

Navegação Autônoma de Robô Quadrúpede para Inspeção de Falhas Estruturais em Ambiente Industrial

Matheus Della Rocca Martins

Prof. Dr. Thiago Boaventura Cunha

Escola de Engenharia de São Carlos / Universidade de São Paulo

matheus.drocca@usp.br | tboaventura@usp.br

Objetivos

A rápida evolução da robótica, especialmente com o uso de robôs quadrúpedes, tem viabilizado aplicações importantes, como o monitoramento e inspeção de infraestruturas. Assim, a capacidade de navegação autônoma é fundamental, abrangendo desde estratégias para percepção e localização até técnicas para planejamento de trajetórias [1]. O objetivo desse projeto é o desenvolvimento de um sistema capaz de proporcionar a navegação autônoma para robôs quadrúpedes. Visa-se possibilitar o planejamento de trajetórias em tempo real, considerando o desvio de obstáculos estáticos e dinâmicos, e, o mapeamento 2D e 3D do ambiente percorrido. Para análise de resultados, o sistema deve ser testado tanto em um ambiente simulado como em um ambiente real controlado. Por fim, o robô deverá percorrer um pátio industrial autonomamente, de modo a possibilitar a inspeção de falhas estruturais em suportes para bobinas de chapas de aço.

Métodos e Procedimentos

Tendo em vista os objetivos propostos, o robô o robô Go1, desenvolvido pela empresa *Unitree Robotics*, foi escolhido para o projeto. Para a obtenção de uma estimativa robusta e confiável da localização do robô, são utilizados diferentes sensores, incluindo a câmera *Intel® RealSense™ Depth Camera D455*, que fornece informações de profundidade junto com a

imagem RGB, a câmera de rastreamento “*Intel® RealSense™ T265*” e o sensor LiDAR “*Unitree 4D LiDAR L1*”.

O desenvolvimento do Sistema de Navegação foi dividido em duas partes: um módulo para localização e mapeamento; e outro módulo para o planejamento de trajetórias. O *Robot Operating System (ROS)* [2] é utilizado para integração do hardware. A estimativa da posição e orientação a partir do sensor LiDAR é realizada utilizando pacotes baseados em algoritmos de *Iterative Closest Point (ICP)* e *Lidar Odometry and Mapping (LOAM)*, que utilizam abordagens distintas para melhorar a precisão da odometria e do mapeamento. Para a câmera D455, é utilizado o algoritmo de odometria visual implementado pelo pacote *RTAB-Map* [3]. Este pacote também foi utilizado para a integração dos algoritmos de *Simultaneous Localization and Mapping (SLAM)* [4], baseados tanto em dados de câmeras RGB-D como em dados coletados pelo LiDAR, capazes de simultaneamente mapear o ambiente percorrido pelo robô e estimar a sua posição no mapa gerado. Para contornar possíveis fragilidades de cada tipo de sensor são utilizadas técnicas de fusão sensorial, como os estimadores *Extended Kalman Filter (EKF)* e *Unscented Kalman Filter (UKF)*. Por fim, o conjunto de pacotes do *Navigation Stack* é usado para a implementação dos planejadores de trajetória e para a integração dos diferentes pacotes.

Ademais, para validar o sistema será criado um ambiente de simulação no software *Gazebo* e um ambiente real controlado.

Resultados

Pode-se destacar os seguintes resultados parciais obtidos para esse trabalho em andamento: integração e desenvolvimento de um sistema de navegação inicial funcional, análise comparativa de planejadores de trajetória implementados em ambiente simulado e testes iniciais do módulo de localização em ambiente real. Para os testes em simulação, definiu-se uma trajetória desejada e realizou-se cinco simulações para cada configuração escolhida.

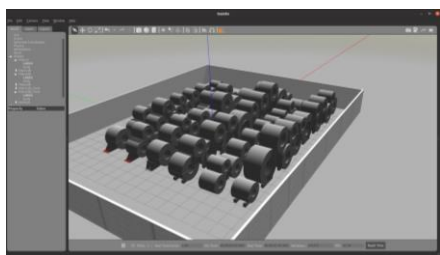


Figura 1: Ambiente de simulação criado

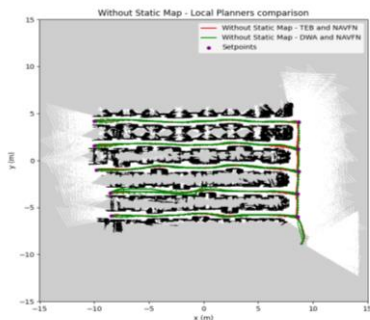


Figura 2: Trajetórias realizadas em simulação

Tabela 1: Comparação entre Planejadores Locais

Planejadores	Algoritmo de planejamento	Tempo médio (s)	Tempo máximo (s)	Tempo mínimo (s)
TEB	Timed Elastic Band	443,11	445,45	440,59
DWA	Dynamic Window Approach	1131,39	1217,55	1067,17

Tabela 2: Comparação entre Planejadores Globais

Planejadores	Algoritmo de planejamento	Tempo médio (s)	Tempo máximo (s)	Tempo mínimo (s)
navfn	Dijkstra's	443,11	445,45	440,59
global_planner	Dijkstra's	437,54	441,88	433,35
global_planner	A*	444,05	459,11	431,66

Conclusões

Os resultados parciais obtidos foram satisfatórios. Assim, considerando o tempo para realização da trajetória e a consistência obtida, foi possível determinar que dentre os planejadores testados, o planejador global "Global Planner", utilizando o algoritmo de Dijkstra's, em conjunto com o planejador local "Teb Local Planner" foi a melhor combinação implementada e se mostrou capaz de realizar as atividades propostas. As próximas etapas do projeto incluem: realização de testes em um ambiente real controlado para validação completa do sistema, otimização do sistema existente e implementação de diferentes pacotes de SLAM a fim de possibilitar uma análise comparativa entre eles.

Agradecimentos

Agradecemos o apoio da Fundação de Apoio A Física e A Química (FAFQ) e do Laboratório de Robôs com Pernas do Centro de Robótica da USP (CRob) para a realização deste trabalho.

Referências

- [1] M. A. K. Niloy et al. (2021). **Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review**. In IEEE Access, 9, 35338-35370, doi: 10.1109/ACCESS.2021.3062557.
- [2] Quigley, Morgan & Conley, Ken & Gerkey, Brian & Faust, Josh & Foote, Tully & Leibs, Jeremy & Wheeler, Rob & Ng, Andrew. (2009). **ROS: an open-source Robot Operating System**. ICRA Workshop on Open Source Software.
- [3] Labbé, M. and Michaud, F. (2019). **Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation**. *Journal of field robotics*, 36(2):416–446.
- [5] T.J. Chong et al (2015). **Sensor Technologies and Simultaneous Localization and Mapping (SLAM)**, *Procedia Computer Science*, Volume 76, 174-179, <https://doi.org/10.1016/j.procs.2015.12.336>.