Revista Tekhnê
Julio - Diciembre 2015, Vol. 12, No. 2, pp. 75– 82

ISSN 1692-8407

Evaluación de estrategia de navegación autónoma basada en comportamiento reactivo para plataformas robóticas móviles

Evaluation of autonomous navigation strategy based on reactive behavior for mobile robotic platforms

Angélica V. Rendón C.
Universidad Distrital Francisco José de Caldas
avrendonc@correo.udistrital.edu.co

En esta investigación se presenta una evaluación de desempeño sobre una plataforma real de una estrategia utilizada para el cálculo del movimiento de un robot no holonómico en un ambiente desordenado. El robot no tiene acceso a la información métrica sobre la ubicación de los obstáculos o de su propia posición, ni tampoco control de odometría o velocidad. Se utiliza el reconocimiento en tiempo real de la señal radiada desde la posición de destino, que permite al robot navegar y aprender información global acerca de la ubicación del destino. Se demuestran tareas de ejemplo que se pueden realizar con este enfoque.

Palabras clave: Planeación de movimiento, robot móvil, sistema no holonómicos, sonar

We presented a performance evaluation on a real platform of an approach to computing motion strategies for a nonholonomic robot in a cluttered environment. The robot has no access to either metric information about the location of obstacles and its own position, or to odometry or speed controls. We use real-time recognition of radiated signal from the target position, that allow the robot to navigate and to learn global information about the target location. We demonstrate example tasks that can be performed using this approach.

Keywords: Mobile robot, nonholonomic system, path planning, sonar

Tipología del artículo: Investigación

Fecha recepción del manuscrito: Mayo 1, 2015
Fecha aceptación del manuscrito: Junio 5, 2015

Investigación financiada por: Universidad Distrital Francisco José de Caldas.

Edición digital: http://revistas.udistrital.edu.co/ojs/index.php/tekhne/issue/view/755
Cómo citar: Rendón, A. (2015). *Evaluación de estrategia de navegación autónoma basada en comportamiento reactivo para plataformas robóticas móviles*. Revista Tekhnê, 12(2), 75-82.

Introducción

Durante la primera mitad del siglo XVIII en la revolución industrial, la humanidad decidió sustituir y/o apoyar procesos de producción con plataformas robóticas, lo cual implicó uno de los mayores avances tecnológicos a lo largo de la historia. Esta forma de desarrollar tareas implicó la parición de un sin número de problemas de investigación, tanto a nivel funcional (control, navegación, etc.) como a nivel de interacción con el ser humano (seguridad, integración, etc.).

Este artículo se enfoca en la evaluación de desempeño de una estrategia de navegación robótica autónoma, con características de exploración, para robots en ambientes desconocidos. Este ha sido un problema de alta investigación en los últimos años debido a sus implicaciones en aplicaciones reales (Bobadilla, Sanchez, Czarnowski, Gossman, y LaValle, 2011; Marques et al., 2013).

La navegación robótica se enfoca en la ubicación espacial de un robot con respecto al entorno que lo rodea, normalmente con la intención de que se desplace desde una ubicación inicial hasta una ubicación deseada. Para coordinar el movimiento normalmente se utilizan estrategias de control como:

- Navegación autónoma, particularmente útil en ambientes dinámicos y desconocidos. En ella se tiene en cuenta variaciones del entorno en tiempo real.
- Pre-programacion de ruta, útil en en entornos estructurados, normalmente estáticos y observables globalmente.

La falta de libre albedrío en un robot limita sus posibilidades de movimiento a una sola, siendo incapaz de reaccionar frente a estímulos externos, lo cual es un problema cuando se enfrenta a situaciones no pre-programadas. Los robots con estrategias de navegación predeterminadas son en este sentido ineficientes, ya que la gran mayoría de los entornos reales se encuentran en constante cambio (ambientes dinámicos).

Una primera posible estrategia de solución, bastante intuitiva, es hacer que el robot siga las paredes del ambiente, como si tratara de salir del laberinto, hasta lograr llegar a su destino (Byoung-Kyun et al., 2015; Byoung-Kyun, Won-Jun, Kyung-Sun, Le, y Sung-Hyun, 2012). Si bien es claro que el robot reconoce la ubicación final al encontrarla, en general no resulta una estrategia eficiente en cuanto a garantía de solución y tiempo requerido para la tarea. Estrategias más eficientes deben permitir establecer una ruta hasta el destino, ya sea a partir de información global o local.

Otras estrategias más robustas, y que hacen parte de la teoría clásica de navegación robótica, parten de la ubicación exacta del robot en el ambiente, así como la ubicación de los obstáculos (a través de procesos previos de mapeos o sensado *on-line*), y ejecutan una estrategia secuencial de control de odometría y velocidad desde el origen hasta el destino (Arroyave y Gallego, 2012; Kuo-Ho, Tan-Phat, Chan-Yun, y Wen-June, 2014; Qiwen, Rekleitis, y Dudek, 2015).

En esta investigación se pretende evaluar el desempeño sobre prototipos reales de una estrategia de control híbrida pensada para robots autónomos en cualquier tipo de ambiente (Tovar, Murrieta, y LaValle, 2007). Se dice que es una estrategia híbrida dado que si bien el modelo y comportamiento del robot real es continuo, sus reacciones son caracterizadas por un conjunto de comportamientos que se activan de acuerdo a las lecturas realizadas en el ambiente, lo cual se asemeja a una conmutación discreta (Carvalho, Motta, y Santos, 2014; Castiblanco y Martínez, 2014).

Formulación del problema

Suponga que uno o más robots (para el caso general n robots) se mueven en un ambiente $\mathcal{W} \subset \mathbb{R}^2$. Para evitar complicaciones con el espacio de configuración, se asume que cada robot es un punto. En éste ambiente $\partial\mathcal{W}$ denota los límites o frontera de $\mathcal{W}$. Si $\mathcal{O}$ es el conjunto de obstáculos en el ambiente, en el que cada $O \in \mathcal{O}$ es cerrado, con fronteras conectadas analíticas a pedazos y finitas en longitud. Estos obstáculos representan regiones inaccesibles a los robots, por lo que sus fronteras hacen parte de la frontera del ambiente $\partial\mathcal{W}$.

Los robots se consideran pequeños con respecto a $\mathcal{W}$, lo cual coincide con la suposición de ser modelados como puntos. Sin embargo, ellos pueden tener una cinemática específica, como en el caso común de los robots diferenciales. Más específicamente, la suposición es que el sub-conjunto libre de colisión de $\mathcal{W}$ (espacio libe) es homeomórfico (desde el punto de vista topológico, es igual) al obtenido para el caso de un robot tipo punto, sin importar la geometría de cada robot. Además, cualquier región $r \in \mathcal{R}$, con $\mathcal{R}$ como sub-conjunto de $\mathcal{W}$, se asume es capaz de contener a todos los n robots sin problemas. El robot navega moviéndose de una región a otra hasta llegar al destino.

Las ecuaciones particulares de movimiento $\dot{x} = f(x)$ para cada robot no son importantes en este enfoque. No se pretende controlar explícitamente los movimientos del robot, y tampoco se intenta medir de forma precisa su estado. En lugar de ello, se confía en el hecho de que los robots se mueven de una forma incontrolable, pero la trayectoria de su movimiento satisface la siguiente propiedad de orden superior: Para cualquier región $r \in \mathcal{R}$, se asume que el robot se mueve en una trayectoria que causa que golpee cada intervalo abierto en ∂r (la frontera de r) un número infinito de veces, con velocidad no tangencial y diferente de cero.

Un conjunto de sistemas que cumple esta condición de movimiento es los *sistemas ergódicos*. En ellos, una partícula newtoniana se mueve a velocidad constante y luego rebota

ISSN 1692-8407

en la frontera con leyes de reflexión estándar. En la mayoría de las regiones planas, se consigue ergodicidad. Un modelo alternativo utilizado en estos experimentos es rebotar en la frontera con un ángulo aleatorio. Esto es mucho más conveniente en robots reales dado que es bastante complejo sensar el ángulo de incidencia. En el caso de n robots, ellos también rebotan un ángulo aleatorio cuando se encuentran unos con otros.

Un *modo de control* es un mapeo de la forma $u : C$, el cual asigna un determinado comportamiento al robot a partir de la lectura de un evento c en el ambiente, donde C es el conjunto de todos los eventos considerados por el robot ($c \in C$). Sea U el conjunto de todos los posibles modos de control ($u \in U$). El modo de control $u(c)$ asigna un determinado movimiento al robot.

Para un único robot, se define el sistema de transición discreto D que simula al sistema híbrido. Por lo tanto, si el espacio de estados del sistema discreto es $\mathcal{R}$, entonces el sistema de transición se define como (ecu. 1):

$$D = (\mathcal{R}, r_0, \rightarrow_1), \tag{1}$$

En este sistema r_0 es la región que inicialmente contiene al robot, y la relación de transición a la región 1 se denota por $r_0 \rightarrow_1$.

Se plantea un sistema basado en eventos. Cada robot comienza con un modo de control inicial, y durante la navegación, el modo de control puede cambiar cuando un sensor observa un determinado evento y. Dependiendo del sistema, los diferentes eventos pueden ser:

1. **Intensidad de una señal:** El robot detecta variaciones de intensidad de la señal en el ambiente, y coordina su respuesta (movimiento) de acuerdo a su gradiente (sonido, luz, etc.). La observación es de la forma $y = f(c)$.

2. **Tiempo:** El robot debe esperar en una determinada región por al menos t segundos. La observación es de la forma $y = f(t)$.

3. **Comunicación:** El robot recibe a mensaje del robot i que indica que el robot i ha avanzado a una nueva región, lo que habilita al robot a avanzar también (seguimiento de un líder). La observación es de la forma $y = f(c, i)$.

Si Y denota al conjunto de todos los posibles eventos observables por el robot en un determinado sistema, entonces $y \in Y$.

Los modos de control son ajustados durante la navegación de acuerdo a una política. Ya que debido a la naturaleza del robot no es posible realizar realimentación de estados (y tampoco se busca), en su lugar se realiza *realimentación de información*. Sea un *filtro* un mapeo de la forma $\phi : \mathcal{I} \times Y \rightarrow \mathcal{I}$, en el que $\mathcal{I}$ denota un *espacio de información* (LaValle, 2006) que es diseñado apropiadamente para la tarea ($\mathcal{I}$ depende de la tarea). Se asume un $\iota \in \mathcal{I}$ inicial, y cada vez que ocurra la observación de un nuevo evento, ocurre una transición de estado en el filtro activada por información. Una *política de control* es definida como un mapeo de realimentación de información de la forma $\pi : \mathcal{I} \rightarrow U$, el cual permite que el modo de control sea ajustado en coherencia con la historia de observaciones detectadas por el sensor (Fig. 1).

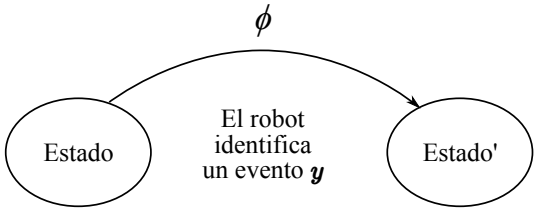

Figura 1. Transición de estado activada por información sensada.

Algoritmo I-Bug

El algoritmo I-Bug fue formulado por Taylor y LaValle (Taylor y LaValle, 2009) como respuesta a la búsqueda de un algoritmo con muy bajos requisitos de sensado y mapeo del ambiente, características identificadas inicialmente por ellos en los esquemas clásicos, y ampliamente conocidos, de algoritmos Bug, tanto seguidores de bordes de obstáculos como buscadores de destino (Bug clásico, TangentBug, WedgeBug y RoverBug entre los más conocidos) (Kamon, Rimon, y Rivlin, 1999; Kamon y Rivlin, 1995; Laubach y Burdick, 1998; Lumelsky, 2005). Sin embargo, estos algoritmos aún requieren de una cantidad apreciable de información, que en el enfoque de I-Bug resulta innecesaria.

Por ejemplo, los algoritmos originales Bug1 y Bug2 (Lumelsky, 2005) requieren para su funcionamiento de cinco elementos:

- Sensor de contacto (o impacto).
- Coordenadas de la posición inicial del robot (origen).
- Coordenadas de la posición actual del robot.
- Coordenadas de la posición final del robot (destino).
- Odometría para obtener la distancia navegada alrededor de la frontera de un obstáculo.

Esto es así debido a que ambos algoritmos calculan la ruta mas corta del origen al destino bordeando a los obstáculos que encuentran en su camino. Las variantes de estos algoritmos utilizan principios de operación similar, con algunas modificaciones que no alteran en general estos cinco requisitos. Por ejemplo, los algoritmos TangentBug, VisBug y WedgeBug agregan diferentes tipos de sensor de alcance para incrementar el desempeño del algoritmo Bug original.

En general, todos estos algoritmos Bug y variante se caracterizan por la necesidad de información relacionada con las coordenadas exactas del robot. Esta es una característica necesaria en procesos de mapeo de ambientes desconocidos, pero no en procesos de navegación, sea o no conocido el ambiente. Un robot autónomo solo necesita saber donde esta el destino, y cómo evitar los obstáculos que encuentre en su

camino. Este enfoque es similar al comportamiento de una persona que se encuentra, por ejemplo, por primera vez en un centro comercial, y empieza la búsqueda y navegación hacia el baño. El éxito de la navegación esta en identificar correctamente la información disponible en el ambiente.

El algoritmo I-Bug parte de este principio, y propone el uso de un sistema localizado en la posición de destino que emite una señal perceptible en todo el ambiente. La señal se distribuye en el ambiente por medio de una función de intensidad sobre $\mathbb{R}^2$, en donde el valor máximo de intensidad, 1, se encuentra en la posición de destino (Fig. 2).

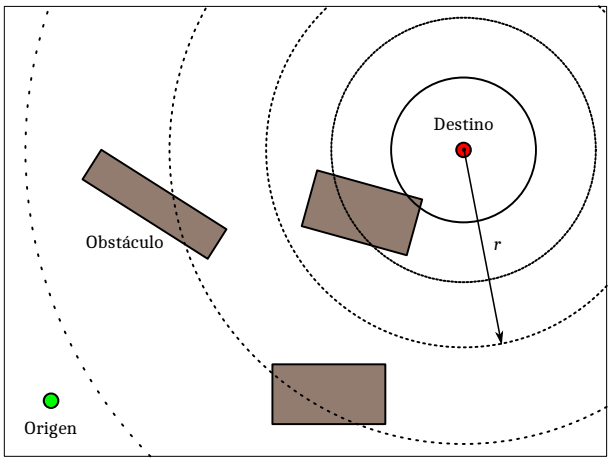

Figura 2. El robot inicia en el punto de origen (inferior izquierdo), y se mueve hacia el punto destino (superior derecho). Los arcos representan a la señal radiada con una misma intensidad, la cual se reduce al alejarse de la fuente (punto destino). El robot navega siguiendo el gradiente de intensidad, mientras esquiva los obstáculos.

En principio, la señal emitida puede ser cualquiera, por lo que el algoritmo permite cualquier tipo de función de intensidad. Por ejemplo, en el caso del sonido (onda sonora esférica), la intensidad es inversamente proporcional al cuadrado de la distancia desde el centro de la esfera (punto de emisión), es decir $1/r^2$. Este comportamiento se puede observar en la Fig. 3.

En un caso mas general, los conjuntos de nivel de intensidad concéntricos con centro en el punto destino, mostrados en la Fig. 2, pueden no ser circulares, dependiendo de la capacidad del medio para transmitir la señal emitida. Aún bajo estas circunstancias de asimetría, en principio se mantiene la característica de intensidad concéntrica con respecto al punto destino, condición mínima suficiente para el funcionamiento del algoritmo.

El algoritmo considera el uso de dos sensores:

• Sensor de contacto: Indica si el robot esta en contacto con la frontera del ambiente $\partial\mathcal{W}$ (como se definió, esto incluye la frontera de los obstáculos). El sensor entrega dos posibles valores (ecu. 2):

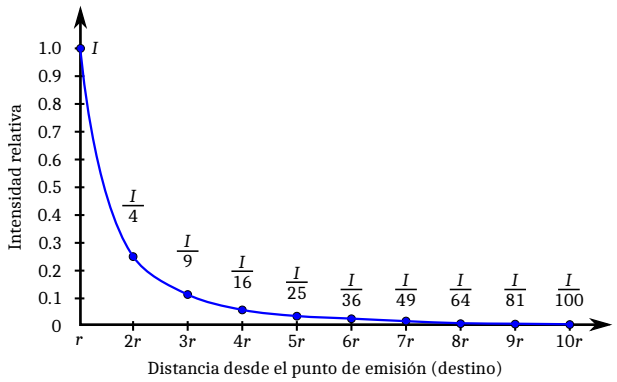

Figura 3. Caída de la intensidad del sonido de acuerdo con la ley del cuadrado inverso.

$$h_{sc}(x,y) = \begin{cases} 1 \text{ si } (x,y) \in \partial\mathcal{W} \\ 0 \text{ en otro caso} \end{cases} \quad (2)$$

• Sensor de intensidad: Entrega el valor de intensidad de la señal en el punto (x, y) del ambiente. Se trata de una variable continua con rango [0, 1], dependiente de la ubicación del sensor, la capacidad del medio para transmitir la señal emitida, la ubicación del punto destino, y de la función de intensidad de la señal, m.

$$h_{si}(x,y) = h(x,y,\mathcal{W},x_{destino},y_{destino},m) \quad (3)$$

El algoritmo I-Bug original considera el uso de un tercer sensor, un sensor de alineación con el punto destino o gradiente. En esta propuesta se ha eliminado dicho sensor, dado que se considera entrega información redundante, si el sistema utiliza dos o mas sensores de intensidad.

Plataforma robótica

La plataforma robótica utilizada en las pruebas es un sistema diferencial con orugas (no ruedas) (Fig. 4). El robot tiene una base de 12.4 cm por 10.5 cm, y una altura total de 24.6 cm. En la parte superior, a cada lado de la cabeza, se colocan dos sensores de sonido (dos sensores en total para el calculo del gradiente de intensidad). El sensor de impacto se coloca al frente del robot. Se instala un sensor adicional de luz en la parte superior de la cabeza.

Como unidad de control se utiliza un microcontrolador ATmega328P de Atmel a 16 MHz.

Sensores

Sensor de sonido. Para implementar el sensor de sonido se utiliza un micrófono común de audio. La señal se amplifica con un transistor bipolar (TIP31C) en configuración emisor común. El circuito base se muestra en la Fig. 5.

ISSN 1692-8407

Figura 4. Plataforma robótica utilizada en las pruebas de desempeño.

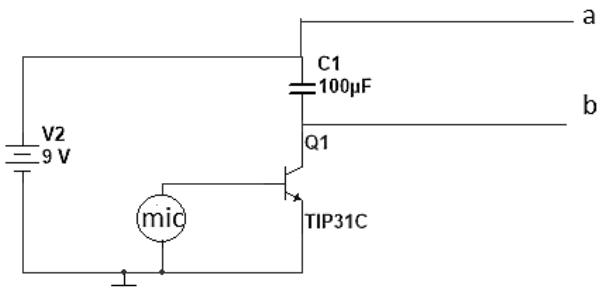

Figura 5. Topología del sensor de sonido.

Para la sintonización del circuito se asigna un voltaje de operación al micrófono de 3.5 V. Cuando es activado satura el transistor de potencia, lo que genera la diferencia de potencial en las salidas *a* y *b*, y en el condensador. Estas salidas serán las que activen las determinadas políticas de control, que a través de un puente H se traducen en cambio de dirección de los motores y del robot. El condensador tan solo conserva esta acción por un tiempo determinado por su constante de tiempo.

El vector gradiente de intensidad se calcula determinando la distancia euclidiana entre las lecturas de los dos sensores de sonido.

Sensor de impacto. El sensor de impacto se implementa con un simple pulsador mecánico instalado al frente del robot, el cual cierra el circuito binario al estrellarse con un obstáculo o la frontera del ambiente. La acción del pulsador de amplia a todo el frente del robot con un pequeño parachoque frontal.

Sensor de luz. Este es un segundo sensor pensado para su uso en la evaluación del algoritmo de acuerdo a la intensidad de señal lumínica, en lugar de la acústica. Sin embargo, en las pruebas de desempeño aquí documentadas no se utiliza como tal, sino como variable de control de movimiento del robot (se condiciona el movimiento del robot a la presencia de luz).

El sensor se implementa utilizando una foto-resistencia o LDR. Debido a que el valor de resistencia de un LDR varía dependiendo de la intensidad de luz incidente (desciende hasta los 50 Ω cuando incide luz en él, y llega a ser del orden de los mega ohmios en la oscuridad), se determinó, mediante pruebas de laboratorio, que en espacios con luz ambiente él LDR tiene un valor resistivo de 1000 Ω. El LDR se utiliza para polarizar un transistor de potencia TIP31C, el cual energiza directamente los motores del robot (Fig. 6).

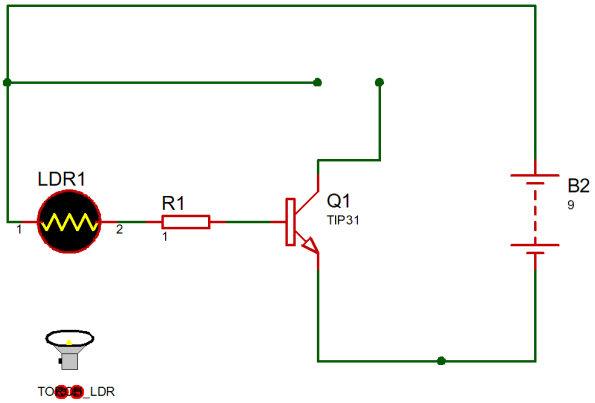

Figura 6. Topología del sensor de luz.

La corriente en colector depende de los motores utilizados. Por medio de pruebas de laboratorio, emulando el peso del robot, se midió la corriente necesaria para el arranque de los motores. En promedio se determinó que la corriente que pide la carga es de 800 mA. La corriente de base será (ecu. 4):

$$I_B = \frac{I_C}{\beta} = \frac{0{,}8}{168} = 4{,}8\,\text{mA} \tag{4}$$

Haciendo ley de tensiones (ecu. 5):

$$-V_{RB} - V_{trans} - V_{LDR} + V_F = 0 \tag{5}$$

Despejando V_{RB} de la ecu. 5 se obtiene la expresión para la resistencia de base.

Actuadores

Como actuadores se utilizan dos pequeños motores DC acoplados por un conjunto de piñones, y activados a través de un puente H. El puente en H usado es el integrado L239B, cuya conexión con los motores esta ilustrada en la Fig. 7.

Política de control

De acuerdo a la configuración de hardware descrita para el robot, y considerando que no se realiza control de velocidad de los motores (avanzan en los dos sentidos a velocidad

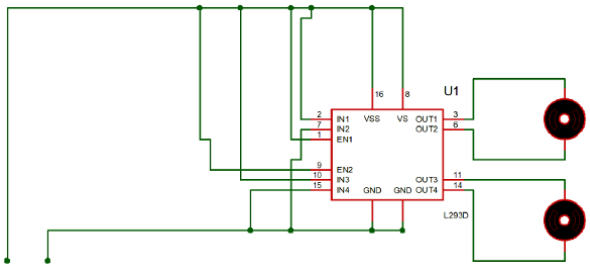

Figura 7. Conexión motores DC y puente H del robot.

nominal, o se quedan quietos, dependiendo del voltaje de alimentación: Vcc, -Vcc y 0 V) es posible definir tres primitivas de movimiento:

- Avance frontal, en caso de que un motor se alimente con Vcc y el otro con -Vcc.
- Avance en reversa, en caso de que los motores se alimenten con voltajes contrarios al caso anterior (-Vcc uno y Vcc el otro).
- Sin avance, en caso de que no se entregue voltaje a los dos motores (0 V de alimentación para los dos).
- Giro a la derecha, en caso de que el motor izquierdo se alimente para avance frontal (con Vcc) y el motor derecho con = 0 V.
- Giro a la izquierda, en caso de que el motor derecho se alimente para avance frontal (con -Vcc) y el motor izquierdo con = 0 V.

Cada primitiva de movimiento es determinada por la unidad de control de acuerdo a la información capturada por los sensores en el ambiente. Cada primitiva de movimiento se ejecuta durante un intervalo de acción calculado a partir de la velocidad de respuesta del robot (mayor al retardo del sistema), para el caso de las pruebas se utilizó un intervalo de acción de dos segundos.

El algoritmo I-Bug se programa entonces en la unidad de control siguiendo las siguientes políticas de control:

1. Sin importar el estado del robot, si el sensor de choque detecta una frontera ($h_{sc} = 1$), el robot debe hacer un pequeño avance en reversa, luego un giro a la derecha o izquierda (el sentido se selecciona aleatoriamente) un ángulo aleatorio entre 0 y 360 grados, y luego continuar su avance frontal.

2. Para definir la primitiva de movimiento, el robot debe calcular la intensidad del sonido en cada uno de los dos sensores de sonido.

3. Si el error de los dos valores con respecto a la referencia de 1 es mayor al 10 %, se procede entonces al cálculo del vector gradiente de intensidad entre los dos sensores de sonido. En caso contrario, se asume que el robot llegó a la región que contiene al punto destino.

4. De acuerdo al vector gradiente de intensidad calculado, el robot debe girar a la derecha o izquierda (la dirección depende del menor ángulo requerido) a fin de alinearse con el vector de gradiente, en el sentido en el que se incrementa su valor.

5. Aplicar un avance frontal durante un intervalo de acción.

6. Reiniciar el paso (2).

Resultados

Las pruebas de desempeño se realizaron sobre un ambiente cuadrado de 4 m por 4 m. Se diseñaron cinco configuraciones diferentes del ambiente cambiando la localización de los obstáculos, pero manteniendo siempre el punto destino en las mismas coordenadas (parte superior derecha del ambiente) y transmitiendo ininterrumpidamente la misma señal con la misma intensidad.

En los cuatro primeros ambientes se colocaron cinco (5) obstáculos, y en la quinta configuración siete (7) obstáculos. Todos los obstáculos son observables por el robot (al igual que las fronteras del ambiente) por medio del sensor de impacto.

En cada configuración del ambiente se realizaron un total de 40 ensayos (con cinco ambientes, se registró un total de 200 pruebas). La única diferencia entre ensayos sobre un mismo ambiente es la localización inicial del robot, la cual fue seleccionada aleatoriamente, así como su orientación inicial.

La Fig. 8 muestra los resultados de estas pruebas. En el eje horizontal se enumeran los cinco ambientes del 1 al 5, y en el eje vertical el tiempo en segundos requerido para la navegación del robot desde el origen (posición inicial aleatoria del robot) hasta el destino (extremo superior derecho del ambiente). El robot logró alcanzar el destino en el 100 % de los casos. La representación para cada ambiente se construye con los cuartiles de la distribución de los tiempos, los límite inferior y superior de la caja azul la conforman los límites del primer y tercer cuartil de la distribución. La línea central corresponde al valor medio. Los rombos en verde en los ambientes 3 y 5 se clasifican como eventos raros y no se incluyen en el análisis estadístico.

El valor medio (tiempo esperado para el cumplimiento de la tarea) para los cuatro ambientes fue:

1. 68 s
2. 85 s
3. 94 s
4. 75 s
5. 105 s

Como se observa, para esta estrategia de navegación no es posible definir empíricamente el tiempo requerido para el desarrollo de la tarea. Sin embargo, si es posible demostrar estadísticamente que la tarea se desarrolla satisfactoriamente en el 100 % de los casos (robustez del sistema) (Erickson, Knuth, Okane, y LaValle, 2008). El tiempo promedio requerido se puede estimar a partir del

ISSN 1692-8407

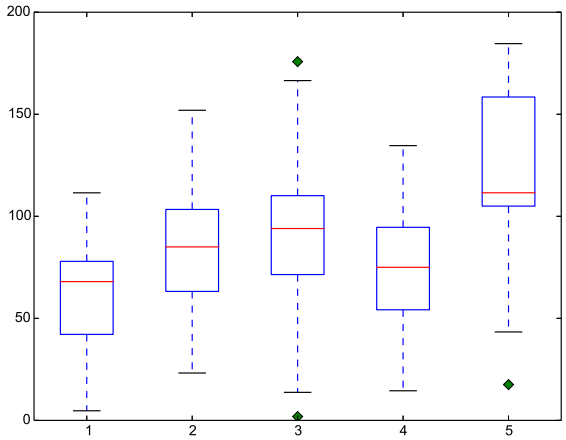

Figura 8. Gráfica box-and-whisker de los tiempos en segundos obtenidos en cinco ambientes diferentes. En cada caso se detallan los cuartiles y el valor medio obtenido en cada ambiente.

estudio de la configuración del ambiente, dado que es fuertemente dependiente de ella. Sin embargo, variables como el robot y la señal misma también afectan el tiempo requerido.

Conclusiones

El artículo presenta la evaluación de desempeño en laboratorio del algoritmo de navegación I-Bug. Para dicha evaluación, se implementa una plataforma diferencial con dos sensores: uno de impacto y otro de intensidad de señal. La señal utilizada, emitida desde el punto destino, es una señal constante esférica de 50 Hz. Con esta configuración del sistema se realizan muchas pruebas del laboratorio, moviendo a lo largo del ambiente tanto el punto de origen como el punto destino. De los resultados, se concluye que el algoritmo es robusto al lograr converger en el 100 % de los casos (lleva al robot a su destino en todos los casos). Sin embargo, se pudo verificar que no es posible determinar analíticamente de forma explícita el tiempo requerido para el desarrollo de la tarea. Este análisis debe realizarse de forma probabilística, y según se observa, es fuertemente dependiente de la configuración del ambiente y la intensidad de la señal.

Con el desarrollo de la investigación aparecen y quedan abiertos muchos problemas de gran interés. La estrategia puede ser aplicada sin mayores alteraciones a una amplia variedad de sensores y señales. Sin embargo, la viabilidad y complejidad de cada una de estas posibilidades debe ser estudiadas como casos particulares. También es interesante estudiar los resultados utilizando otras métricas de distancia para determinar los gradientes de intensidad, y si esto tiene alguna incidencia en el tiempo promedio requerido para el desarrollo de la tarea.

Referencias

Arroyave, S., y Gallego, J. (2012). Path planning applied to the mobile robot gbot. En *Xvii symposium of image, signal processing, and artificial vision (stsiva)* (p. 281-288).

Bobadilla, L., Sanchez, O., Czarnowski, J., Gossman, K., y LaValle, S. (2011). Controlling wild bodies using linear temporal logic. En *Robotics: Science and systems* (p. 17-24).

Byoung-Kyun, S., Jun-Seok, Y., Eok-Gon, K., Yang-Keun, J., Jong, B., y Sung-Hyun, H. (2015). A travelling control of mobile robot based on sonar sensors. En *2015 15th international conference on control, automation and systems (iccas 2015)* (p. 1241-1243).

Byoung-Kyun, S., Won-Jun, H., Kyung-Sun, C., Le, X., y Sung-Hyun, H. (2012). A study on real-time implementation of obstacle avoidance for autonomous travelling robot. En *12th international conference on control, automation and systems (iccas 2012)* (p. 1896-1899).

Carvalho, V., Motta, C., y Santos, F. (2014). A hybrid approach for path planning and execution for autonomous mobile robots. En *Joint conference on robotics: Sbr-lars robotics symposium and robocontrol (sbr lars robocontrol 2014)* (p. 124-129).

Castiblanco, M., y Martínez, F. (2014). Exploración de un modelo comportamental basado en el quorum sensing bacterial para describir la interacción entre individuos. *Tekhnê*, *11*(1), 21-26. (ISSN 1692-8407)

Erickson, L., Knuth, J., Okane, J., y LaValle, S. (2008). Probabilistic localization with a blind robot. En *Ieee international conference on robotics and automation icra 2008* (p. 1821-1827).

Kamon, I., Rimon, E., y Rivlin, E. (1999). Range-sensor based navigation in three dimensions. En *Ieee international conference on robotics and automation* (Vol. 1, p. 163-169).

Kamon, I., y Rivlin, E. (1995). Sensory based motion planning with global proofs. En *Ieee/rsj international conference on intelligent robots and systems 95. 'human robot interaction and cooperative robots'* (Vol. 2, p. 435-440).

Kuo-Ho, S., Tan-Phat, P., Chan-Yun, Y., y Wen-June, W. (2014). Image-based smooth path planning for wheeled robot. En *11th ieee international conference on control & automation (icca)* (p. 203-207).

Laubach, S., y Burdick, J. (1998). Practical autonomous path planner for turn-of-the-century planetary microrovers. En *Mobile robots xiii and intelligent transportation systems* (Vol. 3525).

LaValle, S. (2006). *Planning algorithms*. Cambridge University Press.

Lumelsky, V. (2005). *Sensing, intelligence, motion: How robots and humans move in an unstructured world*. Wiley.

Marques, F., Santana, P., Guedes, M., Pinto, E., Lourenco, A., y Barata, J. (2013). Online self-reconfigurable robot navigation in heterogeneous environments. En *Ieee international symposium on industrial electronics (isie 2013)* (p. 1-6).

Qiwen, Z., Rekleitis, I., y Dudek, G. (2015). Uncertainty reduction via heuristic search planning on hybrid metric/topological map. En *Computer and robot vision (crv 2015)* (p. 222-229).

Taylor, K., y LaValle, S. (2009). I-bug: An intensity-based bug algorithm. En *Ieee international conference on robotics and automation icra '09* (p. 3981-3986).

Tovar, B., Murrieta, R., y LaValle, S. (2007). Distance-optimal navigation in an unknown environment without sensing distances. *IEEE Transactions on Robotics*, *23*(1), 506-518.