

# Implementação da cinemática direta e inversa do protótipo de um braço robótico no MATLAB

1º Mariana Pedroso Naves

Engenharia de Computação

UTFPR - Apucarana

Paraná, Brasil

**Abstract**—Braços robóticos desempenham um papel essencial em diversos campos, desde a indústria até as salas de cirurgia e tarefas cotidianas. Este artigo apresenta o desenvolvimento de um protótipo de braço robótico com 4 graus de liberdade (DOF), projetado e controlado com o auxílio do MATLAB. O protótipo utiliza um Arduino Uno como controlador principal e é operado por meio de dois joysticks, responsáveis por acionar os atuadores que movimentam as juntas do braço robótico. A estrutura foi impressa em PLA utilizando a impressão 3D. Este estudo, portanto, aborda a construção do braço robótico, a seleção de componentes como servomotores e microcontroladores, além da implementação da cinemática direta e inversa no MATLAB. Os principais objetivos incluem: documentar o design completo do protótipo impresso em 3D; detalhar a integração dos joysticks e servomotores no sistema; implementar a cinemática inversa e direta e demonstrar o movimento em tempo real do braço robótico com comandos enviados via comunicação serial.

**Index Terms**—Braço robótico, simulação, cinemática direta, cinemática inversa, MATLAB, educação em robótica.

## I. INTRODUÇÃO

Os braços robóticos têm evoluído significativamente com o avanço da tecnologia, trazendo melhorias em graus de liberdade, precisão e dinamismo. Inspirados no funcionamento do braço humano, eles possuem juntas que desempenham funções semelhantes às do ombro, cotovelo e pulsos [6]. Sua garra pode ser adaptada para diferentes finalidades, permitindo o uso de diferentes acessórios como pinças, ventosas, ferramentas magnetizadas e tantas outras configurações [6]. Desenvolvidos inicialmente para o âmbito comercial, esses robôs estão cada vez mais presentes em diversos setores, substituindo o trabalho humano, como na medicina realizando cirurgias robóticas, e na indústria, em tarefas como soldagem, pintura e movimentação de objetos [3].

Sob essa ótica, percebe-se que os braços robóticos oferecem diversas vantagens, como a capacidade de operar em áreas de risco, garantindo a segurança humana, além de executar tarefas por longos períodos sem interrupção [3]. Ademais, outras vantagens incluem maior precisão e a redução de erros em processos críticos [2]. Assim, nota-se o impacto positivo dos robôs manipuladores e sua evolução contínua nas áreas de elétrica, computação e sistemas de automação.

Este artigo destaca o desenvolvimento de um braço robótico de baixo custo, projetado com impressão 3D em PLA, como uma ferramenta educacional para a área de robótica. Além

disso, o artigo apresenta a implementação do controle do braço, operado por joysticks, além do desenvolvimento da cinemática direta e inversa utilizando MATLAB [2]. A proposta visa possibilitar o controle em tempo real dos movimentos do braço robótico por meio de comunicação serial.

O conteúdo deste artigo está estruturado da seguinte forma: após a introdução na Seção I, os detalhes do design do braço robótico serão discutidos na Seção II, seguidos pelas informações técnicas sobre o funcionamento do protótipo na Seção III. A Seção IV explora a implementação da cinemática inversa e direta no MATLAB juntamente com a simulação, e, por fim, a Seção V traz as considerações finais e sugestões para trabalhos futuros.

## II. DESIGN E MATERIAIS DO BRAÇO ROBÓTICO

### A. Design em 3D

Para o desenvolvimento do braço robótico foi utilizado como base o tutorial desenvolvido por theGHIZmo [7]. As peças foram obtidas por meio da plataforma open source Thingiverse [5], conforme ilustrado na Figura 1. Após a impressão, o braço robótico montado ficou conforme demonstrado na Figura 2.

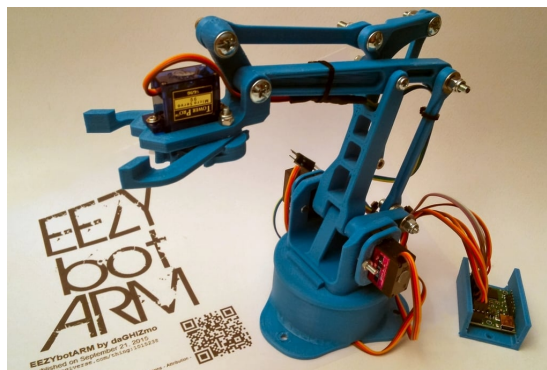


Fig. 1. Modelo de theGHIZmo. Fonte: Thingiverse (2025)

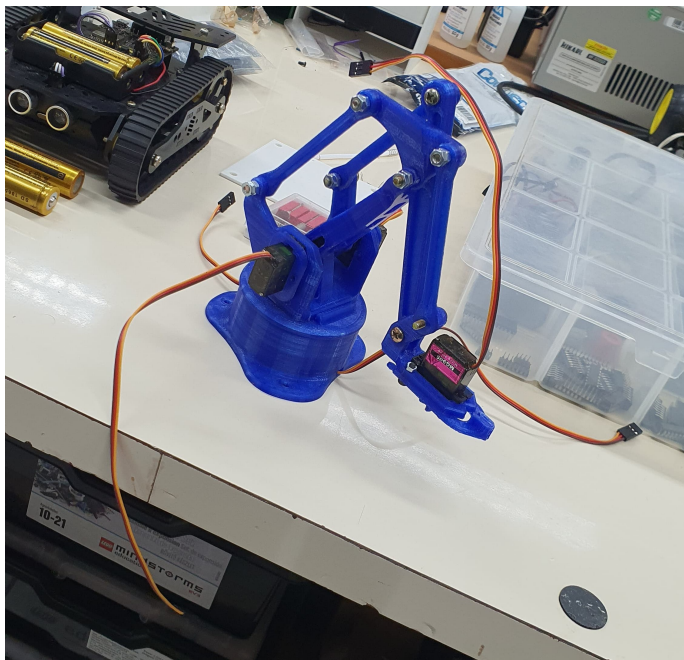


Fig. 2. Modelo final. Fonte: Autoria Própria (2025)

### B. Materiais

Para a confecção do braço robótico como um todo foi utilizado os seguintes materiais:

- Impressão 3D - material PLA;
- 4 servos motores MG90S de 180°;
- 2 módulos joysticks KY-023;
- 4 potenciômetros lineares de 10k;
- Arduino Uno;
- Parafusos e porcas;
- Jumpers;
- Protoboard e
- Shield Servo Motor - 16 Canais.

Os materiais utilizados evidenciam que o projeto consiste em um protótipo de baixo custo, sem comprometer sua funcionalidade. Os servomotores MG90S foram escolhidos devido ao seu tamanho pequeno, custo acessível e capacidade de atender às demandas do braço robótico. A shield de 16 canais foi utilizada para conectar e alimentar os servomotores, permitindo o uso de uma fonte de energia externa de 5V, sem sobrecarregar o Arduino Uno. Já os potenciômetros e joysticks foram utilizados para manipular os movimentos do braço robótico.

## III. FUNCIONAMENTO DO BRAÇO ROBÓTICO

Inicialmente, os movimentos do braço robótico foram testados utilizando 4 potenciômetros, conforme ilustrado na Figura 3. Dois potenciômetros controlavam os elos do braço, enquanto os outros dois eram responsáveis pela movimentação da base e da garra. Todo projeto foi desenvolvido em C++ na IDE do Arduino, com o objetivo de identificar possíveis problemas e limitações relacionadas à angulação dos movimentos do

braço robótico. O sistema foi alimentado com uma fonte externa de 5V-6V.

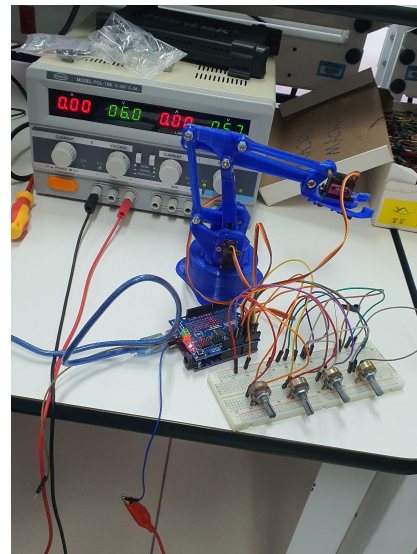


Fig. 3. Teste inicial com potenciômetros. Fonte: Autoria Própria (2025)

Após a fase inicial, os potenciômetros foram substituídos por dois módulos de joystick (Figura 4), que desempenham a mesma função de controle. Cada módulo de joystick possui dois potenciômetros integrados, permitindo que um joystick controle a base e a garra, enquanto o outro controla os dois elos do braço robótico. Essa alteração proporcionou maior praticidade para determinação dos ângulos e melhor controle do sistema.

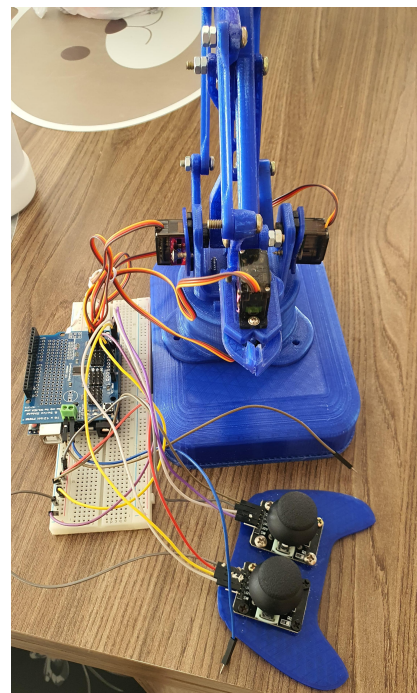


Fig. 4. Trabalho final com os módulos de joystick. Fonte: Autoria Própria (2025)

### A. Grau de liberdade

Para o cálculo do grau de liberdade (*DOF*) de um braço robótico pode ser calculado utilizando a seguinte Fórmula de Grübler [2] que define o número de movimentos independentes que o sistema pode realizar. A fórmula é expressa por:

$$DOF = m(N - 1 - J) + \sum_{i=1}^J f_i, \quad (1)$$

onde:

- $m$  representa o número de graus de liberdade no espaço, sendo ( $m = 6$  para sistemas espaciais e ( $m = 3$  para sistemas planares.
- $N$  é o número total de corpos do sistema, incluindo a base fixa (solo)
- $J$  é o número total de juntas no sistema;
- $f_i$  é o número de graus de liberdade permitido por cada junta ( $f_i = 1$  para uma junta rotacional ou deslizante);

No caso deste projeto, o braço robótico é composto por um sistema espacial, com 4 juntas rotacionais e todas as juntas possuem ( $f_i = 1$  permitindo o movimento em um único eixo resultando em:

$$DOF = 6(4 - 1 - 4) + \sum_{i=1}^4 1 = 4. \quad (2)$$

Portanto, o braço robótico possui 4 graus de liberdade, permitindo movimentos independentes em cada junta.

### B. Classificação do Protótipo

O protótipo desenvolvido é classificado como um robô manipulador articulado limitado (pois possui menos de 6 graus de liberdade), possuindo um manipulador antropomórfico com uma garra voltada para apreensão de objetos cilíndricos. Ademais, o protótipo é composto por 3 juntas de revolução, caracterizadas como RRR (Revolução-Revolução-Revolução), o que permite uma grande flexibilidade e capacidade de realizar movimentos precisos em um espaço tridimensional [2].

Essa configuração é comumente utilizada em aplicações industriais e educacionais, devido à simplicidade de construção e ao alto nível de funcionalidade proporcionado pelos graus de liberdade disponíveis. Além disso, possui uma boa manobrabilidade e controle da velocidade de operação do sistema.

### C. Problemas no funcionamento

1) *Limitações:* Durante a manipulação dos movimentos, tanto com os joysticks quanto com os potenciômetros, foram identificadas limitações na amplitude de movimento de cada elo, da base e da garra. Embora os servomotores utilizados tenham capacidade para realizar movimentos de até 180°, as restrições impostas pela estrutura física do protótipo limitaram os ângulos de atuação efetiva. Essas limitações estão descritas na Tabela I.

TABLE I  
LIMITAÇÃO DO BRAÇO ROBÓTICO

| Elos  | Angulo inicial | Angulo Mínimo | Angulo Máximo |
|-------|----------------|---------------|---------------|
| Base  | 0°             | 0°            | 170°          |
| Elo 1 | 90             | 30°           | 150°          |
| Elo 2 | 90             | 30°           | 137°          |
| Garra | 0              | 10°           | 95°           |

2) *Trepidações:* Inicialmente, o braço robótico apresentava muita trepidação durante os movimentos. Os servos motores estavam traduzindo diretamente o sinal recebido pelo potenciômetro em ângulos, mas devido à grande variação no sinal do potenciômetro, eles tremiam ao tentar atingir o dado ângulo.

Para corrigir esse problema foi implementada a média móvel. Ao invés de enviar o sinal do potenciômetro diretamente aos servos motores, cada leitura foi armazenada em um vetor contendo 10 amostras. Dessa forma, a cada iteração a média dessas amostras foi calculada e enviada a cada servo motor, garantindo maior estabilidade nos movimentos. Importante ressaltar que a cada nova leitura, o valor mais antigo no vetor era substituído, mantendo o cálculo da média atualizado e amenizando os ruídos do sinal, reduzindo significativamente a trepidação do braço.

Além disso, foi implementado um filtro passa-baixa, que contribuiu para suavizar as leituras analógicas, reduzindo ainda mais os ruídos e oscilações. Essas estratégias incorporadas na IDE do Arduino, portanto, minimizaram significativamente a trepidação e melhoraram a fluidez do movimento do protótipo.

3) *Aquecimento:* Os servomotores apresentavam problemas de superaquecimento, especialmente aqueles responsáveis por controlar os dois elos e a garra. O superaquecimento ocorria porque os servos não conseguiam atingir o ângulo estipulado, seja durante a execução de um movimento, seja ao retornar à posição inicial ou final. Além disso, ângulos fora das limitações físicas do servo contribuíam para o esforço excessivo dos servos motores, aquecendo-os mais do que deveriam.

Para solucionar esse problema, foram implementadas no código, além de analisar os ângulos mínimos e máximos, duas abordagens principais: a função `constrain` e um array para controle dos incrementos ou decrementos nos ângulos.

A função `constrain` foi utilizada para limitar os ângulos enviados aos servos, garantindo que permanecessem dentro de seus limites operacionais.

Já o array foi projetado para definir os passos de movimento do servo, com valores que correspondem às variações no ângulo baseado na leitura analógica filtrada. Cada posição do array representa uma magnitude de movimento: valores positivos aumentam o ângulo, valores negativos o diminuem, e o valor zero mantém o servo na posição atual.

Dessa forma, essas implementações proporcionaram ajustes suaves nos movimentos, reduzindo o esforço dos servos e, consequentemente, o aquecimento, prevenindo uma possível queima do componente.

#### IV. IMPLEMENTAÇÃO NO MATLAB

A implementação foi realizada utilizando a versão R2023a do MATLAB, que possibilita o uso da biblioteca RVC3-MATLAB [2], a qual fornece funções para cálculos diretos relacionados ao comportamento cinemático do robô manipulador. Esta biblioteca foi essencial para a análise e modelagem da posição e orientação do efetuador, considerando os ângulos das juntas e os comprimentos dos elos em relação à base do manipulador robótico.

A cinemática é o estudo dos movimentos dos robôs, sem considerar as causas que lhe dão origem, envolvendo apenas as variáveis de posição, velocidade, aceleração e derivadas. O principal objetivo da cinemática é determinar a posição e orientação do robô em um espaço tridimensional. Para isso, são desenvolvidos os modelos de cinemática direta e inversa.

A implementação requer o uso de matrizes de transformações homogêneas, que permitem relacionar os ângulos das juntas do robô às coordenadas do efetuador.

Para determinar a posição do efetuador, representa-se pela seguinte equação:

$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \quad (3)$$

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2), \quad (4)$$

onde:

- $L_1$  e  $L_2$  representam os comprimentos dos elos;
- $\theta_1$  e  $\theta_2$  representam os ângulos das juntas;

Por outro lado, para determinar os ângulos das juntas a partir de uma posição desejada do efetuador, utiliza-se a seguinte equação baseada na tangente inversa:

$$\theta = \tan^{-1} \left( \frac{y}{x} \right). \quad (5)$$

Por sua vez, para calcular o valor do ângulo a partir de uma posição que se deseja, é necessário calcular a partir da tangente inversa (5)

$$\theta = \tan^{-1} \left( \frac{y}{x} \right) \quad (6)$$

Nesta seção, será apresentada a implementação detalhada dos modelos de cinemática direta e inversa, assim como a comunicação serial estabelecida entre o MATLAB e o Arduino, utilizada para o controle do braço robótico. Essa comunicação permite que o sistema receba comandos e envie sinais de controle para os motores, ajustando os ângulos das juntas e, conseqüentemente, a posição do efetuador no espaço.

##### A. Cinemática Direta

Na cinemática direta, a partir dos ângulos das juntas, determina-se a posição e a orientação da braço robótico no espaço cartesiano em relação à base. Esse cálculo é fundamental para o controle de manipuladores, pois permite prever a posição final do efetuador ao executar uma tarefa em específico.

A cinemática direta, como dito anteriormente, relaciona os ângulos das juntas às coordenadas do efetuador. Assim, por meio de matrizes de transformação homogênea, calculadas de modo direto através da biblioteca RVC3-MATLAB, que descrevem as rotações e translações associadas a cada elo do manipulador, resultam na posição final.

##### B. Cinemática Inversa

Na cinemática inversa, a partir da posição final e orientação desejada, encontra-se os valores dos ângulos das juntas. Esse processo é importante quando se trata de aplicações onde o objetivo do robô é alcançar um ponto específico no espaço, como em tarefas de montagem, soldagem ou manipulação de objetos.

Ao contrário da cinemática direta, a cinemática inversa apresenta maior complexidade, especialmente para robôs manipuladores com muitos graus de liberdade. Isso ocorre porque o problema inverso pode levar a:

- Múltiplas soluções: Em muitos casos, existem várias configurações articulares que levam a um mesmo ponto no espaço.
- Ausência de solução: O ponto desejado pode estar fora do espaço de trabalho do robô.
- Ambiguidade ou singularidades: Algumas posições levam a resultados indefinidos ou não únicos.

Esses ângulos serão enviados ao Arduino para acionar os motores do manipulador, posicionando o efetuador na coordenada correta.

##### C. Comunicação Serial

No que se refere à comunicação serial entre o Arduino e o MATLAB, o Arduino é responsável pelo controle dos movimentos das juntas do robô, enquanto o MATLAB realiza os cálculos para determinar os ângulos de rotação. Dessa forma, os servomotores, inicialmente acionados pelo Arduino, aguardam o envio dos ângulos por meio da comunicação serial. Assim que os valores são recebidos, o Arduino movimenta o braço robótico para a posição correspondente. O MATLAB, por sua vez, calcula os ângulos por meio da cinemática inversa, e as posições por meio da cinemática direta.

#### V. RESULTADOS E CONCLUSÃO

Para descrever a posição de um manipulador com quatro graus de liberdade no espaço, é necessário definir quatro variáveis. Neste caso, foram escolhidas as coordenadas  $(x, y, z)$  para representar a posição linear do manipulador, além de três ângulos das juntas, sendo a garra fixa. Além disso, os tamanhos dos elos do braço robótico são considerados para a modelagem cinemática.

A posição linear do braço é expressa:

- $x$  representa a coordenada em relação ao eixo  $x$  do frame da base.
- $y$  representa a coordenada em relação ao eixo  $y$  do frame da base.
- $z$  representa a coordenada em relação ao eixo  $z$  do frame da base.



Os ângulos das juntas determinam os movimentos do braço robótico:

- $\theta_1$  define a rotação da junta da base.
- $\theta_2$  define a rotação da junta 1.
- $\theta_3$  define a rotação da junta 2.

Já os elos (links) do manipulador possuem os tamanhos:

- $L_1 = 7$  cm corresponde à distância entre a base e o primeiro elo (Elo principal).
- $L_2 = 9$  cm representa o comprimento do elo principal.
- $L_3 = 9$  cm define a distância entre o segundo elo e a garra.

Com essas variáveis, foi possível determinar a posição e a orientação do manipulador no espaço, permitindo um controle preciso da cinemática, conforme ilustrado na Figura 5. Ao receber os ângulos calculados, o braço robótico físico executou os movimentos de forma síncrona, resultando em uma trajetória contínua (Figura 6).

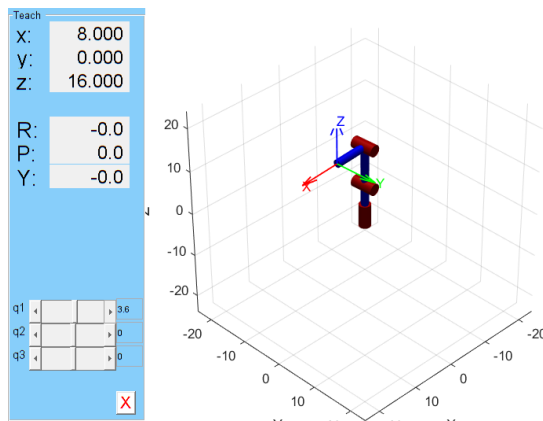


Fig. 5. Esboço do braço robótico. Fonte: MATLAB (2025)

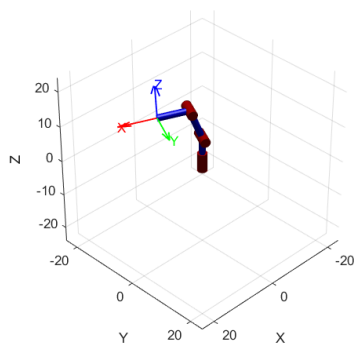


Fig. 6. Braço robótico realizando a trajetória contínua. Fonte: MATLAB (2025)

Em suma, este trabalho desenvolveu um protótipo de um braço robótico com quatro graus de liberdade, totalmente impresso em PLA. Inicialmente, o controle foi realizado por meio do microcontrolador Arduino Uno com o uso de joysticks

e, posteriormente, o controle foi realizado via comunicação serial entre o MATLAB e o Arduino Uno. O MATLAB foi responsável pela implementação da cinemática direta e inversa, possibilitando a execução precisa dos movimentos e até mesmo a geração de trajetórias.

## REFERENCES

- [1] Arnov Sharma. "MicroBot V1 ESP32 Robot (2020)". Disponível em: [https://www.hackster.io/Arnov\\_Sharma\\_makes/microbot-v1-esp32-robot-0c05d3](https://www.hackster.io/Arnov_Sharma_makes/microbot-v1-esp32-robot-0c05d3).
- [2] CORKE, Peter; JACHIMCZYK, Witold; PILLAT, Remo. Robotics, Vision and Control: Fundamental Algorithms in MATLAB®. 3. ed. atual. [S. l.]: Springer, 2023. v. 147. ISBN 978-3-031-07261-1. DOI <https://doi.org/10.1007/978-3-031-07262-8>. Disponível em: <https://link.springer.com/book/10.1007/978-3-031-07262-8>.
- [3] Lorena Teixeira. "PROJETO DE UM BRAÇO ROBÓTICO PARA AUTOMAÇÃO DE BANCADA EXPERIMENTAL". UBERLÂNDIA - MG: UNIVERSIDADE FEDERAL DE UBERLÂNDIA FACULDADE DE ENGENHARIA MECÂNICA, 2022. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/35048/4/ProjetoBra>
- [4] RECK, Miguel Alberto Xavier; ANTONELLO, Ricardo. Projeto para um Braço Robótico de Baixo Custo Baseado em Open Source. DOI: <https://doi.org/10.21166/cpitt.v3i1.2915>.
- [5] Thingiverse. Disponível em: <https://www.instructables.com/EEZYbotARM/>
- [6] UNIVERSAL ROBOTS BRASIL. O que são garras robóticas?. Universal Robots A/S, 18 nov. 2020. Disponível em: <https://www.universal-robots.com/br/blog/o-que-s>
- [7] 3DPRINTED robotic arm. © 2025 Autodesk, Inc. Disponível em: <https://www.instructables.com/EEZYbotARM/>.