-1] A.K. Peters, Publisher of the book " Mobile Robots, Inspiration to Implementation" by Anita Flynn and distributor of Rug Warrior kits and IC" , 289 Linden St., Wellesley, MA 02181-5910, Phone : (617) 235-2210, Fax : (617) 235-2404, E-mail : kpeters@geom.umn.edu o kpeters@math.harvard.edu
- [Co-2] Polaroid Corporation, Fabricante de transductores electrostáticos (para usos de ultrasonido) de la serie 7000 (part # 612366/616736), Dirección en USA : Ultrasonic Components Group, 119 Windsor Street, Cambridge, Massachusetts 02139, Phone : (617) 386-3965, Fax : (617) 386-3966, email : MAILW@cliffy.polaroid.com , Dirección en México : Polaroid de México S.A. de C.V., Paseo de la Reforma # 119-17 (a 3 cuadras del angel de la independendia), Col. Cuahatemoc 06500, Tel : (525) 703-1111, Fax : (525) 566-1359, México D.F.
- [Co-3] Rysh Electrónica S.A. de C.V., Distribuidor de Ultrasonic Transceivers (circuito LM1812), Norte 90 No. 4426-1, Col. Nueva Tenochtitlan, C.P. 07890, Tel : (525) 771-1893, Fax : (525) 771-1857 México D.F.
- [Co-4] Hobby Centro - Jobitiendas S.A. de C.V. , Proveedor de servo motores y accesorios para aeroplanos y carros de radio control (ruedas, engranes, motores, baterias, cargadores, etc), Av. Canal de Miramontes No. 3122, Col. Ex Hacienda de Coapa, C.P. 14300, Tel : (525) 627-8039, México D.F.
- [Co-5] Dimusa - Distribuidor de maquinaria universal S.A. de C.V. y distribuidor oficial de refacciones y accesorios para carros de radio control KYOSHO (sistemas de engranes, diferenciales, servomotores, ruedas, etc), Av. Pacífico 216 Coyoacan, C.P. 04330, Tels : (525) 544-7135 / 544-0809 / 544-0517 / 544-7134 / 689-3590 / 689-4150 / 689-2763 / 689-5529 /689-2855 / 549-6036, Fax : (525) 544-7133, México D.F.

[Co-6] **Mondo-Tronics Inc.** - Distribuidor de productos para la fabricación de robots experimentales (software, servo-motores, sensores, libros, sistemas de engranes, baterías recargables, etc.), Gerente general: Roger G. Gilbertson, 524 San Anselmo Ave # 107-117, San Anselmo, CA 94960, Tels: (415)-455-9330, Fax: (415)-455-9333, E-mail: mondo@mondo.com, <http://www.robotstore.com>, Order toll free: 800-374-5764.