IMPLEMENTAÇÃO DE SISTEMAS DE NAVEGAÇÃO AUTÔNOMA DE ROBÔS MÓVEIS PARA AMBIENTES INTERNOS

Gustavo Gomes Richter, Gabriel Abatti, André Bittencourt Leal

INTRODUÇÃO

O estudo da navegação autônoma de robôs móveis em ambientes internos exige estratégias robustas de controle e coordenação, especialmente para garantir o deslocamento adequado dos robôs sob diferentes condições. Nesse contexto, o objetivo da pesquisa consiste na implementação e avaliação de estratégias de navegação e coordenação de múltiplos robôs autônomos em ambientes virtuais e reais, com base na Teoria de Controle Supervisório (TCS) (Ramadge; Wonham, 1987), utilizando o TurtleBot 4 como plataforma de pesquisa e desenvolvimento.

A transição para os testes reais foi o foco principal das atividades desenvolvidas nesta etapa do trabalho e incorporou técnicas de localização e mapeamento simultâneos (SLAM), localização baseada em mapa e navegação por meio do Nav2 (Nav2 Documentation, 2026), integradas ao controlador MPPI (*Model Predictive Path Integral*). Após os testes iniciais de implementação dessas ferramentas, foram obtidos resultados satisfatórios nos testes de navegação com dois robôs, realizados por meio do envio manual de comandos de navegação. Esses testes permitiram analisar o comportamento do sistema e os recursos necessários para a implementação de estratégias de controle e coordenação baseadas na TCS, considerando a modelagem do sistema como um Sistema a Eventos Discretos (SED) (Cassandras; Lafortune, 2021), com o objetivo de permitir o envio autônomo de objetivos de navegação e evitar a ocorrência de bloqueios (*deadlocks*).

DESENVOLVIMENTO

A bancada experimental é composta por robôs TurtleBot 4, plataformas de pesquisa baseadas em ROS 2 (*Robot Operating System 2*) (*ROS 2 Documentation*, 2026), que utilizam a base iRobot Create 3, integrada a um computador de bordo Raspberry Pi 4B e a um sensor RPLIDAR (*RoboPeak Light Detection and Ranging*) A1M8 (cobertura de 360° e alcance de 12 m) para a geração do mapa 2D e a localização do robô (Clearpath Robotics, 2026).

Para a comunicação entre múltiplos robôs na mesma rede Wi-Fi, adotou-se a infraestrutura de *middleware* do ROS 2 com o Fast DDS (*Fast Data Distribution Service*). Para evitar a saturação do tráfego de rede por *broadcast*, configurou-se o mecanismo Discovery Server. Cada robô opera com uma instância individual do servidor e *namespaces* próprios (/bt2 e /bt3), permitindo que a estação de controle (notebook central) monitore os tópicos e os nós de ambos os robôs e envie comandos coordenados sem interferência cruzada.

A coordenação de múltiplos robôs fundamenta-se na TCS, empregando a abordagem de Controle Supervisório Modular Local (*Local Modular Supervisory Control* – LMSC) (Queiroz; Cury, 2000), adaptada ao ROS 2 (Jazzy). A arquitetura é organizada em quatro níveis: autômatos de estados finitos que representam os subsistemas da planta e os supervisores modulares locais, sendo estes responsáveis pela habilitação ou desabilitação dos eventos controláveis; um nó central do ROS 2, que acompanha os estados da planta e dispara eventos autorizados por todos os supervisores registrados; uma camada de tradução, que converte o disparo de eventos discretos em ações de navegação no Nav2; e, por fim, a execução contínua de baixo nível nos robôs.

Na arquitetura adaptada ao ROS 2, os eventos discretos correspondem a tópicos publicados, enquanto os mecanismos de registro e habilitação de eventos *(Register Model e Enable Event)* operam por meio de serviços. A sincronização entre /bt2 e /bt3 é realizada por meio da observação de tópicos e da habilitação cruzada de serviços entre as instâncias de cada robô.

RESULTADOS

A validação experimental seguiu um fluxo evolutivo. Inicialmente, enfrentou-se uma fase crítica de parametrização da rede e de atribuição de endereços IP estáticos (Protocolo de Internet) aos robôs TurtleBot 4. Superada essa etapa, realizaram-se os testes de SLAM e navegação individual de cada robô. Durante os primeiros ensaios de teleoperação e de navegação simultânea com dois robôs, observaram-se falhas recorrentes de timeout e de perda de pacotes na comunicação Wi-Fi. Além dos problemas de comunicação, foram identificadas dificuldades no planejamento de rotas relacionadas à camada de inflação (*inflation layer*) do Nav2, cujo raio superdimensionado impedia o planejamento de rotas no ambiente restrito do laboratório.

Após o ajuste fino dos parâmetros do Nav2, obteve-se o funcionamento satisfatório da navegação simultânea de dois robôs, o que permitiu avançar para a implementação da estrutura de controle supervisório no sistema. Os supervisores foram integrados à arquitetura de controle em ROS 2 para coordenar o sequenciamento das tarefas de forma assíncrona. No entanto, a execução física contínua apresentou limitações relacionadas à capacidade computacional dos robôs e à latência da comunicação na infraestrutura de rede baseada no *Discovery Server*.

Para mitigar o processamento excessivo nos robôs, adotou-se a estratégia de transferir o processamento do Nav2 e do RViz (*ROS Visualization*) para a estação central e de utilizar uma rede *Wi-Fi* na faixa de 5 GHz. Apesar da melhoria, a comunicação em ambiente sem fio ainda apresentou atrasos significativos na recepção de eventos discretos, resultando em interrupções esporádicas das rotas devido a timeouts.

Como desdobramento atual, estuda-se a modelagem de um cenário de gerenciamento de estação de recarga (*docking*), no qual múltiplos robôs operam de forma coordenada considerando o limiar de 40% do nível de carga da bateria para solicitar a recarga e o limiar de 90% para retornar à operação, visando à otimização da taxa de trabalho.

CONSIDERAÇÕES FINAIS

Os resultados obtidos com a implementação de supervisores modulares locais em ROS 2 demonstraram a viabilidade de sua aplicação no controle e na coordenação de múltiplos robôs autônomos. Contudo, a comunicação por meio do *middleware* Fast DDS com o mecanismo *Discovery Server*, sobre a rede sem fio, representou o principal gargalo prático para a validação experimental de longa duração com os *TurtleBots 4*.

Como continuidade do trabalho, propõe-se o desenvolvimento de uma arquitetura de implementação e comunicação que utilize a rede Wi-Fi quando necessário, mas que permita que o funcionamento principal de cada robô não dependa dessa comunicação. Além disso, estão previstos testes envolvendo comportamentos mais complexos dos robôs, incluindo o gerenciamento da estação de recarga com base no nível de carga da bateria e o uso do sistema de visão disponível no TurtleBot 4.

Palavras-chave: robótica móvel; sistemas multirrobôs; ROS 2; controle supervisório; sistemas a eventos discretos; navegação autônoma.

ANEXOS

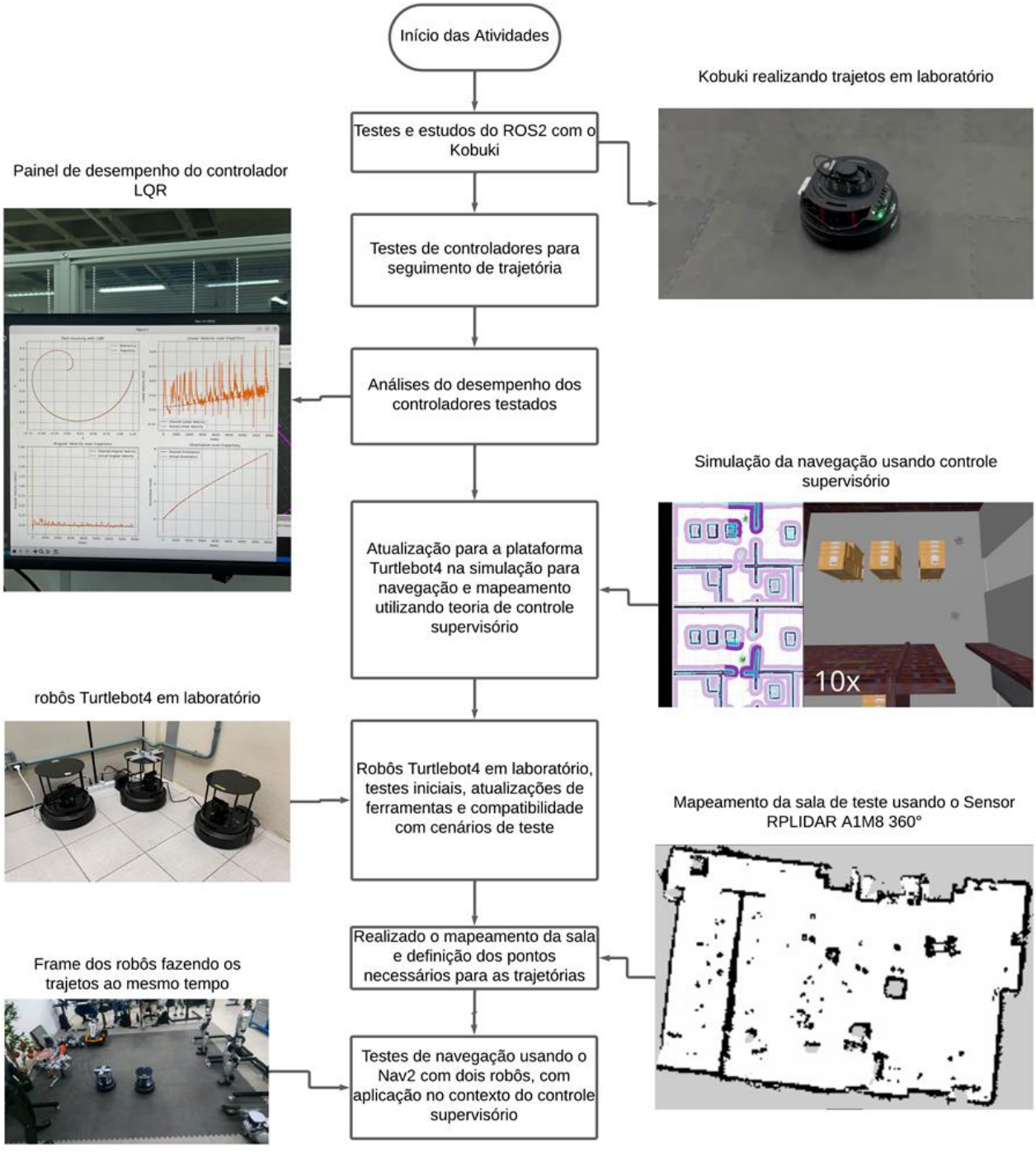


Figura 1. *Linha do tempo das atividades realizadas ao longo da pesquisa.*

REFERÊNCIAS BIBLIOGRÁFICAS

CASSANDRAS, C. G.; LAFORTUNE, S. *Introduction to Discrete Event Systems*. 3. ed. Cham: Springer, 2021.

Clearpath Robotics. TurtleBot 4 user manual: Features & Specifications. Disponível em: https://turtlebot.github.io/turtlebot4-user-manual/overview/features.html. Acesso em: 24 ago. 2026.

Nav2 Documentation. Navigation 2 User Guide. Open Navigation LLC. Disponível em: https://docs.nav2.org/. Acesso em: 24 ago. 2026.

QUEIROZ, M. H. de; CURY, J. E. R. Modular supervisory control of large scale discrete event systems. In: Boel, R.; Stremersch, G. (Org.). *Discrete Event Systems: Analysis and Control*. Boston: Springer, 2000. p. 103-110.

RAMADGE, P. J. G.; WONHAM, W. M. Supervisory control of a class of discrete-event processes. *SIAM Journal on Control and Optimization*, v. 25, n. 1, p. 206-230, 1987.

ROS 2 Documentation. ROS 2 Documentation: Jazzy Jalisco. Open-Source Robotics Foundation (OSRF). Disponível em: https://docs.ros.org/en/jazzy/index.html. Acesso em: 24 ago. 2026.