CENTRO UNIVERSITÁRIO DE VIÇOSA – UNIVIÇOSA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

DANIEL OLIVEIRA TAVARES

Sistema de manipulação robótica baseado em visão computacional

VIÇOSA - MG
2025

DANIEL OLIVEIRA TAVARES

Sistema de manipulação robótica baseado em visão computacional

Projeto apresentado ao Centro Universitário de Viçosa – Univiçosa como parte dos requisitos para a conclusão da disciplina CCE001 – Trabalho de Conclusão de Curso 1.

Orientador: Prof. Me. Vinicius Martins Almeida

VIÇOSA-MG

2025

RESUMO

Este trabalho propõe o desenvolvimento de um sistema de manipulação robótica de baixo custo, utilizando visão computacional como principal fonte de realimentação sensorial. O sistema será composto por um manipulador robótico de cinco graus de liberdade, projetado com estrutura impressa em 3D e utilizando motores de passo e um servo motor, controlados por drivers dedicados. A arquitetura do sistema será baseada em um Raspberry Pi, que executará algoritmos de controle em malha aberta e fechada, incluindo PID e LQR. Para a percepção do ambiente, será utilizada uma câmera USB e a biblioteca OpenCV para realizar a segmentação de objetos por cor, detecção de contornos, formas e estimativa de pose. O sistema de visão fornecerá as coordenadas dos objetos como entrada para os algoritmos de controle, que farão a manipulação autônoma dos itens detectados por meio de uma tarefa *pick and place* (pegar e posicionar). A avaliação do desempenho será realizada em ambiente simulado e físico, considerando métricas como precisão, tempo de resposta e repetibilidade. A solução desenvolvida será de código aberto e visa contribuir para a democratização do acesso à robótica e ao estudo da integração entre percepção e ação em sistemas autônomos.

Palavras-chave: Robótica. Visão computacional. Automação. Controle.

ABSTRACT

This work proposes the development of a low-cost robotic manipulation system, using computer vision as the main source of sensory feedback. The system will consist of a five-degree-of-freedom robotic manipulator, designed with a 3D printed structure and using stepper motors and a servo motor, controlled by dedicated drivers. The system architecture will be based on a Raspberry Pi, which will execute open-loop and closed-loop control algorithms, including PID and LQR. For environment perception, a USB camera and the OpenCV library will be used to perform color-based object segmentation, contour detection, shape recognition, and pose estimation. The vision system will provide object coordinates as input to the control algorithms, which will perform autonomous manipulation of detected items through pick-and-place tasks. Performance evaluation will be conducted in both simulated and physical environments, considering metrics such as accuracy, response time, and repeatability. The developed solution will be open-source and aims to contribute to democratizing access to robotics and the study of perception-action integration in autonomous systems.

Keywords: Robotics. Computer vision. Automation. Control.

SUMÁRIO

TABELA DE FIGURAS

LISTA DE ABREVIATURAS

3D	Três Dimensões
ARE	Algebric Riccati Equation
CAD	Computer-aided Design
DH	Denavit-Hartenberg
DOF	Degrees of Freedom
FDM	Fused Deposition Modeling
LQR	Linear Quadratic Regulator
mm	Milímetro
PCB	Printed Circuit Board
PID	Proporcional-Integral-Derivativo
PLA	Ácido polilático
PWM	Pulse Width Modulation
ROS	Robot Operating System
SBC	Single Board Computer
STL	Stereolithography
URDF	Unified Robot Description Format
XACRO	XML Macros

1 INTRODUÇÃO

Robôs manipuladores interagem fisicamente com o ambiente e, dessa forma, proporcionam uma interface homem-máquina. Tarefas antes consideradas perigosas e exaustivas para os seres humanos podem agora ser executadas por sistemas inteligentes de forma mais segura e em uma fração de tempo e custo (Torres, 2022). A área de manipulação robótica, que estuda como robôs podem realizar essas interações, é muito aplicada na indústria, principalmente em linhas de produção e automação de processos repetitivos.

Historicamente, a manipulação robótica consistia em uma série de movimentos pré-definidos sem a habilidade de adaptação a mudanças do ambiente (Bllard; Kragic, 2019). Contudo, com os avanços tecnológicos do passar dos anos, tais sistemas gradualmente evoluíram. Atualmente, robôs autônomos inteligentes podem exercer inúmeras funções complexas e atuar em ambientes mais dinâmicos.

Apesar dos avanços, ainda existem muitos desafios na manipulação robótica a serem resolvidos. Cui e Trinkle (2021) destacam desafios como a criação de algoritmos de controle robustos para várias tarefas e a implementação robótica em ambientes que estão em constante mudança.

Neste contexto, a integração de sensores visuais aos sistemas robóticos passou a ser essencial para melhor desempenho das atividades, uma vez que esta fornece uma gama de informações, dentre elas identificar a posição e orientação de objetos conhecidos (Tang et al., 2020). Segundo Shahria et al. (2022), a manipulação baseada em visão ocorre quando um robô realiza ações com base em informações visuais captadas por uma ou mais câmeras, e utiliza esses dados como forma de feedback para guiar seu comportamento.

Wells et al. (1996) descrevem o controle visual robótico, dentre outras formas, como uma maneira de alcançar uma posição desejada em relação a algum objeto por meio de informações visuais extraídas de sensores. Essa técnica permite que os manipuladores sejam mais precisos e mais adaptáveis a ambientes dinâmicos.

Apesar dessas inovações, ainda existem obstáculos relacionados à precisão do reconhecimento visual e ao controle eficiente dos manipuladores, principalmente quando o sistema precisa operar de forma autônoma e adaptativa. Diante desse cenário, torna-se relevante explorar soluções eficientes para tarefas de manipulação orientadas por percepção visual.

1.1 Problema e justificativa

A manipulação robótica é utilizada em inúmeros setores industriais para automatizar processos de manufatura e logística e, com isso, garantir maior precisão e eficiência às tarefas.

Robôs manipuladores são utilizados em vários processos, como montagem, soldagem, empacotamento, paletização, inspeção de componentes, entre outros. Segundo Torres (2022), robôs realizam tarefas repetitivas que podem ser perigosas aos seres humanos de forma mais eficiente e mais segura.

No entanto, a implementação desses sistemas exige alto custo operacional, complexidade de desenvolvimento e a necessidade de um controle preciso e eficiente para realizar tarefas específicas. Além do mais, este tipo de sistema deve ser robusto e adaptável a mudanças do ambiente (Cui; Trinkle, 2021). A destreza e a capacidade de manipulação de objetos por meio dos robôs ainda são um grande desafio (Shahria et al., 2022). Pequenos erros em cálculos de trajetória ou reconhecimento de objetos podem resultar em falhas no processo.

Outro desafio é a integração de visão computacional com manipuladores robóticos. Grandes empresas utilizam sensores modernos e algoritmos avançados para garantir precisão na identificação e manipulação de objetos, enquanto a implementação desses recursos em projetos menores apresenta limitações técnicas e financeiras. Técnicas como segmentação de objetos e controle de trajetória precisam ser ajustadas para garantir funcionamento eficiente e confiável (Grassi, 2005).

Logo, existe a necessidade de tornar a manipulação robótica mais acessível e aplicável em diferentes contextos, seja na indústria ou na pesquisa acadêmica. Muitos sistemas robóticos disponíveis no mercado são de alto custo, exigem infraestrutura especializada e possuem códigos e protocolos proprietários, o que dificulta a personalização. Embora existam simulações para auxiliar na compreensão do comportamento do robô em um ambiente controlado, a experimentação em um cenário real continua sendo essencial para validar certos comportamentos (Billard; Kragic, 2019).

Com isso, o projeto poderá ser utilizado para aprofundar estudos sobre algoritmos de controle de manipuladores, robótica, processamento de imagens e estratégias de otimização de movimento por um preço acessível a instituições de ensino. Por fim, desenvolvendo uma solução de código aberto, o projeto se torna mais flexível e acessível para pesquisadores e desenvolvedores e, dessa maneira, novas funcionalidades podem ser adicionadas ou adaptadas conforme diferentes necessidades.

2 OBJETIVOS

2.1 Objetivo geral

Projetar e desenvolver um braço robótico de 5 graus de liberdade (DoF), utilizando visão computacional para identificar e classificar objetos com base na cor.

2.2 Objetivos específicos

1. Modelar e construir a estrutura mecânica e eletrônica do braço robótico de 5 graus de liberdade.
2. Utilizar ações de controle de malha aberta, malha fechada, trajetória e de posição.
3. Implementar as seguintes técnicas de visão computacional para identificar e classificar objetos: segmentação por cor, detecção de contornos e formas, estimativa de posição e orientação.
4. Integrar o sistema de visão computacional ao controle do manipulador.
5. Desenvolver um *software* interativo para desktop para monitoramento e controle do sistema.
6. Realizar testes de precisão, repetibilidade e desempenho do sistema de visão computacional e do controle do manipulador em ambiente simulado e físico.

2.3 Resultados esperados

Espera-se ao final deste trabalho, ter um sistema robótico funcional que seja capaz de identificar e manipular objetos com base em suas características físicas, e de forma totalmente autônoma. O sistema irá capturar imagens dos objetos através de uma câmera integrada, processar essas informações com algoritmos de visão computacional, calcular as ações de controle e realizar a manipulação sem a necessidade de intervenção humana. O manipulador deverá executar a tarefa de *pick and place* com os blocos, depositando-os em seus respectivos compartimentos de acordo com a cor que possuem.

Além disso, espera-se também aplicar diferentes ações de controle no robô para identificar aquela mais eficaz para a tarefa em questão. Técnicas de visão computacional serão empregadas ao manipulador para que a obtenção das imagens pelo sensor da câmera seja a mais precisa possível. Desta forma, será possível validar na prática diferentes estratégias de controle e analisar entre elas a precisão e o tempo de execução. Por fim, acredita-se que o sistema desenvolvido contribuirá para o ensino e aprendizagem nas áreas de robótica, automação e visão computacional, e servirá como recurso pedagógico em ambientes acadêmicos.

3 REFERENCIAL TEÓRICO

3.1 Manipulação robótica e sistemas de *pick and place*

A manipulação robótica tem como objetivo permitir que robôs interajam fisicamente com o ambiente de forma autônoma. Um exemplo é a tarefa de *pick and place*, na qual o robô identifica um objeto, calcula sua posição e realiza o deslocamento até um local de destino predeterminado (Zhang; Liu, 2021). Esse tipo de sistema é utilizado em linhas de produção para movimentar peças com precisão e repetibilidade.

A execução dessa tarefa exige a integração de três conceitos: percepção sensorial, controle dos atuadores e planejamento de movimento. O uso de sensores visuais permite a localização dos objetos no ambiente e fornece feedback contínuo para o sistema se ajustar em tempo real (Oda et al., 2009).

3.2 Modelagem de manipuladores robóticos

A modelagem robótica relaciona os movimentos das juntas do robô com a posição e orientação do atuador final. A convenção de Denavit-Hartenberg (DH) é usada para representar os sistemas de coordenadas ligados a cada elo do robô (Kucuk; Bingul, 2006).

Cada elo é descrito por uma matriz de transformação homogênea T_i (3.1), que representa a relação entre os sistemas de coordenadas de dois elos consecutivos, i e $i+1$.

$$T_i = \begin{bmatrix} cos\,\theta_i & -sin\,\theta_i\,cos\,\phi_i & sin\,\theta_i\,sin\,\phi_i & L_i\,sin\,\theta_i \\ sin\,\theta_i & cos\,\theta_i\,sin\,\phi_i & -cos\,\theta_i\,sin\,\phi_i & L_i\,sin\,\theta_i \\ 0 & sin\,\phi_i & cos\,\phi_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

em que:

- θ_i: ângulo da junta;
- d_i: deslocamento da junta;
- L_i: comprimento do elo;
- ϕ_i: ângulo entre os eixos.

A cinemática direta consiste em determinar a posição do atuador final a partir dos ângulos das juntas. Para isso, é necessário multiplicar sucessivamente as matrizes Ti de todos os elos do atuador, como em (3.2):

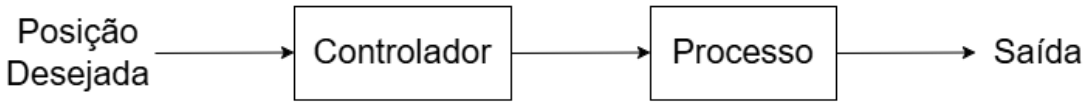
$$T = T_1 \cdot T_2 \cdot \ldots \cdot T_n \quad (3.2)$$

Já a cinemática inversa consiste no oposto: deduz quais os valores das articulações que permitem atingir uma posição desejada. Esse problema pode ser resolvido por métodos geométricos ou numéricos, e pode apresentar múltiplas soluções ou singularidades.

3.3 Sistemas de controle: malha aberta e malha fechada

Sistemas de malha aberta (Figura 1) realizam ações determinadas pelo controlador sem nenhum tipo de realimentação de sinal (Ogata, 2010). Assim, esse controlador não consegue se ajustar em tempo real diante das variações que venham ocorrer no sistema. Sua implementação é simples, contudo, não oferece correção de erros.

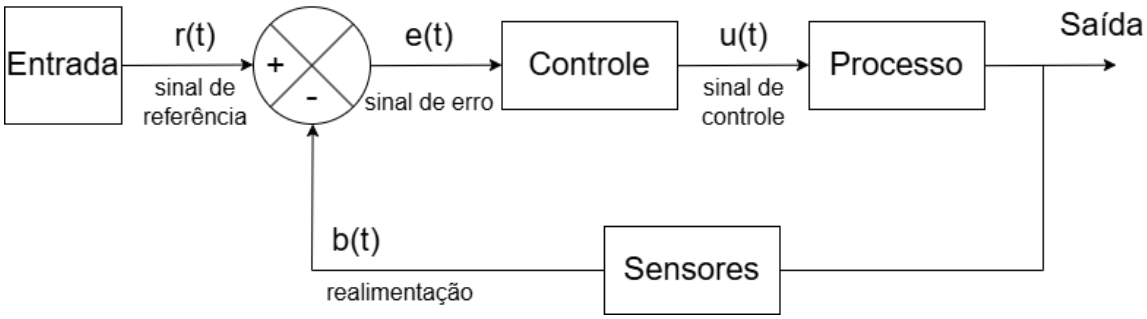
Figura 1 - Controle em malha aberta



Fonte: Autoria Própria (2025).

Já sistemas de malha fechada (Figura 2) utilizam realimentação de sinal através de sensores para comparar a saída real com a saída desejada. Dessa maneira, o controlador está sempre se ajustando para minimizar os erros, de forma automática (Ogata, 2010).

Figura 2 - Controle em malha fechada



Fonte: Autoria Própria (2025).

3.4 Controlador PID (proporcional-integral-derivativo)

Segundo Åström e Hägglund (2000), o controlador PID é um dos mais utilizados na indústria, correspondendo a mais de 90% deles e, por isso, foi escolhido para ser usado neste trabalho. Ele é composto por três termos: o proporcional (P), que reage ao erro atual; o integral (I), que considera a soma dos erros passados; e o derivativo (D), que antecipa as mudanças com base na taxa de variação do erro.

Essa estrutura permite controlar tanto a posição quanto a velocidade dos motores de forma precisa. Contudo, para garantir bom desempenho, é necessário sintonizar corretamente os parâmetros do PID, que pode ser feito por métodos como Ziegler-Nichols ou Cohen-Coon.

Em (3.3) é possível visualizar a representação matemática do PID:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (3.3)$$

em que:

- u(t) é o sinal de controle em função do tempo *t*;
- K_p é o ganho proporcional;
- K_i é o ganho integral;
- K_d é o ganho derivativo;
- e(t) é o erro.

3.5 Controlador LQR (regulador quadrático linear)

O LQR é uma técnica de controle ótimo baseada em modelo matemático do sistema. Seu objetivo é minimizar uma função de custo que penaliza o erro e o esforço de controle. Diferente do PID, o LQR utiliza a modelagem do sistema em espaço de estados, o que o torna adequado para sistemas com múltiplas variáveis (Ribeiro et al., 2019). Dessa maneira, ele foi escolhido como uma alternativa mais robusta sobre o PID.

Em (3.4) é possível visualizar a função de custo quadrática:

$$J = \int_0^\infty (x^T Q x + u^T R u)dt \quad (3.4)$$

em que:

- J é o custo total a ser minimizado;
- x é o vetor de estados dos sistema;
- u é o vetor de entradas de controle;
- Q é a matriz de pesos de estados, mede a performance do sistema;
- R é a matriz de pesos das entradas, mede o esforço do sistema.

A partir da equação anterior, é possível encontrar uma matriz de ganho que produz o menor custo para o sistema de controle considerando o equilíbrio entre precisão e esforço. A matriz de ganho ótimo K é obtida através da Equação Algébrica de Riccati (ARE – Algebric Riccati Equation), que pode ser escrita como em (3.5):

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (3.5)$$

em que:

- A é a matriz de estados;
- B é a matriz de entrada
- P é a matriz simétrica positiva definida que satisfaz a equação.

Com a matriz *P* determinada, a matriz de ganho ótimo K é calculada por (3.6):

$$K = R^{-1}B^{T}P \tag{3.6}$$

em que:

- K é a matriz de ganho ótimo do controlador LQR, que minimiza a função de custo.

A lei de controle pode então ser escrita como em (3.7):

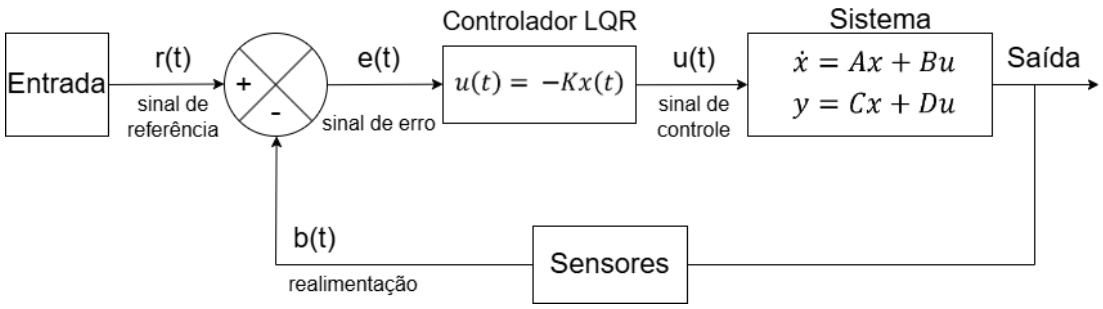
$$u(t) = -Kx(t) \tag{3.7}$$

em que:

- $u(t)$ é o vetor de entrada de controle. Exemplo: torque, tensão;
- $x(t)$ é o vetor de estados do sistema. Exemplo: posição, velocidade.

A Figura 3 apresenta o diagrama de blocos de um sistema de controle com regulador quadrático linear (LQR).

Figura 3 - Controlador LQR



Fonte: Autoria Própria (2025).

Em manipuladores robóticos, o LQR permite maior precisão e estabilidade em trabalhos dinâmicos e complexos. Além disso, pode ser combinado com observadores de estado para estimar variáveis não diretamente mensuradas.

3.6 Visão computacional na robótica

A visão computacional é a área da inteligência artificial que permite a um sistema extrair informações de imagens ou vídeos. Em robótica, é amplamente usada para detecção de objetos, segmentação por cor, estimativa de posição e rastreamento (Shahria et al., 2022).

A integração entre visão e controle permite a realização de tarefas em ambientes dinâmicos. Por exemplo, em um sistema *pick and place*, o robô utiliza a imagem capturada por uma câmera para localizar o objeto, calcular sua pose e executar a tarefa de forma autônoma.

A partir da correspondência entre coordenadas da imagem e coordenadas reais (do mundo), o sistema é capaz de calcular a pose tridimensional T_{obj} do objeto.

A pose pode então ser representada por uma matriz de transformação homogênea (3.8):

$$T_{obj} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \tag{3.8}$$

onde:

- $R \in \mathbb{R}^{3x3}$: matriz de rotação do objeto;
- $t \in \mathbb{R}^3$: vetor de translação do objeto em relação à câmera.

Para calcular a pose, métodos como PnP (*Perspective-n-Point*) são usados em conjunto com a calibração da câmera (Hartley; Zisserman, 2017), cuja equação fundamental é (3.9):

$$s \cdot P_{imagem} = K \cdot [R \mid t] \cdot P_{mundo} \tag{3.9}$$

em que:

- P_{imagem}: ponto na imagem (em pixels);
- K: matriz inerente à câmera;
- P_{mundo}: ponto 3D no espaço real;
- s: fator de escala.

A biblioteca OpenCV tem suporte direto para estas operações, como a estimação de pose pela função "solvePnP" (Bradski, 2000). A biblioteca facilita também a calibração de câmeras, detecção de contornos, e segmentação por cor.

3.7 Simulação com Gazebo

O Gazebo é uma plataforma de simulação 3D que permite a criação de ambientes virtuais realistas para testar robôs, sensores e algoritmos antes da implementação física. Em projetos de manipulação robótica, permite validar movimentos, detectar colisões e ajustar parâmetros de controle (Zhang; Liu, 2021).

Para realizar a simulação, o robô será descrito utilizando o formato URDF, sigla do inglês para "formato de descrição unificada do robô", que define as propriedades físicas de cada elo e junta do manipulador. Como alternativa, é possível utilizar uma extensão do URDF chamado XACRO (XML Macros), que permite modularidade e também reutilizar códigos.

A utilização de simulação reduz o risco de falhas, acelera o desenvolvimento e permite testes repetitivos sem desgastar o hardware. Além disso, possibilita integração com o ROS (Robot Operating System), aumentando os experimentos possíveis a serem realizados.

4 TRABALHOS RELACIONADOS

Billard e Kragic (2019) fizeram uma análise dos principais desafios enfrentados na manipulação robótica moderna. A pesquisa publicada na revista *Science* aponta a crescente necessidade de robôs manipuladores realizarem tarefas em ambientes não estruturados e com objetos desconhecidos. Ademais, explica a dificuldade de criar robôs com destreza suficiente para manipular inúmeros tipos de objetos. Manipuladores feitos de materiais flexíveis e sensores de contato associados com visão computacional são exemplos de tentativas de aperfeiçoamento que vêm sendo feitas nos robôs com o intuito de melhorar suas performances. O presente trabalho compartilha de mesmo objetivo: possibilitar que um robô realize tarefas de manipulação com objetos diversos. Para isso, será utilizada visão computacional integrada ao controle do manipulador.

No contexto de modelagem de robótica, Kucuk e Bingul (2006) fornecem uma descrição detalhada sobre o processo de modelagem matemática de um manipulador robótico abordando a cinemática direta e cinemática inversa com o método de Denavit-Hartenberg (DH). A proposta guarda semelhança com este projeto quanto à utilização dessa convenção para associar sistemas de coordenadas às articulações e calcular a pose final do atuador. Os autores tratam a cinemática direta como um problema direto e sempre solucionável, enquanto a cinemática inversa é um problema "computacionalmente complexo". O artigo explora duas maneiras de resolver a cinemática inversa: a geométrica, mais adequada para manipuladores com baixa complexidade, e a algébrica, para robôs com configurações mais elaboradas. São discutidas ainda as limitações dos métodos analíticos, como singularidades e múltiplas soluções.

Ainda sobre modelagem, Oliveira (2022) trata de um estudo de caso sobre a modelagem de um robô industrial IRB 8700. Ele aborda características físicas gerais de manipuladores industriais juntamente com conceitos de robótica. Ele calcula a cinemática direta usando a convenção de Denavit. Em seguida, a cinemática inversa de posição utilizando o método geométrico e a cinemática inversa de orientação utilizando resultados anteriores. Enquanto o sistema analisado por Oliveira é de porte industrial, este trabalho adota princípios semelhantes de modelagem em um contexto didático.

Sobre técnicas de controle aplicadas a sistemas físicos, Ribeiro et al. (2019) propuseram aplicar um controlador LQR em uma planta didática do tipo Twin Rotor MIMO System (TRMS) juntamente com o MATLAB e a plataforma Arduino. O estudo mostra a modelagem do sistema, considerando sua natureza de múltiplas entradas e múltiplas saídas (MIMO) e os desafios associados a esta abordagem. O controlador foi projetado através da função DQLR no

MATLAB e validado por simulações e testes físicos com a planta. Os autores também implementaram um observador de estados para corrigir falhas ao longo do tempo e, com isso, melhorar a precisão do controle. Como resultado, o sistema alcançou os setpoints desejados e manteve a estabilidade, comprovando a eficiência do LQR em determinados circuitos. Logo, a proposta dos autores guarda semelhança com este projeto quanto à exploração do potencial do LQR em sistemas didáticos, embora com arquiteturas diferentes.

No campo da visão computacional, Oda et al. (2009) apresentam como o controle de movimento baseado em visão aumenta a robustez de sistemas robóticos em ambientes dinâmicos. Os autores destacaram duas abordagens: o Image-Based Visual Servoing (IBVS) e o Position-Based Visual Servoing (PBVS) e mostraram vantagens e limitações destas. Além disso, este trabalho apresenta experimentos com robôs móveis usando visão computacional com feedback para rastrear objetos em movimento. Embora o presente trabalho não implemente servo visual, também se fundamenta na aquisição de dados visuais para a tomada de decisão.

De forma similar, Martinez-Martin, e Del Pobil (2019) apresentam um artigo denominado “Vision for Robust Robot Manipulation” do qual destacam como combinar informações visuais para detectar pontos de contato entre o manipulador robótico e o objeto manipulado. O trabalho combina informações de cor e de profundidade provenientes de câmeras RGB-D que, após processos de segmentação das imagens obtidas, é possível identificar falhas no processo de manipulação de objetos em ambientes não estruturados. Ainda que neste trabalho será empregada câmera RGB convencional e não RGB-D, o princípio de extrair dados a partir do visual para identificar e localizar objetos no ambiente é o mesmo.

Ainda no contexto de visão computacional, Jiang et al. (2014) propuseram um sistema integrado baseado em visão para controle semi-automático de manipuladores robóticos montados em cadeiras de rodas, voltado especialmente para indivíduos com limitações de locomoção. O sistema combina três módulos principais: reconhecimento de gestos com as mãos, detecção facial e reconhecimento de objetos, utilizando duas câmeras Kinect e um manipulador JACO. A abordagem conseguiu reduzir o tempo de execução de tarefas cotidianas ao empregar reconhecimento automático para posicionamento do braço, seguido de controle por gestos para o ajuste fino. O sistema demonstrou que mesmo com hardware comercial acessível é possível atingir alto desempenho em aplicações assistivas ao integrar visão computacional e controle.

Explorando ainda mais a integração de visão e manipulação, a pesquisa de Faria et al. (2015) traz uma ideia da integração entre visão computacional e controle em manipuladores

robóticos. O trabalho apresenta uma abordagem para a manipulação autônoma de válvulas utilizando um robô de dois braços juntamente com sensores visuais. O trabalho sugere o uso de LEDs como marcadores visuais associados às válvulas para permitir a identificação da válvula-alvo e, com isso, estimar sua posição e orientação. Com esta estimativa, é possível alinhar o atuador final do robô com o plano da válvula e realizar a manipulação. O artigo destaca também a modularização do processo em etapas como busca visual, aproximação, alinhamento, correção e manipulação. De modo análogo, o sistema aqui desenvolvido estrutura o processo de manipulação com base na percepção visual ao utilizar segmentação de imagem para identificar objetos no campo de visão e gerar comandos de posicionamento.

Ao se tratar de *softwares* de controle, Li et al. (2022) desenvolveram um sistema de controle para manipuladores robóticos de bancada integrando informações visuais e sonoras. Eles desenvolveram uma interface gráfica para ser executada em Raspberry Pi, responsável pelo controle direto dos motores, reconhecimento de voz e visualização do estado do sistema, tudo em tempo real. O *software* foi estruturado em três camadas: uma camada física (atuadores e sensores), uma camada de drivers e uma camada de aplicação, responsável pelo processamento de comandos e interação com o usuário. A plataforma permite que o manipulador seja treinado para executar movimentos específicos sem a necessidade de reprogramação, integrando comandos por voz e processamento de imagem. Assim, embora o sistema proposto em Li et al. integre o uso de informações sonoras, ele guarda semelhança com este projeto ao desenvolver um *software* para controle e visualização do estado do manipulador.

Por fim, considerando a necessidade de testes em ambiente simulado, Zhang e Liu (2021) propuseram uma metodologia de testes para manipuladores robóticos de 7 graus de liberdade utilizando o simulador Gazebo juntamente com ROS. O estudo envolveu a execução de tarefas de pick-and-place e testes de robustez, onde forças externas foram aplicadas ao robô para avaliar sua capacidade de retornar ao estado normal. Durante os experimentos, foram coletados dados de posição e torque das articulações, para uma análise da precisão e eficiência dos controladores testados. Essa abordagem mostra a importância de simular ambientes robóticos antes da implementação em hardware físico, para validar a eficiência de algoritmos de controle. A proposta guarda semelhança com este projeto quanto à utilização do Gazebo como plataforma de testes para verificar o comportamento do robô. No entanto, enquanto Zhang e Liu focam em manipuladores industriais e testes de resistência mecânica, o presente trabalho utiliza a simulação como etapa de validação em um sistema robótico didático de menor porte.

5 MATERIAL E MÉTODOS

5.1 Caracterização da pesquisa

A natureza desta pesquisa é aplicada, pois tem como foco gerar conhecimento ao solucionar problemas específicos (Nascimento, 2016). Além do mais, o trabalho é de cunho interdisciplinar, uma vez que envolve inúmeros conceitos integrados, a saber: robótica, controle, automação e visão computacional. A abordagem desta pesquisa é quantitativa, pois visa à medição e análise numérica dos fenômenos estudados (Günther, 2006) objetivando melhorar as práticas atuais.

5.2 Modelagem e estrutura mecânica

A modelagem 3D do manipulador será realizada utilizando o *software* SolidWorks e, posteriormente, a informação gerada será transferida para o *software* Ultimaker Cura para o processo de configuração das peças. Os arquivos 3D CAD gerados serão convertidos para o formato STL (estereolitografia) para posterior impressão.

As peças serão impressas utilizando uma impressora Creality Ender3v2 Ns que utiliza a tecnologia FDM para criação das camadas dos objetos. O material de impressão será o filamento termoplástico do tipo PLA, ou ácido polilático, por apresentar maior facilidade de impressão comparado com outros tipos de filamentos (Frunzaverde et al., 2023). A largura da extrusão para todas as peças será mantida no padrão de 0,4 mm e a espessura das camadas em 0,2mm. As peças serão impressas com 20% de preenchimento interno para garantir resistência estrutural ao sistema.

O manipulador projetado possuirá cinco graus de liberdade (DOF), distribuídos da seguinte forma: (i) rotação da base, movimento horizontal; (ii) elevação do braço principal, movimento vertical; (iii) flexão do antebraço, extensão; (iv) rotação do punho, orientação; e (v) abertura/fechamento da garra, preensão do objeto. Essa configuração foi escolhida por ser suficiente para executar tarefas de *pick and place* sem a complexidade adicional de um sexto grau de liberdade, o que simplifica a modelagem e o controle do sistema.

Após projetada e impressa a estrutura física do sistema, será calculado o modelo cinemático do braço robótico usando a notação de Denavit-Hartenberg (DH). A partir disso, serão calculadas as cinemáticas direta e inversa do sistema, servindo de base para os controladores futuros.

5.3 Arquitetura de *hardware* e eletrônica

O diagrama esquemático das conexões e a placa de circuito impresso (PCB) personalizada do circuito serão feitos utilizando o *software* KiCAD. As conexões eletrônicas do sistema serão baseadas nas especificações técnicas de cada componente, seguindo seus respectivos manuais de aplicação (datasheets).

A arquitetura de *hardware* do robô será composta pelos elementos listados a seguir.

5.3.1 Motores de passo NEMA 17

O motor de passo NEMA 17 (Figura 4) é um atuador eletromecânico que converte pulsos elétricos em movimentos angulares (Dejan, 2015). Esses motores não giram de forma contínua, mas sim em passos angulares bem definidos. O NEMA 17 que será usado no projeto possui 200 degraus, sendo assim, cada degrau correspondente a 1,8º (Cook, 2022).

Internamente, o motor de passo possui enrolamentos eletromagnéticos e um rotor com dentes. Os pulsos aplicados sequencialmente nos enrolamentos criam campos magnéticos que puxam o rotor dente a dente. Neste projeto, estes motores serão utilizados para movimentar as juntas do braço robótico.

Figura 4 - Motor de Passo NEMA 17

Fonte: Autoria Própria (2025).

5.3.2 Servo Motor MG996R

O servo motor MG996R (Figura 5) é um motor de corrente contínua que possui um sistema de controle de posição via sinal PWM (Modulação por Largura de Pulso) (Sparkfun, 2021). Os servos recebem comandos de posição por meio de pulsos periódicos e se ajustam automaticamente até atingir o ângulo desejado (Amc, 2024). Este componente irá realizar a abertura e o fechamento da garra do braço robótico para a manipulação dos objetos.

Figura 5 – Servo Motor MG996R

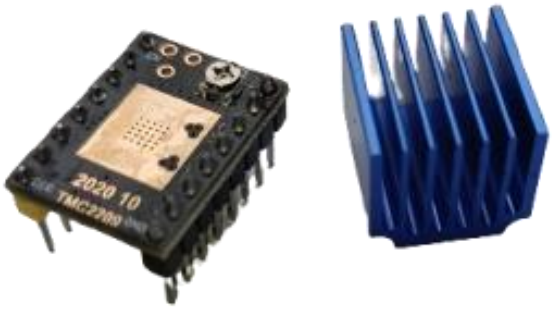
Fonte: Autoria Própria (2025).

5.3.3 *Drivers* de motor TMC2209

Os *drivers* são componentes eletrônicos responsáveis por controlar os motores de passo. A Ic-components (2015) descreve suas características: O *driver* converte sinal digital em correntes elétricas e as enviam para os atuadores. O TMC2209 (Figura 6) possui tecnologia StealthChop2, que proporciona uma movimentação mais silenciosa e mais suave por meio de *microstepping*. Este driver também possui detecção de perda de passo (StallGuard4) e ajuste automático de corrente (CoolStep).

O TMC2209 oferece proteção contra sobrecorrente e sobretemperatura. Suporta até 2A por fase e pode ser controlado via comunicação UART com o microcontrolador.

Figura 6 – *Driver* TMC2209



Fonte: Autoria Própria (2025).

5.3.4 Fonte de Alimentação 24V

A fonte de alimentação chaveada (Figura 7) fornecerá tensão contínua para todo o sistema. Ela converte tensão alternada da rede (110V/220V) em tensão contínua de 24V. Esta

fonte suporta corrente de até 20A, tendo potência suficiente para alimentar simultaneamente todo o circuito. Possui proteção contra sobrecorrente, sobretensão e curto-circuito.

Figura 7 – Fonte 24V

Fonte: RhMateriaisElétricos (s.d.).

5.3.5 Conversor buck

O conversor buck (Figura 8) é um regulador de tensão do tipo *step-down*, que reduz a tensão de entrada de 24V para valores mais baixos. O ajuste de tensão é feito através de um potenciômetro presente no próprio componente. Neste projeto, o conversor será usado para garantir a alimentação do Raspberry Pi e da câmera USB.

Figura 8 – Conversor Buck

Fonte: Autoria Própria (2025).

5.3.6 Raspberry Pi

O Raspberry Pi (Figura 9) é um computador de placa única (SBC – Single Board Computer) baseado em processador de arquitetura ARM. Possui várias interfaces de comunicação, como portas GPIO, portas USB, Wi-Fi e *Bluetooth*, além de ter suporte nativo para Linux. (Raspberry Pi, 2021).

Neste projeto, o Raspberry Pi será utilizado como unidade processamento central do sistema. Será responsável por executar os algoritmos de controle PID e LQR, processar as imagens capturadas pela câmera, calcular a cinemática inversa do manipulador e gerenciar a comunicação entre os módulos do braço robótico e o software de monitoramento.

Figura 9 – Raspberry Pi

Fonte: RaspberryPi (s.d.).

5.3.7 Câmera USB

Será utilizada a câmera Logitech C920 (Figura 10), conectada ao Raspberry Pi via USB para captura de imagens e, com isso, a identificação e classificação de objetos pelo sistema de visão computacional.

Figura 10 – Câmera USB

Fonte: Autoria Própria (2025).

5.4 Lógica de controle

A lógica de controle será implementada no Raspberry Pi, que atuará como unidade central de processamento. A execução das tarefas será realizada de forma autônoma, sem a necessidade de intervenção humana.

Inicialmente, serão testadas ações de controle em malha aberta para avaliação do comportamento básico do sistema. Em seguida, será implementada a ação de controle de malha fechada, o controlador proporcional-integral-derivativo (PID), com o objetivo de reduzir erros de posição. Posteriormente, será explorada uma técnica mais avançada, o regulador quadrático linear (LQR), para otimização do controle com base em modelos matemáticos do robô.

O desempenho de cada estratégia será analisado com base em métricas como precisão no posicionamento, tempo de execução, eficiência na separação e estabilidade dos movimentos. Essas análises permitirão validar qual abordagem é mais adequada ao contexto proposto.

5.5 Técnicas de visão computacional

O sistema terá uma câmera USB conectada ao Raspberry Pi para capturar imagens da área de atuação do manipulador. As imagens adquiridas serão processadas utilizando a biblioteca de visão computacional OpenCV, na linguagem de programação Python, a qual irá identificar as características físicas dos objetos a serem manipulados. A partir das imagens obtidas, serão aplicadas técnicas de segmentação de imagem, como filtragem por limiar de cor, detecção de contornos e formas e estimativa de pose. A identificação será configurada para reconhecer objetos com predominância nas três cores primárias: vermelho, verde e azul, utilizando faixas específicas no espaço HSV.

Com os dados obtidos após a segmentação, o sistema de visão fornecerá como saída as coordenadas da posição e orientação do objeto a ser manipulado, que servirá como entrada para os algoritmos de controle. Assim, o braço robótico poderá planejar sua trajetória com base na posição real do objeto detectado, realizando a movimentação até o ponto desejado.

5.6 *Software*

Será desenvolvido um *software* para desktop utilizando a linguagem C++ e a biblioteca gráfica Qt para monitoramento e controle à parte do sistema. O programa terá uma interface gráfica simples e irá funcionar em uma máquina separada do controlador do braço robótico.

O Raspberry Pi ficará responsável por enviar as informações do braço robótico ao *software* de monitoramento através de uma conexão serial direta.

5.7 Testes e simulação

Antes da implementação prática do braço robótico, serão realizados testes e simulações virtuais para validar o projeto mecânico, a lógica de controle e a interação com o ambiente. O

Gazebo é uma plataforma de simulação de robótica que permite criar ambientes virtuais realistas, modelar robôs, simular sensores e testar atuadores. Neste projeto, o modelo 3D do braço robótico será exportado para o *software* Gazebo para avaliação do funcionamento dos motores, da resposta dos controladores e da interação com objetos. Dessa forma, será possível testar o sistema antes de sua implementação física e, com isso, evitar falhas e otimizar o robô.

Os testes realizados visam avaliar a precisão de posicionamento, a repetibilidade do movimento, e a eficiência do sistema de visão computacional na execução de tarefas de *pick and place*. Serão avaliadas algumas métricas, como: erro de posicionamento entre a posição esperada e a posição final do robô; o tempo total de ciclo desde a identificação do objeto até finalizar a tarefa de separação; e a taxa de sucesso em reconhecer os objetos.

6 ORÇAMENTO

Tabela 1 – Orçamento do Projeto

Item	**Quantidade**	**Preço Unitário (R$)**	**Subtotal (R$)**
Motor de Passo NEMA 17	4	46,80	187,20
Servo Motor MG996R	1	21,75	21,75
Driver TMC2209	4	11,70	46,80
Fonte 24V 20A	1	100,00	100,00
Conversor Buck	1	11,20	11,20
Raspberry Pi 5 4GB	1	669,00	669,00
Material PLA	2	105,40	210,80
Outros*	1	300,00	300,00
TOTAL			**1546,75**

*Outros materiais incluem fios, parafusos, rolamentos e também ferramentas. Será utilizada a câmera USB pessoal, não cabendo aqui estimar seu preço.

7 CRONOGRAMA

Tabela 2 – Cronograma do Projeto

Atividades	**Jun**	**Jul**	**Ago**	**Set**	**Out**	**Nov**
Construção e modelagem do manipulador	X	X				
Montagem do circuito eletrônico		X				
Implementação de algoritmos de controle		X	X			
Implementação de visão computacional			X	X		
Integração visão x robô				X	X	
Desenvolvimento de *software* de monitoramento				X	X	
Testes e simulação						X

REFERÊNCIAS

ANDERSON, B. D. O.; MOORE, J. B. **Optimal Control:** Linear Quadratic Methods. Englewood Cliffs: Prentice Hall, 1990.

AMC. **Advanced Motion Controls**: What is a Servo Motor: Definition, Origins, Components, Types & Applications. 2024. Disponível em: https://www.a-m-c.com/servomotor/. Acesso em: 22 maio 2025.

ÅSTRÖM, K. J.; HÄGGLUND, T. **PID Controllers:** Theory, Design, and Tuning. 2. ed. Research Triangle Park: ISA – The Instrumentation, Systems, and Automation Society, 1995.

ÅSTRÖM, K. J.; HÄGGLUND, T. **The future of PID control**. Control Engineering Practice, *2001*.

BILLARD, A.; KRAGIC, D. Trends and challenges in robot manipulation. **Science**, v. 364, n. 6446, p. eaat8414, 2019.

BRADSKI, G.; KAEHLER, A. **Learning OpenCV.** [s.l.] O'reilly Media, Inc., 2008.

COOK, J. **Stepper motor basics explained.** 2022. Disponível em: https://www.arrow.com/en/research-and-events/articles/stepper-motor-basics-explained. Acesso em: 22 maio 2025.

CUI, J.; TRINKLE, J. Toward next-generation learned robot manipulation. **Science Robotics**, v. 6, n. 54, 2021.

DEJAN. **How a stepper motor works** – how to mechatronics. 2015. Disponível em: https://howtomechatronics.com/how-it-works/electrical-engineering/stepper-motor/. Acesso em: 22 maio 2025.

FARIA, R. O. et al. A Methodology for Autonomous Robotic Manipulation of Valves Using Visual Sensing. **IFAC-PapersOnLine**, 2015.

FRUNZAVERDE, D. et al. The Influence of the Layer Height and the Filament Color on the Dimensional Accuracy and the Tensile Strength of FDM-Printed PLA Specimens. **Polymers**, 2023.

GRASSI, M. V. **Desenvolvimento e aplicação de um sistema de visão para robô industrial de manipulação**. Dissertação – Pós-Graduação Engenharia Mecânica, Porto Alegre, 2005.

GÜNTHER, H. Pesquisa qualitativa versus pesquisa quantitativa: esta é a questão? **Psicologia:** Teoria e Pesquisa, v.22, n. 2, p. 201-209, 2006.

HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. New York: Cambridge University Press, 2017.

IC-COMPONENTS. **Your Guide to TMC2209 Stepper Driver with Microcontroller.** 2015. Disponível em: https://www.ic-components.com/blog/your-guide-to-tmc2209-stepper-driver-with-microcontroller.jsp. Acesso: 10 maio 2025.

JIANG, H.; P. WACHS, J.; S. DUERSTOCK, B. Integrated vision-based system for efficient, semi-automated control of a robotic manipulator. **International Journal of Intelligent Computing and Cybernetics**, v.7, n. 3, p. 253-266, 2014.

KUCUK, S.; BINGUL, Z. Robot Kinematics: Forward and Inverse Kinematics. **Industrial Robotics:** Theory, Modelling and Control, 2006.

LI, C. et al. Skill Acquisition and Controller Design of Desktop Robot Manipulator Based on Audio–Visual Information Fusion. **Machines**, v. 10, n. 9, p. 772, 2022.

MARTINEZ-MARTIN, E.; DEL POBIL, A. Vision for Robust Robot Manipulation. **Sensors**, v. 19, n. 7, p. 1648, 2019.

NASCIMENTO, F. **Classificação da pesquisa:** natureza, método ou abordagem metodológica, objetivos e procedimentos. Brasília: Thesaurus, cap. 6, 2016.

ODA, N.; ITO, M.; SHIBATA, M. Vision-based motion control for robotic systems. **Ieej Transactions on Electrical and Electronic Engineering**, v. 4, n. 2, p. 176-183, 2009.

OGATA, K. **Modern control engineering**. 5. ed. Delhi: Pearson, 2010.

OLIVEIRA, A. **Modelagem cinemática do manipulador robótico IRB 8700**. Trabalho de Conclusão de Curso – Faculdade de Engenharia de Ilha Solteira, Ilha Solteira, 2022.

RASPBERRYPI. **Raspberry Pi hardware**. [s.d]. Disponível em: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html. Acesso: 22 maio 2025.

RHMATERIAISELETRONICOS. **Fonte Chaveada 24V**. [s.d]. Disponível em : https://www.rhmateriaiseletricos.com.br/fonte-chaveada-24v. Acesso: 26 maio 2025.

RIBEIRO, L.; CÔRTES, L. R.; OKADA, K. F.; MORAIS, A. S. **Controlador LQR aplicado à planta didática Twin Rotor MIMO System utilizando MATLAB e Arduino**. Uberlândia: Universidade Federal de Uberlândia, 2019.

SHAHRIA, M. T. et al. A Comprehensive Review of Vision-Based Robotic Applications: Current State, Components, Approaches, Barriers, and Potential Solutions. **Robotics**, v.11, n. 6, p. 139, 2022.

SPARKFUN. **Servos Explained**. 2021. Disponível em: https://www.sparkfun.com/servos. Acesso em: 22 maio 2025.

TANG, Y. et al. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. **Frontiers in Plant Science**, v. 11, 2020.

TORRES, R. **Manipulação robótica na indústria cerâmica**. Tese Mestrado – Instituto Superior de Engenharia de Coimbra, Instituto Politécnico de Coimbra, Coimbra, 2022.

WELLS, G.; VENAILLE, C.; TORRAS, C. Vision-based robot positioning using neural networks. **Image and Vision Computing**, v. 14, n. 10, p. 715-732, 1996.

ZHANG, B.; LIU, P. Control and benchmarking of a 7-DOF robotic arm using Gazebo and ROS. **PeerJ Computer Science**, v. 7, p. e383, 2021.