

# Desenvolvimento de um robô cartesiano *pick-and-place* com 4 graus de liberdade

*Development of a Cartesian Pick-and-Place Robot with 4 Degrees of Freedom*

**Cláudio Lourenço Moreira<sup>1</sup>**  
 [orcid.org/0009-0006-9221-1041](https://orcid.org/0009-0006-9221-1041)

**João Victor Zanoni<sup>2</sup>**  
 [orcid.org/0009-0003-1483-7565](https://orcid.org/0009-0003-1483-7565)

**Lauro Weingartner Neto<sup>3</sup>**  
 [orcid.org/0000-0003-0584-6972](https://orcid.org/0000-0003-0584-6972)

**Marcos Vinicius Matsuo<sup>4</sup>**  
 [orcid.org/0000-0001-6766-3261](https://orcid.org/0000-0001-6766-3261)

**Ebrahim Samer El Youssef<sup>5</sup>**  
 [orcid.org/0000-0001-6722-3899](https://orcid.org/0000-0001-6722-3899)

**Carlos Roberto Moratelli<sup>6</sup>**  
 [orcid.org/0000-0002-0739-0430](https://orcid.org/0000-0002-0739-0430)

<sup>1</sup>Engenharia de Controle e Automação, Universidade Federal de Santa Catarina, Campus Blumenau, Santa Catarina, Brasil. E-mail: claudio.moreira.uni@gmail.com

<sup>2</sup>Engenharia de Controle e Automação, Universidade Federal de Santa Catarina, Campus Blumenau, Santa Catarina, Brasil. E-mail: jvzanoni111@gmail.com.

<sup>3</sup>Engenharia de Controle e Automação, Universidade Federal de Santa Catarina, Campus Blumenau, Santa Catarina, Brasil. E-mail: laurown@gmail.com.

<sup>4</sup>Engenharia de Controle e Automação, Universidade Federal de Santa Catarina, Campus Blumenau, Santa Catarina, Brasil. E-mail: marcos.matsuo@ufsc.br.

<sup>5</sup>Engenharia de Controle e Automação, Universidade Federal de Santa Catarina, Campus Blumenau, Santa Catarina, Brasil. E-mail: ebrahim.el.youssef@ufsc.br.

<sup>6</sup>Engenharia de Controle e Automação, Universidade Federal de Santa Catarina, Campus Blumenau, Santa Catarina, Brasil. E-mail: carlos.moratelli@ufsc.br.

**10.25286/rep.v9i3.2803**

Esta obra apresenta Licença Creative Commons Atribuição-Não Comercial 4.0 Internacional.

Como citar este artigo pela NBR 6023/2018: Cláudio Lourenço Moreira; João Victor Zanoni; Lauro Weingartner Neto; Marcos Vinicius Matsuo; Ebrahim Samer El Youssef; Carlos Roberto Moratelli. Desenvolvimento de um robô cartesiano *pick-and-place* com 4 graus de liberdade. Revista de Engenharia e Pesquisa Aplicada, v.9, n. 3, p. 70-79, 2024.

## RESUMO

Este artigo apresenta o desenvolvimento de um robô cartesiano *pick-and-place* com 4 graus de liberdade, destinado a realizar tarefas de movimentação e posicionamento de objetos em ambientes controlados. O projeto é dividido em três fases: mecânica, elétrica e software. Na fase mecânica, o robô é montado fisicamente com a modelagem e impressão da garra. Na fase elétrica, uma placa de circuito é elaborada e os circuitos de potência e controle do manipulador são construídos, além da definição dos motores de passo, servomotores, fontes de alimentação e reguladores de tensão. Na última fase, é desenvolvido o firmware do dispositivo, que é capaz de receber comandos através de um aplicativo para smartphone, permitindo a movimentação nos três eixos cartesianos, rotação e fechamento da garra. Os principais resultados demonstram a viabilidade de construir um manipulador robótico acessível, funcional e capaz de realizar as ações esperadas de um dispositivo *pick-and-place*.

**PALAVRAS-CHAVE:** Robô Cartesiano; *Pick-and-place*; Manipulador robótico; 4 graus de liberdade.

## ABSTRACT

This article presents the development of a 4-degree-of-freedom Cartesian *pick-and-place* robot, designed to perform object movement and positioning tasks in controlled environments. The project is divided into three phases: mechanical, electrical, and software. In the mechanical phase, the robot is physically assembled with claw modeling and printing. The electrical phase involves designing a circuit board and building power and control circuits for the manipulator, along with selecting stepper motors, servo motors, power supply and voltage regulators. In the final phase, is developed the device firmware, that is capable of receiving commands from a smartphone GUI, enabling motion in three Cartesian axes, rotation, and claw closing. The main results demonstrate the feasibility of constructing an accessible, functional robotic manipulator capable of *pick-and-place* actions.

**KEY-WORDS:** Cartesian Robot; *Pick-and-place*; Robotic Manipulator; 4 degrees-of-freedom;

## 1 INTRODUÇÃO

Futuras aplicações de manufatura irão requerer uma crescente participação de robôs que executam tarefas de pick-and-place [1]. Tais robôs podem ser utilizados para manipulação de objetos em tarefas repetitivas e perigosas, ou ainda onde a presença humana é impossível e indesejável [2]. Por essas razões, os robôs *pick-and-place* são uma solução eficiente para aumentar a segurança e reduzir o risco de acidentes em processos industriais.

A operação de um robô de *pick-and-place* é normalmente dividida em quatro etapas, a primeira consiste no deslocamento do efetuador final até o objeto, para em seguida pegá-lo. Por conseguinte, o robô realiza o reposicionamento do objeto no lugar desejado. Por fim, o objeto é solto no lugar almejado. Para isso, o robô deve ter no mínimo 4 graus de liberdade (GdL), sendo 3 GdLs para translação e 1 GdL para rotação [3]. Essas operações de pick-and-place constituem a maioria das aplicações robóticas industriais da atualidade e possuem um grande potencial de expansão [4].

Os robôs pick-and-place são vitais para o *bin-picking*, uma tarefa desafiadora de coletar objetos desorganizados e posicioná-los corretamente, especialmente devido à diversidade de formatos. A capacidade desses robôs de se adaptar a esses formatos permite a execução eficiente dessa tarefa sem mudanças no hardware, como destacado por [5].

O presente artigo propõe a criação de um robô pick-and-place de baixo custo e acessível, capaz de realizar tarefas características como pegar, posicionar e soltar objetos dentro de sua área de trabalho. Também, o efetuador final do robô é projetado de forma modular, permitindo que seja facilmente trocado segundo as necessidades específicas da tarefa. Além disso, o modelo proposto pode ser utilizado para representar e simular diversas aplicações do ramo industrial, sendo controlado remotamente através de uma interface gráfica para smartphones.

O artigo é estruturado da seguinte forma: na seção 2 é realizada uma revisão bibliográfica, citando alguns trabalhos semelhantes ao desenvolvido no artigo; na seção 3, apresentam-se os materiais e métodos utilizados na elaboração do trabalho, isto é, a estrutura do robô, o efetuador final, o circuito elétrico e o *software*; na seção 4, descreve-se os resultados obtidos e alcançados no projeto; na seção 5, é feita uma conclusão apresentando uma síntese do projeto e passos futuros.

## 2 REVISÃO BIBLIOGRÁFICA

Dada a versatilidade e o suporte que um robô pick-and-place proporciona para diversos setores da indústria, a pesquisa e o desenvolvimento constante dessa solução agregam mais confiabilidade e produtividade nas tarefas para os quais os robôs são designados.

Zhang *et al.* [6] propõe o projeto de avaliação de um robô para a colheita de morangos, visando automatizar um processo repetitivo e oneroso em fazendas com espaço reduzido entre as plantações. A partir da análise dos resultados, constata-se que o robô opera na coleta dos morangos de forma satisfatória, atendendo aos requisitos propostos e se mostrando uma alternativa inovadora no meio agrícola.

Dong *et al.* [7] apresenta outra vertente de estudo: a aplicação de um sistema de controle em malha fechada para manipuladores, monitorando a trajetória realizada pelo robô e eliminando os erros de atuação, gerados por questões construtivas ou perturbações externas ao manipulador. Apesar de não haver implementação física, os resultados apresentados por meio de gráficos evidenciam a melhora gerada pelo sistema em malha fechada dos atuadores do manipulador, apontando evoluções no sistema de *pick-and-place* para qualquer tarefa, principalmente as que exigem maior precisão.

Sabendo do custo elevado para a implementação de manipuladores robóticos na indústria, em muitos casos o investimento se torna inviável. Por isso, existem pesquisas que visam reduzir o capital necessário para a construção de manipuladores, utilizando materiais flexíveis para a estrutura, em tarefas onde o manuseio do material da produção é de baixo peso, como proposto por Kato *et al.* [8]. Tal solução não só reduz o custo do robô ao substituir materiais metálicos por silicone, como também promove versatilidade e amplia a adesão dessa tecnologia em mais setores da produção.

Ainda se tratando de custos, mesmo um manipulador acessível dentro dos seus propósitos ainda tem um custo de energia associado à sua atuação. É importante que este consumo seja o menor possível, pois assim, o valor investido no robô tem o retorno em menor tempo, além de não gerar grandes impactos no custo fixo da fábrica em operação. Por isso, Goya *et al.* [9] propõe melhorar a eficiência energética de um robô SCARA, utilizando dispositivos elásticos adaptativos nas juntas para reduzir o consumo de energia das juntas durante sua operação. Através de experimentos e comparando os resultados de consumo antes e após a implementação dos dispositivos, os autores destacaram uma redução de mais de 72% no consumo de energia do manipulador, propiciando grande melhora nos

custos fixos após a implementação da tecnologia em qualquer atividade proposta.

Em resumo, os contínuos avanços no desenvolvimento de robôs pick-and-place demonstram seu impacto positivo em uma ampla gama de setores. Tais inovações não apenas aumentam segurança e eficiência das tarefas designadas aos robôs, como também redefinem a forma como vários processos são realizados.

### 3 CONSTRUÇÃO DE UM ROBÔ CARTESIANO

Nesta seção, é descrito o desenvolvimento de um robô modular de *pick-and-place* com foco na acessibilidade de custo e na dispensa do controle em malha fechada. Para atingir esse objetivo, escolhe-se o desenvolvimento de um manipulador cartesiano, também conhecido como máquina de Controle Numérico Computadorizado (CNC), que tem a capacidade de traduzir instruções computacionais em movimentos físicos do manipulador. As próximas seções detalham as principais partes e etapas implementadas no mecanismo proposto.

#### 3.1 ESTRUTURA GERAL PARA ROBÔS CARTESIANOS

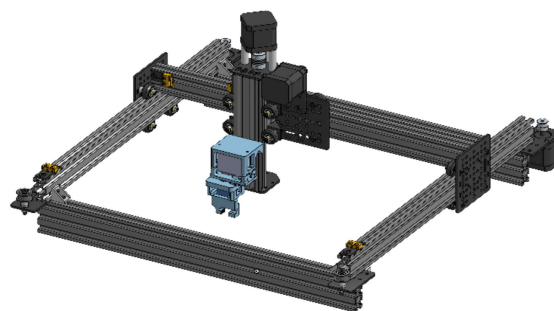
Ao analisar a estrutura de robôs cartesianos, pode-se destacar três tipos de atuadores lineares amplamente empregados em sua cinemática. O atuador com fuso trapezoidal, ao apresentar uma superfície helicoidal, possibilita o movimento linear de objetos rosqueados ao rotacionar o fuso. Sua principal vantagem é a capacidade de suportar cargas, aliada a um baixo custo e a precisão na conversão do movimento rotacional [10]. Em relação ao atuador com correia e polias, este converte o movimento de rotação do motor em translação por meio de uma correia dentada. Nesse caso, a quantidade de dentes na correia influencia na resolução e velocidade do movimento, sendo que correias com mais dentes proporcionam maior precisão, mas resultam em menor velocidade [11]. Por fim, o atuador com correia e pinhão permite ao motor, ao rotacionar, realizar o movimento translacional ao longo da barra onde a correia está fixada, ideal quando se deseja movimentar o atuador que realiza a transmissão do movimento. Sua abordagem é útil em situações em que há limitação no comprimento dos fios que alimentam e realizam a comunicação com o motor [12]. A escolha adequada do tipo de atuador para cada eixo

do robô é essencial para garantir a eficiência, robustez e precisão do sistema, considerando as características estruturais e as demandas específicas de cada aplicação.

#### 3.2 ESTRUTURA PROPOSTA PARA O ROBÔ CARTESIANO

Para elaborar o projeto, a parte mecânica e a construção do manipulador representam uma das grandes frentes a serem abordadas. Sendo assim, um esboço tridimensional do robô *pick-and-place* estimado é elaborado no *Onshape*, uma plataforma online de modelagem 3D, sendo apresentado na Figura 1.

Figura 1 – Estrutura física do robô.



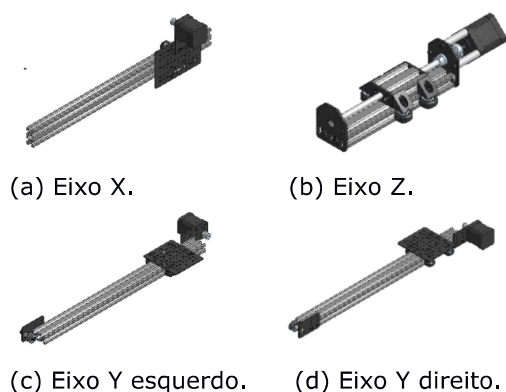
Fonte: Os Autores.

Para sua construção, utilizam-se perfis de alumínio do tipo V-Slot com 50x50x20cm. Os quais, após montados, configuram uma área de trabalho retangular de 28 cm no eixo X, 20 cm no eixo Y e 4 cm no eixo Z. Para garantir os seus limites, com o intuito de encontrar os pontos de origem do sistema, aplicam-se sensores de fim de curso em cada eixo de translação, permitindo assim o controle em malha aberta, como visto em [8, 9].

Uma etapa essencial no desenvolvimento do manipulador é a definição dos atuadores lineares de cada eixo. A fim de facilitar o projeto e permitir a sua montagem em software CAD, escolhe-se utilizar o padrão de peças *Openbuilds* [13], muito difundido na construção de máquinas CNC. Dessa maneira, considerando as características dos três tipos principais de atuadores, pode-se posicioná-los de forma lógica no eixo mais adequado. Inicialmente, escolhe-se um atuador do tipo com pinhão e correia para movimentar o eixo X, como mostra a Figura 2a. Essa escolha é necessária para economizar espaço de montagem e facilitar o acoplamento entre os eixos. Em seguida, define-se

que o eixo Y deve possuir dois atuadores do tipo com correias e polias em paralelo, representados pelas Figuras 2c e 2d. Isso acontece por que esse eixo pertence à base do robô, ou seja, deve suportar o peso de todos os elos subsequentes, fator que exige um maior torque. Por último, o eixo Z é formado por um atuador linear com fuso trapezoidal, mostrado na Figura 2b. Ademais, essa configuração é muito utilizada em diversas máquinas CNC, porque permite um ajuste preciso de altura.

Figura 2 – Atuadores lineares.



Fonte: Os Autores.

Com o resultado, o robô possui 4 graus de liberdade, conseguindo transladar com juntas prismáticas nos eixos X, Y e Z e de rotacionar com uma junta de revolução em torno do eixo Z [8]. Por fim, acopla-se ao eixo Z um efetuator final (ver Seção 3.3) do tipo garra, sendo projetado e manufaturado especialmente para essa aplicação, conseguindo segurar e pinçar componentes pequenos e leves.

### 3.3 EFETUADOR FINAL

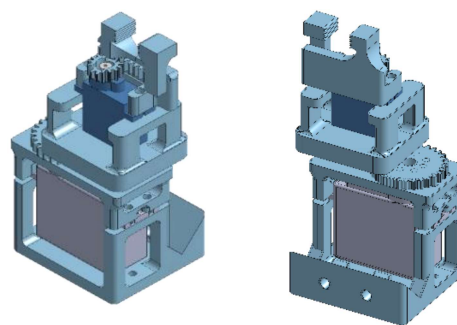
A construção do efetuator final em forma de garra robótica é uma tarefa fundamental para qualquer robô de *pick-and-place* [1], pois a garra é responsável por segurar e soltar objetos com precisão. Para realizar os movimentos, a garra possui dois tipos de motores: um servo MG995, responsável por realizar uma rotação em torno do eixo Z, e um servo 9g, encarregado de acionar o mecanismo de pinhão e cremalheira e fazer a garra abrir e fechar, como uma pinça.

Sabe-se que o motor MG995 possui uma limitação de rotação de 180°. Para contornar esse problema, adota-se uma relação de engrenagens de

1 para 2 que permite a transformação do movimento em uma amplitude de 360° possibilitando assim uma rotação completa em torno do eixo Z. Desse modo, os dois motores trabalham em conjunto para proporcionar a movimentação completa da garra.

Além disso, a garra é projetada em um software de modelagem 3D CAD, permitindo a visualização completa da geometria e dos componentes da mesma antes da sua fabricação [13]. Após tal etapa de projeto, utiliza-se impressão 3D com filamento de ácido polilático (PLA), o qual é um material de fácil manufatura e baixo custo. Ademais, uma das grandes vantagens adicionais da garra é que ela pode ser trocada futuramente, como se fosse um módulo do robô cartesiano, aumentando assim a versatilidade do sistema. Na sequência, apresenta-se nas Figuras 3.a e 3.b, em perspectiva isométrica o efetuator final projetado.

Figura 3 – Efetuator final do tipo garra



(a) Visão frontal. (b) Visão traseira.

Fonte: Os Autores.

### 3.4 CIRCUITO ELÉTRICO

A parte elétrica do robô engloba os sensores de fim de curso, circuitos de comando e potência dos motores, o microcontrolador e as fontes de tensão.

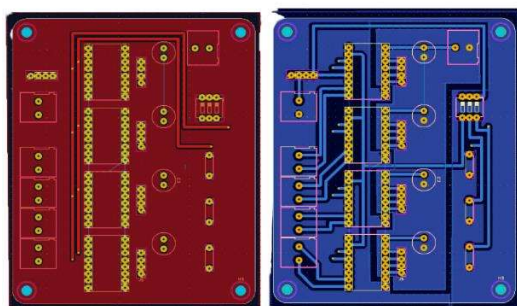
O controle do robô é realizado por meio da utilização do kit de desenvolvimento ESP32. Esse dispositivo desempenha um papel central na tomada de decisões, permitindo comandar os motores dos eixos X, Y e Z, bem como a garra, com base nas informações dos sensores e nas instruções recebidas.

Para proporcionar movimento aos atuadores lineares mencionados na Seção 3.1, escolhem-se motores de passo do tipo NEMA 17. Para controlar esses motores, utilizam-se *drivers* eletrônicos DRV8825. Além disso, é adotada uma resolução de 1/4 de passo para um controle mais preciso [14].

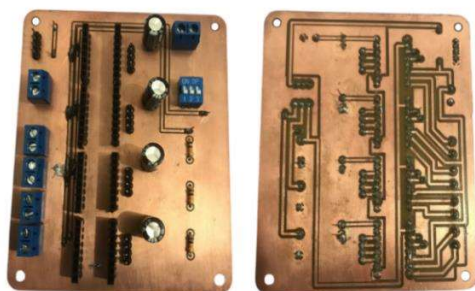
A alimentação dos motores Nema 17 é fornecida por uma fonte de 24 Volts e 20 Amperes, resultando em uma potência de 480 W. Por outro lado, para alimentar os servomotores MG995 e 9g presentes no efetuador, utiliza-se um regulador de tensão LM2596, que recebe a mesma tensão de 24 V dos motores Nema e a reduz para 5V [14].

Visto a abundância de ligações necessárias e as possíveis interferências e resistências causadas, opta-se por projetar e utilizar uma Placa de Circuito Impresso (PCI). Para sua elaboração, utilizou-se o software *KiCad* para projeto juntamente com uma máquina fresadora para impressão. A PCI elaborada é apresentada na Figura 4 e concentra toda a parte de potência e energia dos motores Nema 17 e dos sensores de fim de curso.

Figura 4 – Placa de Circuito Impresso.



(a) Projeto *KiCad*.



(b) Placa Impressa.

Fonte: Os Autores.

### 3.5 ORÇAMENTO DO PROJETO

Com o levantamento de todos os componentes necessários, tanto na parte elétrica quanto na parte estrutural do robô, tem-se na Tabela 1 o orçamento do projeto, totalizando R\$ 1532,58.

Tabela 1- Relação de materiais e preços para construção do projeto.

Fonte: Os autores.

### 3.6 SOFTWARE

Inicialmente, definiu-se que a programação do

Material	Descrição	Preço
Perfis de Alumínio	Perfis Tipo V-Slot	R\$ 213,00
Kit Carrinho	Carrinhos de movimentação	R\$ 360,34
Porcas, parafusos, e acoplamentos	Fixação das peças do robô	R\$ 183,29
Fuso trapezoidal	Estrutura para movimentação do eixo Z	R\$ 33,55
Polias, correias e roldanas	Estrutura para movimentação nos eixos X e Y	R\$ 105,84
Fonte de alimentação 24V	Fornecimento de energia para os componentes	R\$ 78,99
Nema 17	Motor de Passo	R\$ 287,56
MG995	Servo motor	R\$ 89,98
ESP32	Microcontrolador	R\$ 20,51
Sensores, Drivers e Cabos	Componentes da parte elétrica	R\$ 159,52

robô é feita passivamente *off-line*, ponto a ponto. Ou seja, definem-se previamente os pontos iniciais e finais de cada movimento e não se considera a trajetória que o manipulador irá percorrer.

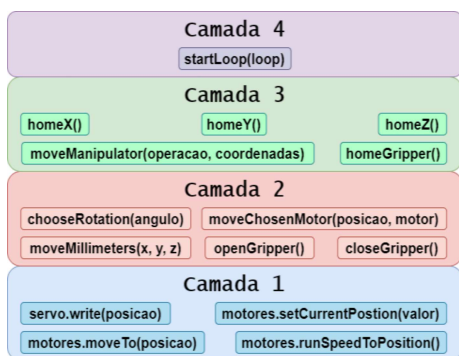
#### 3.6.1 Software Embarcado

Para abstrair o funcionamento do robô, as funções criadas podem ser organizadas em camadas, como se observa na Figura 5. Desse modo, a Camada 1 é composta por métodos fornecidos pelas duas principais bibliotecas utilizadas: a *AccelStepper* [15], para comandar os motores de passo, e a *ESP32Servo* para acionar os servomotores da garra.

Inicialmente os motores de passo somente conseguem ser acionados informando o número de passos que eles precisam se deslocar. Dessa maneira, criam-se as funções da Camada 2 para possibilitar o deslocamento em milímetros de cada atuador linear. Além disso, implementam-se funcionalidades para controle da garra.

A Camada 3 é uma abstração que leva em consideração os sensores de fim de curso. Dessa maneira, definem-se funções para trazer todos os eixos do robô para a origem da área de trabalho. Assim, o erro de posicionamento do robô é reiniciado, dispensando alguma técnica de controle em malha fechada [7]. Nessa camada também é criada uma função que realiza operações de *pick-and-place* em uma coordenada passada como parâmetro. A Camada 4 visa iniciar rotinas de *pick-and-place* predefinidas, com o intuito de demonstrar a funcionalidade do projeto.

Figura 5 – Estrutura de Funções do Software.



Fonte: Os Autores.

Para converter um determinado deslocamento linear (dado em milímetros) na quantidade de passos (S) necessária que cada motor deve realizar, utilizam-se as Equações 1 e 2. Dessa maneira, elas são aplicadas respectivamente nos eixos cujos atuadores utilizam correia e fuso trapezoidal. Por isso, para utilizá-las, devem ser definidos os seguintes parâmetros:

$$S = \text{round} \left( \frac{F \cdot L \cdot N}{R \cdot D \cdot \pi} \right) \quad (1)$$

$$S = \text{round} \left( \frac{L \cdot N}{R \cdot B} \right) \quad (2)$$

Onde:

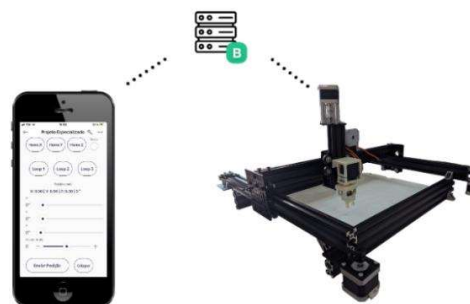
- L: posição desejada absoluta em milímetros;
- N: quantidade de passos necessários para realizar uma revolução em modo *full-step*. Para o NEMA 17 o valor padrão é 200.
- R: resolução de *microstep* escolhida no driver (*full-step*, *half-step*, *1/4 step*, *1/8 step*, *1/16 step* ou *1/32 step*);
- D: diâmetro da polia em milímetros;

- F: fator de correção percentual para ajuste fino do posicionamento. Pode ser obtido com medições experimentais.
- B: deslocamento em milímetros do fuso em uma volta completa. Para o fuso utilizado o valor padrão é 8 milímetros.

### 3.6.2 Interface Gráfica

Visando controlar o robô remotamente, criou-se uma interface gráfica que permite a manipulação do robô a distância. Dessa maneira, utilizou-se uma ferramenta chamada *Blynk* [16]. Ela possibilita a integração entre o microcontrolador do projeto com uma interface de comandos, via *webservices*. Assim, ele integra o microcontrolador do projeto com o painel de comandos via um servidor próprio. Essa integração é realizada por *Wi-fi*, permitindo que o robô e o dispositivo de controle não precisem estar conectados à mesma rede. Todo o envio e recebimento de dados é intermediado pela plataforma, cuja lógica é mostrada na Figura 6.

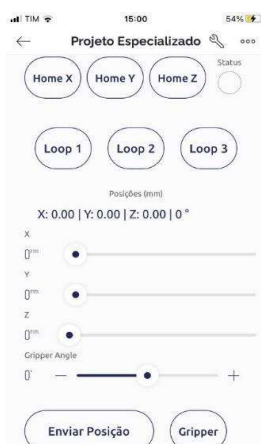
Figura 6 – Esquema Lógico de Comunicação da Interface Gráfica com o Robô.



Fonte: Os Autores.

Dessa forma, dispensa-se a necessidade de implementar um *webservice* [17] com hospedagem em um computador pessoal ou no próprio microcontrolador do robô. Isso proporciona benefícios adicionais, como a alocação completa do processamento para o acionamento dos atuadores e a verificação dos sensores, bem como a facilidade na comunicação remota por meio de um smartphone. A interface gráfica do smartphone, apresentada na Figura 7, permite uma interação intuitiva e eficiente. Com isso, é possível controlar o robô de forma remota, mantendo uma separação clara entre a camada de controle e a camada de interface.

Figura 7 – Interface Gráfica de Comandos do Robô.



Fonte: Os Autores.

Na interface são dispostos comandos que podem ser executados ao pressionar de um botão. Na primeira linha, os botões denominados "Home X", "Home Y" e "Home Z" enviam o robô para a posição inicial dos respectivos eixos, utilizando os sensores de fim de curso para verificar se o manipulador chega na posição de origem. Na segunda linha, os botões *Loop 1*, *Loop 2* e *Loop 3* executam rotinas pré-programadas de *pick-and-place*, movimentando blocos entre as plataformas posicionadas na área de trabalho do robô. Ainda, na segunda metade da interface da Figura 7, implementa-se quatro comandos do tipo *slider* que permitem escolher uma posição específica no espaço de trabalho do robô, além da orientação da garra em ângulos que variam de  $-90^\circ$  a  $90^\circ$ , com passo de  $45^\circ$  graus entre eles.

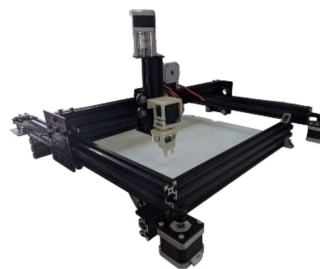
Por fim, o botão "Enviar posição" envia o comando para o robô iniciar sua movimentação até a posição e orientação passada nos *sliders*, enquanto o botão "Gripper" ao lado, envia o comando de abertura e fechamento da garra do manipulador.

Nota-se que a plataforma *Blynk* [16] oferece uma solução robusta e com diversos recursos que facilitam a criação da interface e integração com o microcontrolador desejado, nesse caso, a ESP32.

## 4 RESULTADOS

A partir dos modelos 3D da estrutura física do robô e do efetuator final mostrados nas Figuras 1 e 3, montou-se a estrutura mecânica final do robô. O planejamento e elaboração de modelos virtuais precisos permitiu que a realização física do robô, apresentada na Figura 8, resultasse em uma estrutura adequada e capaz de realizar os movimentos de translação e rotação previstos.

Figura 8 – Estrutura Mecânica do Robô.

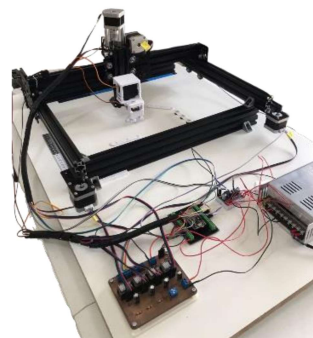


Fonte: Os Autores.

Após essa validação, a parte elétrica do robô é instalada, ligando alimentação, microcontrolador, PCI, sensores e atuadores junto a estrutura mecânica, como mostra a Figura 9.

Com a finalização das frentes mecânica e elétrica, o software é carregado para a ESP32, permitindo a conexão do microcontrolador em uma rede *Wi-fi*, e consequentemente o controle do robô via interface gráfica desenvolvida no *Blynk*, conforme explicado na Seção 3.5.2.

Figura 9 – Estrutura Final do Robô



Fonte: Os Autores.

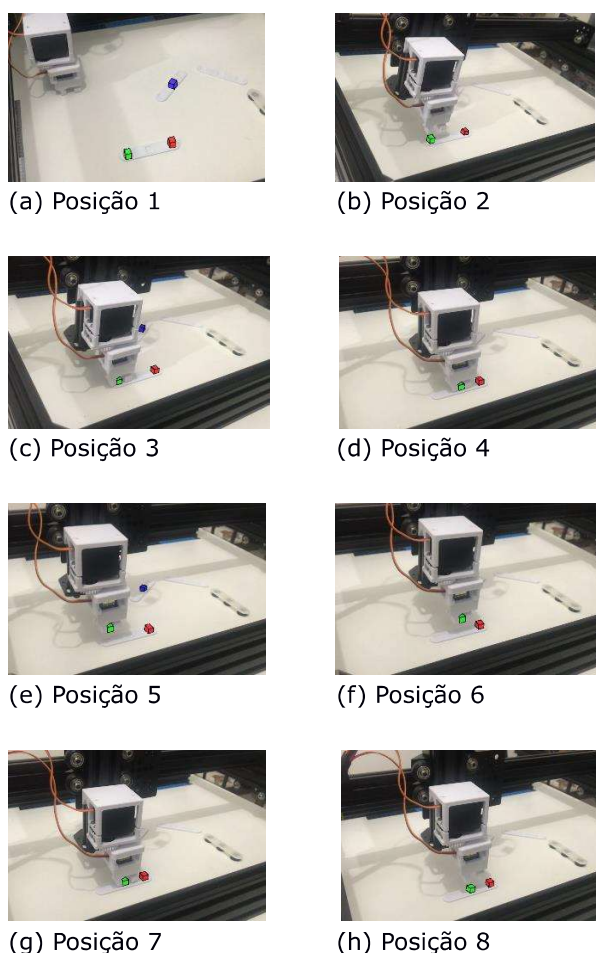
Assim, todos os botões e controles implementados na interface são testados, verificando a robustez tanto no envio dos comandos quanto na execução dos atuadores lineares, rotação, abertura e fechamento da garra. Com essa bateria de testes, evidencia-se o bom funcionamento da integração do sistema, com a interface operando de forma intuitiva e enviando os comandos que por sua vez, são bem executados pela estrutura final do robô.

Para testar a movimentação e execução de comandos do manipulador, são criados dois *loops*, ou rotinas, de *pick-and-place* (ver seções 4.1 e 4.2). Assim, é possível verificar a assertividade do robô ao posicionar pequenos cubos com arestas de 8mm em posições predefinidas.

#### 4.1 ROTINA PRÉ-PROGRAMADA LOOP 1

A Figura 10 apresenta uma demonstração do manipulador realizando a função de *pick-and-place* em um cubo de dimensão 8mm x 8mm x 8 mm para duas posições pré-determinadas em um mesmo plano. Essa demonstração pode ser iniciada através do botão LOOP 1. Inicialmente, a garra vai para a origem da área de trabalho, como pode-se observar na Figura 10a. Em seguida, a garra fica acima da coordenada especificada, e posteriormente se abaixa até alcançar a altura correta, como mostram as Figuras 10b e 10c. Por conseguinte, o manipulador pega a peça, a eleva e se locomove até as coordenadas em X e Y do destino, como se observa nas Figuras 10d, 10e e 10f. Por fim, a garra vai até a coordenada em Z do destino, solta a peça e retorna para a altura anterior, conforme as Figuras 10g e 10h.

**Figura 10** – Operação de *Pick-and-Place* do Loop 1.

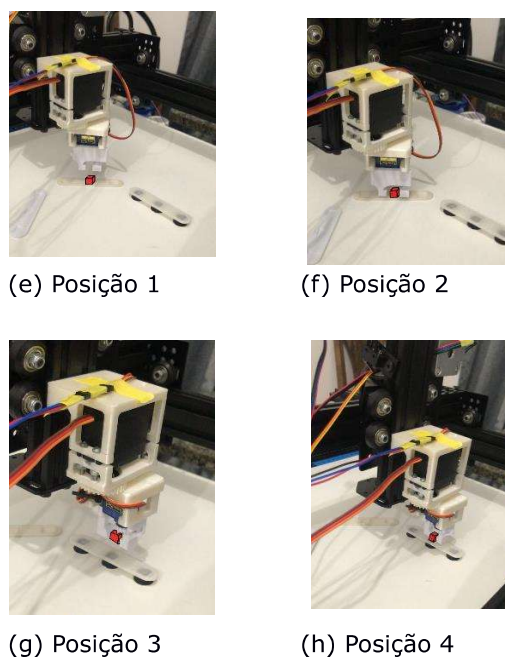


**Fonte:** Os Autores.

#### 4.2 ROTINA PRÉ-PROGRAMADA LOOP 2

Para o botão LOOP 2, após o envio do manipulador para a origem, a garra se movimenta até a primeira posição de *pick*, como mostra a Figura 11a. Em seguida, o manipulador translada no eixo Z, movendo a garra para baixo a fim de pegar o bloco, ilustrado na Figura 11b. Com o bloco preso, o manipulador eleva no eixo Z, dando início ao processo de *place* do *loop*, movimentando o bloco até a outra plataforma, como mostra a Figura 11c. Nota-se que, para esse processo, a plataforma onde é realizado o *place* possui uma elevação em relação as demais. Desse modo, é importante que o robô realize um movimento de translação em Z adequado para que o bloco e a garra não colidam com a plataforma. Assim, na Figura 11d, o manipulador finaliza o *place* do bloco, descendo sobre o eixo Z do destino e soltando a peça.

**Figura 11** – Operação de *Pick-and-Place* do Loop 2.



**Fonte:** Os Autores.

#### 4.3 ANÁLISE DOS RESULTADOS

A análise dos resultados obtidos no artigo científico revela um panorama adequado para diferentes contextos operacionais. Apesar de não enfrentar as restrições espaciais descritas por Zhang et al. [6], é notável que a precisão do robô se mantém satisfatória durante a manipulação de objetos menores. Contudo, observou-se um desvio



significativo na posição quando rotinas pré-programadas são mantidas por longos períodos, sem efetuar a recalibração da referência de origem. Acredita-se que isso possa ser melhorado aplicando-se estratégias de aprimoramento, aliadas a sistemas de realimentação do efetuador final, conforme proposto por *Dong et al.* [7], que visam mitigar esse tipo de erro, permitindo um controle mais eficaz em malha fechada.

Em relação à estrutura do robô, construída com perfis V-Slot, o equilíbrio entre custo e desempenho é destacado como uma conquista significativa. Embora os testes realizados não tenham explorado totalmente a sua capacidade de carga, a análise indica que há margem para redução de custos ao considerar alternativas de materiais, corroborando descobertas anteriores de *Kato et al.* [8]. Esse aspecto ressalta a importância da escolha criteriosa dos componentes estruturais para alcançar uma relação econômica viável sem comprometer a eficiência operacional.

Na parte de controle do manipulador, a interface desenvolvida para smartphones se mostrou indispensável para os testes dos Loops e ajustes finos por meio dos *sliders*, mostrados na Figura 7. A abstração desses comandos permite uma interação intuitiva e didática do manipulador, despertando o interesse de alunos de diferentes faixas etárias no projeto.

Um ponto crucial que emerge da análise dos resultados é a precisão do eixo Z do manipulador, particularmente evidenciada durante a execução contínua da rotina pré-programada Loop 2. A utilização do fuso trapezoidal para movimentação linear se revela como uma escolha acertada, contribuindo para a estabilidade e confiabilidade do sistema. Essas constatações destacam a relevância de selecionar os tipos adequados de atuadores para cada eixo.

## 5 CONCLUSÃO

Este trabalho apresentou as etapas de projeto e construção de um robô cartesiano pick-and-place com 4 graus de liberdade. A elaboração física possibilitou o estudo dos tipos de atuadores lineares usualmente empregados nas soluções comerciais, bem como o aprofundamento em projetos tridimensionais em softwares CAD e na manufatura aditiva a partir da impressão 3D. Para a parte elétrica, foi realizada a confecção de uma placa de circuito impresso modelada no software KiCad. Utilizou-se um microcontrolador ESP32, cujo firmware desenvolvido permite o controle do

equipamento remotamente. Os resultados destacam o potencial do robô como uma solução viável, acessível e promissora para tarefas de manipulação automatizada em diferentes contextos industriais. Como trabalhos futuros, pretende-se utilizar visão computacional para o reconhecimento de objetos e a implementação de trajetórias adaptativas ao movimento do efetuador final.

## REFERÊNCIAS

- [1] LI, Fengchun; JIANG, Yao; LI, Tiemin; FENG, Yixiao; CHEN, Shuqing. **Design of a robot end effector with measurement system for precise pick-and-place of square objects**. *Procedia Manufacturing*, v. 48, p. 172-180. Elsevier BV, 2020. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2351978920314864>. Acesso em: 4 abr. 2023.
- [2] KHURSHID, Rebecca P.; FITTER, Naomi T.; FEDALEI, Elizabeth A.; KUCHENBECKER, Katherine J. **Effects of Grip-Force, Contact, and Acceleration Feedback on a Teleoperated Pick-and-Place Task**. *Ieee Transactions On Haptics*, v. 10, n. 1, p. 40-53, 1 jan. 2017. Institute of Electrical and Electronics Engineers (IEEE). Disponível em: <https://ieeexplore.ieee.org/document/7479565>. Acesso em: 10 abr. 2023.
- [3] ZHANG, Yong; CHEN, B.K.; LIU, Xinyu; SUN, Yu. **Autonomous Robotic Pick-and-Place of Microobjects**. *Ieee Transactions On Robotics*, v. 26, n. 1, p. 200-207, fev. 2010. Institute of Electrical and Electronics Engineers (IEEE). Disponível em: <https://ieeexplore.ieee.org/document/6282812>. Acesso em: 4 abr. 2023.
- [4] MNYUSIWALLA, Hussein; Pavlos TRIANTAFYLLOU; Panagiotis SOTIROPOULOS; Maximo A. ROA; Werner FRIEDL; Ashok M. SUNDARAM; Duncan RUSSELL; Graham DEACON. **A Bin-Picking Benchmark for Systematic Evaluation of Robotic Pick-and-Place Systems**. *IEEE Robotics And Automation Letters*. Hatfield, p. 1-18. dez. 2019. Disponível em: <https://ieeexplore.ieee.org/document/8954746>. Acesso em: 4 abr. 2023.
- [5] D'AVELLA, Salvatore; TRIPICCHIO, Paolo; AVIZZANO, Carlo Alberto. **A study on picking objects in cluttered environments**: Exploiting depth features for a custom low-cost universal jamming gripper. *Robotics And Computer-Integrated Manufacturing*. Pisa, p. 1-10. out. 2019.

Disponível em:

<https://www.sciencedirect.com/science/article/pii/S0736584519307276>. Acesso em: 10 abr. 2023.

- [6] ZHANG, Yaohui; ZHANG, Kailiang; YANG, Li; ZHANG, Dongxing; CUI, Tao; YU, Yang; LIU, Hui. **Design and simulation experiment of ridge planting strawberry picking manipulator**. Computers And Electronics In Agriculture. Pequim, p. 1-18. 30 jan. 2023. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169923000789>. Acesso em: 04 abr. 2023.
- [7] DONG Jian; HE Bin; MING Ma; ZHANG Chengchong; LI Gang. **Design of Open-Closed-Loop Iterative Learning Control With Variable Stiffness for Multiple Flexible Manipulator Robot Systems**. IEEE Access, vol. 7, p. 23163-23168. 2019. Disponível em: <https://ieeexplore.ieee.org/document/8638774>. Acesso em: 04 abr. 2023.
- [8] KATO Gustavo; ONCHI Diego; ABARCA Monica. **Low Cost Flexible Robot Manipulator for Pick and Place Tasks**. 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), p. 677-680. 2013. Disponível em: <https://ieeexplore.ieee.org/document/6677451>. Acesso em 04 abr. 2023.
- [9] GOYA Hidemasa; MATSUSAKA Kento; UEMURA Mitsunori; NISHIOKA Yasutaka; KAWAMURA Sadao. **Realization of High-Energy Efficient Pick-and-Place Tasks of SCARA Robots by Resonance**. International Conference on Intelligent Robots and Systems, p. 7-12. Out. 2012. Disponível em: <https://ieeexplore.ieee.org/document/6386084>. Acesso em 10. Abr. 2023.
- [10] GRECHISHNIKOV, V. A.; PETUKHOV, Yu. E.; ROMANOV, V. B.; ISAEV, A. V.; DOMNIN, P. V.; PIVKIN, P. M.. **Improved Precision of Trapezoidal Thread**. Russian Engineering Research, v. 38, n. 12, p. 1018-1021, dez. 2018. Allerton Press. Disponível em: <https://link.springer.com/article/10.3103/S1068798X18120079>. Acesso em 10 abr. 2023.
- [11] JUVINALL, Robert C.; MARSHEK, Kurt. **Projeto de Componentes de Máquinas**. 4. ed. Rio de Janeiro: Ltc. 500 p. 2008.
- [12] EHRMANN, Christopher; ISABEY, Philippe; FLEISCHER, Jürgen. **Condition Monitoring of Rack and Pinion Drive Systems** and challenges in production environments. Procedia Cirp, v. 40, p. 197-201. Elsevier BV. 2016. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2212827116001165>. Acesso em: 10 abr. 2023.
- [13] OPENBUILDS. **Openbuilds**. Disponível em: <https://openbuilds.com/>. Acesso em: 15 fev. 2023.
- [14] ACARNLEY, Paul. **Stepping Motors: A Guide to Theory and Practice**. London, England: Institution of Engineering and Technology, 2004.
- [15] MCCAULEY, MICHAEL. **AccelStepper**. Disponível em: <https://arduino.cc/reference/en/libraries/accelstepper/>. Acesso em: 30 fev. 2023.
- [16] BLYNK. **Blynk**. Disponível em: <https://blynk.io/>. Acesso em: 12 abr. 2023.
- [17] RICHARDSON, Leonard; RUBY, Sam. **Restful Web Services**. Sebastopol, California: O'reilly Media, Inc., 2008.