Revista de Ciencias Tecnológicas (RECIT). Universidad Autónoma de Baja California ISSN 2594-1925
Volumen 9 (3): e465. Julio-Septiembre, 2026. https://doi.org/10.37636/recit.v9n3e465

Artículo de investigación

Sistema de mapeo y planificación de rutas con fusión multisensorial para navegación autónoma de un robot móvil tipo uniciclo

Multi-sensor fusion-based mapping and path planning system for autonomous navigation of a unicycle-type mobile robot

Gilberto Ramos, Luis Arturo García Delgado, Ricardo Ramón Pérez Alcocer

Universidad de Sonora, Blvd. Luis Encinas J, Calle Av. Rosales &, Centro, 83000 Hermosillo, Sonora, México

Autor de correspondencia: Gilberto Ramos, Universidad de Sonora, Blvd. Luis Encinas J, Calle Av. Rosales &, Centro, 83000 Hermosillo, Sonora, México. Correo electrónico: brtdevkit@gmail.com. ORCID: 0009-0001-4388-733X.

Recibido: 15 de abril del 2026 **Aceptado:** 4 de agosto del 2026 **Publicado:** 14 de agosto del 2026

Resumen. - *En este trabajo se presenta un sistema de mapeo y planificación de rutas para un robot móvil con ruedas tipo uniciclo (WMR, Wheeled Mobile Robot), implementado sobre la plataforma AmigoBot, con el objetivo de permitir la navegación autónoma desde una posición inicial hasta un destino definido en un entorno estructurado. El sistema se organiza en tres componentes principales: generación y visualización de mapas de ocupación, planificación de trayectorias y control de movimiento con evasión de obstáculos. La implementación se realizó utilizando el Sistema Operativo para Robots (ROS, Robot Operating System) y la arquitectura ARIA (Advanced Robotics Interface for Applications). La representación del entorno se construye mediante la fusión de datos provenientes de un arreglo de sensores ultrasónicos y una cámara de profundidad Kinect. La fusión se realiza mediante la intersección de mapas binarios, complementada con un filtrado espacial basado en vecindad para suavizar inconsistencias locales en la rejilla de ocupación. La planificación de rutas se basa en el algoritmo* **A***, *seguido de una etapa de simplificación geométrica mediante trazado de rayos (ray casting), con el objetivo de reducir la cantidad de nodos en la trayectoria. El seguimiento de la ruta se implementa mediante un controlador proporcional de error de postura (posición y orientación), incorporando un esquema de evasión de obstáculos reactivo basado en campos potenciales repulsivos. Los resultados experimentales muestran que el sistema logra una reducción del 92 % en el número de nodos de la ruta original y del 62.5 % respecto a sus puntos de inflexión gracias al proceso de simplificación geométrica. En las pruebas de navegación autónoma en tiempo real, el robot completó exitosamente la trayectoria en un tiempo de 1 minuto y 32 segundos, manteniendo el error de seguimiento respecto a los puntos de control acotado por debajo de 0.25 m durante las transiciones. Ante la presencia de obstáculos adicionales no cartografiados, el módulo reactivo garantizó una tasa de éxito del 100 % en la evasión de colisiones, registrando una distancia mínima de seguridad de 3.6 cm en las regiones con mayor restricción espacial del entorno. Estos resultados evidencian que la arquitectura de software propuesta es capaz de generar mapas de ocupación consistentes, planificar trayectorias eficientes con baja complejidad geométrica y ejecutar navegación autónoma segura en el entorno experimental.*

Palabras clave: Planificación de rutas; WMR; A*; Ray casting; Mapas de ocupación.

Abstract. – *This paper presents a mapping and path planning system for a unicycle-type wheeled mobile robot (WMR), implemented on the AmigoBot platform, aimed at enabling autonomous navigation from a starting position to a defined destination within a structured environment. The system comprises three main components: occupancy map generation and visualization, trajectory planning, and motion control with obstacle avoidance. Implementation was carried out using the Robot Operating System (ROS) and the Advanced Robotics Interface for Applications (ARIA) architecture. The environmental representation is constructed by fusing data from an ultrasonic sensor array and a Kinect depth camera. Data fusion is achieved through the intersection of binary maps, complemented by neighborhood-based spatial filtering to smooth out local inconsistencies in the occupancy grid. Path planning relies on the A* algorithm, followed by a geometric simplification stage using ray casting to reduce the number of nodes in the trajectory. Path following is implemented via a proportional controller based on pose error (position and orientation), incorporating a reactive obstacle avoidance scheme based on repulsive potential fields. Experimental results demonstrate that the system achieves a 92% reduction in the number of nodes from the original path and a 62.5% reduction in inflection points, thanks to the geometric simplification process. In real-time autonomous navigation tests, the robot successfully completed the trajectory in 1 minute and 32 seconds, maintaining the tracking error relative to control points below 0.25 m during transitions. In the presence of additional unmapped obstacles, the reactive module ensured a 100% success rate in collision avoidance, recording a minimum safety distance of 3.6 cm in the environment's most spatially constrained areas. These results demonstrate that the proposed software architecture is capable of generating consistent occupancy maps, planning efficient trajectories with low geometric complexity, and executing safe autonomous navigation within the experimental environment.*

Keywords: Path planning; WMR; A*; Ray casting; Occupancy maps.

ISSN: 2594-1925

1. Introducción

Los vehículos autónomos han adquirido una relevancia significativa en las últimas décadas debido a su creciente aplicación en diversos ámbitos, incluyendo sistemas de asistencia doméstica como robots aspiradores, transporte autónomo de personas y aplicaciones industriales en entornos logísticos y de almacenamiento, como centros de distribución automatizados. Asimismo, en el ámbito de exploración y rescate, estos sistemas son empleados en entornos peligrosos o de difícil acceso para el ser humano.

Para que un robot sea considerado autónomo, debe ser capaz de percibir su entorno, tomar decisiones y ejecutar acciones de navegación sin intervención humana directa. El desempeño en tareas de navegación depende directamente de la calidad de la información sensorial disponible y de la capacidad del sistema para estimar su pose, entendida como la posición y orientación del robot dentro del entorno operativo [1].

A partir de estas necesidades surge el problema de localización y mapeo simultáneo, conocido como *SLAM* (*Simultaneous Localization and Mapping*). Este problema consiste en construir un mapa del entorno mientras se estima simultáneamente la pose del robot dentro de dicho mapa [2]. Sus aplicaciones son particularmente relevantes en entornos sin acceso a sistemas de posicionamiento global, tales como interiores, minas o ambientes subacuáticos [3]–[7]. El problema de *SLAM* puede descomponerse en submódulos como la adquisición de datos sensoriales, la representación del entorno, la estimación de la pose y la planificación de rutas [8]–[11].

Sin embargo, la implementación de arquitecturas de navegación autónoma en plataformas robóticas experimentales presenta desafíos asociados a restricciones de integración de hardware y software, disponibilidad limitada de documentación técnica y dependencia de herramientas de software heredadas o con soporte limitado. Estas condiciones son comunes en plataformas de investigación o educativas basadas en hardware descontinuado, lo que dificulta la implementación directa de soluciones modernas de navegación.

En este contexto, el presente trabajo propone una arquitectura de navegación autónoma basada en el Sistema Operativo para Robots (*ROS, Robot Operating System*), un *framework* de software de código abierto ampliamente utilizado en robótica para la gestión coordinada de percepción, control y planificación. Este sistema se utiliza para la integración de percepción sensorial, generación de mapas en rejilla, planificación de trayectorias y control de movimiento, implementado sobre el robot móvil tipo uniciclo (*WMR, Wheeled Mobile Robot*) AmigoBot™ desarrollado por Adept MobileRobots Inc.

La principal contribución de este trabajo es el desarrollo de un sistema de *software* de visualización y planificación de rutas que integra la construcción del mapa, su representación en rejilla y la selección interactiva de objetivos de navegación dentro de un entorno gráfico intuitivo. A diferencia de herramientas de visualización estándar como *RViz*, que se enfocan principalmente en la visualización de información del sistema, la plataforma desarrollada permite la interacción directa con el mapa para la definición de metas de navegación y la generación de trayectorias planificadas dentro del mismo entorno.

Adicionalmente, el sistema cuenta con una arquitectura modular que permite la incorporación de distintos algoritmos de planificación de trayectorias y control de movimiento. En este trabajo se implementa una estrategia de control proporcional, aunque la estructura propuesta facilita su extensión a otros métodos de control y su adaptación a diferentes plataformas robóticas.

ISSN: 2594-1925

Finalmente, debido a las limitaciones de documentación y soporte del *hardware* utilizado, correspondiente a un robot móvil diferencial AmigoBot (modelo 2004, basado en el microcontrolador Renesas H8S) con software heredado (*legacy*), se desarrolló un sistema de percepción y adquisición de datos que permite su integración en un entorno moderno basado en ROS, habilitando su operación dentro del flujo de navegación propuesto.

El documento se organiza de la siguiente manera. En la Sección 2 se presentan los antecedentes relevantes. La Sección 3 describe la metodología propuesta, incluyendo el modelo cinemático del robot, el esquema de generación de mapas y el algoritmo de planificación de trayectorias. La Sección 4 presenta la arquitectura del sistema y el esquema de control utilizado para la validación experimental. La Sección 5 expone los resultados obtenidos y, finalmente, la Sección 6 presenta las conclusiones y trabajo futuro.

2. Antecedentes

En robótica móvil se emplea una amplia variedad de sensores para la percepción y representación del entorno, entre los que destacan sensores ultrasónicos, sistemas LiDAR y cámaras de profundidad [12]–[16]. A partir de estos dispositivos es posible obtener información espacial del entorno, la cual puede procesarse para construir modelos virtuales que permitan la ejecución de tareas de navegación autónoma.

Entre las representaciones más utilizadas para modelar el entorno se encuentran los mapas métricos, topológicos, basados en características y los mapas de ocupación [17]–[21]. Dentro de estas alternativas, los mapas de ocupación constituyen una solución ampliamente adoptada debido a su representación discreta del entorno mediante rejillas bidimensionales, en las que cada celda codifica el estado de ocupación de una región del espacio [22]. Esta representación ofrece un equilibrio adecuado entre simplicidad computacional, capacidad descriptiva y facilidad de integración con algoritmos de planificación.

Una vez obtenida la representación del entorno, la navegación requiere el uso de algoritmos de planificación de rutas que permitan determinar trayectorias seguras entre una posición inicial y un destino. Entre los métodos clásicos se encuentra el algoritmo de *Dijkstra* [23], del cual derivan enfoques heurísticos más eficientes como el algoritmo *A** (*A-star*) [24]. También existen métodos como *D* Lite* [25], orientados a la replanificación en entornos dinámicos, así como técnicas de exploración basadas en muestreo como los árboles aleatorios de exploración rápida (*RRT*, *Rapidly-Exploring Random Trees*) [26], utilizados principalmente en espacios continuos de alta dimensión.

En este trabajo se emplean mapas de ocupación debido a su adecuación para representar entornos estructurados y su compatibilidad directa con algoritmos de búsqueda en rejillas discretas. En particular, se selecciona el algoritmo *A** debido a su eficiencia computacional y su capacidad para generar trayectorias libres de colisión, lo cual lo hace adecuado para la arquitectura de navegación propuesta.

ISSN: 2594-1925

3. Metodología

El sistema de mapeo y planificación de rutas se implementó sobre la arquitectura distribuida mostrada en la Figura 1.

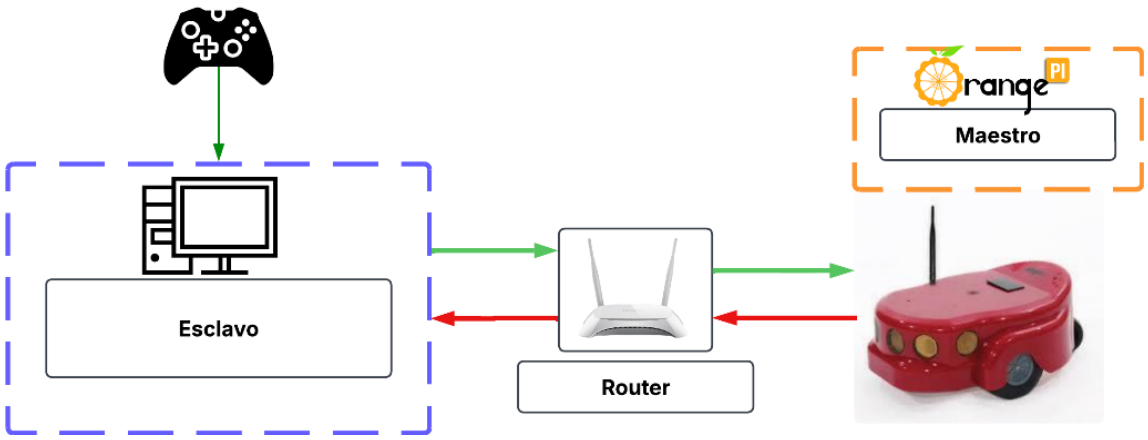


Figura 1. Arquitectura distribuida del sistema y flujo de datos de comunicación inalámbrica (protocolo ROS) entre la estación maestra a bordo (Orange Pi 5 Plus) y la estación esclava en tierra (PC de control).

El sistema consta de dos estaciones: una estación maestra, integrada por un robot móvil con configuración cinemática tipo uniciclo, modelo AmigoBot de la marca Adept Mobile Robots™, y una monoplaca Orange Pi 5 Plus encargada de la adquisición de datos sensoriales y la ejecución de los algoritmos de control; y una estación esclava, compuesta por una computadora de escritorio destinada a la visualización del entorno y a la planificación de rutas. Ambas estaciones operan bajo Ubuntu 20.04 y utilizan ROS Noetic, implementado bajo una arquitectura distribuida basada en nodos que se comunican mediante mecanismos de publicación y suscripción.

La plataforma AmigoBot dispone de *encoders* en ambas ruedas motrices para la estimación de odometría y un arreglo de ocho sensores ultrasónicos, de los cuales se emplean únicamente los seis orientados al frente. El sistema se extendió físicamente (Figura 2) mediante una cámara de profundidad Kinect y una unidad de medición inercial (IMU) 3DM-GX4-45 para la orientación del vehículo.

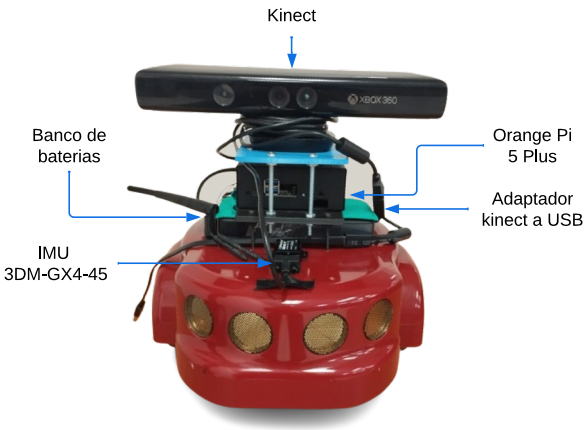


Figura 2. Distribución física de la instrumentación y sensores integrados en el WMR AmigoBot.

ISSN: 2594-1925

La adquisición de datos se realiza en ROS mediante el paquete *RosAria* [27] para los datos de sonares y *encoders* de la plataforma Amigobot. Por otra parte, la información de la cámara Kinect se captura a través del controlador *libfreenect* [28] y el paquete *laserscan_kinect* [29], el cual transforma la imagen de profundidad en un escáner láser equivalente para su integración en la rejilla de ocupación. La comunicación inalámbrica entre ambas estaciones permite delegar el procesamiento gráfico y de planificación a la estación esclava, reduciendo la carga computacional a bordo del vehículo.

3.1 Modelo cinemático

Se considera un WMR como el mostrado en la Figura 3.

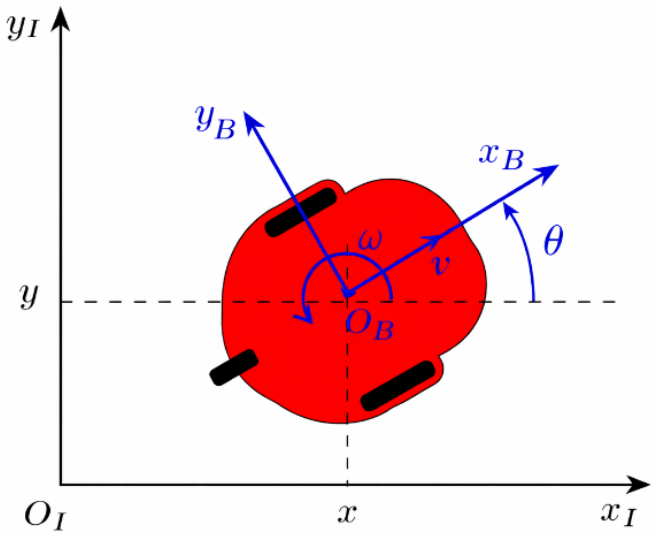


Figura 3. Representación del modelo cinemático no holonómico del robot tipo uniciclo.

El vehículo opera bajo la restricción de rodamiento puro sin deslizamiento lateral, lo que implica una restricción no holonómica que impide la existencia de velocidad lateral en el robot. Bajo esta consideración, la pose del robot respecto a un marco inercial $\{I\}$ se define mediante el vector de coordenadas generalizadas

$$q = [x, y, \theta]^T \tag{1}$$

donde x y y representan la posición del centro de masa del robot en el plano, y θ representa la orientación del eje x_B del sistema de referencia del cuerpo $\{B\}$ respecto al marco inercial.

El modelo cinemático del WMR está dado por

$$\begin{aligned} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \end{aligned} \tag{2}$$

ISSN: 2594-1925

$$\dot{\theta} = \omega$$

donde v y ω denotan la velocidad lineal y angular respectivamente, constituyendo las entradas de control del sistema.

3.2 Generación de mapas

Para capturar la geometría del espacio de trabajo sin requerir un modelado explícito de la incertidumbre de los sensores, se adoptó un enfoque de adquisición directa de datos durante una fase de exploración teleoperada en un entorno estacionario.

El proceso de construcción se implementa de forma incremental (Figura 4). Las mediciones de distancia y profundidad se transforman a un sistema de referencia común y se proyectan al plano de trabajo. Posteriormente, estas observaciones se discretizan en índices de una rejilla de ocupación de resolución fija, asignando un valor binario igual a 1 a las celdas correspondientes.

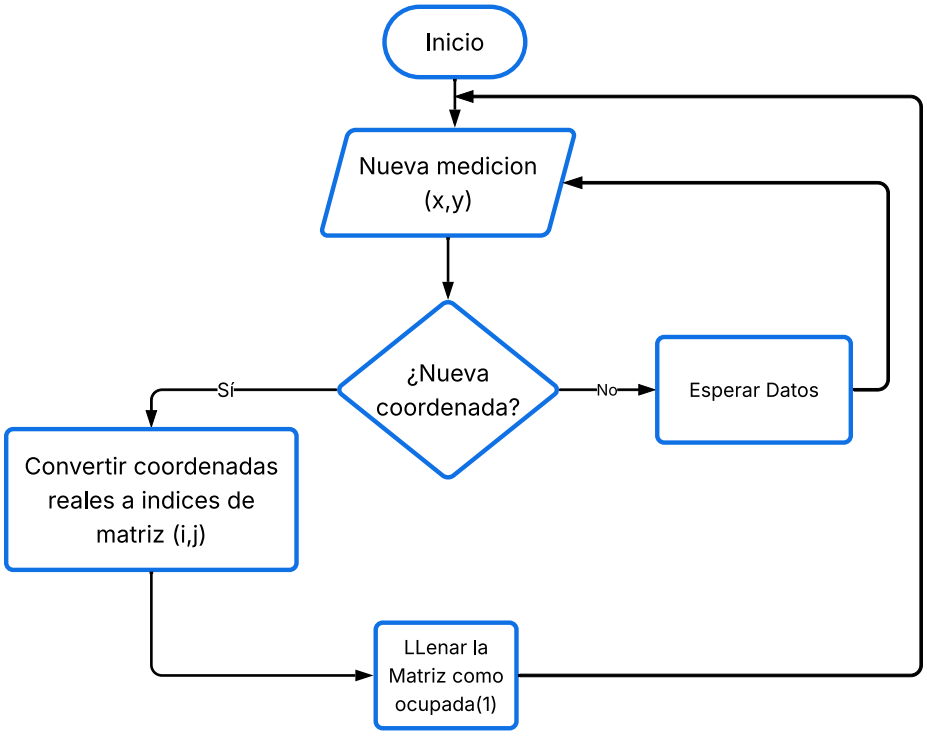


Figura 4. Algoritmo de mapeo.

3.3 Transformaciones geométricas

Para garantizar la consistencia espacial entre las mediciones provenientes de los distintos sensores y su correcta integración en la rejilla de ocupación, es necesario expresar todas las observaciones dentro de un mismo sistema de referencia global. En este contexto, se emplean transformaciones geométricas homogéneas que permiten proyectar los puntos medidos desde los marcos locales de los sensores hacia el marco inercial del mapa, asegurando coherencia en el proceso de construcción del entorno.

ISSN: 2594-1925

Sea $p^B = \left[p_x^B \; p_y^B \; 1\right]^T$ un punto, obtenido mediante los sensores ultrasónicos del robot, expresado en el marco del robot $\{B\}$. Su transformación al marco inercial $\{I\}$ se realiza mediante

$$p^I = T_B^I p^B \tag{3}$$

donde $T_B^I \in SE(2)$ es una matriz de transformación homogénea de 3×3 que expresa representa la orientación y translación del robot en el plano.

Para el sensor Kinect, las mediciones se encuentran en el marco del sensor $\{C\}$ se obtienen en coordenadas polares (r, α), las cuales se convierten a coordenadas cartesianas mediante

$$x^C = r\cos(\alpha)\ , \qquad y^C = r\sin(\alpha) + y_{offset} \tag{4}$$

donde y_{offset} es el desplazamiento del sensor respecto del centro del robot. Por lo tanto $p^C = \left[p_x^C \; p_y^C \; 1\right]^T$ se transforma al marco inercial mediante

$$p^I = T_B^I T_C^B p^C \tag{5}$$

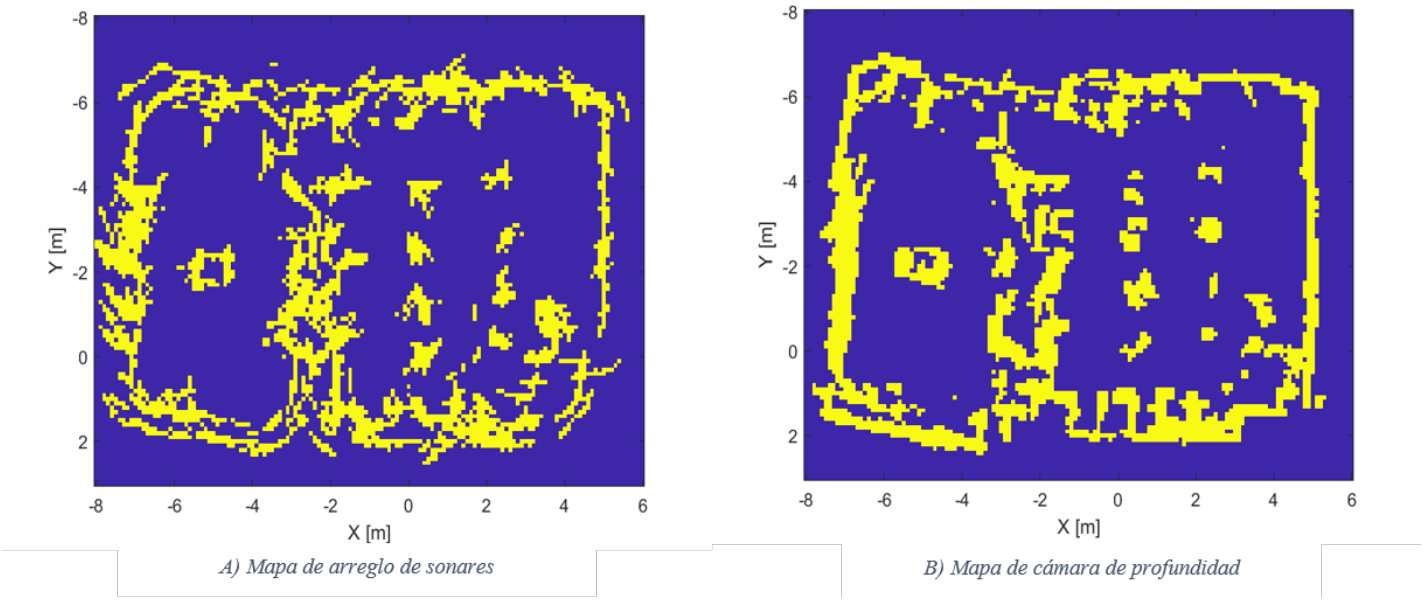
A diferencia de las validaciones en entornos controlados que emplean pasillos despejados y geometrías uniformes, las condiciones experimentales de este trabajo se desarrollaron en un entorno de laboratorio real, bajo condiciones estáticas o no cambiantes del entorno. En la Figura 5 se ilustra la distribución física y complejidad del escenario utilizado para las pruebas.

Figura 5. Escenario experimental de navegación.

ISSN: 2594-1925

Para la generación adecuada de mapas, se realiza un barrido angular controlado alrededor del eje del robot, permitiendo que los seis sonares frontales y el sensor de profundidad capturen información del entorno de forma simultánea, lo que permite una evaluación directa de la consistencia estructural entre ambos mapas sin requerir procesos adicionales de registro espacial.

La Figura 6 presenta la comparativa entre el mapa generado por sonares y el obtenido por la cámara de profundidad, configurados con una dimensión de 200 x 200 celdas y una resolución de 0.1 m por celda, lo que cubre un área de 20 m × 20 m (400 m²). Ambas modalidades ofrecen una representación cualitativamente consistente de la distribución real del laboratorio.



A) Mapa de arreglo de sonares

B) Mapa de cámara de profundidad

Figura 6. Comparativa estructural de las rejillas de ocupación antes de la fusión: (A) Mapa por sonares, (B) Mapa por cámara de profundidad.

El mapa de ultrasonido exhibe mayor dispersión y ruido, mientras que el de profundidad aporta contornos más definidos y continuidad geométrica. Al evaluar la similitud entre ambas representaciones mediante una métrica de coincidencia de celdas ocupadas, se obtuvo un valor del 59.39%, lo que refleja las diferencias en la naturaleza de los sensores y la ausencia de una optimización global tipo SLAM.

3.4 Fusión sensorial

Para obtener una representación más consistente del entorno, se propone una estrategia de combinación basada en la intersección de ambos mapas.

Sean S_{map} y C_{map} los mapas generados a partir de los sensores ultrasónicos y de la cámara de profundidad, respectivamente. Entonces, el mapa combinado se define como la intersección $S_{map} \cap C_{map}$, conservando únicamente aquellas celdas clasificadas como ocupadas en ambas representaciones. Esta operación permite reducir detecciones inconsistentes y reforzar las estructuras espaciales comúnmente observadas por ambos sensores.

ISSN: 2594-1925

La Figura 7 muestra el mapa resultante de intersección $S_{map} \cap C_{map}$. Se observa una reducción significativa del ruido, con una mejor definición de paredes y estructuras del entorno, aunque con una ligera pérdida de detalle en regiones con baja coincidencia entre sensores.

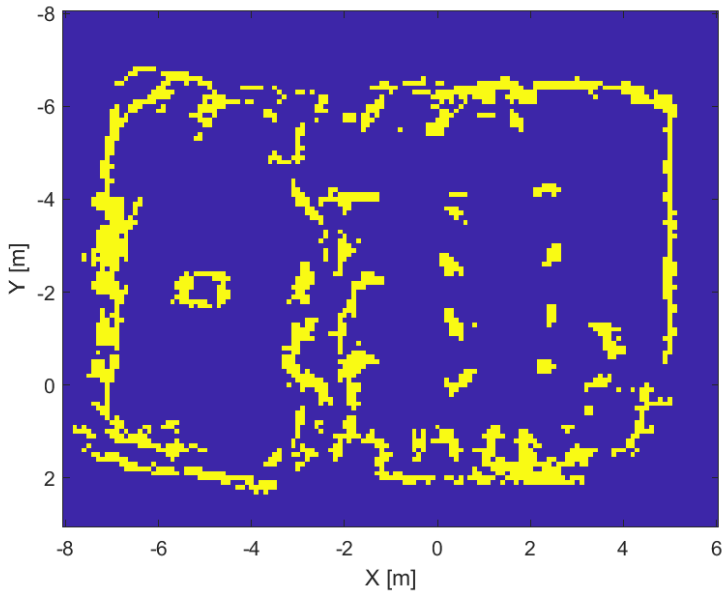


Figura 7. Mapa $S_{map} \cap C_{map}$

La intersección de los mapas permite eliminar ruido y detecciones espurias. No obstante, este proceso puede introducir discontinuidades locales debido a la pérdida de celdas aisladas durante la operación de intersección. Para mejorar la continuidad espacial y la definición de los bordes, se aplica un filtro de vecindad sobre el mapa binario resultante [30].

El filtrado se realiza mediante una vecindad 3×3 celdas alrededor de una celda central; es decir, se consideran las ocho direcciones adyacentes. Para cada celda (i, j) se calcula el número de vecinos ocupados $N(i, j)$.

El estado final de la celda se determina mediante una regla basada en umbrales: una celda permanece ocupada si ya lo estaba y cuenta con al menos un vecino ocupado, o se marca como ocupada si estaba libre y tiene al menos dos vecinos ocupados; en caso contrario, se considera libre. Este procedimiento es análogo a operaciones de filtrado espacial y morfológico empleadas en procesamiento de imágenes, donde el estado de cada elemento se actualiza en función de la información local de su vecindad. En este contexto, dicho filtrado permite reducir ocupaciones espurias y mejorar la coherencia estructural del mapa generado.

El mapa resultante en la Figura 8 recupera el detalle perdido en la Figura 7 durante la operación de intersección, preservando al mismo tiempo la reducción de ruido obtenida mediante la fusión. Esto permite disponer de una representación más consistente del entorno para la etapa posterior de planificación de rutas.

ISSN: 2594-1925

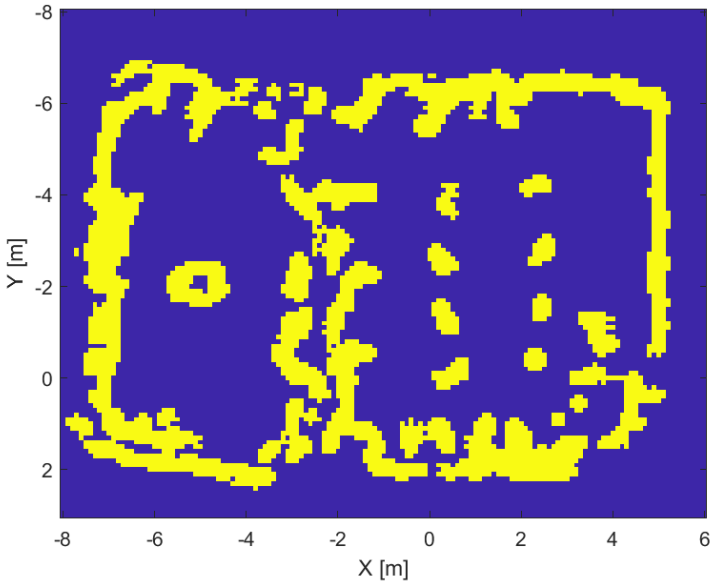


Figura 8. Mapa $S\_map \cap C_{map}$ filtrado.

*3.5 Planificación de rutas: A**

La planificación de rutas se lleva a cabo mediante el algoritmo A* [31]. Este tipo de algoritmos modela la rejilla de ocupación como un grafo, donde los nodos corresponden a cada una de las celdas de la matriz, mientras que las aristas representan las conexiones entre celdas adyacentes.

El criterio de selección se basa en la función de evaluación

$$f(n) = g(n) + h(n) \tag{6}$$

donde $g(n)$ representa el costo acumulado desde el nodo de inicio hasta el nodo actual n, mientras que $h(n)$ corresponde a una estimación heurística del costo restante hasta el nodo objetivo.

La heurística permite al algoritmo priorizar aquellos nodos que tienen mayor probabilidad de encontrarse más próximos al objetivo.

En este trabajo se emplea la distancia octil [32], la cual permite movimientos diagonales y resulta adecuada para entornos discretizados con conectividad en ocho direcciones. Sean $p(x, y)$ y $p'(x', y')$ dos puntos definidos sobre una cuadrícula discreta. La distancia octil entre ambos se define como

$$d_{\text{octil}}(p, p') = \sqrt{2}\,\min(\Delta x, \Delta y) + |\,\Delta x - \Delta y\,| \tag{7}$$

donde $\Delta x = |x - x'|$ y $\Delta y = |y - y'|$ representan las diferencias absolutas en las coordenadas horizontal y vertical, respectivamente.

ISSN: 2594-1925

Esta métrica asigna un costo $\sqrt{2}$ a los desplazamientos diagonales, mientras que los movimientos cardinales tienen un costo unitario implícito en el término $|\Delta x - \Delta y|$, proporcionando una estimación consistente del costo de navegación sobre la rejilla.

Durante el proceso de búsqueda, el algoritmo expande iterativamente el nodo más prometedor, generando sus vecinos y actualizando sus costos asociados hasta alcanzar el nodo objetivo o agotar el espacio de búsqueda.

La ruta obtenida mediante A* es óptima dentro de la rejilla discreta; sin embargo, el entorno real de navegación es continuo. Por ello, se aplica un procedimiento de simplificación basado en trazado de rayos (*ray casting*) [33], con el objetivo de eliminar puntos intermedios redundantes y obtener trayectorias más suaves.

El principio fundamental de este procedimiento es la verificación de línea de vista entre pares de puntos de la trayectoria. Este enfoque constituye la base de métodos como Theta* [34], los cuales extienden A* al permitir conexiones directas entre nodos con visibilidad mutua.

Si no se detecta colisión entre dos puntos, se considera que existe visibilidad directa y, en consecuencia, los puntos intermedios pueden eliminarse sin comprometer la seguridad de la ruta.

Estos métodos basados en visibilidad han sido ampliamente utilizados en distintos dominios, como la planificación de movimiento en robótica móvil, donde se emplean para la simplificación de trayectorias y la mejora del seguimiento en entornos discretizados [35], así como en generación de rutas eficientes y realistas para agentes autónomos dentro de entornos virtuales [36].

La verificación de línea de vista se realiza mediante la ecuación de la recta que une dos puntos $A(x_A, y_A)$ y $B(x_B, y_B)$. El algoritmo distingue tres casos, según la pendiente de la recta:

- Línea vertical: si $x_A = x_B$, se recorren todas las celdas entre y_A y y_B, verificando la ocupación en cada una de ellas.
- Pendiente suave ($| m | \leq 1$): se itera sobre el eje x y se calcula $y = \text{round}\left(\frac{y-b}{m}\right)$, evaluando la ocupación de cada celda correspondiente.
- Pendiente pronunciada ($|m| > 1$): se itera sobre el eje y para evitar omitir celdas; para cada valor discreto de y, se calcula $x = \text{round}\left(\frac{y-b}{m}\right)$ y se verifica la ocupación.

El algoritmo de trazado de rayos se aplica sobre los puntos de inflexión de la ruta obtenida mediante A*, es decir, en los cambios de dirección de la trayectoria.

4. Implementación del Sistema

La implementación de los métodos descritos se llevó a cabo mediante el desarrollo de un software de mapeo, visualización y planificación de rutas implementado en C++, el cual se ejecuta en la estación esclava. Este software constituye una de las principales contribuciones del presente trabajo, al proporcionar un entorno gráfico unificado para la construcción, procesamiento y visualización de mapas de ocupación, así como para la selección interactiva de objetivos y la generación de trayectorias de

ISSN: 2594-1925

navegación. El sistema genera las referencias de posición que posteriormente son enviadas al controlador de navegación, encargado del seguimiento de trayectoria y la evasión local de obstáculos. Asimismo, su arquitectura modular permite la incorporación de diferentes estrategias de control y su adaptación a otras plataformas robóticas con modificaciones mínimas.

4.1 Interfaz de usuario

Con el fin de facilitar la visualización de mapas y la planificación de rutas, se desarrolló una interfaz gráfica de usuario basada en las bibliotecas *SFML* (*Simple and Fast Multimedia Library*) y *TGUI*. La biblioteca *SFML* se emplea para la gestión de la ventana de renderizado, el manejo de eventos y la representación gráfica de los mapas de rejilla, mientras que *TGUI* se utiliza para la construcción de los elementos interactivos de la interfaz, tales como menús, botones, ventanas de configuración y controles de usuario.

La interfaz desarrollada permite visualizar en tiempo real el mapa generado, seleccionar de forma intuitiva posiciones objetivo sobre la rejilla de ocupación, ejecutar el proceso de planificación de trayectorias y supervisar la ruta calculada antes de su envío al robot móvil.

La interfaz resultante se muestra en la Figura 9.

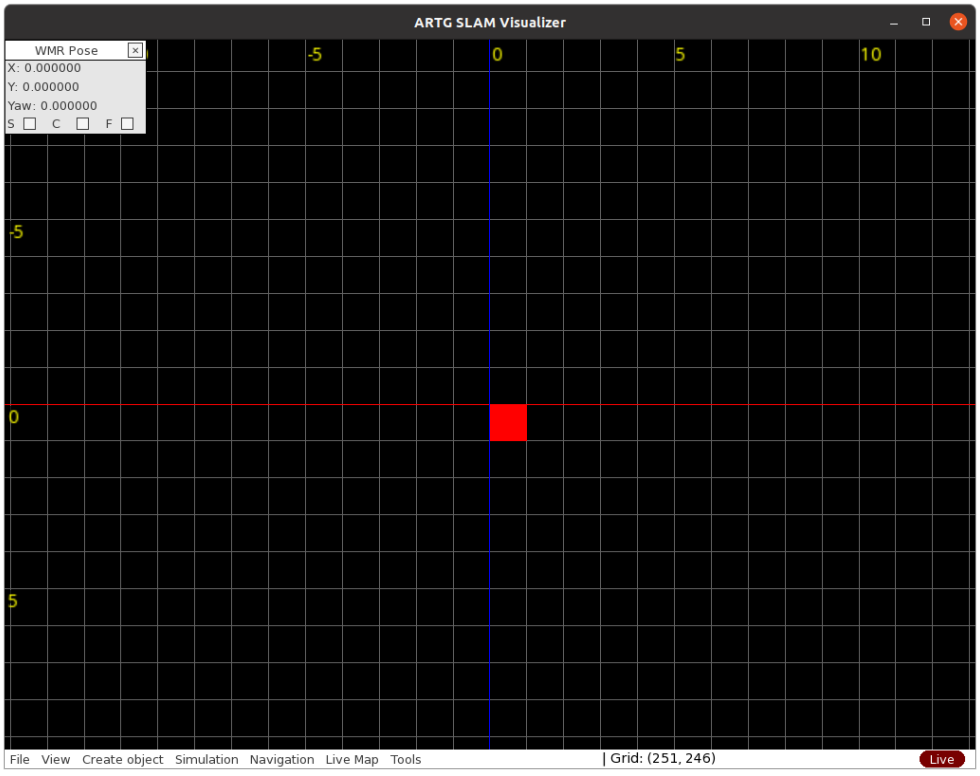


Figura 9. Interfaz de usuario del visualizador de mapas y planificador de rutas. El cuadro rojo en el centro representa la ubicación actual del vehículo.

ISSN: 2594-1925

Como puede observarse, la interfaz cuenta con un área principal de visualización de mapas de rejilla, donde la celda marcada en rojo identifica la posición instantánea del vehículo durante la navegación. Asimismo, incorpora una ventana flotante que despliega la pose estimada del robot.

Adicionalmente, se incluye una barra de menú que integra funciones para la carga y exportación de mapas, la ejecución del proceso de planificación y el inicio de la navegación. También dispone de un indicador de coordenadas de rejilla, el cual señala en tiempo real la celda sobre la que se encuentra el cursor, facilitando la selección precisa de objetivos.

El interruptor denominado *Live* habilita la adquisición continua de datos sensoriales, permitiendo la generación y actualización dinámica del mapa durante la operación del sistema.

4.2 Control de navegación

Para llevar a cabo el seguimiento de la ruta planificada, se empleó un controlador proporcional de posición y orientación, al cual se le actualiza dinámicamente la referencia deseada conforme el robot se aproxima a los puntos sucesivos de la ruta.

Sea el error de posición definido por

$$\begin{aligned} e_x &= x_d - x \\ e_y &= y_d - y \end{aligned} \tag{8}$$

donde x y y son las componentes de la posición actual del robot, mientras que x_d y y_d corresponden a las coordenadas del punto objetivo.

A partir del error de posición se obtiene la orientación deseada mediante

$$\theta_d = \operatorname{atan2}(e_y, e_x) \tag{9}$$

La función de referencia de la velocidad lineal se calcula en función de la magnitud del error de posición y se define como

$$v = k_p \sqrt{e_x^2 + e_y^2} \tag{10}$$

donde k_p es una constante proporcional asociada al error de posición.

Por otro lado, la referencia de velocidad angular se calcula en función del error de orientación

$$\omega = k_\theta e_\theta \tag{11}$$

donde k_θ es la constante proporcional asociada al error de orientación, definido como

$$e_\theta = \theta_d - \theta \tag{12}$$

ISSN: 2594-1925

donde θ es la posición angular estimada por la IMU.

Con el fin de dotar al sistema de capacidad reactiva ante obstáculos no representados en el mapa, se emplean las mediciones del arreglo frontal de sensores ultrasónicos y del sensor de profundidad. La distancia al obstáculo detectado se obtiene a partir del promedio de las mediciones válidas, denotado como s.

Cuando un obstáculo se encuentra dentro de un radio de influencia r_o, se genera un vector de repulsión lateral, perpendicular a la dirección del obstáculo. Este vector se define como

$$\begin{aligned} r_x &= \pm k_r(r_o - s)e_y \\ r_y &= \mp k_r(r_o - s)e_x \end{aligned} \tag{13}$$

donde k_r es la ganancia del campo repulsivo. El signo se selecciona en función de la posición relativa del obstáculo, permitiendo definir el sentido de la desviación lateral.

Para incorporar el efecto de la repulsión en el control, se modifica el error de posición como

$$\begin{aligned} e'_x &= (e_x + r_x)\alpha \\ e'_y &= (e_y + r_y)\alpha \end{aligned} \tag{14}$$

donde α es una constante que escala el error.

A partir de este nuevo error, se recalcula la orientación deseada, permitiendo modificar la trayectoria de forma reactiva para evitar colisiones mientras se preserva el objetivo de seguimiento de ruta.

5. Resultados

Con el fin de validar el funcionamiento de los métodos propuestos y del sistema desarrollado, se empleó el mapa mostrado en la Figura 8 para la planificación de trayectorias y la ejecución de pruebas de navegación autónoma.

La Figura 10 muestra el mapa dentro del software de visualización, además de la ruta planificada mediante el método de **A***.

ISSN: 2594-1925

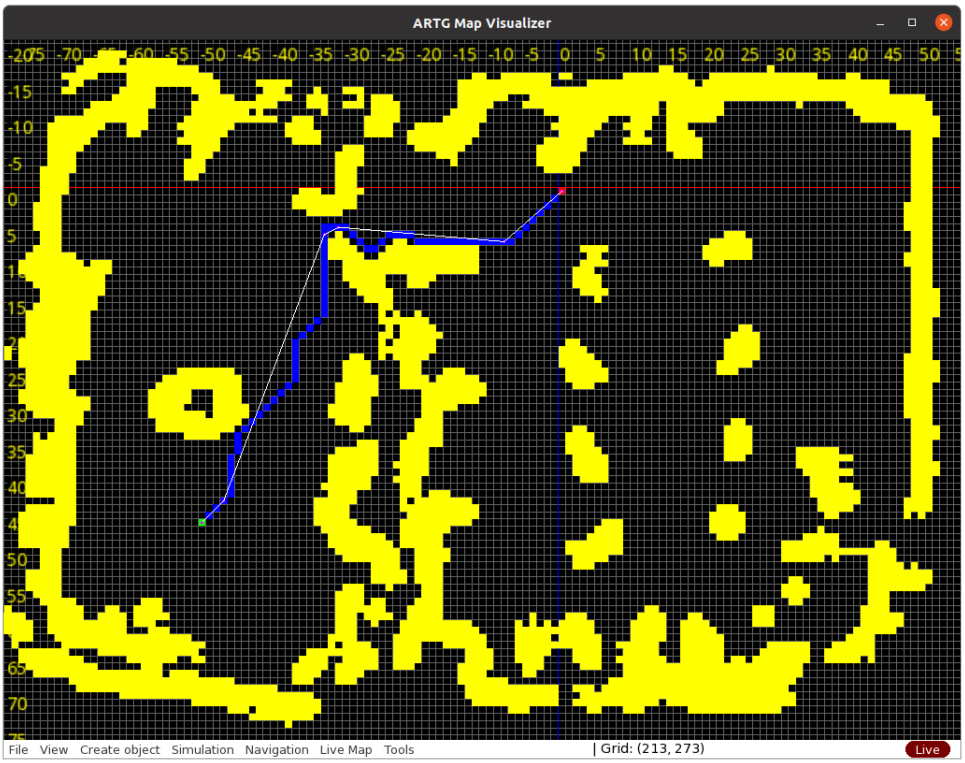


Figura 10. Visualización del mapa junto con la ruta planificada mediante el algoritmo A*.

La ruta planificada inicia en el origen del mapa, indicado en color rojo, mientras que la meta se representa en color verde. La trayectoria calculada por A* se muestra en color azul y evita adecuadamente las celdas ocupadas del entorno (celdas amarillas). La línea blanca corresponde a la ruta simplificada obtenida mediante el procedimiento de trazado de rayos.

La planificación se realizó desde la posición inicial (0,0) hasta la posición (200,296) en coordenadas discretas del mapa, equivalente a la posición física (-4.932, -4.644) m. Al tratarse de una ruta de puntos discretos, los puntos de control (*waypoints*) en la realización del avance del vehículo se actualizan cuando el error de navegación es menor a 0.25 m, lo que permite un seguimiento continuo y un movimiento fluido del mismo.

La Tabla 1 muestra los resultados de optimización de ruta planeada, desde el punto inicial a la meta se obtuvo una ruta con 75 waypoints.

Tabla 1. Métricas de optimización de trayectoria.

Métrica	Valor
Nodos de la trayectoria original	75
Puntos de inflexión detectados	16
Nodos de la trayectoria optimizada	6
Reducción respecto a puntos de inflexión	62.5 %
Reducción respecto a la trayectoria original	92 %

ISSN: 2594-1925

Los resultados muestran una reducción significativa en el número de nodos requeridos para describir la trayectoria final. En particular, la optimización logró una reducción del 92.0% respecto al número total de nodos o waypoints generados por A*, evidenciando la capacidad del procedimiento de simplificación para eliminar redundancias geométricas introducidas por la discretización de la rejilla.

5.1. Resultados de navegación

Para llevar a cabo la navegación se emplearon las siguientes constantes:

- Ganancia proporcional en posición $K_p = 0.8$
- Ganancia proporcional angular $k_\theta = 2.4$
- Ganancia del campo repulsivo $k_r = 2.8$
- Radio de influencia del obstáculo $r_o = .6\text{m}$
- Constante de escalamiento de e' $\alpha = .75$

La Figura 11 presenta el recorrido del robot dentro del mapa generado a partir de la información sensorial. Como se observa, la trayectoria experimental permite al robot desplazarse desde la posición inicial hasta la meta sin colisiones con los obstáculos presentes en el entorno, lo que evidencia la correcta integración entre el sistema de mapeo y el módulo de navegación.

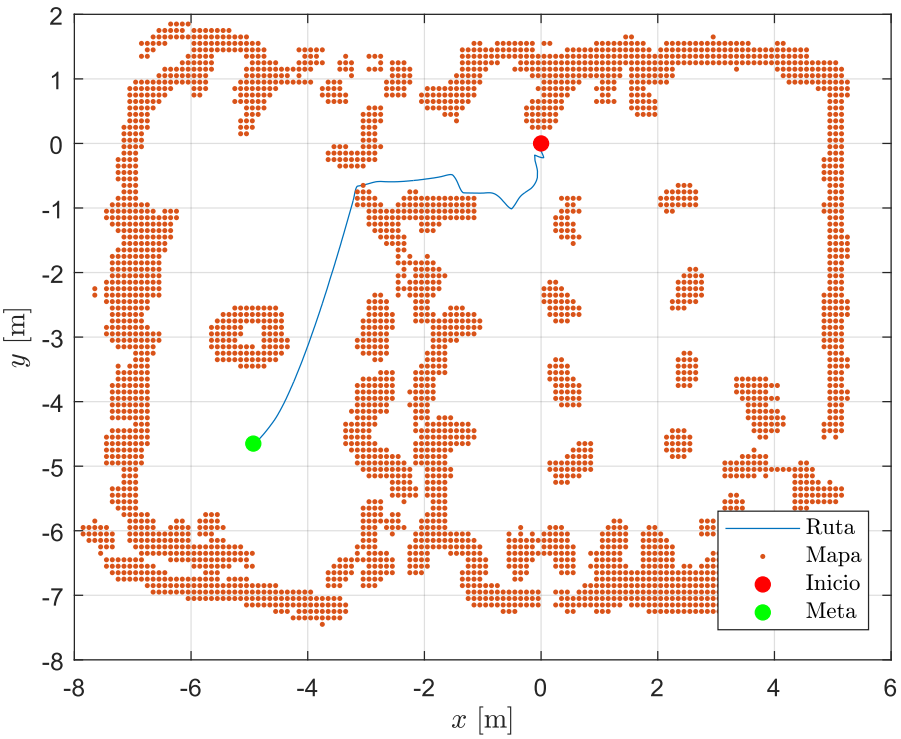


Figura 11. Ruta del robot experimental sobre el mapa generado.

Asimismo, se aprecia que la trayectoria ejecutada sigue la estructura general de la ruta planificada, presentando desviaciones locales en ciertas regiones del entorno. Estas variaciones son esperables debido

ISSN: 2594-1925

a la naturaleza continua del sistema dinámico, la discretización del mapa de ocupación y la acción del término de evasión reactiva.

La Figura 12 compara los puntos de control (en color rosa), correspondientes a los puntos obtenidos mediante el procedimiento de simplificación por *ray casting*, con la secuencia de referencia percibida por el controlador (en gris claro), la cual consiste en la repetición temporal de los mismos puntos de la ruta discreta hasta que se cumple la condición de cambio de *waypoint*. La trayectoria ejecutada por el robot se muestra en color azul. Además, los obstáculos no considerados en el mapa, pero registrados por los sensores durante la ejecución del experimento, se representan mediante cruces en color negro. Estos puntos corresponden a obstáculos detectados en línea durante la navegación, los cuales podrían haber provocado una colisión en ausencia del campo potencial de repulsión.

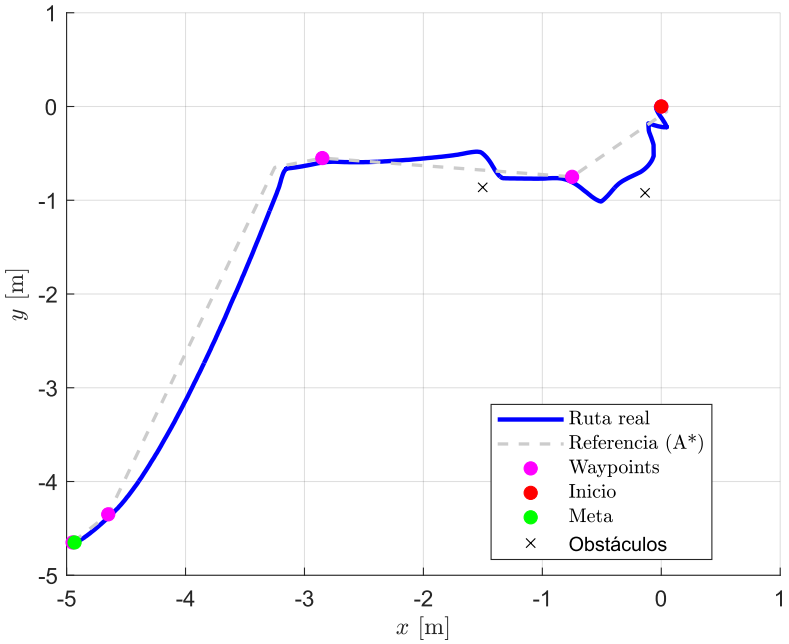


Figura 12. Comparación entre trayectoria ejecutada, ruta de referencia y puntos de control.

Como se puede observar, el robot inició su recorrido con una orientación opuesta a la meta. Durante la ejecución se registraron dos obstáculos, adicionales al mapa, los cuales fueron evitados mediante la aplicación del campo potencial, desviando la trayectoria y permitiendo completar la tarea.

En la Figura 13 se observa que el error de distancia a los puntos de control, el cual disminuye a medida que el robot se aproxima a cada punto, alcanzando valores inferiores a 0.25 m en sus proximidades. Una vez que se selecciona el siguiente punto de control como referencia, el error aumenta en consecuencia y posteriormente vuelve a disminuir conforme el robot se acerca a dicho punto. Este comportamiento se repite de manera consistente a lo largo de toda la ruta.

ISSN: 2594-1925

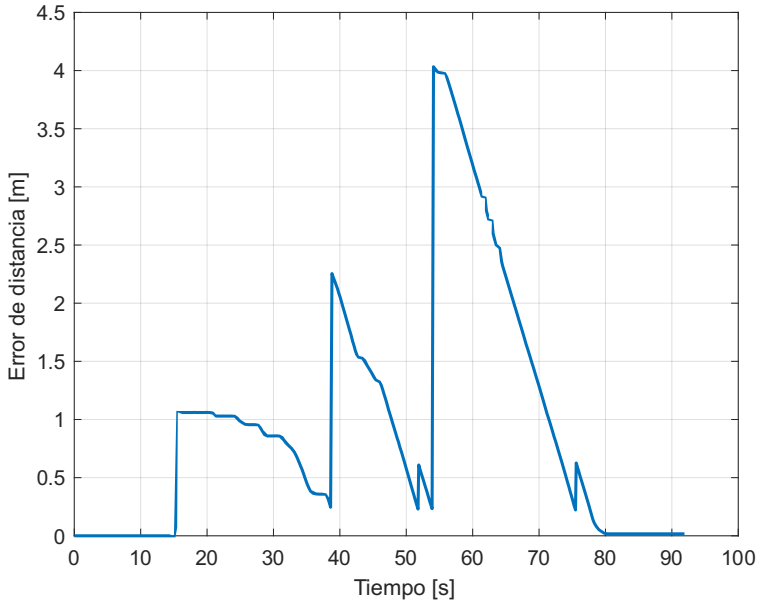


Figura 13. Error de distancia respecto a los puntos de control.

El robot es capaz de esquivar cada uno de los obstáculos dentro del mapa, así como los obstáculos adicionales detectados mediante los sensores frontales. La Figura 14 muestra la distancia mínima a los obstáculos con respecto a la trayectoria.

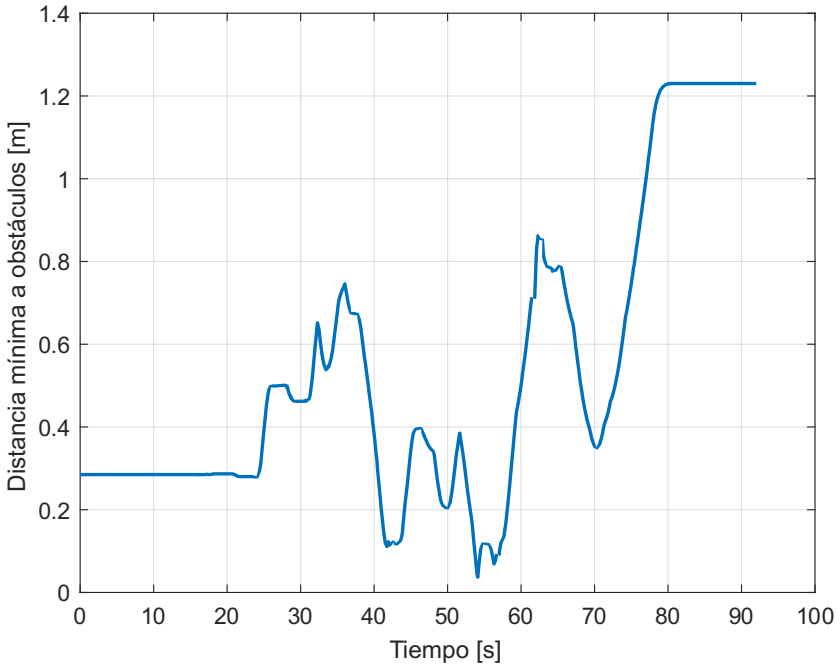


Figura 14. Distancia mínima del robot a los obstáculos a lo largo de la trayectoria seguida.

ISSN: 2594-1925

Estos resultados muestran que el sistema mantiene un seguimiento consistente de la ruta de referencia, además de mantener una distancia mínima de seguridad a los obstáculos, siendo ésta de 3.6 cm en la región donde la geometría del entorno es más estrecha.

5. Conclusiones

El sistema desarrollado permite la generación de mapas de ocupación, la planificación de rutas y la regulación del movimiento del robot, logrando la ejecución exitosa de tareas de navegación en el entorno experimental. En términos generales, los mapas obtenidos presentan una calidad suficiente para representar de forma funcional la estructura del entorno, lo que permite la planificación de rutas consistentes. El algoritmo de simplificación basado en línea de vista (*ray casting*) demostró ser eficiente para mitigar las redundancias de la discretización por rejilla, logrando una reducción del 92% en los nodos de la trayectoria original del algoritmo A^* y del 62.5% respecto a sus puntos de inflexión. Esto se traduce en rutas cinemáticamente viables y con transiciones de dirección más suaves, reduciendo la complejidad del camino computado. La estrategia de control, que combina la regulación de error de postura con campos potenciales repulsivos, cumplió con los requerimientos de diseño. El sistema garantizó el seguimiento de los *waypoints* y una tasa de éxito del 100% en la evitación de colisiones en el escenario de prueba, reaccionando ante obstáculos dinámicos o no cartografiados inicialmente. A pesar de los resultados obtenidos, se identificaron limitaciones inherentes al sistema. La fidelidad de la cartografía depende directamente del alcance y la susceptibilidad a la oclusión de los sensores empleados. Asimismo, la ausencia de un algoritmo de optimización de lazo cerrado (como Loop Closure) o de un sistema de localización global puede provocar una deriva acumulativa (drift) en la estimación de la odometría a largo plazo. En entornos dinámicos, estas limitantes podrían comprometer la consistencia del mapa respecto al estado real del entorno. Como trabajo futuro, se propone la incorporación de técnicas de localización y mapeo simultáneo (SLAM), que permitan mejorar la estimación de la posición del robot mientras se construye el mapa del entorno en tiempo real. Asimismo, se plantea la integración de mecanismos de optimización del mapeo y filtrado probabilístico, con el objetivo de mejorar la consistencia espacial y reducir la influencia del ruido sensorial en la representación del entorno.

7. Agradecimientos

Los autores desean agradecer el apoyo brinda por SECIHTI mediante la asignación de la beca CVU 2064835, y por el apoyo recibido por parte del Departamento de Investigación en Física de la Universidad de Sonora.

8. Agradecimientos de autoría

Luis Arturo García Delgado: Conceptualización; Recursos; Ideas; Metodología; Análisis formal; Investigación; Recursos; Análisis de datos; Borrador original; Revisión y edición. *Ricardo Ramon Pérez Alcocer:* Conceptualización; Recursos; Ideas; Metodología; Análisis formal; Investigación; Recursos; Análisis de datos; Borrador original; Revisión y edición. *Gilberto Ramos Valenzuela*: Conceptualización; Ideas; Metodología; Análisis formal; Investigación; Análisis de datos; Escritura; Borrador original; Revisión y edición.

ISSN: 2594-1925

Referencias

[1] A. Giannaros, A. Karras, L. Theodorakopoulos, C. Karras, P. Kranias, N. Schizas, et al., "Autonomous vehicles: Sophisticated attacks, safety issues, challenges, open topics, blockchain, and future directions," Journal of Cybersecurity and Privacy, vol. 3, no. 3, pp. 493–543, 2023, doi: 10.3390/jcp3030025.

[2] A. Taheri and Z. C. Xia, "SLAM; definition and evolution," Engineering Applications of Artificial Intelligence, vol. 97, 2021, Art. no. 104032, doi: 10.1016/j.engappai.2020.104032.

[3] Z. Ren, L. Wang, and L. Bi, "Robust GICP-based 3D LiDAR SLAM for underground mining environment," Sensors, vol. 19, no. 13, 2019, Art. no. 2915, doi: 10.3390/s19132915.

[4] M. Hao, J. Ren, X. Ji, Y. Bi, S. Zhao, and M. Wu, "Research on multimodal data enhanced SLAM algorithm for global mapping of underground coal mines," Scientific Reports, vol. 15, no. 1, 2025, Art. no. 37124, doi: 10.1038/s41598-025-37124-0.

[5] Y. Huang, J. McConnell, X. Lin, and B. Englot, "DRACo-SLAM2: Distributed Robust Acoustic Communication-efficient SLAM for Imaging Sonar Equipped Underwater Robot Teams with Object Graph Matching," in 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2025, pp. 3587–3594, doi: 10.1109/IROS62453.2025.10970000.

[6] M. Heshmat, L. Saad Saoud, M. Abujabal, A. Sultan, M. Elmezain, L. Seneviratne, and I. Hussain, "Underwater SLAM meets deep learning: challenges, multi-sensor integration, and future directions," Sensors, vol. 25, no. 11, 2025, Art. no. 3258, doi: 10.3390/s25113258.

[7] J. Lim and K. H. Chon, "Minimax Rao-blackwellized particle filtering in 2D LiDAR SLAM," International Journal of Control, Automation and Systems, vol. 22, no. 6, 2024, pp. 1947–1957, doi: 10.1007/s12555-023-0268-z.

[8] T. P. Kucner, M. Magnusson, S. Mghames, L. Palmieri, F. Verdoja, C. S. Swaminathan, et al., "Survey of maps of dynamics for mobile robots," The International Journal of Robotics Research, vol. 42, no. 11, pp. 977–1006, 2023, doi: 10.1177/02783649231169812.

[9] I. Ullah, D. Adhikari, H. Khan, M. S. Anwar, S. Ahmad, and X. Bai, "Mobile robot localization: Current challenges and future prospective," Computer Science Review, vol. 53, 2024, Art. no. 100651, doi: 10.1016/j.cosrev.2024.100651.

[10] L. Yang, P. Li, S. Qian, H. Quan, J. Miao, M. Liu, et al., "Path planning technique for mobile robots: A review," Machines, vol. 11, no. 10, 2023, Art. no. 980, doi: 10.3390/machines11100980.

[11] Z. Zhou, X. Feng, S. Di, and X. Zhou, "A LiDAR mapping system for robot navigation in dynamic environments," IEEE Transactions on Intelligent Vehicles, 2023, doi: 10.1109/TIV.2023.3298456.

ISSN: 2594-1925

[12] L. Sun, Y. Zhang, and H. Yu, "Multi-sensor perception and spatial mapping for mobile robots in dynamic environments: A survey," Robotics and Autonomous Systems, vol. 172, Art. no. 104590, 2024, doi: 10.1016/j.robot.2023.104590.

[13] J. Jung, Y. Lee, J. Park, and T. K. Yeu, "Multi-modal sonar mapping of offshore cable lines with an autonomous surface vehicle," Journal of Marine Science and Engineering, vol. 10, no. 3, p. 361, 2022, doi: 10.3390/jmse10030361.

[14] H. Huang, L. Li, F. Cheng, and S. K. Yeung, "Photo-SLAM: Real-time simultaneous localization and photorealistic mapping for monocular stereo and RGB-D cameras," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21584–21593, 2024, doi: 10.1109/CVPR52733.2024.02039.

[15] T. Grebner, A. Grathwohl, P. Schoeder, V. Janoudi, and C. Waldschmidt, "Probabilistic SAR processing for high-resolution mapping using millimeter-wave radar sensors," IEEE Transactions on Aerospace and Electronic Systems, vol. 59, no. 5, pp. 4800–4814, 2023, doi: 10.1109/TAES.2023.10164269.

[16] S. Y. Alaba, "GPS-IMU Sensor Fusion for Reliable Autonomous Vehicle Position Estimation," arXiv preprint arXiv:2405.08119, 2024, doi: 10.48550/arXiv.2405.08119.

[17] M. Münzinger, N. Prechtel, and M. Behnisch, "Mapping the urban forest in detail: From LiDAR point clouds to 3D tree models," Urban Forestry & Urban Greening, vol. 74, 2022, Art. no. 127637, doi: 10.1016/j.ufug.2022.127637.

[18] X. Liu, A. Prabhu, F. Cladera, I. D. Miller, L. Zhou, C. J. Taylor, and V. Kumar, "Active metric-semantic mapping by multiple aerial robots," in 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 3282–3288, 2023, doi: 10.1109/ICRA48891.2023.10161564.

[19] S. Zhao, P. Bhattacharya, and M. Han, "A review of visual-LiDAR fusion based simultaneous localization and mapping (SLAM) and metric-semantic mapping," Robotics and Autonomous Systems, vol. 174, Art. no. 104642, 2024, doi: 10.1016/j.robot.2024.104642.

[20] K. Guo, W. Liu, and J. Pan, "End-to-End Trajectory Distribution Prediction Based on Occupancy Grid Maps," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2242–2251, 2022, doi: 10.1109/CVPR52688.2022.00228.

[21] S. Sunil, S. Mozaffari, R. Singh, B. Shahrrava, and S. Alirezaee, "Feature-Based Occupancy Map-Merging for Collaborative SLAM," Sensors, vol. 23, no. 6, 2023, Art. no. 3114, doi: 10.3390/s23063114.

[22] Z. An, X. Rui, and C. Gao, "Improved A* Navigation Path-Planning Algorithm Based on Hexagonal Grid," ISPRS International Journal of Geo-Information, vol. 13, no. 5, 2024, Art. no. 166, doi: 10.3390/ijgi13050166.

ISSN: 2594-1925

[23] M. E. Miyombo, Y. K. Liu, C. M. Mulenga, A. Siamulonga, M. C. Kabanda, P. Shaba, et al., "Optimal path planning in a real-world radioactive environment: A comparative study of A-star and Dijkstra algorithms," Nuclear Engineering and Design, vol. 420, 2024, Art. no. 113039, doi: 10.1016/j.nucengdes.2024.113039.

[24] H. Wang, S. Lou, J. Jing, Y. Wang, W. Liu, and T. Liu, "The EBS-A* algorithm: An improved A* algorithm for path planning," PLOS ONE, vol. 17, no. 2, 2022, Art. no. e0263841, doi: 10.1371/journal.pone.0263841.

[25] X. Li, Y. Lu, X. Zhao, X. Deng, and Z. Xie, "Path planning for intelligent vehicles based on improved D* Lite," Journal of Supercomputing, vol. 80, no. 1, 2024, Art. no. 1294, doi: 10.1007/s11227-023-05528-1.

[26] H. Tu, Y. Deng, Q. Li, M. Song, and X. Zheng, "Improved RRT global path planning algorithm based on Bridge Test," Robotics and Autonomous Systems, vol. 171, 2024, Art. no. 104570, doi: 10.1016/j.robot.2024.104570.

[27] "ROSARIA," ROS Wiki. [Online]. Available: https://wiki.ros.org/ROSARIA

[28] "libfreenect," ROS Wiki. [Online]. Available: https://wiki.ros.org/libfreenect

[29] "laserscan_kinect," ROS Wiki. [Online]. Available: https://wiki.ros.org/laserscan_kinect

[30] Y. Gong, "OSBF: One-sided box filter for edge-preserving image processing," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.10942596.

[31] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968, doi: 10.1109/TSSC.1968.300136.

[32] H. K. Balakrishnan, A. Suryan, A. P. Alex, S. S. Raj, and G. Zhang, "Heuristic evaluation in A-star algorithm for enhanced urban UAV path optimization," SSRN Electronic Journal, 2025, doi: 10.2139/ssrn.5079937.

[33] U. Karki and P. J. Parikh, "Visibility-based layout of a hospital unit—An optimization approach," Health Care Management Science, vol. 27, no. 2, 2024, doi: 10.1007/s10729-023-09662-1.

[34] Y. Zhang, Y. Hu, J. Lu, and Z. Shi, "Research on path planning of mobile robot based on improved Theta* algorithm," Algorithms, vol. 15, no. 12, Art. no. 477, 2022, doi: 10.3390/a15120477.

[35] J. Zhang, P. Tsiotras, and Y. Yan, "Visibility-aware trajectory optimization for mobile robots in cluttered environments," in 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 9823–9829, 2021, doi: 10.1109/ICRA48506.2021.9561021.

ISSN: 2594-1925

[36] D. A. Daohong Liu, “Research of the Path Finding Algorithm A* in Video Games,” Highlights in Science, Engineering and Technology, vol. 39, pp. 763–768, 2023, doi: 10.54097/hset.v39i.6642.

ISSN: 2594-1925