


Sistema robótico autónomo para la exploración y construcción de mapas en entornos estructurados

Autonomous robot system for exploration and map building in structured environments

Gustavo Alonso Acosta-Amaya 

Politécnico Colombiano Jaime Isaza Cadavid, Colombia

Andrés Felipe Acosta-Gil 

Universidad Nacional de Colombia, Colombia

Jovani Alberto Jiménez-Builes 

Universidad Nacional de Colombia, Colombia

Open Access

Recibido:

5 agosto de 2019

Aceptado:

4 noviembre de 2019

Publicado:

1 enero de 2020

Correspondencia:

gaacosta@elpoli.edu.co

afacostag@unal.edu.co

jajimen1@unal.edu.co

DOI:

<https://doi.org/10.17081/invinno.8.1.3593>



© Copyright: Investigación e Innovación en Ingenierías

Resumen

Objetivo: Construir un sistema robótico para la exploración y el mapeo de entornos interiores de trabajo con una arquitectura y una metodología de diseño propia. **Metodología:** Se definió una arquitectura y un diseño en la parte de hardware y en la parte lógica. En la parte lógica, se utilizó una representación del entorno basada en celdas de ocupación y una arquitectura de control fundada en comportamientos que fue implementada a través de un control *fuzzy*. **Resultados:** Se probó la plataforma robótica en escenarios estructurados como pasillos y salones. Su efectividad se comprobó mediante mapas construidos en un aplicativo desarrollado en Matlab. **Conclusiones:** Se demostró la capacidad de la plataforma robótica para construir mapas de su entorno de forma explícita, que, aun cuando pueden resultar más demandantes en lo computacional, permite una representación precisa y comprensible.

Palabras claves: Robótica móvil, mapeo, control basado en comportamientos, arquitectura de subsunción, control *fuzzy*, inteligencia artificial.

Abstract

Objective: To build a robotic system for exploration and mapping of indoor working environments with our own architecture and design methodology. **Methodology:** We defined an architecture and a design methodology for both the hardware and the logical levels. At the logical level, we used an environment representation based in occupancy cells and a behavior-based control architecture which was implemented through *fuzzy* control. **Results:** The robotic platform was tested in structured scenarios such as corridors and rooms. The effectivity of the platform could be verified using maps constructed with an application developed in Matlab. **Conclusions:** We demonstrated the ability of the robotic platform of constructing maps of its environment explicitly, which although could result computationally demanding, allows a precise and comprehensive representation.

Keywords: Mobile robotics, mapping, behavior-based control, subsumption architecture, *fuzzy* control, artificial intelligence.

Como citar (IEEE): G. Acosta-Amaya, A. Acosta-Gil., y J. Jiménez-Builes, "Sistema robótico autónomo para la exploración y construcción de mapas en entornos estructurados", *Investigación e Innovación en Ingenierías*, vol. 8, n°. 1, 2020. DOI: <https://doi.org/10.17081/invinno.8.1.3593>

Introducción

Los Robots Móviles Autónomos (RMA) deben resolver una amplia gama de problemas complejos sin intervención humana, como la navegación segura en entornos de trabajo cambiantes y la elaboración de representaciones computacionales de los sitios explorados [1].

En los últimos años, el paradigma de control basado en comportamientos, BBC (Behavior-Based Control), ha probado ser una poderosa herramienta para soportar procesos de navegación de agentes robóticos móviles en escenarios caracterizados por altos niveles de incertidumbre, imprecisión sensorial y falta de información [2, 3]. Este paradigma busca descomponer el problema de la navegación autónoma, no en funciones, sino en tareas simples de programar, depurar y mantener. Cada una de estas tareas recibe el nombre de comportamiento y puede ejecutarse de manera concurrente o no.

En la implementación de comportamientos básicos se han empleado diferentes técnicas de inteligencia artificial y sus combinaciones. Por ejemplo, en la implementación de comportamientos como evasión de obstáculos, seguimiento de paredes o búsqueda de espacio libre, se emplean sistemas de inferencia borrosos [3, 4]. Pandey et ál. [5] proponen un sistema neuro-borroso para la extracción automática de reglas y funciones de pertenencia; en tanto que Tahboub y Al-Din [6], presentan un método, también neurodifuso, en el que se ponderan las entradas proporcionadas por los sensores. En ambos casos se consigue una reducción significativa del número de reglas IF-THEN requeridas. Pero también es posible ajustar parámetros como funciones de pertenencia (Membership Functions MFs), factores de escala y reglas de los controladores borrosos (Fuzzy Logic Controllers FLCs) mediante algoritmos genéticos [7].

El mapeo de entornos constituye otro de los problemas más frecuentes que deben resolver los RMA. Este problema se refiere a la adquisición de información sensorial y la construcción de representaciones espaciales de entornos físicos a partir de la información sensorial adquirida por el robot durante la exploración del entorno [8, 9].

Ahora bien, los paradigmas predominantes en el mapeo robótico son los mapas topológicos y los mapas métricos. De este modo, mientras que en los modelos métricos se hace explícita la geometría del entorno, pudiéndose establecer relaciones métricas exactas entre los objetos encontrados; en las representaciones topológicas se busca establecer conexiones entre localizaciones reconocibles y distintivas del entorno [8, 9, 10].

El presente artículo describe justamente un sistema robótico para la exploración y el mapeo de entornos interiores de trabajo y está distribuido de la siguiente manera: en el siguiente apartado se presentan los materiales y métodos; luego, se expone la metodología de uso de la plataforma

robótica y, enseguida, los resultados y la discusión. Finalmente, se despliegan las conclusiones y referencias.

Metodología

Representación computacional del entorno y navegación

El mapeo de entornos, que corresponde al problema de la construcción y actualización de modelos computacionales, es un asunto complejo, pues exige considerar factores como: incertidumbre sensorial, condiciones desfavorables del entorno y limitaciones propias del alcance. Depende también, en alto grado, de otros problemas como la navegación, localización y percepción [9, 10, 11].

En el caso de entornos desconocidos, los RMA deben contar con la capacidad de elaborar, in situ, mapas del entorno explorado. A medida que el robot navega y construye un modelo, debe también determinar, de forma simultánea, su localización en tal representación. Cabe aclarar que en la literatura científica se denomina SLAM (Simultaneous Localization and Mapping) al problema del mapeo y localización simultánea de robots en su entorno [10, 12, 13].

En este trabajo, el entorno se modela mediante una representación métrica conocida como mapa de celdas de ocupación (Occupancy Grid, OC). Inicialmente propuestas por Kim et ál. [11], las representaciones por OC discretizan el entorno mediante una malla de celdas uniformes o retícula, en la que cada celda representa una pequeña parte del área de trabajo. A cada celda se asigna una o más propiedades que indican si esa porción del espacio se encuentra o no ocupada [10, 14], por lo que los mapas de celdas constituyen una herramienta de modelado probabilístico para la representación de obstáculos en el entorno de trabajo.

Los mapas de celdas reproducen de manera explícita la estructura del entorno. Por esta razón, resulta relativamente fácil determinar la localización del robot en el modelo a partir de su localización en el mundo físico [15]. Precisamente, esta correspondencia geométrica del modelo con el entorno hace de los mapas de celdas descripciones fáciles de construir y mantener, pero, también, muy sensibles a errores odométricos acumulativos.

La principal desventaja de los mapas de celdas es que son modelos muy demandantes de recursos de memoria cuando se quieren representar espacios de grandes dimensiones.

De otro lado, las arquitecturas de control constituyen una guía para dar respuesta al problema de navegación segura, tanto en entornos estáticos como dinámicos. Estas arquitecturas permiten describir cada componente del sistema, su organización y los mecanismos de interacción entre ellos [16, 17].

Obedeciendo a las relaciones que se pueden establecer entre las componentes de percepción, planificación y actuación, las arquitecturas de control se clasifican en deliberativas, reactivas e híbridas [17].

Para los años 80, el paradigma predominante era el deliberativo, centrado en los procesos de modelamiento y planificación. En esta arquitectura el sistema de percepción provee información del entorno y del estado del robot, con la que se construye o actualiza un modelo interno del mundo. Una vez se dispone de tal modelo, el sistema de planificación elabora un plan de las futuras acciones que ejecutará el robot [17, 18].

A mediados de los 80, Brooks propuso eliminar el sistema de planificación, componente principal de las arquitecturas deliberativas, privilegiando una respuesta reactiva del robot a los estímulos del entorno y consiguiendo así una significativa reducción del elevado costo computacional asociado a la planificación [18].

Metodología de uso de la plataforma robótica en la exploración del entorno

Para la elaboración de mapas de entornos interiores se construyó el sistema Map-Bot, el cual integra módulos de hardware y software para la navegación, adquisición de información, transferencia de datos y elaboración de mapas. Al sistema se le incorpora un computador en el que corre el software responsable de la construcción de representaciones bidimensionales del entorno, un agente robótico autónomo que recoge información sensorial del sitio visitado y un módulo de comunicaciones inalámbricas para la transferencia de datos entre el computador y el robot [12].

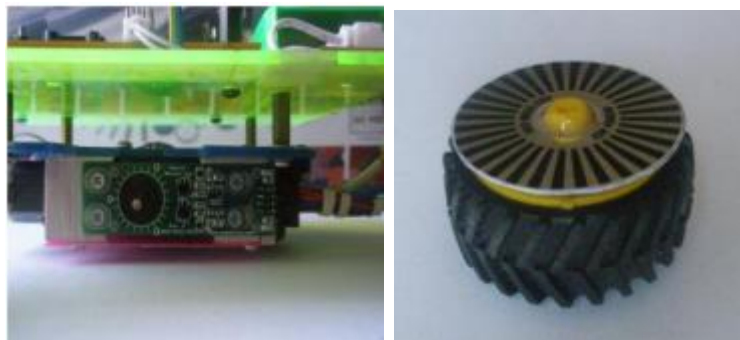
La exploración del entorno la lleva a cabo el robot autónomo Walling, que dispone de subsistemas de percepción, control y actuación que le permiten explorar su entorno de trabajo de forma segura, evitando obstáculos a su alrededor. Durante la exploración, el robot recaba y transmite información de distancias a objetos circundantes, como paredes, sillas y mesas, además de la longitud de los trayectos recorridos. El robot cuenta con un sistema de efectores constituido por una caja motorreductora doble con dos motores eléctricos DC, dos ruedas de tracción en configuración diferencial y una rueda de soporte tipo caster ball [12].

Las funciones de control de navegación, captura y procesamiento de datos sensoriales son llevadas a cabo por un microcontrolador Coldfire MCF51QE128 de 32 bits, que se ejecuta a una frecuencia interna de bus de 26,6 MHz.

El sistema de percepción exteroceptiva y de comunicaciones incorpora tres módulos de ultrasonido SRF02 que entregan datos de distancia sobre obstáculos localizados al frente y al costado derecho del robot. Para la transmisión y recepción de información se emplean módulos de comunicaciones inalámbricas XBee que soporta el protocolo IEEE 802.15.4. En síntesis, la plataforma robótica de exploración fue diseñada según criterios de modularidad, potencia computacional y una sensorial básica. Sensorial propioceptiva del robot Walling

La elaboración automática de mapas del entorno requiere robots dotados de dispositivos de percepción que permitan determinar, no solo las distancias a los obstáculos circundantes, sino también los trayectos lineales que recorre y su orientación relativa a un sistema de coordenadas [12]. Las distancias recorridas por el robot durante la exploración del entorno, se determinan con base en un sistema odométrico que emplea una fuente de luz infrarroja y un disco reflectivo para generar un tren de pulsos eléctricos. Como el perímetro de las ruedas de tracción es de 11.62 cm, resulta fácil determinar el recorrido lineal que transita el robot al contar los pulsos que genera el sistema (ver Figura 1).

Figura 1. Encoder anclado al chasis del robot y concéntrico al eje del motor y su respectivo disco reflectivo acoplado a una rueda para el sistema odométrico

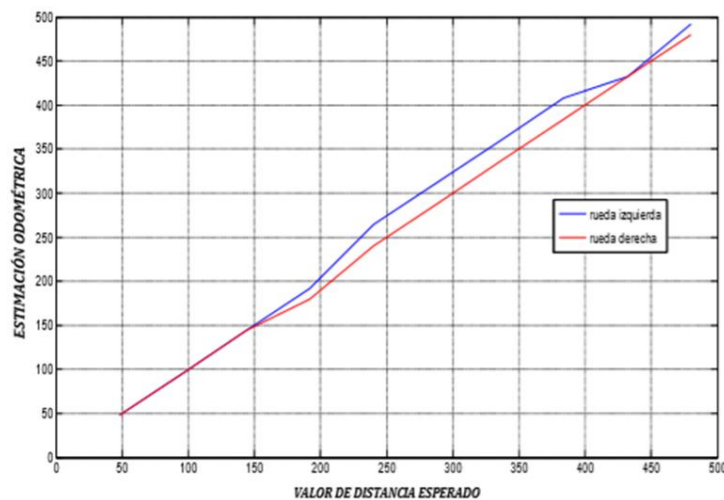


Fuente: Elaboración propia

De otra parte, en la Figura 2 se presentan los resultados de las pruebas experimentales efectuadas al sistema odométrico del robot. Se tomaron lecturas de recorridos lineales realizados por la rueda derecha e izquierda para trayectos de navegación de 480 cm. Durante las pruebas, un algoritmo de control de seguimiento de pared mantuvo al robot a una distancia

cercana a 50 cm (set-point) respecto a la pared derecha. La desviación respecto al comportamiento puramente lineal esperado (modelo teórico) se debe a fluctuaciones alrededor del set-point provocadas por la incertidumbre sensorial asociada a los sensores de ultrasonido y la odometría. Así mismo, se aprecia una mayor desviación sobre la rueda izquierda que sobre la derecha, debido a que el esquema de control de navegación efectúa correcciones sobre el efector derecho y no sobre el izquierdo. La rueda izquierda “seguirá” siempre a la derecha.

Figura 2. Prueba de estimación odométrica para trayectos de navegación de 480 cm



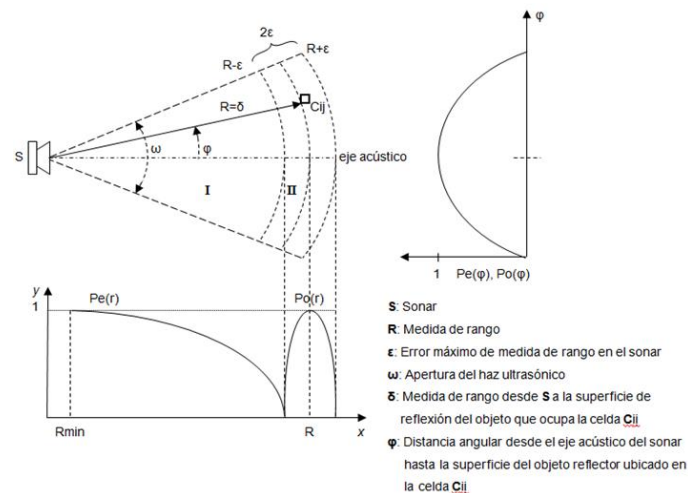
Fuente: Elaboración propia

Caracterización de los sensores exteroceptivos del robot Walling

El robot autónomo cuenta con un sistema de sonares dispuestos ortogonalmente sobre una placa de circuito impreso. Esta disposición permite obtener medidas de distancia a obstáculos localizados en su entorno próximo. A partir de estos datos, el módulo de control genera acciones que permiten la navegación segura del robot en su entorno. La sensoria exteroceptiva del robot Walling se implementó con base en el sonar SRF02, un dispositivo ultrasónico que opera a una frecuencia de 40 KHz. La transferencia de comandos y datos entre el sonar y el procesador central se realiza a través de una interfaz UART con formato de trama de 9600, 8, N, 1 y niveles TTL. Los sensores ultrasónicos proveen información de una superficie o volumen y no de un punto particular del entorno y operan de la siguiente forma: un pulso acústico de frecuencia f , emitido por el sonar, alcanza un obstáculo y devuelve un eco. Con la medida del tiempo de vuelo (Time Of Fly, TOF) de la señal acústica, desde la emisión del pulso hasta la recepción del primer eco, se obtiene, a su vez, una medida de distancia desde el sonar hasta el obstáculo. Las medidas de rango que

resultan de ello no son suficientes para la construcción de mapas [14, 19]. Se debe contar al respecto con un modelo en el que se tengan en cuenta las incertidumbres propias del sistema de percepción, la manera en que se interpretan las medidas de rango y la forma como se trasladan estas lecturas a valores para indicar el grado de ocupación de las celdas C_{ij} en el mapa. En este trabajo se emplea el modelo bidimensional gaussiano (2Dgauss) para el sonar (ver Figura 3). Debido a la gran apertura angular del lóbulo principal de emisión del transductor, las medidas de rango proporcionan información indirecta del grado de ocupación de las celdas ubicadas en el campo de visión del sensor. El obstáculo que devuelve el eco puede localizarse en cualquier lugar al interior de la región semicircular comprendida entre $[R-\epsilon, R+\epsilon]$ y $[-\omega/2, \omega/2]$ que corresponde a la región II en la Figura 3. Se asume entonces que no hay obstáculos localizados en la región I, es decir, a una distancia menor a $R-\epsilon$ [12].

Figura 3. Modelo bidimensional gaussiano (2Dgauss) para el sonar SRF02



Fuente: Elaboración propia

Actualización del espacio libre mediante el modelo 2Dgauss

Para las celdas localizadas en la región I de la Figura 3, el valor de probabilidad de encontrarse libres de obstáculos se calcula mediante [12]:

$$P_E(x, y) = E_r(\delta) * E_a(\phi) \quad (1)$$

Donde:

$P_E(x, y)$: Probabilidad de que la celda C_{ij} , localizada en las coordenadas (x, y) , se encuentre vacía.

$E_r(\delta)$: Estimación de celda vacía con base en la medida de rango.

$E_a(\phi)$: Estimación de celda vacía con base en su distancia angular al eje acústico del sensor.

Además, los valores de $E_r(\delta)$ y $E_a(\phi)$ se calculan mediante las ecuaciones (2) y (3) respectivamente.

$$E_r(\delta) = \begin{cases} 1 - ((\delta - R_{min}) / (R - \varepsilon - R_{min}))^2, & \text{para } R_{min} \leq \delta \leq R - \varepsilon \\ 0, & \text{en otro caso} \end{cases} \quad (2)$$

$$E_a(\phi) = 1 - \left(\frac{2\phi}{\omega}\right)^2, \quad \text{para } \frac{\omega}{2} \leq \phi \leq \frac{\omega}{2} \quad (3)$$

Actualización del espacio ocupado mediante el modelo 2Dgauss

Las celdas localizadas en la región II de la Figura 3 presentan una probabilidad de encontrarse ocupadas dada por [12]:

$$P_o(x, y) = O_r(\delta) * O_a(\phi) \quad (4)$$

Donde:

$P_o(x, y)$: Probabilidad de que la celda Cij, localizada en las coordenadas (x, y), se encuentre ocupada (occupied).

$O_r(\delta)$: Estimación de celda ocupada con base en la medida de rango.

$O_a(\phi)$: Estimación de celda ocupada con base en su distancia angular al eje acústico del sensor.

Las ecuaciones (5) y (6) permiten calcular los valores $O_r(\delta)$ y $O_a(\phi)$.

$$O_r(\delta) = 1 - ((\delta - R) / \varepsilon)^2, \quad \text{para } R - \varepsilon \leq \delta \leq R + \varepsilon \quad (5)$$

$$O_a(\phi) = 1 - \left(\frac{2\phi}{\omega}\right)^2, \quad \text{para } -\frac{\omega}{2} \leq \phi \leq \frac{\omega}{2} \quad (6)$$

Comunicaciones

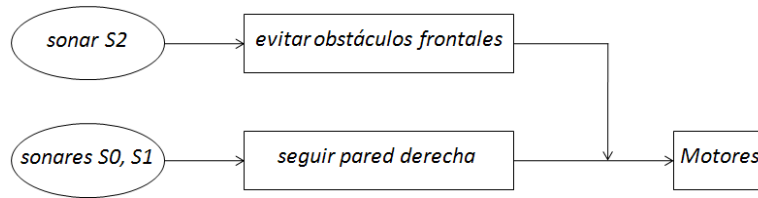
Para la transferencia bidireccional de datos entre el computador y el robot, se diseñó y construyó una pequeña estación base (Base Station) cuyo componente principal es el módulo electrónico Xbee, con el cual se consigue implementar un enlace punto a punto que cumple con el estándar IEEE 802.15.4. La estación base permite mantener la integridad de datos a distancias de hasta 30 metros en entornos interiores [12].

Arquitectura de control

En robótica móvil una arquitectura de control corresponde a una estructura de hardware y software con la que se adquiere y procesa información sensorial del entorno, a fin de llevar a cabo funciones cognitivas y generar señales para el control de efectores. En tal sentido, la arquitectura representa, organiza e interconecta la percepción, el razonamiento y la actuación [9]. En la Figura 4 se aprecia la arquitectura de control implementada para la plataforma robótica Walling, que se ajusta al modelo

de subsunción de Brooks y que incorpora dos comportamientos para la navegación [12].

Figura 4. Arquitectura de subsunción del robot Walling



Fuente: Elaboración propia

El comportamiento “Evitar obstáculos frontales” corresponde a un controlador duro que se activa cuando el sensor S2 detecta un obstáculo a una distancia no segura y subsume el comportamiento “Seguir pared derecha”. Una vez se detecta un obstáculo a una distancia menor o igual a 50 cm, el módulo de control procesa la medida y genera acciones de control de acuerdo con el algoritmo 1 (ver Tabla 1).

Tabla 1. Algoritmo 1 Implementación del comportamiento evitar obstáculos frontales [12]

1: While(distancia_obstáculo_frontal (sonar S2) \leq 50 cm)
2: Detener motores
3: While(desplazamiento_angular < 90°)
4: Activar motor derecho
5: End
6: End
7: Ceder control de navegación al comportamiento seguir pared derecha

Fuente: Elaboración propia

El comportamiento de menor jerarquía “Seguir pared derecha” se implementó mediante un sistema de inferencia fuzzy (Fuzzy Inference System FIS) que recibe dos medidas de rango, una del sensor S0 y otra del S1.

Para este FIS se definieron dos variables lingüísticas de entrada (e_dilad0, e_dilad1), una variable de salida (Pulse Width Modulation PWM), los universos del discurso para cada variable, las particiones en conjuntos borrosos con sus correspondientes funciones de pertenencia y el conjunto de reglas. El controlador fuzzy implementado permite mantener una distancia de alrededor de 50 cm con relación a objetos localizados al costado derecho del robot.

El FIS modifica el ciclo de trabajo (Duty Cycle) de la señal PWM que se aplica al efector derecho. Para el efector izquierdo se mantiene constante el valor

PWM inicial. El algoritmo 2 corresponde al FIS empleado para implementar el comportamiento “Seguir pared derecha”.

Tabla 2. Algoritmo 2 FIS para la implementación del comportamiento “Seguir pared derecha” [12].

1: while (distancia_obstáculo_fronal (sonar S2) > 50 cm)
2: Obtener medidas de distancia m0 y m1 de los sonares S0 y S1
Respectivamente
3: Determinar los errores de distancia:
$e_{dilad0}=r-m0$
$e_{dilad1}=r-m1$
4: Fuzificación: conversión de los valores precisos (crisp) que adoptan las variables de entrada $e_{dilade0}$ y e_{dilad1} en valores difusos
5: Inferencia Lógica: aplicación de reglas lingüísticas IF-THEN para determinar el valor difuso de salida a partir de los valores difusos de entrada
6: Defuzificación: procedimiento de cálculo para determinar valores crisp de salida a partir de un valor fuzzy de entrada, el método empleado es el del singleton
7: Actualizar ciclo de trabajo (Duty Cycle) del PWM del motor derecho
8: End

Fuente: Elaboración propia

Resultados y discusión

Para la realización de las pruebas de navegación y mapeo, se seleccionaron espacios con características típicas de entornos estructurados de trabajo, con dimensiones ajustadas al entorno virtual de trabajo definido en el aplicativo de software diseñado para el mapeo. Los espacios de prueba seleccionados incorporaron una o varias de las siguientes características: muros de obra con o sin acabado enlucido o empapelado, columnas de hormigón de sección transversal cualquiera y puertas de aglomerado con acabado en railite o fórmica.

La captura de datos se llevó a cabo en condiciones normales de presión y temperatura, cuyos valores de referencia son 1 atm y 20oC, respectivamente. Para la exploración del entorno se programa una velocidad lineal promedio de 12cm/s. Durante el recorrido el robot transfiere dos distancias laterales (pared derecha), una distancia frontal y la distancia lineal que recorre cada rueda. La toma de datos se hace a razón de diez muestras por segundo.

Puesto que todo sensor incorpora incertidumbre en la medida, es necesario establecer un parámetro para cuantificar el grado de aproximación de la representación computacional obtenida con las dimensiones reales del entorno físico. Para tal fin, se define el parámetro Nivel de Aproximación del Mapa al Entorno Real (NAMER) dado por [12]:

$$NAMER = \frac{D_{map}}{D_{real}} \quad (7)$$

Donde:

D_{map} : Distancia calculada sobre el mapa.

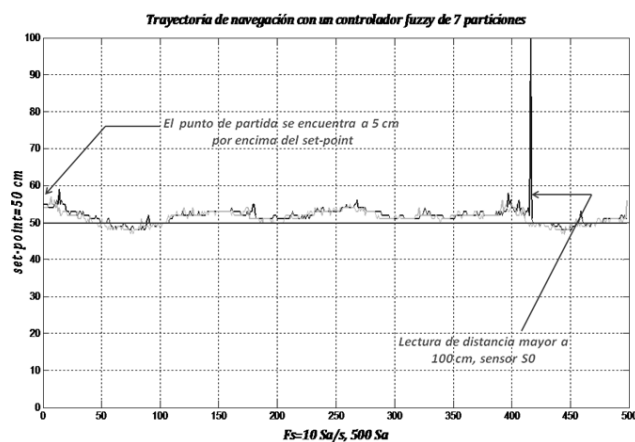
D_{real} : Distancia real medida sobre el entorno.

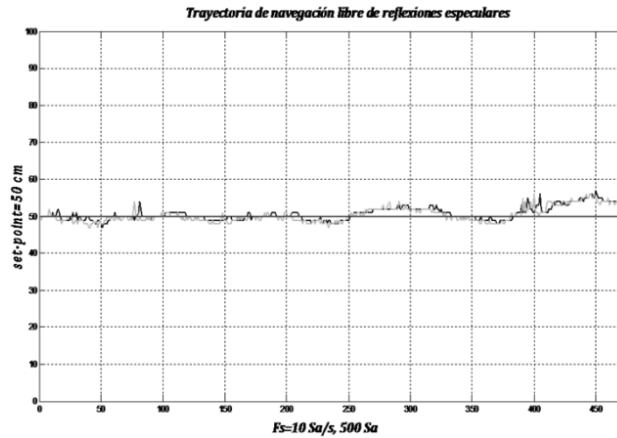
Mientras más cercano se encuentre este parámetro a la unidad, mayor será el grado de correspondencia entre la distancia real y la distancia virtual calculada sobre el mapa. Valores inferiores a la unidad indican una distancia menor a la distancia real, y valores superiores, una distancia mayor.

Navegación

Inicialmente se llevaron a cabo pruebas de navegación para el comportamiento “Seguir pared derecha” empleando un solo sonar. Los recorridos realizados presentaron fuertes oscilaciones alrededor del set-point de 50 cm. A fin de corregir estas oscilaciones, se agregó un segundo sonar, con lo que se mejoró el desempeño en la navegación. Este segundo esquema con redundancia en la medida permitió reducir también la incertidumbre en la medida por reflexiones especulares en los sonares. Así, mientras que uno de los sensores puede estar suministrando una lectura errónea, el otro puede proporcionar una lectura correcta, con lo que se logra mantener el control de navegación. Para reducir aún más el impacto negativo provocado por el fenómeno de especularidad, se incorporó en el módulo de navegación un control de saturación por software. En la Figura 5 se observan curvas experimentales de navegación, una que presenta reflexión especular y otra libre de este fenómeno [12].

Figura 5. Pruebas de navegación con recorridos de 560 cm. (con reflexión especular en el sensor S0 y con recorrido libre de reflexiones)





Fuente: Elaboración propia

Mapeo

Se consideraron dos entornos típicos de interiores como oficinas, laboratorios, pasillos y aulas. A continuación, se presentan pruebas para dos de los entornos considerados.

Entorno 1: área de oficinas con dos paredes y una puerta. En esta prueba se consideró un entorno interior de oficinas con dos paredes de ladrillo separadas por una puerta en madera con revestimiento de fórmica (Figura 6). En el experimento, el agente explorador realizó un recorrido de 237 centímetros, desde donde adquirió y transmitió un total de 1220 datos de distancia (ultrasonido y odometría) a una frecuencia de muestreo de 10 Hz [12].

Figura 6. Mapeo de un entorno de oficinas



Fuente: Elaboración propia

En la Tabla 3 se aprecian los valores de NAMER y el cálculo de algunas distancias representativas: distancia de la trayectoria seguida por el robot respecto al muro (d_1), profundidad a la que se encuentra la puerta respecto al muro (d_2) y distancia de apertura para la puerta (d_3).

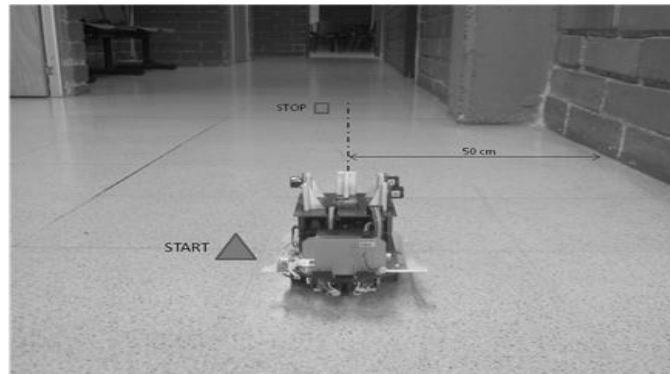
Tabla 3. Valores NAMER y cálculo de distancias para el entorno 1

	d1	d2	d3
Distancia Medida en el Mapa (D_{map}), en cm	45	85	105
Distancia Real (D_{real}), en cm	50	95	104
NAMER (parámetro adimensional)	0.9	0.895	1.01
<i>Promedio NAMER: 0.935</i>			
Cálculo de distancias en el mapa con celdas de 5 cm:			
d1=9 celdas x 5 cm/celda=45 cm			
d2=17 celdas x 5 cm/celda= 85 cm			
d3=21 celdas x 5 cm/celda=105 cm			

Fuente: Elaboración propia

Entorno 2: pasillo con dos paredes y una columna. Este experimento se llevó a cabo en un entorno de pasillo con dos paredes no colineales en ladrillo y entre las cuales se encuentra una columna de hormigón con sección transversal rectangular. El trayecto recorrido por el robot explorador fue de 250 cm y se adquirieron 1060 muestras a una frecuencia de 10 Hz. La Figura 7 registra el entorno físico en el que se realizó la prueba con resolución de celda de 5 cm [12].

Figura 7. Mapeo de un tramo de pasillo con paredes no colineales y columna



Fuente: Elaboración propia

En la Tabla 4 se pueden apreciar los valores NAMER para el mapa de la Figura 7. Se consideraron las distancias de referencia: distancia entre la cara de la columna que da al pasillo y la pared del lado derecho del punto de start (d1), distancia de la trayectoria seguida por el robot respecto al muro (d2) y ancho de la cara de la columna de sección transversal rectangular que da al pasillo (d3).

Tabla 4. Cálculo del NAMER para el mapa de la Figura 7

	d1	d2	d3
Distancia Medida en el Mapa (D_{map}), en cm	20	45	50
Distancia Real (D_{real}), en cm	25	50	40
NAMER (parámetro adimensional)	0.8	0.9	1.25
<i>Promedio NAMER: 0.983</i>			
Cálculo de distancias en el mapa con celdas de 5 cm:			
d1=4 celdas x 5 cm/celda=20 cm			
d2=9 celdas x 5 cm/celda= 45 cm			
d3=10 celdas x 5 cm/celda=50 cm			

Conclusiones

Después de analizar los paradigmas predominantes en la construcción de representaciones computacionales del entorno, se decidió emplear la técnica de mapeo por celdas de ocupación. Si bien esta resulta ser más demandante en recursos computacionales que las representaciones topológicas, también reproduce de manera explícita la estructura del entorno, lo que facilita la localización del robot en el modelo a partir de su localización en el mundo físico.

La combinación de un controlador fuzzy con un controlador duro, en lugar de los esquemas tradicionales de control, permitió obtener una navegación segura sin necesidad de disponer de modelos matemáticos del robot, validándose de esta forma el potencial de los métodos intuitivos en el diseño de controladores.

El sistema fue puesto a prueba en diferentes tipos de entornos interiores: pasillos, oficinas, laboratorios y aulas de clase. Los modelos obtenidos se ajustaron a las características morfológicas de los entornos de prueba y a sus características métricas.

Agradecimientos

Los autores expresan su agradecimiento a los ingenieros en Instrumentación y control Alejandra Aristizábal y Christian Toro, por el soporte brindado en la realización de algunas pruebas experimentales y en la codificación de rutinas de mapeo en Matlab.

Referencias bibliográficas

1. J. Orsulić, D. Miklić., and Z. Kovačić, "Efficient Dense Frontier Detection for 2-D Graph SLAM Based on Occupancy Grid Submaps", *IEEE Robotics and Automation Letters*, vol. 4, n°. 4, pp. 3569-3576, 2019.
2. J. Huang, N. Zhou., y M. Cao, "Adaptive fuzzy behavioral control of second-order autonomous agents with prioritized missions: Theory and experiments", *IEEE Transactions on Industrial Electronics*, vol. 66, n°. 12, pp. 9612-9622, 2019.
3. A. Fatmi, A.A. Yahmadi, L. Khriji., and N. Masmoudi, "A Fuzzy Logic Navigation of a Mobile Robot", *Proceedings of World Academy of Science and Technology*, Vol. 15, pp. 255-260, October 2008.
4. S. Islam, M. Zaman, B. Madon., and M. Othman, "Designing Fuzzy Based Mobile Robot Controller using VHDL", *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 2, pp. 138-142, 2008.
5. A. Pandey, A.K. Kashyap, D.R. Parhi., and B.K. Patle, "Autonomous mobile robot navigation between static and dynamic obstacles using multiple ANFIS architecture", *World Journal of Engineering*, vol. 16l, n°. 2, pp. 275-286, 2019.
6. K. K. Tahboub., and M. S. N. Al-Din, "A Neuro-Fuzzy Reasoning System for Mobile Robot Navigation", *Jordan Journal of Mechanical and Industrial Engineering*, vol. 3, No. 1, pp. 77-88, 2009.
7. S. F. Desouky, H.M. Schwartz, "Generic Based Fuzzy Logic Controller for a Wall-Following Mobile Robots", *American Control Conference ACC 2009, St. Louis Missouri, USA*, 2009.
8. L. Delobel, R. Aufrere, C. Debain, R. Chapuis., and T. Chateau, "A Real-Time Map Refinement Method Using a Multi-Sensor Localization Framework", *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, n°. 5, pp. 1644-1658, 2019.
9. M. Emharraf, M. Bourhaleb, M. Saber., and M. Rahmoun, "Mobile robot: SLAM implementation for unknown indoor environment exploration", *Journal of Computer Science*, vol. 2, n°. 2, pp. 106-112, 2019.
10. M.K. Habib, "Real Time Mapping and Dynamic Navigation for Mobile Robots", *International Journal of Advanced Robotic Systems*, Vol. 4, No. 3, pp. 323-338, 2007.
11. B. Kim, H. Cho, H. Joe., and S.C. Yu, "Optimal strategy for seabed 3D mapping of AUV based on imaging sonar", *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans, OCEANS - Kobe*, 2018.

12. G. Acosta. "Ambiente Multi-agente Robótico para la Navegación Colaborativa en Escenarios Estructurados". M. Sc. Thesis, Universidad Nacional de Colombia, Medellín, Colombia, 2010.
13. G. Acosta. "SLAM monocular en tiempo real". Tesis de doctorado, Universidad Nacional de Colombia, 2019.
14. C. West, F. Arvin, W. Cheah, M. Giuliani, and B. Lennox, "A Debris Clearance Robot for Extreme Environments". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11649 LNAI, pp. 148-159, 2019.
15. X.Z. Li, Y. Gong, X.Y. Zhang, S.M. Jia., and X.N. Liang, "An autonomous exploration method for an indoor mobile robot". *Kongzhi y Juece/Control and Decision*. Vol. 34, Issue. 6, 1 June pp.1227-1233, 2019,
16. M. Imre, E. Oztop, Y. Nagai. and E. Ugur, "Affordance-based altruistic robotic architecture for human–robot collaboration", *Adaptive Behavior*, vol. 27, n°. 4, pp. 223-241, 2019.
17. D. Nakhaeinia, S. H. Tang, S. B. Mohd Noor, O. Motlagh, "A Review of Control Architectures for Autonomous Navigation of Mobile Robots", *International Journal of the Physical Sciences*, Vol. 6, n°. 2, pp. 169-174, 2011,
18. R. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, 1986, pp.14-23.
19. V.S. Kalogeiton, K. Ioannidis, G.C. Sirakoulis., and E.B. Kosmatopoulos, "Real-Time Active SLAM and Obstacle Avoidance for an Autonomous Robot Based on Stereo Vision", *Cybernetics and Systems*, vol. 50, n°, 3, pp. 239-260, 2019.