

NAVEGAÇÃO AUTÔNOMA: DESEMPENHO E DESAFIOS EM ROBÔS MÓVEIS UTILIZANDO ROS E ROS2

Marcus Vinícius Rodrigues Araújo¹; Rebeca Tourinho Lima²

¹ Bolsista; Centro de Competência; marcus.araujo@fbter.org.br

² Centro Universitário SENAI CIMATEC; Salvador - BA; rebeca.lima@fieb.org.br

RESUMO

Este estudo investigou a aplicação de técnicas de navegação autônoma em robôs móveis, utilizando os frameworks ROS e ROS2, Navigation Stack e Navigation 2, além dos algoritmos SLAM GMapping e Cartographer. A metodologia incluiu revisão de literatura, implementação de algoritmos e realização de testes indoor com dois robôs Turtlebot modelo Burger. Os resultados revelaram que tanto o GMapping quanto o Cartographer produziram mapas representativos da arena de testes, sendo o Cartographer mais preciso. Na aplicação dos algoritmos de navegação, tanto o Navigation Stack (ROS) quanto o Navigation 2 (ROS2) demonstraram eficácia, com o Navigation 2 apresentando melhor desempenho. O algoritmo de controle local DWB superou o DWA em termos de tempo de execução e taxa de falhas, destacando a importância da escolha do algoritmo na minimização de falhas e na eficiência da navegação autônoma.

PALAVRAS-CHAVE: Navegação autônoma, Robôs móveis, ROS2, Algoritmos de navegação

1. INTRODUÇÃO

A navegação autônoma de robôs móveis representa um campo de pesquisa em constante evolução na robótica, desempenhando um papel fundamental em diversas aplicações práticas, desde logística até exploração espacial. Essa tecnologia permite que os robôs se movam de forma autônoma em ambientes complexos e dinâmicos, sem intervenção humana direta, aumentando significativamente sua autonomia e utilidade em uma ampla gama de cenários¹.

Neste contexto, o desenvolvimento de frameworks de software eficazes desempenha um papel crucial. O ROS (Robot Operating System) emergiu como um dos principais *frameworks* para o desenvolvimento de sistemas robóticos autônomos, fornecendo uma plataforma flexível e poderosa para a implementação de uma variedade de algoritmos de navegação autônoma. Com o lançamento do ROS2, uma versão mais recente do sistema, surgem novas oportunidades e desafios para a comunidade de pesquisa robótica.

Além disso, o Navigation Stack, um conjunto de pacotes de software do ROS, oferece funcionalidades essenciais para a navegação autônoma, incluindo localização, mapeamento e planejamento de trajetória². Com o advento do ROS2, uma nova versão do Navigation Stack, chamada Navigation 2, foi introduzida, prometendo melhorias significativas em termos de desempenho e escalabilidade.

A tecnologia SLAM (*Simultaneous Localization and Mapping*) é outra ferramenta importante para a navegação autônoma, permitindo que os robôs construam mapas de seus ambientes e se localizem dentro deles em tempo real³. Dentro do ecossistema ROS, dois dos algoritmos SLAM mais proeminentes são o GMapping e o Cartographer, cada um com suas próprias vantagens e limitações⁴.

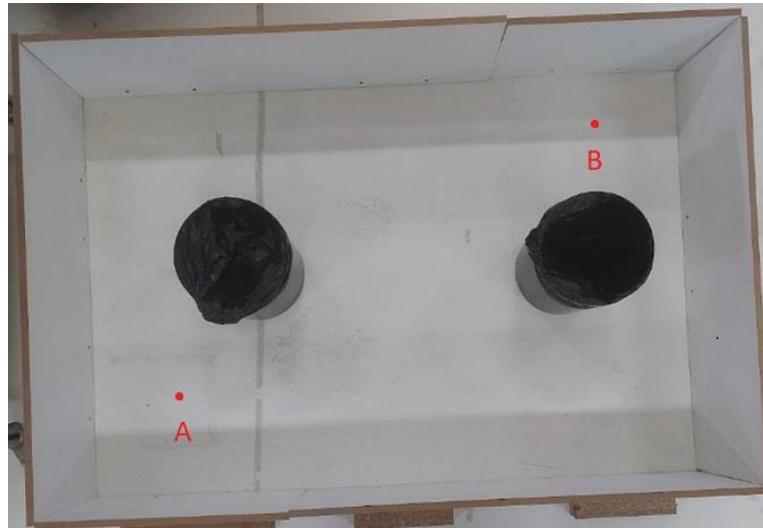
Neste artigo, será apresentada uma análise comparativa das aplicações de navegação autônoma no robô TurtleBot 3 Burger. O objetivo principal é investigar e avaliar a eficácia desses sistemas em diferentes cenários de navegação autônoma, utilizando as funcionalidades oferecidas pelo ROS, ROS2, Navigation Stack e Navigation 2, bem como os algoritmos SLAM GMapping e Cartographer. Espera-se que esta análise forneça insights valiosos para pesquisadores e desenvolvedores interessados em navegação autônoma de robôs móveis e contribua para o avanço contínuo da robótica autônoma.

2. METODOLOGIA

Este estudo foi estruturado em quatro etapas essenciais, iniciando com uma revisão da literatura para situar o estudo no contexto atual e identificar lacunas de conhecimento, especialmente na análise de metodologias de controladores locais para navegação autônoma. Em seguida, os algoritmos de SLAM foram implementados para mapeamento do ambiente controlado, adaptando metodologias consolidadas às peculiaridades do TurtleBot 3. A etapa subsequente envolveu a realização de testes *indoor* de navegação autônoma, nos quais os robôs foram submetidos a experimentos em uma arena controlada. Utilizando dois Turtlebots modelo Burger equipados com ROS Noetic e ROS2 Humble, ambos com uma Raspberry Pi 4 como controlador, foram coletadas 20 amostras de tempo de execução para cada robô e cada algoritmo

implementado, envolvendo a navegação do ponto A ao ponto B, conforme ilustrado na Figura 01, com o controlador determinando o melhor percurso em todas as iterações.

Figura 01 – Arena utilizada para testes com os robôs Turtlebot



Fonte: Autores (2024).

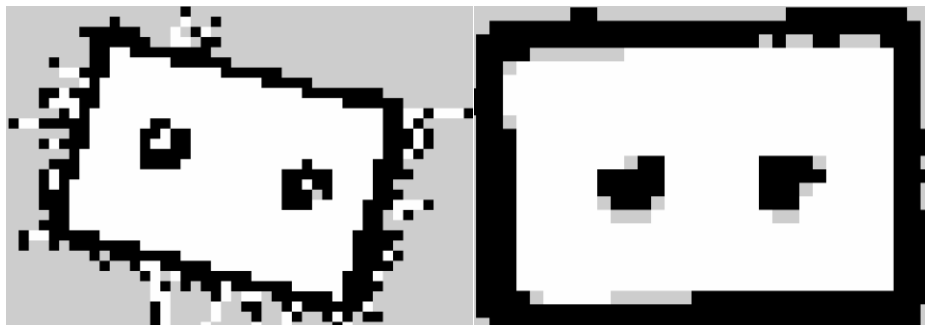
Por fim, os dados coletados foram submetidos a uma análise rigorosa, culminando na interpretação dos resultados e formulação de conclusões embasadas.

3. RESULTADOS E DISCUSSÃO

Através da revisão bibliográfica foi decidido que a melhor aplicação de algoritmos controladores de trajetória, ou planejadores locais, seriam o DWA para ROS e DWB para ROS2¹, pois esses algoritmos têm uma resposta mais rápida e correta. O DWA utiliza o método de janela dinâmica, assim realizando um desvio de obstáculo ótimo⁴. Já o DWB é uma evolução do DWA e dá novas oportunidades de gerenciar melhor as opções e ter ajustes melhores levando em consideração o peso do robô¹.

Na aplicação dos algoritmos de SLAM, tanto o GMapping quanto o Cartographer produziram resultados semelhantes, gerando mapas que representam a arena criada para os testes, conforme demonstrado na Figura 02. Observou-se que o Cartographer produziu um mapa mais limpo, i.e. com menos ruídos, em comparação ao GMapping, que capturou alguns elementos fora da arena, resultando em distorções no resultado do mapa de ocupação.

Figura 02: À esquerda mapa gerado pelo Gmapping. À direita mapa gerado pelo Cartographer.



Fonte: Autor (2024).

No que diz respeito à implementação dos algoritmos de navegação, tanto o Navigation Stack quanto o Navigation 2 revelaram-se robustos e eficazes na aplicação nos robôs estudados. Durante a realização do teste comparativo entre os controladores locais DWB e DWA, conforme detalhado na Tabela 01, que registra o tempo de execução de cada amostra em segundos e destaca as ocorrências de falhas com a letra “F”, observou-se um desempenho diferenciado entre as abordagens. Notavelmente, o DWB, apresentou uma

média de tempo de 22,12 segundos, demonstrou uma capacidade superior de alcançar o objetivo de navegação de forma mais rápida quando comparado ao DWA, que registrou uma média de 32,44 segundos.

Tabela 01: Testes dos Controladores locais

Teste DWB	1	2	3	4	5	6	7	8	9	10
Tempo (s)	F	24,88	34,66	10,9	13,94	F	F	17,56	14,62	11,26
Teste DWB	11	12	13	14	15	16	17	18	19	20
Tempo (s)	F	51,62	15,39	36,53	17,62	24,08	8,67	15,39	36,4	F

Teste DWA	1	2	3	4	5	6	7	8	9	10
Tempo (s)	18,56	25,89	45,34	F	20,52	37,25	67,87	27,59	F	F
Teste DWA	11	12	13	14	15	16	17	18	19	20
Tempo (s)	32,25	19,55	F	42,56	17,25	39,15	F	F	27,9	32,45

No entanto, é importante ressaltar que o algoritmo DWB demonstrou uma incidência apenas ligeiramente inferior de falhas em relação ao DWA. Ao longo do experimento, foram registradas cinco ocorrências de falhas para o DWB, em comparação com seis para o DWA. Das cinco falhas observadas no DWB, duas decorreram da incapacidade do robô em transpor um obstáculo, resultando em uma falha por limite de tempo e consequente interrupção da navegação, que pode ser classificada como uma falha simples e comum. Nas três outras ocasiões, o robô indicou ter completado o percurso, contudo parou antes de alcançar o ponto de destino. Uma análise detalhada do mapa de navegação revelou um desvio do mapa, indicando que os sensores de localização do robô forneceram uma posição errônea, alterando a representação da *pose* do robô no mapa elaborado. Tal situação é considerada uma falha moderada, pois a perda de localização do robô prejudica a precisão da navegação.

Por outro lado, todas as seis falhas registradas no DWA foram decorrentes da inabilidade do algoritmo em criar uma rota viável para o objetivo, destacando algumas limitações do algoritmo DWA no que concerne à geração de rotas viáveis, também sendo categorizadas como falhas moderadas.

Esses resultados ressaltam a maior agilidade de resposta na aplicação do algoritmo DWB em comparação com o DWA, bem como a ocorrência de um número inferior de falhas, e falhas menos significativas, na implementação do DWB. Isso indica a relevância da seleção do algoritmo de navegação na minimização de falhas e na garantia da eficácia do sistema de navegação autônoma.

4. CONSIDERAÇÕES FINAIS

Com base nos resultados e análises, este estudo permitiu avaliar os algoritmos de navegação autônoma DWB e DWA, juntamente com os algoritmos de mapeamento SLAM GMapping e Cartographer, utilizando robôs TurtleBot 3 em um ambiente controlado.

Os testes revelaram que tanto o GMapping quanto o Cartographer foram capazes de gerar mapas representativos do ambiente, com o Cartographer demonstrando maior precisão. Além disso, o algoritmo DWB apresentou melhor desempenho em termos de velocidade e incidência de falhas em comparação com o DWA.

Esses resultados sugerem um avanço nas aplicações em ROS 2 em relação ao ROS, destacando a importância de escolher algoritmos adequados para garantir a eficácia dos sistemas de navegação autônoma.

5. REFERÊNCIAS

- ¹SANT'ANNA, Tiago B.; ARGOLO, Miguel B.; LIMA, Rebeca T. **Comparative analysis in real environment of trajectory controllers on ROS2**. IEEE Latin American Robotics Symposium - LARS, 2023.
- ²LEE, Ung-Gyo; CHOI, Kyung-Jea; PARK, Soon-Yong. **The Design and Implementation of Autonomous Driving Pallet Robot System using ROS**. In: Twelfth International Conference on Ubiquitous and Future Networks (ICUFN), 2021, p. 372-374. DOI: 10.1109/ICUFN49451.2021.9528735.
- ³ZAKARIA, W. N. W.; MAHMOOD, I. A.-T.; SHAMSUDIN, A. U.; RAHMAN, M. A. A.; TOMARI, M. R. M. **ROS-based SLAM and Path Planning for Autonomous Unmanned Surface Vehicle Navigation System**. In: IEEE 5th International Symposium in Robotics and Manufacturing Automation (ROMA), 2022, Malacca, Malaysia, p. 1-6. DOI: 10.1109/ROMA55875.2022.9915665.
- ⁴LIU, B.; GUAN, Z.; LI, B.; WEN, G.; ZHAO, Y. **Research on SLAM Algorithm and Navigation of Mobile Robot Based on ROS**. In: IEEE International Conference on Mechatronics and Automation (ICMA), 2021, Takamatsu, Japão, p. 119-124. DOI: 10.1109/ICMA52036.2021.9512584.