

Navegación de un robot móvil autónomo utilizando el concepto de campo de fuerza combinado

Navigation of an autonomous mobile robot using the combined force field concept

JOSÉ L. SUSA¹ y DIEGO A. RAMOS²

1. Ingeniero electrónico, investigador grupo de investigación Ecitrónica, Escuela Colombiana de Ingeniería, Bogotá, Colombia. jose.susa@escuelaing.edu.co.

2. Ingeniero electrónico e ingeniero mecánico, grupo de investigación Ecitrónica, Escuela Colombiana de Ingeniería, Bogotá, Colombia. diego.ramos@escuelaing.edu.co.

Recibido: 29/06/2009 Aceptado: 12/12/2009

Disponible en <http://www.escuelaing.edu.co/revista.htm>

Resumen

En este artículo se presentan el desarrollo, prueba y resultados obtenidos de un algoritmo de evasión de obstáculos basado en el método de campo de potencial (PFM, por su sigla en inglés) y combinado con el método de seguimiento de contornos, para resolver el problema del mínimo local que posee el PFM.

Adicionalmente, se divulgan los resultados de la investigación, cuyo propósito fue desarrollar un algoritmo que permitiera a un robot móvil desplazarse en forma autónoma, con el fin de alcanzar una meta, evitando los obstáculos que encontrara en su trayectoria. Los requerimientos para el diseño del algoritmo fueron alta velocidad de respuesta, bajo consumo de recursos de *hardware* y capacidad de respuesta ante situaciones no previstas.

Las simulaciones demuestran que el algoritmo soluciona el problema del mínimo local, inherente al uso del PFM, y puede implementarse en un robot real, cumpliendo con las características citadas previamente.

Palabras claves: inteligencia artificial, evasión de obstáculos, método de campo de potencial y navegación autónoma.

Abstract

This article presents the development, testing and results from an obstacle avoidance algorithm based on potential field method (PFM), combined with the contour following method to solve the problem of local minimum in the PFM.

In this paper, we report the findings of the research, whose objective was to develop an algorithm that allowed a mobile robot to navigate autonomously to reach a goal, avoiding obstacles in its path. Requirements for the design of the algorithm were high response speed, low consumption of hardware resources and capacity answer to unforeseen situations.

The simulations show that the algorithm solves the problem of local minimum-inherent in the use of PFM-and it can be implemented in a real robot, since it fulfills the mentioned characteristics.

Keywords: artificial intelligence, obstacle avoidance, potential field method, autonomous navigation.

INTRODUCCIÓN

Ir de un punto a otro (*path planning*), ir en busca de alimento (*research goal*) y caminar sin tropezar (*avoiding obstacles*) son comportamientos ejecutados fácil y rápidamente por un ser vivo; sin embargo, hasta el día de hoy los intentos por imitar tales funciones en agentes robóticos autónomos han resultado en grandes desafíos para los investigadores, ya que para lograr una navegación autónoma en un robot hay que desarrollar sofisticados algoritmos que deben detectar las condiciones de su entorno y de sí mismo para desplazarse, tratar rápida y eficientemente la información (tiempo real), ser robustos y flexibles (resolver problemas que no se previeron en su programación inicial).

El objetivo de la investigación de que trata este artículo fue desarrollar un algoritmo de navegación autónoma para un robot móvil, con el fin de que éste pueda desplazarse en busca de un objetivo, evite los obstáculos que encuentre en su trayectoria y cumpla con las siguientes características para poder implementarlo en un robot real:

- Flexible (respuesta ante situaciones no previstas, adaptación a cualquier medio).
- Eficiente (bajo consumo de recursos del sistema, ejemplo: “memoria”).
- Rápido (pocas operaciones, respuesta en tiempo real).
- Fiable (respuesta correcta a pesar de los errores de percepción).

Aunque hay diferentes tipos de algoritmos para la navegación autónoma, seguimiento de trayectorias y evasión de obstáculos, éstos pueden clasificarse en cuatro categorías [21]: los algoritmos gráficos, los métodos clásicos, los algoritmos heurísticos y, finalmente, los que hacen uso del campo de potencial. El trabajo realizado en esta investigación se encuentra en el último grupo.

El contenido del presente artículo está dividido en cinco secciones: la primera es el “estado del arte”, en la que se exponen los principales métodos que se basan en el PFM y corrigen el problema del mínimo local; la segunda sección es la fundamentación teórica, donde se explican los dos comportamientos que se usaron en el algoritmo; la tercera sección es la discusión de los experimentos realizados, la cuarta sección son las conclusiones, y en la quinta sección se muestra el futuro de

esta investigación, en aras de implementar los resultados en un entorno real.

ESTADO DEL ARTE

En esta parte del artículo se introducirán los principales métodos de navegación autónoma para robots móviles desarrollados en los últimos años, que han usado el concepto del campo de potencial para guiar al robot a un objetivo mientras éste evade los obstáculos encontrados en su camino.

Khatib (1985) [1] fue el precursor de la propuesta del PFM (*Potential Field Method*): éste tomó como punto de partida el sistema de fuerzas planteado por Craig Reynolds [3], en el cual se relacionan los obstáculos, un objetivo (punto de llegada) y la dirección del robot con fuerzas vectoriales, de tal manera que la suma de sus magnitudes y direcciones proporcionaba un vector resultante que tiende a apartar al robot de los obstáculos y al mismo tiempo intenta acercarlo al objetivo. En 1990, Borenstein y Koren [7] identificaron claramente la gran limitación de este método, el problema del mínimo local (*Local Minima Problem*, LMP), el cual se presenta cuando el robot, en su trayectoria hacia el objetivo, se encuentra en el interior de un obstáculo cóncavo.

Moravec y Elfes [2] [4] [5] tomaron igualmente el PFM para generar un método capaz de resolver el LMP. Su solución consistió en modelar el mundo del robot en una forma matricial, donde cada celda de la matriz correspondía a un valor de probabilidad de existencia de obstáculos “ $C(i,j)$ ”; cuanto más grande era el valor de la celda ($C(i,j)$), más grande era la certeza de que un obstáculo ocupaba esa celda. La fuerza de repulsión generada por los obstáculos era directamente proporcional al valor “ $C(i,j)$ ” e inversamente proporcional al cuadrado de la distancia entre las celdas y el robot, lo que generaba un mapa de trayectorias por donde el robot se podía mover sin chocarse. Este método llamado VFF (*Virtual Force Field*), precedió al VFH[8] y sus posteriores mejoras VFH+[11], VFH*[12] en los años 1998 y 2000. En el 2000, Im y Oh [13] hicieron uso del VFF para desarrollar su método extendido (EVFF), que incluía redes neuronales y una metodología heurística para controlar la trayectoria de un robot sin chocarse y sin permanecer en ciclos infinitos.

En los años 2000[14] y 2003[15] se plantearon dos algoritmos basados en conceptos abstractos sobre obstáculos y objetivos virtuales. El primero obligaba al robot a usar nuevas trayectorias, cambiando los obstáculos cóncavos por obstáculos convexos virtuales, y el segundo se encarga de cambiar el objetivo principal por subobjetivos temporales que guían al robot fuera de obstáculos cóncavos, y una vez afuera continúa con su objetivo.

ABEO - BASES TEÓRICAS

Método del campo de potencial (PFM)

Craig W. Reynolds fue el precursor de la idea de representar el ambiente y comportamiento de un robot por un sistema vectorial de fuerzas. En su artículo *Steering Behaviors for Autonomous Characters*[3], él explica la modelización de diferentes comportamientos animales, tales como la búsqueda o evasión de un objetivo estático y móvil, seguimiento de una trayectoria, seguimiento de un flujo de movimiento, separación y cohesión entre individuos.

A renglón seguido se muestra el primer comportamiento Reynolds para buscar un objetivo (figura 1).

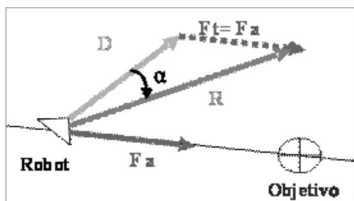


Figura 1. Búsqueda de un objetivo .

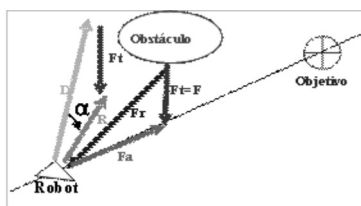


Figura 2. Fuerza de atracción y de repulsión mientras se busca un objetivo.

En la figura 1 se puede observar la existencia de dos vectores de fuerza: la fuerza de movimiento del robot D y la fuerza total F_t , que representa el entorno, pero

que en este ejemplo sólo es la fuerza de atracción que genera el objetivo. La suma de estas dos fuerzas es la fuerza R , la cual corregirá la dirección de movimiento del robot y lo hará avanzar hacia el objetivo.

En la figura 2 se muestra cómo se aplica el mismo principio pero ahora con un obstáculo y la fuerza generada por éste. En el presente caso la fuerza total (F_t) es la suma de las fuerzas de atracción (F_a) y repulsión (F_r) generadas por el objetivo y el obstáculo, respectivamente; el vector resultante (R) se obtiene sumando el vector de movimiento (D) con la fuerza total (F_t). Este vector R representa la nueva dirección del robot que lo llevará al objetivo, al mismo tiempo que evita el obstáculo.

La simpleza y, por ende, su rapidez para calcular las fuerzas de repulsión y de atracción fueron las principales razones consideradas para usar el método PFM. A continuación se muestra su representación:

$$\overline{F_r} = \frac{F_{cr}}{d(i, j)} \left[\frac{x_{obj} - x_{robot}}{d(i, j)} \hat{x} + \frac{y_{obj} - y_{robot}}{d(i, j)} \hat{y} \right]$$

Ecuación 1. Fuerza de repulsión

$$\overline{F_a} = F_{ca} * \left[\frac{x_{obj} - x_{robot}}{d(i, j)} \hat{x} + \frac{y_{obj} - y_{robot}}{d(i, j)} \hat{y} \right]$$

Ecuación 2. Fuerza de atracción

$$\overline{F_t} = \overline{F_a} + \overline{F_r}$$

Ecuación 3. Fuerza total

$$\overline{R} = \overline{D} + \overline{F_t}$$

Ecuación 4. Fuerza resultante

En la ecuación 1 F_{cr} y ecuación 2 son constantes, X_{obj} y Y_{obj} son las coordenadas del objetivo, X_{robot} y Y_{robot} son las coordenadas del robot, $d(i, j)$ la distancia del centro del robot al objetivo, y en la ecuación 4 D es la fuerza de movimiento o dirección del robot.

A partir de las anteriores ecuaciones se puede notar que cuando el robot está lejos del obstáculo la fuerza de repulsión (F_r) generada es baja o nula; en caso contrario, si el robot está cerca del obstáculo la magnitud del vector de repulsión es mayor para evitar que el robot choque. Así mismo, la fuerza de atracción (F_a) es inversamente proporcional a la distancia entre el robot y el objetivo, mientras más cerca esté el robot del objetivo, más gran-

de es la magnitud del vector de atracción.

La simplicidad de las operaciones realizadas para obtener los vectores F_r y F_a es una ventaja cuando se piensa en aplicar el PFM a un robot real, ya que el procesador tomará menos ciclos para calcular cada vector de fuerza y su consecuente vector de movimiento resultante, y se ganará con esto que el robot tenga reacción en tiempo real.

Método de seguimiento de contornos (wall-following)

La idea en general del método de seguimiento de contornos es mantenerse a cierta distancia de un objeto, moviéndose paralelo a la trayectoria que marca el contorno de éste. Dichos objetos pueden ser tanto convexos como cóncavos. El seguimiento de contornos se realiza de la siguiente manera: primero se debe hacer un acercamiento del robot al objeto hasta alcanzar el umbral de cercanía deseado, es decir, la distancia que el robot debe mantener con el objeto; al conseguirlo, el robot debe buscar orientarse de tal manera que el frente esté libre para poder avanzar y su costado se halle a cierta distancia del objeto; si al avanzar la distancia entre el robot y el objeto es mayor o menor que el umbral definido, éste deberá posicionarse nuevamente a la distancia establecida para seguir avanzando.

Borenstein (1989) [6] y Yun y Tan (1997) [10] plantearon dos algoritmos para resolver el problema del mínimo local con la ayuda del método de seguimiento de contornos. El algoritmo presentado difiere en las condiciones que determinan cuándo dejar el PFM y comenzar a usar el seguimiento de contornos para no caer nunca en los mínimos locales. Estas condiciones se presentan a continuación.

Condiciones de estado cíclico y libre (combinación de los dos métodos anteriores)

Estas condiciones definen cuándo el robot debe usar el PFM o cuándo utilizar el método de seguimiento de contornos, con el fin de que el robot pueda sortear cualquier obstáculo sin importar su forma y llegar a su objetivo, solucionando el problema del mínimo local.

Mediante el siguiente ejemplo se explica el problema del mínimo local inherente al método PFM.

En la figura 4A, la fuerza de atracción guía al robot al objetivo; luego de avanzar una cierta distancia (figura

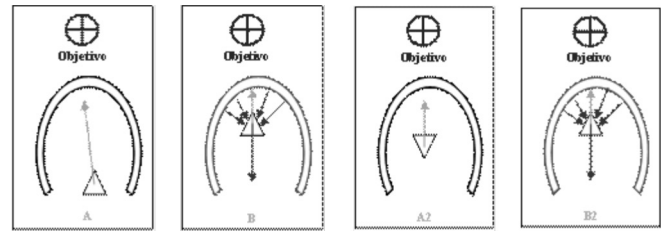


Figura 4. Problema del mínimo local (LMP) generado por usar el PFM con un obstáculo cóncavo.

4B), se encontrará con el obstáculo que lo repele con una fuerza en sentido contrario a la fuerza de atracción. Debido a esto, el robot girará y se alejará del objetivo, evitando el obstáculo; sin embargo, luego de estar nuevamente sin obstáculos (figura 4A2) el robot tratará de dirigirse otra vez al objetivo, regresando al interior del obstáculo y entrando en una condición cíclica (figura 4B2).

Para evitar caer en la trayectoria cíclica ocasionada por el problema del mínimo local se definió la llamada “condición de estado cíclico (CEC)”, que obliga al robot a cambiar de método. La primera condición consiste en que si todos los sensores del robot se activan indicando una distancia menor o igual al radio del robot, éste usará el método de seguimiento de contornos. Seguidamente se muestra un ejemplo:

Cuando la condición de estado cíclico se activa en el robot (figura 5A), se calcula la llamada “recta de detección de estado cíclico (RDEC)”, que pasa por los puntos (X_{obj}, Y_{obj}) y (X_{robot}, Y_{robot}) ,

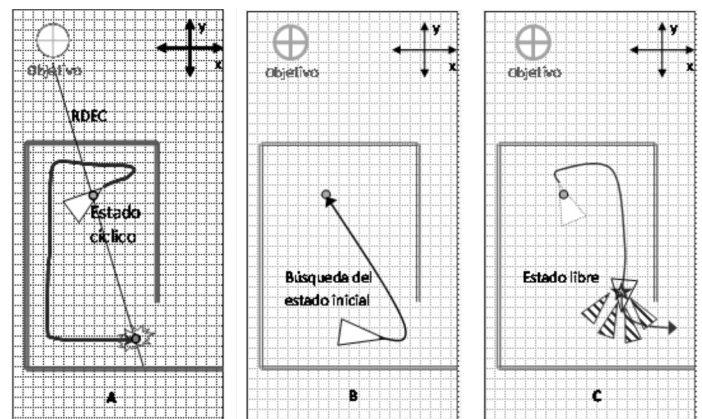


Figura 5. Uso de la recta de detección de estado cíclico, y las condiciones de estado cíclico y libre.

$$y = \left[\begin{matrix} y_{obj} - y_{robot} \\ x_{obj} - x_{robot} \end{matrix} \right] * (x - x_{obj}) + y_{obj}$$

Ecuación 5. Recta de detección de estado cíclico

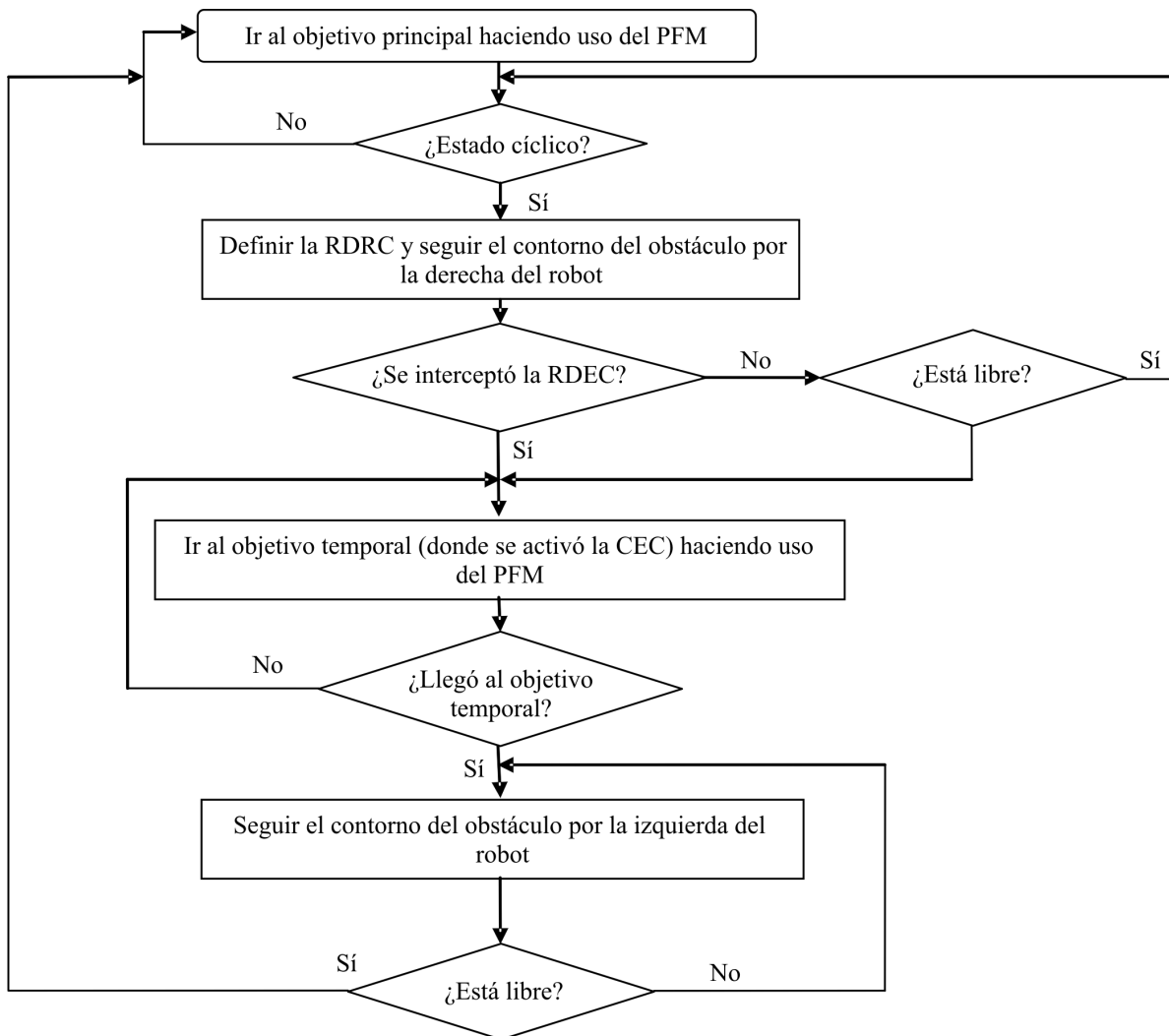
Luego de calcular la RDEC, el robot comienza a usar el método de seguimiento de contornos recorriendo el obstáculo por su derecha; se evalúa si la posición actual cumple, con un grado de tolerancia, la ecuación de la recta RDEC; si cumple, significa que entró en un bucle (figura 5A) y debe regresar al punto en el cual comenzó la condición CEC, usando el PFM para cambiar temporalmente su objetivo, y empieza a seguir el contorno del obstáculo por su izquierda (figura 5B).

No obstante, si esta condición no se activa, a medida que el robot avanza rodeando el muro, cada nueva

posición es evaluada en la RDEC; si en algún momento del recorrido del robot su posición (x, y) satisface la ecuación de la RDEC, significa que éste la cruzó. Para llegar a este punto se hace uso del PFM, con la salvedad de que el objetivo es el punto de activación de la CEC. En el momento en que el robot llega al objetivo temporal, éste comienza a seguir el contorno del obstáculo por su izquierda (figura 5B).

Si en algún momento del recorrido el robot no detecta obstáculos se generará la segunda condición, llamada “condición de estado libre” (CEL), en la cual el robot volverá a usar el método PFM, dirigiéndose al objetivo principal (figura 5C).

A continuación se muestra el diagrama de flujo del algoritmo:



DISCUSIÓN DEL EXPERIMENTO

Para probar el algoritmo se empleó el simulador de robots móviles Mobotsim, en tanto que para poder comparar el desempeño del algoritmo se usaron configuraciones de obstáculos similares a las propuestas por otros autores [21].

Geometría del robot

En la simulación se utilizó un robot con las siguientes características:

- Cinemática de tipo diferencial.
- Diámetro de 40 cm (*platform diameter*).
- Distancia entre las ruedas de 25 cm (*distance between wheels*).
- Diámetro de las ruedas de 20 cm.
- Cuatro sensores, separados un ángulo de 45 grados, con un rango de medición de 4 cm hasta 80 cm (valores similares a los de un sensor de distancia infrarrojo), con un cono de radiación de 15 grados y con error producido por una falsa detección de un objeto (*misreading*) del 5% (un *misreading* del 10% significa que de cien detecciones de obstáculos, diez de éstas son inexistentes).

Entorno

El *software* MobotSim posee un mapa de coordenadas rectangulares (x, y) donde pueden ponerse obstáculos

de diferentes formas, simplemente dibujándolos encima del mapa. En este entorno, el robot puede conocer su posición actual en cualquier momento de manera exacta, la posición del objetivo, y las distancias precisas entre el robot y los objetos por medio de los sensores de distancia.

Programación

El MobotSim dispone del lenguaje Sax Basic© para programar los comportamientos del robot. Éste se modela como un objeto (entendido éste con la referencia de la programación por objetos) con diferentes funciones para hacer giros y desplazamientos, conocer su posición actual y la del objetivo, entre otras.

Pruebas

Para comenzar, se hicieron pruebas en ambientes cuyas configuraciones fueron difíciles para otros algoritmos (Ayomoh y Olunloyo [21]):

Como se observa en la figura 6, el resultado de los experimentos demostró que el algoritmo es efectivo en ambientes similares a los probados [21] y sus predecesores. Se observa además que las trayectorias no son siempre iguales, puesto que dependen no sólo de la posición inicial y la ubicación de los obstáculos, sino también del ángulo inicial del robot.

Se efectuaron otras pruebas utilizando los mismos mapas de la figura 6, pero con un porcentaje de error

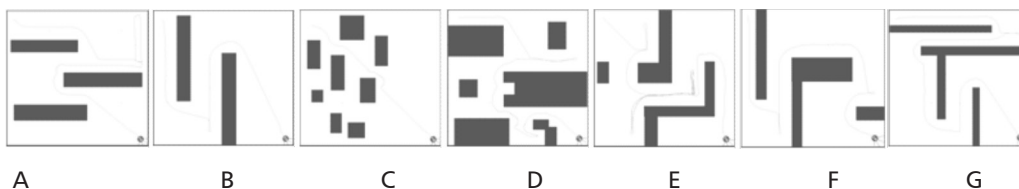


Figura 6. Entornos de prueba comunes a otros algoritmos.

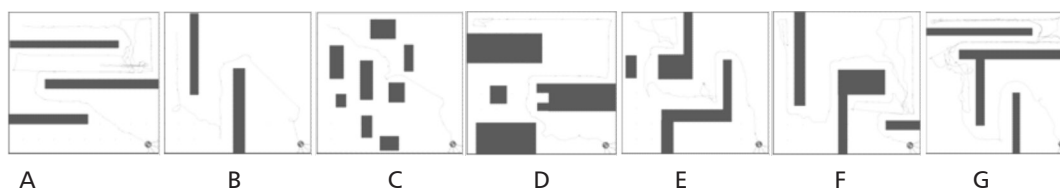


Figura 7. Entornos de prueba comunes a otros algoritmos.

del 10% en las mediciones de los sensores, simulando al error que un sensor infrarrojo podría presentar.

El resultado fue exitoso en todas las pruebas realizadas, lo que demuestra que el algoritmo es capaz de sobrellevar estos errores y hacer que el robot alcance el objetivo, evitando el problema del mínimo local.

El algoritmo propuesto usa sólo cuatro sensores, comparado con los algoritmos basados en el VFF [21] que necesitan 24 o más; por tanto, requiere adquirir menos datos, y realizar menos número de conversiones y operaciones, ya que éstas son simples, por lo que toman pocos ciclos de operación del procesador. A menor cantidad de ciclos de procesador, más rápida es la respuesta del robot, por lo cual se pueden alcanzar mayores velocidades.

La última prueba incluye un ambiente más complejo, como recrear espacios y objetos en una casa.

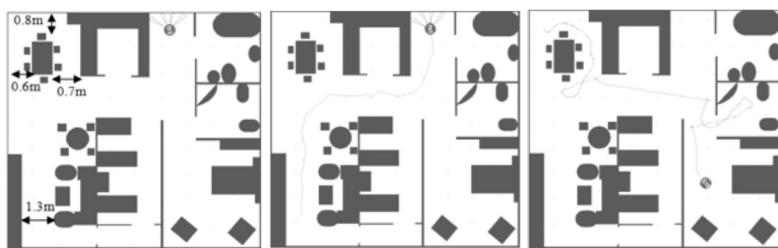


Figura 8. Mundo virtual de prueba, resultado sin error de medición y con error del 10%.

El robot fue capaz de llegar al objetivo luego de pasar por lugares estrechos hasta con error del 10% en las mediciones de distancias y sorteando los problemas del mínimo local.

CONCLUSIONES

Un robot móvil, que implemente el algoritmo de evasión de obstáculos basado en el concepto del campo de fuerza combinado, está en capacidad de alcanzar un objetivo esquivando obstáculos de cualquier tamaño o cualquier forma sin detenerse, sin permanecer en estados cíclicos debido a espacios semicerrados cóncavos, y respondiendo rápida y correctamente, a pesar de posibles errores de percepción.

De los experimentos realizados se determinó la alta fiabilidad del algoritmo propuesto, ya que con errores de percepción de hasta un 10%, el robot puede evitar obstáculos y llegar al objetivo sin caer en el error del mínimo local.

El método propuesto requiere menos *hardware*, en comparación con otros métodos basados en el PFM como el VFF, el VHF y sus derivados, el VOC, el VGC, o el HVFF. El robot únicamente necesita cuatro sensores, lo que disminuye no solo costos, sino tiempo de procesamiento al requerir menos operaciones de adquisición, conversión, procesamiento de información y cálculo de fuerzas de repulsión.

Comparado con otros algoritmos basados en el PFM que resuelven el problema del mínimo local, el algoritmo propuesto responde más rápido, al efectuar operaciones muy sencillas (sumas, restas y comparaciones) para calcular la condición de estado cíclico, la condición de estado libre y el método de seguimiento de contornos, además de necesitar menos operaciones por usar tan sólo cuatro sensores.

Utilizar como base el método PFM permite al robot tener una respuesta en tiempo real ante cambios en el ambiente, no requiere guardar en memoria la descripción del entorno para generar la trayectoria que lo conduzca al objetivo, y su combinación con el método de seguimiento de contornos permite resolver en una forma sencilla el problema del mínimo local inherente al PFM.

TRABAJOS POSTERIORES

La investigación posterior se centrará en la aplicación del algoritmo a un robot real diferencial con posicionamiento basado en GPS (*Global Position System*), medición de distancia a los obstáculos utilizando sensores infrarrojos y estimación de la posición angular del robot por métodos odométricos empleando codificadores ópticos.

REFERENCIAS

- [1] Khatib, O. (1985). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *IEEE International Conference on Robotics and Automation*.
- [2] Moravec, H.P. & Elfes, A. (1985). High Resolution Maps from Wide Angle Sonar, *IEEE Conference on Robotics and Automation*. Washington, D.C., pp. 116-121.
- [3] Reynolds, C.W. (1986). *Steering Behaviors For Autonomous Characters*, Sony Computer Entertainment America.

- [4] Elfes, A. (1987). Sonar-based Real-World Mapping and Navigation, *IEEE Journal of Robotics and Automation*, vol. RA-3, No. 3, pp. 249-265.
- [5] Moravec, H.P. (1988). Sensor Fusion in Certainty Grids for Mobile Robots, *AI Magazine*, Summer, pp. 61-74.
- [6] Borenstein, J. & Koren, Y. (1989). Real-time obstacle avoidance for fast mobile robots, *IEEE Trans. on Systems, Man and Cybernetics*, 19(5): 1179-1187.
- [7] Borenstein, J. & Koren, Y. (1990). Real-time obstacle avoidance for fast mobile robots in cluttered environments, *Proceedings of IEEE International Conference on Robotics and Automation*. Cincinnati, Ohio, pp. 572-577.
- [8] Borenstein, J. & Koren, Y. (1991, June). The Vector Field Histogram - Fast Obstacle Avoidance For Mobile Robots, *IEEE Journal of Robotics and Automation*, vol. 7, No. 3, pp. 278-288.
- [9] Connolly, C.I. (1992). Applications of harmonic functions to robotics, *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 498-502.
- [10] Yun, X.P. & Tan, K.C. (1997). A wall-following method for escaping local minima in potential field based motion planning, *Proceedings of International Conference on Advanced Robotics*. Monterrey, pp. 421-426.
- [11] Ulrich, I. & Borenstein, J. (1998). VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*. Leuven, Belgium, May 16-21, pp. 1572-1577.
- [12] Ulrich, I. & Borenstein, J. (2000). VFH*: Local Obstacle Avoidance with Look-Ahead Verification, *IEEE International Conference on Robotics and Automation*. San Francisco, CA, April 24-28, pp. 2505-2511.
- [13] Im, K.Y. & Oh, S.Y. (2000). An Extended Virtual Force Field Based Behavioral Fusion with Neural Networks and Evolutionary Programming for Mobile Robot Navigation, *Evolutionary Computation, IEEE Congress*, vol. 2, pp. 1238-1244.
- [14] Chengqing, L., Hang, M., Krishnan, H. & Ser Yong, L. (2000). Virtual Obstacle Concept for Local-minimum-recovery in Potential-field Based Navigation, *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*.
- [15] Zou Xi-yong, Zhu Jing (2003). Virtual local target method for avoiding local minimum in potential field based robot navigation, ISSN 1009 - 3095, *Journal of Zhejiang University Science*, vol. 4, No. 3, pp. 264-269.
- [16] Seul Jung, Eun Soo Jang, Hsia, T.C. (2005). Collision Avoidance of a Mobile Robot Using Intelligent Hybrid Force Control Technique, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. Barcelona.
- [17] Zhiqiang Yu, Gao Meng, Huaping Liu, Xiaoyan Deng, Jianhua Liu, Qiurui Wu & Yuewei Liu (2008). Dynamic Obstacle Avoidance in Polar Coordinates for Mobile Robot Based on Laser Radar, *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*.
- [18] Hiroaki Seki, Satoshi Shibayama, Yoshitsugu Kamiya & Masatoshi Hikizu (2008). Practical Obstacle Avoidance Using Potential Field for a Nonholonomic Mobile Robot with Rectangular Body, *Proceedings of the IEEE International Conference on Automation and Logistics Qingdao*. China.
- [19] Filliat, D. (2008). Interactive learning of visual topological navigation, Paris: Ensta - UEI.
- [20] Fan Wen, Zhenshen Qu, Changhong Wang & Bin Hu (2008). Study on Real-Time Obstacle Avoidance of Mobile Robot Based on Vision Sensor, *Proceedings of the IEEE International Conference on Automation and Logistics Qingdao*. China.
- [21] Ayomoh, M.K.O. & Olunloyo, V.O.S. (2009). Autonomous Mobile Robot Navigation Using Hybrid Virtual Force Field Concept, *European Journal of Scientific Research*, ISSN 1450-216X, vol. 31, No. 2, pp. 204-228, EuroJournals Publishing, Inc., <http://www.eurojournals.com/ejsr.htm>.