VII Congresso Nacional de Inovação e Tecnologia - INOVA 2024
08 de novembro de 2024
São Bento do Sul / Santa Catarina - Brasil
ISSN 2526-3145

Projeto e Controle de Manipulador Robótico de Baixo Custo baseado em ROS

Design and Control of Low-Cost Robotic Manipulator based on ROS

Elton João Wassoaski Junior*, Gisele Fernanda Koman Fucilini†, Dieisson Martinelli‡, Vivian Cremer Kalempa§

RESUMO

Este artigo demonstra o desenvolvimento de um manipulador robótico de baixo custo, destinado à comunidade acadêmica e projetos com recursos limitados. O objetivo é desenvolver um manipulador capaz de interagir com o ambiente, realizando tarefas automatizadas. A ênfase está em utilizar o *Robot Operating System* (ROS), como plataforma central para integração e controle eficiente, aproveitando os recursos disponíveis nas comunidades *open source*. Ao destacar a viabilidade da aplicação de técnicas avançadas de robótica, sem depender de orçamentos substanciais, este trabalho evidencia que a excelência em robótica não deve ser exclusiva de grandes orçamentos, mas sim da busca por soluções inteligentes e acessíveis, inspirando o desenvolvimento de novos projetos de robótica, impulsionando assim, avanços significativos na área.

PALAVRAS-CHAVE: Manipulador robótico; Robot Operating System (ROS); Baixo Custo.

ABSTRACT

This article demonstrates the development of a low-cost robotic manipulator, aimed at the academic community and projects with limited resources. The goal is to create a manipulator capable of interacting with the environment, performing automated tasks. The emphasis is on utilizing the Robot Operating System (ROS) as the central platform for efficient integration and control, leveraging the resources available in open-source communities. By highlighting the feasibility of applying advanced robotics techniques without depending on substantial budgets, this work underscores that excellence in robotics should not be exclusive to large budgets. Instead, it advocates for intelligent and accessible solutions, inspiring the development of new robotics projects, thus propelling significant advances in the field.

KEYWORDS: Robotic Manipulator; Robot Operating System (ROS); Low Cost.

1 INTRODUÇÃO

A robótica tem experimentado um notável crescimento no mercado, marcado pela integração de máquinas, algoritmos e dispositivos nos cenários reais e virtuais (Moraes, 2021). Esse avanço é impulsionado pela demanda crescente na indústria, abrangendo desde questões relacionadas ao alto volume de produção e eficiência do fluxo produtivo, até o controle e a garantia da qualidade dos produtos fornecidos (Brasil, 2022). Diante desse cenário, a robótica surge como um facilitador essencial para atender às exigências do mercado, operando com alta precisão e superando desafios associados a tarefas repetitivas, características que a distinguem positivamente em comparação com a mão de obra humana. Ao longo dos anos, a aplicação da robótica tem evoluído significativamente, destacando-se a inclusão de novas tecnologias que proporcionam maior precisão e autonomia nas tarefas

* Udesc, São Bento do Sul, SC, Brasil. eltonjoaow@gmail.com.
† Udesc, São Bento do Sul, SC, Brasil. gisele.fucilini@edu.udesc.br .
‡ Udesc, São Bento do Sul, SC, Brasil. dieisson.martinelli@udesc.br.
§ Udesc, São Bento do Sul, SC, Brasil. vivian.kalempa@udesc.br.

realizadas. Essas inovações resultam de extensas pesquisas conduzidas por importantes universidades e instituições de pesquisa globais, visando desenvolver soluções para as principais demandas do mercado (Santoni; Lucato, 2021).

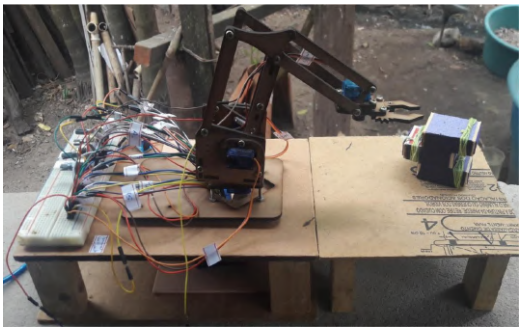
Nesse sentido, este trabalho propõe a implementação do ROS, uma plataforma de *software* de código aberto amplamente utilizada no campo da robótica (Robotics, 2023b), em um modelo de braço robótico de baixo custo, acessível no mercado (Ômega, 2023). Isso possibilitará a aplicação de técnicas avançadas de controle de movimento, utilizando os recursos robustos do ROS, para potencializar as capacidades do braço robótico em desempenhar tarefas e agregando um nível superior de funcionalidade ao sistema. Dessa forma, visa-se promover a utilidade do braço robótico em diferentes contextos de maneira prática, especialmente para a acessibilidade de comunidades acadêmicas e projetos com recursos limitados.

O artigo está estruturado da seguinte forma: na Seção 2 é apresentada uma revisão bibliográfica de trabalhos relacionados ao objetivo deste artigo. Na Seção 3, é apresentada a abordagem proposta para o desenvolvimento do manipulador robótico de baixo custo, abrangendo a escolha e integração dos componentes utilizados em sua construção, bem como a utilização do ROS e seus recursos como uma plataforma fundamental para o desenvolvimento lógico deste trabalho. Na Seção 4, são descritos os experimentos e resultados dos testes realizados para avaliar o comportamento do manipulador, tanto em ambientes virtuais como em ambientes reais. Finalmente, na Seção 5, as conclusões obtidas deste projeto são apontadas, destacando as contribuições do trabalho, suas limitações e possíveis direções para pesquisas futuras.

2 TRABALHOS RELACIONADOS

Ao realizar uma pesquisa bibliográfica, é possível encontrar diversos trabalhos relacionados à prototipação robótica de diferentes contextos, bem como a utilização de diversos recursos para demostrar distintas tarefas automatizadas. Nesse sentido, o trabalho desenvolvido por (Ferreira da Silva, 2016) apresenta um braço mecânico multipropósito (Figura 1), que utiliza nos seus componentes estruturais peças de MDF pré-moldadas, montado com 4 servomotores, proporcionando ao protótipo 4 graus de liberdade para a execução dos seus movimentos com mais precisão. A autora enfatiza diversos pontos de melhoria para o projeto, como implantação de sensores ao protótipo e a possibilidade de controle a partir de dispositivos móveis.

Figura 1 – Braço mecânico multipropósito

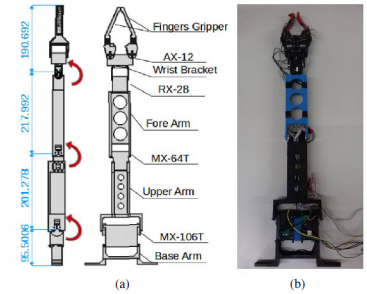


Fonte: Ferreira da Silva (2016).

Um trabalho de grande proveito para o estudo deste artigo é o projeto desenvolvido por (Hernandez-Mendez *et al.*, 2017), apresentado na Figura 2, o qual projeta e implementa um braço robótico utilizando ROS e MoveIt, que pode manipular e agarrar objetos para transportá-los de um local para outro.

A metodologia baseada em ROS é adotada para conseguir a manipulação do objeto com o braço e sua visualização. Quanto aos trabalhos futuros, os autores sugerem melhorar a precisão do planejamento e controle do movimento do braço, bem como explorar o uso de técnicas de aprendizado de máquina para melhorar a capacidade do braço de agarrar e manipular objetos.

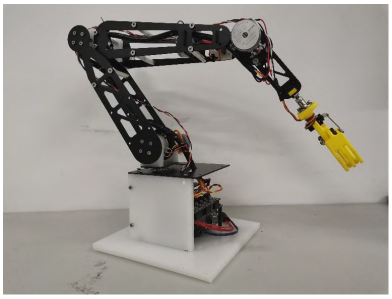
Figura 2 – a) Projeto mecânico do manipulador b) Braço robótico montado.



Fonte: Hernandez-Mendez *et al.* (2017).

Com foco nos trabalhos que fazem o uso do ROS para o desenvolvimento de protótipos, o projeto de (Zhu *et al.*, 2023) apresentam como proposta um novo esquema de projeto para um braço robótico baseado em ROS, que seja diferente da maioria dos braços robóticos do mercado, que utilizam materiais puros de impressão 3D como principal suporte estrutural (Figura 3). Os resultados experimentais de (Zhu *et al.*, 2023) mostram que o projeto proposto do sistema de braço robótico é bem-sucedido e possui boa escalabilidade. Os autores sugerem que trabalhos futuros se concentrem na pesquisa de servo visual e no aprimoramento da atual solução de cinemática inversa do braço robótico.

Figura 3 – Braço mecânico.



Fonte: Zhu *et al.* (2023).

Ao discorrer a pesquisa bibliográfica é perceptível ver as diversas aplicações da automatização de tarefas, como também os autores deixam explícito ao final das suas obras, as oportunidades e a gama de melhorias e incrementos que podem dar ainda mais robustez à robótica e podem ser aplicadas a diferentes cenários. Visto isso, este trabalho inova ao introduzir um manipulador robótico de baixo custo com a ênfase na utilização do

ROS, capitalizando os inúmeros recursos disponíveis na comunidade *open source*. Ao adotar um protótipo de baixo custo e recursos da comunidade aberta, o projeto promove a maior acessibilidade na área de robótica e aproveita a robustez e eficiência do ROS para aprimorar as capacidades do manipulador.

3 ABORDAGEM PROPOSTA

Nesta seção, é apresentado em detalhes o desenvolvimento e implementação do manipulador robótico de baixo custo.

3.1 BRAÇO ROBÓTICO E SEUS COMPONENTES

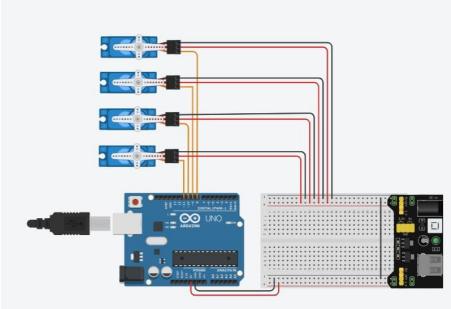
O modelo proposto para este trabalho é um braço articulado construído para dispor de 4 graus de liberdade, feito e uma impressora 3D, com os encaixes dos micro servo motores já planejados para a movimentação da estrutura, conforme Figura 4. Este modelo foi adquirido em *site* especializado na distribuição de componentes para projetos de eletrônica (Ômega, 2023).

Figura 4 – Modelo 3D Braço Robótico.

Fonte: Gabriel (2023).

Para montar o circuito elétrico do protótipo, conforme apresentado na Figura 5, é utilizada uma placa de prototipagem Arduino Uno, quatro servomotores e uma fonte ajustável *protoboard*, que pode ser conectada diretamente a uma fonte de alimentação DC para converter a tensão de saída em 5V. As conexões foram feitas utilizando os *jumpers* para alimentação dos servomotores e conexão às portas digitais do Arduino para controle dos servos. No circuito elétrico da Figura 5, é possível visualizar a integração do Arduino ao sistema, componente crucial para o desenvolvimento do protótipo.

Figura 5 – Circuito elétrico do protótipo.

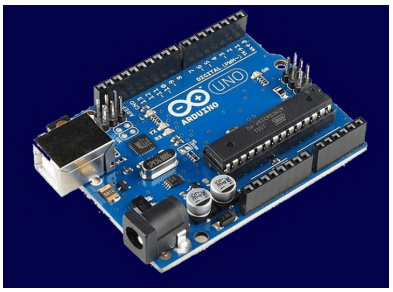


Fonte: Elaborada pelos autores (2023).

Existem algumas variantes das placas Arduino, cada uma com seu próprio processador e características únicas. Para o projeto desse artigo é utilizado o Arduino Uno, que pode

ser observado na Figura 6, uma escolha popular para projetos de robótica, devido à sua facilidade de uso, versatilidade, custo acessível e uma comunidade ativa de suporte. Ele é adequado para integrar os atuadores do projeto e a sua programação é de fácil compreensão, podendo ser usado em conjunto com o ROS para projetos de robótica mais complexos.

Figura 6 – Placa Arduino Uno.



Fonte: Gabriel (2023).

Para integração dos atuadores do projeto, conforme o modelo do braço adquirido, é utilizado os servomotores, dispositivos eletromecânicos que convertem um sinal de controle em movimento rotacional preciso. Para a aplicação dos servos no protótipo, utilizou-se os micro servos, uma versão compacta e de menor tamanho de um servomotor, que geralmente são utilizados em projetos de pequena escala. Os micro servos possuem as mesmas características básicas de um servomotor convencional, mas em uma escala menor, como pode ser visto na Figura 7.

Figura 7 – Micro Servo Motor.

Fonte: Araujo (2018).

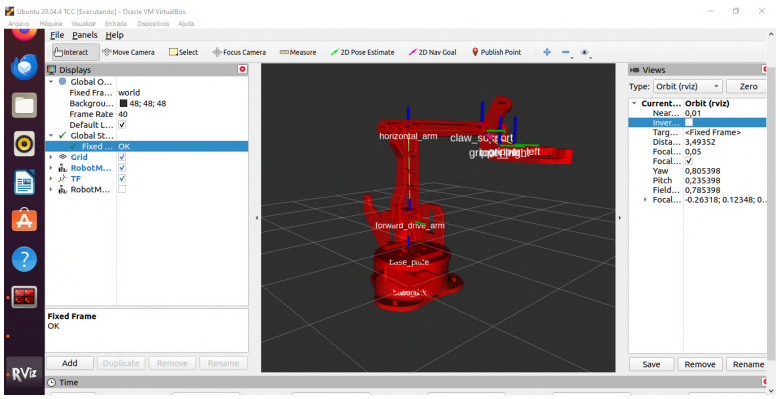
3.2 REPRESENTAÇÃO VIRTUAL - *DIGITAL TWIN*

Para criar a representação virtual da estrutura do robô no ROS, baseado no trabalho de Brandi (2023), é utilizado o formato denominado URDF, sendo essencialmente, um modelo 3D com informações sobre juntas, motores, massa, dentre outros aspectos. O modelo do robô é um arquivo XML, no qual cada componente do robô é definido com uma *tag* XML. O objetivo deste arquivo XML é definir o esqueleto do robô, unindo os vários elos do robô e, eventualmente, atribuindo algumas propriedades físicas e cinemáticas a esses componentes. Nesse contexto, é criado um arquivo URDF aprimorado com extensões Xacro, visando descrever a estrutura física do robô destinado à simulação no ambiente Gazebo. O URDF, uma linguagem de marcação XML amplamente adotada, é complementado pelo Xacro, uma extensão que possibilita uma descrição mais concisa e modular. Com a extensão do Xacro, incorpora-se o uso de propriedades, como PI, esforço e velocidade, proporcionando uma abordagem organizada e facilmente ajustável (Robotics, 2022). Além disso, a implementação de macros contribui para a definição eficiente de padrões de inércia e transmissão, promovendo uma estrutura de código mais compreensível (Brandi, 2023).

Utilizando o RViz, uma ferramenta de visualização tridimensional projetada para integração com o ROS (Robotics, 2018), o robô é decomposto em vários elos, requerendo a presença dos arquivos STL que contêm as informações da malha 3D de cada elo do robô. Cada elo representa diferentes partes físicas do robô, como a base, braços e garras, caracterizado por propriedades específicas, incluindo inércia, visualização e colisão. Isso proporciona uma representação detalhada e abrangente do modelo físico.

Na Figura 8 é possível visualizar o modelo URDF do protótipo. Na interface da ferramenta RViz é gerado o modelo de visualização tridimensional do robô, onde os elementos em vermelho são os elos de cada componente do braço e as legendas em branco fornecem a descrição de cada um desses elos. Em meio a estrutura do robô, é possível visualizar os elementos em azul e verde, que tratam-se das juntas definidas para simular as partes móveis do robô.

Figura 8 – Modelo URDF na ferramenta RViz.



Fonte: Elaborada pelos autores (2023).

3.3 CONTROLES

Com o *Digital Twin* do robô concluído, é dado sequência ao desenvolvimento das funcionalidades, para que seja possível visualizar o comportamento da estrutura em um ambiente simulado. O ponto de partida será a funcionalidade mais básica que todo robô deve implementar, que consiste na atuação dos motores. Mesmo sendo a mais básica, essa funcionalidade é essencial para todos os outros módulos e nós que compõem o *software* de um robô e atuam no robô com uma certa lógica. Uma vez efetuada essa funcionalidade básica, é possível enviar comandos para os motores que atuam na junta e obter um movimento do robô como consequência do comando enviado.

O controle é implementado usando a biblioteca ROS *Control*, que permite o desenvolvimento do seu próprio controlador ou a reutilização de alguns controladores que já estão implementados (Robotics, 2023c). O manuseio dos controladores ROS é composto por dois componentes principais: o gerenciador de controladores e a interface de *hardware*. A interface de *hardware* é responsável por interagir com o *hardware* do robô, seja ele um robô real ou apenas um simulado no Gazebo. A vantagem deste componente é que, mesmo sendo específico para o *hardware*, ele expõe a outros módulos uma interface padrão, então qualquer outro nó ou aplicativo que queira enviar comandos para o motor pode fazer isso, independentemente do tipo de motores, dos seus protocolos de comunicação e pode usar a interface padrão oferecida pela interface de *hardware*.

O segundo componente do sistema de controle no ROS é chamado de Gerenciador de Controladores. Esse componente permite desenvolver a lógica do controlador que recebe a entrada do usuário e a saída do motor. Assim, ele conhece o alvo e o *status* atual do motor e garante que a saída se torne igual à entrada e, de preferência, no menor período possível.

3.4 MOVEIT

Um dos grandes pontos para desenvolver projetos de manipuladores é a resolução do problema de cinemática, pois pode haver várias configurações de juntas que levam a garra à mesma posição no espaço. A cinemática inversa pode ter soluções alternativas e, em alguns casos, até soluções infinitas. Diferentemente da cinemática direta, a cinemática inversa não é resolvida com equações em forma fechada, mas utiliza, frequentemente, técnicas numéricas ou otimizadores (Silva, 2019).

Ao resolver o problema de cinemática inversa, é possível mover diretamente a garra no espaço, sem a necessidade de se preocupar com as variações correspondentes nos ângulos das juntas. Mesmo que esses valores de ângulo sejam necessários para acionar os motores, a cinemática inversa simplifica o processo, relacionando a posição da garra no espaço com os valores dos ângulos das juntas (Brandi, 2023).

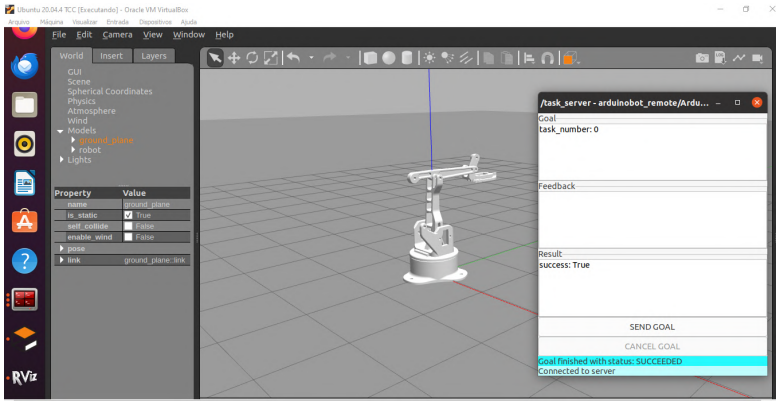
O MoveIt é uma plataforma de planejamento de movimento amplamente utilizada em robótica, projetada para facilitar o controle de manipuladores robóticos. O MoveIt simplifica tarefas complexas, como o planejamento de trajetórias, tornando-as acessíveis mesmo para robôs com diferentes graus de liberdade (Robotics, 2023a).

4 EXPERIMENTOS E RESULTADOS

Nesta seção são apresentadas as experimentações realizadas a fim de explorar as oportunidades oferecidas na implementação dos recursos do ROS. Para desenvolver aplicações que utilizam a API do MoveIt é introduzida uma nova ferramenta chamada ROS *Action*. Uma ação que implementa um novo protocolo de comunicação entre os nós. A comunicação por ações é semelhante à comunicação por serviços, mas com algumas peculiaridades. Ela ainda envolve um nó cliente e um nó servidor. A principal diferença é que, ao contrário dos serviços, a comunicação por ações não bloqueia o nó cliente. Isso permite que o nó cliente continue executando as suas funcionalidades, mesmo enquanto aguarda a conclusão da operação do servidor (Saito, 2018). Uma vez que a ação é concluída, independentemente de ter sido bem-sucedida ou não, o servidor de ação envia uma mensagem de resultado para o cliente de ação. Além disso, o cliente de ação pode enviar uma mensagem de cancelamento para o servidor durante a execução da ação, fazendo com que o servidor tente interromper a ação.

Para o projeto do manipulador, um servidor de tarefas é criado como interface para a API MoveIt, permitindo ao robô executar várias tarefas baseadas nos objetivos recebidos. A lógica do servidor é desenvolvida em Python, usando as bibliotecas *ActionLib* e *MoveIt Commander*. Nos terminais do Ubuntu, são iniciados os arquivos de simulação Gazebo, o controlador do robô, o núcleo do MoveIt e o nó do servidor de tarefas, com o cliente axcliente.py da *ActionLib* para planejar e executar ações. A interface gráfica da *Action Tools Library* permite enviar novas tarefas para o robô. A Figura 9 mostra a simulação no Gazebo.

Figura 9 – Simulação do projeto no Gazebo.



Fonte: Elaborada pelos autores (2023).

4.1 EXPERIMENTO ROBÔ REAL

Depois de bem-sucedida a simulação do protótipo nos ambientes virtuais, é dado foco a simulação com o robô real, iniciando com código de programação do Arduino Uno, utilizando a linguagem C++, pelo aplicativo Arduino IDE. Este projetado para controlar os quatro servomotores do manipulador conectados ao Arduino e integrá-los ao sistema ROS, permitindo o controle independente de cada servomotor.

A inicialização dos servomotores ocorre pela biblioteca Servo, definindo as posições iniciais desejadas para os mesmos. Seguidamente, é definido os limites de alcance de movimento, com um mínimo de 0 grau e um máximo de 180 graus, evitando o controle dos motores fora dos seus limites físicos. O *script* conta também com uma função para controlar suavemente o movimento dos servomotores em direção às posições desejadas. Essa função calcula a diferença entre a posição atual e a posição alvo de cada servo, permitindo que os motores se movam gradualmente em incrementos de um grau de cada vez, evitando movimentos bruscos e imprecisos. Após carregada a programação na placa Arduino, é feita a atualização do arquivo de controle do manipulador, possibilitando que o ROS e o Arduino possam trocar mensagens, pela comunicação serial, usando o protocolo rosserial.

Por fim, como na simulação virtual, repete-se o processo de execução do controlador do robô, o núcleo do MoveIt e o nó de comunicação com o servidor para fornecer as tarefas de movimento para o manipulador.

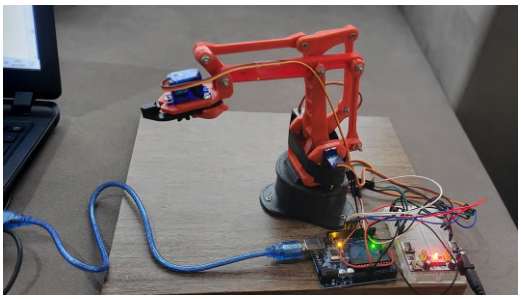
4.2 RESULTADOS DO EXPERIMENTO

Na Figura 10 é possível visualizar o resultado após a montagem da parte elétrica e mecânica do manipulador robótico e os seus componentes instalados.

O manipulador opera utilizando o ROS e o MoveIt sendo inicializado e configurado para o modelo cinemático do robô. Ele recebe comandos de movimento, planeja trajetórias suaves considerando restrições cinemáticas e obstáculos, bem como executa as ações desejadas em tempo real. A interface de controle permite interação com operadores humanos, pelo servidor de tarefas, enquanto o *feedback* contínuo e a integração com os atuadores contribuem para uma operação coordenada e eficiente. O ciclo de operação inclui desde o recebimento de objetivos até a conclusão da tarefa, podendo ser repetido conforme necessário.

Os resultados foram obtidos por três tarefas específicas definidas, cada uma associada

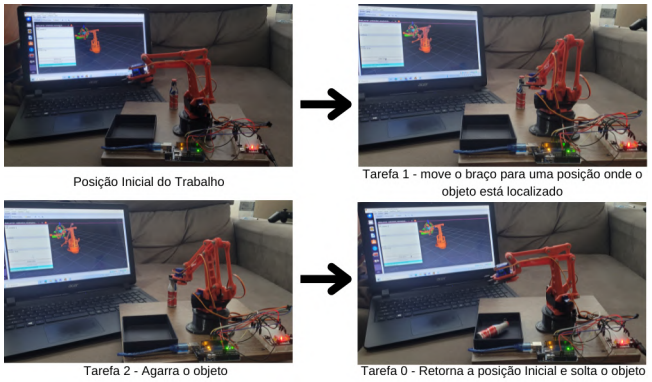
Figura 10 – Protótipo Final.



Fonte: Elaborada pelos autores (2023).

a um número de tarefa. Primeiramente carrega-se o Arduino para o manipulador assumir a posição inicial definida. Depois executa-se a primeira tarefa utilizando a interface de comunicação da biblioteca *ActionLib*. Ao executar a primeira tarefa o braço se movimenta para a posição na qual o objeto está localizado. Na execução da segunda tarefa, o braço se movimenta para apanhar o objeto que está situado à sua frente. Por fim, é executada a terceira tarefa que transporta o objeto para a posição inicial do trabalho e o solta em um recipiente localizado abaixo da garra, conforme a Figura 11.

Figura 11 – Sequência de Imagens da funcionalidade do Protótipo Final.



Fonte: Elaborada pelos autores (2023).

Se a tarefa fornecida não corresponder a nenhuma das três tarefas predefinidas, é registrado um erro. As metas são enviadas ao MoveIt API para planejamento e execução de trajetórias e o sucesso ou a falha da execução é comunicado de volta ao cliente. O código também lida com solicitações predefinidas e fornece *feedback* durante a execução.

O MoveIt desempenha um papel crucial na boa execução do projeto, ao fornecer uma interface eficiente para o planejamento e execução de movimentos para a manipulação do robô, simplificando a complexidade do controle de movimento, permitindo que o robô execute as tarefas específicas de maneira suave e precisa.

A realização de simulações no Gazebo e no RViz pode apresentar desafios de desempenho quando o *hardware* disponível não atende aos requisitos robustos necessários. Por se tratarem de programas pesados, as limitações de desempenho podem resultar em atrasos significativos, tornando a simulação menos precisa e impactando a capacidade de visualização em tempo real no RViz. Além disso, a complexidade de modelos detalhados e interações dinâmicas pode sobrecarregar sistemas menos potentes, comprometendo a

fidelidade das simulações.

Outro ponto crucial, é a montagem cuidadosa do braço robótico, que devido à natureza das peças produzidas em impressora 3D que, embora sejam inovadoras, podem ser mais suscetíveis à fragilidade em comparação com peças tradicionalmente fabricadas. A precisão na montagem se torna essencial, pois qualquer desalinhamento ou falta de estabilidade pode comprometer a integridade estrutural e, consequentemente, afetar os movimentos do braço robótico. A atenção aos detalhes durante o processo de montagem é vital para garantir a funcionalidade adequada das articulações e a resistência necessária para lidar com as demandas do uso.

5 CONSIDERAÇÕES FINAIS

Neste projeto é apresentada a implementação bem-sucedida de um protótipo de robô manipulador de baixo custo, demonstrando como é possível alcançar resultados significativos com recursos acessíveis. Ao abordar diversos conceitos, desde o desenvolvimento do modelo URDF até a integração com a plataforma do MoveIt o resultado do trabalho reflete a viabilidade e a eficácia de um protótipo acessível para fins acadêmicos e de pesquisa. A utilização do ROS, como estrutura principal, permite uma abordagem modular e colaborativa, facilitando o desenvolvimento e a integração de diferentes funcionalidades. O controle de movimento, a cinemática inversa e a interação cliente e servidor implementados de forma eficiente, demonstraram que soluções de baixo custo podem ser altamente eficazes. As limitações deste trabalho foram principalmente relacionadas ao peso dos arquivos de simulação, o que impactou negativamente no desempenho das análises. A complexidade e o tamanho desses arquivos tornaram mais lenta a execução das simulações, prejudicando a eficiência dos testes nos ambientes virtuais. Como proposta de melhoria futura, sugere-se a integração de sensores para aperfeiçoar a capacidade do manipulador em perceber e interagir com objetos no ambiente. Isso não apenas ampliaria as habilidades do robô, como contribuiria para um desempenho mais robusto e versátil em diversas aplicações.

A contribuição significativa deste projeto, ao tornar a robótica ainda mais acessível, abre portas para estudantes, entusiastas e comunidades com recursos limitados, explorarem e experimentarem tecnologias avançadas. A maneira como é possível integrar com sucesso um protótipo de baixo custo com o ROS destaca o potencial dessas tecnologias em tornar a interação humano-robô uma experiência intuitiva e acessível para todos.

Conflito de interesse

Não há conflito de interesse.

REFERÊNCIAS

ARAUJO, Pedro Filipe Costa. Protótipo de braço robótico capaz de capturar objetos fazendo uso de processamento de imagens e redes neurais. **RE3C-Revista Eletrônica Científica de Ciência da Computação**, v. 13, n. 1, 2018.

BRANDI, Antonio. **Robotics-and-ROS-Learn-by-Doing-Manipulators**. [*S. l.: s. n.*], 2023. Disponível em: https://github.com/AntoBrandi/Robotics-and-ROS-Learn-by-Doing-Manipulators.git.

BRASIL, Universal Robots. **Tipos de robôs industriais e suas aplicações**. [*S. l.: s. n.*], 2022. Acesso em: 05 nov. 2023. Disponível em: https://www.universal-robots.com/br/blog/.

FERREIRA DA SILVA, Jaqueline. **Braço mecânico multipropósito controlado pela plataforma de prototipagem Arduino**. 2016. F. 60. Diss. (Mestrado) – Faculdade de Engenharia Elétrica, Universidade Federal do Pará - Tucuruí - Pará, Pará.

GABRIEL, Rangel. **E-book Robótica para iniciantes**. [*S. l.: s. n.*], 2023. Acesso em: 5 nov. 2023. Disponível em: https://www.eletronicaomega.com/.

HERNANDEZ-MENDEZ, Sergio *et al.* Design and implementation of a robotic arm using ROS and MoveIt! *In*: PROCEEDINGS of the 2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC). Ixtapa, Mexico: IEEE, 2017. P. 1–6. DOI: 10.1109/ROPEC.2017.8261666.

MORAES, Rodrigo Bombonati de Souza. **Indústria 4.0: Impactos sociais e profissionais**. 1. ed. São Paulo: Blusher, 2021.

ÔMEGA, Eletrônica. **Kit Braço Robótico 3D**. [*S. l.: s. n.*], 2023. Acesso em: 10 out. 2023. Disponível em: https://www.eletronicaomega.com/.

ROBOTICS, Open. **MoveIt**. [*S. l.: s. n.*], 2023a. Acesso em: 05 nov. 2023. Disponível em: https://moveit.ros.org/.

ROBOTICS, Open. **ROS - Robot Operating System**. [*S. l.: s. n.*], 2023b. Acesso em: 05 nov. 2023. Disponível em: https://www.ros.org/.

ROBOTICS, Open. **ros control**. [*S. l.: s. n.*], 2023c. Acesso em: 05 nov. 2023. Disponível em: https://www.ros.org/ros_control.

ROBOTICS, Open. **rviz**. [*S. l.: s. n.*], 2018. Acesso em: 10 out. 2023. Disponível em: http://wiki.ros.org/rviz.

ROBOTICS, Open. **xacro**. [*S. l.: s. n.*], 2022. Acesso em: 05 nov. 2023. Disponível em: http://wiki.ros.org/xacro.

SAITO, Isaac. **actionlib**. [*S. l.: s. n.*], 2018. Acesso em: 28 out. 2023. Disponível em: http://wiki.ros.org/actionlib.

SANTONI, Fabiano; LUCATO, André Vicente Ricco. Robótica colaborativa: A utilização de robôs nos processos produtivos. **RECIMA21-Revista Científica Multidisciplinar**, v. 1, n. 1, e210914–e210914, 2021.

SILVA, Fernando Rafael Arnhold da. **Projeto e construção de um braço robótico SCARA com controle da cinemática**. 2019. Diss. (Mestrado) – Universidade Federal de Pernambuco.

ZHU, Mingyi *et al.* The Design of Robotic Arm Based on ROS. *In*: PROCEEDINGS of the 2023 3rd International Conference on Computer, Control and Robotics (ICCCR). Shanghai, China: IEEE, 2023. P. 203–207. ISBN 978-1-6654-9212-6, 978-1-6654-9211-9, 978-1-6654-9213-3. DOI: 10.1109/ICCCR56747.2023.10194215.