

Mecha Electra CBR 2025 RoboCup@Home

Felipe Trentin

Depto. Eng. de Controle e Automação
Universidade Estadual Paulista-UNESP
Sorocaba, Brasil
felipe.trentin@unesp.br

Gustavo Henrique Bacci

Depto. Eng. de Controle e Automação
Universidade Estadual Paulista-UNESP
Sorocaba, Brasil
gustavo.bacci@unesp.br

Gabriel Ramalho Correa

Depto. Eng. de Controle e Automação
Universidade Estadual Paulista-UNESP
Sorocaba, Brasil
gabriel.ramalho@unesp.br

Julio Gustavo Brendler Sabka

Depto. Eng. de Controle e Automação
Universidade Estadual Paulista-UNESP
Sorocaba, Brasil
jgb.sabka@unesp.br

Pedro Augusto Lemos Gonçalves

Depto. Eng. de Controle e Automação
Universidade Estadual Paulista-UNESP
Sorocaba, Brasil
pedro.a.goncalves@unesp.br

Marilza Antunes de Lemos

Depto. Eng. de Controle e Automação
Universidade Estadual Paulista-UNESP
Sorocaba, Brasil
marilza.lemos@unesp.br

Abstract—This paper presents the Mecha Electra robotic platform, developed by the Mecha Electra team at UNESP Sorocaba for the RoboCup@Home 2025 competition. The system integrates a repurposed Festo Robotino omnidirectional mobile base, modernized with lithium-ion batteries and a ROS 2-based architecture, with a custom 5-DOF ViperX-250 robotic arm. Key innovations include a fully omnidirectional repurposed mobile base, a vision subsystem leveraging 6D object detection with pose estimation, and a modular Rasa and Llama 2 chatbot with Whisper speech recognition. The navigation stack combines a LiDAR-based AMCL and Ceres solver for localization and a Dijkstra global planner, while MoveIt enables manipulator control.

Index Terms—RoboCup Brazil, RoboCup@Home, Service Robotics, Mobile Manipulation, Human-Robot Interaction, Omnidirectional Navigation.

I. INTRODUÇÃO

A equipe do Instituto de Ciência e Tecnologia da UNESP Sorocaba (ICTS/UNESP), atualmente denominada Mecha Electra, tem um histórico de participação em competições da liga RoboCup Festo Logistics, nos eventos da Competição Brasileira de Robótica (CBR), desde 2010 [1]. Após o término da existência da liga em 2018 e também da pandemia, a equipe retomou os estudos em robótica para competição, escolhendo a liga RoboCup@Home para dar continuidade na pesquisa e desenvolvimento em robótica numa área que tem se expandido em todo o mundo: a dos robôs domésticos. Essa decisão levou em conta a possibilidade de reuso das bases móveis já existentes no ICTS/UNESP, os Robotinos da Festo [2], componentes como sensores LiDAR, Kinect, câmeras e o interesse em continuar o desenvolvimento sob o *Robot Operating System* (ROS) [3]. Algumas modificações relevantes foram necessárias para habilitar o uso da base móvel, descontinuada pelo fabricante, e que precisava suportar novas tecnologias e versões de software para um robô doméstico, assim como acrescentar um braço robótico para realizar as tarefas de manipulação.

A. Inovação

1) Inovações Mecânicas e Elétricas:

- **Construção do braço do robô:** O braço robótico foi montado a partir do braço ViperX-250 [4], braço com 5 graus de liberdade, que foi modificado para melhor se adequar a um robô móvel, constituído principalmente de motores Dynamixel e extrusões de alumínio.
- **Modernização do sistema de energia:** As baterias do robô, originalmente de chumbo, foram substituídas por baterias de íon-lítio comercialmente disponíveis com proteção e gerenciamento de bateria embutidos.

2) Integração da base móvel com ROS2:

- **Implementação do driver para ROS2:** O computador interno, antigo e obsoleto, foi removido completamente, e em seu lugar foi instalada uma interface RS232-USB da base móvel para o notebook principal do robô. Para a integração com o ROS2, um driver personalizado foi implementado, contendo os protocolos e controladores necessários. Esta solução permite controle completo da base móvel, seus atuadores e sensores, além de portas de entrada e saída.

II. SISTEMA ELÉTRICO E MECÂNICO

Por esta ser a primeira participação da equipe na RoboCup@home, o robô foi arquitetado de forma a utilizar o máximo de componentes e subsistemas prontos e comercialmente disponíveis. Como já mencionado, houve o reuso da base móvel Robotino e o acréscimo de uma estrutura em alumínio capaz de suportar o braço robótico, novos sensores, notebook e novas baterias. A estrutura foi montada a partir de extrusões de alumínio e peças estruturais personalizadas, como o suporte do braço robótico e do notebook. A Figura 1 mostra o robô Mecha Electra em sua atual configuração.

A. Configuração do Robô

O robô é formado majoritariamente por uma base móvel Robotino, uma plataforma de robótica móvel para pesquisa e



Figura 1. Robô Mecha Electra.

educação, desenvolvida pela Festo. Equipado com um acionamento omnidirecional de três rodas, em uma base circular de 352 mm de diâmetro, o robô possui flexibilidade em sua operação e livre movimento. Esse conjunto de rodas é movido, de forma individual, por motores Dunkermotoren GR 42x40. Além disso, possui um total de nove sensores infravermelhos dispostos em um ângulo de 40° entre si ao redor da base, úteis na identificação de obstáculos durante sua movimentação. Essencialmente modular, é acoplado a base um perfil de alumínio que abriga o sistema de manipulação, a interface visual e de comunicação. O robô é alimentado por três fontes principais de energia, sendo elas a bateria de 12 V do braço robótico, a bateria de 20 V da base móvel e a bateria interna do notebook. Os circuitos de 12 V e 20 V estão conectados a um botão de emergência, em circuitos distintos, para desenergização do robô. A base móvel também inclui um sensor LiDAR RPLiDAR A1, e um interruptor de contato em seu perímetro, que desativa o movimento do robô em caso de contato.

B. Braço Robótico

O braço robótico construído tem 5 graus de liberdade, alcance de 650 mm e carga útil de 450 g que pode ser visto na Figura 1. Pertence à família de braços da série ViperX da Trossen Robotics [4], com os servomotores inteligentes DYNAMIXEL XM-Series [5]. Os atuadores da série XM oferecem alto torque, dissipação de calor eficiente e grande durabilidade. Os servomotores oferecem alta resolução de 4096 posições por rotação e controlador integrado com parâmetros PID definidos pelo usuário, bem como monitoramento de temperatura, feedback posicional, tensão e corrente. A interface Robotis DYNAMIXEL U2D2 permite a integração com o

computador principal, e um ROS driver permite comunicação com o ROS2.

III. ARQUITETURA DE SOFTWARE DO ROBÔ

O *framework* principal utilizado na arquitetura do robô Mecha Electra é o ROS2, que permite alta flexibilidade e integração com diversos algoritmos de estado da arte. A arquitetura é dividida nos subsistemas de localização e mapeamento, navegação, manipulação, visão e interface humano-robô. Os subsistemas são unidos por uma máquina de estados YASMIN [6] que define o comportamento do robô através de uma máquina de estados finitos.

A. Localização e Mapeamento

Para localizar o robô na arena, assim como detectar obstáculos e fornecer uma visualização intuitiva do espaço do robô, foi implementado o método *Simultaneous Localization and Mapping* (SLAM) através do pacote SLAM Toolbox [7], configurado para utilizar o LiDAR do robô como sensor principal, por meio do algoritmo *Adaptive Monte Carlo Localization* (AMCL) [8] junto com um resolvidor Ceres, que geram e comparam um mapa local. A localização do robô é derivada de informações obtidas pela odometria da base móvel, e utiliza a correspondência do rastro do LiDAR com o mapa para refinar sua posição e resolver o problema de desvio acumulado da estimativa da posição ao longo do tempo. Um grafo com as leituras dos sensores também é gerado, o que refina o mapa conforme o robô se desloca, compensando divergências entre mapa e arena. A figura 2 mostra um exemplo de mapeamento e localização, onde o robô foi colocado em um ambiente simulado de um restaurante e mapeou o ambiente utilizando somente seus sensores embarcados.

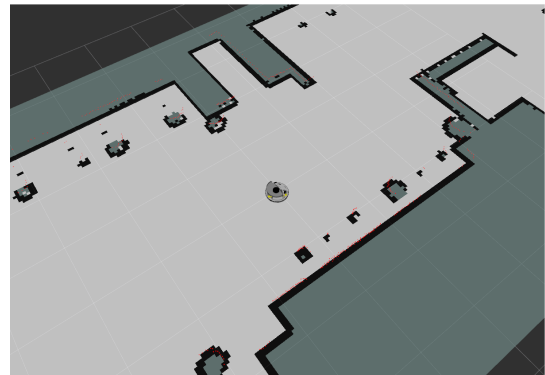


Figura 2. Ambiente mapeado pelo Robotino em simulação.

B. Navegação

Para a implementação do sistema de navegação autônoma, foi utilizado o Navigation 2 [9], um *framework* robusto e amplamente utilizado para aplicações de robótica móvel [10]. Esse pacote de ferramentas permite a geração de trajetórias otimizadas por meio do planejador NavFn, utilizando o algoritmo de Dijkstra para o cálculo da trajetória global. Posteriormente, o caminho gerado passa por um processo de suavização para

garantir continuidade e baixa aceleração no movimento, que é então encaminhado a um controlador local responsável por executar o trajeto planejado, comandando os atuadores do robô.

Adicionalmente, o sistema incorpora Árvores Comportamentais para gerenciar decisões em tempo real, permitindo ao robô lidar de forma dinâmica com situações imprevistas durante a navegação. Essa abordagem garante maior flexibilidade e robustez ao sistema, adaptando-se a diferentes cenários operacionais.

Atualmente, a equipe busca implementar um controlador projetado para explorar as capacidades holonômicas da base móvel.

C. Controle e Movimentação do Braço Robótico

Para o controle do braço robótico são utilizados os pacotes Interbotix ROS Toolboxes e Interbotix ROS Manipulators [11], que fornecem integração com o ROS2. Dentre as formas de controle disponíveis nos pacotes, foi escolhido o uso da plataforma MoveIt [12] com o planejador OMPL [13] para realizar os movimentos do braço em espaço cartesiano, assim como planejar e construir tarefas pelo MoveIt Task Constructor.

A plataforma também permite adicionar obstáculos que o braço não pode encostar, como a estrutura do robô, além de possuir ferramentas para manipulação e interação com objetos. Atualmente a equipe está desenvolvendo uma integração entre o subsistema de visão computacional e o sistema de controle do braço para realizar as tarefas de manipulação.

A Figura 3 demonstra a integração da plataforma MoveIt com o braço robótico em simulação. Em laranja, está representada a posição alvo do braço em que será gerado um plano de movimentação.

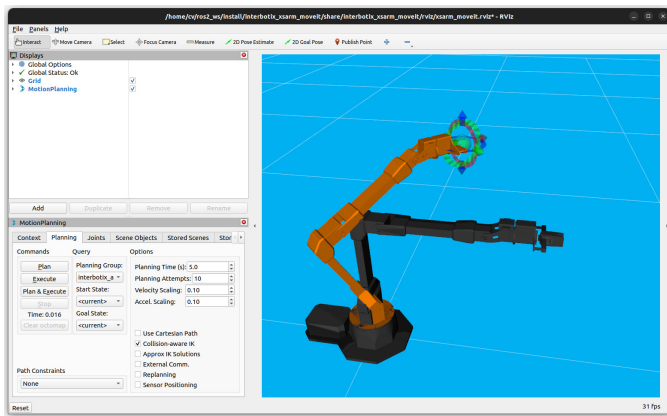


Figura 3. Representação em 3D da posição atual do braço simulado e sua posição alvo.

D. Visão de Máquina

Para a arquitetura de visão computacional foi adotada como base a ferramenta FoundationPose, desenvolvida pela NVIDIA Research (NVLabs) [14], a qual permite reconhecimento visual com alta acurácia mesmo em condições de iluminação

variável, visão parcial e com base em um número reduzido de imagens de referência. O subsistema de visão é composto por uma câmera RGB de alta resolução posicionada de forma que seu campo de visão inclua a detecção frontal de objetos em superfícies como mesas, prateleiras e bancadas. A aquisição das imagens é contínua e sincronizada com o sistema de planejamento de tarefas e navegação do robô.

A ferramenta atua em duas etapas principais: Primeiro o reconhecimento de objetos com base em exemplos visuais. Durante a fase de configuração do robô, uma base visual é construída a partir da captura de imagens dos objetos que o robô poderá manipular ou reconhecer em prova. Essas imagens de referência são então utilizadas para treinar internamente uma rede de extração de características visuais, baseada em modelos fundacionais, capazes de detectar *keypoints* dentro das detecções. Após isso, é realizada a estimativa de pose 6D com correspondência geométrica. No momento da inferência, o FoundationPose compara a imagem da cena atual com os dados de referência, identificando os objetos presentes e estimando sua pose 6D com relação à câmera. A estimativa é feita por meio da correspondência entre os *keypoints* 2D da imagem atual e os pontos 3D inferidos dos objetos, resolvendo o problema Perspective-n-Point (PnP) e aplicando refinamentos geométricos. O resultado final é uma matriz de transformação homogênea que representa a pose do objeto no sistema de coordenadas do robô.

Essas informações são integradas ao sistema de transformações do ROS 2, permitindo que módulos de navegação, manipulação robótica e interação homem-máquina utilizem as poses estimadas de acordo com suas necessidades.

O principal diferencial dessa abordagem é a sua generalização e eficiência: o sistema não exige treinamento extenso para novos objetos e pode ser rapidamente adaptado a novas configurações de ambiente, como reorganizações de objetos em uma casa.

Para o reconhecimento do operador, foi utilizada uma câmera Intel Realsense F455 capaz de realizar reconhecimento facial. A figura 4 mostra a IDE da câmera, onde é possível cadastrar e reconhecer pessoas, assim como modificar parâmetros de detecção.

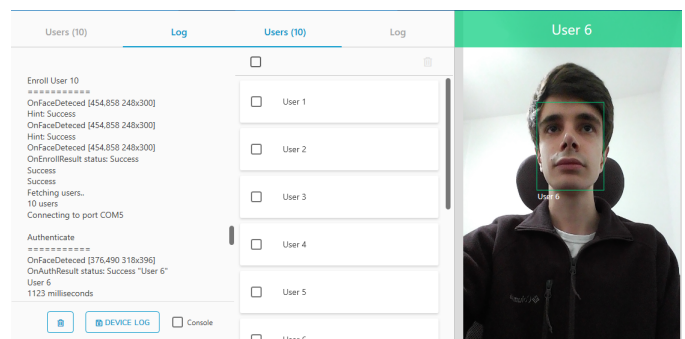


Figura 4. Teste de reconhecimento facial da câmera F455

Foram conduzidos testes experimentais para avaliar a robustez do sistema de reconhecimento baseado em visão computacional, considerando diferentes condições de distância. A Tabela I mostra os resultados do experimento, que contou com a participação de 10 usuários em um ambiente controlado, onde foram registrados três parâmetros fundamentais para a análise de desempenho: tempo de detecção, distância do usuário em relação à câmera, e índice de confiança do reconhecimento.

Tabela I
RESULTADOS DE DETECÇÃO DE USUÁRIOS

Usuário	Tempo de Detecção (ms)	Distância (m)	Confiança
Usuário 1	1112	1.5	Alta
Usuário 2	1113	2.1	Média
Usuário 3	1092	0.8	Alta
Usuário 4	1138	1.2	Alta
Usuário 5	1110	2.5	Média
Usuário 6	1115	1.1	Alta
Usuário 7	1099	0.7	Alta
Usuário 8	1093	1.8	Média
Usuário 9	1125	2.0	Alta
Usuário 10	1071	0.9	Alta

E. Interação Humano Robô

Para os desafios envolvendo Interação Humano-Robô (*Human Robot Interaction-HRI*), o robô Mecha Electra foi equipado com um sistema de *chatbot* capaz de realizar reconhecimento de voz e gerar respostas adequadas a comandos ou perguntas vindas de operadores. Neste contexto, é importante destacar que o *chatbot* implementado foi concebido seguindo as boas práticas de Dale [15], que define *chatbots* como aplicações de software que simulam conversação humana por comandos de texto ou voz.

O *Rasa Open Source* [16] foi o modelo selecionado por viabilizar a personalização modular de diversos atributos, permitindo definir intenções, entidades, fluxos de diálogo e ações customizadas [16]. A figura 5 apresenta a arquitetura do sistema HRI utilizada no projeto.

O *chatbot* foi programado para reconhecer perguntas dentro de contextos fixos, como a categoria da competição e as informações da cidade sede do evento. O robô opera com tecnologia de Inteligência Artificial em *Natural Language Processing* (NLP) supervisionado, e cada entrada do usuário é classificada em intenções e processada de modo a fornecer a resposta mais assertiva. Embora as respostas sejam pré-definidas, a estrutura do modelo permite rápida atualização do sistema para capacitar conversação sobre outros assuntos.

O sistema de interação por voz do Mecha Electra é composto por módulos integrados responsáveis por processar comandos naturais. Na Figura 5, a interação começa com a captura de áudio por um microfone, que é processado pelo sistema de reconhecimento de voz Whisper [17], convertendo a fala em texto. Esse texto é enviado ao Rasa, que realiza o reconhecimento de intenção e entidade. Caso a intenção seja reconhecida com confiança, uma resposta pré-definida na programação do *chatbot* é retornada. Se não for reconhecida, um *fallback* ativa uma *custom action* com um modelo de



Figura 5. Fluxo de decisão do sistema Rasa com fallback para modelo Llama.

linguagem, integrando o modelo Llama [18] para gerar uma resposta textual alternativa. A resposta, seja de ação ou *fallback*, é convertida em áudio pela ferramenta OpenTTS [19] e reproduzida ao usuário por um alto-falante.

O grande diferencial técnico do projeto HRI está na integração com serviços externos via *custom actions*, recurso nativo do Rasa integrado à sua arquitetura. Este permite que o robô execute tarefas além de fornecer as respostas básicas pré-definidas, como consultas em APIs, manipulação de banco de dados ou acionamento de módulos de IA generativa. No Rasa, o recurso *fallback* é viabilizado via *custom actions*.

O mecanismo de *fallback* é uma estratégia de resposta de segurança, acionada quando o sistema de *Natural Language Understanding* (NLU) não atinge um nível de confiança mínimo na classificação de uma entrada. *Fallback handlers* são essenciais para lidar com entradas inesperadas, evitando rupturas na experiência de interação [20]. O Rasa implementa esse mecanismo nativamente por meio do recurso *fallback action*, que pode ser usado para retornar respostas genéricas ou acionar fluxos alternativos [16]. Neste projeto, o *fallback* do *chatbot* gera uma resposta fora da base treinada do Rasa, uma vez que foi integrado com uma IA Llama [18], um modelo LLM (*Large Language Model*) capaz de gerar respostas livres ampliando o alcance e flexibilidade da solução.

A Figura 6 apresenta o fluxo de informações na arquitetura do sistema HRI utilizada no projeto. O sistema responde corretamente perguntas sobre informações específicas, como o estado-sede da RoboCup@Home Brasil e dúvidas sobre a competição, além de apresentar capacidade de lidar com *intents* novas adicionadas recentemente com respostas contextuais e coerentes.


```

Your input -> hello!
Hey there! How can I help you?
Your input -> what is robocup @home?
RoboCup @Home is a competition that aims to advance robotics for domestic environments.
Your input -> who can participate?
Anyone with a functional robot can register and join, after one test.
Your input -> how are challenges scored?
Scoring is based on test completion, time, and interaction.
Your input -> hello!
Hey there! How can I help you?
Your input -> what is robocup @home?
RoboCup @Home is a competition that aims to advance robotics for domestic environments.
Your input -> who can participate?
Anyone with a functional robot can register and join, after one test.
Your input -> how are challenges scored?
Scoring is based on test completion, time, and interaction.
Your input -> give me one scoring criterion
An example criterion is 'Task completion time'.
Your input -> what state is hosting @home Brazil?
In 2025, RoboCup @Home Brazil will be held in the state of Espírito Santo. Espírito Santo is
in the Southeast region of Brazil, with Vitória as its capital, known for its ports, beaches,
and balanced economy.

```

Figura 6. Testes na Interface Humano Robô do Mecha Electra.

F. Execução de tarefas baseada em Máquina de estados finitos

A fim de coordenar os subsistemas do robô para execução de tarefas complexas, foram implementadas máquinas de estados finitos, desenvolvidas com a biblioteca YASMIN. Cada máquina de estado implementa uma estratégia para realização de uma tarefa complexa, como mostra a figura 7. Cada estado representa uma ação que o robô deve executar. O final da execução gera uma pré-condição que habilita a execução de um próximo estado.

A Figura 7, representa a tarefa de limpar a mesa, em que o robô deve retirar os itens da mesa e levá-los ate uma bandeja em outro local. Ao iniciar a tarefa, o robô primeiro deve navegar até a mesa, nesse ponto ele deve detectar os objetos e a bandeja destino. Na sequencia, o robô deve segurar o objeto e colocá-lo na bandeja. Caso o robô não detecte objetos na mesa, é realizada uma verificação final e a máquina de estados finaliza. Caso o robô não detecte a bandeja ou falhe na navegação, a máquina de estados pode finalizar com erro e o robô pode solicitar assistência humana.

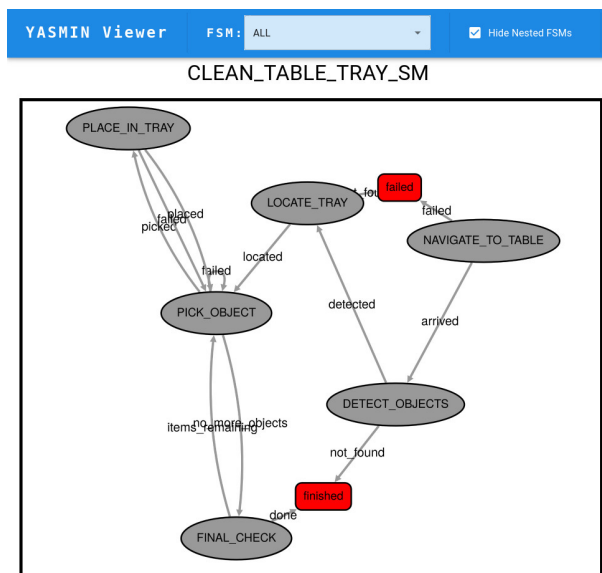


Figura 7. Máquina de estados implementada para limpar a mesa.

AGRADECIMENTOS

A equipe registra agradecimentos à PROGRAD/UNESP, ICTS/UNESP e GASI/UNESP. Este estudo foi suportado pela PROGRAD/UNESP na forma de auxílio e bolsas.

REFERÊNCIAS

- [1] Mecha Electra. Mecha electra homepage. [Online]. Available: <https://www.sorocaba.unesp.br/#!/comunidade/grupos-de-competicao/home/>
- [2] Festo Didactic. Robotino®: plataforma móvel autônoma para ensino e pesquisa. [Online]. Available: <https://ip.festo-didactic.com/InfoPortal/Robotino/Overview/EN/index.html>
- [3] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. J. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>
- [4] Trossen Robotics / Interbotix. Viperx-250 robot arm specifications — interbotix x-series arms. [Online]. Available: https://docs.trossenrobotics.com/interbotix_xsarms_docs/specifications/vx250.html
- [5] ROBOTIS Co., Ltd., *DYNAMIXEL XM540-W270 (Protocol 2.0) eManual*, ROBOTIS, 2025, acesso em: 28 jul. 2025. [Online]. Available: <https://emanual.robotis.com/docs/en/dxl/x/xm540-w270/>
- [6] M. González-Santamarta, F. Rodríguez-Lera, C. Llamas, F. Rico, and V. Olivera, "Yasmin: Yet another state machine library for ros 2," *arXiv preprint arXiv:2205.13284*, 2022, disponível em <https://arxiv.org/abs/2205.13284>.
- [7] S. Macenski and I. Jambrecic, "Slam toolbox: Slam for the dynamic world," *Journal of Open Source Software*, vol. 6, no. 61, p. 2783, 2021. [Online]. Available: <https://doi.org/10.21105/joss.02783>
- [8] S. He, T. Song, P. Wang, C. Ding, and X. Wu, "An enhanced adaptive monte carlo localization for service robots in dynamic and featureless environments," *Journal of Intelligent and Robotic Systems*, vol. 108, no. 1, May 2023. [Online]. Available: <http://dx.doi.org/10.1007/s10846-023-01858-7>
- [9] S. Macenski, T. Moore, D. V. Lu, A. Merzlyakov, and M. Ferguson, "From the desks of ros maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2," *Robotics and Autonomous Systems*, 2023.
- [10] A. K. Kashyap and K. Konathalapalli, "Autonomous navigation of ros2 based turtlebot3 in static and dynamic environments using intelligent approach," *International Journal of Information Technology*, Mar. 2025. [Online]. Available: <http://dx.doi.org/10.1007/s41870-025-02500-5>
- [11] S. Wiznitzer, L. Schmitt, and M. Trossen, "interbotix_ros_manipulators." [Online]. Available: https://github.com/Interbotix/interbotix_ros_manipulators
- [12] P. R. . M. Developers. Moveit 2 documentation. [Online]. Available: <https://moveit.picknik.ai/>
- [13] I. A. Şucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec. 2012, <https://ompl.kavrakilab.org>.
- [14] S. B. Wen, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [15] R. Dale, "The return of the chatbots," *Natural Language Engineering*, vol. 22, no. 5, pp. 811–817, 2016.
- [16] T. Bocklisch *et al.*, "Rasa: Open source language understanding and dialogue management," *arXiv preprint arXiv:1712.05181*, 2017, disponível em <https://arxiv.org/abs/1712.05181>. Acesso em: 28 jul. 2025.
- [17] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," OpenAI, 2022, acesso em: 28 jul. 2025. [Online]. Available: <https://arxiv.org/abs/2212.04356>
- [18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. Xiang Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez,

R. Stojnic, S. Edunov, and T. Scialom, “Llama 2: Open foundation and fine-tuned chat models,” *CoRR*, vol. abs/2307.09288, 2023. [Online]. Available: <https://arxiv.org/abs/2307.09288>

- [19] E. Casanova, C. Shulby, E. Gölge, A. C. Junior *et al.*, “Opentts: A modular, open-source text-to-speech framework,” *Software open-source*, 2023, acesso em: 28 jul. 2025. [Online]. Available: <https://github.com/synesthesiam/opentts>
- [20] N. Radziwill and M. Benton, “Evaluating quality of chatbots and intelligent conversational agents,” *Journal of Software Quality Professional*, vol. 19, no. 3, pp. 25–36, 2017.