

## PROGRAMAÇÃO DE MOVIMENTOS DO MANIPULADOR *OPENMANIPULATOR-X USANDO ROS*

Milena Martins de Oliveira Krauss, [d202310304@uftm.edu.br](mailto:d202310304@uftm.edu.br)<sup>1</sup>  
Guilherme Augusto Nascimento, [d202110599@uftm.edu.br](mailto:d202110599@uftm.edu.br)<sup>1</sup>  
Fabian Andres Lara-Molina, [fabian.molina@uftm.edu.br](mailto:fabian.molina@uftm.edu.br)<sup>2</sup>

<sup>1</sup>Univesidade Federal do Triângulo Mineiro, Departamento de Engenharia Elétrica, Av. Randolfo Borges Júnior, 1400 - Univerdecidade, Uberaba - MG, 38064-200

<sup>2</sup>Univesidade Federal do Triângulo Mineiro, Departamento de Engenharia Mecânica, Av. Randolfo Borges Júnior, 1400 - Univerdecidade, Uberaba - MG, 38064-200

*Na atualidade, diversos manipuladores são inseridos em nosso cotidiano para realizar tarefas em ambientes não estruturados, como logística, ambientes domésticos e até mesmo resgate em desastres. Programar os movimentos dos manipuladores na realização das diversas tarefas permite adaptabilidade em diferentes cenários. Portanto, tem-se como objetivo implementar, em Python a programação dos movimentos do OpenManipulator-X, assim como analisar e comparar a resposta obtida por simulação numérica e dados do protótipo experimental do sistema. Tendo isso em vista, foi utilizado o ROS (Robotic Operational System), um framework colaborativo que reúne bibliotecas e ferramentas para facilitar o desenvolvimento de software na área da robótica. O openMANIPULATOR-X é um hardware de um braço articulado de 5 graus de liberdade, sendo controlado por pacotes em simulação. A metodologia foi avaliada mediante simulação computacional mostrando a simulação dos movimentos do manipulador.*

**Palavras chave:** *Robótica, Manipuladores Robóticos, Programação, Robotic Operational System.*

*Currently, several manipulators are inserted into our daily lives to perform tasks in unstructured environments, such as logistics, domestic environments and even disaster rescue. Programming the movements of the manipulators to perform different tasks allows adaptability in different scenarios. Therefore, the objective is to implement the programming of OpenManipulator-X movements in Python, as well as analyze and compare the response obtained by numerical simulation and data from the experimental prototype of the system. With this in mind, ROS (Robotic Operational System) was used, a collaborative framework that brings together libraries and tools to facilitate software development in the area of robotics. openMANIPULATOR-X is an articulated arm hardware with 5 degrees of freedom, controlled by packages in simulation. The methodology was evaluated through computer simulation showing the simulation of the manipulator's movements.*

**Keywords:** *Robotics, Robotic Manipulators, Programming, Robotic Operational System.*

### 1. INTRODUÇÃO

Na era contemporânea, assistimos à crescente e ubíqua presença de manipuladores robóticos, os quais desempenham papéis fundamentais em vários cenários desafiadores e não estruturados (Machida *et al.*, 2022). Essa onipresença se manifesta em uma variedade de contextos, desde otimização logística até a execução de tarefas complexas em ambientes domésticos, além de sua participação essencial em operações de resgate durante desastres. A versatilidade desses manipuladores torna-se evidente, destacando a importância crítica da programação precisa de seus movimentos (Molina, 2012). Essa capacidade não apenas confere agilidade, mas também a adaptabilidade necessária para enfrentar uma diversidade de cenários complexos. Neste sentido diversos trabalhos de pesquisa têm sido desenvolvidos visando melhorar a resposta dinâmica nos movimentos dos manipuladores robóticos (Lara-Molina *et al.*, 2014; Costa *et al.*, 2018; Lara-Molina *et al.*, 2018, 2011, 2015).

Além da implementação, conduzimos uma análise aprofundada e uma comparação abrangente. Investigaremos não apenas as respostas obtidas por meio de simulações numéricas, mas também os dados oriundos do protótipo experimental do sistema. Essa abordagem holística visa não apenas validar a eficácia do modelo proposto, mas também compreender profundamente suas nuances de desempenho em cenários do mundo real.

Em meio ao cenário desafiador e em constante evolução que caracteriza o ambiente tecnológico, a adoção e estudo do Robotic Operational System (ROS) assumem um papel de relevância incontestável. Pesquisas recentes, segundo contribuições da comunidade (Joseph and Cacace, 2018), destacam o ROS não apenas como uma plataforma, mas como um framework robusto que agrega bibliotecas e ferramentas desenvolvidas de maneira colaborativa pela comunidade robóti-

cista. Sua essência reside na simplificação do desenvolvimento de softwares, oferecendo códigos abertos provenientes de especialistas que, de forma colaborativa, contribuem para aprimorar e expandir o universo da robótica.

O ROS tem sido amplamente utilizado em diversos trabalhos científicos no campo da robótica. Um estudo realizado por Kaoud Abdelaziz (2020) explorou a aplicação do ROS em sistemas autônomos de navegação subaquática, demonstrando sua eficácia na integração de sensores, controle de movimento e comunicação em ambientes aquáticos desafiadores. Além disso, a pesquisa de Ferrari *et al.* (2024) investigou a implementação do ROS em sistemas de robótica médica para cirurgias assistidas por robô, destacando sua capacidade de fornecer uma plataforma flexível e modular para desenvolver algoritmos de planejamento de trajetória e controle de movimento. Outro estudo relevante foi conduzido por Jing *et al.* (2018), que utilizou o ROS para desenvolver um sistema de inspeção robótica em ambientes industriais, mostrando como o framework facilita a integração de sensores e a execução de tarefas complexas de manipulação e navegação. Esses trabalhos exemplificam a versatilidade e a importância do ROS como uma ferramenta fundamental para pesquisa e desenvolvimento em robótica, abrindo caminho para avanços significativos em uma variedade de aplicações.

O OpenMANIPULATOR-X tem sido adotado em vários estudos científicos como uma plataforma de manipulação robótica versátil e de código aberto. Uma pesquisa conduzida por Kim and Ryu (2023) explorou a aplicação do OpenMANIPULATOR-X em tarefas de montagem e manuseio de objetos em ambientes industriais, destacando sua precisão e eficiência no cumprimento de tarefas repetitivas com alta precisão. Além disso, o trabalho de Zhou (2023) investigou a integração do OpenMANIPULATOR-X em sistemas de assistência em saúde, demonstrando sua capacidade de realizar procedimentos médicos delicados com segurança e precisão. Outro estudo relevante foi realizado por Lexau (2022), que utilizou o OpenMANIPULATOR-X em pesquisas de interação humano-robô, explorando sua aplicação em ambientes de convivência humana e sua capacidade de realizar tarefas colaborativas de forma eficiente. Esses estudos exemplificam a versatilidade e o potencial do OpenMANIPULATOR-X como uma plataforma de pesquisa em robótica, contribuindo para avanços significativos em uma variedade de domínios de aplicação.

O presente trabalho tem como objetivo desenvolver um programa baseado em ROS para programar os movimentos do OpenMANIPULATOR-X e assim movimentar o manipulador visando realizar uma determinada tarefa. O restante do manuscrito está organizado em cinco seções. Na seção 2 é apresentada a metodologia e materiais usados. Os resultados baseados em simulação foram apresentados na seção 3. Finalmente, a seção 4 apresenta as conclusões.

## 2. MATERIAIS E MÉTODOS

### 2.1 *Robotic Operational System- ROS*

O Sistema Operacional Robótico (ROS) é uma plataforma de código aberto desenvolvida inicialmente pela Willow Garage e atualmente mantida pela Open Robotics. Desde sua introdução, o ROS tornou-se uma ferramenta fundamental para o desenvolvimento de sistemas robóticos em uma variedade de domínios. Neste artigo, fornecemos uma visão geral do ROS, discutindo sua definição, vantagens e aplicações em diferentes contextos da robótica (Quigley *et al.*, 2009).

O ROS é um conjunto de ferramentas, bibliotecas e convenções que visam simplificar o desenvolvimento de software para robótica. Ele fornece uma plataforma flexível e distribuída que permite aos desenvolvedores escreverem software robusto e modular para uma ampla variedade de robôs. O ROS facilita a integração de diferentes sensores, atuadores e algoritmos de controle, oferecendo uma abstração de hardware que simplifica o processo de desenvolvimento.

As vantagens do ROS incluem sua arquitetura modular, comunidade ativa de desenvolvedores, ferramentas integradas e suporte a múltiplas plataformas. A arquitetura modular do ROS divide os sistemas robóticos em unidades de software independentes chamadas de "pacotes", permitindo que os desenvolvedores escrevam e testem módulos de software separadamente, facilitando a reutilização e a manutenção do código. A comunidade global de desenvolvedores e pesquisadores do ROS promove a inovação e a rápida adoção de novas tecnologias na área da robótica. As ferramentas integradas do ROS facilitam o desenvolvimento e teste de algoritmos robóticos em um ambiente virtual antes da implantação em hardware real. Além disso, o ROS é compatível com uma variedade de sistemas operacionais, tornando-o uma escolha versátil para desenvolvedores em diferentes plataformas.

O ROS é amplamente utilizado em diversos domínios da robótica, incluindo robótica móvel, manipulação robótica, robótica aérea e subaquática. Na robótica móvel, o ROS é usado no desenvolvimento de robôs móveis autônomos para aplicações como mapeamento, navegação, patrulha de segurança e logística interna. Na manipulação robótica, o ROS é utilizado para controlar braços robóticos industriais e manipuladores móveis em ambientes como linhas de montagem, centros de distribuição e ambientes de pesquisa. Além disso, o ROS é adaptado para aplicações de robótica aérea (drones) e subaquática, permitindo o desenvolvimento de sistemas autônomos para inspeção, mapeamento e monitoramento ambiental Joseph and Cacace (2018).

Em resumo, o Sistema Operacional Robótico (ROS) é uma plataforma versátil e poderosa para o desenvolvimento de sistemas robóticos. Sua arquitetura modular, comunidade ativa e ampla gama de ferramentas integradas o tornam uma escolha popular entre os desenvolvedores e pesquisadores na área da robótica. Com suas vantagens e aplicações diversas, o ROS continua a impulsionar a inovação e o avanço da tecnologia robótica em todo o mundo.

## 2.2 OpenMANIPULATOR-X

O OpenMANIPULATOR-X é um manipulador robótico de código aberto desenvolvido pela ROBOTIS, uma empresa líder no fornecimento de soluções robóticas. Projetado para ser acessível, flexível e de fácil utilização, o OpenMANIPULATOR-X é amplamente utilizado em ambientes acadêmicos, de pesquisa e industriais para uma variedade de aplicações robóticas. Neste artigo, fornecemos uma visão geral do OpenMANIPULATOR-X, discutindo suas características, aplicações e contribuições para a robótica.

O OpenMANIPULATOR-X apresenta uma arquitetura mecânica modular e leve, composta por links de alumínio e atuadores dinâmicos. Ele é equipado com uma variedade de sensores integrados, incluindo encoders, IMUs (unidade de medição de inercial) e sensores de proximidade, que permitem um controle preciso e feedback sensorial em tempo real. Além disso, o OpenMANIPULATOR-X é totalmente compatível com o ROS (Robot Operating System), facilitando o desenvolvimento de software e a integração com outros sistemas robóticos (ver Fig. 1).

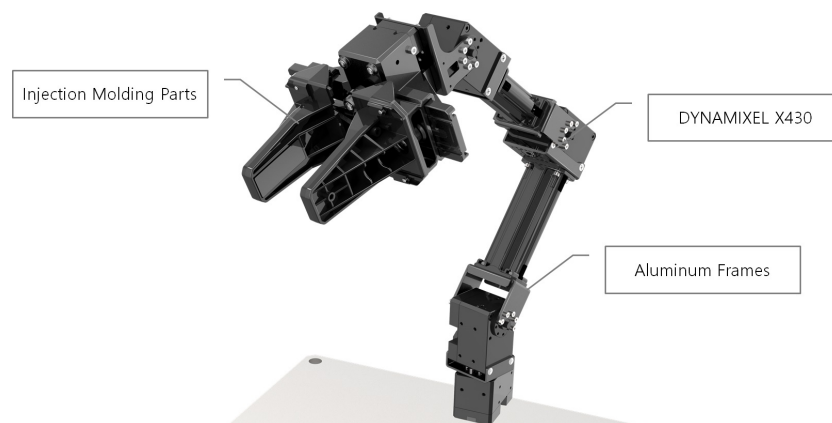


Figura 1: Manipulador Robótico OpenMANIPULATOR-X (ROBOTIS, 2022)

O OpenMANIPULATOR-X é utilizado em uma ampla gama de aplicações robóticas, incluindo manipulação de objetos, montagem automatizada, inspeção visual, pesquisa em inteligência artificial e educação em robótica. Sua versatilidade e facilidade de uso o tornam uma escolha popular entre pesquisadores, educadores e desenvolvedores de protótipos que buscam uma plataforma acessível e flexível para experimentação e desenvolvimento.

O OpenMANIPULATOR-X tem contribuído significativamente para o avanço da tecnologia robótica, fornecendo uma plataforma acessível e de código aberto para pesquisa e desenvolvimento. Sua integração com o ROS facilitou a colaboração e o intercâmbio de recursos entre a comunidade robótica, resultando em avanços rápidos e inovações na área. Além disso, o OpenMANIPULATOR-X tem sido fundamental no treinamento de estudantes e profissionais em robótica, preparando-os para enfrentar os desafios do futuro da automação e da inteligência artificial.

O OpenMANIPULATOR-X é uma ferramenta poderosa e versátil para pesquisa, educação e desenvolvimento de protótipos em robótica. Com suas características inovadoras, ampla gama de aplicações e contribuições para o avanço da tecnologia robótica, o OpenMANIPULATOR-X continua a desempenhar um papel crucial na transformação do cenário robótico global. Seu impacto na comunidade acadêmica e industrial é evidente, impulsionando a inovação e promovendo o progresso em direção a uma sociedade mais automatizada e tecnologicamente avançada.

## 2.3 Implementação e Planejamento dos Movimentos

Afim de automatizar o processo, um programa em *Python* foi adaptado com base em (Arruda., 2020). O código começa importando as bibliotecas necessárias para se comunicar com o ROS e para gerenciar o tempo de espera entre os movimentos. Isso inclui `sys`, `rospy` (para interagir com o ROS), `SetKinematicsPose` e `SetJointPosition` (para os serviços ROS), e `sleep` (para pausar a execução do código).

O código define três funções principais para controlar o OpenMANIPULATOR-X:

- `set_position_client(x, y, z, time)`: Esta função move o manipulador para uma posição específica no espaço, especificada pelos argumentos `x`, `y` e `z`. O argumento `time` define o tempo que o movimento deve levar.
- `set_joint_position(joint_pos, time)`: Esta função move as articulações do manipulador para uma configuração específica, especificada pela lista de posições das articulações `joint_pos`. O argumento `time` define o tempo que o movimento deve levar.
- `set_gripper(joint_pos, time)`: Esta função controla a abertura/fechamento da garra do manipulador. O argumento `joint_pos` especifica a posição da garra, e `time` define o tempo que o movimento deve levar.

Uma matriz de trajetória é definida, na qual cada linha representa um ponto no espaço que o manipulador deve percorrer. Cada ponto é definido por suas coordenadas  $x$ ,  $y$ ,  $z$  e um tempo de trajetória. Um loop é iniciado para percorrer cada linha da matriz de trajetória. Em cada iteração, o manipulador é movido para a posição especificada na matriz e espera-se um tempo antes de prosseguir para o próximo ponto. No final do script, são feitas chamadas adicionais para controlar as articulações do manipulador e a garra, ajustando sua posição e estado conforme necessário.

Essencialmente, o código funciona movendo o OpenMANIPULATOR-X através de uma série de posições no espaço especificadas em uma matriz de trajetória. Cada posição é alcançada usando as funções de controle definidas, que se comunicam com os serviços ROS responsáveis por mover as articulações e a garra do manipulador para as posições desejadas. O tempo entre os movimentos é controlado para garantir uma execução suave e coordenada dos movimentos do manipulador. O diagrama do algoritmo é exibido na Fig. 2.

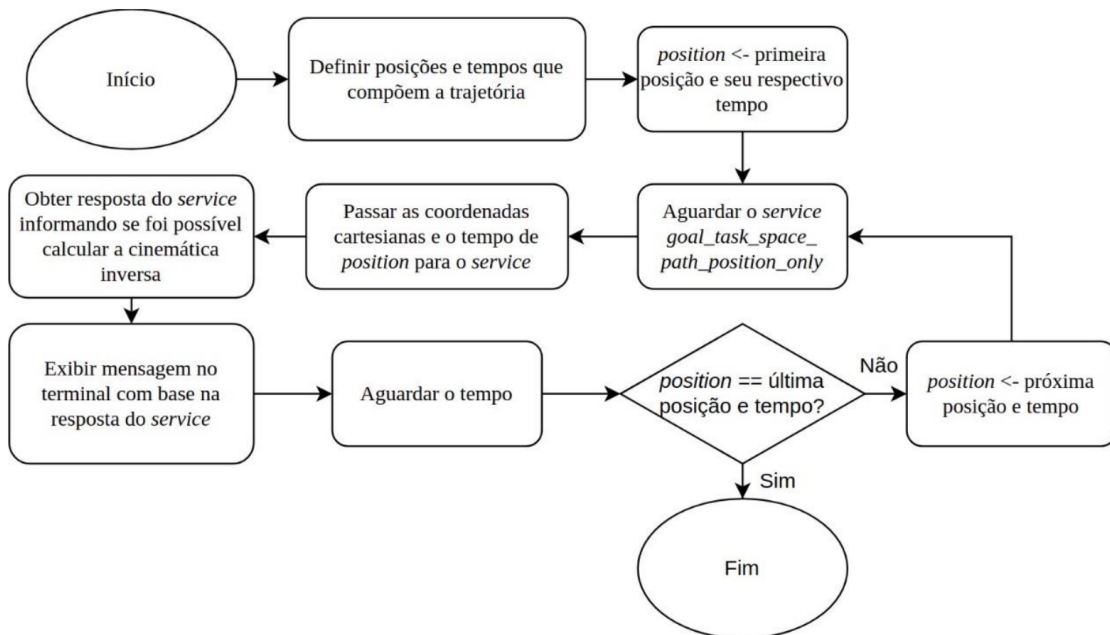


Figura 2: Algoritmo do programa.

### 3. RESULTADOS

Após as execuções dos devidos comandos, a janela do Gazebo foi aberta com a simulação pausada, conforme mostra a Fig. 3. Ao iniciar a simulação e aguardar, não foram exibidas mensagens de erro nos terminais e o manipulador manteve-se em sua posição inicial. O próximo passo era, então, tentar operá-lo.

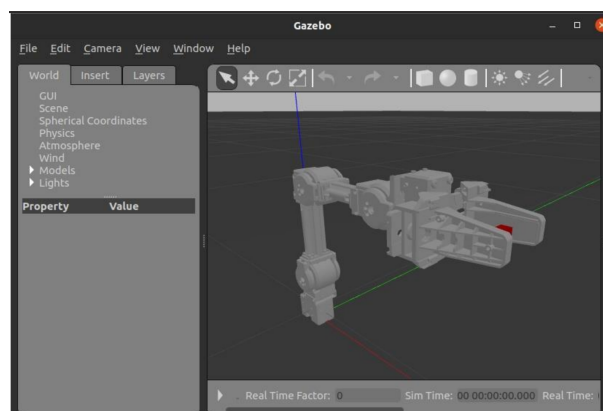


Figura 3: Simulação no Gazebo.

Na simulação, o robô foi capaz de seguir todas as trajetórias submetidas, sendo uma delas de 4 posições ilustrada na Fig. 4. Seus movimentos mostraram-se suaves, com boa precisão e executados no tempo certo, evidenciando que o controle dos motores e do manipulador como um todo foi bem implementado e configurado.

Durante a execução do algoritmo, o gráfico das coordenadas  $x$ ,  $y$  e  $z$  pôde ser gerado sem problemas para os robôs

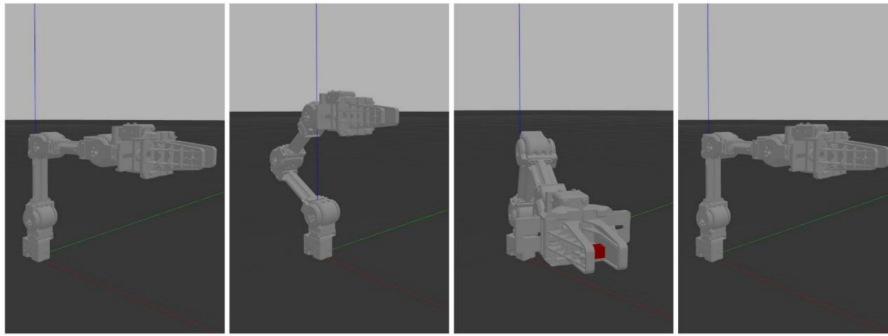


Figura 4: Trajetória do manipulador simulado.

real e simulado (ver Fig. 5); este resultado permite avaliar o movimento do efetuador final no espaço Cartesiano para os movimentos programados no planejamento das trajetórias, desta forma é possível observar separadamente a evolução temporal das coordenadas Cartesianas do movimento do efetuador final durante a simulação que também corresponde aos movimentos implementados experimentalmente no manipulador. Os resultados similares entre os ambientes analisados revelam quão semelhantes esses ambientes também são, um excelente resultado pois é desejável que a simulação seja o mais fiel possível da realidade.

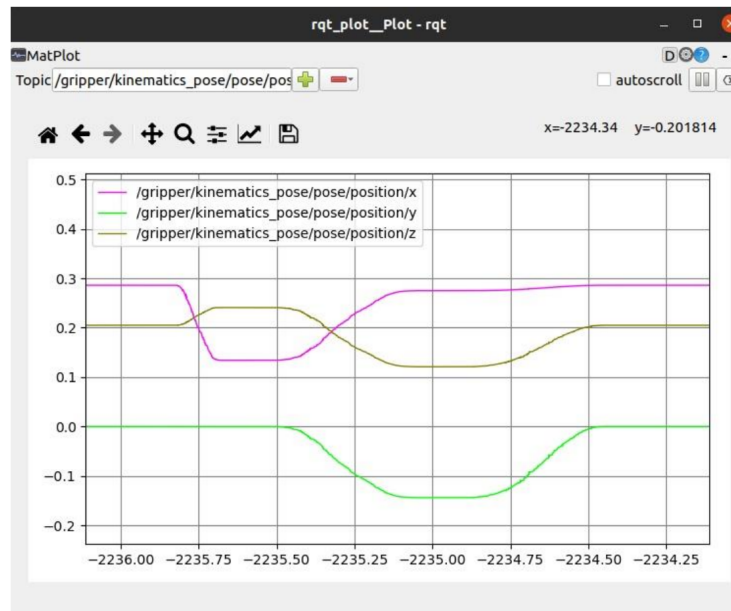


Figura 5: Trajetória do manipulador simulado.

#### 4. CONCLUSÕES

Este trabalho teve como objetivo principal tornar o manipulador OpenMANIPULATOR-X apto para alcançar objetos em ambientes dinâmicos mediante a programação dos movimentos. Para cumprir esse objetivo, foi necessário familiarizar-se com a plataforma OpenMANIPULATOR-X, testar o funcionamento do manipulador e automatizar sua trajetória mediante simulação.

Os resultados positivos nos testes permitem afirmar que o objetivo principal foi cumprido, estando o OpenMANIPULATOR-X apto para alcançar objetos em ambientes dinâmicos. Pretende-se estudar, em trabalhos futuros, o emprego de *Deep Reinforcement Learning* para controlar o manipulador robótico durante essa tarefa.

#### 5. REFERÊNCIAS

- Arruda., M., 2020. "Programming a service client for robotis open manipulator". <https://www.theconstruct.ai/programming-a-service-client-for-robotis-open-manipulator/>.
- Costa, T.L., Lara-Molina, F.A., Junior, A.A.C. and Taketa, E., 2018. "Robust  $h_{\infty}$  computed torque control for manipulators". *IEEE Latin America Transactions*, Vol. 16, No. 2, pp. 398–407.

- Ferrari, L., Nicolaou, S. and Adams, K., 2024. “Implementation of a robotic surgical practice in inflammatory bowel disease”. *Journal of Robotic Surgery*, Vol. 18, No. 1, p. 57.
- Jing, W., Goh, C.F., Rajaraman, M., Gao, F., Park, S., Liu, Y. and Shimada, K., 2018. “A computational framework for automatic online path generation of robotic inspection tasks via coverage planning and reinforcement learning”. *IEEE Access*, Vol. 6, pp. 54854–54864.
- Joseph, L. and Cacace, J., 2018. *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing Ltd.
- Kaoud Abdelaziz, S., 2020. *Vision-aided Inertial System for Near-real-time Positioning and Navigation of Unmanned Ground Survey Vehicles in GNSS-denied Environments*. Ph.D. thesis.
- Kim, S.C. and Ryu, S., 2023. “Robotic kinesthesia: Estimating object geometry and material with robot’s haptic senses”. *IEEE Transactions on Haptics*.
- Lara-Molina, F., Takano, K. and Koroishi, E., 2015. “Set-point regulation of a robot manipulator with flexible joints using fuzzy control”. In *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*. IEEE, pp. 103–108.
- Lara-Molina, F.A., Dumur, D. and Assolari Takano, K., 2018. “Multi-objective optimal design of flexible-joint parallel robot”. *Engineering Computations*, Vol. 35, No. 8, pp. 2775–2801.
- Lara-Molina, F.A., Rosário, J.M. and Dumur, D., 2011. “Robust generalized predictive control of stewart-gough platform”. In *IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, 2011 IEEE*. IEEE, pp. 1–6.
- Lara-Molina, F.A., Rosário, J.M., Dumur, D. and Wenger, P., 2014. “Robust generalized predictive control of the orthoglide robot”. *Industrial Robot: An International Journal*, Vol. 41, No. 3, pp. 275–285.
- Lexau, S.J.N., 2022. *Combining Reinforcement Learning and Robotic Vision for Human-Machine Cooperation*. Master’s thesis, NTNU.
- Machida, J.S., Lara-Molina, F.A. and Koroishi, E.H., 2022. “Geração de trajetórias de manipuladores robóticos usando deep reinforcement learning”. *Revista Mundi Engenharia, Tecnologia e Gestão (ISSN: 2525-4782)*, Vol. 7, No. 5.
- Molina, F.A.L., 2012. *Simulação e implementação experimental de um controlador preditivo generalizado para um robô orthoglide baseado na modelagem dinâmica*. Ph.D. thesis, [Universidade Estadual de Campinas].
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y. *et al.*, 2009. “Ros: an open-source robot operating system”. In *ICRA workshop on open source software*. Kobe, Japan, Vol. 3, p. 5.
- ROBOTIS, 2022. “Openmanipulator-x”. [https://emmanual.robotis.com/docs/en/platform/openmanipulator\\_x/overview/](https://emmanual.robotis.com/docs/en/platform/openmanipulator_x/overview/).
- Zhou, Z., 2023. “A mobile robot with a manipulator to alleviate the shortage of health workers in hospitals in covid 19”. *Highlights in Science, Engineering and Technology*, Vol. 39, pp. 872–880.

## 6. RESPONSABILIDADE PELAS INFORMAÇÕES

Os autores são os únicos responsáveis pelas informações incluídas neste trabalho.