

Sistema FIEB



PELO FUTURO DA INOVAÇÃO

CENTRO UNIVERSITÁRIO SENAI CIMATEC

## Relatório Final

# Implementação do Planejador Probabilistic Roadmap

Apresentada por: Mateus Zarth Seixas

Orientado por: Prof. Marco Reis, M.Eng.

Dezembro de 2021

Mateus Zarth Seixas

# **Implementação do Planejador Probabilistic Roadmap**

Salvador  
Centro Universitário SENAI CIMATEC  
2020

---

# Sumário

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	1
1.2	Justificativa . . . . .	2
1.3	Organização do documento . . . . .	2
<b>2</b>	<b>Fundamentação Teórica</b>	<b>3</b>
2.1	Turtlebot3 . . . . .	3
2.2	Probabilistic Roadmap . . . . .	4
2.3	Move Base . . . . .	4
<b>3</b>	<b>Desenvolvimento do projeto</b>	<b>6</b>
3.1	Simulação . . . . .	6
3.2	Prática . . . . .	6
<b>4</b>	<b>Resultados</b>	<b>8</b>
<b>5</b>	<b>Conclusão</b>	<b>9</b>
	<b>Referências</b>	<b>10</b>

---

## Lista de Figuras

---

1.1	Turtlebot3 . . . . .	1
2.1	Turtlebot3 e seus Componentes . . . . .	3
2.2	Probabilistic Roadmap . . . . .	4
2.3	Move Base . . . . .	5
3.1	Simulação no Gazebo . . . . .	7
3.2	Simulação no Gazebo . . . . .	7

---

## Lista de Tabelas

---

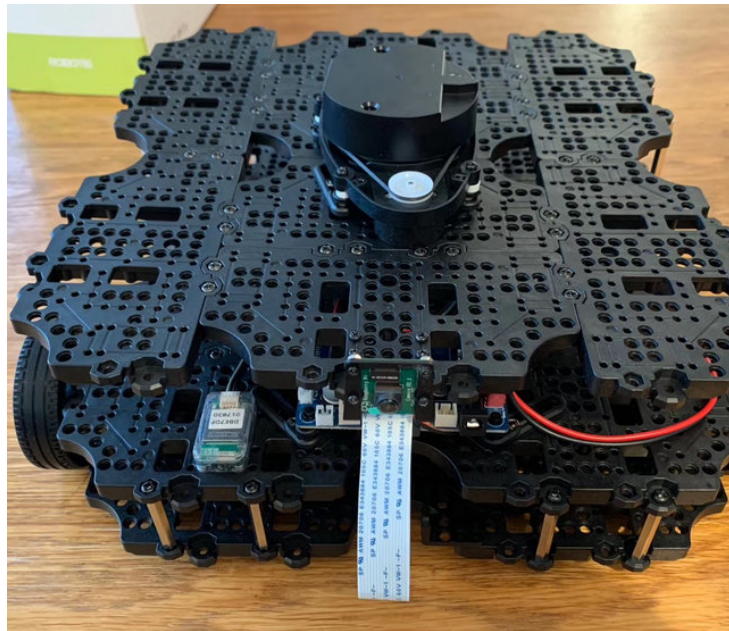
---

## Introdução

---

Nesse estudo foi feita a implementação do algoritmo de planejamento de trajetória Probabilistic Roadmap (PRM) em um robô diferencial não-holonômico chamado Turtlebot3 para permitir a navegação autônoma desse robô. Os testes foram realizados em ambiente de simulação e em um labirinto real construído.

Figura 1.1: Turtlebot3



Fonte: Autoria própria.

### **1.1** *Objetivos*

O objetivo desse estudo foi fazer a implementação do algoritmo de planejamento de trajetória PRM no Turtlebot3 para ser integrada à navegação desse robô como planejador global, a fim de que esse robô pudesse ser capaz de navegar autonomamente em um labirinto, após mapeado, indo de um ponto a outro sem colidir com os obstáculos.

## **1.2    *Justificativa***

Em um trabalho futuro, o algoritmo de planejamento aqui desenvolvido será comparado com outras técnicas de planejamento e trajetória, como  $A^*$ ,  $D^*$  e Dijkstra, para comparar seus resultados estatisticamente.

## **1.3    *Organização do documento***

Este documento apresenta 5 capítulos e está estruturado da seguinte forma:

- **Capítulo 1 - Introdução:** É apresentado o estudo realizado, com a descrição do problema, objetivo e justificativa.;
- **Capítulo 2 - Fundamentação Teórica:** É apresentada a base teórica que sustenta o estudo desenvolvido;
- **Capítulo 3 - Desenvolvimento do projeto:** Nesse capítulo é apresentado o desenvolvimento realizado para implementar o planejador;
- **Capítulo 4 - Resultados:** Nesse capítulo são apresentados os resultados obtidos;
- **Capítulo 5 - Conclusão:** Apresenta as conclusões do trabalho e trabalhos futuros.

## Fundamentação Teórica

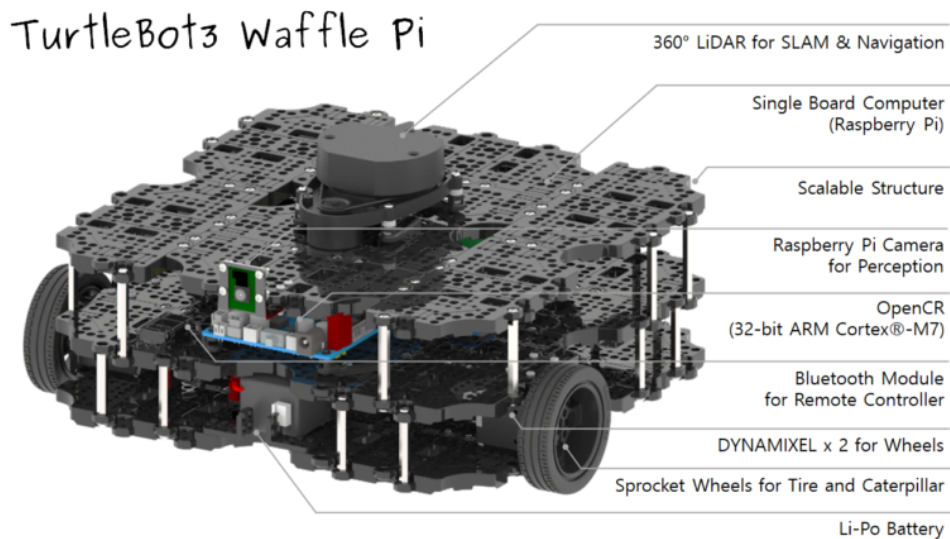
O estudo tinha como objetivo fazer a implementação do planejador de trajetória PRM no robô diferencial Turtlebot3. Nesse capítulo serão apresentadas as características do robô e do planejador de trajetória Probabilistic Roadmap.

### 2.1 Turtlebot3

O Turtlebot3 é um robô diferencial não-holonômico desenvolvido pela empresa Robotis e tem como o ambiente de desenvolvimento padrão o ROS (Robot Operating System). O Turtlebot3 tem como unidade central de processamento uma Raspberry Pi e o sistema operacional instalado nela foi o Ubuntu 20.04 com o ROS Noetic.

O Turtlebot3 tem 2 modelos, o Burger e o Waffle Pi. O modelo escolhido para esse desenvolvimento foi o Waffle Pi, que conta o motor DYNAMIXEL (XM430-W210-T) e uma câmera Raspberry Pi, componentes que o diferenciam do modelo Burger, além do seu formato. O Turtlebot3 conta com um sensor de escaneamento a laser LiDar, uma OpenCR, módulo Bluetooth e uma bateria Li-Po. Pode-se ver o robô e seus componentes na Figura 2.1.

Figura 2.1: Turtlebot3 e seus Componentes



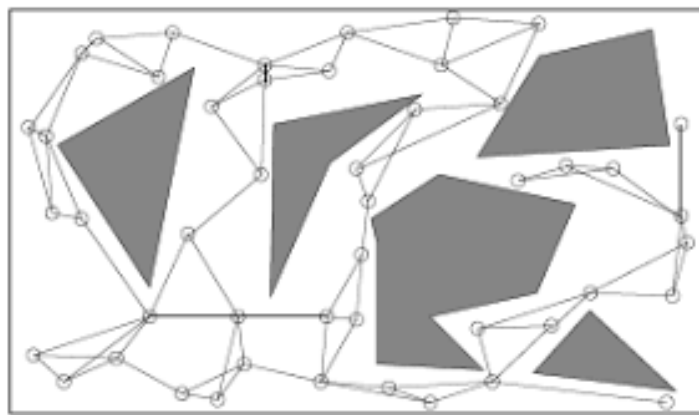
Fonte: Autoria própria.



## 2.2 Probabilistic Roadmap

O algoritmo Probabilistic Roadmap é um planejador de trajetória que permite que um robô se desloque de um ponto inicial a um ponto final sem a interferência de um operador evitando a colisão com obstáculos. A ideia do planejador é gerar pontos aleatórios no mapa, verificando se esses pontos são espaços livres ou não, e tentando conectar eles com os outros pontos mais próximos já gerados, caso não tenha nenhuma barreira entre esses pontos. A medida que o número de pontos cresce, vão se formando trajetórias entre todos os espaços livres do mapa, como mostrado na Figura 2.2.

Figura 2.2: Probabilistic Roadmap



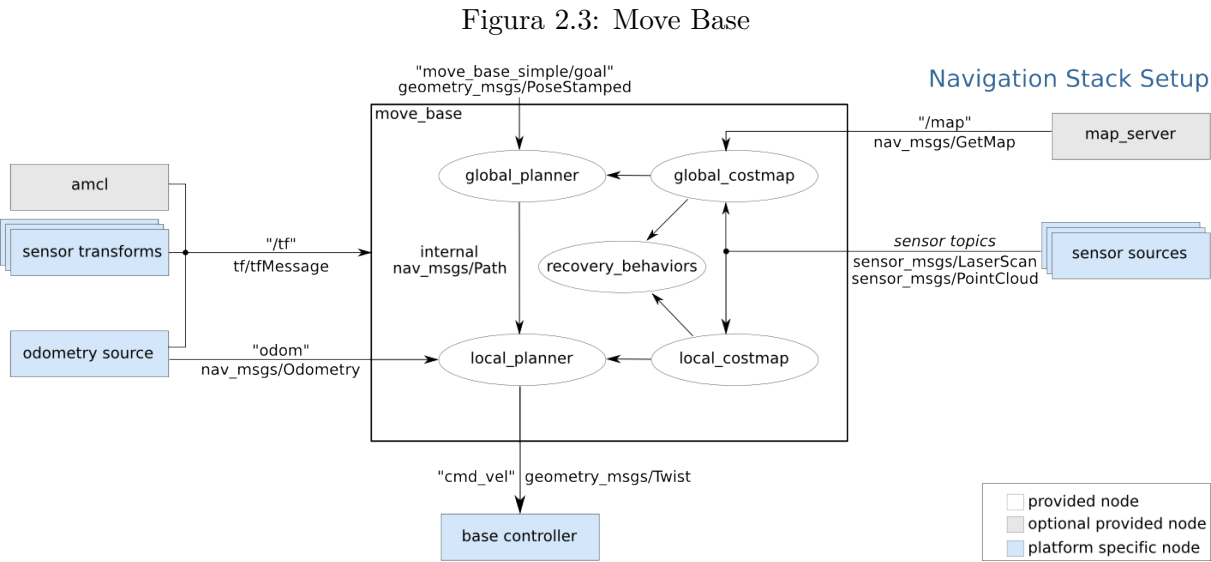
Fonte: Autoria própria.

O algoritmo PRM tem duas fases, uma fase de construção do mapa de rotas e uma fase de questionamento. Na fase de construção, são gerados nós em pontos aleatórios do mapa que estejam em espaços livres. Cada nó desse é então ligado a número  $k$  de nós mais próximos, verificando se não há colisão no caminho. O mapa de rotas é construído então de maneira incremental e armazenado.

Na fase de questionamento, são declarados no algoritmo o ponto inicial e o ponto final do robô, sendo utilizado alguma técnica para calcular o caminho de menor custo. O algoritmo utilizado nesse caso foi o  $A^*$ .

## 2.3 Move Base

O pacote de navegação para ROS utilizado para auxiliar nessa funcionalidade do robô é o Move Base. Através dele é possível declarar uma ação para o robô se deslocar de uma posição inicial para uma posição final utilizando um planejador global e um planejador local. Nesse estudo o planejador global que será utilizado é o planejador de trajetória PRM. Na Figura 2.3 é possível ver um esquemático de como o Move Base funciona.



Fonte: Autoria própria.

---

## Desenvolvimento do projeto

---

O primeiro passo para o desenvolvimento desse estudo foi a montagem do robô Turtlebot3 e a realização do tutorial disponível no site ([ROBOTIS, 2021](#)). Foi instalada a imagem do Ubuntu 20.04 na Raspberry Pi, o ROS Noetic, as dependências necessárias e os pacotes do Turtlebot3 `ros-noetic-dynamixel-sdk`, `ros-noetic-turtlebot3-msgs` e o `ros-noetic-turtlebot3`.

Para implementar o planejador de trajetória PRM foi feita inicialmente uma busca das soluções já disponíveis na internet. Foi encontrado um pacote no GitHub no Link ([MWSWARTWOUT, 2015](#)) que implementa o planejador PRM como Plugin no ROS. O pacote foi desenvolvido para outra versão de ROS, foram necessário fazer algumas modificações no mesmo e também instalar algumas dependências necessárias: `occupancy_grid_utils`, `ros_noetic_tf2_bullet`, `ros_noetic_ompl` e `ros_noetic_ompl_dbgsym`.

No site <http://wiki.ros.org/navigation/Tutorials/Writing%20A%20Global%20Path%20Planner%20As%20Plugin%20in%20ROS> é encontrado um tutorial de como desenvolver o planejador como Plugin e como utilizá-lo. Para selecionar o PRM como planejador global do Move Base é necessário alterar o parâmetro `base_global_planner` do pacote `turtlebot3_navigation` para o PRM.

Foi desenvolvido um repositório no GitHub com o planejador PRM com as modificações e o tutorial de como instalar as dependências necessárias. O repositório se encontra nesse site [https://github.com/seixasxbr/prm\\_planner](https://github.com/seixasxbr/prm_planner).

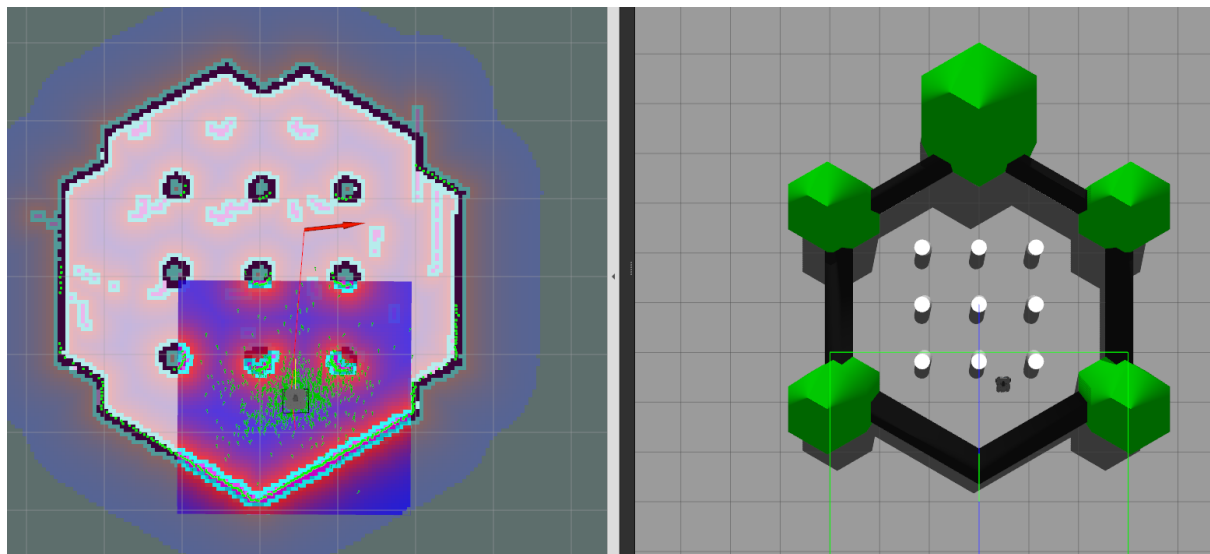
### 3.1 Simulação

A simulação do sistema foi realizada no software Gazebo para validar o funcionamento do planejador de trajetórias PRM. A simulação pode ser vista na Figura 3.1.

### 3.2 Prática

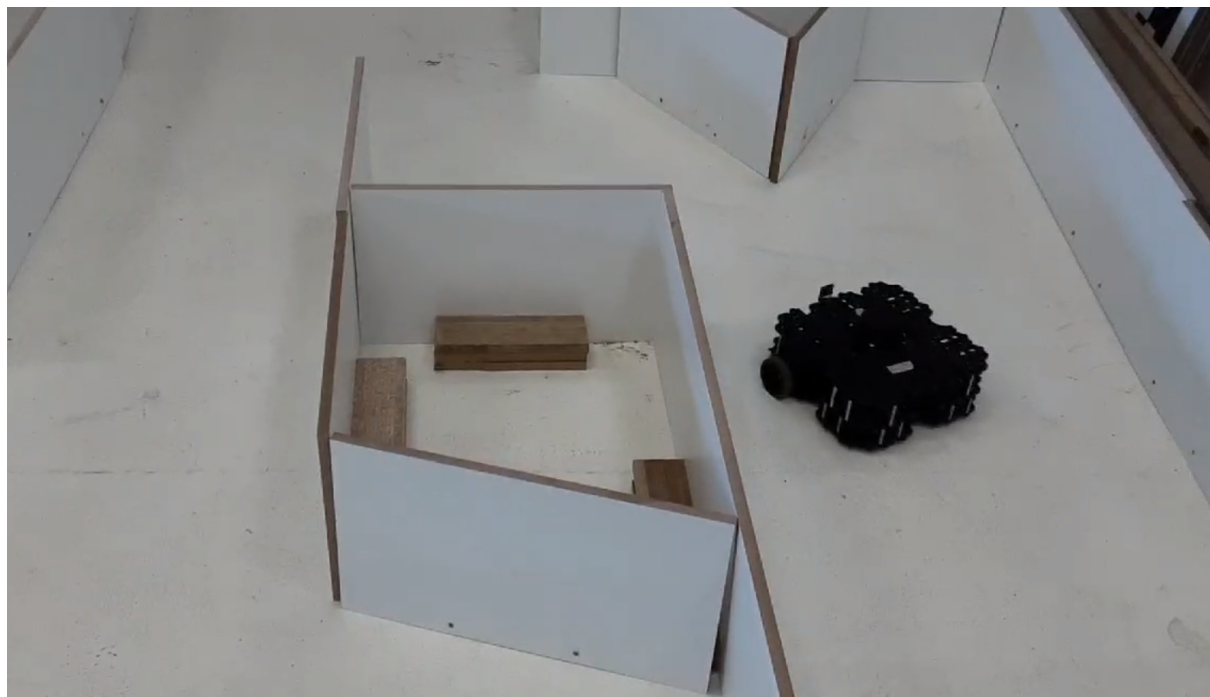
O planejador foi testado também no mundo real, sendo utilizado um labirinto para avaliar a navegação autônoma do robô utilizando o PRM como planejador global. A foto do experimento pode ser visto na Figura 3.2.

Figura 3.1: Simulação no Gazebo



Fonte: Autoria própria.

Figura 3.2: Simulação no Gazebo



Fonte: Autoria própria.

---

## Resultados

---

Foi possível implementar o planejador Probabilistic Roadmap (PRM) no Turtlebot3, carregando o algoritmo desse planejador como Plugin e sendo integrado ao pacote de navegação `move_base`. Na simulação o planejador apresentou bons resultados, conseguindo lidar com os obstáculos na trajetória do robô. Na prática, o algoritmo funcionou, entretanto para alguns trajetos, principalmente que envolviam curvas fechadas, o planejador não conseguiu em alguns casos alcançar a solução. Alguns parâmetros do `move_base` podem ser reconfigurados para tentar achar uma melhor implementação do planejador.

---

## Conclusão

---

O estudo realizado trouxe maior entendimento sobre as funcionalidades de planejamento de trajetória e navegação, além de trazer conhecimento sobre o funcionamento do planejador Probabilistic Roadmap. Em ambiente de simulação o planejador teve bons resultados, porém na prática, para algumas trajetórias mais complicadas, o planejador não conseguiu obter solução, mas conseguiu lidar bem com trajetórias simples. Alguns parâmetros do `move_base` podem ser ajustados para encontrar um melhor resultado da implementação do algoritmo. Para trabalhos futuros o serão implementadas os ajustes dos parâmetros e o resultado obtido com esse planejador será comparado com outros planejadores, como A\*, D\* e Dijkstra, para comparar seus resultados estatisticamente.

---

## Referências

---

MWSWARTWOUT. *PRM Planner*. 2015. <[https://github.com/mwswartwout/turtlebot\\_prm](https://github.com/mwswartwout/turtlebot_prm)>. Accessed: 2021-12-19. Citado na página 3.

ROBOTIS. *Turtlebot3 e-Manual*. 2021. <<https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>>. Accessed: 2021-12-19. Citado na página 3.

*Implementação do Planejador*  
*Probabilistic Roadmap*

Mateus Zarth Seixas

Salvador, Dezembro de 2021.