

Relatório de prática da disciplina de Robótica – 2025.1

João Gustavo Cavalcanti Beltrão da Silva^{1,2}  orcid.org/0000-0001-8392-3115

Davi Viana Gouveia ^{1,2}  <https://orcid.org/0009-0008-8432-8391>

José Lucas Bessa de Oliveira ^{1,2}

¹ Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil,

² Graduação em Engenharia de Controle e Automação, Escola Politécnica de Pernambuco, Pernambuco, Brasil,

Git Hub do projeto: <https://github.com/jgc77/Manipulador-2EE>

Resumo

Este artigo apresenta o desenvolvimento de um manipulador robótico com três graus de liberdade, configurado com três juntas rotacionais (RRR), de configuração antropomórfica e controlado por meio de um Arduino. O sistema foi estruturado com uma arquitetura modular, separando os componentes de hardware dos cálculos de cinemática, o que facilitou a organização e o desenvolvimento do projeto. Foram realizadas medições dos elos do robô para permitir a aplicação correta dos cálculos de cinemática direta e inversa, além da montagem mecânica das juntas e elos utilizando peças impressas em 3D. O controle dos motores de passo foi feito com drivers dedicados, sendo que todos dos motores (28BYJ-48) foram modificados para funcionar como motor bipolar, ampliando sua precisão e controle. Para organizar e documentar o projeto, foi criado um repositório no GitHub, e bibliotecas específicas, como ControlMotor e Cinematic, foram desenvolvidas para facilitar o controle e a reutilização do código.

Palavras-Chave: manipulador robótico, juntas rotacionais, Arduino, cinemática direta e inversa, motores de passo.

Abstract

This article presents the development of a robotic manipulator with three degrees of freedom, configured with three rotational joints (RRR) and controlled using an Arduino platform. The system was structured with a modular architecture, separating hardware components from kinematic calculations, which facilitated the organization and development of the project. Measurements of the robot's links were carried out to enable the correct application of forward and inverse kinematics calculations, along with the complete mechanical assembly of the joints and links using 3D-printed parts. Stepper motors were controlled using dedicated drivers, and one of the motors (28BYJ-48) was modified to operate as a bipolar motor, improving its precision and control. A GitHub repository was created to organize and document the project, and specific libraries, such as main and controlMotor, were developed to simplify control and enable code reuse.

Key-words: robotic manipulator, rotational joints, Arduino, forward and inverse kinematics, stepper motors.

1 Introdução

Os manipuladores robóticos são sistemas mecânicos desenvolvidos para reproduzir movimentos semelhantes muitas vezes de um braço humano, sendo amplamente utilizados em processos automatizados que exigem precisão e repetibilidade. Um tipo comum de manipulador é o que possui três graus de liberdade com juntas rotacionais (RRR), de configuração antropomórfica, permitindo que o sistema realize movimentos em diferentes direções por meio da rotação de seus elos. Esse tipo de configuração é especialmente útil em aplicações que exigem movimentação em planos tridimensionais, como em linhas de montagem e tarefas de manipulação leve.

Para projetar e controlar corretamente um manipulador (RRR), de configuração antropomórfica é essencial aplicar os conceitos de cinemática direta e inversa. A cinemática direta permite determinar a posição final do atuador (ou ferramenta) a partir dos ângulos das juntas, enquanto a cinemática inversa realiza o processo inverso: calcula os ângulos necessários das juntas para alcançar uma determinada posição no espaço. Para isso, foram feitas medições precisas dos elos do robô, servindo como base para os cálculos matemáticos.

Durante o desenvolvimento do projeto, foram utilizadas técnicas de programação no Arduino para integrar os motores de passo ao controle das juntas, além do uso de bibliotecas específicas para organizar o código e facilitar a modularização do sistema.

2 Materiais e Métodos

Para a realização da prática referente ao manipulador robótico com três graus de liberdade com juntas rotacionais (RRR), de configuração antropomórfica foi utilizado os seguintes componentes:

- 1x Arduino Uno
- 3x Drive a4988
- 3x Capacitor Eletrolítico 100uf 50v
- 1x Fonte 12v 2a
- 1x Protoboard
- 1x Conector P4 Fêmea
- 1x Conector p4 Macho
- 1x Chave Alavanca

- 1x Chave Gangorra 2 Terminais
- 1x Kit Jumpers Macho-Fêmea
- 3x Motores de Passo 28BYJ-48
- 3x Elos Impressos em 3d

2.1 Análise Teórica

Para compreender o comportamento do manipulador robótico de três graus de liberdade (configuração RRR e antropomórfica), é essencial realizar uma análise teórica com base nos princípios da cinemática. Essa análise permite descrever, de forma matemática, o movimento do sistema e prever a posição do efetuador final (*end-effector*) a partir dos ângulos das juntas, ou vice-versa.

A cinemática é dividida em duas etapas principais:

- Cinemática Direta
- Cinemática Inversa

Cinemática Direta:

Na cinemática direta, o objetivo é determinar a posição final do efetuador (x, y, z) a partir dos ângulos das juntas ($\theta_1, \theta_2, \theta_3$) e dos comprimentos dos elos (L_1, L_2, L_3). Essa etapa é importante para saber onde o braço estará no espaço depois de receber um determinado conjunto de comandos.

O processo seguido foi o seguinte:

- **Definição dos Parâmetros D-H (Denavit-Hartenberg):**

O manipulador foi modelado na posição inicial com elos dispostos totalmente verticalmente a partir da base.

- **Montagem das Matrizes de Transformação:**

Com os parâmetros D-H definidos, foram calculadas as matrizes de transformação entre os elos, respeitando as convenções da modelagem.

- **Implementação no Código**

A equação final da cadeia de transformações foi inserida diretamente no código.

As entradas são os ângulos das juntas ($\theta_1, \theta_2, \theta_3$) e a saída é a posição cartesiana (x, y, z) do efetuador.

$$\theta_3 = \arccos(\text{resultado acima})$$

Cinemática Inversa:

A cinemática inversa faz o caminho contrário: a partir de uma posição desejada do efetuador (x, y, z), calcula-se os ângulos que cada junta deve assumir para atingir esse ponto. Esse cálculo é mais complexo, pois:

- Pode haver mais de uma solução possível.
- Em alguns casos, a solução pode ser inexistente, devido às limitações físicas do robô.

Lógica da Cinemática Inversa:

Dado o ponto (x, y, z), os objetivos são:

θ_1 : ângulo da base

θ_2 : ângulo do ombro

θ_3 : ângulo do cotovelo

Cálculo de θ_1 (ângulo da base):

Este é o ângulo mais simples de determinar. Basta utilizar a função arco-tangente para encontrar o ângulo entre o eixo X e o ponto projetado no plano XY:

$$\theta_1 = \arctan\left(\frac{y}{x}\right)$$

Redução para manipulador plano:

Com θ_1 determinado, o problema tridimensional é reduzido a um plano bidimensional (plano R-R), considerando apenas os comprimentos dos elos e a altura. Define-se dois valores auxiliares:

r: distância horizontal no plano XY

$$r = \sqrt{x^2 + y^2}$$

s: altura relativa ao elo L_1

$$s = z - L_1$$

Cálculo de θ_3 (ângulo do cotovelo):

Utiliza-se a Lei dos Cossenos no triângulo formado pelos elos L_2 , L_3 e a linha direta entre o ombro e o ponto final (r, s):

$$\cos(\theta_3) = \frac{r^2 + s^2 - L_2^2 - L_3^2}{2L_2L_3}$$

O valor de θ_3 é obtido com:

Observação: É adotada a convenção do cotovelo voltado para cima (solução "cotovelo para cima"), portanto escolhe-se a solução com o maior valor possível de θ_3 (ângulo interno obtuso).

Cálculo de θ_2 (ângulo do ombro):

Para encontrar θ_2 , utiliza-se dois ângulos auxiliares:

β (beta): ângulo interno entre os elos L_2 e L_3

$$\beta = \arctan\left(\frac{L_3 \times \sin(\theta_3)}{L_2 + L_3 \times \cos(\theta_3)}\right)$$

γ (gama): ângulo formado entre a linha do ombro ao ponto (r, s) e o eixo horizontal

$$\gamma = \arctan\left(\frac{s}{r}\right)$$

Com isso:

$$\theta_2 = \gamma - \beta$$

E para viabilizar a implementação prática dessa lógica no sistema embarcado, foram desenvolvidas duas bibliotecas principais: ControlMotor e Cinematic.

ControlMotor, essa biblioteca é responsável por:

- Integrar a biblioteca AccelStepper.h com os drivers A4988;
- Automatizar toda a configuração inicial dos motores;
- Definir os pinos de controle e inicializar os sinais de ativação (nível lógico LOW);
- Implementar a lógica da chave seletora com controle de bouncing;
- Converter os ângulos obtidos da cinemática para valores discretos de passos de motor;
- Implementar os limites físicos do manipulador e o modo de calibração manual.

Cinematic, essa biblioteca é responsável por:

- Concentrar todas as funções relacionadas à Cinemática Direta e Inversa;
- Organizar os cálculos que transformam ângulos das juntas em posições no espaço e vice-versa;
- Utilizar as funções matemáticas da biblioteca math.h para realizar operações trigonométricas e geométricas;

2.2 Simulação

Para validar o funcionamento do manipulador robótico RRR antes da montagem final, foi desenvolvida uma representação teórica e visual do sistema. Essa representação buscou simular o comportamento do robô com base nos componentes reais utilizados.

Na simulação, consideramos o uso de três motores de passo 28BYJ-48, que foram modificados para funcionar no modo bipolar. Cada motor foi conectado a um driver A4988, que realiza o controle de corrente e direção com maior estabilidade. A comunicação e o controle dos motores foram gerenciados por meio de uma placa Arduino Uno, que executa os códigos implementados nas bibliotecas Cinematic e ControlMotor que é integrado pelo código principal main.

Abaixo, apresentamos a representação teórica criada para ilustrar o funcionamento do sistema com os três motores conectados ao Arduino e aos drivers:

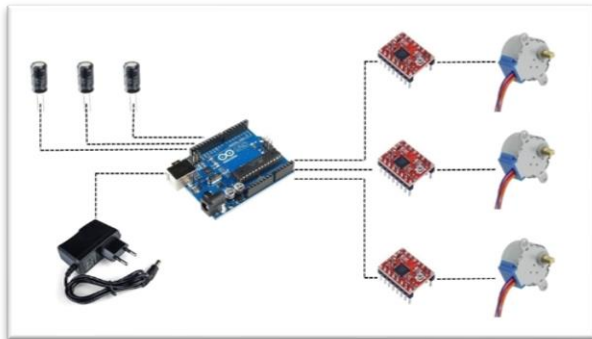


Figura 01: Representação da estrutura do manipulador robótico.

Além da representação teórica, também foi elaborado um diagrama elétrico completo do sistema, utilizando o software fritzing de simulação para projetos com Arduino. Esse diagrama detalha todas as conexões entre os componentes, incluindo os pinos digitais do Arduino, os sinais de controle dos A4988, a alimentação dos motores e os demais elementos que compõem a arquitetura do projeto.

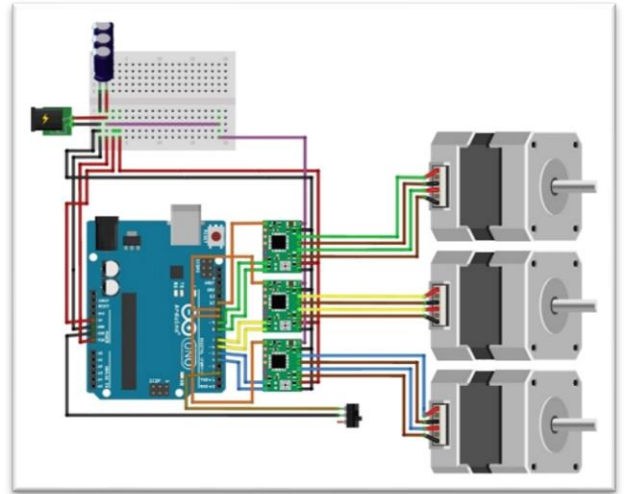


Figura 02: Representação do diagrama elétrico do manipulador robótico.

3 Resultados

Durante os testes práticos, o manipulador foi colocado em diferentes posições de forma controlada, demonstrando a eficácia do sistema tanto na parte mecânica quanto na lógica de movimentação. As posições foram calculadas com base na representação teórica e executadas com boa estabilidade, respeitando os limites físicos definidos no código.

A seguir, apresentamos imagens da montagem final do manipulador, bem como exemplos das posições alcançadas pelo braço robótico durante a execução dos testes:



Figura 03: Manipulador na posição inicial dos nossos cálculos (0, 0, 0).

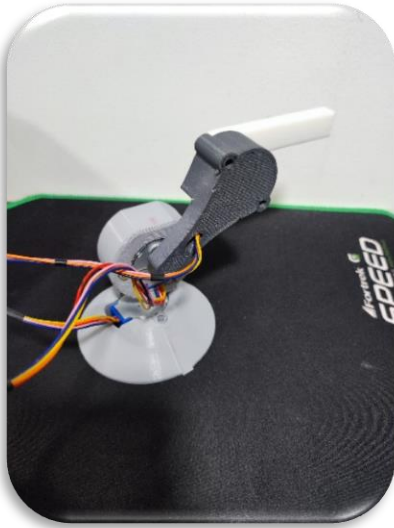


Figura 04: Manipulador na posição (90, -45, -45).

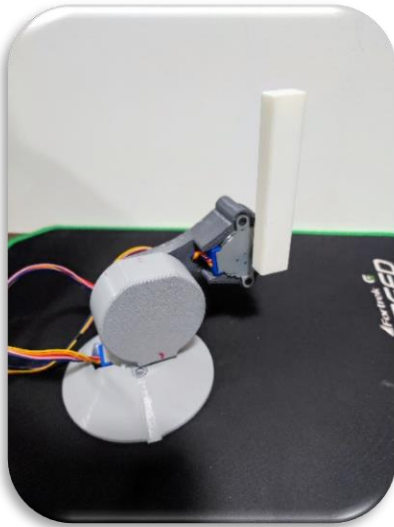


Figura 05: Manipulador na posição (-130, 90, -80).

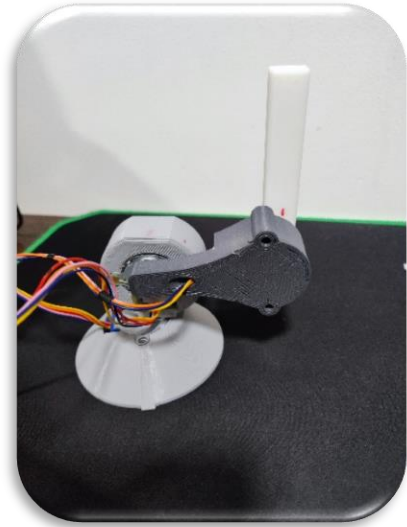


Figura 06: Manipulador na posição (130, -80, 90).

4 Discussão

Após realizar uma análise comparativa dos resultados teóricos, práticos e simulados, constatou-se que a organização modular do software separando leitura de comandos, lógica de controle e acionamento de motores foi fundamental para o bom desempenho do sistema. Implementar a cinemática direta foi relativamente rápido: bastou enviar os valores dos ângulos direto para os motores e verificar se a posição calculada condizia com a observada.

Já a cinemática inversa tem sido mais trabalhosa. Converter um alvo cartesiano em três ângulos viáveis, sem infringir limites mecânicos ou gerar soluções ambíguas, exigiu um refinamento do modelo matemático e tratamento cuidadoso de singularidades.

A etapa de simulação mostrou-se essencial para evitar erros de fiação ou escolha de pinos. Mesmo assim, alguns ajustes finos como correntes dos A4988 e compensação de jogo mecânico foram feitos apenas após os primeiros testes reais.

Em resumo, o projeto atingiu os objetivos iniciais: montar um manipulador funcional, validar a cinemática direta e criar a base para a cinemática inversa.

5. Conclusões

Neste artigo, discutimos o funcionamento do manipulador robótico com três graus de liberdade. O mesmo foi finalizado com sucesso, cumprindo os objetivos propostos desde a fase de planejamento até os testes práticos. Consequimos montar toda a estrutura física, implementar o controle dos motores via Arduino e aplicar tanto a cinemática direta quanto a inversa de forma funcional. Apesar dos desafios, especialmente na parte da cinemática inversa e na parte do controle de qualidade do Drivers A4988 o sistema se mostrou preciso.

A separação do código em bibliotecas facilitou bastante a organização do projeto e a realização de ajustes durante os testes. Além disso, o uso da simulação e do diagrama elétrico ajudou a montagem na prática. No final, o manipulador conseguiu executar movimentos programados de forma controlada, validando toda a lógica construída ao longo do processo.

Referências

[1] ARDUFOCUS. Como modificar o motor de passo 28BYJ-48 para bipolar. ArduFocus, [s.d.]. Disponível em: <https://ardufocus.com/howto/28byj-48-bipolar-hw-mod/>. Acesso em: 7 jul. 2025.

[2] ELETROGATE. Tudo sobre DRIVER A4988 e motor de passo usando o Arduino. Blog Eletrogate, 29 mar. 2018 (atualizado em 9 set. 2022). Disponível em: <https://blog.eletrogate.com/driver-a4988-motor-de-passo-usando-o-arduino/>. Acesso em: 7 jul. 2025.

[3] AIRSPAYCE. AccelStepper Class Reference. [S.l.]: Airspayce, [s.d.]. Disponível em: <https://www.airspayce.com/mikem/arduino/AccelStepper/classAccelStepper.html>. Acesso em: 7 jul. 2025.