

(ABEO)-algoritmo bioinspirado de evasión de obstáculos

(ABEO) algorithm for obstacle avoidance bioinspired

JOSÉ L. SUSÁ

Estudiante de Ingeniería Electrónica e investigador del grupo de investigación Electrónica de la Escuela Colombiana de Ingeniería

DIEGO A. RAMOS

Estudiante de Ingeniería Electrónica e investigador del Grupo de investigación Electrónica de la Escuela Colombiana de Ingeniería

Clasificación del artículo: Investigación (conciencias)

Fecha de recepción: 27 de agosto de 2009

Fecha de aceptación: 26 de enero de 2010

Palabras claves: robótica móvil, evasión de obstáculos, bioinspirado, algoritmo

Key words: mobile robotic, obstacle avoidance, bioinspired, algorithm

RESUMEN

En este artículo se presenta el desarrollo, prueba y resultados obtenidos del Algoritmo Bioinspirado de Evasión de Obstáculos - (ABEO). El objetivo de este artículo es divulgar los resultados de la investigación que tuvo como propósito desarrollar un algoritmo que le permitiera a un robot móvil desplazarse de forma autónoma, a fin de alcanzar un objetivo evitando los obstáculos que encuentran 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.

La prueba del algoritmo se realizó en un entorno simulado, con un robot que solo usa cuatro sensores de distancia, de características similares a uno real infrarrojo, en vez de veinticuatro sensores como en otros métodos de evasión de obstáculos similares. En la pruebas realizadas se comprobó la fiabilidad del método, se observó que ante situaciones no preestablecidas, tales como nuevos obstáculos en el entorno y con errores de medición de hasta del 50% el robot alcanzaba el objetivo, al

mismo tiempo que evadía obstáculos.

ABSTRACT

This paper presents the development, testing and results of bioinspired algorithm for obstacle avoidance - (ABEO). The objective of this paper reveals the results of research's purpose to develop an algorithm that would allow a mobile robot to navigate autonomously, to achieve a goal avoiding obstacles in its path. The requirements for the design of the algorithm were: high-speed response, low consumption of hardware resources and capacity to respond to unforeseen situations.

The proof of the algorithm is performed in a simulated environment with a robot that uses only four distance sensors, with characteristics similar to a real infrared sensors instead of twenty-four as in other methods like obstacle avoidance. The tests proved the reliability of the method, it was noted that no predetermined situations, such as new obstacles in the environment and measurement errors of up to 50% the robot reached the goal, while dodging obstacles.

1. Introducción

Una tarea tan simple para un ser vivo como es la de buscar alimento (objetivo), esquivando obstáculos y retornando a su madriguera (punto de partida) es un problema que aún no ha sido totalmente resuelto para un robot móvil, el cual tiene un limitado conocimiento del entorno y reducida flexibilidad para resolver problemas nuevos que no fueron previstos en su programación inicial.

El objetivo de la investigación que presenta este artículo fue desarrollar un algoritmo capaz de controlar el movimiento de un robot móvil diferencial, a fin de que éste alcance un objetivo evitando los obstáculos que encuentre en su trayectoria, y cumpla con las siguientes características:

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 responda correctamente a pesar de los errores de percepción.

Las características anteriores son importantes para en un futuro inmediato lograr la implementación del algoritmo en un robot real, la flexibilidad le permitirá adaptarse a cualquier medio, la eficiencia y la rapidez reducirán los requerimientos y costos del hardware¹ y, finalmente, la fiabilidad será la clave para superar los errores de medición o el ruido que afectan el resultado del algoritmo y, por ende, el control del robot.

El símil que hay entre las anteriores características citadas con las que posee un cerebro animal simple (de un insecto, o un ratón), demuestra el objetivo inicial de este trabajo de investigación; se buscó que las respuestas del algoritmo fueran

1 En un trabajo posterior se desea implementar el ABEO en un microprocesador, el cual tiene restricciones de memoria y velocidad que obligan al ABEO a que sea de alto rendimiento y de limitado número de operaciones.

semejantes al de un comportamiento animal y por esto se integraron dos conductas bioinspiradas: la primera es la acción de avanzar siguiendo contornos de objetos (muros y obstáculos), propia de los roedores. La segunda conducta es natural a todo ser vivo, la de dirigirse a un objetivo, como a una madriguera, a una presa o simplemente al árbol adecuado para alimentarse, guiándose, ya sea por medio del olor, de la visión, o de la posición como en el caso de los insectos². Esta conducta de dirigirse a un objetivo, junto con la primera conducta ya citada, conforman el fundamento del control de movimiento del robot.

No es suficiente con tener dos comportamientos definidos si no se determina una condición que establezca en qué momento uno u otro debe ser efectuado; por esto, adicional al trabajo de generar las conductas, el algoritmo debe manejar un sistema de condiciones que le permita al robot decidir entre el uso del comportamiento de seguimiento de contornos o el de direccionamiento a un objetivo. En la sección de las bases teóricas se ampliará este tema y se analizarán las condiciones de estado atrapado y de estado libre, encargadas de hacer la conmutación entre conductas.

Al final de este artículo se mostrarán los resultados obtenidos en las pruebas del ABEO en un entorno simulado, en el cual el robot fue capaz de pasar obstáculos cóncavos, convexos, pequeños y de gran tamaño.

2. Estado del arte

En esta parte del artículo se muestran los principales métodos desarrollados para evasión de obstáculos que han precedido al ABEO. El primero de éstos llamado VFF nació en 1989 con el planteamiento de Khatib [1]; éste tomaba como punto de partida el sistema de fuerzas planteado por Craig

2 Un insecto reconoce un lugar y es capaz de dirigirse a otro, gracias a un mapa mental que forma con algunos puntos característicos del entorno (Landmarks) con los que reconocen los objetos que lo componen y con los que a partir de éstos pueden decidir cómo alcanzar ciertas posiciones de acuerdo con la actual.

Reynolds [2], en el cual se relacionan los obstáculos, un objetivo (punto de llegada) y la dirección del robot de manera vectorial. Los obstáculos se representan como vectores de repulsión, y el objetivo como un vector de atracción, sus magnitudes proporcionales al inverso de la distancia entre el obstáculo/objetivo y el robot se sumaban junto con la dirección de este último para formar un vector resultante que apartara al robot de los obstáculos y al mismo tiempo lo acercara al objetivo. Sin embargo el VFF, poseía limitaciones que limitaban su aplicación.

El VFF, fue el punto de partida para los estudios en Carnegie- Mellon University (CMU) [3] [4] [5] para modelar el mundo del robot en una forma matricial, la celdas de la matriz se definían como valores de probabilidad de existencia de obstáculos " $C(i,j)$ ". Entre más grande era el valor de la celda ($C(i,j)$), más grande es la certeza de que un obstáculo ocupa 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. Esto generaba un mapa de trayectorias por donde el robot se movía sin chocarse. Este método fue ampliamente probado en diferentes robots móviles hasta descubrir luego de algunas investigaciones las limitaciones del algoritmo a causa de obstáculos con geometrías cóncavas -debido a una permanencia en bucles o ciclos infinitos- y se postuló un nuevo método, en 1991, llamado VFH[6] y sus posteriores mejoras VFH+[7], VFH*[8] en 1998 y 2000.

En estos últimos métodos se abandonó la idea del campo de fuerzas vectoriales, pero se guardó la modelización del mundo en forma de matriz de probabilidades; así mismo, se sumó al nuevo método un análisis de histogramas -en coordenadas polares, distancia versus ángulo al obstáculo- y funciones de costo para encontrar las mejores rutas posibles que llevarán al robot a su objetivo; al final se generó un algoritmo complejo que le permitía al robot hacer movimientos suaves, rápidos y una evasión de obstáculos relativamente confiable. Sin embargo, estos métodos requieren un aprendizaje previo del lugar por donde se desplazaba el robot y sus costos son elevados ya que el robot debe disponer de 24 sensores de ultrasonido.

Más tarde, se desarrollaron nuevos algoritmos que buscaban hacer un mapeo de los objetos circundantes al robot en cada instante para de esta manera reconocer los espacios libres por donde el robot podía moverse sin estrellarse; en estos algoritmos se hacía un cálculo de áreas de peligro de choque y seguridad para moverse, dependían del ángulo en el que se encontraba el robot en relación con los obstáculos analizados, sus velocidades y posiciones presentes y futuras. De esta manera se le ordenaba al robot pasar por las áreas que implicaban menos peligro de colisión. El primero de estos métodos fue llamado VOA por sus siglas en inglés "Obstacle Velocity Approach" [9], y su mejora fue el método BOA [10] en 2005. La ventaja de estos métodos con respecto a los anteriores es su capacidad para prevenir choques contra objetos estáticos, u objetos móviles (otros robots) de forma muy confiable.

En la actualidad, los investigadores prueban métodos de reconocimiento de lugares utilizando visión artificial[11], apoyados en algoritmos de tratamiento y caracterización de imágenes como el "Bag of Words", "Sift", Visual LandMarks, entre otros, con los cuales se busca obtener más información sobre el entorno del robot. En estos métodos no se cuenta con ningún tipo de sensor métrico o mapa preestablecido. Sin embargo, al hacer uso de complejos métodos de procesamiento de imágenes y por requerir un alto consumo de memoria, la rapidez de respuesta del robot es sustancialmente más baja que con cualquier otro método que no use visión artificial.

3. ABEO - Bases teóricas

3.1. Sistema vectorial de fuerzas

Craig W. Reynolds fue el precursor de la idea de representar el ambiente y comportamiento de un robot por un sistema vectorial. En su artículo, "Steering Behaviors For Autonomous Characters" [2], se explica la modelización realizada de diferentes comportamientos animales, tales como:

- Búsqueda o evasión de un objetivo estático y móvil.

- Seguimiento de una trayectoria.
- Seguimiento de un flujo de movimiento.
- Separación y cohesión entre individuos.

En el siguiente ejemplo se muestra el primer comportamiento de búsqueda de un objetivo. En la figura 1 se puede observar la existencia de dos vectores de fuerza D y Ft; estos representan la fuerza de movimiento del robot y la Fuerza Total generada por el entorno; esta última 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 haciéndolo avanzar hacia el objetivo.

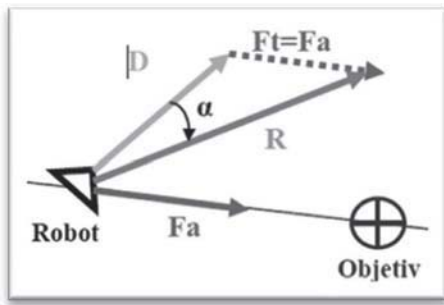


Figura 1. Búsqueda de un objetivo

En la figura 2 se muestra como se aplica el mismo principio pero ahora con un obstáculo y la fuerza generada por éste.

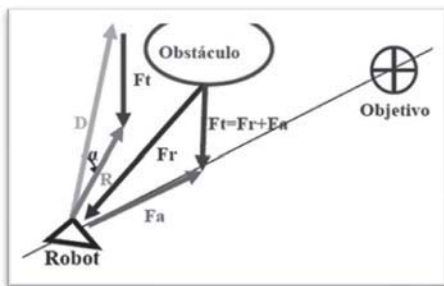


Figura 2. Fuerza de atracción y de Repulsión mientras se busca un objetivo

En este caso, la fuerza de total (Ft) es la suma de

las fuerzas de atracción (Fa) y repulsión (Fr) generadas por el objetivo y el obstáculo, respectivamente. El vector resultante (R) se obtiene sumando el vector de movimiento (D) con la fuerza total (Ft), este vector R representa la nueva dirección del robot que lo llevará al objetivo al mismo tiempo que evita el obstáculo.

El método VFF fue el primero en usar el comportamiento de búsqueda en un robot real, su modo de calcular las fuerzas de repulsión y de atracción se usó para el desarrollo del ABEO. En su forma matemática estas fuerzas están descritas en las ecuaciones 1, 2 y 3.

Fuerza de repulsión:

$$\overline{Fr} = \frac{Fcr}{d(i,j)} \left[\frac{X_{obj} - X_{robot}}{d(i,j)} \hat{x} + \frac{y_{obj} - y_{robot}}{d(i,j)} \hat{y} \right] \quad (1)$$

Fuerza de atracción:

$$\overline{Fa} = Fca * \left[\frac{X_{obj} - X_{robot}}{d(i,j)} \hat{x} + \frac{y_{obj} - y_{robot}}{d(i,j)} \hat{y} \right] \quad (2)$$

Fuerza total:

$$\overline{Ft} = \overline{Fa} + \overline{Fr} \quad (3)$$

Fuerza R que hay que aplicar al robot para obtener el comportamiento deseado es: $R = D + Ft$

En estas ecuaciones se tiene:

- Fcr y Fca constantes de repulsión y atracción.
- Xobj e Yobj son las coordenadas del objetivo.
- Xrobot e Yrobot son las coordenadas del robot o la posición actual de éste.
- d(i,j) = Distancia del centro del robot al objetivo.
- 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 “Fr” que éste le genera 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 se choque. Igualmente la fuerza de atracción “Fa” es inversamente proporcional a la distancia entre el robot y el objetivo, entre más cerca esté el robot del objetivo, más grande es la magnitud del vector de atracción.

Las ventajas significativas de este método vectorial en relación a un método exhaustivo de reglas fijas (conductas predefinidas), es que en el primero los movimientos del robot son suaves (giro progresivo), en un método común cuando un sensor detecta un obstáculo el robot reacciona bruscamente girando a veces hasta 180 grados. En este método de fuerzas, el robot poco a poco se va alejando del obstáculo y va desviando su trayectoria a pasos cortos. La simplicidad de las operaciones realizadas para obtener los vectores Fr y Fa es una ventaja al aplicarse en un robot real, ya que le permite reaccionar rápidamente para evadir obstáculos, debido al bajo número de operaciones

que debe realizar. A diferencia de algunos robots actuales que se deben detener para saber cuál es el camino más libre por donde ir [4], un robot que use un sistema de fuerzas puede moverse y reaccionar en tiempo real.

3.2. Seguimiento de contornos

En este apartado se explican los fundamentos del método de seguimiento de contornos ampliamente usado en el mundo animal (roedores, hormigas, entre otros) y en aplicaciones de robots móviles autónomos, como en el caso de los robots usados en competencias estudiantiles para salir de laberintos.

Este método resulta tanto eficaz como ineficiente, aunque puede hacer que cualquier robot salga de un laberinto simplemente manteniéndose cerca de uno de los muros, el tiempo requerido para poder encontrar la salida puede ser excesivo. Esto se puede ver en las siguientes figuras:

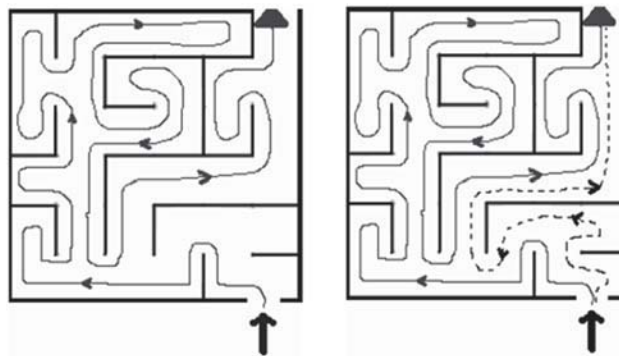


Figura 3. Recorrido del laberinto por la izquierda (Línea continua) y por la derecha (Línea punteada)

En las dos figuras de arriba el robot puede salir, ya sea si sigue el muro por su izquierda o por su derecha, la diferencia es el tiempo que se toma cada solución en llevar al robot a la salida; para corregir esto se tendría que conocer la configuración del laberinto para saber cuál ruta se debe tomar para llegar más rápido, y la generalidad del algoritmo quedaría limitada al tipo de laberinto. Como la idea no es tener un algoritmo para cada laberinto

se debe optar por tomar uno de los dos caminos y, por tanto, se puede caer en uno que lo haga recorrer todo el laberinto antes de salir. Por eso este tipo de solución es muy discutida para usarla únicamente en robots que tengan que navegar dentro de un lugar cerrado. En un caso más real, si el robot se mueve en un apartamento y se quiere que el robot vaya de la cocina a la alcoba, podría recorrer todo el lugar antes de llegar.

4. Condiciones de estado atrapado y libre (Combinación de los dos métodos anteriores)

Éste es uno de los puntos clave del algoritmo ABEO. Su éxito radica en la condición que determina cuándo el robot debe usar el método de fuerzas o el método de seguimiento de contornos; su combinación permite que el robot sortee cualquier obstáculo y llegue a su objetivo.

La primera condición que se estudiará será la de “atrapado”, ésta le dará el punto de partida a la segunda. Para lo cual se usará un ejemplo con un obstáculo de tipo cóncavo o una esquina, generalmente, imposible de pasar, si se utiliza solamente el método de fuerzas.

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. Estos objetos pueden ser tanto muros, como objetos cuyos contornos sean menos extensos. Técnicamente hablando, 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, el cual está determinado por la distancia al objeto que el robot debe mantener. Al conseguirlo, el robot debe buscar posicionarse de tal manera que su frente esté libre para poder avanzar y su costado esté a cierta distancia del objeto; si al avanzar la distancia entre el robot y el objeto es mayor o menor al umbral definido, se deberá posicionarse nuevamente a la distancia establecida para seguir avanzando.

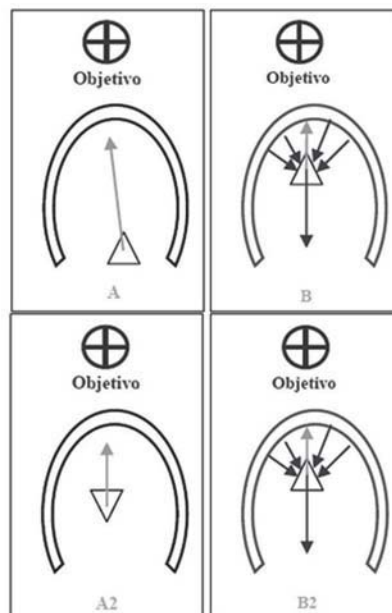


Figura 4. Problema de bucle generado por solo usar el algoritmo de fuerzas

En la figura 4, el robot es guiado al objetivo (figura 4A); luego de avanzar un cierto espacio (figura 4B) se va a topor con un obstáculo que lo repele con una fuerza en sentido contrario a la fuerza de atracción (vector que apunta al objetivo); debi-

do 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 de nuevo al objetivo, por lo cual vuelva a regresar de nuevo al mismo punto

de donde se alejó una vez más (figura 4B2). A este estado se le ha denominado estado atrapado o bu-

cle (ciclo infinito).

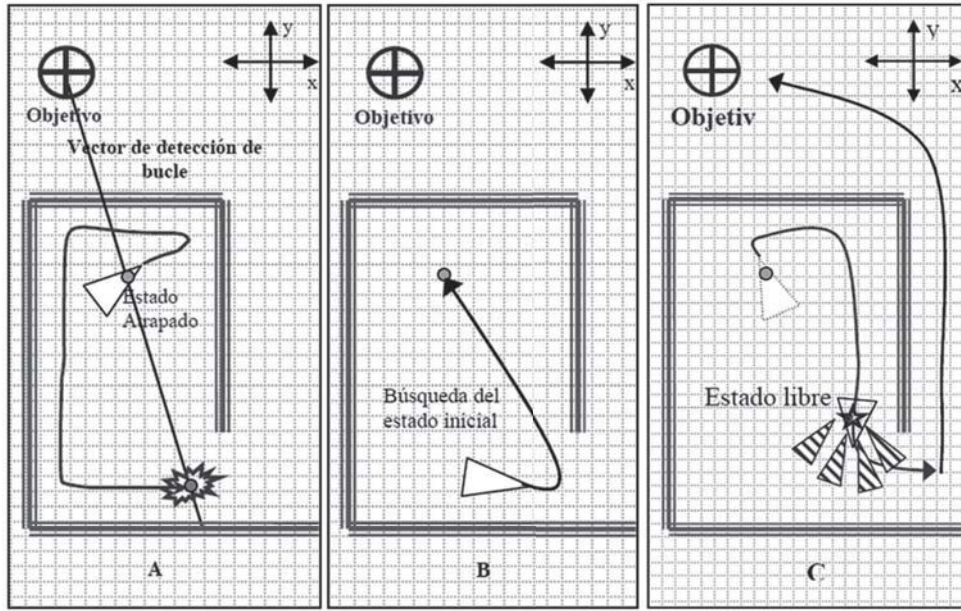


Figura 5. Uso del vector de detección de bucle y la condición de estado libre

Si la condición de estado atrapado se activa en el robot, éste comienza a usar el método de seguimiento de contornos (figura 5A). Se genera un vector de detección de bucle, el cual pasa por el punto en el que se sintió atrapado. En el momento de interceptar el vector de detección de bucle, el robot regresa al punto de partida (figura 5B), y desde ese punto comienza el seguimiento del muro por su izquierda, como lo muestra la figura 5C. En el punto marcado con la estrella el robot “sentirá que está libre”, ya que todos sus sensores no detectan ningún obstáculo y no hay más muro por seguir. Entonces, El robot realiza un pequeño avance y un giro de tal manera que termine de salir del obstáculo y pueda continuar en la búsqueda del objetivo mediante el sistema de fuerzas (figura 5C).

5. Desarrollo del experimento

A continuación se explicarán las condiciones de

la simulación

5.1. Geometría del robot:

El robot simulado es de tipo diferencial, su morfología es determinada por una serie de parámetros configurables, los cuales son: el diámetro del robot (*Platform diameter*), distancia entre el centro del robot y las ruedas (*Distance between wheels*), diámetro de las ruedas (*Wheels diameter*).

Un número de sensores de distancia puede ser definido por el usuario en todos los 360° de perímetro del robot, configurables igualmente bajo los siguientes parámetros: Rango de medición (*Range*), distancia mínima (*zona muerta*) y máxima medible, distancia entre sensores (*Angle Between Sensors*), cono de radiación (*Radiation Cone*), error producido por una falsa detección de un objeto (*Misreadings*). Por ejemplo, si se tiene un misreading del 10%, de 100 detecciones de obstáculos,

Luego de varios tests se observó que con sólo tres sensores se generaba un problema porque se evitaban los obstáculos frontales, debido a la activación simultánea de los sensores. Se decidió agregar un sensor más para conseguir una separación mayor entre los sensores de los extremos para así reducir la posibilidad de que se activaran al mismo tiempo y se anularan las fuerzas que generaban al detectar un obstáculo frontal. Con esta nueva configuración de 4 sensores el robot al encontrarse con un muro primero por su derecha giraba correctamente para evitar el choque. Con estas mejoras se lograron obtener los siguientes resultados:

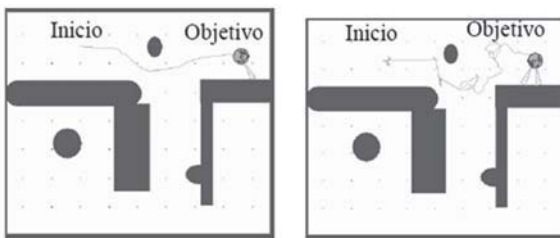


Figura 8. Alcanzar el objetivo. A) Misreadings = 1%, C) Misreadings = 50%

En las figura 8, se muestra la robustez y confiabilidad del método ABEO. Este algoritmo es capaz de dirigir al robot hacia el objetivo rápidamente hasta con un 50% de error de medición de sus sensores, lo que evita perfectamente el obstáculo que se interpone en el camino. En la primera prueba con un error bajo, el robot llegó al objetivo sólo con el uso del sistema de fuerzas; en el recorrido nunca se sintió atrapado y esta percepción lo llevó rápidamente al objetivo. En las siguientes imágenes el error aumenta proporcionalmente los movimientos del robot, se vuelve más inestable pero también alcanza el objetivo. Para ver la sensibilidad de los sensores, se pusieron obstáculos de un tamaño reducido:

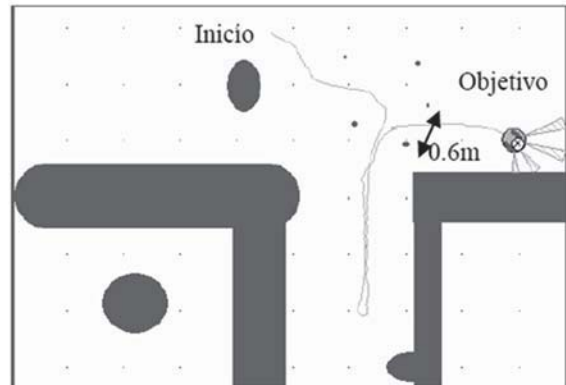


Figura 9. Alcanzar el objetivo pasando por obstáculo de reducido tamaño. Misreadings=1%

Tan bien como lo hizo con un obstáculo de un tamaño proporcional al robot, con los obstáculos pequeños el robot no tuvo inconveniente para llegar al objetivo.

Luego de los anteriores test, se hicieron pruebas del algoritmo ABEO con obstáculos cóncavos que, como se explicó anteriormente, generan un comportamiento de bucle.

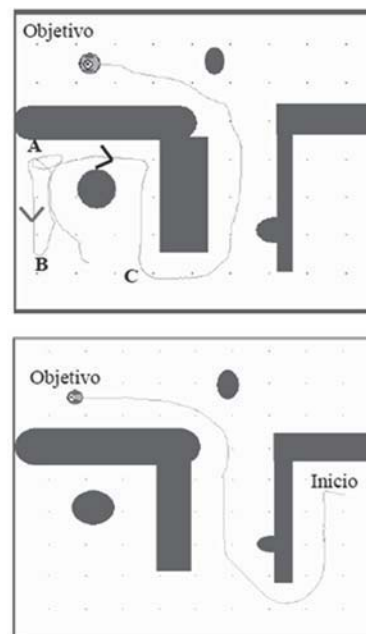


Figura 10. Alcanzar el objetivo pasando por obstáculos cóncavos. Misreadings =1%

y rápida. Igualmente, la capacidad de memorizar permitirá una reducción de los cambios de dirección en intervalos cortos (movimiento más suave), ya que memorizando las posiciones y ordenes de

giro se podrá evitar que el robot gire de forma contraria en dos instantes continuos, y tenga un movimiento de zigzag innecesario.

Referencias

- [1] O. Khatib. “*Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*”. IEEE International Conference on Robotics and Automation, 1985.
- [2] W. Craig Reynolds. “*Steering Behaviors For Autonomous Characters*”. Sony Computer Entertainment America. 1986.
- [3] A. Elfes “*Sonar-based Real-World Mapping and Navigation*”. IEEE Journal of Robotics and Automation, Vol. RA-3, No 3, 1987, pp. 249-265.
- [4] H.P Moravec and and A. Elfes “*High Resolution Maps from Wide Angle Sonar*” IEEE Conference on Robotics and Automation, Washington D.C., 1985, pp. 116-121.
- [5] H.P Moravec. “*Sensor Fusion in Certainty Grids for Mobile Robots*”. AI Magazine, 1988, pp. 61-74.
- [6] J. Borenstein and Y. Koren “*The Vector Field Histogram - Fast Obstacle Avoidance For Mobile Robots*” IEEE Journal of Robotics and Automation Vol 7, No 3, June 1991, pp. 278-288.
- [7] I. Ulrich and Borenstein. “*VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots*”. Proceedings of the 1998 IEEE International Conference on Robotics and Automation. Leuven, Belgium, May 16–21, 1998, pp. 1572 – 1577.
- [8] I. Ulrich and Borenstein. “*VFH*: Local Obstacle Avoidance with Look-head Verification*”. 2000 IEEE International
- [9] P. Fiorini and Shillerz “*Motion Planning in dynamic Environments*”. Int. Journal of Robotics Research, Vol 17 No.(7), 1998 pp. 760 - 772.
- [10] M. Becker and D. Meirelles, Carolina and Perdigão M., Weber. “*Obstacle Avoidance Procedure for Mobile Robots*”. Congreso Internacional de Ingenieria Mecanica edición 18, 2005.
- [11] D. Filliat “*Interactive learning of visual topological navigation*”. ENSTA – EI, Paris- Francia, 2008.