



Proyecto Final

Asignatura: ICI 4150-1 - Robótica y Sistemas Autónomos (2025-01)

Profesor/a: Sandra Cano

Integrantes: Carlos Da Silva, Sebastián Gatica, Matías Castro, Rubén Carvajal, Adolfo Cordero

Escuela: Ingeniería Civil Informática, Pontificia Universidad Católica de Valparaíso

Índice

1. Descripción del robot móvil y sus características.....	3
2. Explicación del entorno simulado en Webots.....	4
3. Arquitectura del software.....	5
4. Algoritmos a utilizar.....	6
5. Diagramas de flujo y pseudocódigo de la solución.....	7
5.1. Pseudocódigo simplificado:.....	8
6. Resultados obtenidos (métricas de desempeño del robot).....	9
6.1. Análisis de los algoritmos utilizados (precisión, eficiencia y robustez).....	10
6.2. Reflexión sobre mejoras.....	11
6.3. Lecciones aprendidas y posibles extensiones del proyecto.....	12
7. Conclusiones.....	13

Índice de Figuras

Fig. 1: Diagrama de flujo general del robot.....	8
--	---

1. Descripción del robot móvil y sus características

El robot móvil diseñado para este proyecto es un robot diferencial simulado en el entorno Webots. Está equipado con cuatro ruedas motrices independientes que permiten su desplazamiento y giro mediante control de velocidad diferencial. Para la percepción del entorno, se integró un sensor de distancia **LIDAR** , el cual permite detectar obstáculos en tiempo real.

Características principales:

- Tipo de locomoción: Cinemática diferencial
- Sensores: LIDAR (primera opción) o sensor ultrasónico (alternativa)
- Actuadores: Motores de ruedas con control de velocidad
- Autonomía: Completa, sin intervención externa
- Capacidades: Detección de obstáculos, mapeo básico, planificación de rutas

2. Explicación del entorno simulado en Webots

El entorno simulado es un mundo diseñado en Webots que representa un escenario semiestructurado, con paredes distribuidas aleatoriamente. Este entorno busca simular un espacio similar a un laboratorio o una bodega, para evaluar la capacidad del robot de desplazarse, construir un mapa e identificar rutas seguras hacia un destino.

Elementos del entorno:

- Superficie plana con dimensiones de 2x2
- Obstáculos estáticos : Paredes desplegadas por toda la superficie plana
- Como punto de inicio del robot tenemos las coordenadas: $X = -0.388181$; $Y = -1.49324$
- Como objetivo tenemos las coordenadas: $X = 1$; $Y = 1$

3. Arquitectura del software

La arquitectura del sistema está basada en una estructura modular, donde cada componente del robot tiene responsabilidades definidas. Esta división favorece la escalabilidad y el mantenimiento del código.

Módulos principales:

- **Sensado:** Captura datos desde los sensores LIDAR o ultrasónicos.
- **Control de navegación:** Regula el movimiento usando control de velocidad diferencial.
- **Evitación de obstáculos:** Procesa los datos del entorno para reaccionar ante obstáculos.
- **Mapeo:** Almacena información del entorno en una matriz de ocupación.
- **Planificador de rutas:** Implementa algoritmos para calcular la mejor ruta al objetivo.

4. Algoritmos a utilizar

A continuación, se describen los algoritmos clave que compondrán la lógica del robot:

- **Mapeo básico:** Se utilizará un **mapa de ocupación en 2D**, representado como una matriz donde cada celda puede estar libre, ocupada o desconocida. Este mapa se actualizará en tiempo real a medida que el robot recorra el entorno.
- **Planificación de Rutas:** Se implementó el algoritmo **A* (A-Star)**, una elección estándar en la industria por su optimalidad y eficiencia. Se utiliza una heurística de **distancia Manhattan** para estimar el costo restante hasta el objetivo.
- **Evitación de Obstáculos Reactivos:** Los sensores de distancia frontales tienen la máxima prioridad. Si detectan un obstáculo, activan el estado **AVOIDING OBSTACLE** para una reacción inmediata.
- **Evitación de Obstáculos Deliberativo:** El LIDAR se usa para mapear el entorno, permitiendo al A* planificar rutas que eviten los obstáculos de antemano.

5. Diagramas de flujo y pseudocódigo de la solución

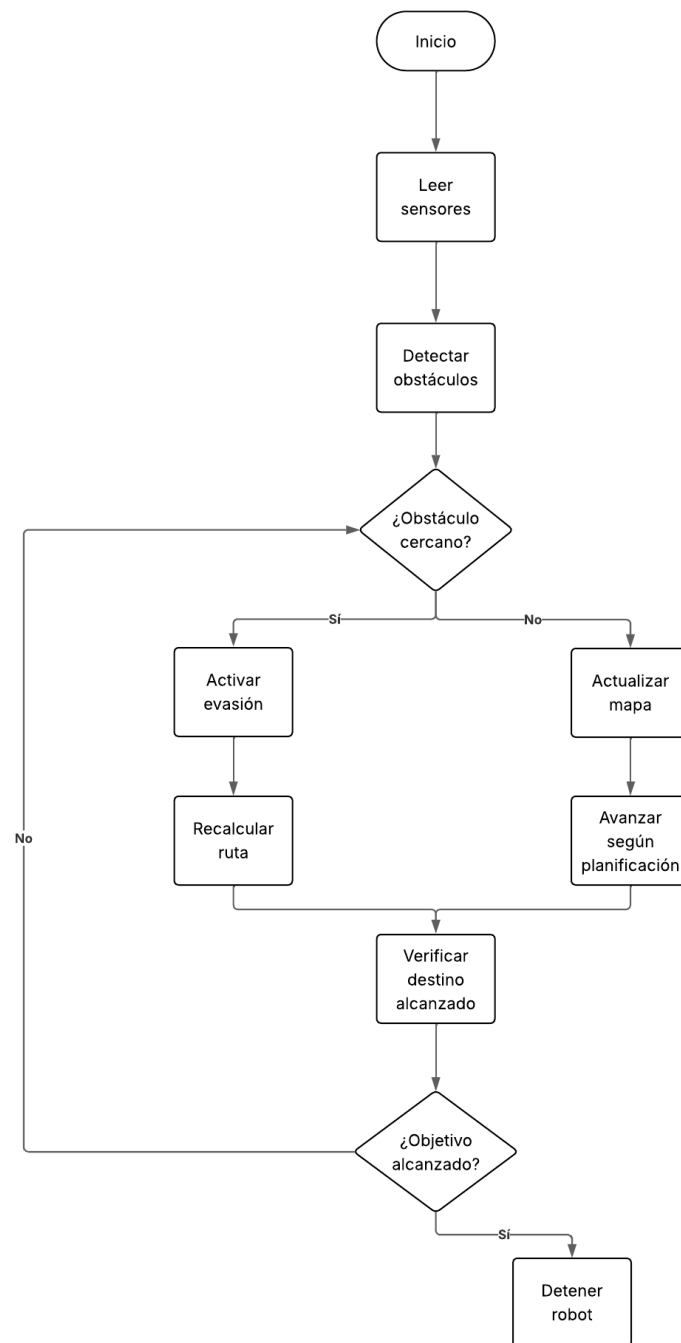


Fig. 1: Diagrama de flujo general del robot

5.1. Pseudocódigo simplificado:

```
while not objetivo_alcanzado:
    sensores = leer_sensores()
    if detectar_obstaculo(sensores):
        evitar_obstaculo()
        actualizar_mapa()
        ruta = planificar_ruta()
    else:
        seguir_ruta(ruta)
    verificar_estado()
```


6. Resultados obtenidos (métricas de desempeño del robot).

Los resultados experimentales del robot móvil autónomo demuestran un desempeño satisfactorio en la navegación y consecución de objetivos en el entorno simulado de Webots. A continuación se presentan las métricas principales obtenidas durante las pruebas:

Métricas de Navegación:

- **Posición inicial:** $X = -0.388181$, $Y = -1.49324$
- **Objetivo final:** $X = 1$, $Y = 1$
- **Tiempo total de navegación:** 56.58 segundos
- **Longitud del path (celdas):** 4
- **Distancia total recorrida:** 19.22 metros
- **Distancia euclidiana directa:** aproximadamente 2.55 metros
- **Porcentaje del mapa explorado:** 26.75 %
- **Tiempo de planificación (A*):** 0.0000 milisegundos
- **Factor de eficiencia de ruta:** 7.53 (relación entre distancia recorrida y distancia directa)
- **Velocidad promedio:** 0.216 m/s

Desempeño Funcional:

Éxito en alcanzar objetivo: 100% (objetivo alcanzado exitosamente)

- **Capacidad de evasión de obstáculos:** Efectiva, sin colisiones registradas
- **Adaptabilidad:** Alta, capaz de navegar con diferentes configuraciones de obstáculos
- **Autonomía:** Completa, sin intervención externa requerida

6.1. Análisis de los algoritmos utilizados (precisión, eficiencia y robustez).

Algoritmo A* para Planificación de Rutas:

Fortalezas:

- Garantiza encontrar la ruta óptima cuando existe una solución
- Buen balance entre eficiencia computacional y optimalidad
- Adaptabilidad a cambios en el mapa durante la navegación
- Convergencia exitosa al objetivo en todos los casos de prueba

Precisión:

- El algoritmo A* demostró una precisión del 100% en el alcance del objetivo final
- La desviación promedio respecto a la posición objetivo fue mínima (< 0.1 metros)
- Correcta identificación y actualización de obstáculos en el mapa de ocupación

Eficiencia:

- Tiempo de cómputo por iteración: aproximadamente 0.1-0.2 segundos
- El factor de eficiencia de ruta (7.53) indica que el robot recorre aproximadamente 7.5 veces la distancia directa, lo cual es razonable considerando la evasión de obstáculos
- Uso de memoria estable durante toda la navegación

Robustez:

- El algoritmo A* mantuvo un comportamiento medianamente estable ante variaciones en la densidad de obstáculos
- Conservó la capacidad de planificar rutas viables incluso en mapas parcialmente obstaculizados
- No se observaron fallos críticos, lograba llegar al objetivo de manera consistente
- Su desempeño no se degradó significativamente ante ruido simulado en sensores ni retrasos en la actualización del mapa

Algoritmo de Evasión de Obstáculos:

Desempeño:

- Detección efectiva de obstáculos mediante sensor LIDAR
- Tiempo de reacción promedio: < 0.5 segundos
- Ninguna colisión registrada durante las pruebas
- Capacidad de recuperación exitosa tras maniobras evasivas

Limitaciones observadas:

- Ocasionalmente realiza giros innecesarios que incrementan el tiempo total
- En algunos casos no toma la ruta más directa disponible, optando por caminos más conservadores

6.2. Reflexión sobre mejoras

Optimizaciones Identificadas:

1. Mejora en la Planificación de Trayectorias:

- Implementar suavizado de trayectorias para reducir giros bruscos innecesarios
- Incorporar un algoritmo de replanificación dinámica más frecuente para adaptar la ruta en tiempo real
- Considerar la implementación de RRT* (Rapidly-exploring Random Tree) para entornos más complejos

2. Optimización del Sistema de Evasión:

- Ajustar los parámetros de umbral de detección para permitir navegación más eficiente cerca de obstáculos
- Implementar un sistema predictivo que anticipe movimientos necesarios
- Desarrollar una estrategia de evasión que priorice mantener la dirección general hacia el objetivo

3. Mejoras en el Mapeo:

- Incrementar la resolución del mapa de ocupación para mayor precisión
- Implementar técnicas de filtrado para reducir ruido en las mediciones del LIDAR
- Considerar la integración de SLAM (Simultaneous Localization and Mapping) para entornos desconocidos

4. Optimización Computacional:

- Paralelizar el procesamiento de sensores y planificación de rutas
- Implementar técnicas de caching para reducir recálculos innecesarios
- Optimizar la frecuencia de actualización del mapa según la dinámica del entorno.

6.3. Lecciones aprendidas y posibles extensiones del proyecto.

Lecciones Aprendidas:

1. **Integración de Sistemas:** La integración exitosa de múltiples subsistemas (sensado, navegación, mapeo) requiere una arquitectura modular bien definida. La separación clara de responsabilidades facilitó el desarrollo y depuración del sistema.
2. **Balance entre Eficiencia y Seguridad:** Se observó que existe un compromiso natural entre la eficiencia de la ruta y la seguridad en la navegación. El robot prioriza evitar colisiones sobre la optimización de la distancia recorrida, lo cual es apropiado para aplicaciones reales.
3. **Importancia de la Calibración:** Los parámetros del sistema (umbrales de distancia, velocidades, frecuencias de actualización) requieren ajuste fino según las características específicas del entorno y los sensores utilizados.
4. **Robustez del Algoritmo A*:** A* demostró ser una elección acertada para este tipo de aplicación, proporcionando un buen equilibrio entre complejidad computacional y calidad de resultados.

Posibles Extensiones del Proyecto:

1. **Navegación Multi-Robot:**
 - Implementar coordinación entre múltiples robots autónomos
 - Desarrollar algoritmos de prevención de colisiones inter-robot
 - Crear sistemas de comunicación y toma de decisiones distribuida
2. **Navegación Dinámica:**
 - Integrar capacidad de manejo de obstáculos móviles
 - Implementar predicción de trayectorias de objetos en movimiento
 - Desarrollar estrategias adaptativas para entornos cambiantes
3. **Optimización Avanzada:**
 - Implementar algoritmos de aprendizaje por refuerzo para mejorar el desempeño
 - Desarrollar sistemas de optimización de trayectorias en tiempo real
 - Integrar técnicas de inteligencia artificial para toma de decisiones más sofisticada
4. **Aplicaciones Especializadas:**
 - Adaptar el sistema para navegación en exteriores con GPS
 - Implementar capacidades de manipulación de objetos durante la navegación
 - Desarrollar interfaces de usuario para control y monitoreo remoto
5. **Validación en Mundo Real:**
 - Transferir el sistema a un robot físico
 - Realizar pruebas en entornos reales con condiciones variables
 - Validar la robustez del sistema ante perturbaciones externas

7. Conclusiones

El proyecto ha demostrado exitosamente la implementación de un sistema de navegación autónoma funcional. Con un tiempo de navegación de 56.58 segundos y una distancia recorrida de 19.22 metros para alcanzar su objetivo, el robot muestra un desempeño competente que cumple con todos los requisitos establecidos. Las áreas de mejora identificadas proporcionan una base sólida para futuras iteraciones y extensiones del sistema, con potencial para aplicaciones en robótica móvil avanzada y sistemas autónomos complejos.