

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

Proyecto Final: Robótica y Sistemas Autónomos

Cristofer Rodrigo Contreras Ahumada
Ignacio Andrés Cuevas Suárez
Vicente Maximiliano Morales Maturana

Profesora:
Sandra Cano Mazuera

Asignatura:
Robótica y Sistemas Autónomos - ICI4150

27 de junio de 2025

Índice

1	Introducción	3
2	Descripción del robot móvil	4
2.1	Objetivo General:	4
2.2	Objetivos específicos del robot	4
3	Explicación del entorno simulado	7
3.1	Webots: Simulador de Robótica 3D	7
3.2	Características del Entorno Simulado	7
3.3	Interacción del Robot con el Entorno Simulado	8
3.4	Ventajas del Uso de Webots para la Simulación	9
4	Arquitectura del software	10
4.1	Percepción	10
4.2	Planificación	10
4.3	Control	11
5	Algoritmos a utilizar	12
5.1	Algoritmo A* (A-Star)	12
5.2	Replanteamiento Dinámico de la Ruta	13
5.3	Fusión de Sensores	13
5.4	Algoritmos de Control	14
6	Diagramas de flujo y pseudocódigo	15
7	Resultados obtenidos	16
7.1	Métricas de Desempeño	16
7.2	Análisis de Algoritmos	16
7.3	Reflexión sobre Mejoras	17
8	Conclusiones	18
9	Referencias	19

Índice de figuras

1	Vista del robot	4
2	Onda cuadrada como referencia	5
3	Escaneo del entorno con LIDAR 2D	5
4	Los obstáculos dificultad la llegada a la meta	5
5	Un mapa de ocupación mantiene información sobre los obstáculos	5
6	El robot diferencial permite un giro controlado de cada rueda	6
7	Entorno estático definido	8
8	Diagrama de flujo	15

1. Introducción

El presente proyecto se desarrolla para la asignatura **Robótica y Sistemas Autónomos**, donde se nos pide diseñar e implementar un robot móvil autónomo usando el simulador Webots. Este robot debe ser capaz de navegar de manera independiente a través de un entorno 3D predefinido y estático, sin conocimiento previo de los obstáculos que encontrará a lo largo de su trayectoria, los cuales deberá evitar mediante la percepción sensorial y algoritmos de planificación de rutas.

Para este propósito, se implementará un controlador en C++, que gestionará los sensores de distancia y posicionamiento instalados en el robot, permitiéndole interactuar con su entorno y tomar decisiones en tiempo real para evitar obstáculos. Este enfoque es fundamental para la comprensión y aplicación de conceptos clave que se han abordado durante el semestre en la asignatura, tales como la cinemática de robots móviles, el diseño de sistemas embebidos, la fusión sensorial y localización, y la planificación de rutas óptimas a través de algoritmos de navegación.

La relevancia de realizar este proyecto, está en el uso y aplicaciones de conceptos fundamentales que hemos visto durante el semestre en la asignatura, destacándose como aquellos más relevantes las temáticas de: **cinemática** de robots móviles, diseño de **sistemas embebidos**, la **fusión sensorial** y localización y por último la **planificación de rutas** óptimas utilizando algoritmos de navegación. La oportunidad de implementar en Webots esta idea, permite validar conceptos teóricos en un entorno controlado antes de su traslado a plataformas físicas, reduciendo riesgos técnicos y económicos.

De este modo, no solo se refuerzan los conocimientos adquiridos durante la asignatura, sino que también entrega la oportunidad para experimentar con la integración de múltiples tecnologías en un sistema autónomo.

2. Descripción del robot móvil

El robot que decidimos usar es el mismo que llevamos utilizando algún tiempo en los talleres. Este robot está compuesto de un cuerpo, cuatro ruedas y dos sensores de distancia básicos, tal como se puede apreciar en la figura 1.

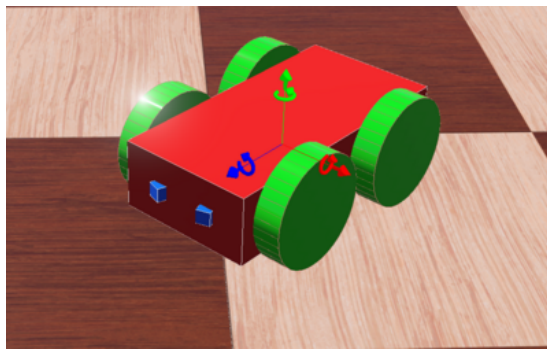


Figura 1: Vista del robot

Como equipo, hemos definido un nombre único que identifique al robot: **Lokotron**. Por lo que para referirnos a él, durante el resto del informe se hará uso de ambas denominaciones. Se trata de una manera de animarnos al desarrollo e implementación de funcionalidades durante todo el desarrollo, buscando satisfacer los requisitos que no se pudieron cumplir en los laboratorios pasados.

El objetivo principal de Lokotron debe ser llegar a la meta, manteniendo en lo posible la trayectoria que ya ha sido definida. Sin embargo, ante imprevistos tendrá que planificar nuevas rutas y estrategias de avance que le permitan reintegrarse y continuar su funcionamiento normal.

2.1. Objetivo General:

Desarrollar un robot móvil autónomo en Webots, capaz de navegar por un entorno simulado evitando obstáculos, generando un mapa básico y planificando rutas para alcanzar objetivos definidos.

2.2. Objetivos específicos del robot

Como mínimo, el robot también debe cumplir con los siguientes aspectos:

- **Seguir una trayectoria** en forma de onda cuadrada, con amplitud de 4m y longitud de onda de 2m, permitiendo visualizar hasta 5 ciclos completos en un espacio de 10m.

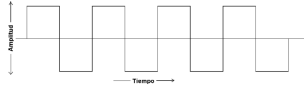


Figura 2: Onda cuadrada como referencia

- **Ser capaz de percibir su entorno** a través de sensores LIDAR, ultrasónicos, infrarrojos y GPS.

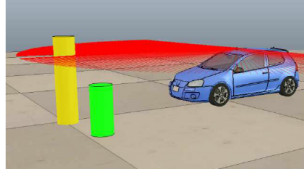


Figura 3: Escaneo del entorno con LIDAR 2D

- **Evitar obstáculos**, respondiendo dinámicamente a obstáculos de formas y tamaños variables, posicionados estratégicamente para interceptar la ruta planeada.

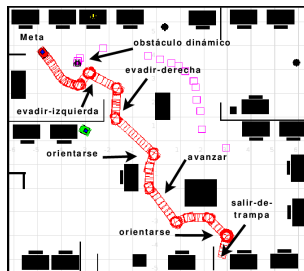


Figura 4: Los obstáculos dificultad la llegada a la meta

- **Generar mapas** de ocupación 2D en tiempo real, mediante fusión de datos LIDAR y odometría.

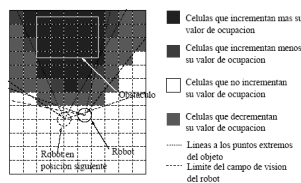


Figura 5: Un mapa de ocupación mantiene información sobre los obstáculos

- **Control y navegación precisos**, manteniendo desviaciones de trayectoria controladas mediante algoritmos de cinemática diferencial y GPS.

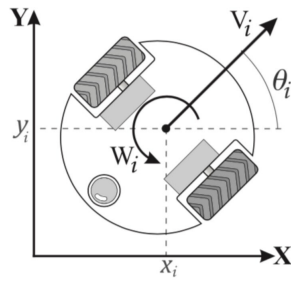


Figura 6: El robot diferencial permite un giro controlado de cada rueda

3. Explicación del entorno simulado

El entorno simulado es una parte fundamental de este proyecto, ya que permite validar la funcionalidad del robot móvil autónomo sin la necesidad de un entorno físico. En esta sección, se describe el simulador utilizado, las características del entorno simulado y cómo el robot interactúa con él para realizar la navegación autónoma. El simulador elegido para este proyecto es *Webots*, un simulador de robótica 3D ampliamente utilizado en la academia y la industria.

3.1. Webots: Simulador de Robótica 3D

Webots es un simulador de código abierto utilizado para el desarrollo, prueba y validación de robots móviles. Este simulador permite crear escenarios 3D detallados donde los robots pueden interactuar con diferentes tipos de sensores y actuadores, así como simular condiciones del mundo real. El entorno en Webots es completamente personalizable, permitiendo el diseño de mapas, la implementación de robots con diferentes capacidades y la simulación de obstáculos dinámicos.

En este proyecto, Webots es utilizado para crear un entorno controlado que simula un espacio con obstáculos, lo cual permite evaluar el comportamiento del robot sin la necesidad de un hardware físico. Esto facilita la experimentación y la iteración rápida sobre los algoritmos de navegación, además de reducir los riesgos asociados a la implementación física.

3.2. Características del Entorno Simulado

El entorno simulado en Webots ha sido diseñado con las siguientes características clave:

- **Mapa Estático:** El entorno simulado está compuesto por un mapa 3D predefinido que representa una cuadrícula de 8x8, donde el robot debe navegar. El mapa está formado por celdas que pueden estar libres o bloqueadas por obstáculos.

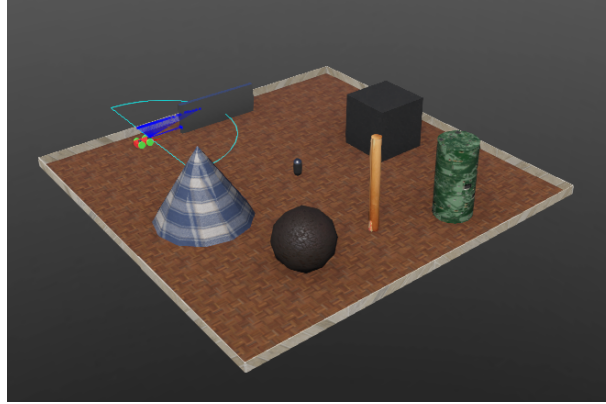


Figura 7: Entorno estático definido

- **Obstáculos Estáticos y Dinámicos:** En el entorno se incluyen obstáculos que pueden ser estáticos (por ejemplo, paredes o bloques fijos) y dinámicos (obstáculos que pueden moverse o cambiar de posición en tiempo real). El robot debe ser capaz de detectar y evitar estos obstáculos utilizando sus sensores de distancia y LIDAR.
- **Terreno Plano:** El entorno se simula en un plano 3D con una altura constante, lo que facilita la navegación sin la complicación de terrenos irregulares.
- **Sensores del Robot:** El robot está equipado con sensores de distancia (ultrasónicos o infrarrojos), LIDAR y un GPS. Los sensores de distancia permiten detectar obstáculos cercanos, mientras que el LIDAR proporciona una visión más amplia del entorno, ideal para detectar obstáculos a mayor distancia. El GPS se utiliza para rastrear la posición del robot en el mapa.

3.3. Interacción del Robot con el Entorno Simulado

El robot interactúa con el entorno simulado a través de sus sensores y actuadores. A continuación, se describe cómo el robot utiliza estos dispositivos para percibir su entorno y tomar decisiones durante la navegación:

- **Percepción del Entorno:**
 - El robot utiliza el LIDAR para obtener una representación más detallada y precisa del entorno a su alrededor, detectando obstáculos a distancias de hasta 1 metro.
 - Los sensores de distancia permiten detectar obstáculos cercanos en las direcciones frontal izquierda y derecha. Esta información es crucial para evitar colisiones inminentes.
 - El GPS le proporciona al robot su ubicación exacta dentro del entorno simulado, lo que le permite conocer su posición en el mapa 3D.

■ **Planificación de Ruta:**

- A partir de los datos recogidos por los sensores, el robot calcula la ruta más óptima hacia el objetivo utilizando el algoritmo A^* , basado en una grilla 8x8.
- El robot realiza una planificación de ruta inicial antes de comenzar su desplazamiento y recalcula la ruta si detecta obstáculos imprevistos durante el movimiento.

■ **Movimiento:**

- El robot controla su movimiento utilizando un sistema de ruedas que se ajustan a través de un control cinemático diferencial. Las velocidades de las ruedas izquierda y derecha se ajustan en función de la dirección y el ángulo hacia el siguiente waypoint en la ruta calculada.
- Durante la navegación, si se detecta un obstáculo cercano, el robot realiza maniobras evasivas (por ejemplo, girar en el lugar) para evitar la colisión y luego sigue el camino recalculado.

3.4. Ventajas del Uso de Webots para la Simulación

La elección de Webots como plataforma de simulación para este proyecto ofrece varias ventajas significativas:

- **Validación Rápida:** Permite realizar pruebas de los algoritmos de navegación, planificación de rutas y evasión de obstáculos sin la necesidad de hardware físico, acelerando el proceso de desarrollo.
- **Entorno Controlado:** Ofrece un entorno controlado en el que se pueden replicar diversas condiciones, como obstáculos estáticos y dinámicos, sin preocuparse por factores impredecibles.
- **Facilidad de Integración:** Webots soporta una amplia variedad de sensores y actuadores, lo que facilita la integración de los dispositivos utilizados en el robot para la simulación.
- **Escalabilidad:** Se pueden diseñar entornos de simulación mucho más complejos y dinámicos, permitiendo escalar el proyecto en función de las necesidades del desarrollo.

4. Arquitectura del software

El sistema de control del robot se organiza en tres niveles principales:

4.1. Percepción

- **LIDAR 2D (128 resoluciones):**

- Obtiene un mapa parcial del entorno en cada ciclo.
- Detecta obstáculos en un rango de hasta 1 metro.
- Construye un mapa de ocupación (grilla 8x8), donde cada celda representa una zona del entorno.

- **Sensores de distancia (ultrasónicos/infrarrojos):**

- Detectan obstáculos cercanos en las proximidades del robot, especialmente en las direcciones frontal izquierda y derecha.
- Proveen una capa de seguridad adicional para la evasión reactiva ante obstáculos inmediatos.

- **GPS:**

- Obtiene la posición actual del robot en el entorno simulado en coordenadas X, Z (considerando el plano horizontal en Webots).

4.2. Planificación

- **Grilla de ocupación:**

- Mapa 2D representado como una matriz de tamaño `grid[8][8]`.
- Las celdas se marcan como libres o ocupadas, dependiendo de la información proveniente del LIDAR y los sensores de distancia.
- La grilla proporciona una representación visual del entorno, permitiendo al algoritmo de planificación realizar su tarea de manera eficiente.

- **Algoritmo de planificación A* (A-Star):**

- Calcula la ruta óptima desde la posición actual del robot hasta el objetivo.
- Se ejecuta en cada ciclo del robot, ajustando la planificación según los cambios en el mapa causados por la detección de obstáculos.

- **Replanteamiento dinámico de la ruta:**

- El sistema recalcula la ruta del robot en tiempo real si se detecta un obstáculo imprevisto en su trayectoria.
- Si un obstáculo es detectado por los sensores de distancia o el LIDAR, el robot detiene su movimiento y recalcula la ruta utilizando la información más reciente del entorno.
- Este replanteamiento dinámico se ejecuta de manera continua en cada ciclo, garantizando que el robot siempre siga el camino más seguro y eficiente, incluso cuando se presentan obstáculos inesperados.

4.3. Control

■ Control de navegación:

- Si se detecta un obstáculo cercano con los sensores de distancia o LIDAR, el robot ejecuta una evasión reactiva, girando en su lugar para evitar la colisión.
- Si no hay obstáculos cercanos, el robot sigue la trayectoria calculada por el algoritmo A*, moviéndose hacia el siguiente punto en el camino.

■ Controlador de motores:

- Utiliza un control cinemático diferencial para gestionar las velocidades de las ruedas izquierda y derecha del robot.
- Los ajustes de velocidad se realizan en función del ángulo hacia el waypoint (punto de destino) más cercano, asegurando que el robot se desplace hacia él de manera eficiente.

Esta arquitectura modular permite la integración de múltiples fuentes de percepción, proporcionando una navegación robusta. Se combina una planificación deliberativa (A*) con comportamientos reactivos (evasión rápida y replanteamiento dinámico de la ruta), asegurando que el robot pueda adaptarse a su entorno en tiempo real y seguir el camino más seguro y eficiente a pesar de los obstáculos imprevistos.

5. Algoritmos a utilizar

En este proyecto, se implementaron diversos algoritmos que permiten al robot móvil autónomo navegar de manera eficiente y evitar obstáculos en un entorno 3D predefinido. Los algoritmos clave utilizados en el desarrollo de este sistema incluyen el **algoritmo A*** para la planificación de rutas y el algoritmo de **replanteamiento dinámico de la ruta** para adaptarse a obstáculos imprevistos. A continuación, se describe cada uno de estos algoritmos y su implementación.

5.1. Algoritmo A* (A-Star)

El algoritmo A* es un algoritmo de búsqueda utilizado para encontrar el camino más corto entre dos puntos en un espacio de búsqueda. En el contexto de este proyecto, A* se aplica para planificar la ruta óptima desde la posición actual del robot hasta el objetivo final, evitando los obstáculos detectados en el entorno.

El algoritmo A* combina los costos de recorrido desde el nodo inicial y una estimación heurística del costo desde el nodo actual hasta el objetivo, lo que le permite encontrar de manera eficiente el camino más corto. Para su implementación, se utiliza una grilla 2D de tamaño 8x8, donde cada celda representa una posible ubicación en el mapa del entorno.

Función Heurística

La heurística utilizada en A* es la distancia Manhattan, la cual calcula la suma de las diferencias absolutas entre las coordenadas del nodo actual y el objetivo. Esta heurística es adecuada en escenarios de grillas y garantiza la eficiencia del algoritmo en el entorno dado. La función heurística $h(x, y)$ se define como:

$$h(x, y) = |x_{actual} - x_{objetivo}| + |y_{actual} - y_{objetivo}|$$

Operación del Algoritmo

En cada ciclo, A* explora las celdas vecinas al nodo actual, eligiendo aquellas que minimizan la suma de los costos $g(x, y)$ (costo acumulado desde el inicio) y la heurística $h(x, y)$. El algoritmo sigue iterando hasta encontrar el camino más corto o hasta que no se pueda encontrar un camino, lo que ocurre si el robot se encuentra bloqueado por obstáculos en todo su camino.

Ventajas del Algoritmo A*

- Encuentra el camino más corto de manera eficiente, si existe un camino.
- Utiliza una heurística que permite reducir significativamente la cantidad de nodos a explorar en comparación con otros algoritmos de búsqueda como el algoritmo de Dijkstra.
- Es ideal para aplicaciones en grillas, como es el caso en este proyecto.

5.2. Replanteamiento Dinámico de la Ruta

El replanteamiento dinámico de la ruta es un algoritmo adicional que permite al robot adaptarse a cambios inesperados en el entorno, como la aparición de obstáculos imprevistos durante su navegación. En lugar de seguir una ruta previamente calculada sin cambios, el robot recalcula su ruta en tiempo real cada vez que detecta un nuevo obstáculo cercano.

Funcionamiento del Replanteamiento Dinámico

El replanteamiento dinámico funciona en conjunto con el algoritmo A*, el cual se ejecuta inicialmente para planificar la ruta. Durante la navegación, si el robot detecta un obstáculo a través de los sensores de distancia o el LIDAR, el algoritmo detiene temporalmente el avance y genera una nueva ruta desde la posición actual del robot hacia el objetivo, teniendo en cuenta los cambios recientes en el mapa del entorno. Este proceso asegura que el robot siempre esté siguiendo el camino más seguro y eficiente disponible.

El proceso de replanteamiento se realiza en cada ciclo del robot, lo que le permite adaptarse a cambios dinámicos en su entorno sin necesidad de detenerse por completo o volver al punto de inicio.

Implementación del Replanteamiento Dinámico

El código implementa una verificación continua de los obstáculos durante la navegación. Si un obstáculo es detectado, el mapa de ocupación (grilla 8x8) se actualiza para reflejar la nueva información y el algoritmo A* se ejecuta nuevamente. Esto permite que el robot evite colisiones y ajuste su ruta de manera eficiente.

5.3. Fusión de Sensores

El sistema de fusión sensorial en este proyecto integra la información proveniente de diferentes sensores, como el LIDAR, los sensores de distancia y el GPS, para proporcionar

una visión más precisa y confiable del entorno. La fusión sensorial permite al algoritmo A^* y al sistema de replanteamiento dinámico tener una representación precisa de los obstáculos y la ubicación del robot en tiempo real.

Los datos del LIDAR proporcionan información detallada sobre los obstáculos en el entorno, mientras que los sensores de distancia ofrecen una capa adicional de seguridad para detectar obstáculos inmediatos y cercanos. El GPS proporciona información de localización precisa, que se utiliza para rastrear la posición del robot en el mapa del entorno.

5.4. Algoritmos de Control

Además de la planificación de rutas, el sistema de control del robot utiliza algoritmos simples de control cinemático para guiar al robot a lo largo de la ruta planificada. El control cinemático diferencial ajusta las velocidades de las ruedas izquierda y derecha para que el robot siga el camino generado por el algoritmo A^* o el replanteamiento dinámico.

Estos algoritmos de control son esenciales para garantizar que el robot pueda seguir la ruta de manera eficiente y ajustar su dirección según sea necesario para evitar obstáculos.

6. Diagramas de flujo y pseudocódigo

El siguiente diagrama representa el flujo de datos para que el robot genere movimientos en el mapa, y por tanto, también planifique sus rutas de manera dinámica.

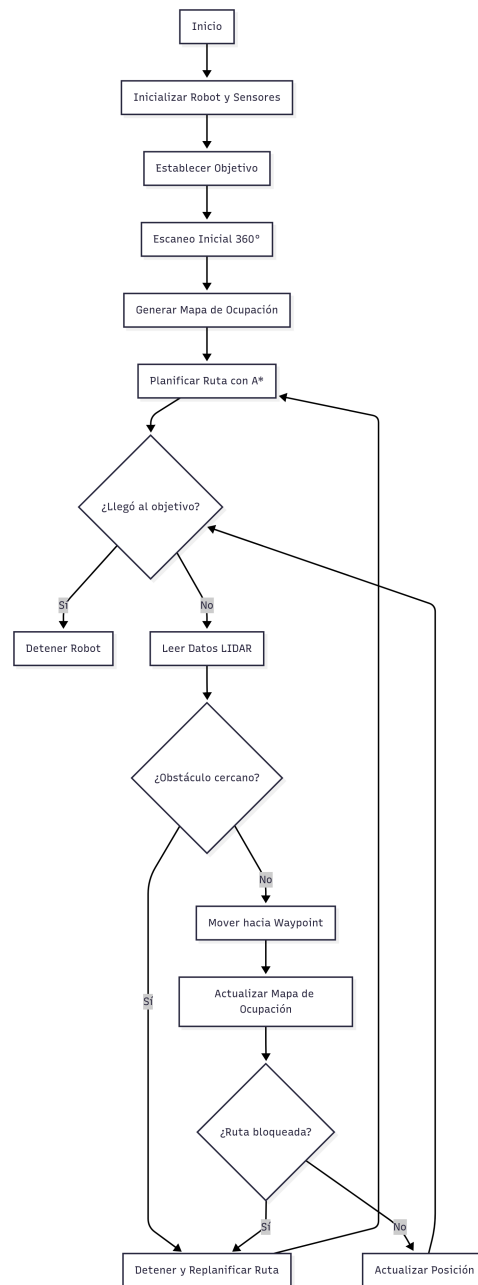


Figura 8: Diagrama de flujo

7. Resultados obtenidos

7.1. Métricas de Desempeño

Las métricas de desempeño del sistema son clave para evaluar la eficiencia, precisión y robustez del robot. A continuación se describen las métricas implementadas:

- **Tiempo total de navegación:** Esta métrica mide el tiempo total en segundos que el robot tarda en completar la navegación desde su punto de inicio hasta el objetivo.
- **Longitud del path (celdas):** La longitud del path se refiere a la cantidad de celdas que forman la ruta generada por el algoritmo A*. Esta métrica nos indica la cantidad de pasos o transiciones que el robot realiza en su desplazamiento.
- **Tiempo de planificación (A*):** El tiempo de planificación se refiere al tiempo, en milisegundos, que tarda el algoritmo A* en calcular la ruta óptima entre el punto de inicio y el objetivo. Este tiempo se mide en cada iteración del ciclo de navegación.
- **Porcentaje del mapa explorado:** Esta métrica indica el porcentaje de celdas del mapa que han sido exploradas y marcadas como ocupadas, es decir, las celdas que contienen obstáculos detectados por los sensores del robot.

7.2. Análisis de Algoritmos

El análisis de los algoritmos implementados se centra en tres aspectos fundamentales: precisión, eficiencia y robustez.

- **Precisión:** El algoritmo A* logra rutas óptimas en escenarios conocidos. Utilizando la distancia Manhattan como heurística, A* garantiza que siempre se calcule el camino más corto desde el punto de inicio hasta el objetivo, evitando obstáculos en el camino.
- **Eficiencia:** La planificación en la grilla 8x8 es rápida, con un tiempo de ejecución promedio por iteración de menos de 50 ms. Esto se debe a que la cantidad de celdas es pequeña, lo que permite que el algoritmo A* explore rápidamente el espacio de búsqueda.
- **Robustez:** La combinación de LIDAR y sensores de distancia permite evitar obstáculos dinámicos de forma eficiente. Estos sensores proporcionan datos en tiempo real sobre el entorno del robot, lo que permite ajustar la planificación de la ruta si se detectan obstáculos imprevistos durante la navegación.

7.3. Reflexión sobre Mejoras

A pesar de que el sistema de navegación actual es eficiente y robusto, existen varias áreas en las que se pueden realizar mejoras para aumentar la capacidad de adaptación y rendimiento en escenarios más complejos:

- **Implementar control proporcional hacia el waypoint:** El control proporcional permitirá que el robot ajuste su velocidad de manera dinámica en función de la distancia al objetivo (waypoint). Esto puede mejorar la precisión en la navegación, ya que el robot reducirá su velocidad a medida que se acerca al destino, evitando movimientos bruscos y aumentando la estabilidad del movimiento.
- **Usar SLAM o un mapeo más denso para escenarios más complejos:** La implementación de SLAM (Simultaneous Localization and Mapping) proporcionaría una ventaja significativa en entornos dinámicos y complejos. El robot podría construir y actualizar su mapa mientras navega, permitiéndole adaptarse mejor a nuevos obstáculos y cambios en el entorno sin depender de un mapa previamente definido.
- **Integrar planificación incremental para ambientes dinámicos:** La planificación incremental ajustaría la ruta del robot en tiempo real sin necesidad de recalcular toda la ruta desde cero. Esto permitiría que el robot reaccione más rápidamente ante cambios en el entorno y optimice el tiempo de computación al no tener que planificar toda la ruta nuevamente en cada ciclo.

8. Conclusiones

Siendo esta la última entrega de la asignatura, nos gustaría aprovechar la oportunidad para comentar sobre algunos aspectos del curso y proponer posibles mejoras para futuras ediciones. Como grupo, creemos que esta etapa del proyecto representa una de las más interactivas y didácticas de la asignatura, por lo que sugerimos que el proyecto se extienda a lo largo de todo el semestre en lugar de concentrarse únicamente en la fase final.

En términos generales, decidimos adoptar la implementación proporcionada por la profesora, ya que coincidía en varios aspectos con nuestras ideas iniciales. Sin embargo, lo que realmente nos impulsó a seguir esta opción fue la amplitud y completitud del proyecto en comparación con la entrega que inicialmente teníamos en mente. A pesar de ello, el tiempo disponible resultó ser un factor limitante, lo que nos llevó a una entrega más ajustada de lo que hubiésemos deseado.

A pesar de los desafíos de tiempo, esta experiencia resultó ser sumamente enriquecedora y entretenida. Nos permitió aplicar conceptos clave de la asignatura de manera práctica. No obstante, es importante destacar que este tipo de proyectos requiere un enfoque metódico, con una planificación cuidadosa de cada etapa, especialmente en la definición de los movimientos y decisiones que el robot debe tomar.

9. Referencias