

Plataforma baseada em ROS2 e Gazebo para avaliação de controladores em robótica móvel com visão computacional e inteligência artificial

ROS2 and Gazebo-based platform for evaluating controllers in mobile robotics using computer vision and artificial intelligence

¹Fabio Romero De Souza Junior, ²Daniel Costa Ramos

¹Mestrando em Engenharia Eletrica – Universidade Federal de Uberlândia (fabio.junior@ufu.br)

²Professor Efetivo – Doutorado – Universidade Federal de Uberlândia (danielramos@ufu.br)

RESUMO: Este trabalho apresenta uma plataforma experimental baseada em ROS2 e Gazebo para avaliação reprodutível de algoritmos de controle aplicados à robótica móvel. A arquitetura utiliza visão computacional como entrada, a partir de imagens capturadas por câmera, e permite a integração de diferentes abordagens sob as mesmas condições experimentais. Como validação, são comparados um controlador PID e um agente TD3 (Twin Delayed Deep Deterministic Policy Gradient) em uma tarefa de seguimento de linha. Os resultados mostram que a plataforma é capaz de gerar métricas quantitativas consistentes e suportar a comparação entre métodos, além de permitir a transposição para sistemas físicos, contribuindo para a redução do gap sim-to-real.

Palavras Chave: Aprendizado por Reforço Profundo, Gazebo, ROS2, Robótica móvel, Seguimento de linha, Visão computacional.

ABSTRACT: This work presents an experimental platform based on ROS2 and Gazebo for reproducible evaluation of control algorithms applied to mobile robotics. The architecture uses computer vision as input, based on images captured by a camera, and enables the integration of different approaches under the same experimental conditions. As validation, a PID controller and a TD3 agent (Twin Delayed Deep Deterministic Policy Gradient) are compared in a line-following task. The results show that the platform is capable of generating consistent quantitative metrics and supporting comparisons between methods, as well as enabling transfer to physical systems, contributing to the reduction of the sim-to-real gap.

Keywords: Computer vision, Deep Reinforcement Learning, Gazebo, Line following, Mobile robotics, ROS2.

Recebido em:
04/06/2026
Publicado em:
06/07/2026

1. INTRODUÇÃO

O avanço das técnicas de controle e de inteligência artificial tem impulsionado significativamente o desenvolvimento da robótica móvel, permitindo a execução de tarefas cada vez mais complexas em ambientes dinâmicos (GHODSI et al., 2026). Nesse contexto, ferramentas de simulação têm se tornado amplamente utilizadas, pois possibilitam o desenvolvimento, teste e validação de algoritmos em ambientes controlados, reduzindo custos e riscos associados à experimentação em sistemas físicos (KOJIMA et al., 2025). Entre essas ferramentas, destacam-se ambientes que integram simulação física e sistemas de comunicação, permitindo maior fidelidade na representação do comportamento robótico (FLORES GONZALEZ et al., 2025).

Apesar desses avanços, a avaliação comparativa de diferentes abordagens de controle e aprendizado ainda apresenta limitações importantes (GUNES et al., 2022). A ausência de padronização nos ambientes experimentais, aliada às variações nos modelos de robôs, sensores

e condições de teste, dificulta a reprodução dos resultados e compromete a análise consistente de desempenho entre diferentes métodos (TANG et al., 2024). Esse problema é particularmente relevante em tarefas clássicas, como o seguimento de trajetória, nas quais pequenas variações no ambiente podem impactar significativamente os resultados obtidos (BRIGIDO; OLIVEIRA, 2025).

Diante desse cenário, observa-se uma lacuna na disponibilidade de plataformas que permitam a avaliação reprodutível e comparável de algoritmos de controle e inteligência artificial em robótica móvel. Embora ambientes de simulação baseados em ROS e Gazebo sejam amplamente utilizados (FARLEY et al., 2022), as ferramentas existentes frequentemente carecem de suporte integrado para substituição de algoritmos (TANG et al., 2024), coleta estruturada de métricas e padronização experimental (DUAN et al., 2016), bem como integração com aprendizado de máquina (XIANG et al., 2019), limitando a transposição para sistemas reais (NEVO et al., 2025). Essa lacuna motiva o presente trabalho.

Como principal contribuição, este trabalho apresenta uma plataforma integrada baseada em ROS2 e Gazebo para avaliação de controladores em robótica móvel. Esta plataforma permite a execução de diferentes abordagens sob condições reprodutíveis e padronizadas, viabilizando a coleta estruturada de dados e a geração de métricas quantitativas para facilitar a comparação entre métodos. Além disso, a integração entre simulação e modelo de controle contribui para a redução do gap sim-to-real, estabelecendo um ambiente experimental consistente para validação e desenvolvimento de algoritmos.

2. MATERIAIS E MÉTODO

Esta seção apresenta a arquitetura do sistema, bem como a modelagem do ambiente e do robô utilizados nos experimentos.

2.1 Visão Geral da Arquitetura

A arquitetura proposta integra ROS2 e Gazebo para avaliação reprodutível de controladores em robótica móvel, sendo estruturada de forma modular para permitir a substituição de algoritmos sem alterações na infraestrutura. O ROS2 é responsável pela comunicação entre os módulos, enquanto o Gazebo realiza a simulação do ambiente e da dinâmica do robô, disponibilizando dados sensoriais que são utilizados pelos controladores para geração de comandos.

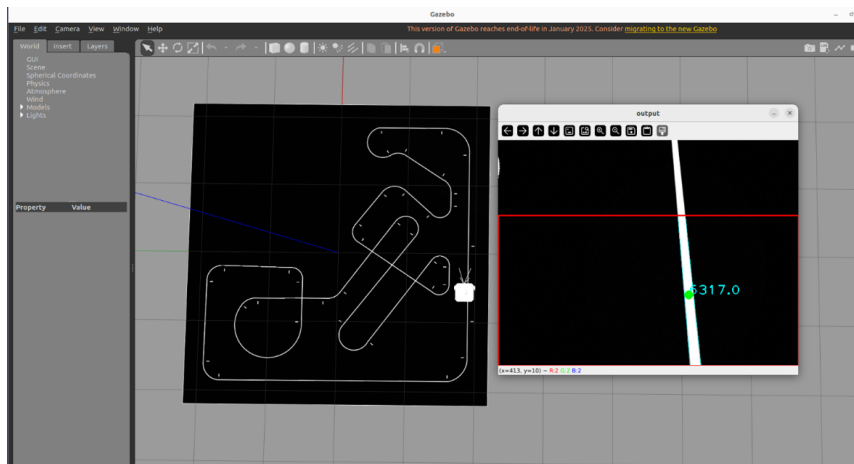
O sistema é composto por módulos de aquisição de dados, controle e registro de informações. O módulo de controle pode ser implementado por diferentes estratégias, incluindo abordagens clássicas e baseadas em inteligência artificial. Já o módulo de coleta registra métricas como erro, comandos de controle e estados do robô, possibilitando a análise do desempenho e garantindo consistência na avaliação entre diferentes métodos.

2.2 Modelagem do Ambiente

Para os experimentos, foi adotado um cenário com trajetória complexa, composta por curvas e mudanças de direção que aumentam a exigência no seguimento da trajetória. A Figura 1 apresenta o ambiente de simulação desenvolvido no Gazebo. À esquerda, é mostrado o mapa com a trajetória de referência, caracterizada por um percurso fechado. À direita, é apresentada a imagem capturada pela câmera embarcada no robô, juntamente com o processamento aplicado

para detecção da trajetória, incluindo a definição da região de interesse e a estimativa da posição da linha no plano da imagem.

Figura 1 – Ambiente de simulação no Gazebo com o robô e a pista de seguimento de linha.



A percepção do ambiente é realizada por meio de visão computacional. A configuração proposta permite a utilização de diferentes geometrias de pista em condições controladas, mantendo a consistência na aquisição e processamento dos dados.

2.3 Modelo do Robô

A proposta é independente de um modelo específico de robô, podendo ser aplicada a diferentes plataformas que utilizem comandos de velocidade e sensores adequados ao seguimento de trajetória. Essa abstração permite a substituição do robô sem impacto na arquitetura do sistema. Para validação, foi utilizado o modelo TurtleBot3 Waffle, equipado com sensor visual e controle baseado no envio de comandos de velocidade, com o sistema organizado em módulos de percepção, controle e atuação.

Para adequação ao problema de seguimento de linha, foram realizadas modificações no modelo do robô, incluindo a desativação do LiDAR, reposicionamento da câmera e ajustes nos parâmetros dinâmicos das rodas, tornando a resposta ao controle com dinâmica mais complexa. A escolha do TurtleBot3 Waffle Pi se justifica pela disponibilidade de câmera e IMU, além da compatibilidade com aplicações reais, permitindo a transposição dos experimentos para plataformas físicas.

2.4 Visão Computacional

A percepção do ambiente é realizada por meio de uma câmera embarcada no robô, responsável pela captura de imagens utilizadas na identificação da trajetória. A linha a ser seguida é representada por uma região de alto contraste no ambiente, permitindo sua detecção a partir de técnicas simples de processamento de imagem.

A partir das imagens adquiridas, é estimada a posição da linha no campo de visão, sendo então calculado o erro de rastreamento em relação ao centro da imagem. Esse erro é utilizado como variável de entrada para os algoritmos de controle, estabelecendo a conexão entre percepção e tomada de decisão no sistema. O módulo de percepção é desacoplado, permitindo a substituição das técnicas de processamento sem alterações no restante da arquitetura.

2.5 Pipeline de Controle

O sistema opera em malha fechada, em um ciclo contínuo de percepção e ação. A cada iteração, imagens capturadas pela câmera são utilizadas para estimar a posição da linha no campo de visão. A partir dessa estimativa, calcula-se o erro de rastreamento, que é fornecido ao controlador para geração dos comandos de velocidade aplicados ao robô. Esse processo iterativo é descrito no Figura 2.

O controle da malha fechada pode ser realizado por diferentes abordagens, incluindo métodos clássicos, como o controlador PID, e técnicas baseadas em inteligência artificial. Entre estas, destacam-se modelos de aprendizado por reforço, capazes de aprender políticas de controle representadas por redes neurais, nas quais o ator é responsável por mapear diretamente os estados do sistema em ações de controle.

Figura 2 – Laço de controle percepção-ação para seguimento de linha

Algoritmo 1: Laço de controle percepção-ação para seguimento de linha

```

1: Carregar a configuração do controlador (ganhos do PID
   ou política treinada  $\pi_\theta$ )
2: Definir a velocidade linear constante  $v \leftarrow v_0$ 
3: for cada instante de tempo  $t$  até a finalização do
4:   Adquirir a imagem da câmera  $I_t$ 
5:   Detectar a linha em  $I_t$  e calcular o centróide  $c_t$ 
6:   Calcular o erro lateral  $e_t \leftarrow c_t - c_{center}$ 
7:   Limitar o erro  $e_t \leftarrow \text{clip}(e_t, -300, 300)$ 
8:   if controlador = PID then
9:      $\omega_t \leftarrow \text{PID}(e_t)$ 
10:  else
11:     $\omega_t \leftarrow \pi_\theta(e_t)$ 
12:  end if
13:  Normalizar  $\omega_t$  conforme os limites do atuador
14:  Publicar o comando  $(v_0, \omega_t)$ 
15:  Registrar  $(e_t, \omega_t, t)$ 
16: end for

```

2.6 Integração com IA

A integração com técnicas de inteligência artificial ocorre no módulo de controle, substituindo o controlador clássico por modelos baseados em aprendizado. Nessa abordagem, a política de controle é aprendida a partir dos dados de entrada do sistema, sendo utilizada diretamente para a geração dos comandos aplicados ao robô.

A arquitetura proposta permite a incorporação desses modelos sem alterações na infraestrutura, possibilitando a avaliação de diferentes estratégias em condições padronizadas. Para validação, foi utilizado o algoritmo TD3 (Twin Delayed Deep Deterministic Policy Gradient), adequado a espaços de ação contínuos, sendo empregado como controlador na tarefa de seguimento de linha (LIN et al., 2025).

2.7 Coleta de Dados e Métricas

Durante a execução dos experimentos, a plataforma realiza a coleta automática de dados provenientes dos diferentes módulos do sistema, incluindo imagens da câmera embarcada, erro lateral ao longo do tempo, comandos de velocidade (linear e angular), dados de orientação obtidos pela IMU e tempo de simulação. Essa aquisição é realizada de forma padronizada para diferentes controladores, garantindo consistência nas análises.

A partir desses dados, são calculadas métricas quantitativas de desempenho, como erro médio absoluto (MAE), erro quadrático médio (RMSE), overshoot, esforço de controle e desvio padrão. Essas métricas são obtidas diretamente dos sinais registrados, permitindo a avaliação objetiva do comportamento dos controladores e a comparação em um ambiente de teste padronizado.

2.8 Reprodutibilidade

Os experimentos foram realizados utilizando o mesmo mapa, configurações do robô e parâmetros, garantindo a avaliação dos métodos sob condições equivalentes e possibilitando a análise comparativa entre diferentes abordagens de controle.

Foram considerados dois controladores para a avaliação do desempenho do robô: um baseado em controle clássico, utilizando um controlador PID, e outro baseado em aprendizado por reforço por meio do algoritmo TD3. O controlador PID foi configurado com ação de controle limitada a $u_{\max} = 1.0$, adotando ganhos de $K_p = 0,018$, $K_i = 8 \times 10^{-7}$ e $K_d = 3,5 \times 10^{-8}$. Já o modelo TD3 foi previamente treinado em ambiente de simulação e, durante a etapa de avaliação, foi empregado exclusivamente para inferência, sem atualização adicional dos parâmetros do agente.

Ambos os métodos foram executados com velocidade linear constante de 0.125 m/s e saturação da velocidade angular limitada a 1 rad/s. Cada experimento foi repetido cinco vezes, com sementes fixas quando aplicável, assegurando consistência entre execuções. A arquitetura permite a substituição do controlador sem modificar o restante do sistema, preservando o ambiente experimental.

3. RESULTADOS/ DISCUSSÕES

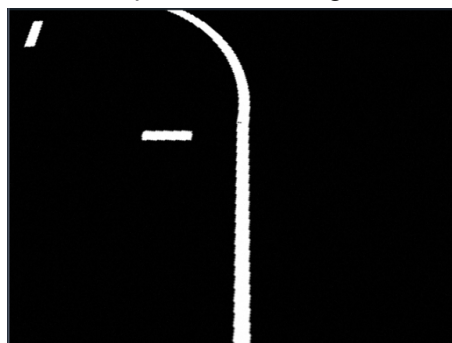
Os experimentos foram conduzidos com o objetivo de validar a arquitetura proposta e demonstrar sua capacidade de suportar diferentes estratégias de controle sob as mesmas condições experimentais. Foi utilizado um cenário com trajetória complexa na tarefa de seguimento de linha.

As métricas consideradas incluem o erro de rastreamento ao longo do tempo e variáveis relacionadas ao movimento do robô, permitindo a avaliação do desempenho dos controladores.

3.1 Capacidades de Aquisição e Registro

A plataforma permite a aquisição e o armazenamento contínuo de imagens RGB provenientes da câmera simulada no ROS, viabilizando a análise do comportamento do sistema e a reprodução dos experimentos. Esses dados possibilitam a aplicação e comparação de diferentes algoritmos de visão computacional, incluindo abordagens baseadas em OpenCV, bem como o treinamento de modelos como CNNs e YOLO a partir de dados reutilizáveis. Como limitação, o desempenho depende da integração entre os módulos de percepção e controle, podendo ser influenciado por atrasos no processamento das imagens. A Figura 3 apresenta um exemplo de imagem capturada durante a execução da tarefa de seguimento de linha, na qual a linha branca representa a trajetória a ser seguida.

Figura 3 - Exemplo de imagem capturada e armazenada pela plataforma durante a execução da tarefa de seguimento de linha.



A imagem capturada constitui a entrada do módulo de percepção, a partir da qual são extraídas características visuais utilizadas na estimação do erro lateral empregado pelo controlador.

A plataforma coleta automaticamente dados de odometria e IMU por meio de nós no ROS, armazenando posição e orientação ao longo do tempo para cada execução. Esses dados são utilizados para analisar o deslocamento do robô e sua dinâmica angular, permitindo comparar diretamente o comportamento dos controladores com parâmetros experimentais fixos. Como limitação, os resultados dependem da precisão dos modelos de sensores no ambiente simulado. A Figura 4 e a Figura 5 mostram a evolução da posição e da orientação para os controladores PID e TD3.

Figura 4 - Evolução da posição (odometria) e orientação (IMU) ao longo do tempo utilizando o controlador PID.

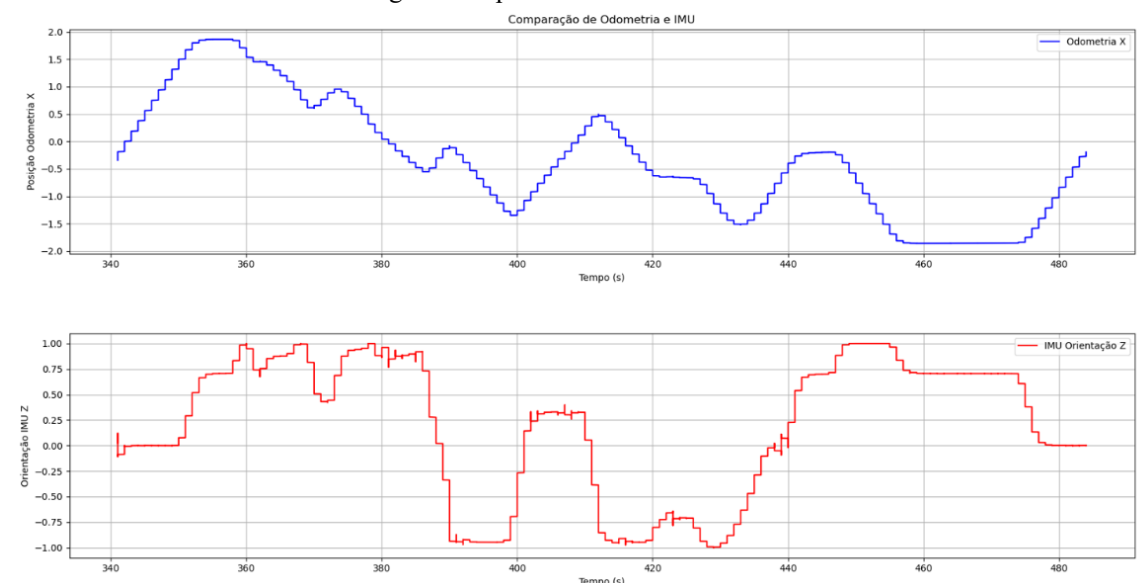
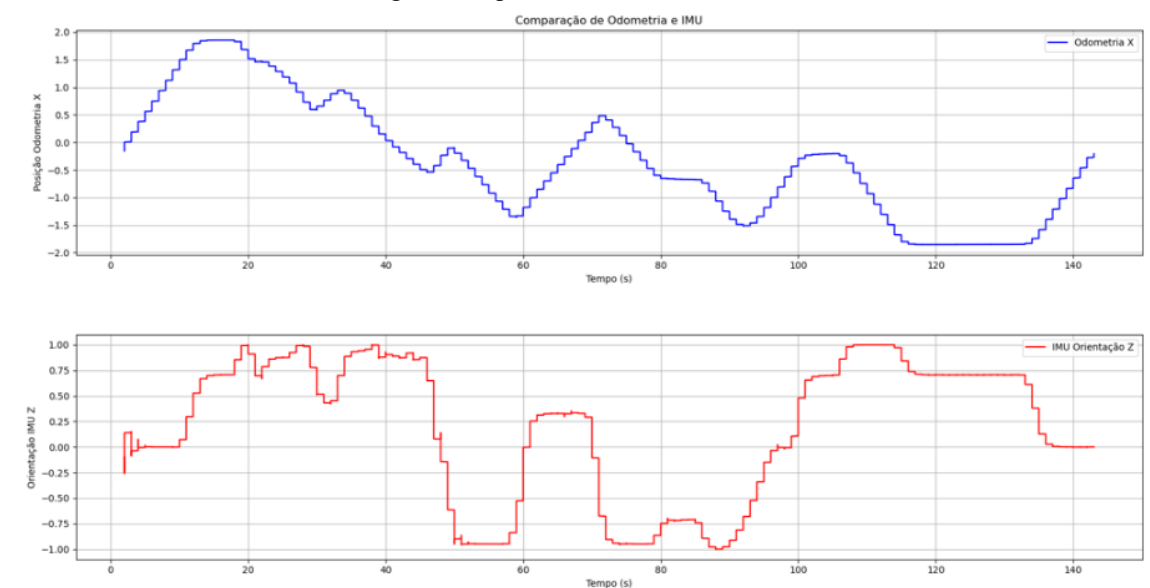


Figura 5 - Evolução da posição (odometria) e orientação (IMU) ao longo do tempo utilizando o controlador TD3.



Os dados indicam que a evolução da orientação reflete diretamente as correções realizadas pelo controlador ao longo do percurso, permitindo analisar a dinâmica angular e a estabilidade do sistema durante a execução.

A trajetória do robô é reconstruída a partir dos dados de posição obtidos por odometria e IMU, aplicando o mesmo processamento para todos os controladores. Esses dados permitem analisar o deslocamento no ambiente e comparar desvios em relação à pista sob as mesmas condições experimentais, podendo também ser utilizados em avaliações comparativas. Como limitação, a precisão da trajetória depende da estimativa de posição, podendo acumular erro ao longo do tempo. A Figura 6 e a Figura 7 mostram as trajetórias dos controladores PID e TD3, que seguem o formato da pista sem desvios significativos.

Figura 6 - Trajetória do robô utilizando o controlador PID, onde a linha em azul representa o percurso executado no ambiente.

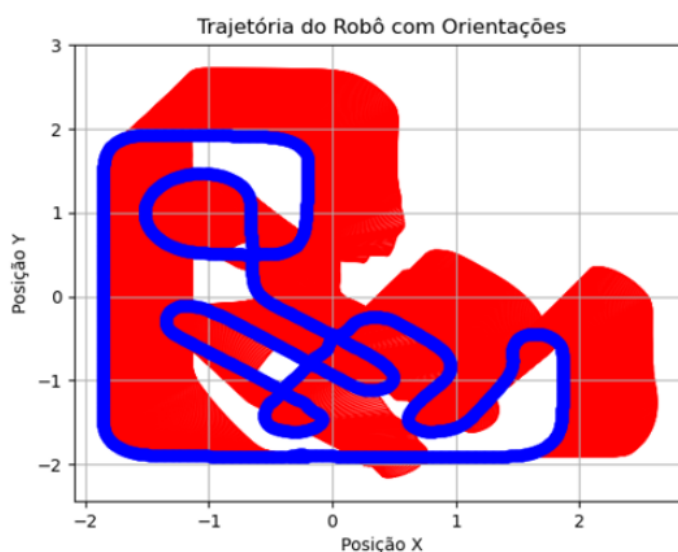
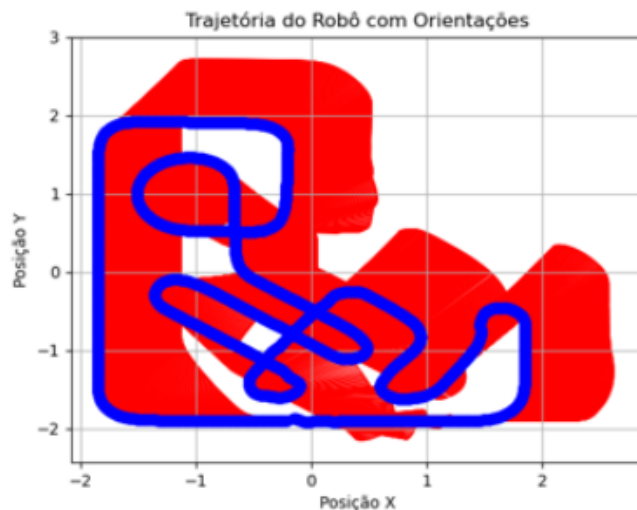


Figura 7 - Trajetória do robô utilizando o controladorTD3, onde a linha em azul representa o percurso executado no ambiente.



As trajetórias reconstruídas apresentam aderência ao formato da pista, permitindo avaliar o desempenho espacial dos controladores e indicando ausência de desvios relevantes ao longo da execução.

As componentes da velocidade linear são obtidas a partir dos comandos aplicados ao robô, com taxa de amostragem configurável e igual para todos os controladores. Esses dados permitem analisar o comportamento longitudinal e lateral, incluindo variações, oscilações e estabilidade ao longo do tempo, além de comparar diretamente as estratégias em condições controladas e padronizadas. Como limitação, a análise depende da taxa de amostragem, que influencia a resolução temporal dos sinais, e das restrições do robô, como o limite de velocidade angular. A Figura 8 e a Figura 9 mostram as componentes da velocidade linear para os controladores PID e TD3.

Figura 8 - Componentes da velocidade linear ao longo do tempo utilizando o controlador PID.

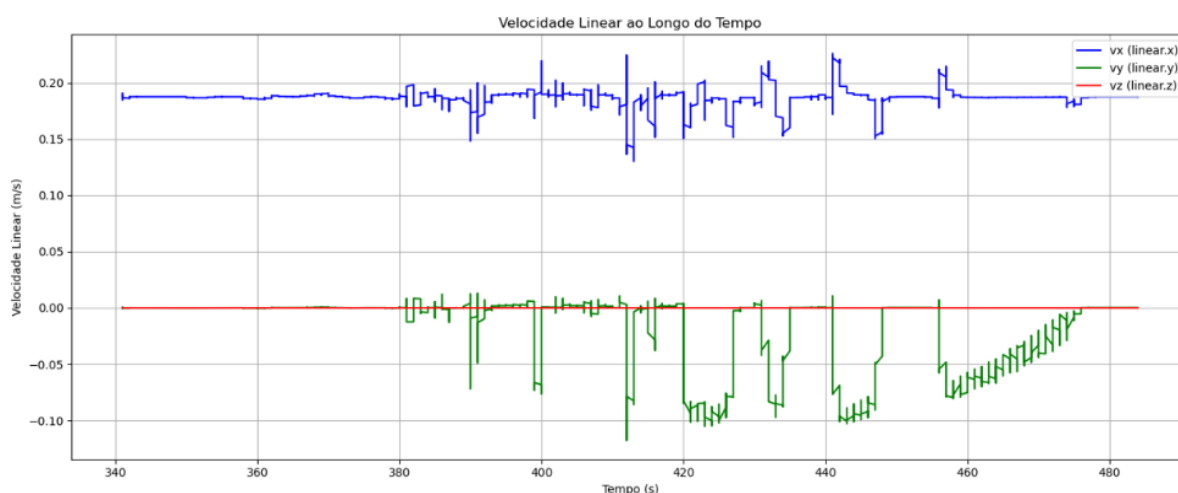
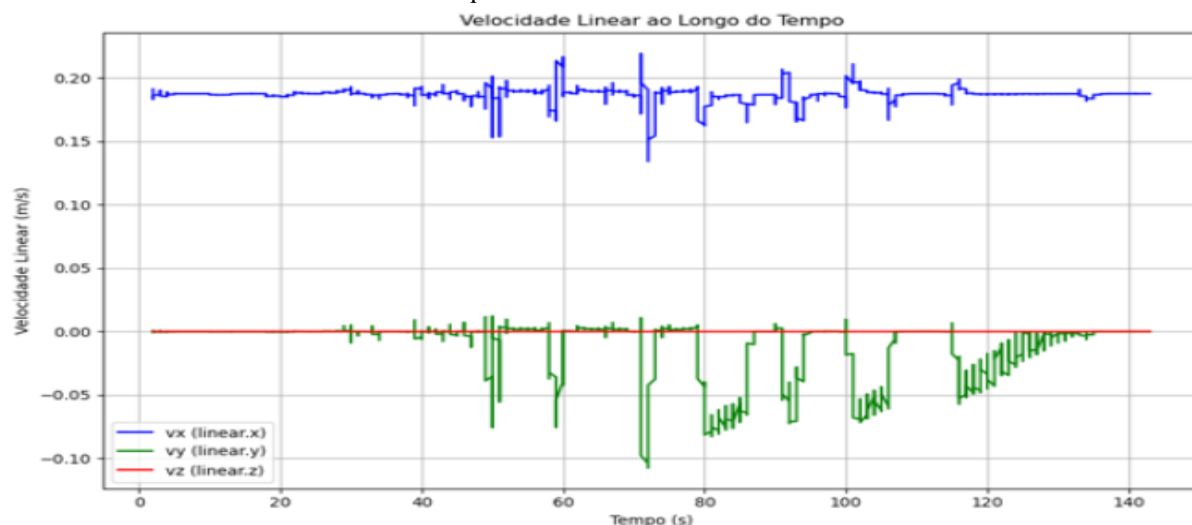


Figura 9 - Componentes da velocidade linear ao longo do tempo utilizando o controlador TD3.



Observa-se que a componente longitudinal permanece aproximadamente constante ao longo do percurso, enquanto a componente lateral apresenta variações associadas às correções de trajetória, sem evidências de instabilidade ou oscilações significativas, indicando comportamento consistente entre os controladores nas condições avaliadas.

A velocidade angular ao longo do tempo é obtida a partir dos dados do robô durante a execução. Esses dados permitem analisar o comportamento rotacional, incluindo variações, oscilações e estabilidade, além de comparar diretamente as estratégias nas mesmas condições de execução. Como limitação, os resultados dependem da fidelidade do modelo do robô no ambiente simulado. As Figura 10 e a Figura 11 mostram as componentes da velocidade angular para os controladores PID e TD3.

Figura 10 - Componentes da velocidade angular ao longo do tempo utilizando o controlador PID.

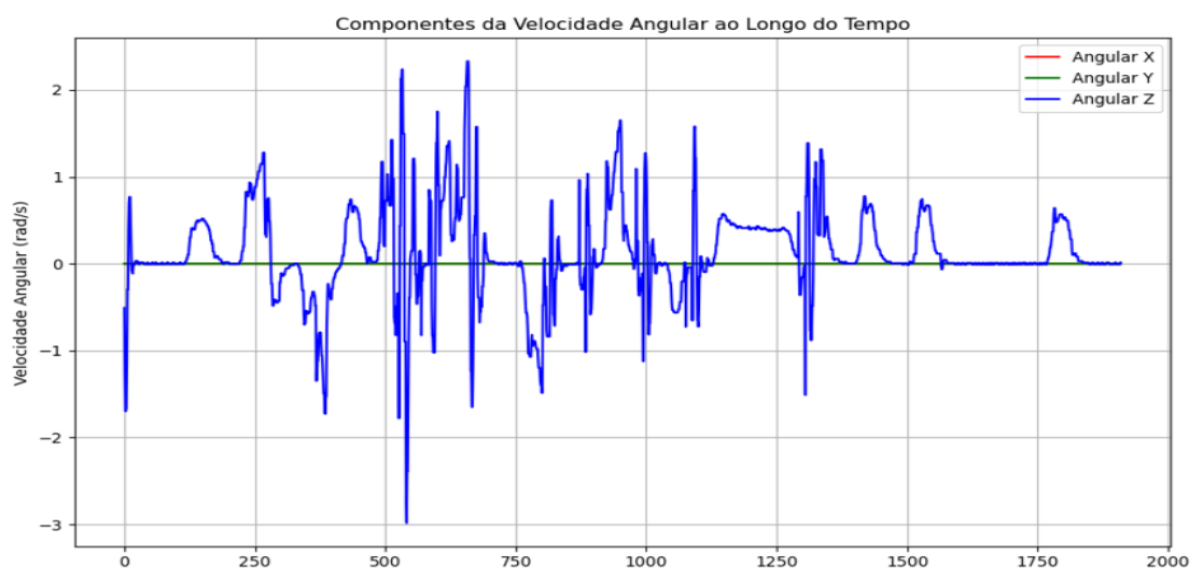
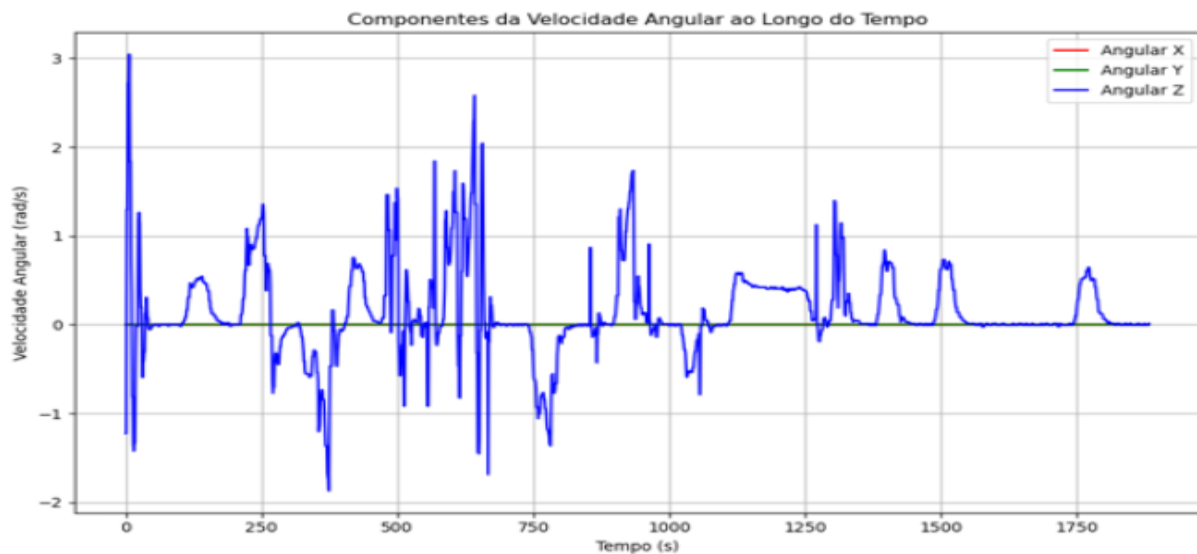


Figura 11 - Componentes da velocidade angular ao longo do tempo utilizando o controlador TD3.



Observa-se que os sinais apresentam variações associadas às correções de direção ao longo do percurso, com presença de oscilações que refletem os ajustes realizados pelos controladores, sem evidências de saturação, indicando comportamento rotacional consistente no contexto experimental analisado.

A plataforma permite adquirir o erro lateral a partir da imagem da câmera, definido como a diferença entre o centróide da linha detectada e o centro da imagem, fornecendo uma métrica quantitativa para avaliação do seguimento de trajetória. Esses dados possibilitam a análise de desvios, estabilidade e resposta ao longo do percurso, bem como a comparação entre diferentes controladores sob as mesmas condições experimentais. Como limitação, a métrica depende do desempenho da detecção da linha, podendo ser afetada por falhas na percepção.

3.2 Análise dos Resultados

A Tabela 1 apresenta as métricas de desempenho dos controladores PID e TD3 sob as mesmas condições experimentais.

Tabela 1 - Comparação de desempenho entre PID e TD3

Métrica	PID	TD3
MAE	18.68	16.91
RMSE	28.77	28.51
Overshoot	113.80	127.20
Esforço de controle	35401.40	31939.80
Desvio padrão	26.47	27.31
Tempo médio (s)	148	149

Os resultados demonstram que a plataforma distingue quantitativamente diferentes estratégias de controle de forma consistente. Como o ambiente, a percepção e o pipeline de dados são mantidos fixos e apenas o módulo de controle é substituído, as variações observadas nas métricas de MAE, RMSE, overshoot e esforço de controle são atribuídas diretamente aos controladores avaliados, eliminando fontes externas de variação presentes em outros trabalhos. Esses resultados indicam que a plataforma é capaz de produzir comparações críveis entre métodos distintos, com diferenças mensuráveis e reproduzíveis sob as mesmas condições experimentais.

Os dados gerados, incluindo trajetória, odometria, IMU e erro lateral, são contínuos e sincronizados no tempo, o que permite a análise temporal do comportamento do robô e a extração de métricas comparativas entre controladores. A simulação utiliza um modelo baseado no TurtleBot, com sensores e dinâmica compatíveis com plataformas reais, e o comportamento observado é plausível, conforme evidenciado pelas trajetórias e pelos sinais de velocidade linear e angular durante o seguimento da trajetória. O ambiente simulado reproduz com fidelidade o funcionamento interno de um robô e as interações entre seus diversos módulos, especialmente aqueles que incorporam técnicas de inteligência artificial e processamento de imagens. Apesar de mais controlado e previsível, o ambiente simulado reproduz com precisão a dinâmica de um robô móvel e o funcionamento de seus componentes físicos.

A arquitetura modular da plataforma permite a alteração independente de sensores, mapas e cenários, ampliando seu uso em diferentes configurações experimentais. Essa flexibilidade possibilita a avaliação de uma variedade maior de controladores e estratégias de navegação, sem a necessidade de modificações estruturais no sistema. Os experimentos apresentados demonstram que não apenas é possível integrar esses softwares, como também ampliar sua utilização para incorporar técnicas avançadas e gerar relatórios detalhados que auxiliam na comparação técnica dos dados durante as simulações, demonstrando sua viabilidade para estudos comparativos em robótica móvel.

A experiência com as implementações dessas simulações também evidenciou possibilidades concretas de extensão das pesquisas, incluindo a avaliação de algoritmos como SAC e PPO, a substituição do pipeline de visão por CNNs ou modelos baseados em transformers, e a variação de cenários e geometrias de pista sem alterações estruturais no sistema. Essas possibilidades indicam o potencial para o desenvolvimento de benchmarks padronizados e reforçam de forma consistente a plataforma como um ambiente de referência para pesquisas em controle e navegação autônoma.

4. CONCLUSÃO

Este trabalho apresentou uma plataforma integrada baseada em ROS 2 e Gazebo para avaliação de controladores em robótica móvel, com foco na padronização experimental e na aquisição estruturada de dados. A arquitetura modular proposta permite substituir o módulo de controle sem alterar os demais componentes do sistema, garantindo que as diferenças observadas entre estratégias sejam atribuídas exclusivamente ao controlador avaliado.

A validação com os controladores PID e TD3 demonstrou que a plataforma produz resultados críveis e mensuráveis, evidenciando comportamentos distintos por meio de métricas quantitativas. O uso de simulação baseada em modelos com sensores e dinâmica compatíveis com plataformas reais contribui para a plausibilidade dos experimentos, ainda que o ambiente simulado não reproduza integralmente os efeitos presentes em sistemas físicos, como ruído sensorial, atrasos e incertezas de modelagem.

Como próximos passos, propõe-se a validação da plataforma em robôs físicos, com foco na transferência de controladores entre simulação e ambiente real, bem como na investigação do impacto da percepção no desempenho do sistema e no uso da plataforma para execução reprodutível de experimentos. Pretende-se também ampliar o conjunto de avaliações, incluindo diferentes condições de operação e estratégias de controle, mantendo a mesma estrutura experimental para garantir comparabilidade. Adicionalmente, busca-se utilizar a plataforma como base para o desenvolvimento de benchmarks voltados à avaliação de métodos em robótica móvel.

5. REFERÊNCIAS

BRIGIDO, W. J. H.; OLIVEIRA, J. M. P. de. *The line follower robot: A meta-analytic approach*. PeerJ Computer Science, v. 11, p. e2744, 2025. <https://doi.org/10.7717/PEERJ-CS.2744>

DUAN, Y.; CHEN, X.; SCHULMAN, J.; ABBEEL, P. *Benchmarking Deep Reinforcement Learning for Continuous Control*. 2016.

FARLEY, A.; WANG, J.; MARSHALL, J. A. *How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion*. Simulation Modelling Practice and Theory, v. 120, p. 102629, 2022. <https://doi.org/10.1016/J.SIMPAT.2022.102629>

FLORES GONZALEZ, J. M.; CORONADO, E.; YAMANOBÉ, N. *ROS-Compatible Robotics Simulators for Industry 4.0 and Industry 5.0: A Systematic Review of Trends and Technologies*. Applied Sciences, v. 15, n. 15, p. 8637, 2025. <https://doi.org/10.3390/app15158637>

GHODSI, M. H.; SHAHBAZI, H.; TORABI, K. *A Comprehensive Survey of Deep Reinforcement Learning Techniques for Soft Mobile Robots*. Drones and Autonomous Vehicles, v. 3, n. 1, p. 10022, 2026. <https://doi.org/10.70322/dav.2025.10022>

GUNES, H.; BROZ, F.; CRAWFORD, C. S.; ROSENTHAL-VON DER PÜTTEN, A.; STRAIT, M.; RIEK, L. *Reproducibility in Human-Robot Interaction: Furthering the Science of HRI*. Service and Interactive Robotics, v. 1, p. 3, 2022. <https://doi.org/10.1007/s43154-022-00094-5>

KOJIMA, T.; ZHU, Y.; IWASAWA, Y.; KITAMURA, T.; YAN, G.; MORIKUNI, S.; TAKANAMI, R.; SOLANO, A.; MATSUSHIMA, T.; MURAKAMI, A.; MATSUO, Y. *A Comprehensive Survey on Physical Risk Control in the Era of Foundation Model-enabled Robotics*. International Joint Conference on Artificial Intelligence (IJCAI), p. 10517-10527, 2025. <https://doi.org/10.24963/ijcai.2025/1168>

LIN, Y.; ZHANG, Z.; TAN, Y.; FU, H.; MIN, H. *Efficient TD3 based path planning of mobile robot in dynamic environments using prioritized experience replay and LSTM*. Scientific Reports, v. 15, n. 1, p. 18331, 2025. <https://doi.org/10.1038/s41598-025-02244-z>

NEVO, M. D.; TURCATO, N.; SINIGAGLIA, A.; LORIGIOLA, R.; CASADEI, O.; CARLI, R.; CENEDESE, A.; SUSTO, G. A. *Deep Reinforcement Learning for Autonomous Navigation:*

Sim-to-Real Transfer on TurtleBots. IFAC-PapersOnLine, v. 59, n. 26, p. 13-18, 2025. <https://doi.org/10.1016/j.ifacol.2025.12.003>

TANG, C.; ABBATEMATTEO, B.; HU, J.; CHANDRA, R.; MARTÍN-MARTÍN, R.; STONE, P. *Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes*. Proceedings of the AAAI Conference on Artificial Intelligence, v. 39, n. 27, p. 28694-28698, 2025. <https://doi.org/10.1609/aaai.v39i27.35095>

XIANG, J.; LI, Q.; DONG, X.; REN, Z. *Continuous Control with Deep Reinforcement Learning for Mobile Robot Navigation*. Proceedings of the Chinese Automation Congress (CAC), p. 1501-1506, 2019. <https://doi.org/10.1109/CAC48633.2019.8996652>

NOTAS

CONTRIBUIÇÃO DE AUTORIA

Concepção e elaboração do manuscrito: F. R. Souza Junior, D. C. Ramos

Desenvolvimento de software: F. R. Souza Junior

Análise de dados: F. R. Souza Junior

Discussão dos resultados: F. R. Souza Junior, D. C. Ramos

Supervisão e orientação metodológica: D. C. Ramos

Revisão e aprovação: F. R. Souza Junior, D. C. Ramos

PREPRINTS

O manuscrito não é um preprint.

FINANCIAMENTO

Este trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, e do Projeto RoboPatos (APQ-02616-22

CONFLITO DE INTERESSES

Os autores declaram não haver conflito de interesses.

DISPONIBILIDADE DE DADOS DE PESQUISA E OUTROS MATERIAIS

Os dados experimentais gerados e analisados neste estudo, incluindo erro de rastreamento, comandos de velocidade e estados do robô, estão disponíveis publicamente em repositório de acesso aberto. Os dados podem ser acessados por meio do DOI: <https://doi.org/10.5281/zenodo.20646254>

ANUÊNCIA DE AVALIAÇÃO ABERTA

Não aplicável.



O conteúdo deste trabalho pode ser usado sob os termos da licença Creative Commons Attribution 4.0. Qualquer outra distribuição deste trabalho deve manter a atribuição ao(s) autor(es) e o título do trabalho, citação da revista e DOI.