



# Estudo e adaptação de um manipulador robótico para ambientes dinâmicos

## *Study and adaptation of a robotic manipulator for dynamic environments*

Jaqueline Sayuri Machida\*, Fabian Andres Lara Molina<sup>†</sup>

### RESUMO

Um manipulador robótico por si só pode lidar apenas com objetos estáticos e semelhantes uns aos outros. Uma vez acoplada uma câmera ao sistema, o *software* do robô pode planejar seus movimentos em tempo real, graças ao *feedback* fornecido pelas imagens. Isso permite que o manipulador atue em ambientes dinâmicos, onde objetos distintos estão localizados em posições e orientações diferentes, podendo até se moverem. O objetivo deste trabalho é adaptar o manipulador OpenMANIPULATOR-X para atuação em ambientes dinâmicos, através do acoplamento da câmera Realsense D435. Para isso, foi necessário familiarizar-se com o Robot Operating System (ROS), um meta-sistema operacional sobre o qual a plataforma do OpenMANIPULATOR-X opera, além de familiarizar-se com a própria plataforma. Por fim, foi realizada uma série de tarefas que serviram como testes para certificar o funcionamento do conjunto ROS, manipulador e câmera. Os resultados obtidos revelaram que a sincronização do conjunto foi obtida com sucesso, deixando o manipulador OpenMANIPULATOR-X apto para atuar em ambientes dinâmicos em trabalhos futuros.

**Palavras-chave:** manipulador robótico, câmera, visão, ambiente dinâmico.

### ABSTRACT

A robotic manipulator itself can lead only with static and equal objects. When a camera is coupled to the system, the robot software can plan its movements in real time, due to the images feedback. This allows the manipulator to act in dynamic environments, where distinct objects are in different positions and orientations, those objects may even be moving. The goal of this research is adapting the OpenMANIPULATOR-X manipulator to act in dynamic environments coupling the Realsense D435 camera. For this, it was necessary to become familiarized with the Robot Operating System (ROS), a meta-operating system where the OpenMANIPULATOR-X runs, it was also necessary to become acquainted with the platform itself. Lastly, a series of tasks was done to test the performance of the system (ROS, manipulator and camera). The results reveal that the synchronization of the system was successful, enabling the OpenMANIPULATOR-X manipulator to act in dynamic environments in future works.

**Keywords:** robotic manipulator, camera, vision, dynamic environment.

## 1 INTRODUÇÃO

Segundo Craig (2005), manipuladores robóticos são robôs formados por uma sequência de elos conectados por juntas, iniciando-se em uma base fixa. Essas juntas possibilitam que os elos se movimentem de forma rotacional (no caso de uma junta rotacional) ou translacional (junta prismática). Ao final dessa sequência é acoplado um dispositivo (o efetuator-final) escolhido de acordo com a aplicação do robô, como soldagem,

\* Engenharia de Controle e Automação, Universidade Tecnológica Federal do Paraná, Cornélio Procopio, Paraná, Brasil;  
[jaquelinemachida@alunos.utfpr.edu.br](mailto:jaquelinemachida@alunos.utfpr.edu.br)

<sup>†</sup> Universidade Tecnológica Federal do Paraná, Campus Cornélio Procopio; [fabianmolina@utfpr.edu.br](mailto:fabianmolina@utfpr.edu.br)



corte com jato d'água, pintura, montagem, aplicação de adesivos, paletização e controle de qualidade (LEWIS (2006); BARRIENTOS (1997)). Ao somar-se a quantidade de movimentos independentes possíveis de cada articulação, temos a medida de graus de liberdade do manipulador (BARRIENTOS, 1997). As principais configurações de manipuladores robóticos são: articulado, SCARA I, SCARA II, robô de coordenadas cartesianas, de coordenadas cilíndricas, de coordenadas esféricas e paralelo (LEWIS, 2006).

Um manipulador por si só pode lidar apenas com objetos estáticos e semelhantes uns aos outros. Uma vez acoplada uma câmera a ele, o robô pode planejar seus movimentos em tempo real, graças ao feedback fornecido pelas imagens. Isso permite que o manipulador atue em ambientes dinâmicos, onde objetos distintos estão localizados em posições e orientações diferentes, podendo até se moverem (JAMALUDDIN (2006); ZHANG (2020)). Logo, torna-se desejável equipar o robô com uma câmera, a fim de que ele seja capaz de lidar com tarefas complexas.

OpenMANIPULATOR-X é um projeto desenvolvido pela ROBOTIS (ROBOTIS, 2021). Ele conta com o *hardware* aberto de um braço articulado com 5 graus de liberdade; *softwares* abertos organizados em pacotes de controle, operação, simulação e algumas aplicações do manipulador no ROS. Decidiu-se trabalhar com esse manipulador graças ao custo menor quando comparado aos demais e simplicidade de manuseio.

Escolhido o manipulador, seria possível adaptá-lo para lidar com ambientes dinâmicos? O presente trabalho tem como objetivo principal tornar o manipulador OpenMANIPULATOR-X apto para realizar tarefas em ambientes dinâmicos, integrando a câmera Realsense D435 a ele (REALSENSE, 2021). Dessa forma, será necessário familiarizar-se com a plataforma OpenMANIPULATOR-X, testar o funcionamento do manipulador OpenMANIPULATOR-X real e em simulação, especificar a trajetória do efetuador-final em ambiente real e simulado e acoplar a câmera Realsense D435 ao manipulador.

## 2 MÉTODO (OU PROCEDIMENTOS OPERACIONAIS DA PESQUISA)

O manipulador OpenMANIPULATOR-X possui 5 graus de liberdade, contando com seu efetuador-final em formato de pinça, e é do tipo articulado. Apesar de ser um modelo pequeno e simples, toda a teoria aplicada ao OpenMANIPULATOR-X também é válida para braços robóticos com mais graus de liberdade.

Além de familiarizar-se com o robô real, também buscou-se dominar o robô em ambiente simulado. Para isso, foi utilizado o software de simulação Gazebo.

O OpenMANIPULATOR-X possui a interface GUI, onde é possível determinar a posição do manipulador, tanto no espaço cartesiano quanto no espaço de juntas, e o tempo que ele possui para atingi-la. A fim de testar a interface, ordenou-se que o manipulador fosse para as posições inicial e *home* e que seu efetuador-final fosse aberto e fechado. Também foram exploradas algumas coordenadas do efetuador-final no espaço angular e no espaço cartesiano.

A abordagem de operação com o GUI possui uma limitação notável: caso o operador deseje que o efetuador-final do manipulador siga uma trajetória específica, ele deve continuamente fornecer as posições e seus tempos. Almejando automatizar esse processo, foi adaptado um programa em Python de (THE CONSTRUCT, 2020). O diagrama do algoritmo é exibido na Fig. 1. O teste foi feito tanto em ambiente real quanto simulado.

Para observar a trajetória do efetuador-final em termos de coordenadas cartesianas foi utilizado o Rqt Plot, um plugin do ROS que permite visualizar dados na forma de gráficos.

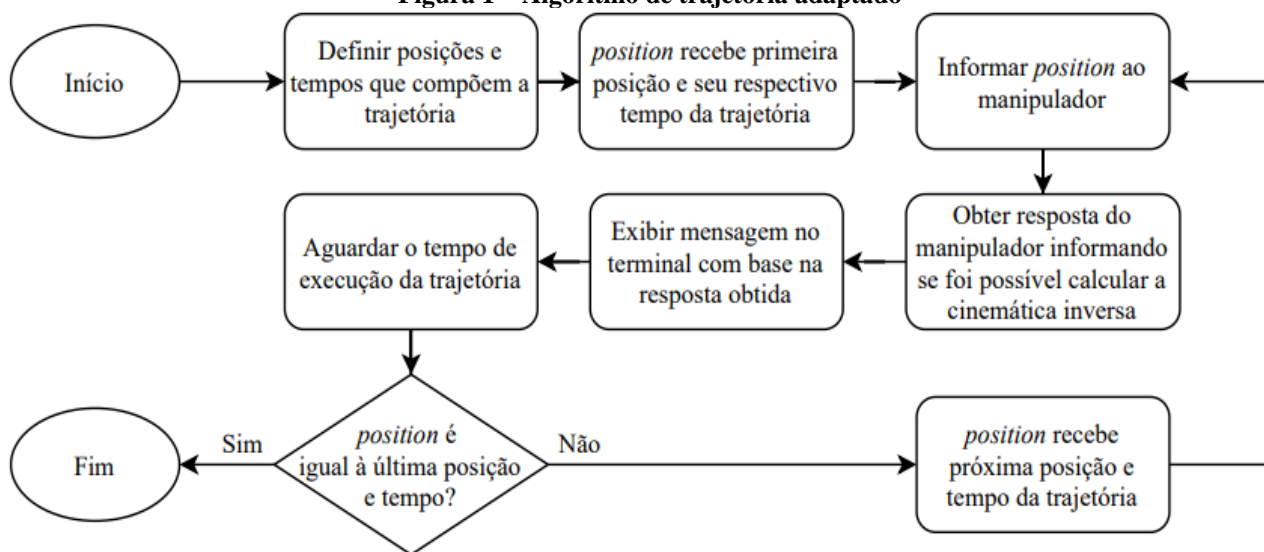
Para desempenhar tarefas, é indispensável que o algoritmo do manipulador seja capaz de localizar-se no ambiente ao qual ele está inserido, assim como localizar objetos ao seu redor. O sensor que garante essa

percepção é a câmera de profundidade Intel Realsense D435, que foi acoplada no topo do braço robótico e integrada ao ROS.

Nesse trabalho, o Rviz foi responsável por exibir as imagens da câmera, revelando o campo de visão do manipulador e os objetos observados por ele. Para isso, um cilindro foi introduzido na simulação, posicionado perto do manipulador e, depois, longe.

No teste real, optou-se por verificar o funcionamento da câmera a partir de uma aplicação *pick and place* já desenvolvida pela fabricante do OpenMANIPULATOR-X (GITHUB, 2021). O objetivo do robô nessa aplicação é localizar blocos, pegá-los e empilhá-los.

**Figura 1 – Algoritmo de trajetória adaptado**



Fonte: autoria própria (2021).

O quadro abaixo resume os testes realizados nesse trabalho:

**Quadro 1 – Testes realizados com o OpenMANIPULATOR-X**

Ambiente	Real	Simulado
Testes	Operar o manipulador pelo GUI.	
	Executar trajetória através do algoritmo adaptado.	
	Obter gráfico da trajetória cartesiana do efetuador-final.	
	Movimentar o manipulador com a câmera acoplada.	Testar aplicação <i>pick and place</i> .

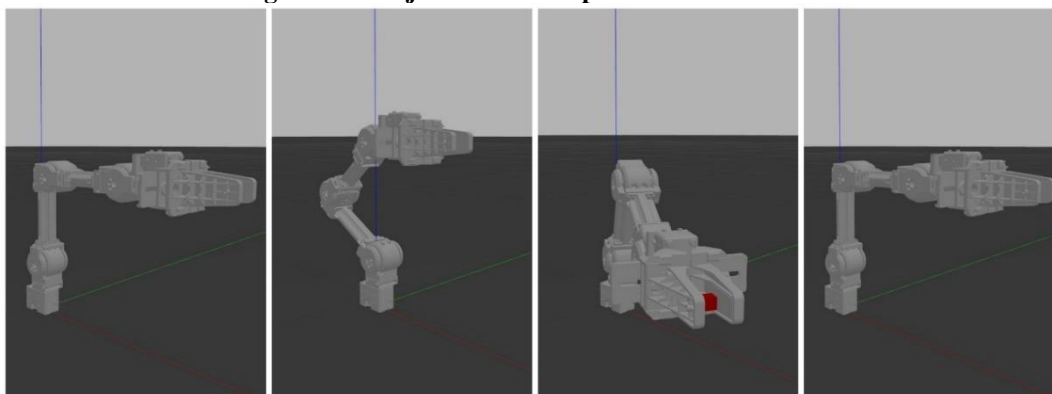
Fonte: Autoria própria (2021).

### 3 RESULTADOS

O manipulador comportou-se de acordo com as instruções durante a operação com o GUI, em ambos os ambientes real e simulado. Tanto em ambiente real quanto no Gazebo, o manipulador não teve dificuldade em atingir as coordenadas do efetuador-final no espaço de juntas. O mesmo não ocorreu com o espaço cartesiano, pois era necessário informar os pontos exatos que o efetuador-final é capaz de alcançar, para que fosse possível resolver a cinemática inversa.

Os robôs real e simulado foram capazes de seguir todas as trajetórias submetidas no algoritmo, sendo uma delas de 4 posições ilustrada na Fig. 2. 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.

**Figura 2 – Trajetória do manipulador simulado**



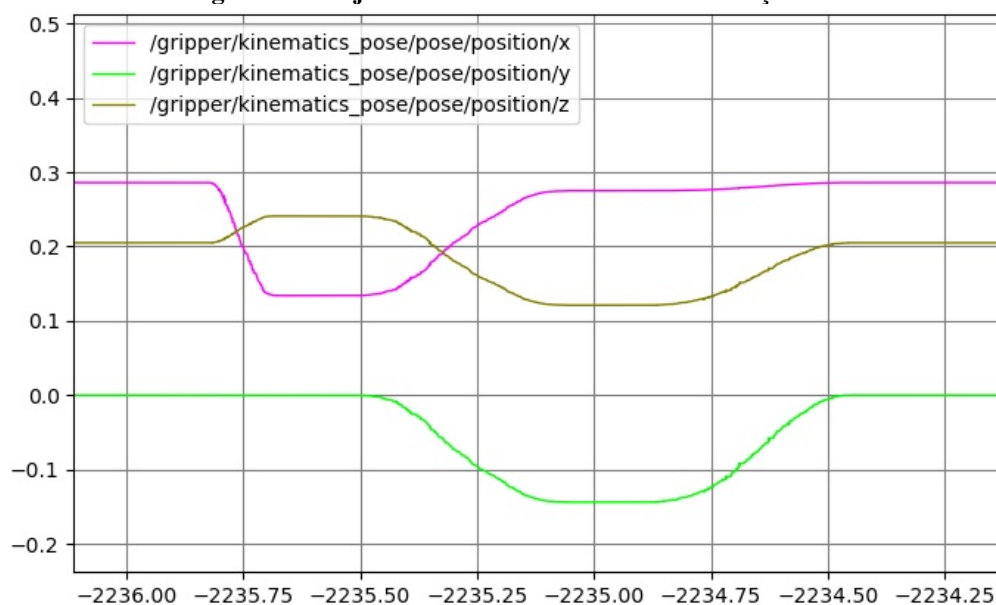
Fonte: Autoria própria (2021).

Durante a execução do algoritmo, o gráfico das coordenadas x, y e z pôde ser gerado pelo Rviz sem problemas para os robôs real e simulado (Fig. 3).

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.

Ao executar a tarefa *pick and place* em ambiente real (vide Fig. 4), foi possível atestar o funcionamento da câmera, pois a posição e a orientação dos blocos foram reconhecidas, permitindo que o robô se dirigisse até eles e completasse sua tarefa.

**Figura 3 – Trajetória do efetuador-final na simulação**



Fonte: Autoria própria (2021).



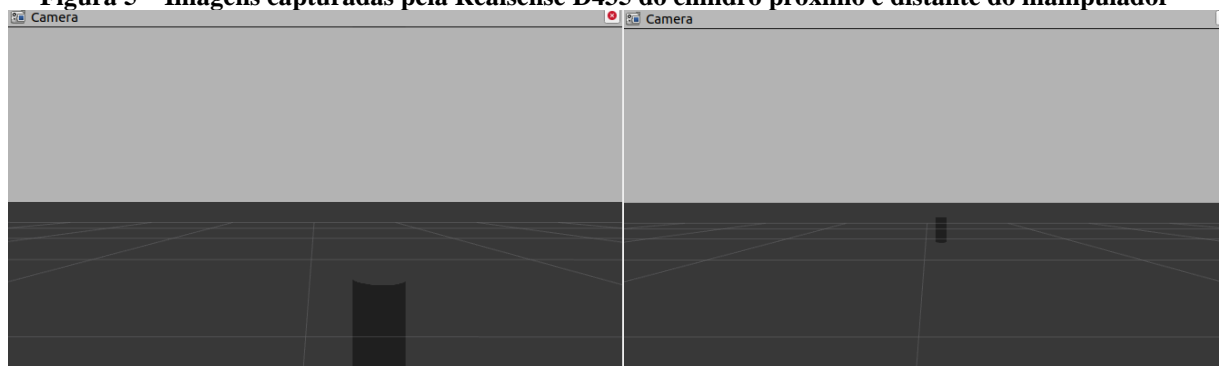
**Figura 4 – Câmera Realsense acoplada à garra do manipulador**



**Fonte: Autoria própria (2021).**

A seguir, é descrito o teste realizado em simulação. A Fig. 5 mostra o Rviz exibindo as imagens publicadas pela Realsense D435, com o cilindro localizado próximo e longe do manipulador. Através do Rviz comprovou-se o funcionamento da câmera na simulação em conjunto com o ROS, afinal foi possível ver as imagens com o objeto em diferentes profundidades.

**Figura 5 – Imagens capturadas pela Realsense D435 do cilindro próximo e distante do manipulador**



**Fonte: Autoria própria (2021).**

## 4 CONCLUSÃO

Esse trabalho teve como objetivo principal tornar o manipulador OpenMANIPULATOR-X apto para atuar em ambientes dinâmicos, integrando a câmera Realsense D435 a ele. Para cumprir esse objetivo, foi necessário familiarizar-se com a plataforma OpenMANIPULATOR-X, testar o funcionamento do manipulador e automatizar sua trajetória, tanto em ambiente real quanto em simulação. Por fim, a câmera de profundidade Realsense D435 foi acoplada ao manipulador, tanto real quanto simulado, e integrada ao ROS, permitindo a visualização de suas imagens.

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





## AGRADECIMENTOS

Os autores agradecem à Fundação Araucária pelas bolsas concedidas.

## REFERÊNCIAS

- CRAIG, John J. **Introduction to Robotics: Mechanics and Control**. 3. ed. Upper Saddle River: Prentice Hall, 2005.
- L. LEWIS, Frank; M. DAWSON, Darren; T. ABDALLAH, Chaouki. **Robot Manipulator Control: Theory and Practice**. 2. ed. New York: Marcel Dekker, 2006.
- BARRIENTOS, Antonio; F. PEÑÍN, Luís; BALAGUER, Carlos; ARACIL, Rafael. **Fundamentos de Robótica**. 2. ed. Madrid: Concepción Fernández Madrid, 1997.
- ROBOTIS E-MANUAL. **OpenMANIPULATOR-X**. Disponível em: [https://emanual.robotis.com/docs/en/platform/openmanipulator\\_x/overview/](https://emanual.robotis.com/docs/en/platform/openmanipulator_x/overview/).
- JAMALUDDIN, M. H.; SAID, M. A.; SULAIMAN, Marizan.; HORNG, C. S. Vision Guided Manipulator for Optimal Dynamic Performance. **2006 4th Student Conference on Research and Development**, 27-28 jun. 2006, Malaysia. Disponível em: <https://ieeexplore.ieee.org/document/4339328>.
- ZHANG, Hui; JIN, Hongzhe; GE, Mingda; ZHU, Yanhe; ZHAO, Jie. Real-Time Joint Angular-Acceleration Planning for Vision-Based Kinematically Redundant Manipulator in Dynamic Environment. **IEEE Access**, v. 8, p. 110796-110809, jun. 2020. Disponível em: <https://ieeexplore.ieee.org/document/9113243>.
- INTEL REALSENSE. **Depth Camera D435**. Disponível em: <https://www.intelrealsense.com/depth-camera-d435/>.
- THE CONSTRUCT. **Programming a Service client for ROBOTIS Open Manipulator**. Disponível em: <https://www.theconstructsim.com/programming-a-service-client-for-robotis-open-manipulator/>.
- GITHUB. **Open\_manipulator\_find\_object\_2d**. Disponível em: [https://github.com/minwoominwoominwoo7/open\\_manipulator\\_find\\_object\\_2d](https://github.com/minwoominwoominwoo7/open_manipulator_find_object_2d).
- ROS Wiki. Rviz. Disponível em: <http://wiki.ros.org/rviz>.