

Implementación de comportamientos para navegación inteligente de robots móviles

Implementation of Behaviors for Intelligent Navigation of Mobile Robot

MAXIMILIANO BUENO LÓPEZ

Ingeniero Electricista. Estudiante de la Maestría en Ingeniería Eléctrica (Universidad Tecnológica de Pereira). Profesor catedrático Universidad Tecnológica de Pereira. Miembro del grupo GIROPS.

max@ohm.utp.edu.co

LUIS HERNANDO RÍOS GONZÁLEZ

Ingeniero Electrónico MsC. Estudiante Doctorado UPV. Profesor titular (Universidad Tecnológica de Pereira). Miembro del grupo GIROPS.

lhgonza@utp.edu.co

Fecha de recepción: 13 de septiembre de 2007

Clasificación del artículo: investigación

Fecha de aceptación: 31 de enero de 2008

Palabras clave: PMETIN, robótica móvil, sistema basado en comportamientos y sistema de inferencia difuso.

Key words: behavior-based system, fuzzy inference systems, mobile robot, PMETIN.

RESUMEN

En este artículo se implementa un sistema de control para la navegación reactiva de un robot móvil en ambientes dinámicos, basado en comportamientos difusos. Un anillo de sensores ultrasónicos es utilizado para detectar los obstáculos del entorno por donde se desplaza el robot. El controlador difuso se compone de cuatro comportamientos básicos y su respectivo conjunto de reglas, dichos comportamientos son: navegar por pasillo, seguir pared, alcanzar objetivo y evitar obstáculos. Los algoritmos implementados para cada compor-

tamiento utilizan funciones de pertenencia del tipo triangular. El control desarrollado se simuló e implementó en la plataforma móvil P-METIN, construida por el grupo GIROPS. Los resultados obtenidos son satisfactorios para navegación en ambientes dinámicos.

ABSTRACT

In this paper is showed the development of fuzzy logic control system for reactive navigation of mobile robot in dynamic environments. A ring of ultrasonic sensors is employed to sense the obstacles

of environment. A fuzzy control system consists of four behaviors: Navigation by a corridor, follow a wall, reach an objective and Obstacle avoidance.

Each one of behaviors designed has triangular membership function. The control was probe in P-METIN platform, designed for the GIROPS group. Experimental results verify this approach.

* * *

1. Introducción

Son múltiples las aplicaciones de los robots móviles, tanto en el campo industrial como en el de experimentación e investigación. En lo que respecta a la navegación de robots móviles, es necesario tener en cuenta que en condiciones iniciales no se conoce la localización del robot en el entorno de trabajo; además, el conocimiento a priori del entorno de trabajo es incompleto, incierto y aproximado. La información adquirida por el robot, en la mayoría de los casos es poco fiable; los entornos reales son complejos y de dinámica impredecible. Para la navegación autónoma en tiempo real, el robot debe ser capaz de interpretar la información para obtener el conocimiento de su posición en el ambiente y planificar su ruta de desplazamiento en tiempo real [1].

Uno de los principales retos de la robótica móvil es la elaboración de sofisticadas estrategias de control, que le permitan al robot realizar tareas complejas a pesar de la incertidumbre del ambiente. Estas tareas complejas son más fáciles de llevar a cabo cuando son divididas en pequeñas estrategias o comportamientos [2 y 3]. Cada uno de estos comportamientos se diseña considerando las lecturas de los sensores, en este caso las lecturas de los sensores del anillo ultrasónico [1]. En este artículo se muestran los resultados de la implementación de algunos comportamientos básicos de navegación, utilizando lógica difusa, lo cual le permitirá al robot navegar por un ambiente de trabajo, así como evitar obstáculos y alcanzar objetivos definidos previamente. El uso de inteligencia artificial para la implementación de comportamientos es altamente recomendado, debido a la naturaleza de la información que se obtiene del ambiente. [4, 5 y 6].

Los sistemas difusos tienen la capacidad de tratar la incertidumbre e imprecisión de la información a través de la utilización de un conjunto de reglas del tipo IF-THEN y el conocimiento de expertos sobre el tema en estudio [7].

Por último, la lógica difusa permite diseñar cada uno de los comportamientos deseados, usando como entradas del sistema, las lecturas obtenidas de los sensores de ultrasonido, y como salidas la velocidad y dirección del robot [8].

2. Descripción de p-metin, plataforma móvil empleada

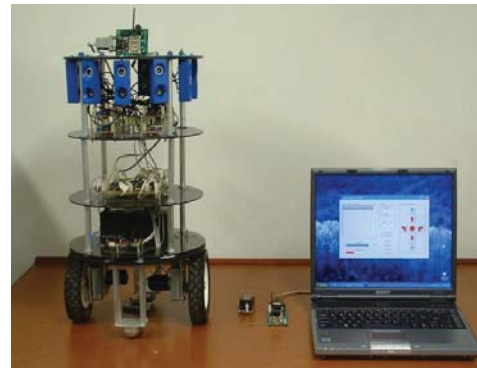


Figura 1. Plataforma móvil PMETIN.

La plataforma móvil mostrada en la figura 1 está dotada de un módulo de sensores de ultrasonido, un sistema central de control, un sistema de alimentación, un sistema de regulación y filtrado, un sistema controlador de motores y encoders de rotación.

El sistema de control se encarga de manejar el sistema de locomoción; activar el módulo de ultrasonido; monitorear el estado de la batería y establecer comunicación con el PC [9]. La arquitectura de control de

la plataforma móvil construida muestra una estructura multiprocesador (múltiples microcontroladores de la familia MOTOROLA, MC68HC908GP32, MC68HC908JK1 y el MC68HC908JK3), en la que cada procesador realiza una serie de tareas específicas y algunas tareas las realizan de manera compartida.

En los actuales momentos, con dicha plataforma (P-METIN), se realizan pruebas de navegación empleando técnicas inteligentes (lógica difusa y sistemas neurodifusos). Para facilitar en el procesamiento de datos, los sensores han sido divididos en grupos, los cuales son mostrados en la figura 2.

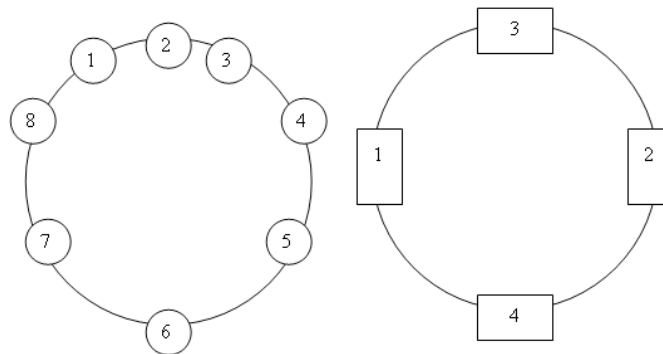


Figura 2. Distribución de los sensores ultrasónicos en la plataforma móvil.

3. Lógica difusa aplicada en robótica móvil

La lógica difusa se utiliza cuando la complejidad del proceso que se va a estudiar es muy alta o no existen modelos matemáticos precisos (procesos altamente no lineales), o cuando se utilizan definiciones o conocimiento impreciso o subjetivo. La lógica difusa puede controlar procesos por medio de reglas de sentido común [8 y 10]. El uso de la lógica difusa se hace cada vez más extensivo en el campo del control de agentes móviles. La capacidad de tratar con información imprecisa la convierte en una herramienta muy adecuada en el diseño de aplicaciones de control.

4. Diseño de comportamientos (t2)

La arquitectura Saphira diseñada por Saffiotti, Ruspini y Konolige [11 y 12] se basa en la descomposición de las tareas complejas en comportamientos. Estos comportamientos están codificados usando una base de reglas difusas del tipo *if Antecedente then Consecuente*. Tanto el antecedente como el

consecuente son conjuntos difusos. El valor del antecedente determina la “deseabilidad” de aplicar esa regla en el estado actual. En este caso se han definido cuatro comportamientos básicos (aproximación a la pared más cercana, seguir la pared, evitar obstáculos y alcanzar objetivo). Cada comportamiento se ha implementado como una base de reglas difusas; la salida de cada comportamiento es un conjunto difuso que representa la velocidad y ángulo de giro del robot, y los comportamientos se encargan del control del movimiento del robot.

4.1 Comportamiento alcanzar objetivo

Para este comportamiento se parte de unas coordenadas iniciales y unas coordenadas finales a las que se desea llegar. Se determina inicialmente la distancia mínima al objetivo, y el ángulo de giro que debe realizar el robot para iniciar su recorrido hacia las coordenadas finales. En este comportamiento no se consideran obstáculos; la figura 3 muestra gráficamente cuál es el objetivo de este comportamiento.

Los grupos denominados como 1, 2 y 4 no se consideran, sólo el grupo 3 (frente) tiene peso en este diseño. Las variables de entrada al controlador difuso son la distancia al objetivo y la lectura del grupo 3. La salida del sistema es la velocidad del robot, que puede ser alta (0,45 m/s) o baja (0,40 m/s). Las funciones de pertenencia empleadas para las variables de entrada se muestran en la figura 5.

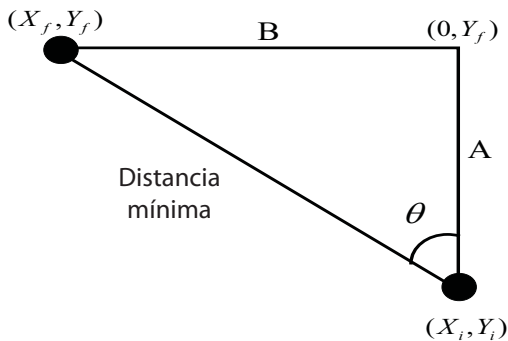
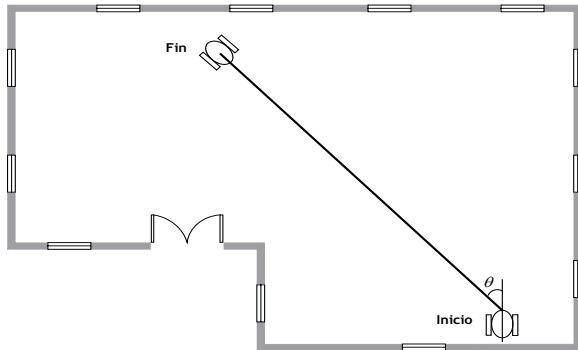


Figura 3. Llegar a objetivo.

$$\begin{aligned}
 A &= Y_f - Y_i \\
 B &= X_f - 0 \\
 \text{Distancia mínima} &= \sqrt{A^2 + B^2} \\
 \theta &= \tan^{-1}\left(\frac{B}{A}\right)
 \end{aligned}
 \tag{1}$$

Figura 4. Cálculo del ángulo θ .

En la figura 4 se presenta la construcción geométrica para calcular el ángulo θ . Se toman las coordenadas iniciales como punto de referencia (0,0) y se calcula el valor de los catetos A y B que se presentan en la figura 4.

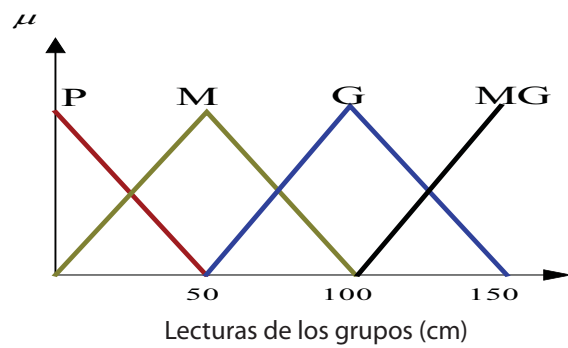


Figura 5. Funciones de pertenencia para las variables de entrada.

Las etiquetas lingüísticas tienen el siguiente significado:

- MP: distancia muy pequeña.
- P: distancia pequeña.
- N: distancia normal.
- ME: distancia media.
- G: distancia grande.
- MG: distancia muy grande.

La salida como se mencionó anteriormente sólo tiene dos posibles valores, debido a esto se emplean valores constantes, buscando que la velocidad resultante siempre corresponda a uno de estos dos valores.

En la figura 6 se muestra la interfaz diseñada para realizar las simulaciones de este comportamiento. Los datos de entrada a la interfaz son las coordenadas iniciales y finales, las lecturas de los sensores y la distancia de avance. El algoritmo implementado entrega la velocidad del robot y evalúa si es posible o no llegar al objetivo deseado. Las razones por las que el robot no puede cumplir con la tarea propuesta son:

- Lectura del grupo 3 es menor que la distancia mínima al objetivo.
- Si se encuentra un obstáculo en el camino que no le permite al robot avanzar.

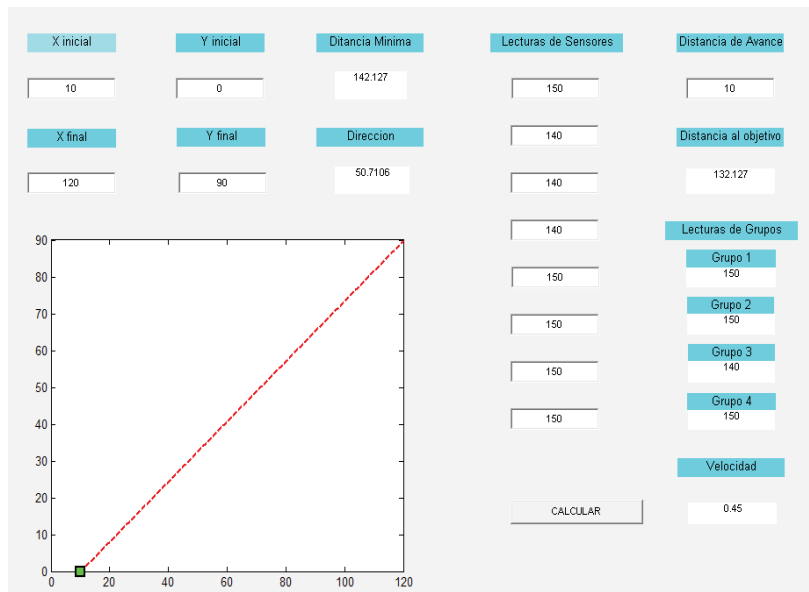


Figura 6. Interfaz para comportamiento alcanzar objetivo.

La distancia de avance hace referencia al desplazamiento que tiene el robot entre cada toma de datos, es necesario definirla para tener completo conoci-

miento sobre la distancia recorrida por el robot. En la tabla 1 se presentan las reglas planteadas para este comportamiento.

	Distancia al objetivo					
Grupo 3	MP	P	N	ME	G	MG
MP	cero	cero	cero	cero	cero	cero
P	cero	baja	baja	baja	baja	baja
N	cero	baja	alta	alta	alta	alta
ME	cero	baja	alta	alta	alta	alta
G	cero	baja	alta	alta	alta	alta
MG	cero	baja	alta	alta	alta	alta

Tabla 1. Reglas para comportamiento alcanzar objetivo.

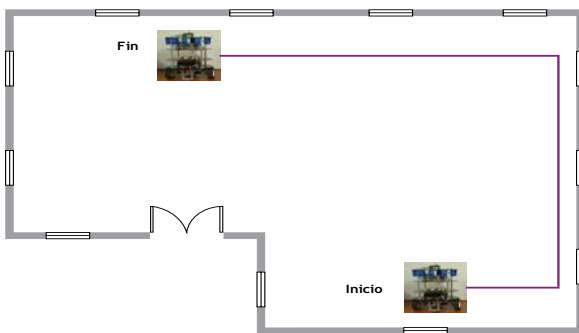


Figura 7. Seguir pared.

4.2 Comportamiento seguir pared

En este comportamiento el robot tiene como referencia en todo momento para su desplazamiento la pared más cercana. La figura 7 muestra gráficamente cuál es el objetivo de este comportamiento.

Las entradas al sistema en este comportamiento son las lecturas de los sensores y las salidas son la velocidad y dirección del robot. Se han implementado dos algoritmos empleando diferentes números de funciones de pertenencia, a fin de comprobar

la disminución en el tiempo computacional. En la figura 8 se muestran las funciones de pertenencia empleadas.

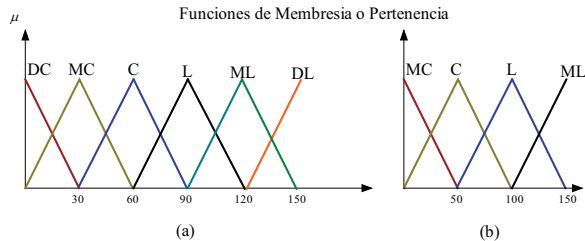


Figura 8. Funciones de pertenencia para las variables de entrada.

En la figura 8 (a) se presentan un conjunto de funciones de pertenencia que le dan al sistema una buena exactitud, pero el tiempo computacional es muy elevado, mientras en la figura 8 (b) se reduce

la exactitud y el tiempo computacional es menor. Las etiquetas lingüísticas tienen el siguiente significado:

- DC: demasiado cerca.
- MC: muy cerca.
- C: cerca.
- L: lejos.
- ML: muy lejos.
- DL: demasiado lejos.

En la figura 9 se presenta un diagrama de flujo para este comportamiento. En la figura 10 se presentan dos graficas de entornos simulados para el comportamiento seguir pared.

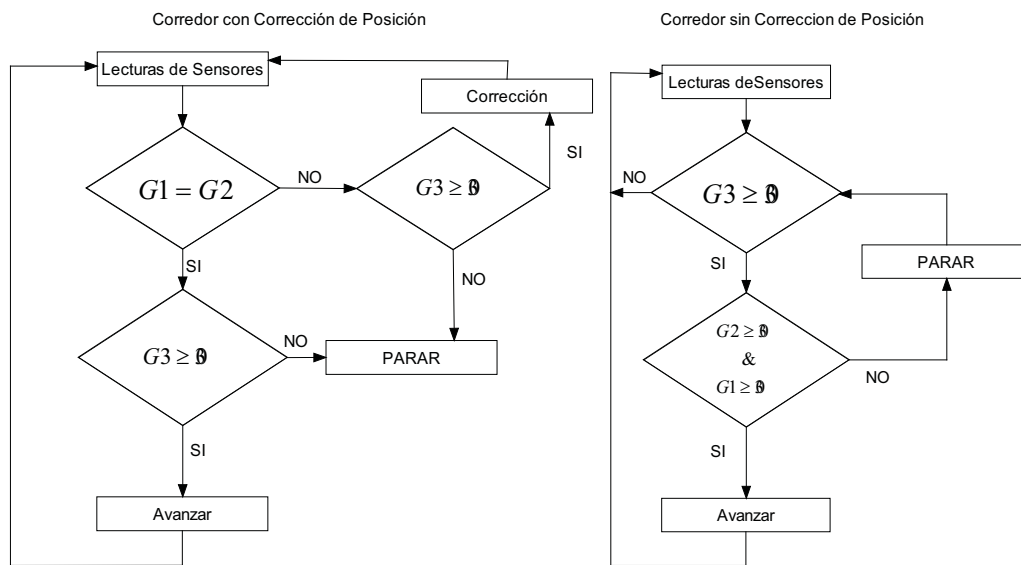


Figura 9. Diagrama de flujo para comportamiento seguir pared.

En la tabla 2 se presentan algunos resultados obtenidos mediante simulaciones empleando el grupo de funciones de pertenencia de la figura 8(a).

4.3 Comportamiento navegar a través de un pasillo

Para este comportamiento se han diseñado dos casos; en el primero de ellos el robot se desplaza

hasta que detecta un obstáculo frente a él. En el segundo caso el robot realiza corrección de su posición, buscando ir en línea recta a través del pasillo. En la figura 10 se muestran algunas simulaciones realizadas y en la figura 11 se muestra gráficamente el objetivo de este comportamiento.

Pared						
Grupo de funciones de pertenencia						
Grupo 1	Grupo 2	Grupo 3	Grupo 4	Velocidad(m/s)	Desviación angular	Tiempo de ejecución(s)
37,0	73,0	52,3	51,0	0,00	81,4	0,078
80,5	56,5	23,0	32,0	0,00	54,9	0,000
90,0	25,5	82,3	150,0	0,00	45,0	0,015
46,0	90,5	59,3	98,0	0,00	68,8	0,016
43,5	61,5	90,0	63,0	0,00	107,5	0,000
84,0	30,0	125,6	37,0	0,00	45,0	0,000
13,0	98,5	41,0	41,0	0,00	45,0	0,000
55,0	58,5	24,6	124,0	0,40	0,0	0,015
130,0	76,5	75,6	61,0	0,00	44,6	0,016
106,0	42,0	65,0	97,0	0,00	45,0	0,000
100,0	77,0	85,6	93,0	0,00	54,9	0,015

Tabla 2. Resultados obtenidos para comportamiento seguir pared.

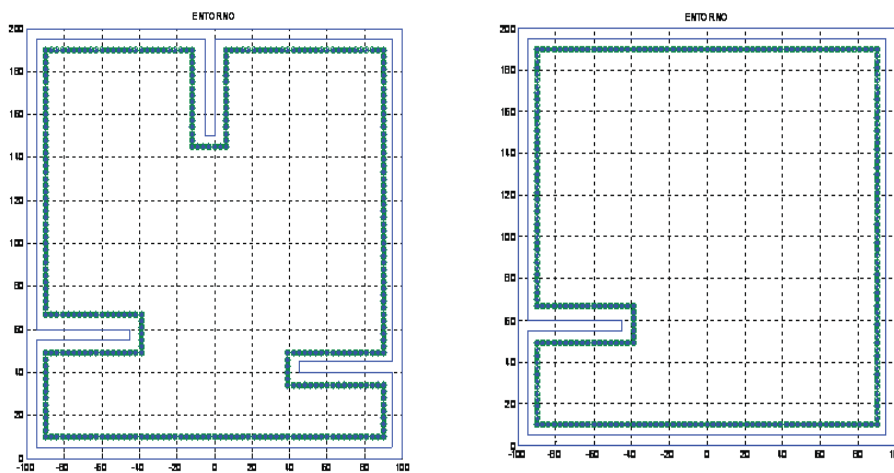


Figura 10. Simulaciones realizadas para comportamiento seguir pared.

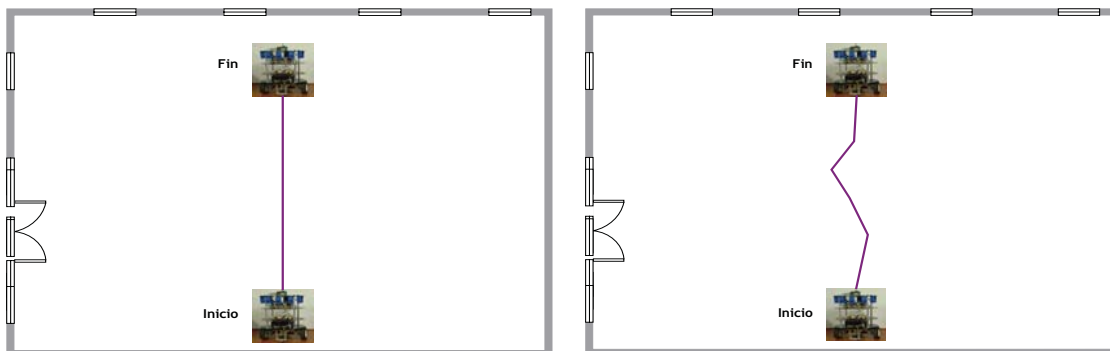


Figura 11. Navegar a través de un pasillo.

Las entradas al sistema en este comportamiento son las lecturas de los sensores y las salidas son la velocidad y dirección del robot. Las funciones de pertenencia empleadas son iguales a las del comportamiento seguir pared. En la figura 12 se presenta la interfaz diseñada para este comportamiento y en la figura 13 los diagramas de flujo correspondientes.

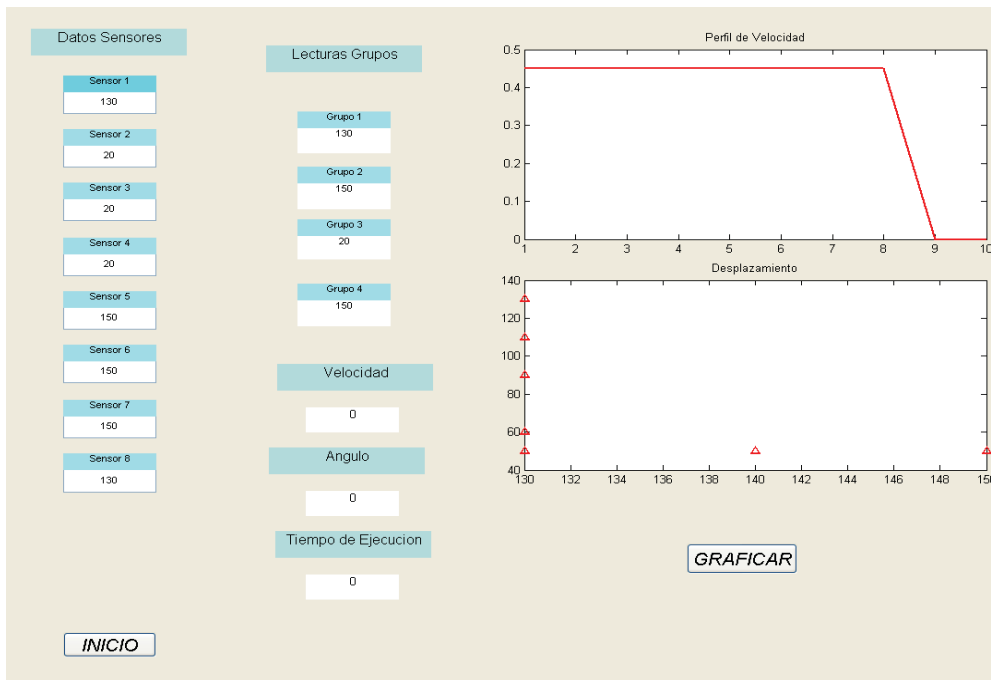


Figura 12. Navegar a través de un pasillo con corrección de posición.

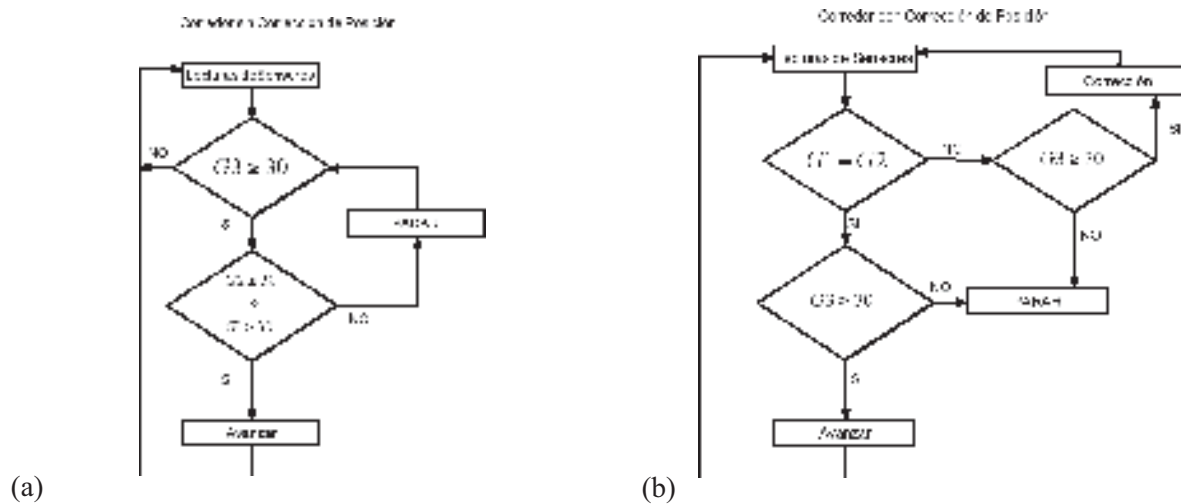


Figura 13. Navegar a través de un pasillo con corrección de posición.

En la tabla 3 se presentan los resultados obtenidos mediante simulación para la navegación por un corredor sin corrección de posición empleando las funciones de pertenencia de la figura 8(b), y en la tabla 4 se presentan los resultados para el caso con corrección de posición.

Pasillo						
Sin corrección de posición/grupo óptimo de funciones de pertenencia						
Grupo 1	Grupo 2	Grupo 3	Grupo 4	Velocidad(m/s)	Desviación angular	Tiempo de ejecución(s)
37,0	73,0	52,3	51,0	0,40	0,0	0,046
80,5	56,5	23,0	32,0	0,00	0,0	0,000
90,0	25,5	82,3	150,0	0,40	0,0	0,000
46,0	90,5	59,3	98,0	0,40	0,0	0,000
43,5	61,5	90,0	63,0	0,40	0,0	0,000
84,0	30,0	125,6	37,0	0,40	0,0	0,000
13,0	98,5	41,0	41,0	0,00	0,0	0,000
55,0	58,5	24,6	124,0	0,00	0,0	0,000
130,0	76,5	75,6	61,0	0,45	0,0	0,000
106,0	42,0	65,0	97,0	0,45	0,0	0,000
100,0	77,0	85,6	93,0	0,45	0,0	0,000
79,5	65,0	86,0	78,0	0,40	0,0	0,000
20,0	111,5	105,0	74,0	0,40	0,0	0,000
18,0	83,0	32,3	139,0	0,00	0,0	0,000
86,0	114,0	55,0	87,0	0,45	0,0	0,000
118,0	82,5	42,6	19,0	0,00	0,0	0,000

Tabla 3. Navegar por pasillo sin corrección de posición.

Corredor						
Con corrección de posición/grupo óptimo de funciones de pertenencia						
Grupo 1	Grupo 2	Grupo 3	Grupo 4	Velocidad(m/s)	Desviación angular	Tiempo de ejecución(s)
37,0	73,0	52,3	51,0	0,00	81,4	0,078
80,5	56,5	23,0	32,0	0,00	54,9	0,000
90,0	25,5	82,3	150,0	0,00	45,0	0,015
46,0	90,5	59,3	98,0	0,00	68,8	0,016
43,5	61,5	90,0	63,0	0,00	131,5	0,000
84,0	30,0	125,6	37,0	0,00	45,0	0,000
13,0	98,5	41,0	41,0	0,00	45,0	0,000
55,0	58,5	24,6	124,0	0,00	131,4	0,015
130,0	76,5	75,6	61,0	0,00	44,6	0,016
106,0	42,0	65,0	97,0	0,00	45,0	0,000
100,0	77,0	85,6	93,0	0,00	53,1	0,015
79,5	65,0	86,0	78,0	0,45	0,0	0,047
20,0	111,5	105,0	74,0	0,00	45,0	0,000
18,0	83,0	32,3	139,0	0,00	87,1	0,016
86,0	114,0	55,0	87,0	0,00	92,9	0,015
118,0	82,5	42,6	19,0	0,00	27,4	0,016
108,0	110,5	119,0	139,0	0,45	0,0	0,032
28,0	30,0	70,3	59,0	0,00	134,5	0,000

Tabla 4. Navegar por pasillo sin corrección de posición.

4.4 Comportamiento alcanzar objetivo evitando obstáculo

En este comportamiento el objetivo es llegar hasta un punto que es definido por un par de coordenadas (x,y), evitando los obstáculos que aparezcan en la trayectoria. La figura 14 muestra gráficamente cuál es el objetivo de este caso.

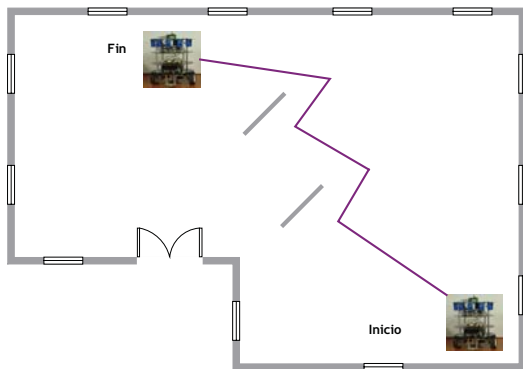


Figura 14. Llegar a objetivo evitando obstáculo.

Para iniciar el desplazamiento se verifica la lectura de cada uno de los grupo; éstos deben entregar una lectura superior a 30 cm, si esta condición no se cumple no se inicia el recorrido.

Se definen cuatro (4) entradas para el sistema de control difuso, las cuales son:

- Lectura de sensores que conforman el grupo 1 (izquierda);
- Lectura de sensores que conforman el grupo 2 (derecha);
- Lectura de sensores que conforman el grupo 3 (frente);
- Distancia al objetivo.

El sistema posee dos salidas las cuales son:

- Velocidad;
- Dirección del robot.

Para las entradas correspondientes a las lecturas de los sensores, el universo de discurso está en el rango [0 150]. Para la entrada correspondiente a la distancia al objetivo, el universo de discurso esta en el rango [0 30]. Para las cuatro entradas se emplean las siguientes etiquetas lingüísticas.

P: distancia pequeña (conjunto 1).

M: distancia media (conjunto 2).

G: distancia grande (conjunto 3).

MG: distancia muy grande (conjunto 4).

En la figura 15 se presentan las formas empleadas para las funciones de membresía o pertenencia.

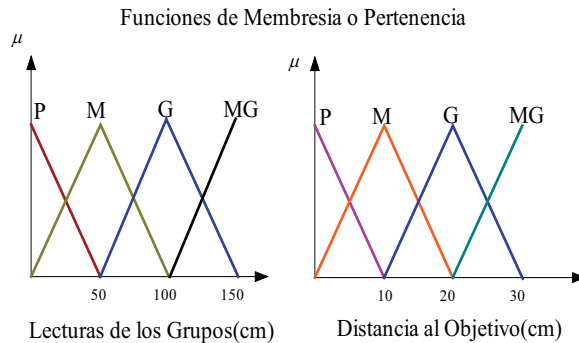


Figura 15. Funciones de Pertenencia para las entradas del sistema.

Además de las variables de entrada se debe conocer en todo momento la distancia que ha recorrido el robot, para lograr esto se define al inicio del recorrido una distancia de avance, la cual determinara el intervalo en el que se tomaran medidas por parte de los sensores. La distancia de avance debe ser pequeña para evitar errores de tipo odométrico. El robot inicia el desplazamiento hacia el objetivo en línea recta, hasta que detecta un obstáculo a una distancia no mayor a los 30 cm.

Como se observa en la figura 16, se asigna una letra a las rectas que tienen pendiente positiva, y otra a las que poseen pendiente negativa. Esto ayudará al proceso de difuminado. Para el caso de la Entrada d1 las letras empleadas son: i, j. En la figura 15 los números asignados a cada función de pertenencia tienen el siguiente significado:

$$1 \boxtimes P, 2 \boxtimes M, 3 \boxtimes G, 4 \boxtimes MG$$

Se observa que cada valor de entrada activará dos variables difusas, cada una con un valor de pertenencia, “muei” y “muej” respectivamente para el caso mostrado en la figura.

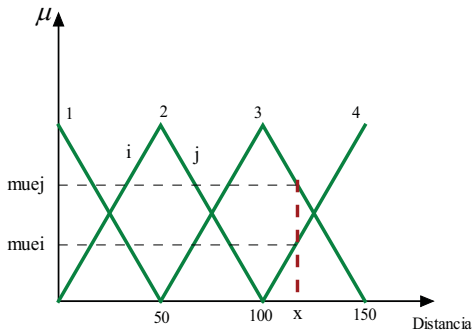


Figura 16. Proceso de difuminado.

Cada función de pertenencia estará etiquetada por dos letras que corresponden a cada una de sus rectas, además a cada letra se le asigna el número correspondiente al conjunto difuso que representa, por ejemplo, el conjunto difuso Distancia Media (M) al cual se le asigna el número 2 se representa por una función de pertenencia identificada por $i=2$ y $j=2$.

De los mecanismos de inferencia existentes se escogió el mecanismo de inferencia tipo Mamdani,

el cual es representado en la figura 17. Debido a la forma triangular de las funciones de pertenencia y al traslape del 50% entre ellas, para cada valor de entrada, se activan cuatro reglas, también llamadas metareglas, las cuales forman parte del cálculo de la salida. Los valores de los niveles de disparo de cada una de estas reglas se obtienen como:

$$ndikmr = \min(Mud1i, Mud1k, Mud1m, Mud1r) \quad (4)$$

La base de reglas difusas derivada para esta aplicación se obtuvo mediante la aplicación de un método heurístico. Después de analizar el comportamiento deseado se dedujo la relación (de tipo *if-then*) que debe existir entre las entradas al controlador difuso ($d1, d2, d3$ y $d4$). Estas relaciones se resumen en la tabla 5.

Mientras no se detecta obstáculo, la única entrada que se considera es la que corresponde al grupo 3. Cuando es detectado un obstáculo, se consideran los sensores ubicados en la parte lateral del robot 1 y 2. El ángulo de giro será $+90$ grados (izquierda) o -90 grados (derecha).

Grupo 2 \ Grupo 1	1	2	3	4
1	Parar	-90	-90	-90
2	90	-90	-90	-90
3	90	90	-90	-90
4	90	90	90	-90

Tabla 5. Conjunto de reglas para comportamiento alcanzar objetivo evadiendo obstáculo.

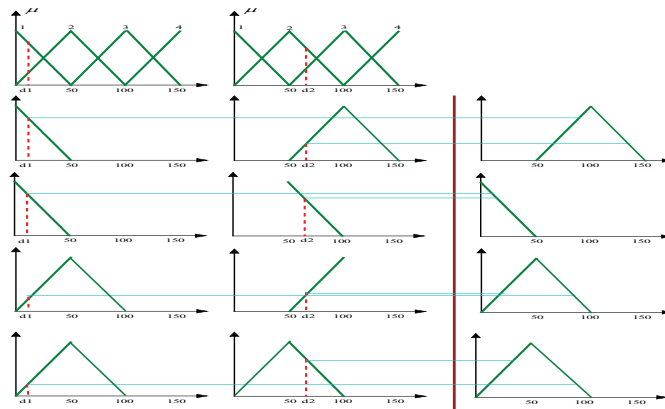


Figura 17. Mecanismo de inferencia difusa.

En la figura 18 se muestra un entorno de simulación para este comportamiento y la trayectoria generada.

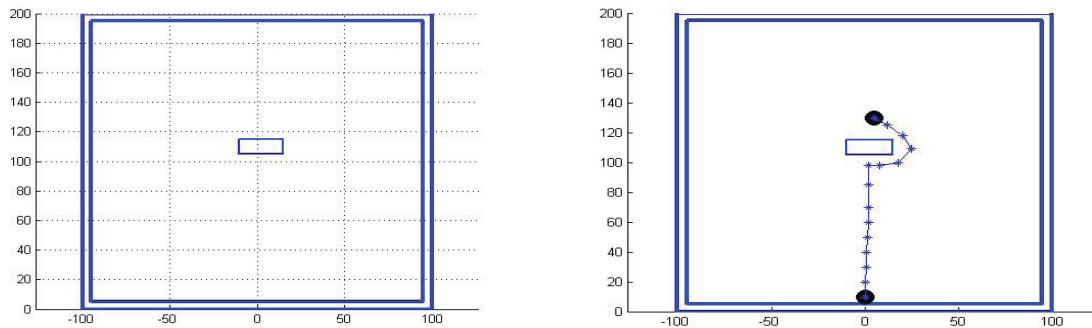


Figura 18. Entorno de simulación para comportamiento alcanzar objetivo evadiendo obstáculos.

5. Pruebas realizadas en la plataforma móvil (pmetin)

Si se emplea la plataforma móvil PMETIN se realizó un conjunto de pruebas a los diversos comportamientos. Las figuras muestran algunas de las trayectorias obtenidas para los comportamientos seguir pared y navegar a través de un pasillo. En la figura 19 se observa que la plataforma móvil se desplaza cerca a la pared izquierda y su despla-

zamiento no es en línea recta debido a los errores odométricos. En la figura 20 se observa una de las trayectorias generadas por el robot para el comportamiento navegar a través de un pasillo sin corrección de posición. En la figura 21 se observa una de las trayectorias generadas por la plataforma para el comportamiento navegar a través de un pasillo corrigiendo posición, este comportamiento busca que la plataforma se desplace por el centro del pasillo en el que se encuentra.

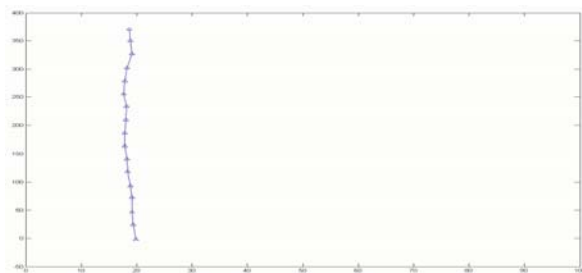


Figura 19. Comportamiento seguir pared.

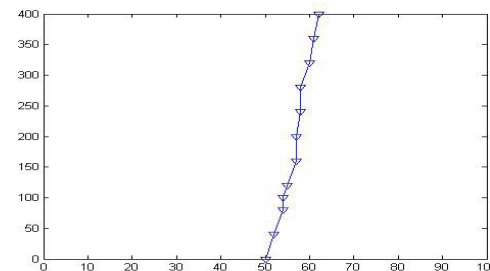


Figura 20. Comportamiento navegar a través de pasillo sin corrección de posición.

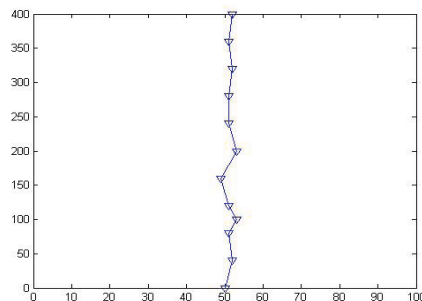


Figura 21. Comportamiento navegar a través de pasillo con corrección de posición.

6. Conclusiones

Los resultados obtenidos muestran el buen comportamiento del sistema difuso presentado para el control de navegación de un robot móvil. La estrategia empleando lógica difusa permite fácilmente trabajar con sistemas de múltiples variables, los cuales generalmente son difícil de modelar empleando estrategias de control clásico. En el grupo GIROPS se han realizado trabajos utilizando controladores clásicos y el gran problema que se presenta es el modelamiento matemático del robot, debido a la cantidad de elementos que se ven involucrados. El tiempo de ejecución para un controlador clásico es menor comparado con el controlador difuso: éste es uno de los mayores problemas de los controladores difusos, y se suele denominar “maldición de la dimensionalidad”, e indica que a mayor cantidad de funciones de pertenencia (lo cual origina un error pequeño), el tiempo de ejecución es mayor, debido a

la cantidad de reglas que se deben plantear. Esto se ha demostrado en las tablas presentadas en la sección de resultados. Además se han implementado estrategias de navegación utilizando sistemas neurodifusos y FCM (Fuzzy cluster mean); obtuvieron mejores resultados en cuanto a tiempo de ejecución. Por último, la descomposición de un problema en múltiples tareas permite al robot mostrar un alto grado de inteligencia.

Agradecimientos

Este trabajo fue desarrollado en el marco del proyecto implementación de nuevas metodologías que involucren técnicas de inteligencia artificial para navegación de robots móviles en ambientes dinámicos financiado por la Universidad Tecnológica de Pereira. Los autores agradecen la colaboración del grupo de Sensorización y Robótica Industrial de la Universidad Politécnica de Valencia dirigido por el Doctor Ginés Benet Gilabert.

Referencias bibliográficas

- | | |
|--|---|
| <p>[1] Araújo, Rui y Almeida, Aníbal. (April, 1999) Learning Sensor-Based Navigation of a Real Mobile Robot in Unknown Worlds. <i>IEEE Transactions on Systems, Man, and Cybernetics—Part b: Cybernetics</i>, 29, (2).</p> <p>[2] Bueno, Maximiliano; Ríos, Luis Hernando y Parra, Hernando. (2007) Navegación de robots móviles mediante comportamientos utilizando lógica difusa. <i>Scientia Et Técnica, Año XIII</i> (34), 79-84.</p> <p>[3] Maaref, H. and Barret, C. (2002) Sensor-Based Navigation of a Mobile Robot in an Indoor Environment. <i>Elsevier Robotics and Autonomous Systems</i>, 38, 1-18.</p> <p>[4] Akbarzadeh-T.; Kumbala, K.; Tunstel E. and Jamshidi M. (2000) Soft Computing for Autonomous Robotic Systems. <i>Computers and Electrical Engineering, Vol. 26</i>, 5-32.</p> <p>[5] Ríos, Luis H.; Bueno, Maximiliano and Restrepo Carlos. (Agosto de 2007) Behaviors design for a mobile robot using fuzzy logic. En 23rd ISPE Internacional Conference on CAD/CAM, Robotics & Factories of the future. CARS & FOF 2007. Bogotá D.C. Colombia .</p> <p>[6] Abdessamed, Foudil.; Benmahammed, Khier and Monacelli, Eric. (2004) A Fuzzy-Based Reactive Controller for a Non-Holonomic Mobile Robot. <i>Elsevier: Robotics and Autonomous Systems</i>, 47.</p> | <p>[7] Vadakkepat, Prahlad; Peng, Xiao; Kiat Quek, Boon and Tong Heng, Lee. (2007) Evolution of Fuzzy Behaviors for Multi-Robotic System. <i>Elsevier Robotics and Autonomous Systems</i>, 55, 146-161.</p> <p>[8] Saffiotti, Alessandro. (1998) <i>Autonomous Robot Navigation: A Fuzzy Logic Approach</i>. Tesis Doctoral. Université Libre de Bruxelles. Faculté de Sciences Appliquées.</p> <p>[9] Ríos, Luis Hernando; Córdoba, Juan Pablo y Restrepo, Carlos. (2005) Descripción de una plataforma móvil implementada para el estudio de técnicas inteligentes de navegación :PMETIN. <i>Scientia Et Técnica, Año XI</i> (27), 37-42.</p> <p>[10] Mohammad Al-Khatib and Jean J., Saade. (2003) An Efficient Data-Driven Fuzzy Approach to the Motion Planning Problem of a Mobile Robot. <i>Elsevier Fuzzy Sets and Systems and Autonomous Systems</i>, 134, 65-82.</p> <p>[11] Konolige, K. and Myers, K. (1998) The Saphira architecture for autonomous mobile robots. en: <i>Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems</i> (chapter 9). The MIT Press, pp. 211-242.</p> <p>[12] Saffiotti, A.; Konolige, K. and Ruspini, E. (1995) A Multivalued Logic Approach to Integrating Planning and Control. <i>Artificial Intelligence</i>, 76, 481-526.</p> |
|--|---|