



DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

UNIVERSIDAD DE CHILE

EL7009-1 ROBÓTICA MÓVIL

NAVEGACIÓN: STACK DE NAVEGACIÓN AUTÓNOMA

INFORME PROYECTO FINAL

Autor: Matías Urrea
Profesores: Javier Ruiz del Solar
Felipe Inostroza
Auxiliar: Francisco Leiva
Ayudantes: Eduardo Jorquera
Gonzalo Olguín

Fecha: 22 de julio de 2025
Santiago, Chile

1. Introducción

En este proyecto se buscará implementar una combinación de dos algoritmos de planificación, siendo RRT* (*Rapidly-exploring Random Tree*) el algoritmo utilizado para la planificación global, y DWA (*Dynamic Window Approach*) el algoritmo dedicado a la planificación local. Para esto se generó cada algoritmo por separado y se realizaron las pruebas necesarias para verificar el funcionamiento de ambos.

Luego fue posible incorporar ambos algoritmos para planificar la trayectoria de un robot diferencial, el cual fue visto en trabajos anteriores. Para su correcto funcionamiento, se utilizó un mapa y los sensores LiDAR entregados por el cuerpo docente en trabajos anteriores.

Finalmente, la integración de ambos algoritmos fue exitosa, por lo que fue posible generar ambas planificaciones y el robot fue capaz de llegar a la pose objetivo.

Índice de Contenidos

1. Introducción	1
2. Metodología	3
2.1. RRT*	3
2.2. DWA	3
3. Resultados	4
4. Anexos	9

Lista de Figuras

2. Metodología

A continuación, se presenta la metodología del proyecto, donde se presentarán ambos algoritmos con sus configuraciones y sus resultados correspondientes.

2.1. RRT*

En un inicio, se generó en *Python* el código capaz de simular el algoritmo de RRT*, es decir, el algoritmo RRT optimizado. Para esto, se reutilizó el código generado en trabajos anteriores sobre RRT y se hicieron los cambios correspondientes para la optimización del mismo.

La optimización del algoritmo consiste en asignar costos a los caminos que se van generando, con el fin de compararlos y escoger el camino más corto. Además, el algoritmo genera un 'rewiring', lo que implica que además de escoger el nuevo camino más barato, se examinan los nodos vecinos con el fin de buscar si existe una configuración con un camino aún más corto. Esto permite establecer trayectorias más eficientes que las generadas en un algoritmo RRT común.

Con respecto a los parámetros, se escogieron acorde al mapa, por lo que la distancia transversal es de 0.3, con un máximo de 5000 iteraciones.

2.2. DWA

Luego de generado el algoritmo de planificación global, es necesario configurar un algoritmo de planificación local, donde, como se mencionó anteriormente, se escogió *Dynamic Window Approach*. Este algoritmo, como su nombre lo dice, genera ventanas dinámicas, es decir, toma un rango de valores de velocidad, tanto lineal como tangencial, y con éstas genera una trayectoria al simular que el robot se mueva con estos valores en una cierta ventana de tiempo. Luego de generadas las simulaciones correspondientes, se evalúan dándoles un puntaje, que depende de la alineación de la pose con el objetivo, de la velocidad lineal y de la distancia mínima hacia un obstáculo. La siguiente ecuación modela la función puntaje:

$$G(v, w) = \sigma(\alpha \cdot \text{heading}(v, w) + \beta \cdot \text{dist}(v, w) + \gamma \cdot \text{vel}(v, w))$$

Donde *heading* corresponde a la función que determina el ángulo que forma el robot con respecto al objetivo, *dist* corresponde a la distancia mínima a un obstáculo, *vel* corresponde a la velocidad lineal, y los demás valores son constantes que denotan los pesos de cada función. Los valores finales están dados por:

$$0,5 \cdot (1,4 \cdot \text{heading}(v, w) + 1,0 \cdot \text{vel}(v, w) + 0,1 \cdot \text{dist}(v, w))$$

La distancia tiene un factor pequeño dado que si la distancia es muy pequeña (generando un choque) el puntaje se vuelve menos infinito automáticamente, por lo que para otros valores (como infinito positivo) se debe reducir.

Con esto en cuenta, fue posible generar el algoritmo en *Python*, donde fueron necesarias algunas configuraciones.

En primera instancia, se deben definir los límites de las velocidades y los límites de la ventana, es

decir, la aceleración. Esto depende del ambiente en el que se encuentre el robot, en nuestro caso, las velocidades deben tomar valores pequeños (entre 0.1 y 0.3) ya que se miden en metros por segundo y radianes por segundo.

Luego, con respecto a lo anterior, el robot habrá llegado al objetivo siempre que se encuentre a una distancia de 0.2 m o menos, esto con el fin de ir actualizando los objetivos entregados por el algoritmo RRT*.

Sobre las ventanas de integración, se evaluarán 10 puntos entre las velocidades mínimas y máximas posibles.

Finalmente, es posible realizar las pruebas de ambos algoritmos.

3. Resultados

A continuación se presentan algunos resultados del algoritmo RRT* para diferentes puntos en el mapa a evaluar.

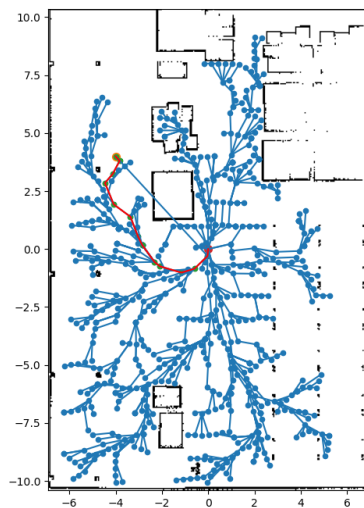


Figura 1: Resultado de RRT* al punto $(-4,4)$

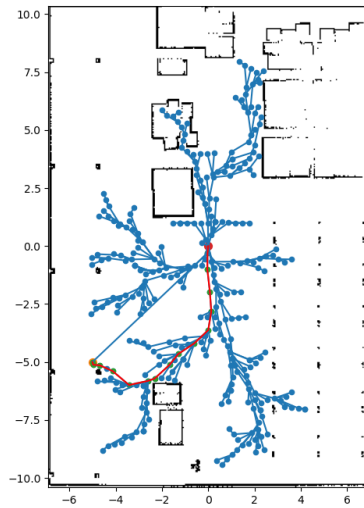


Figura 2: Resultado de RRT* al punto (-5,-5)

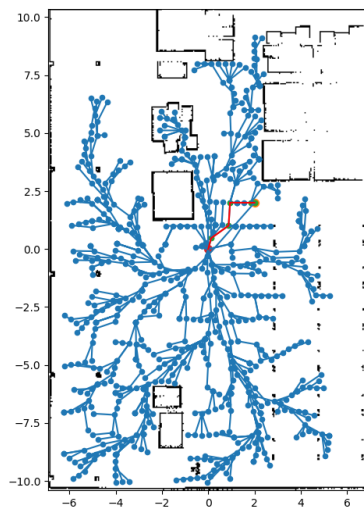


Figura 3: Resultado de RRT* al punto (2,2)

Es posible notar que en algunos casos no es capaz de identificar los obstáculos, para esto, hay dos soluciones, disminuir la distancia transversal (lo que significa mayor tiempo de cómputo) o aumentar el número de puntos entre nodos que se verifican en caso de posible obstáculo intermedio. Luego, se presentan los resultados de DWA para los mismos objetivos.

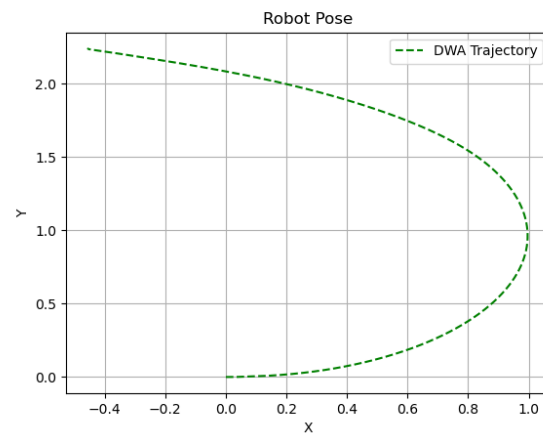


Figura 4: Trayectoria del DWA al objetivo $(-4,4)$

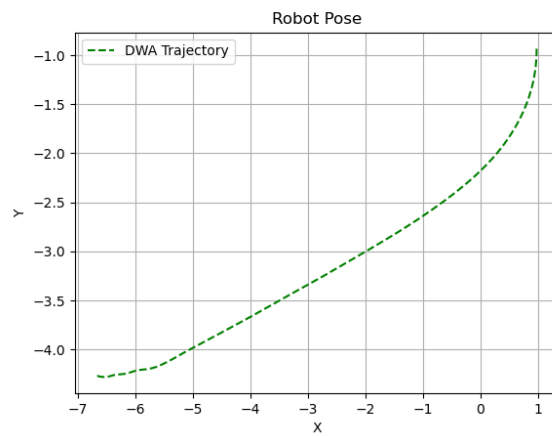


Figura 5: Trayectoria del DWA al objetivo $(-5,-5)$

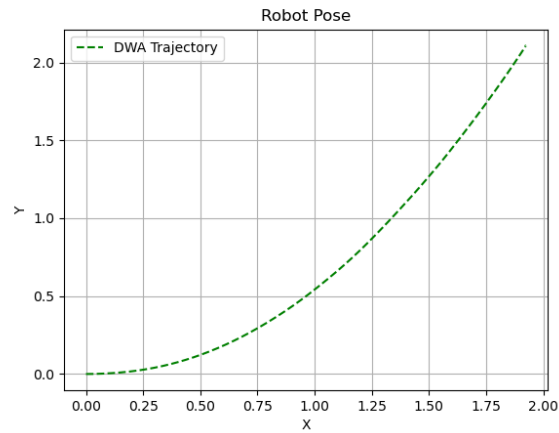
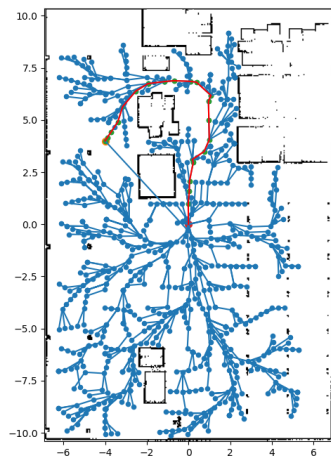
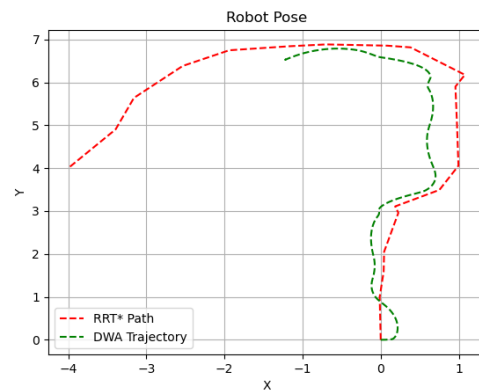


Figura 6: Trayectoria del DWA al objetivo (2,2)

Se puede apreciar que el algoritmo DWA no es capaz de generar una buena predicción cuando el objetivo se encuentra a una distancia lejana, provocando choques o tomando rutas erróneas. Finalmente, se tienen los resultados de la implementación de ambos algoritmos, donde se espera que la planificación global apoye a la planificación local.



(a)



(b)

Figura 7: Resultados finales al objetivo (2,2)

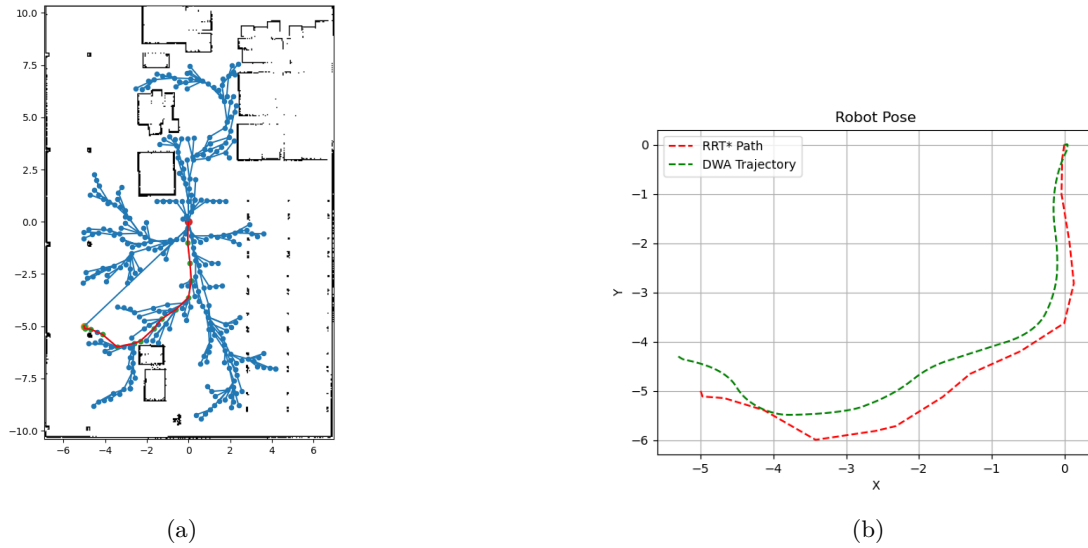


Figura 8: Resultados finales al objetivo (2,2)

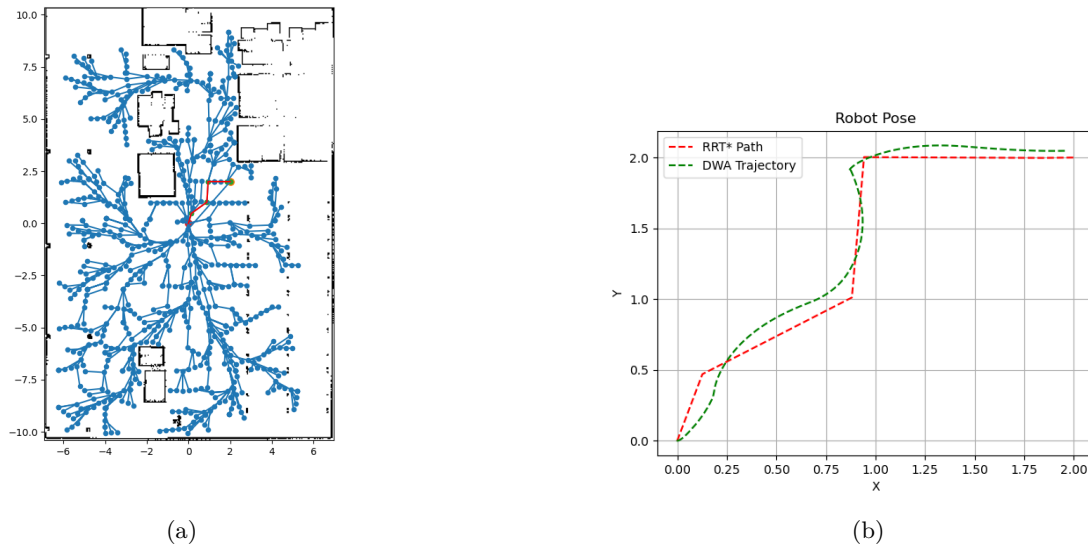


Figura 9: Resultados finales al objetivo (2,2)

Como se pudo ver anteriormente, para la planificación local de la última prueba con objetivo en $(-4,4)$ no fue posible llegar al objetivo. Se estima que esto es debido a que el robot debe pasar por una zona estrecha, por lo que disminuir el costo de la distancia mínima hacia un obstáculo mejoraría el problema. Sin embargo, para las demás pruebas, el robot logra llegar al objetivo. Además, se obtuvieron mejoras al fusionar ambos algoritmos, que era lo esperado. Finalmente, se puede mejorar el desempeño del algoritmo disminuyendo la distancia que toma una posición objetivo lograda y los parámetros de la función de puntaje de las trayectorias generadas por DWA.

4. Anexos