

ESTUDIO COMPARATIVO DE TRES TÉCNICAS DE NAVEGACIÓN PARA ROBOTS MÓVILES

HERNANDO GONZÁLEZ ACEVEDO

*Candidato a Maestría En Ingeniería, Universidad Industrial De Santander
Grupo de Investigación en Automatización, Modelamiento,
Simulación y Control de Productos y Procesos
Universidad Santo Tomas – Seccional Bucaramanga
hernando_gonza@hotmail.com*

CARLOS ALBERTO MEJIA CASTAÑEDA

*Grupo de Investigación en Automatización, Modelamiento,
Simulación y Control de Productos y Procesos
Universidad Santo Tomas – Seccional Bucaramanga
ingcarlosmejia@hotmail.com*

*Fecha Recepción: 27/03/2007
Fecha Aceptación: 28/05/2007*

RESUMEN

El artículo se centra en la problemática de la navegación de robots móviles que se desplazan por un entorno parcialmente estructurado. El trabajo se divide en tres partes: diseño de un sistema mecatrónico, denominado Robot IO, estudio de las ventajas y desventajas de aplicar tres técnicas de navegación, redes neuronales, algoritmos genéticos y lógica difusa, por último la implementación de un algoritmo que permita identificar el espacio de trabajo por el cual se desplaza IO.

PALABRAS CLAVE: Robótica móvil, Redes RBF, Lógica difusa, Algoritmos genéticos, Mapas de ocupación.

ABSTRACT

This paper described the problem of path-planning in mobile robots in a partially structured environment. The work was divided in three parts: design of a mechatronic system, denominated Robot IO, study of advantages and disadvantages of three path-planning algorithms: neural nets, genetic algorithms and fuzzy logic, and implementation of algorithms that allow to identify the workspace where IO moves.

KEYWORDS: Mobile robots, Neural nets RBF, Fuzzy logic, Genetic algorithms, Occupation maps

1. INTRODUCCIÓN

En la actualidad se han desarrollado diversos métodos para lograr que un robot se mueva en un entorno para lograr variados propósitos como evadir obstáculos, navegar a un sitio específico o hacer un mapa interno de un entorno. Los métodos clásicos de planificación se fundamentan en una representación del espacio libre, según información obtenida del entorno previamente, para posteriormente emplear un algoritmo de búsqueda que encuentre el camino óptimo según cierta función de costo, entre ellos se encuentra los grafos de visibilidad y los diagramas

de Voronoy. Problemas de planeamiento de trayectorias complejos, como el enfrentarse a un laberinto o a un entorno variable ponen a prueba estos métodos, por lo tanto se hace necesario aplicar otros sistemas de control basados en técnicas de inteligencia artificial, como redes neuronales, lógica difusa o algoritmos genéticos. El primero tiene la capacidad de generalización para resolver problemas que se plantean en la navegación de un robot móvil en un ambiente desconocido, para tal fin, se debe seleccionar los casos de entrenamiento que permitan emular las posibles situaciones a las cuales se va enfrentar el móvil. La lógica difusa se puede definir como un

lenguaje que traslada sentencias sofisticadas del lenguaje natural a un formalismo matemático, tiene la ventaja de ser robusta frente a cambios en el sistema, así como su capacidad de manejar información que contiene ruido y gran incertidumbre, pero la probabilidad de resultados óptimos depende en un gran factor de la experiencia del programador. Los algoritmos genéticos se enfocan a la optimización de una función sin la restricción de tener una base de datos inicial. Esta característica la hace interesante si se trata de resolver problemas variables donde las soluciones parecen ser siempre únicas e independientes de cualquier otra solución.

Algunos autores han utilizado redes neuronales de tipo perceptrón [1] para el desarrollo de generadores de trayectorias, con la ventaja que pueden ser implementada sobre un microchip con tecnología FPGA, otros utilizan redes neuronales de base radial [2] [3] que permiten dar respuesta a casos que no son especificados en la etapa de entrenamiento, debido a su carácter probabilístico. La lógica fuzzy es otra alternativa, algoritmos basados en los conceptos de Mandani y Takani-Sugeno [4] [5] para el control de la dirección de un robot móvil en presencia de obstáculos fijos. Estas técnicas no garantizan la generación de una ruta óptima, caso contrario a los algoritmos genéticos, afirmación que se fundamenta en dos razones principales: es una técnica adecuada para realizar búsquedas en espacios de dimensión elevada e impone pocas restricciones de tipo matemático en la forma de la función a optimizar, de tal manera que es aplicable a la generación de trayectorias para cualquier tipo de comportamiento (evitar obstáculos, seguir paredes, etc), como los que se observan en las referencias [6] [7] [8].

En el presente artículo se propone comparar estas tres técnicas de navegación (redes neuronales, lógica difusa, algoritmos genéticos) sobre un entorno estático que el robot determinará previamente, conformado por pasillos, paredes u obstáculos en forma de T, L o U. Los indicadores que se utilizarán para comprobar la efectividad de cada una son el número de ocasiones en las cuales el robot no cumple con su objetivo, lo llevan a que se choque con un obstáculo y por último como es el desempeño del robot al seguir los movimientos determinados por las diferentes acciones de control.

2. ROBOT IO

El robot IO se implementó con el fin de verificar el desempeño de cada uno de los algoritmos de navegación. Se encuentra construido en piezas técnicas LEGO, dotado con un sensor de proximidad GP2D12 para detectar obstáculos entre 10 a 45 cm, cinco sensores

IS471F para distancias inferiores a 15 cm, cuatro CNY70 que permiten identificar una marca significativa en el escenario y dos encoders adaptados internamente a los motores para garantizar movimientos en línea recta. Además, se le adecuo un módulo de comunicación inalámbrico utilizando los integrados TLP-RLP434 de Laipac para el envío de toda la información captada a un computador, donde se ejecutan los algoritmos que rigen su comportamiento. El móvil se desplazará por un escenario de 2,4 metros de largo por 1,5 metros de ancho, dividido en celdas de 15x15 cm., lo que constituye un total de 160 posibles posiciones de exploración.

3. REDES NEURONALES DE BASE RADIAL

Las redes neuronales de base radial o RBF, a pesar de su reciente introducción, cada vez cuentan con más aplicaciones prácticas gracias a su simplicidad, generalidad y facilidad de aprendizaje. Las neuronas son de respuesta localizada, porque sólo responden con una intensidad apreciable, cuando el vector de entradas es cercano al valor del centroide de la neurona.

Para el desarrollo del algoritmo, la red neuronal se entrena en base de un aprendizaje híbrido, por lo tanto la matriz de entrenamiento debe incluir, además de los vectores de entradas, las salidas deseadas para cada uno de ellos. La red en función de la información que capta a través de los sensores tiene que proporcionar una acción de control adecuada para que el vehículo se desplace por el entorno sin colisionar. Las entradas a la red neuronal se codifican en un vector columna, como se observa en la Figura 1.

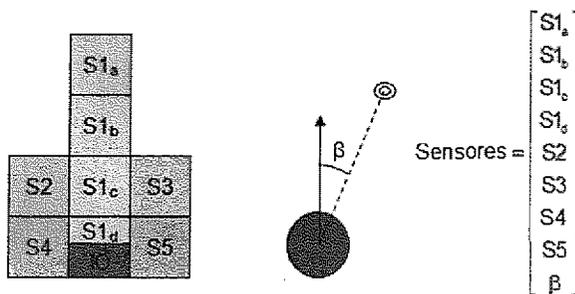


Figura 1. Datos de Entrada a una Red RBF

S_i representa la señal digitalizada del sensor GP2D12, indicando la distancia que se encuentra un objeto frente a IO, codificada de la siguiente forma

- [0,0,0,0] No existe obstáculo
- [x,x,x,1] Obstáculo a una distancia de 0 a 7 cm
- [x,x,1,0] Obstáculo a una distancia de 7 a 15 cm
- [x,1,0,0] Obstáculo a una distancia de 15 a 30 cm
- [1,0,0,0] Obstáculo a una distancia de 30 a 45 cm

S2 a S5 corresponde a la lectura digital de los sensores IS471F, asumiendo un valor de uno para la presencia de un obstáculo o cero en caso contrario. β es el ángulo que existe entre la horizontal y la línea imaginaria que une al robot con la meta, el cual se calcula por la ecuación (1),

$$\beta = a \tan\left(\frac{y_m - y_r}{x_m - x_r}\right) \quad (1)$$

donde (x_m, y_m) es la ubicación del punto de llegada y (x_r, y_r) la ubicación actual del robot. El ángulo se divide en pasos de cuarenta y cinco grados, obteniendo ocho valores posibles, los cuales se codifican en un número de -3 a 3 para los rangos que se observan en la tabla No 1.

Tabla 1. Codificación del Ángulo a Meta β

Código	Ángulo
0	$-22.5 < \beta < 22.5$
1	$22.5 < \beta < 67.5$
2	$67.5 < \beta < 112.5$
3	$112.5 < \beta < 157.5$
4	$157.5 < \beta < 202.5$
-1	$-22.5 > \beta > -67.5$
-2	$-67.5 > \beta > -112.5$
-3	$-112.5 > \beta > -157.5$

Se considera que el robot tiene en cada celda la posibilidad de moverse a la siguiente, en cualquier dirección. Según estos criterios las acciones básicas de control que debe proporcionar la red son el incremento en el eje X y eje Y del sistema de coordenadas.

El entrenamiento se realizó a partir de 41 casos, donde se tienen en cuenta diferentes posibilidades de activación de los sensores y para cada uno, el ángulo que hay entre la horizontal y la línea imaginaria que une al robot con su objetivo, dando un total de 287 patrones de entrenamiento. En la figura 2 se muestran algunos casos que se utilizaron para el entrenamiento de la red RBF

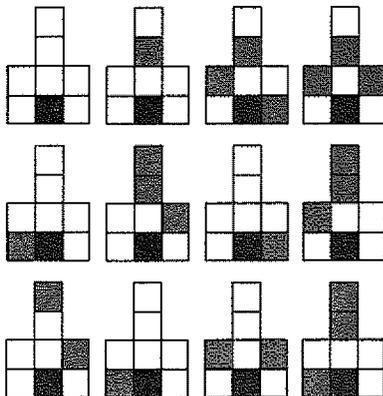


Figura 2. Casos de Entrenamiento para una Red RBF

Para programar la red se empleó el comando *newrb* de la toolbox de MATLAB definiendo un ancho de la campana de uno. En la Figura 3 se observan algunos resultados para entornos de distinta complejidad, destacándose que la lógica de control no genera cambios bruscos en la dirección.

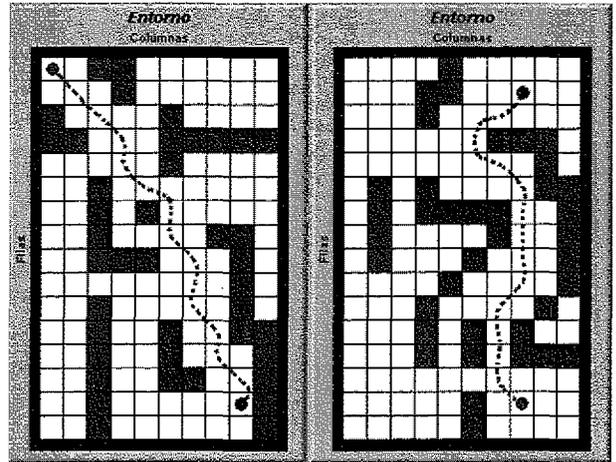


Figura 3. Resultados de la Simulación en un Entorno Aleatorio - Redes RBF

4. ALGORITMOS GENÉTICOS

Un algoritmo genético es una técnica de programación que imita la evolución biológica como estrategia para resolver problemas. El tamaño del cromosoma que se eligió para el planificador de trayectorias es variable, depende de la distancia que existe entre el punto de partida $(C_{inicial}, F_{inicial})$ y el punto de llegada (C_{final}, F_{final}) .

$$\text{Longitud} = 2 \cdot (|F_{final} - F_{inicial}| + |C_{final} - C_{inicial}|) \quad (2)$$

La ecuación (2) busca evitar procesos repetitivos, o generar individuos con muy poca información. Para el algoritmo, cada gen de un cromosoma se define como un movimiento respecto a la posición y orientación actual del robot, los cuales se realizan de forma secuencialmente.

Para el entorno de la figura 4, el algoritmo inicialmente genera tres individuos aleatoriamente.

- C1 = [1,5,3,1,1,1,4,1,1,0,0,0,0,0,0]
- C2 = [1,1,1,1,1,1,1,3,0,0,0,0,0,0,0]
- C3 = [2,2,1,1,1,1,1,3,1,2,1,1,1,3,3,3]

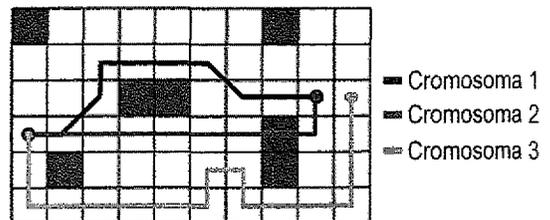


Figura 4. Secuencia de Movimientos Generados Aleatoriamente en un Entorno Estructurado

La primera acción es ejecutar los movimientos determinados por cada cromosoma. Si durante su ejecución se logra llegar al objetivo, el resto de los movimientos descritos se definen como nulos. Luego se procede a evaluar la función de aptitud definida por la ecuación (3)

$$\text{Aptitud} = 15 \cdot \sum M_{LR} + 20 \cdot \sum \frac{1}{\text{Mov}} - 50 \cdot \sum C - 10 \cdot d \quad (3)$$

La anterior expresión da un valor cuantitativo al comportamiento del robot durante la navegación, penalizando aquellos movimientos que lo llevan lejos de su objetivo (d) y lo hacen chocar con obstáculos del entorno (C), premiando a su vez los movimientos en línea recta (M_{LR}) y la menor cantidad de ellos (Mov). Los coeficientes que multiplican a cada factor se eligieron después de realizar varias pruebas en el simulador, en el cual se observó que era muy importante penalizar con un valor alto el caso en el cual el vehículo choca con un obstáculo y favorecer la menor cantidad de movimientos, porque se deseaba que la trayectoria fuera la óptima.

El individuo con el menor valor en la función de aptitud es reemplazado. El nuevo individuo se genera, bien sea por un proceso de mutación (probabilidad del 4%) o un proceso de cruce (probabilidad del 60%). El algoritmo se detiene si la trayectoria cumple con las siguientes condiciones: la cantidad de movimientos que se deben realizar es inferior a un 30% de la cantidad de cuadrículas que ocupan la diagonal que unen el punto de partida con la meta, además la cantidad C y d deben ser iguales a cero. El número máximo de iteraciones con las cuales se programó el algoritmo genético fue de 5000.

Para evaluar la efectividad de algoritmo genético se ejecuto 25 veces sobre los dos escenarios de la figura 5, obteniéndose un 85% y 79% respectivamente, de cumplimiento con el objetivo propuesto. Esta diferencia se debe al grado de aleatoriedad en el cual se basa la técnica.

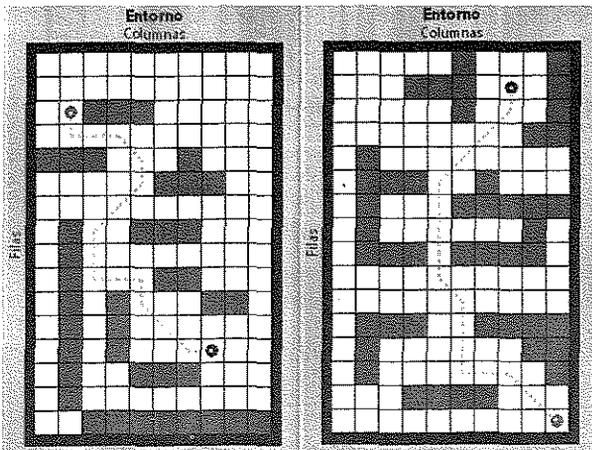


Figura 5. Resultados de la Simulación en un Entorno Aleatorio – Algoritmos Genéticos

5. LÓGICA DIFUSA

La lógica difusa se puede definir como un lenguaje que permite trasladar sentencias sofisticadas del lenguaje natural a un formalismo matemático. El sistema implementado lo conforman tres conjuntos difusos, el primero es denominado d_i , que relaciona la distancia entre el robot y el obstáculo más cercano, el segundo corresponde al ángulo entre ellos (α), y el tercero es el ángulo entre IO y la meta (β). El máximo rango de d_i lo determina el alcance del sensor GP2D12 (45 cm), el ángulo α se asume positivo en la dirección en contra de las manecillas del reloj (0° a 180°) y β es positivo en dirección de las manecillas del reloj (0° a 180°), como se observa en la figura 6.

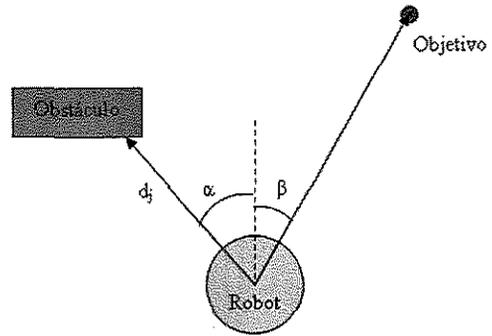
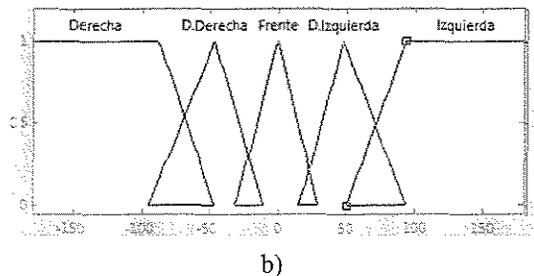
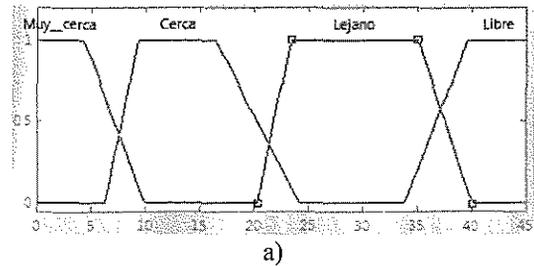
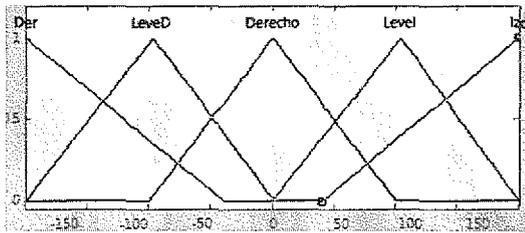


Figura 6. Variables del Controlador Difuso

Las funciones de pertenencia para los conjuntos de entrada (figura 7) y el de salida (figura 8) se determinaron de forma intuitiva, teniendo en cuenta la característica de los sensores y el comportamiento que se deseaba en los movimientos.



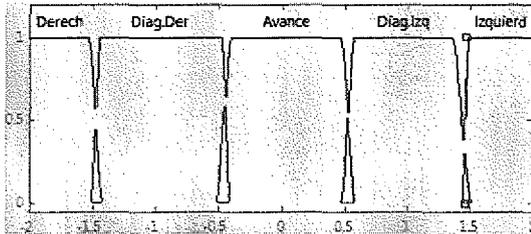
b)



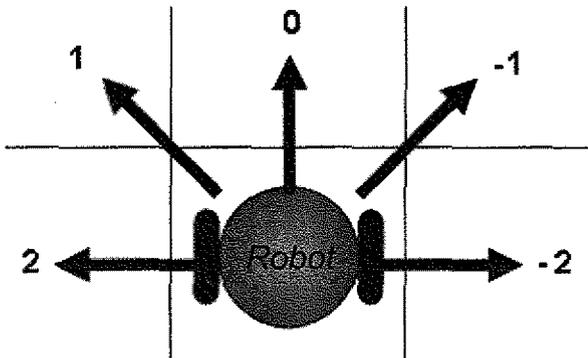
c)

Figura 7. Conjunto de entrada en el controlador Fuzzy. a) Distancia al Obstáculo b) Ángulo a Obstáculo c) Ángulo a Meta

Se implementaron 35 reglas, utilizando conceptos cualitativos simples, como tratar de seguir la línea imaginaria que va desde el punto de partida hasta la meta si no existe algún obstáculo al frente de IO, o en caso contrario, el robot debe bordearlo por el lado derecho. Por ejemplo si la entrada al controlador fuzzy es "Muy Cerca" & "D.Izquierda" & "LeveD" indica que el objeto se encuentra al lado izquierdo del robot y al lado opuesto es el camino a meta, entonces la salida que debe enviar el controlador es "Diagder". En el caso que la entrada sea "Muy Cerca" & "D.Derecha" & "LeveD" refleja que el obstáculo interfiere en la línea a meta, entonces la salida es "Avance".



a)



b)

Figura 8.

a) Conjunto de salida para el controlador Fuzzy
b) Movimientos de IO

Algunos resultados del comportamiento del controlador fuzzy se observan en la Figura 9. Una característica interesante es que el controlador genera algunos giros de 90 grados en la trayectoria, esto se debe a la forma como se programó el conjunto de salida, funciones trapezoidales con pendiente abrupta, lo cual no es recomendable a la hora de implementar la trayectoria en el móvil para que la ejecute.

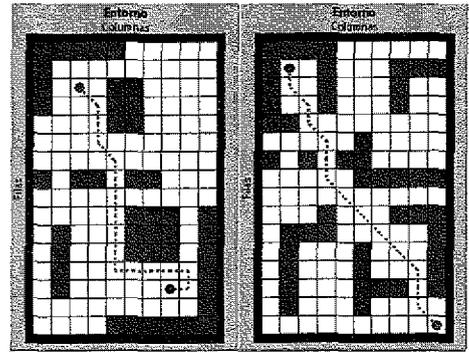


Figura 9. Resultados de la Simulación en entornos aleatorios - Lógica Difusa

6. RESULTADOS DE LA IMPLEMENTACIÓN

Los algoritmos se implementaron sobre dos plataformas de software, LabVIEW 8 y Matlab R2006a. Para establecer un porcentaje de efectividad, se trabajaron dos tipos de escenarios, uno de complejidad media (5-20 obstáculos) y otro de complejidad alta (21-40 obstáculos), generados aleatoriamente, definiendo a su vez distintos puntos de partida y llegada para cada uno. En cada nivel de complejidad se realizaron 20 pruebas, evaluando en cada una las tres técnicas de navegación: redes neuronales, algoritmos genéticos y lógica difusa.

En la figura 10 se observa como la efectividad de los algoritmos decae cuando su tarea se hace más compleja. En el caso de las redes neuronales, el porcentaje del 80%, para una complejidad media, decae al 70% en entornos con complejidad alta, lo que demuestra que el algoritmo tiende a no encontrar una solución para estas situaciones. Una alternativa sería incrementar los casos de entrenamiento, analizando a su vez el valor de los nodos radiales definidos en la red RBF, debido a que existen estimulaciones que están fuera de los centroides.

En la lógica difusa, la efectividad del algoritmo decae cuando su tarea se hace más compleja, lo cual demuestra que el algoritmo tiende a no encontrar una solución a entornos muy estructurados. Este algoritmo es restrictivo en el sentido que el robot tiene prioridad a seguir la línea recta imaginaria que se forma entre él y su objetivo, esto puede ser un problema si el camino pasa por un pasadizo

que resulta ser bloqueado por un obstáculo imprevisto. Una solución para incrementar la efectividad, requiere desarrollar algoritmos que generen un nuevo conjunto de reglas optimizando los resultados.

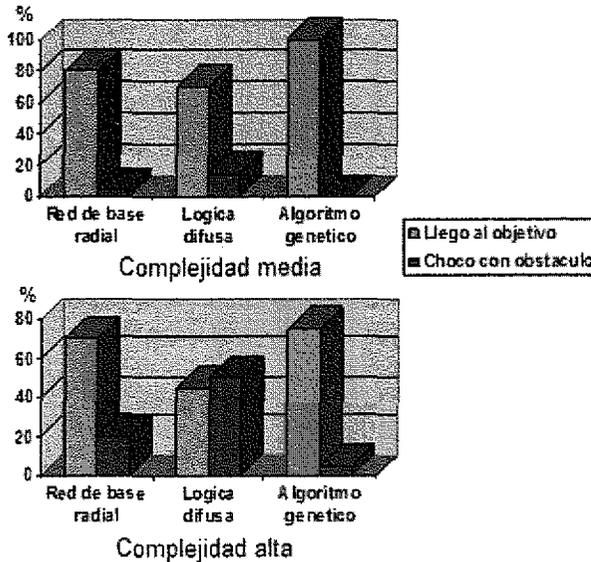


Figura 10. Grafico de comparación entre las tres técnicas de navegación

La técnica con la mejor efectividad es el algoritmo genético, aunque el costo computacional es mayor con relación a las otras dos. Cada tipo de escenario, de acuerdo a su nivel de complejidad requiere una cantidad proporcional de generaciones, y en algunos de los casos, en ambientes complejos, se debe reiniciar el algoritmo ya que tiende a no encontrar una respuesta después de muchas generaciones evaluadas. Sin embargo, una importante consideración al crearlo es definir una representación del problema. El lenguaje utilizado para especificar posibles soluciones debe ser robusto, capaz de tolerar cambios aleatorios que no produzcan constantemente errores fatales o resultados sin sentido.

Al ejecutar los movimientos ordenados por el algoritmo se pueden observar las limitaciones mecánicas que presentan la estructura, cambios en la dirección generan pequeños errores en trayectorias largas, ocasionando que el robot se desvíe de su objetivo y choque con obstáculos, esto se debe a la inercia del robot y pequeñas diferencias de respuesta de los motores. El algoritmo que menos errores presenta en su implementación física son los algoritmos genéticos, por que su función de aptitud premia los movimientos uniformes y continuos, caso contrario al de redes neuronales y lógica difusa.

7. MAPAS DE OCUPACIÓN Y AUTOLOCALIZACIÓN

En robótica móvil, un mapa de ocupación es una representación del entorno en forma de mosaico o rejilla compuesta de celdas, que almacenan información del entorno. Esta rejilla es generada por un robot en movimiento, que va actualizando de forma recursiva cada celda con el cambio de posición. Utilizando esta información se pueden programar tareas de localización o generación de trayectorias.

Con el fin de explorar su entorno, el robot IO realiza movimientos en línea recta. Si detecta algún obstáculo, y este bloquea la ruta programada, se debe generar una acción evasiva, que depende de la celda que se encuentre ocupada y de la orientación del robot, primando los movimientos que rodeen los obstáculos por una zona anteriormente explorada (Figura 11). Las rutinas de evasión no tienen en cuenta salir de callejones, debido a que el IO no cuenta con sensores posteriores, luego no se puede garantizar un movimiento sin saber lo que se encuentra en la celda a la que se desea que se mueva el robot. Una vez identificado el espacio de trabajo se define el tipo de técnica de navegación, que IO guiará hasta un punto significativo en el entorno.

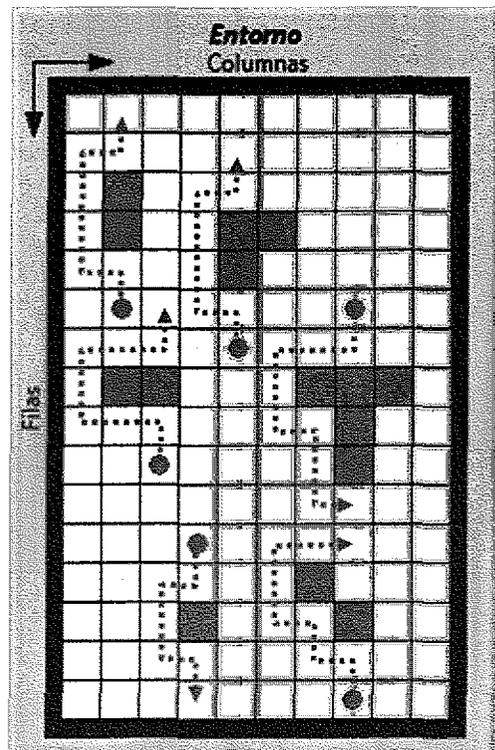


Figura 11. Principales rutinas de evasión de obstáculos implementadas en el software de control.

Una limitante al implementar este método de exploración se observa durante los recorridos extensos, debidos a errores mecánicos el robot comienza a desviarse de las celdas que está explorando, lo cual introduce un error de lectura en el sensor, porque los obstáculos se pueden salir del campo de visión. Una alternativa para mejorar los resultados es implementar técnicas de fusión sensorial, información de diferentes sensores sobre un mismo objeto del entorno darán una estimación del mismo más fiel reduciendo la incertidumbre en la medida ocasionada por el desvío en la ruta ideal del robot, logrando de esta forma un estimativo del error en el desplazamiento. Un ejemplo de esta técnica se observa en la referencia [9], la navegación del robot depende de la información de sensores ultrasónicos y ópticos, los cuales se caracterizan por producir una percepción imprecisa del entorno debido a problemas de ruido, por lo tanto se destinan diferentes agentes que procesen la información, reduciendo la complejidad del sistema e incrementando la fiabilidad del mismo.

8. CONCLUSIONES

Los algoritmos genéticos son adecuados para generar la trayectoria que debe seguir un robot móvil en un entorno estructurado, con respecto a las redes neuronales y la lógica difusa, cumpliendo con una ecuación de aptitud creada por la persona que implementa el algoritmo, aunque necesitan conocerlo a priori, por eso son utilizados en optimización de controladores, pero no se pueden emplear para explorar entornos desconocidos. Es importante resaltar la eficiencia de la técnica de redes neuronales, que fue mayor al 70%, lo cual la convierte en una alternativa que presenta buenos resultados con un menor costo computacional, con relación a los algoritmos genéticos, para esta clase de problemas.

El controlador difuso ofrece importantes ventajas en el guiado de un robot móvil al permitir la incorporación en el control de la lógica que usaría un ser humano para alcanzar su objetivo, no demandando cálculos muy elaborados para sintonizar los parámetros de las funciones de pertenencia, para lo cual solo basta conocer los parámetros máximos de tolerancia de los sensores y actuadores.

Las simulaciones realizadas en todo el desarrollo del trabajo no tienen en cuenta parámetros como las constantes de inercia del robot y las limitantes mecánicas. La implementación de algoritmos más robustos, teniendo en cuenta estos parámetros, pueden generar simuladores más cercanos a la realidad y mejorar la navegación del robot, buscando llevar el error de movimiento por deslizamiento e inercia a un valor muy cercano a cero.

Para que un robot esté dotado de la máxima versatilidad y pueda, tanto interactuar como adaptarse a su entorno, es necesario que disponga de los mecanismos necesarios para poder observar, reconocer e interpretar. La combinación de varios sensores de diferente clase permite dotar al robot de un grado de flexibilidad y autonomía mayor, lo cual se conoce como fusión sensorial, sin embargo, esto provoca una sobrecarga de información en el sistema, que debe ofrecer una respuesta correcta en unos márgenes de tiempo adecuados. Además son necesarias dos operaciones, por un lado combinar datos que en ocasiones pueden ser contradictorios, y por otro filtrarlos de forma que se elimine, tanta información redundante como datos erróneos.

9. BIBLIOGRAFÍA

- [1] GUSTAVO RAMIREZ. *Método de Aprendizaje Simple para Navegación de Minirobots Móviles Rodantes*. Escuela de Ingeniería Eléctrica y Mecánica. Facultad de Minas, Universidad Nacional de Colombia, Medellín. Enero de 2003.
- [2] MAC GABRIEL VEGA YANCE. *Concepción e Implementación de un Simulador de Trayectorias para Robots Autónomos Móviles*. Trabajo de grado, Ingeniería Electrónica, Universidad del Valle. 2006.
- [3] JULIO CESAR ROJAS GUALDRON. *Desarrollo de Estrategias para el Control de Móviles Autónomos Basado en Redes Neuronales*. Universidad Industrial de Santander. 2007.
- [4] RODRIGO CARRASCO, ALDO CIPRIANO. *Sistema de Guiado para un Robot Móvil Basado en Lógica Difusa*. Departamento de Ingeniería Eléctrica. Pontificia Universidad Católica de Chile.
- [5] M. BROX, A. GERSNOVIEZ, S. SÁNCHEZ SOLANO. *Controlador Difuso para Problemas de Navegación en Presencia de Obstáculos Fijos*. Instituto de Microelectrónica de Sevilla – Centro Nacional de Microelectrónica. 2006.
- [6] DOMINGO GALLARDO, OTTO COLOMINA, FRANCISCO FLÓREZ. *Generación de Trayectorias Robustas Mediante Computación Evolutiva*. Departamento de Tecnología Informática y Computación. Universidad de Alicante.
- [7] FABIO ALFONSO GONZÁLEZ, JAIME BARRERO PÉREZ. *Algoritmos Genéticos Aplicados al Planeamiento de Trayectorias de Robots Móviles. Modelado y Simulación*. Universidad Industrial de Santander.

[8] ALFREDO TORIZ PALACIOS, MARÍA AUXILIO OSORIO. *Planificación de Rutas para Robots Móviles Utilizando un Método PRM-Genético*. Facultad de Ciencias de la Computación. Universidad de Puebla.

[9] M.A. ZAMORA, L.M. TOMAS BALIBREA. *"Navegación Planificada de un Robot Móvil en Entornos Interiores Desconocidos"*. Universidad de Murcia.

[10] DIEGO ALEXANDER TIBADUIZA BURGOS. *Planeamiento de Trayectorias de un robot móvil*. Universidad Industrial de Santander. Enero de 2006.

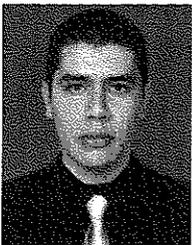
[11] MARCO CAVALCANTI. *Algoritmos Genéticos: Principios y Aplicaciones*. Rio de Janeiro. 2000. Universidad Pontificia de Rio de Janeiro. Departamento de Ingeniería Eléctrica.

10. AUTORES



HERNANDO GONZÁLEZ ACEVEDO
Candidato a Maestría en Ingeniería:
Área Electrónica

Universidad Industrial de Santander.
 Director del Grupo de Investigación
 en Automatización, Modelamiento,
 Simulación y Control de Productos y
 Procesos - Universidad Santo Tomas –
 Seccional Bucaramanga. Colombia.
 hernando_gonza@hotmail.com



CARLOS ALBERTO MEJÍA
CASTAÑEDA

Ingeniero Mecatrónico

Cordinador del Semillero de Robótica
 Móvil ASIMOV. Universidad Santo
 Tomas – Seccional Bucaramanga.
 Colombia.

ingcarlosmejia@hotmail.com