

**CENTRO UNIVERSITÁRIO DE VIÇOSA  
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**DANIEL OLIVEIRA TAVARES**

**Sistema de Manipulação Robótica baseado em Visão  
Computacional**

**Viçosa  
2025**

**DANIEL OLIVEIRA TAVARES**

## **Sistema de Manipulação Robótica baseado em Visão Computacional**

Projeto apresentado ao Curso de Engenharia de Computação do Centro Universitário de Viçosa, como parte dos requisitos para conclusão da disciplina CCE001 Trabalho de Conclusão de Curso 1.

Orientador: Prof. Me. Vinicius Martins Almeida

**Viçosa  
2025**

## RESUMO

Coloque o seu resumo aqui

**Palavras-chave:** PC1. PC1. PC1. PC1.

## SUMÁRIO

1	INTRODUÇÃO . . . . .	4
1.1	Problema e Justificativa . . . . .	5
2	OBJETIVOS . . . . .	6
2.1	Objetivo Geral . . . . .	6
2.2	Objetivos Específicos . . . . .	6
2.3	Resultados Esperados . . . . .	6
3	REVISÃO BIBLIOGRÁFICA . . . . .	8
3.1	seção . . . . .	8
4	TRABALHOS RELACIONADOS . . . . .	9
5	MATERIAL E MÉTODOS . . . . .	12
5.1	Caracterização da Pesquisa . . . . .	12
5.2	Modelagem e Estrutura Mecânica . . . . .	12
5.3	Arquitetura de Hardware e Eletrônica . . . . .	12
5.4	Lógica de controle . . . . .	15
5.5	Técnicas de Visão Computacional . . . . .	16
5.6	<i>Software</i> . . . . .	16
5.7	Testes e Simulação . . . . .	16
6	CRONOGRAMA . . . . .	17
7	ORÇAMENTO . . . . .	18
	REFERÊNCIAS . . . . .	19

## 1 INTRODUÇÃO

Robôs manipuladores interagem fisicamente com o ambiente e, desta 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 aplicada muito na indústria, principalmente em linhas de produção e automação de processos repetitivos. A manipulação robótica, área que estuda como robôs interagem fisicamente com o ambiente, vem sendo aplicada há muitos anos, principalmente na indústria para manufatura e automação de produtos. Este tipo de interação pode ser feito de inúmeras formas, dependendo da complexidade da tarefa para a qual o robô foi projetado.

Historicamente, a manipulação robótica consistia em uma série de movimentos pré-definidos sem a habilidade de adaptação a mudanças do ambiente (Trends, 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, 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, 2020). Segundo Shahira (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, Venaille, Torras (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 desafios importantes relacionados à precisão do reconhecimento visual e ao controle eficiente da trajetória dos manipuladores, especialmente 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, 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. Táticas 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 (Trends, 2019)(Shahria, 2022).

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, 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

VEM NO FINAL DA INTRODUCAO Espera-se ao final deste trabalho, possuir 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 ordenar de maneira eficiente os blocos e depositá-los 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 REVISÃO BIBLIOGRÁFICA**

PODE COMCAR COM OS TRABALHOS CORRELATOS E DEPOIS VA PARA OS CONCEITOS

#### **3.1 seção**

INTERDICIPLINAR

CRONO 2 SEMESTRE MAIS ESPECIFICO

## 4 TRABALHOS RELACIONADOS

Billard, Kraige (2019) fizeram uma análise dos principais desafios enfrentados na manipulação robótica moderna. A pesquisa publicada na revista americana *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.

No contexto de modelagem de robótica, Kucuk, 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 (D-H). O trabalho demonstra como associar sistemas de coordenadas às articulações para obter a pose final do atuador. Os autores tratam a cinemática direta como um problema direto e que sempre há uma solução, 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.

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.

No campo da visão computacional, Oda,Ito,Shibata (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.

De forma similar, Martin,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 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.

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.

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.

## 5 MATERIAL E MÉTODOS

### 5.1 Caracterização da Pesquisa

A natureza desta pesquisa é de caráter aplicada, pois tem como foco gerar conhecimento ao solucionar problemas específicos (Nascimento, 2016). O objetivo é desenvolver um sistema robótico funcional.

### 5.2 Modelagem e Estrutura Mecânica

A modelagem 3D do manipulador será feita 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 serem então impressos.

As peças serão impressas utilizando uma impressora *Creality Ender3v2 Neo* 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 ser de mais fácil impressão comparado com outros tipos de filamento (Frunzaverde, 2023). A largura da extrusão para todas as peças será mantida no padrão de 0,4 milímetros.

Após projetada e impressa a estrutura física do sistema, será calculado o modelo cinemático do braço robótico usando a notação de Denavit-Hartenberg. A partir disso, será calculada a cinemática direta e a cinemática 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.

#### Motores de passo NEMA 17

O motor de passo NEMA 17 é um atuador eletromecânico que realiza movimentos precisos baseados em sinais discretos de pulso. Cada pulso enviado ao motor gera um movimento angular específico, o que permite controle fino da posição do seu eixo. Neste projeto, estes motores serão utilizados para movimentar as juntas do braço robótico.



Figura 1 – Motor de Passo NEMA 17  
Fonte: Autoria Própria

### Servo Motor MG996R

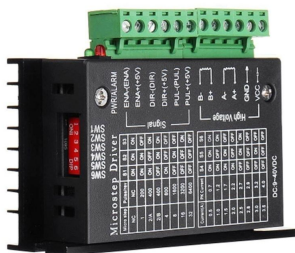
O servo motor MG996R é um motor de corrente contínua que possui um sistema de controle de posição via sinal PWM (Pulse Width Modulation). Este componente irá realizar a abertura e fechamento da garra do braço robótico para manipulação dos objetos.



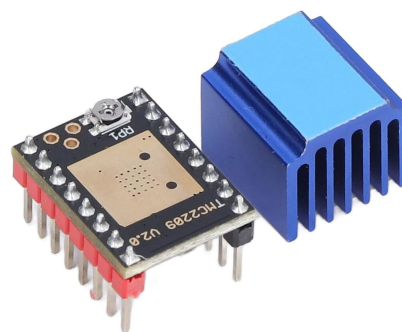
Figura 2 – Servo Motor MG996R  
Fonte: Autoria Própria

### Drivers de motores TB6600 e TMC2209

Os drivers são componentes que convertem sinal digital em correntes elétricas e as enviam para os atuadores. O driver TB6600 permite maior corrente elétrica, além de oferecer proteção contra sobrecorrente e sobretemperatura. Já o driver TMC2209 permite aos motores de passo uma movimentação mais silenciosa e maior precisão.



(a) Driver TB6600



(b) Driver TMC2209

Figura 3 – Caption for this figure with two images

### Fonte de Alimentação 24V

A fonte de alimentação de 24v fornecerá tensão contínua para todo o sistema.



Figura 4 – Fonte 24v  
Fonte: Autoria Própria

### Conversor Buck

O conversor buck é responsável por reduzir a tensão de 24V para níveis mais baixos, e será usado para alimentar o Raspberry Pi e a câmera USB.

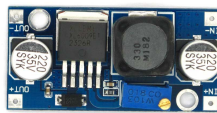


Figura 5 – Conversor Buck  
Fonte: Autoria Própria

### Raspberry Pi

O Raspberry Pi é uma plataforma de computação compacta (SBC – Single Board Computer) que será utilizada como o processamento central do sistema. Será responsável por executar os algoritmos de controle dos motores, processar as imagens capturadas pela câmera e gerenciar a comunicação entre os módulos do braço robótico.

### Câmera USB

Será utilizada a câmera Logitech C920, 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 6 – Raspberry Pi  
Fonte: Autoria Própria



Figura 7 – Câmera USB  
Fonte: Autoria Própria

#### 5.4 Lógica de controle

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, serão exploradas técnicas mais avançadas como o regulador quadrático linear (LQR) para otimização do controle com base em modelos matemáticos do robô, e controle preditivo baseado em modelo (MPC) da qual considera restrições físicas do sistema e prevê estados futuros.

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.

O desempenho de cada estratégia será analisado com base em métricas como precisão no posicionamento, tempo de execução da tarefa, 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 posição e orientação.

Com os dados obtidos após a segmentação, o sistema de visão fornecerá como saída a 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 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 simulados. Dessa forma, será possível testar o sistema antes de sua implementação física e, com isso, evitar falhas e otimizar o robô.

## 6 CRONOGRAMA

Tabela 1 – Cronograma 1º Semestre 2025

<b>Atividades</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>	<b>06</b>
Tema, Problema e Objetivo	X				
Resultados Esperados e Metodologia		X			
Introdução e Justificativa			X		
Cronograma, Orçamento			X		
Resumo e palavras chave			X		
Entrega final do projeto				X	
Correção do Projeto					X
Registro no NUPEX					X

Fonte: Elaborado pelo autor

## 7 ORÇAMENTO

Tabela 2 – Orçamento do Projeto

Item	Qtd	Preço Unitário (R\$)	Preço Total (R\$)
Motor de Passo Nema 17	4	46,80	187,20
Servo Motor	1	21,75	21,75
Driver TB6600	3	25,00	75,00
Driver TMC2209	2	11,70	23,40
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
<b>Total Geral</b>			<b>1.411,15</b>

Fonte: Elaborado pelo autor

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

## REFERÊNCIAS