DESENVOLVIMENTO DE UMA PLATAFORMA PARA SISTEMAS MULTIRROBÔS EM ROS 2 COM MAPEAMENTO DINÂMICO E MODELAGEM POR AUTÔMATOS

Lúcia Rosa Silva Castañeda, Gabriel Abatti, André Bittencourt Leal

INTRODUÇÃO

Segundo Wei *et al*. (2025), com o avanço contínuo da Inteligência Artificial, dos sensores embarcados e das tecnologias de controle inteligente, os *Autonomous Mobile Robots* (*AMRs*) vêm sendo cada vez mais utilizados em diversos setores, incluindo a manufatura industrial, a logística inteligente, os serviços urbanos e a assistência à saúde. Esses sistemas podem operar continuamente, adaptar-se a mudanças no ambiente e executar manobras complexas que seriam inseguras ou inviáveis para humanos (Ba *et al*., 2026). Comparados aos equipamentos automatizados tradicionais, os AMRs apresentam maior flexibilidade operacional e capacidade de adaptação (Wei et *al*., 2025). Segundo Zohaib *et al*. (2023), um dos desafios da robótica móvel é o mapeamento de ambientes dinâmicos para fins de navegação. Nesse sentido, o planejamento eficiente de trajetórias e a prevenção de colisões dependem das informações disponíveis sobre o ambiente. O papel crítico dessas informações pode ser exemplificado por acidentes envolvendo veículos autônomos (Zohaib *et al*., 2023). Nesse contexto, a segurança constitui um aspecto importante no desenvolvimento da robótica móvel, demandando sistemas capazes de perceber e lidar com as incertezas inerentes a ambientes dinâmicos.

Dessa forma, este trabalho busca contribuir para a navegação autônoma por meio da implementação de um mapa dinâmico, capaz de incorporar novos objetos e manter atualizadas as informações sobre o ambiente diante das mudanças que possam ocorrer durante a operação do robô. Em conjunto com o desenvolvimento do mapa dinâmico, foram estudados e desenvolvidos modelos em autômatos para representar comportamentos do sistema robótico. Os autômatos permitem representar estados e transições de Sistemas a Eventos Discretos (SEDs) e, no contexto da Teoria de Controle Supervisório (TCS), podem ser utilizados para modelar plantas e especificações de controle, possibilitando a síntese de supervisores para restringir comportamentos indesejados do sistema (Cassandras; Lafortune, 2021). Assim, o objetivo deste trabalho é contribuir para o desenvolvimento de uma plataforma para sistemas multirrobôs em ROS 2 (*Robot Operating System 2)*, incorporando recursos de mapeamento dinâmico e modelos em autômatos associados ao comportamento dos robôs, no contexto da TCS.

DESENVOLVIMENTO

A base do trabalho iniciou-se com o estudo e a implementação de um sistema chamado *Cartographer*, que fornece recursos para *Simultaneous Localization and Mapping* (*SLAM*) em 2D e 3D em diversas plataformas e configurações de sensores. A partir do estudo do funcionamento desse sistema, verificou-se a possibilidade de implementar um mapa dinâmico. Para essa implementação, foi utilizado o pacote *Slam Toolbox*, que cria um mapa 2D do espaço e permite serializar os dados e o estado do mapa de SLAM, para que possam ser recarregados e utilizados posteriormente para continuar o mapeamento, realizar a localização, a fusão de mapas ou outras manipulações. Essa funcionalidade foi o ponto de partida para o desenvolvimento do trabalho, pois, para se obter um mapa dinâmico, é necessário realizar a desserialização do mapa atual. Para isso, foi utilizado *Deserialize PoseGraph* do pacote *Slam Toolbox*, que permite ler um arquivo previamente salvo em disco e reconstruir seus dados na memória do computador. Nesse arquivo estão armazenadas informações necessárias para representar o mapa e a trajetória do robô, como os nós (*poses*), as restrições ou relações entre essas *poses* e outros dados utilizados pelo algoritmo. Assim, ao realizar a desserialização, o *Slam Toolbox* recupera essas informações do arquivo e as carrega novamente em

sua estrutura interna, permitindo continuar o mapeamento, realizar a localização ou trabalhar com mapa que já havia sido salvo anteriormente, sem precisar reconstruir todo o mapa desde o início.

O *Slam Toolbox* foi utilizado em conjunto com o Nav2 (*Navigation 2*), que é a pilha (*stack*) de navegação do ROS 2, que reúne os componentes que um robô precisa para ir de um ponto A a um ponto B com segurança: saber onde está, planejar um caminho, seguir esse caminho controlando os motores e reagir a obstáculos e imprevistos. Essa integração resultou na implementação de um mapa dinâmico, capaz de incorporar novos objetos em tempo real.

Outra parte importante do trabalho foi o desenvolvimento e a discussão de autômatos para controlar os movimentos do robô. Até o momento, foram desenvolvidas seis plantas. Uma delas foi desenvolvida para substituir o *NavigateToPose* do Nav2, que permite que o robô se desloque diretamente até um único objetivo final, desacelerando e parando exatamente na posição e orientação solicitadas. As demais plantas têm o intuito de verificar a disponibilidade do robô, que pode estar indisponível ou disponível para as atividades, o consumo da bateria do robô, a identificação de novos objetos e o estado da missão do robô, indicando se ele está em missão, indo para casa, em espera, com a missão cancelada ou se chegou ao destino com sucesso. Também foi desenvolvida uma planta que contém as configurações de controle, que são divididas em configuração normal, como o *start* do robô, configuração dinâmica, como o recálculo da trajetória, e configuração crítica, como o desvio de objetos.

RESULTADOS

O mapa dinâmico foi inicialmente implementado com um único robô, o *Kobuki*, que foi o primeiro a ser utilizado nos testes realizados pela equipe. Os resultados obtidos com a implementação desse mapa foram os arquivos correspondentes ao mapa atualizado e à sua serialização, permitindo a continuidade do mapeamento. O mapa é salvo a cada 45 segundos, utilizando-se o *save map* do *Slam Toolbox* para salvar o arquivo de imagem do mapa do gráfico de poses, que pode ser utilizado para exibição ou localização por meio do AMCL (*Adaptive Monte Carlo Localization*). Após a verificação do funcionamento com o *Kobuki*, implementou-se o mapa dinâmico com múltiplos robôs (Figura 1). Nesse caso, foram utilizados robôs *TurtleBot*, plataformas robóticas de baixo custo e com *software* de código aberto, adquiridas pela universidade.

Na implementação desse mapa dinâmico com múltiplos robôs, foi necessário renomear os tópicos e os nós utilizando *namespaces*. Com essa alteração, o nó do *Slam Toolbox* também passou a utilizar um *namespace*, o que gerou um problema ao salvar o mapa atualizado utilizando o *save map*. Dessa forma, utilizou-se o *map server* para realizar o salvamento do mapa. Este é um nó do ROS 2 usado pelo Nav2 que pode carregar um mapa salvo, disponibilizá-lo para outros nós do sistema e salvar alterações no mapa. O arquivo de serialização, por sua vez, não apresentou problemas com o uso de *namespace* e foi salvo normalmente.

O autômato ilustrado na Figura 2 corresponde ao *NavigateToPose* e apresenta os eventos associados à navegação do robô. No contexto da TCS, esses eventos podem ser classificados como controláveis ou não controláveis. Os eventos controláveis podem ser habilitados ou desabilitados pelo supervisor, enquanto os eventos não controláveis não podem ter sua ocorrência impedida pelo supervisor. No autômato apresentado, *goal_send* é um evento controlável e representa o envio de uma nova posição para navegação; *goal_final* é não controlável e representa a chegada à posição desejada; *cancel_goal* é controlável e representa o cancelamento da navegação; *fail_navigation* é não controlável e representa uma falha na navegação; *robot_stop* é controlável e representa a parada do robô; *rec_path* é controlável e representa o recálculo da trajetória; e *fail_path* é não controlável e representa a ocorrência de um erro na navegação.

CONSIDERAÇÕES FINAIS

O trabalho ainda está em andamento e uma das próximas etapas do desenvolvimento da plataforma consiste em integrar o mapa dinâmico às plantas desenvolvidas no contexto da TCS. Foram identificadas limitações no trabalho desenvolvido, relacionadas à capacidade computacional disponível, especialmente durante a execução simultânea dos diferentes processos do sistema robótico. Essas limitações foram mais evidentes durante a execução do SLAM, devido à maior demanda de recursos computacionais para o processamento dos dados dos sensores, a construção do mapa e a atualização das informações necessárias à navegação do robô. Como consequência, observou-se lentidão na movimentação dos robôs. Entretanto, essas limitações não comprometeram os resultados obtidos até esta etapa do trabalho.

Palavras-chave: mapeamento dinâmico; robôs móveis; ROS 2; SLAM; Sistemas a Eventos Discretos; Teoria de Controle Supervisório.

ANEXOS

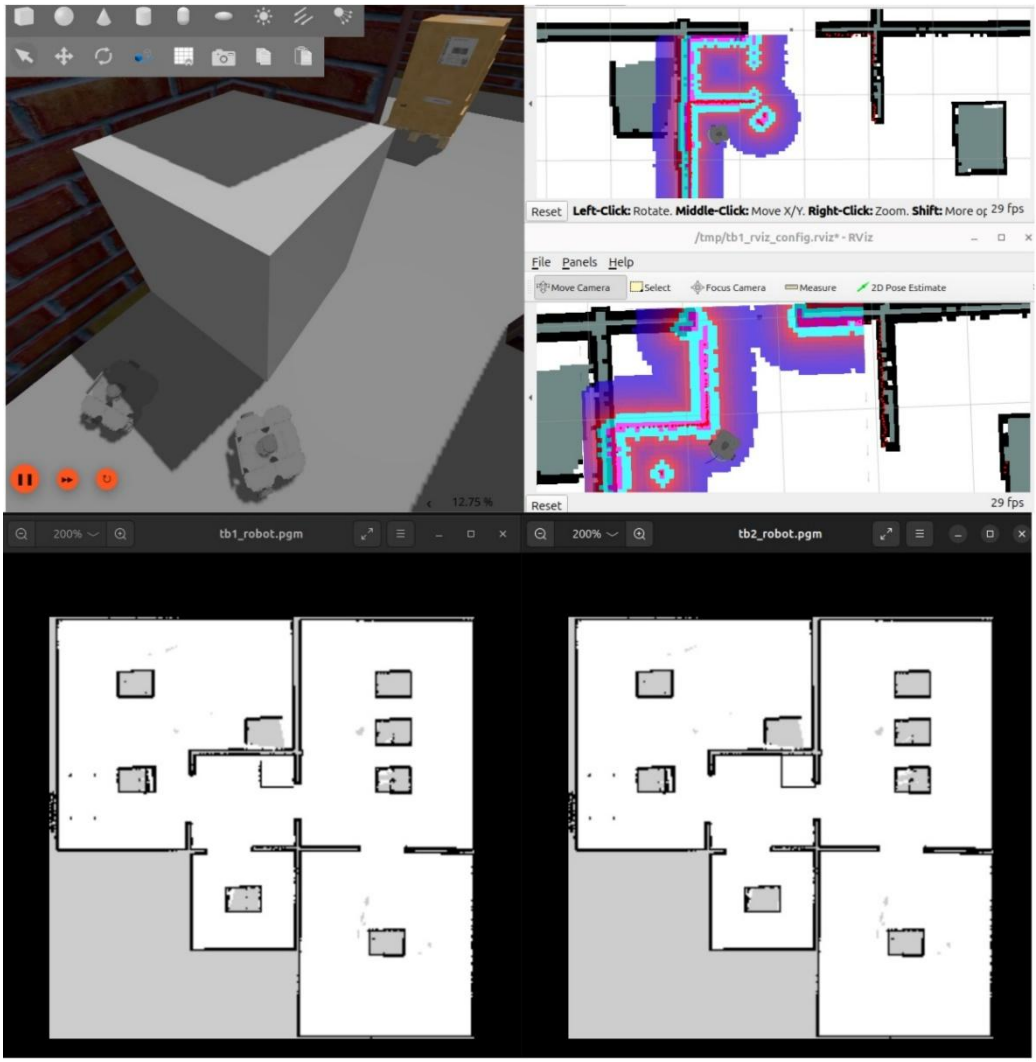


Figura 1. *Mapa Dinâmico.*

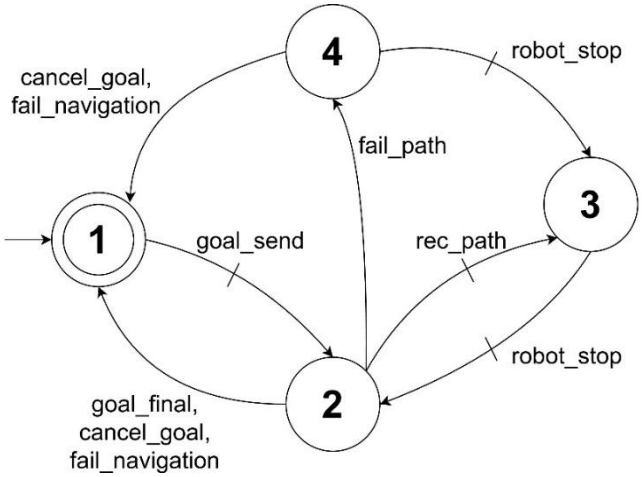


Figura 2. *Autômato par o NavigateToPose.*

REFERÊNCIAS BIBLIOGRÁFICAS

BA, R.; MISHRA, P.; KALAIMANI, M.; JUYAL, P.; ANJANI, P. K.; DUA, R.; MAMORIA, P. Real-time obstacle avoidance in mobile robots using deep reinforcement learning. *Future Technology*, v. 5, n. 3, p. 69–76, 2026.

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

WEI, Z.; WANG, S.; CHEN, K.; WANG, F. ROS-Based Navigation and Obstacle Avoidance: A Study of Architectures, Methods, and Trends. *Sensors*, v. 25, n. 14, p. 1–22, 2025.

ZOHAIB, M.; AHSAN, M.; KHAN, M.; IQBAL, J. A featureless approach for object detection and tracking in dynamic environments. *PLOS ONE*, v. 18, n. 1, p. 1–20, 2023.