



UNIVERSIDADE FEDERAL DE PERNAMBUCO

**CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

GABRIEL HENRIQUE CAVALCANTI DE OLIVEIRA XAVIER

**COMPARAÇÃO DO DESEMPENHO DE CONTROLE DE UM SISTEMA *BALL ON PLATE* COM DIFERENTES MÉTODOS DE REALIMENTAÇÃO:
MÉTODO TRADICIONAL E POR VISÃO COMPUTACIONAL**

Recife
2025

GABRIEL HENRIQUE CAVALCANTI DE OLIVEIRA XAVIER

**COMPARAÇÃO DO DESEMPENHO DE CONTROLE DE UM SISTEMA *BALL ON PLATE* COM DIFERENTES MÉTODOS DE REALIMENTAÇÃO:
MÉTODO TRADICIONAL E POR VISÃO COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador(a): Prof. Dr. Douglas Contente Pimentel Barbosa

Coorientador: Prof. Dr. Rafael Cavalcanti Neto

Recife
2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Xavier, Gabriel Henrique Cavalcanti de Oliveira.

Comparação do desempenho de controle de um sistema Ball on Plate com diferentes métodos de realimentação: método tradicional e por visão computacional / Gabriel Henrique Cavalcanti de Oliveira Xavier. - Recife, 2025.

125 p. : il., tab.

Orientador(a): Douglas Contente Pimentel Barbosa

Coorientador(a): Rafael Cavalcanti Neto

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia de Controle e Automação - Bacharelado, 2025.

Inclui referências, apêndices.

1. Ball on Plate. 2. Sistema de controle PID. 3. Visão computacional. 4. Tela resistiva. 5. ESP32. I. Barbosa, Douglas Contente Pimentel. (Orientação). II. Cavalcanti Neto, Rafael. (Coorientação). IV. Título.

620 CDD (22.ed.)

GABRIEL HENRIQUE CAVALCANTI DE OLIVEIRA XAVIER

**COMPARAÇÃO DO DESEMPENHO DE CONTROLE DE UM SISTEMA *BALL ON PLATE* COM DIFERENTES MÉTODOS DE REALIMENTAÇÃO:
MÉTODO TRADICIONAL E POR VISÃO COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Aprovado em: 13/08/2025

BANCA EXAMINADORA

Prof. Dr. Douglas Contente Pimentel Barbosa (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Calebe Hermann de Oliveira Lima (Examinador Interno)
Universidade Federal de Pernambuco

Eng. M.Sc. Gustavo Azevedo Teixeira (Examinador Interno)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Primeiramente, agradeço a Deus por me conceder saúde, força e determinação para chegar até aqui. Foram anos de dedicação, desafios, preocupações e também de muitas alegrias.

Expresso minha mais profunda gratidão à minha família: meu pai, Gerailton Xavier, minha mãe, Cybelle Xavier, meu irmão, Guilherme Xavier, e minha avó, Maria José. Sou imensamente grato por todo o suporte e incentivo ao longo desta jornada. Vocês são meu alicerce e a razão da minha perseverança. Espero, um dia, poder retribuir todo o amor, dedicação e esforços que tiveram por mim.

À minha namorada, Samary Vieira, registro um agradecimento especial pelo apoio constante ao longo dos últimos dois anos, em especial na reta final deste trabalho. Além de me oferecer suporte emocional diário, esteve presente nos momentos mais difíceis, ajudando-me nas decisões e contribuindo diretamente para o desenvolvimento do TCC. Sem você, esse caminho teria sido muito mais árduo.

Agradeço ao meu orientador, Prof. Dr. Douglas Contente, pela dedicação, paciência e compromisso demonstrados ao longo de toda a orientação. Ao meu coorientador, Prof. Dr. Rafael Cavalcanti, expresso meus sinceros agradecimentos pelo apoio constante, mesmo diante de suas inúmeras responsabilidades. A colaboração de ambos foi essencial para a realização deste projeto, e seus ensinamentos se tornaram fundamentais não apenas para a conclusão deste trabalho, mas também para minha formação pessoal e profissional.

Ao Grupo de Eletrônica de Potência e Acionamentos Elétricos (GEPAE) e ao GRE3D da UFPE, agradeço pelo suporte e contribuições que viabilizaram o desenvolvimento deste trabalho.

Aos meus amigos, de vida e de faculdade, sou grato pela companhia, pelos momentos de descontração e pelo apoio em todas as fases dessa caminhada. Vocês tornaram meus dias mais leves e felizes.

Por fim, desejo a todos paz, felicidade e sucesso, e encerro com meu mais sincero e profundo obrigado!

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de controle dinâmico do tipo *Ball on Plate*, cujo objetivo é equilibrar uma esfera livre sobre uma superfície plana por meio do acionamento de servomotores controlados por realimentação. Foram implementadas e comparadas duas abordagens distintas de detecção da posição da esfera: uma utilizando uma tela resistiva sensível ao toque (*touchscreen*) e outra baseada em visão computacional, com processamento de imagem em tempo real. O sistema foi modelado matematicamente e controlado por um algoritmo PID implementado no microcontrolador ESP32. Além disso, foi realizada uma comparação do comportamento dinâmico com duas diferentes esferas, variando massa, diâmetro e atrito. Os resultados experimentais demonstraram que o método de visão computacional apresentou desempenho superior, com menor sobressinal e tempo de acomodação, além de maior estabilidade frente a ruídos. A proposta se mostra eficaz para fins didáticos e de pesquisa, possibilitando o estudo prático de controle em sistemas dinâmicos não-lineares.

Palavras-chave: *Ball on Plate*; Sistema de controle PID; Visão computacional; Tela resistiva; ESP32.

ABSTRACT

This work presents the development of a dynamic control system of the Ball on Plate type, whose objective is to balance a free-moving sphere on a flat surface through the actuation of servomotors controlled via feedback. Two distinct approaches for detecting the sphere's position were implemented and compared: one using a resistive touchscreen and the other based on computer vision with real-time image processing. The system was mathematically modeled and controlled by a PID algorithm implemented on the ESP32 microcontroller. Additionally, a comparison was made of the dynamic behavior using two different spheres, varying in mass, diameter, and friction. Experimental results demonstrated that the computer vision method achieved superior performance, with lower overshoot and settling time, as well as greater robustness to noise. The proposed system proves to be effective for educational and research purposes, enabling practical study of control in nonlinear dynamic systems.

Keywords: Ball on Plate; PID Control System; Computer Vision; Resistive Touchscreen; ESP32.

LISTA DE ILUSTRAÇÕES

Figura 1 – Regulador de Watt.	21
Figura 2 – Representação de um sistema SISO.	23
Figura 3 – Sistema MIMO.....	24
Figura 4 – Sinal degrau unitário de tempo contínuo e de tempo discreto.	25
Figura 5 – Especificações de projeto ($\zeta = 0.4$, $\omega_n = 4.0$).....	29
Figura 6 – Localização dos pólos de um sistema instável no plano s	30
Figura 7 – Sistema de controle digital.	32
Figura 8 – Saída do amostrador para uma entrada contínua $u(t) = 1 - e^{-t}$, com período de amostragem $T = 1s$	32
Figura 9 – Saída de um retentor de ordem zero (ZOH).....	33
Figura 10 – Exemplo de fluxo de visão computacional com detecção de pedestres.....	37
Figura 11 – Representação de uma imagem digital bidimensional.	39
Figura 12 – Ilustração da operação de convolução entre uma imagem digital e um kernel 3×3	41
Figura 13 – Representação do sistema RGB.....	47
Figura 14 – Representação do sistema HSV.	48
Figura 15 – Diagrama do processo de segmentação por cor no espaço HSV.....	49
Figura 16 – Modulo de processamento de imagem no OpenCV.....	53
Figura 17 – Estrutura de um sistema <i>Ball on Plate</i>	55
Figura 18 – Representação dos eixos ortogonais do sistema <i>Ball on Plate</i> em vistas superior e frontal.	56
Figura 19 – Sinal PWM.	57
Figura 20 – Componentes internos de um servomotor.	58
Figura 21 – Representação de uma ponte H.....	59
Figura 22 – Composição de uma tela sensível ao toque resistiva de 4 fios.	61
Figura 23 – Circuito equivalente da tela sensível ao toque resistiva de 4 fios.	61
Figura 24 – Leitura da posição x e y utilizando divisor de tensão.	62
Figura 25 – Estrutura de suporte para o servomotor.....	66
Figura 26 – Estrutura de suporte para o servomotor, vistas frontal e lateral esquerda com dimensões.	66
Figura 27 – Vista superior da estrutura de suporte para o servomotor.	67

Figura 28 – Haste central do sistema.....	67
Figura 29 – Desenho técnico da Haste central do sistema com vistas lateral e superior cotadas.....	68
Figura 30 – Junta universal.	69
Figura 31 – Eixo central superior.....	69
Figura 32 – Eixo central superior.....	70
Figura 33 – Desenho técnico da face da base inferior, com medidas em milímetros.	71
Figura 34 – Desenho técnico da face da base superior, com medidas em milímetros.	72
Figura 35 – Haste de conexão.	72
Figura 36 – Placa de desenvolvimento ESP32 DevKit V1.	73
Figura 37 – Diagrama de pinagem da placa ESP32 DevKit V1.....	74
Figura 38 – Servomotor DS3225MG acoplado em sua estrutura de suporte.....	74
Figura 39 – Controle de posição do servomotor DS3225MG por largura de pulso PWM.	75
Figura 40 – Tela resistiva de 4 fios.....	76
Figura 41 – Webcam NB307 Full HD 1080p.	76
Figura 42 – Esfera.....	77
Figura 43 – Esfera pintada.	77
Figura 44 – Fonte chaveada CA/CC MS-60-12.....	79
Figura 45 – Conversor buck XL4005.....	79
Figura 46 – Diagrama esquemático do circuito de alimentação do sistema.....	79
Figura 47 – Placa de circuito impresso utilizada para controle e aquisição de dados do sistema.....	80
Figura 48 – Modelo do <i>Ball on Plate</i> simplificado (eixo X).	82
Figura 49 – Projeções horizontal e vertical da posição da esfera.	83
Figura 50 – Simulação em malha aberta (sem controlador) do sistema com entrada em degrau unitário.	91
Figura 51 – Simulação do sistema em malha fechada através do MATLAB/Simulink.	92
Figura 52 – Implementação de um controlador utilizando o PID Turner no MATLAB.	93

Figura 53 – Resultado da simulação em malha fechada do sistema com entrada em degrau unitário.	94
Figura 54 – Arquitetura do sistema de controle com realimentação tradicional.	95
Figura 55 – Fluxograma do algoritmo de controle implementado para o sistema <i>Ball on Plate</i> no método tradicional de realimentação.....	98
Figura 56 – Arquitetura do sistema de controle por visão computacional no <i>Ball on Plate</i>	101
Figura 57 – Etapa de calibração dos parâmetros H, S e V para detecção da esfera.	103
Figura 58 – Definição dos limites da região de controle para posicionamento da esfera no centro.	104
Figura 59 – Fluxograma do algoritmo final de rastreamento da esfera.	106
Figura 60 – Fluxograma do algoritmo de controle implementado para o sistema <i>Ball on Plate</i> no método de realimentação por visão computacional.	108
Figura 61 – Estrutura de suporte para posicionamento da câmera.....	110
Figura 62 – Estrutura de suporte desenvolvida para fixação e posicionamento da câmera de aquisição de imagens.....	111
Figura 63 – Circuito de alimentação e placa de controle utilizados no sistema abrigados em uma caixa de acrílico.	112
Figura 64 – Estrutura completa do sistema integrando os módulos mecânico, eletrônico e de visão computacional.	112
Figura 65 – Resposta dos eixos X e Y no método tradicional de realimentação com a esfera verde.	113
Figura 66 – Resposta dos eixos X e Y no método tradicional de realimentação com a utilização da esfera de aço (esfera prata).	114
Figura 67 – Resposta dos eixos X e Y no método de realimentação por visão computacional com a esfera verde.....	116

LISTA DE TABELAS

Tabela 1 – Diagrama de ligação da tela resistiva touchscreen de 4 fios.....	62
Tabela 2 – Especificações técnicas servomotor DS3225MG.....	75
Tabela 3 – Parâmetros do controlador utilizados na simulação em malha fechada e resultados.....	94
Tabela 4 – Diagrama de ligação da Esp32 com a tela resistiva touchscreen de 4 fios.	95
Tabela 5 – Diagrama de ligação da Esp32 com os servomotores e configurações. .	99
Tabela 6 – Parâmetros do controlador utilizados na implementação em malha fechada.	99
Tabela 7 – Valores dos parâmetros H, S e V utilizados para a detecção da esfera.	103
Tabela 8 – Medidas de desempenho obtidas para os eixos X e Y no método tradicional de realimentação com a esfera verde.	114
Tabela 9 – Medidas de desempenho obtidas para os eixos X e Y no método tradicional de realimentação com a esfera de aço (esfera prata).....	115
Tabela 10 – Medidas de desempenho obtidas para os eixos X e Y no método de realimentação por visão computacional com a esfera verde.....	116

LISTA DE ABREVIATURAS E SIGLAS

FT	Função de transferência
MIMO	Múltiplas entradas e múltiplas saídas
SISO	Única entrada e única saída
CV	Visão computacional
DAC	Conversor digital analógico
ADC	Conversor analógico digital
ZOH	Retentor de ordem zero

LISTA DE SÍMBOLOS

L	Função de Euler-Lagrange
K	Energia cinética total do sistema
U	Energia potencial total do sistema
m_b	Massa da bola/esfera
v_b	Velocidade linear da bola/esfera
x	Posição linear da bola/esfera, medida ao longo do eixo X
y	Posição linear da bola/esfera, medida ao longo do eixo Y
J_b	Momento de inércia da bola/esfera
ω_b	Velocidade angular da bola/esfera
J_p	Momento de inércia do prato
ϕ_x	Deslocamento angular do prato em torno do eixo X (inclinação da mesa no eixo horizontal frontal)
ϕ_y	Deslocamento angular do prato em torno do eixo Y (inclinação da mesa no eixo horizontal lateral)
θ_x	Ângulo de rotação do servomotor responsável pelo controle no eixo
θ_y	Ângulo de rotação do servomotor responsável pelo controle no eixo Y
l	Distância entre o centro do prato e o ponto de conexão da haste com o prato (raio de atuação da esfera)
d	Raio de alavanca do servomotor

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVOS	17
1.1.1	Geral.....	17
1.1.2	Específicos	18
1.2	ORGANIZAÇÃO DO TRABALHO.....	18
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	TEORIA DE CONTROLE	20
2.1.1	Modelagem de Sistemas de Controle.....	21
2.1.1.1	<i>Classificação de Sistemas</i>	22
2.1.1.2	<i>Modelagem no Domínio da Frequência</i>	26
2.1.2	Especificações de Projeto	27
2.1.3	Análise de Estabilidade	29
2.1.4	Sistemas de Controle Digitais e Discretização	31
2.1.4.1	<i>Técnica de Discretização de Tustin</i>	35
2.2	VISÃO COMPUTACIONAL (COMPUTER VISION, CV).....	36
2.2.1	Etapas da Visão Computacional.....	39
2.2.2	Operações Sobre as Imagens no Domínio Espacial	40
2.2.2.1	<i>Filtragem de Suavização da Imagem</i>	41
2.2.2.2	<i>Limiarização (Tresholding)</i>	42
2.2.2.3	<i>Operações Morfológicas</i>	43
2.2.2.4	<i>Detecção de Bordas</i>	44
2.2.3	Representações e Espaços de Cor	46
2.2.3.1	<i>Sistema RGB</i>	46
2.2.3.2	<i>Sistema HSV</i>	47
2.2.3.3	<i>Segmentação de Imagens com Base em Cor no Espaço HSV</i>	48
2.2.4	Comparação entre Métodos Baseados em IA e Tradicionais.....	49
2.2.4.1	<i>Deep Learning (DL)</i>	49
2.2.4.2	<i>Método Tradicional de Detecção</i>	50
2.2.5	OpenCV	51
3	DESENVOLVIMENTO DA PLANTA DIDÁTICA BALL ON PLATE	54
3.1	FUNCIONAMENTO DO SISTEMA.....	54
3.1.1	Arquitetura Geral	55
3.1.2	Atuadores	56
3.1.3	Sensores	59
3.1.3.1	<i>Tela Sensível ao Toque Resistiva</i>	60
3.1.3.2	<i>Webcam</i>	63
3.1.4	Unidade de Controle.....	63
3.2	CONSTRUÇÃO DA ESTRUTURA MECÂNICA.....	65

3.2.1	Estrutura de Suporte para o Servomotor	66
3.2.2	Haste Central.....	67
3.2.3	Junta Universal.....	68
3.2.4	Eixo Central Superior.....	69
3.2.5	Bases Inferior e Superior	70
3.2.6	Hastes de Conexão	72
3.3	SELEÇÃO DOS EQUIPAMENTOS E COMPONENTES.....	73
3.3.1	ESP32 DevKit V1.....	73
3.3.2	Servomotor DS3225MG	74
3.3.1	Tela Resistiva de 4 Fios	75
3.3.2	Webcam NB307.....	76
3.3.3	Esfera	77
3.3.4	Circuito de Alimentação	78
3.3.5	Placa para Conexão da ESP32 e Servomotores	80
3.4	MODELAGEM DO SISTEMA	80
3.4.1	Linearização	86
3.4.2	Função de Transferência do Sistema em Malha Aberta.....	89
3.4.3	Simulação do Sistema em Malha Aberta.....	91
3.4.4	Simulação do Sistema em Malha Fechada	92
3.5	CONTROLE DO SISTEMA POR REALIMENTAÇÃO TRADICIONAL	94
3.5.1	Aquisição da Posição da Esfera com Tela Resistiva de 4 Fios	95
3.5.2	Algoritmo de Controle	96
3.5.3	Definição dos Parâmetros PID do Controlador Utilizados na Planta e Ajuste Fino	99
4	CONTROLE DO SISTEMA <i>BALL ON PLATE</i> POR VISÃO COMPUTACIONAL	101
4.1	DESCRIÇÃO DO SISTEMA	101
4.2	PROCESSAMENTO DE IMAGEM PARA DETECÇÃO DA ESFERA.....	102
4.2.1	Calibração de Parâmetros para Segmentação por Cor	102
4.2.2	Delimitação da Região de Atuação da Esfera	103
4.2.3	Algoritmo Final de Rastreamento da Esfera	104
4.3	FECHAMENTO DE MALHA	106
5	RESULTADOS EXPERIMENTAIS E ANÁLISE.....	110
5.1	ESTRUTURA MECÂNICA.....	110
5.2	ESTRUTURA SUPORTE PARA CÂMERA	111
5.3	CIRCUITO DE ALIMENTAÇÃO E PLACA DE CONTROLE	111
5.4	ESTRUTURA COMPLETA	112
5.5	RESULTADOS OBTIDOS NO MÉTODO TRADICIONAL DE REALIMENTAÇÃO.....	113
5.6	RESULTADOS OBTIDOS NO MÉTODO DE REALIMENTAÇÃO POR VISÃO COMPUTACIONAL	115
5.7	ANÁLISE DOS RESULTADOS	117
6	CONCLUSÕES E PROPOSTAS DE CONTINUIDADE	118

6.1	DIFICULDADES ENCONTRADAS	118
6.2	CONCLUSÕES FINAIS	119
6.3	PONTOS DE MELHORIA.....	120
6.4	PROPOSTAS DE CONTINUIDADE	121
REFERÊNCIAS.....		122
APÊNDICES		125
APÊNDICE A – LINK PARA OS VÍDEOS DE DEMONSTRAÇÃO DOS RESULTADOS EXPERIMENTAIS		125
APÊNDICE B – LINK DO GITHUB COM OS CÓDIGOS DO PROJETO		125

1 INTRODUÇÃO

Como afirmam Dorf e Bishop (2001), é papel de um engenheiro de controle conhecer e controlar os sistemas em sua volta, com o objetivo de trazer benefícios para humanidade. Assim, a sociedade vem experimentando avanços nessa disciplina, que contribuíram para sua grande difusão e relevância, principalmente ao longo do século XX, período em que os principais conceitos teóricos foram desenvolvidos. Atualmente, essa área é amplamente aplicada em setores como geração e transmissão de energia, aeroespacial, transporte, ciência dos materiais, processos industriais, redes e sistemas de computação, robótica, instrumentação e até mesmo economia, sendo essencial para o controle eficiente de inúmeros processos (ÅSTRÖM e MURRAY, 2021). Além disso, com o desenvolvimento da indústria 4.0 e internet das coisas (*Internet of things*, IoT) as soluções de controle se tornaram mais complexas e robustas à medida que a tecnologia aumentou.

Combinada à teoria de controle, a visão computacional tem ganhado destaque no meio científico, especialmente a partir de 2012, impulsionada pelo avanço do *deep learning* e das redes neurais convolucionais (*Convolutional Neural Networks*, CNNs). Esses avanços revolucionaram tarefas como classificação de imagens, segmentação e detecção de objetos. O modelo do YOLO, por exemplo, possibilita a detecção em tempo real e vêm sendo amplamente aplicados em áreas como automação industrial, saúde, veículos autônomos e agricultura (ALI e ZHANG, 2024). Além disso, os métodos tradicionais de visão computacional ainda são bastante utilizados, seja como suporte a modelos baseados em inteligência artificial, seja em aplicações nas quais se mostram mais vantajosos em relação ao uso exclusivo de técnicas baseadas em inteligência artificial (IA).

Paralelamente aos dois temas mencionados, as metodologias de ensino utilizadas no processo de aprendizagem em engenharia constituem um fator crucial para a formação de profissionais mais preparados e alinhados às demandas contemporâneas (MASSON et al., 2012). No Brasil, a formação em engenharia ainda é predominantemente baseada em abordagens tradicionais e teóricas, com pouca ênfase em atividades práticas e experimentais (DE REZENDE FREITAS e FORTES, 2020). Essa ausência de metodologia prática (ou metodologia ativa) compromete o preparo do futuro profissional. Diante desse cenário, a Aprendizagem Baseada em

Projetos (ABP), ou *Project-Based Learning* (PjBL), surge como uma alternativa de metodologia eficaz. Segundo Kokotsaki, Menzies e Wiggins (2016), a Aprendizagem Baseada em Projetos é uma metodologia centrada no aluno, fundamentada em princípios construtivistas, que promove a autonomia, a colaboração e o desenvolvimento de competências por meio da investigação de problemas autênticos e da construção de soluções concretas. Estudos apontam que, quando aplicada ao ensino de engenharia, a ABP é mais efetiva que as aulas teóricas (MASSON et al., 2012), sendo especialmente eficiente em disciplinas com alta carga teórica, como controle de sistemas. Além disso, essa abordagem contribui para o desenvolvimento de habilidades como liderança, pensamento crítico, resolução de problemas e trabalho em equipe (CAMARGO e DAROS, 2018). Como destacado por Rocha (2024), algumas das implementações mais recorrentes da ABP na área de controle incluem plantas didáticas como *Ball and Beam*, aeropêndulos e *Ball on Plate*. Neste trabalho, optou-se pelo desenvolvimento da planta *Ball on Plate* devido à sua relevância pedagógica e ao fato de permitir a observação, em tempo real, do comportamento do sistema sob controle.

O sistema *Ball on Plate*, por meio de diferentes implementações, oferece uma plataforma didática versátil para o desenvolvimento de conceitos relacionados à teoria de controle e à visão computacional. Trata-se de um sistema projetado para o estudo da dinâmica de sistemas e a realização de experimentos laboratoriais, permitindo a aplicação de abordagens de controle tanto clássicas quanto modernas. Além de seu uso consolidado no ensino de engenharia, sua integração com técnicas de visão computacional como forma de realimentação permite explorar, de forma prática, conceitos fundamentais dessa área em expansão, com aplicações em diversos setores tecnológicos (TUDIĆ, ŠIMIĆ e PERIĆ, 2022).

1.1 Objetivos

1.1.1 Geral

Este trabalho tem como objetivo projetar, desenvolver e comparar duas diferentes implementações de sistemas de controle para uma planta didática do tipo

Ball on Plate. A primeira estratégia a ser implementada utiliza a realimentação tradicional por meio de uma tela *touchscreen* resistiva, enquanto a segunda é baseada em visão computacional, empregando uma webcam posicionada estrategicamente acima da plataforma didática.

1.1.2 Específicos

- Desenvolver a estrutura mecânica de uma planta didática baseada no sistema *Ball on Plate*;
- Implementar um sistema de leitura de coordenadas da esfera do sistema *Ball on Plate* utilizando uma tela resistiva e um microcontrolador ESP32;
- Desenvolver um mecanismo de escalonamento das coordenadas adquiridas, a fim de converter os dados brutos em valores físicos (em milímetros);
- Integrar o algoritmo de controle tradicional com as leituras obtidas da tela resistiva, controlando a posição da esfera em tempo real;
- Estabelecer um sistema alternativo de detecção de posição baseado em visão computacional utilizando técnicas de segmentação por cor;
- Comparar o desempenho e a precisão entre os dois métodos de aquisição de posição (tela resistiva e visão computacional);
- Validar o sistema por meio de testes práticos, analisando sua resposta dinâmica e a efetividade do controle sobre a posição da esfera.

1.2 Organização do Trabalho

A estrutura desse trabalho divide-se em 6 capítulos, conforme descrito a seguir:

- **Capítulo 2 – Fundamentação teórica:** Discute os principais conceitos necessários para o entendimento do trabalho (Teoria de Controle e Visão Computacional);

- **Capítulo 3 – Desenvolvimento da planta didática *Ball on Plate*:** Apresenta de forma detalhada o desenvolvimento da planta didática, abordando desde o princípio de funcionamento e construção do sistema até a modelagem matemática e a implementação do controle tradicional;
- **Capítulo 4 – Controle do sistema *Ball on Plate* por visão computacional:** Apresenta o desenvolvimento do controle do sistema baseado em realimentação por visão computacional;
- **Capítulo 5 – Resultados experimentais e análise:** Apresenta os resultados obtidos ao longo do desenvolvimento do trabalho, bem como sua respectiva análise, destacando os diferentes métodos de realimentação empregados;
- **Capítulo 6 – Conclusões e propostas de continuidade:** Apresenta as conclusões do trabalho, abordando as principais dificuldades encontradas durante a implementação, conclusões finais, pontos de melhorias e propostas de continuidade.

2 FUNDAMENTAÇÃO TEÓRICA

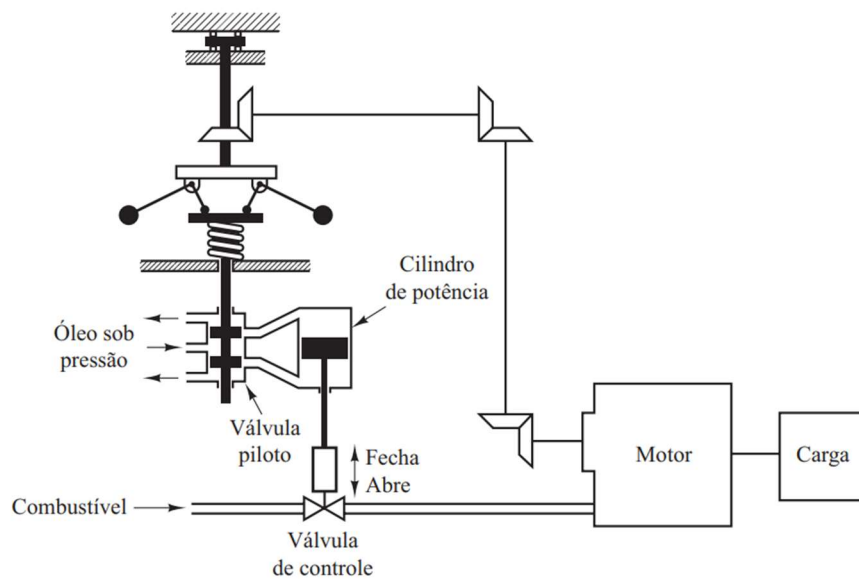
Neste capítulo, são apresentados os principais conceitos que fundamentam este trabalho. Inicialmente, aborda-se a teoria de controle e, em seguida, são introduzidos os conceitos de visão computacional, com ênfase nas técnicas de processamento de imagens utilizadas para a detecção e rastreamento de objetos, fundamentais para a implementação do controle baseado em câmera.

2.1 Teoria de Controle

O termo controle, utilizado na engenharia, corresponde a utilização de técnicas e métodos para controlar sistemas dinâmicos, garantindo respostas desejadas que atendam certos requisitos de desempenho previamente estabelecidos. Em outras palavras, o controle de sistemas manipula variáveis de entrada para produzir variáveis de saídas específicas, atendendo à critérios de projeto (NISE, 2012).

Apesar de ser aplicado nas tecnologias mais recentes e nos mais variados ramos (indústria, medicina, robótica, entre outros), o primeiro projeto nessa área não é tão atual, como afirma Ogata (2010): “O primeiro trabalho significativo de controle automático foi o regulador centrífugo construído por James Watt para o controle de velocidade de uma máquina a vapor, no século XVIII”. O princípio de um regulador de Watt, apresentado na Figura 1, baseia-se no controle da velocidade do motor, ajustando a diferença entre o valor de referência da velocidade e seu valor real. Com base nessa diferença, a quantidade de combustível fornecida ao motor é regulada para manter a velocidade desejada.

Figura 1 – Regulador de Watt.



Fonte: (OGATA, 2010).

A partir do regulador de James Watt, diversos avanços teóricos e práticos na disciplina foram observados, à medida que a complexidade de sistemas e processos cresciam. Entre os benefícios trazidos com o desenvolvimento dos conceitos de controle, destacam-se: automação de processos com alto desempenho, diminuição de custos e redução de mão de obra.

Nas subseções a seguir, são apresentados conceitos acerca do tema de engenharia de controle, com objetivo de fornecer os conhecimentos teóricos essenciais para compreensão do trabalho.

2.1.1 Modelagem de Sistemas de Controle

Um modelo dinâmico é uma representação matemática precisa da dinâmica de um sistema, que pode ser de natureza física, biológica ou de informação (ÅSTRÖM e MURRAY, 2021). Ele é utilizado para análises e simulações do comportamento do sistema em diferentes condições. Vale destacar que um mesmo sistema pode ser descrito por diferentes modelos, dependendo do nível de detalhamento, das variáveis consideradas e dos objetivos da modelagem. No caso de sistemas que obedecem a

leis físicas, a modelagem é frequentemente realizada por meio de equações diferenciais.

Um modelo pode ser obtido por meio da aplicação de princípios físicos (criando o que é chamado de modelo caixa-branca), ou de forma experimental (criando o que é chamado de modelo caixa-preta), em que dados são coletados a partir da aplicação de entradas conhecidas ao sistema, permitindo a análise da resposta de saída (FRANKLIN, POWELL e EMAMI-NAEINI, 2010).

É importante ressaltar a correspondência que existe entre simplicidade e precisão: quanto mais simples o modelo obtido, mais são as propriedades físicas ignoradas do sistema, o que pode causar problemas no desenvolvimento do controle (OGATA, 2010). Ao abordar sistemas lineares por meio de equações diferenciais, sempre são ignoradas as não-linearidades inerentes ao sistema, assumindo-se um comportamento aproximadamente linear dentro de uma faixa de operação. A essa técnica, dar-se o nome de linearização, e visa facilitar a modelagem e análise do sistema. Além disso, considera-se que o sistema possui parâmetros concentrados, ou seja, as propriedades como massa, resistência, indutância e capacitância são tratadas como concentradas em pontos específicos, em vez de distribuídas ao longo do sistema. Portanto, deve-se levar em consideração os efeitos que essas propriedades e não-linearidades ignoradas produzem sobre a resposta: se forem mínimos, então o modelo é utilizável, caso o contrário, deve-se levar em consideração na fase de modelagem as características desprezadas.

É apresentado a seguir como os sistemas são classificados e o método da modelagem no domínio da frequência.

2.1.1.1 *Classificação de Sistemas*

Na teoria de controle, os sistemas podem ser classificados quanto a sua linearidade, quanto ao número de entradas e saídas, em sua natureza temporal (contínuos ou discretos) e também quanto à sua dependência do tempo, podendo ser variantes ou invariantes.

Sistemas lineares são definidos como aqueles em que o princípio da superposição e o princípio da homogeneidade podem ser empregados (NISE, 2012),

ou seja, a resposta causada no sistema por duas entradas simultâneas pode ser calculada manipulando as duas entradas separadamente, calculando-se as duas respostas e somando-as. Matematicamente falando, sistemas lineares obedecem às equações:

$$y(t) = y_1(t) + y_2(t); \quad (2.1)$$

$$\text{Se } u(t) \Rightarrow y(t), \text{ então } k \cdot u(t) \Rightarrow k \cdot y(t), \quad (2.2)$$

em que:

- $y(t)$ representa a resposta completa do sistema;
- $y_1(t)$ representa a resposta particular devido à entrada 1;
- $y_2(t)$ representa a resposta particular devido à entrada 2;
- $u(t)$ é a entrada do sistema;
- k é uma constante qualquer.

Em contrapartida, os sistemas não-lineares não obedecem ao princípio da superposição e da homogeneidade, e, portanto, não podem ser representados pelas equações (2.1) e (2.2). O sistema mecânico do Pêndulo Invertido, por exemplo, é um sistema não-linear.

Na classificação dos sistemas de controle quanto ao número de entradas e saídas, um sistema de entrada e saída única (*Single Input Single Output*, SISO) é aquele que possui apenas uma entrada e uma única saída, sendo a abordagem no domínio da frequência o método mais utilizado para controlar essa classe de sistemas. A Figura 2 mostra um sistema SISO, onde $u(t)$ e $y(t)$ representam, respectivamente, a entrada e saída do sistema.

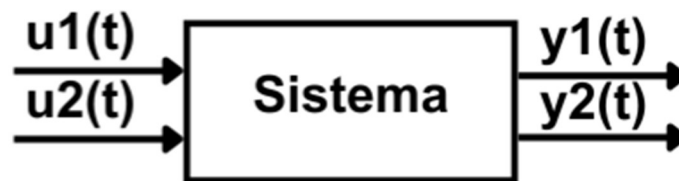
Figura 2 – Representação de um sistema SISO.



Fonte: O autor, 2025.

Por outro lado, um sistema de múltiplas entradas e múltiplas saídas (*Multiple Input Multiple Output*, MIMO) é aquele que possui mais de uma entrada e/ou mais de uma saída, sendo a abordagem no espaço estado o método mais utilizado para controlar essa classe de sistema. A Figura 3 representa um sistema MIMO, de duas entradas ($u1(t)$ e $u2(t)$) e duas saídas ($y1(t)$ e $y2(t)$).

Figura 3 – Sistema MIMO.

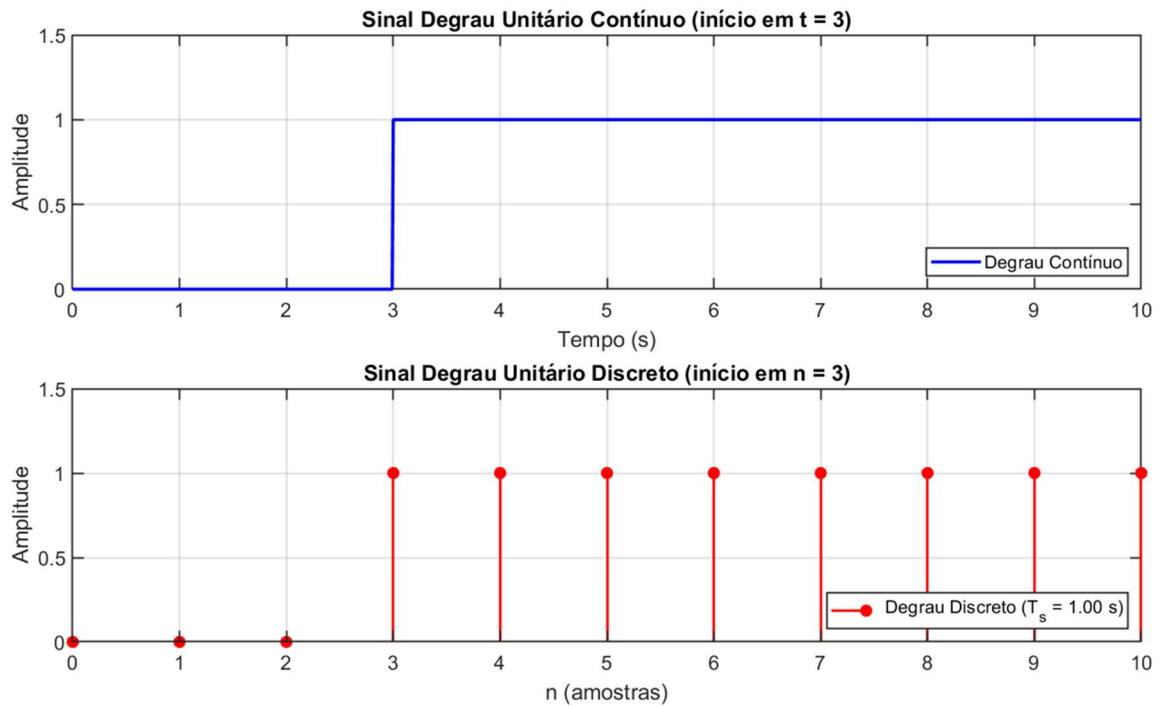


Fonte: O autor, 2025.

Um sistema de tempo contínuo é aquele em que os sinais manipulados estão definidos para qualquer instante dentro de um intervalo contínuo de tempo. Ou seja, suas entradas e saídas são especificadas para todo valor de tempo (t), onde t pertence a um conjunto contínuo de valores. Esses sistemas são geralmente modelados por equações diferenciais e são amplamente utilizados em processos dinâmicos que envolvem as leis da física.

Nos sistemas de tempo discreto, por outro lado, as entradas e saídas são definidas apenas em instantes específicos de tempo (amostras), normalmente espaçados por um período T , denominado período de amostragem. Esses sistemas são geralmente descritos por equações diferenças. A Figura 4 apresenta dois tipos de sinais degrau unitário. O primeiro é um sinal contínuo, definido no intervalo de tempo $t \in [0, 10]$ segundos. O segundo é um sinal discreto, definido em termos de um número finito de amostras n , com período de amostragem igual a 1 segundo. A comparação entre os dois permite observar as diferenças entre representações contínuas e discretas de um mesmo tipo de entrada.

Figura 4 – Sinal degrau unitário de tempo contínuo e de tempo discreto.



Fonte: O autor, 2025.

Sistemas invariantes no tempo são caracterizados por não possuírem parâmetros que dependem do tempo nas equações que descrevem sua dinâmica. Um circuito RLC, por exemplo, que possui resistor, capacitor, indutor e fonte de alimentação, pode ser modelado através da seguinte equação:

$$L \cdot \frac{d^2 i(t)}{dt^2} + R \cdot \frac{di(t)}{dt} + \frac{1}{C} \cdot i(t) = v(t), \quad (2.3)$$

onde, os parâmetros L, R e C são constantes.

Sistemas variantes no tempo são aqueles que possuem parâmetros que não são constantes, ou seja, nas equações que descrevem sua dinâmica, os coeficientes dependem do tempo. Um circuito elétrico RC com resistência variável, por exemplo, pode ser modelado através da seguinte equação diferencial linear variante ao tempo:

$$R(t) \cdot C \frac{dV_c(t)}{dt} + V_c(t) = V_{in}(t), \quad (2.4)$$

em que o parâmetro R não é constante.

2.1.1.2 Modelagem no Domínio da Frequência

O método de modelagem no domínio da frequência é realizado por meio de função de transferência (FT). A principal vantagem da utilização desse método é que ele substitui a solução mais difícil de equações diferenciais por uma solução algébrica mais fácil no domínio da frequência, através da transformada de Laplace (DORF e BISHOP, 2001). Em geral, um sistema físico que pode ser representado por uma equação diferencial linear invariante no tempo pode ser modelado como uma FT (NISE, 2012).

A transformada de Laplace é definida através da equação:

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{\infty} f(t) \cdot e^{-st} dt, \quad (2.5)$$

onde:

- $f(t)$ é a função no domínio do tempo;
- $F(s)$ é a função transformada no domínio de s ;
- s é uma variável complexa definida como $s = \sigma + j\omega$.

Com isso, pode-se transformar uma função no domínio do tempo, $f(t)$, para sua correspondente no domínio de Laplace, $F(s)$, contanto que a integral exista e convirja. A transformada Inversa de Laplace permite obter $f(t)$ a partir de $F(s)$ e é definida como:

$$f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi j} \int_{\sigma-j\omega}^{\sigma+j\omega} F(s) \cdot e^{st} ds. \quad (2.6)$$

De acordo com Esfandiari e Lu (2010), uma FT é definida como a razão entre a transformada de Laplace da saída e a transformada de Laplace da entrada do sistema, assumindo que todas as condições iniciais são nulas. Essa definição é expressa na equação abaixo:

$$FT = G(s) = \frac{\mathcal{L}[saída]}{\mathcal{L}[entrada]}. \quad (2.7)$$

A forma base de uma FT é mostrada a seguir:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 \cdot s^m + b_1 \cdot s^{m-1} + \dots + b_{m-1} \cdot s + b_m}{a_0 \cdot s^n + a_1 \cdot s^{n-1} + \dots + a_{n-1} \cdot s + a_n}. \quad (2.8)$$

As raízes do denominador são os polos do sistema, enquanto que as raízes do numerador são os zeros do sistema. A função de transferência é amplamente utilizada na modelagem e análise de sistemas dinâmicos. Ela permite efetuar a análise de estabilidade do sistema, além de determinar sua resposta a diferentes tipos de sinais de entrada, como impulso, degrau e rampa. A FT da planta pode ser obtida diretamente da aplicação de um impulso, analisando-se a resposta do sistema. Com isso, é possível determinar o comportamento do sistema em malha aberta e em malha fechada. O método da modelagem no domínio da frequência, utilizando a representação por FT, é aplicado a sistemas lineares invariantes ao tempo de entrada e saída única.

2.1.2 Especificações de Projeto

De acordo com Dorf e Bishop (2001), medidas de desempenho padrões de projetos de sistemas de controle são comumente definidas em termos da resposta transitória do sistema em malha fechada a uma entrada do tipo degrau. Em geral, a entrada do tipo degrau unitário é a mais utilizada em razão da sua facilidade em ser gerada.

A resposta de um sistema a uma entrada do tipo degrau unitário depende também das condições iniciais do sistema, e dessa forma geralmente é considerado o sistema em repouso, ou seja, com condições iniciais nulas, para comparar as medidas de desempenho do sistema (OGATA, 2010).

Dessa forma, os projetistas de sistemas de controle devem atender a determinados requisitos de projeto, conhecidos como especificações de desempenho, os quais definem faixas ou valores desejados para o comportamento dinâmico do sistema, garantindo que ele atenda aos critérios estabelecidos de estabilidade, rapidez, precisão e robustez.

Dentre as medidas de desempenho mais comuns, destacam-se os listados abaixo:

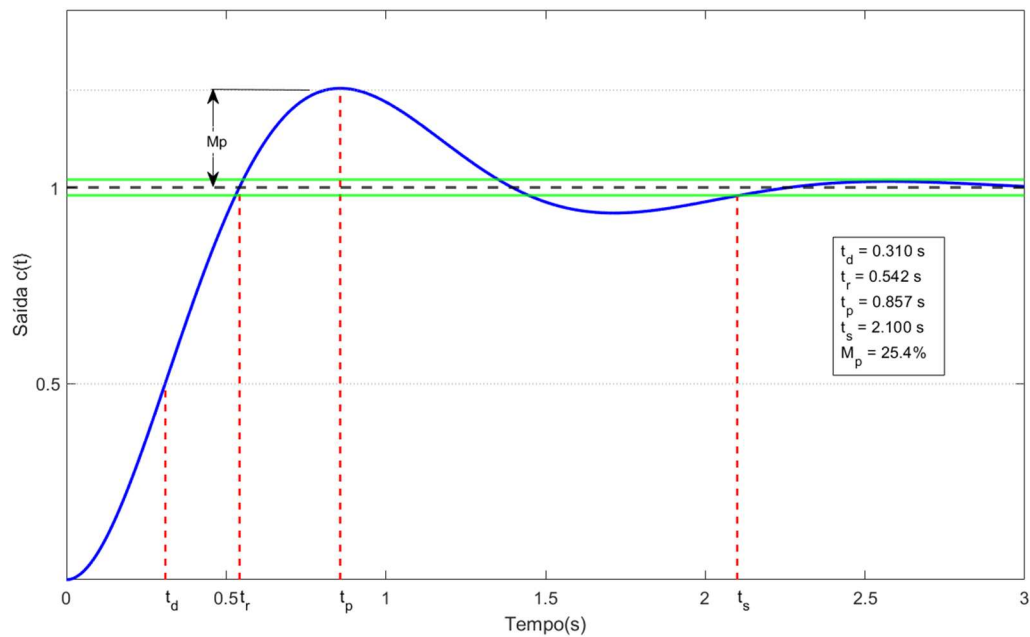
- **Tempo de pico (t_p):** é o tempo necessário para a resposta atingir o seu valor máximo;
- **Sobressinal máximo percentual (M_p):** é definido como o valor máximo de pico da curva de resposta medido a partir da unidade, para quando a resposta em regime permanente for igual a 1. Para os casos mais diversos é definido através da equação:

$$M_p = \frac{c(tp) - c(\infty)}{c(\infty)} \cdot 100\%, \quad (2.9)$$

onde $c(tp)$ representa o valor da saída no instante de pico e $c(\infty)$ o valor em regime permanente;

- **Tempo de atraso (t_d):** é o tempo em que a resposta do sistema leva para atingir a metade de seu valor em regime permanente, pela primeira vez;
- **Tempo de subida (t_r):** é o intervalo de tempo que a resposta leva para cruzar um determinado limite inferior e um determinado limite superior, pela primeira vez. Os critérios mais utilizados são os de 10-90% e 0-100% do valor final;
- **Tempo de assentamento (t_s):** é o tempo necessário para que a resposta atinja e permaneça dentro de uma determinada faixa de tolerância em torno do valor final, sem mais ultrapassá-la. Os critérios mais utilizados para essa faixa são de $\pm 2\%$ e $\pm 5\%$ do valor de regime permanente;
- **Erro estacionário:** é a diferença entre a entrada desejada e a saída do sistema quando o tempo tende ao infinito. Em sistemas de controle, geralmente se deseja que a saída acompanhe a entrada com precisão, e o erro estacionário quantifica o desvio persistente entre ambas. Um sistema pode apresentar erro nulo para determinados tipos de entrada (como degrau), mas ainda assim exibir erro distinto de zero para outros tipos de sinais, como rampa ou parábola.

A Figura 5 representa as especificações de desempenho para uma resposta de segunda ordem subamortecida.

Figura 5 – Especificações de projeto ($\zeta = 0.4$, $\omega_n = 4.0$).

Fonte: O autor, 2025.

2.1.3 Análise de Estabilidade

A estabilidade de um sistema de controle é um tema muito importante, pois caso um sistema seja instável em malha fechada, as definições já vistas de caracterização de respostas e especificações de projeto não são aplicáveis. Ainda de acordo com Nise (2012), um sistema instável pode causar danos ao sistema, às instalações adjacentes e à vida humana, em razão de que as repostas desse tipo de sistema crescem ilimitadamente.

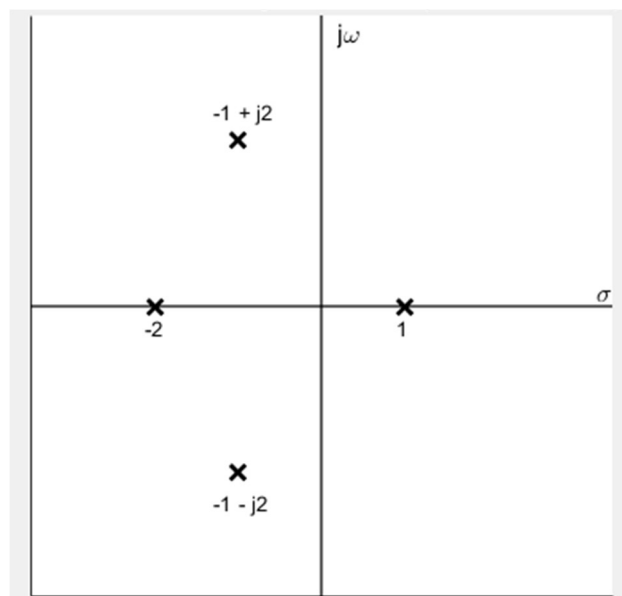
Dessa forma, de acordo com Dorf e Bishop (2001), a estabilidade de um sistema pode ser definida da seguinte maneira: um sistema é considerado estável se, ao ser submetido a uma entrada limitada, sua resposta também permanece limitada em magnitude. Por outro lado, um sistema instável gera respostas ilimitadas mesmo quando submetidos a entradas limitadas. Ainda segundo os autores, existem dois tipos de estabilidade: a estabilidade absoluta, que se refere diretamente a definição mencionada anteriormente, e a estabilidade relativa, que avalia o grau de estabilidade do sistema, ou seja, o quão próximo ele está da instabilidade. O termo estabilidade, tratado neste trabalho, refere-se exclusivamente à estabilidade absoluta. Sempre que

for necessário tratar da estabilidade relativa, essa será devidamente especificada e descrita em detalhes.

Em termos da função de transferência do sistema, pode-se definir estabilidade, segundo Dorf e Bishop (2001), da seguinte forma: “uma condição necessária e suficiente para um sistema com retroação ser estável é que todos os polos da função de transferência do sistema tenham parte real negativa”, ou seja, estejam no semiplano esquerdo do plano s . Quando o sistema possui pólos simples sobre o eixo imaginário ($j\omega$), ele é classificado como marginalmente estável, pois sua resposta é ilimitada apenas para alguns tipos de entradas limitadas. Caso existam polos com multiplicidade maior que 1 sobre o eixo imaginário, o sistema torna-se instável, pois apresenta crescimento ilimitado das respostas mesmo para entradas limitadas. Por fim, um sistema é considerado instável sempre que possuir pelo menos um polo no semiplano direito do plano s , ou seja, com parte real positiva.

A Figura 6 ilustra a localização dos polos ($s = \sigma \pm j\omega$) de um sistema linear instável no plano s . Conforme pode ser observado, um dos pólos está localizado no semiplano direito de s , ou seja, com parte real positiva, o que, de acordo com os critérios de estabilidade apresentados, caracteriza um sistema instável.

Figura 6 – Localização dos pólos de um sistema instável no plano s .



Fonte: O autor, 2025.

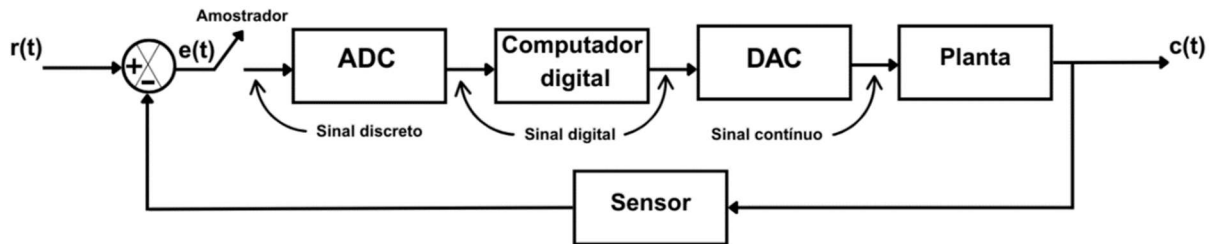
2.1.4 Sistemas de Controle Digitais e Discretização

O controle digital, segundo Dorf e Bishop (2001), refere-se à utilização de computadores digitais para desempenhar a função de controlador em sistemas de controle com realimentação. Nos últimos anos, os avanços tecnológicos possibilitaram o aumento da capacidade computacional, como o desenvolvimento das unidades centrais de processamento (*Central Processing Units*, CPUs) e o aumento da memória, além da diminuição do custo e do tamanho desses componentes. Como resultado, microprocessadores e microcontroladores modernos são capazes de executar, com alta precisão e confiabilidade, algoritmos de controle em tempo real mesmo em sistemas dinâmicos complexos. O controle digital é amplamente utilizado em processos industriais, sistemas de aviação, gerenciamento de tráfego, robótica, automóveis e sistemas embarcados.

Segundo Nise (2012), as principais vantagens do controle digital em relação ao controle analógico, implementado por componentes físicos analógicos, incluem o menor custo, maior flexibilidade para ajustes e modificações no projeto, além de uma maior imunidade a ruídos. Portanto, os controladores digitais são capazes de realizar cálculos complexos com alta precisão e velocidade, de maneira eficiente e com um custo relativamente baixo.

A principal diferença em um sistema de controle digital, ilustrado na Figura 7, está na forma como os sinais são processados. Inicialmente, o sinal analógico contínuo é amostrado no tempo, resultando em um sinal de tempo discreto. Em seguida, cada amostra é quantizada e codificada por um conversor analógico-digital (ADC). O controlador digital então processa esses dados por meio de algoritmos, gerando sinais de controle também digitais. Para que o sinal de controle discreto possa ser aplicado à planta, é necessário convertê-lo de volta para o domínio do tempo contínuo por meio de um conversor digital-analógico (DAC).

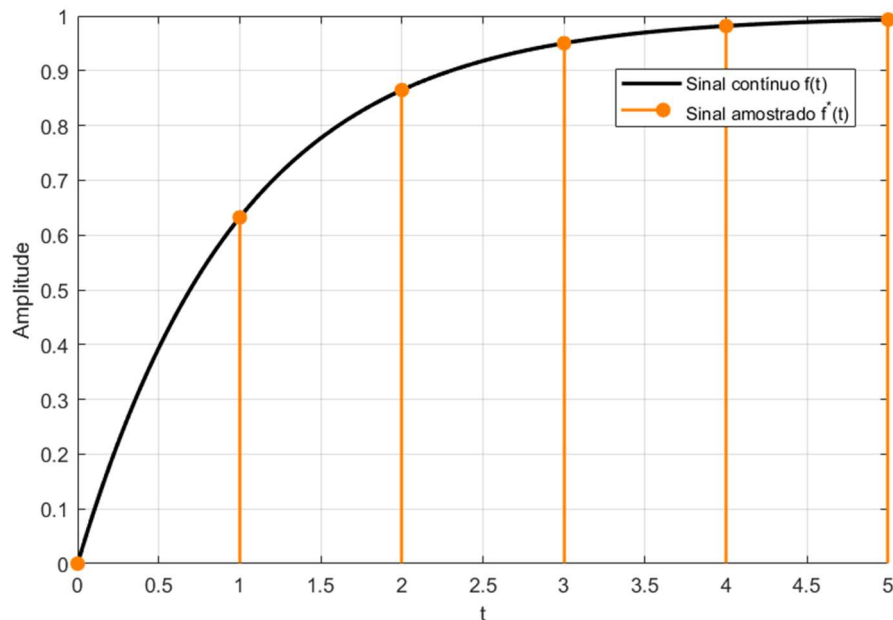
Figura 7 – Sistema de controle digital.



Fonte: Adaptado de (TORRES, 2023).

Um amostrador pode ser entendido como uma chave que se fecha a cada T segundos, extraíndo o valor da variável de tempo contínuo naquele instante específico (DORF e BISHOP, 2001). Como mostrado na Figura 8, a saída do amostrador é uma sequência de impulsos espaçados por um período T . O parâmetro T é chamado período de amostragem.

Figura 8 – Saída do amostrador para uma entrada contínua $u(t) = 1 - e^{-t}$, com período de amostragem $T = 1s$.



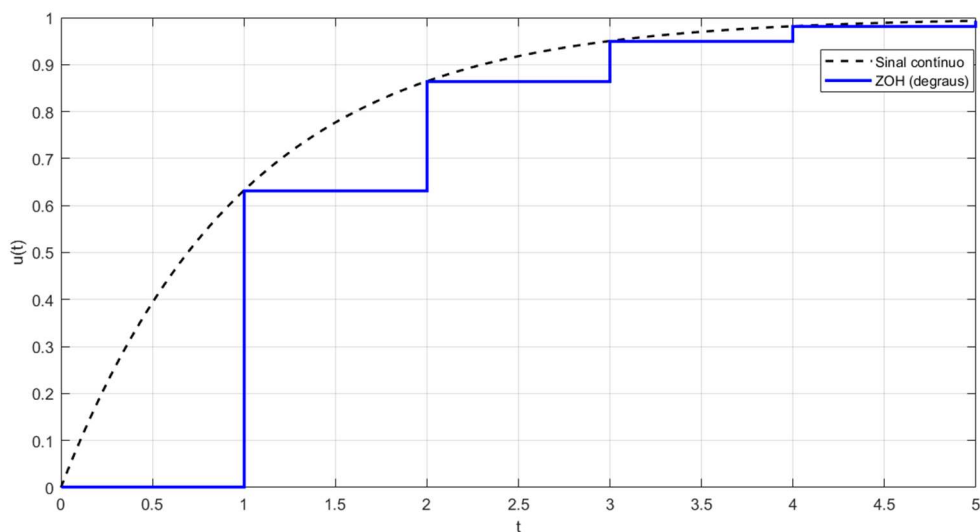
Fonte: O autor, 2025.

A Quantização é o processo de aproximar cada valor amostrado para o mais próximo dentro de um conjunto finito de níveis, determinado pela resolução do ADC. A codificação atribui uma palavra digital (geralmente em formato binário) a cada valor

quantizado, possibilitando sua manipulação pelo sistema digital. Como a quantização representa uma aproximação do valor real da variável analógica, a diferença entre o valor original e o valor quantizado é denominada erro de quantização. Na maioria das vezes, esse erro é desprezado por ser muito pequeno em relação à amplitude do sinal (DORF e BISHOP, 2001).

Um DAC é comumente modelado por um retentor de ordem zero (ZOH – *Zero-Order Hold*) (DORF e BISHOP, 2001), que atua mantendo constante o valor de cada amostra durante o intervalo entre amostragens. Especificamente, o ZOH sustenta o valor da amostra $u(kT)$ ao longo do intervalo $kT \leq t < (k+1)T$, gerando um sinal contínuo em tempo com degraus. A Figura 9 apresenta a saída de um retentor de ordem zero (ZOH) utilizada para reconstruir o sinal discreto ilustrado na Figura 8. Para fins de comparação, o sinal contínuo original é exibido em tracejado.

Figura 9 – Saída de um retentor de ordem zero (ZOH).



Fonte: O autor, 2025.

Um fator importante a ser considerado, na implementação do controle digital, é que a frequência de amostragem deve ser, no mínimo, o dobro da maior frequência presente no sinal analógico, conforme estabelece o Teorema de *Nyquist*. Caso contrário, ocorre um fenômeno conhecido como *aliasing*, onde diferentes componentes de frequência se sobrepõem tornando-se indistinguíveis, resultando em um sinal distorcido que compromete sua reconstrução no tempo contínuo. Além disso,

para o correto funcionamento do ADC, é necessário realizar o condicionamento do sinal analógico antes da conversão. As etapas mais comuns incluem amplificação ou atenuação, para adequar a amplitude do sinal à faixa de entrada do ADC, e filtragem *anti-aliasing*, que elimina componentes de alta frequência que podem causar distorções. Essas ações garantem que o sinal respeite os limites do conversor e preserve a resolução e fidelidade na digitalização.

Analogamente aos sistemas de tempo contínuo, que utilizam a transformada de Laplace, os sistemas de tempo discreto fazem uso da transformada Z, representada pela equação abaixo:

$$Z\{u(kT)\} = U(z) = \sum_{k=0}^{\infty} u(kT) \cdot z^{-k}. \quad (2.10)$$

A transformada Z permite obter a função de transferência no domínio Z a partir das equações diferenças, desempenhando, no contexto discreto, um papel equivalente ao da transformada de Laplace no domínio contínuo.

A variável z é definida da seguinte forma:

$$z = e^{sT}, \quad (2.11)$$

em que s é a variável complexa da transformada de Laplace e T é o período de amostragem (DORF e BISHOP, 2001). Portanto, a equação (2.11) define um mapeamento do plano s no plano z .

Um controlador discreto pode ser projetado por diversas abordagens, como o projeto direto no domínio z , o projeto de controladores PID diretamente em tempo discreto, o controle ótimo discreto, entre outros. Dentre essas metodologias, destaca-se o projeto via equivalente discreto, descrito por Franklin, Powell e Emami-Naeini (2010), que segue os seguintes passos:

1. Projetar um controlador no tempo contínuo, utilizando técnicas clássicas no domínio do tempo, como lugar das raízes e resposta em frequência;
2. Digitalizar o controlador contínuo;
3. Utilizar a análise discreta, simulada ou experimentada, para verificar o projeto.

Essa abordagem é amplamente utilizada por aproveitar as ferramentas consolidadas do projeto contínuo e permitir a implementação digital do controlador resultante.

2.1.4.1 Técnica de Discretização de Tustin

Uma das abordagens mais utilizadas para a discretização de controladores projetados no domínio contínuo é o método de Tustin, também chamado de transformação bilinear. Essa técnica utiliza a operação de integração para abordar a implementação do controlador digital (FRANKLIN, POWELL e EMAMI-NAEINI, 2010). Dessa forma, a integração contínua do erro $e(t)$ é aproximada como mostrado nas equações:

$$u(kT) = \int_0^{KT-T} e(t) dt + \int_{KT-T}^{KT} e(t) dt, \quad (2.12)$$

e

$$u(kT) = u(KT - T) + \text{área sob } e(t) \text{ sobre o último } T. \quad (2.13)$$

Para aproximar a área mais recente, o método de Tustin utiliza a regra do trapézio, que consiste em estimar a integral pela média dos valores da função nos dois instantes de amostragem consecutivos:

$$u(k) = u(K - 1) + \frac{T}{2} \cdot [e(k - 1) + e(k)]. \quad (2.14)$$

Aplicando a transformada Z nessa equação, obtém-se a função de transferência do integrador:

$$\frac{U(Z)}{E(Z)} = \frac{T}{2} \cdot \left(\frac{1+Z^{-1}}{1-Z^{-1}} \right). \quad (2.15)$$

A partir do resultado, é possível derivar a substituição que relaciona as variáveis nos domínios contínuo (s) e discreto (z):

$$s = \frac{z}{T} \cdot \left(\frac{1-z^{-1}}{1+z^{-1}} \right). \quad (2.16)$$

O método de Tustin tem a vantagem de sempre mapear um controlador estável, no domínio de s , em um controlador estável no domínio de z , preservando a estabilidade do sistema nos dois casos.

2.2 Visão Computacional (Computer Vision, CV)

De acordo com Szeliski (2010), CV é o processo de descrever o mundo visível a partir de uma ou mais imagens e reconstruir suas propriedades como forma, iluminação e distribuição de cores. Para isso, são utilizados algoritmos baseados em modelos físicos e técnicas probabilísticas.

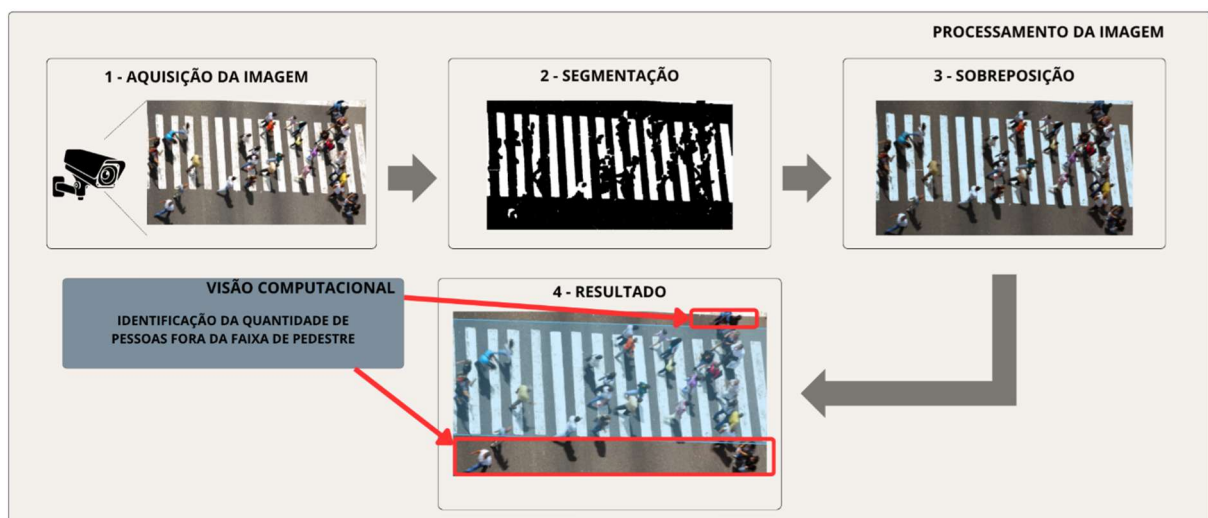
Já Gonzalez e Woods (2018) definem CV como o processo de usar computadores para emular a visão humana, permitindo que sistemas aprendam, façam inferências e tomem decisões a partir de imagens ou vídeos. Segundo os autores, não há uma divisão clara entre os temas como processamento de imagens e CV. Um paradigma útil, descrito por eles, é definir o processamento de imagens em três níveis: baixo, médio e alto.

O processamento de baixo nível diz respeito a operações primitivas, como o pré-processamento de uma imagem para redução de ruído. Nesse caso, a entrada e a saída são imagens. Já o processamento de nível médio envolve a extração de atributos a partir da imagem, como detecção de bordas, segmentação ou reconhecimento de objetos. Nesse caso, a entrada é uma imagem e a saída é um conjunto de características. Por fim, o processamento de nível superior, o mais próximo da CV, envolve dar sentido a um conjunto de objetos reconhecidos, como por exemplo, detectar se um humano está correndo ou parado a partir da imagem, ou ainda detectar ações suspeitas em um sistema de vigilância.

Dessa forma, assim como abordado por Marengoni e Stringhini (2009), as operações realizadas diretamente sobre as imagens, como filtragens, mudanças na forma da imagem, segmentações e realces, caracterizam o processamento de imagem. Já a CV representa o nível mais alto, sendo responsável por interpretar os

dados extraídos das imagens processadas. Seu objetivo é extrair informações significativas e tomar decisões ou realizar ações automatizadas com base nessa interpretação, aproximando-se da forma como os seres humanos agem. Assim, o processamento prepara a imagem e a CV entende e age. Como exemplo, a Figura 10 ilustra o fluxo de processamento de uma imagem de uma faixa de pedestres. Na etapa 1, a imagem é extraída por uma câmera. Em seguida, na etapa 2, são aplicadas operações de segmentação, utilizando o filtro HSV para detectar as faixas brancas com base na diferença de cor. Na etapa 3, realiza-se a sobreposição dos contornos detectados à imagem original. O resultado final, mostrado na etapa 4, é então interpretado por um sistema de CV, que identifica e contabiliza os pedestres fora da faixa, possibilitando, por exemplo, ações automáticas como alertas ou aplicação de penalidades.

Figura 10 – Exemplo de fluxo de visão computacional com detecção de pedestres.



Fonte: O autor, 2025.

De acordo com Szeliski (2010), um dos marcos iniciais da CV foi no ano de 1966, com um desafio proposto em um verão aos universitários do MIT, em que os alunos deveriam conectar um computador a uma câmera, e deveria fazer com que o computador descrevesse o que via, encontrando grande dificuldade por parte dos alunos. Ainda, segundo o autor:

- Entre 1970 e 1980 os trabalhos em CV se concentraram na inferência de estruturas tridimensionais a partir de imagens com linhas bidimensionais, utilizando técnicas geométricas e baseadas em operações básicas de imagem, como a detecção e extração de bordas;
- Entre as décadas de 1980 e 1990, técnicas matemáticas mais sofisticadas foram desenvolvidas para realizar análises quantitativas de imagens e cenas. Um dos avanços marcantes foi o uso das pirâmides de imagens para análise em múltiplas escalas. Também foram observados avanços teóricos na reconstrução 3D e segmentação;
- A partir de 1990, abordagens mais estatísticas e baseadas em aprendizado de máquina ganharam destaque, melhorando as áreas de reconstrução tridimensional, segmentação, rastreamento de objetos e reconhecimento facial. A visão baseada em física e a fusão com a computação gráfica (área que se concentra na criação e renderização de imagens sintéticas, baseando-se em modelos geométricos e físicos) ampliaram as aplicações e o realismo nas reconstruções visuais;
- Dos anos 2000 em diante, a CV se aproximou fortemente da computação gráfica, originando a fotografia computacional, uma área que aplica algoritmos avançados para aprimorar ou expandir as capacidades da fotografia tradicional. Avanços significativos ocorreram em síntese de texturas, reconhecimento de objetos baseado em características, otimização global e aprendizado de máquina com grandes bases de dados. Essas tendências transformaram tanto a pesquisa quanto as aplicações práticas da área.

A partir de 2012, com o avanço dos recursos computacionais e a evolução das tecnologias, a área de CV passou por grandes transformações, impulsionadas especialmente pelo desenvolvimento do *deep learning*, das redes neurais e, em particular, das redes neurais convolucionais (CNNs). Uma série de avanços foram feitos em diversas áreas envolvendo classificação de imagens, segmentação e detecção de objetos com o rápido desenvolvimento das CNNs (ZHAO, 2024). Modelos mais recentes de detecção de objetos em tempo real, como *You Only Look Once* (YOLO) e *Faster Region Convolutional Neural Network* (*Faster R-CNN*), são usados

nas mais diversas áreas, como automação industrial, construção civil, saúde, veículos autônomos, além de outros (RANE, 2023).

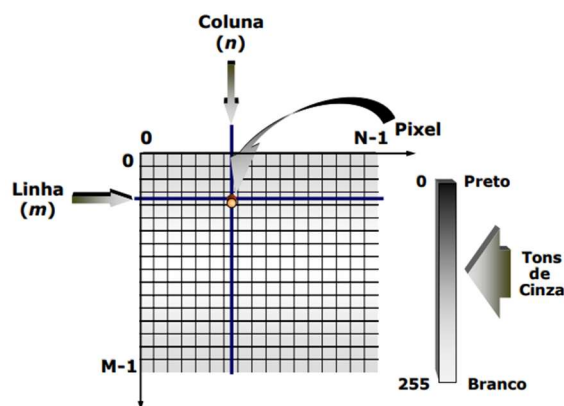
2.2.1 Etapas da Visão Computacional

Para compreender o fluxo de uma imagem em aplicações de CV, é essencial entender o que caracteriza uma imagem. Segundo Gonzalez e Woods (2018), uma imagem é uma função bidimensional $f(x, y)$, onde x e y são coordenadas espaciais, e o valor de f representa o nível de cinza naquele ponto. No caso de imagens digitais, tanto as coordenadas espaciais quanto os valores de amplitude são discretos e finitos. Como descreve Neto (2020), uma imagem digitalizada é uma matriz onde os índices de linhas e colunas identificam um ponto na imagem, e o valor correspondente determina o brilho médio amostrado.

Um computador digital realiza o processamento dessas imagens através da manipulação dos pixels, definidos como as menores unidade da matriz que representa uma imagem digital. Os pixels são finitos e sua unidade é representada pela posição (x, y) e pelo valor de intensidade do pixel.

A Figura 11 ilustra esse conceito, mostrando uma imagem composta por M linhas e N colunas, com destaque para a representação de um pixel individual. A resolução da imagem é determinada pela quantidade de colunas (N) por linhas (M), sendo expressa na forma NxM.

Figura 11 – Representação de uma imagem digital bidimensional.



Fonte: (DE QUEIROZ e GOMES, 2001) .

Conforme abordado por De Queiroz e Gomes (2001), o processamento digital de imagem segue as etapas abaixo:

- **Aquisição:** é o processo de captura da imagem, que envolve os mecanismos de aquisição e digitalização para manipulação em computador digital;
- **Pré-processamento:** envolve passos como filtragem de ruídos introduzidos na etapa de aquisição, redimensionamento e melhoria de contraste. O pré-processamento corresponde as operações básicas feitas na imagem para sua melhor manipulação;
- **Extração de características:** etapa de extração de bordas, formato, cor, dentre outros atributos em uma imagem que identificam um determinado objeto de interesse;
- **Segmentação:** é a etapa onde o objeto é separado do plano de fundo da imagem, através de características que permanecem constantes no objeto de interesse e que possui variação abrupta quando encontra uma região diferente da do objeto;
- **Classificação:** é o processo que, a partir dos resultados das etapas anteriores, permite identificar e reconhecer objetos

Dessa forma, é muito importante ressaltar as operações mais comuns feitas no processamento digital da imagem.

2.2.2 Operações Sobre as Imagens no Domínio Espacial

Uma operação no domínio espacial de uma imagem é definida como um processo aplicado diretamente sobre os pixels que a compõem. Segundo Marengoni e Stringhini (2009), esse tipo de operação pode ser representado pela equação:

$$g(x, y) = T(f(x, y)), \quad (2.17)$$

em que $f(x, y)$ é a imagem original, T é a função de transformação aplicada e $g(x, y)$ corresponde à imagem resultante após a transformação.

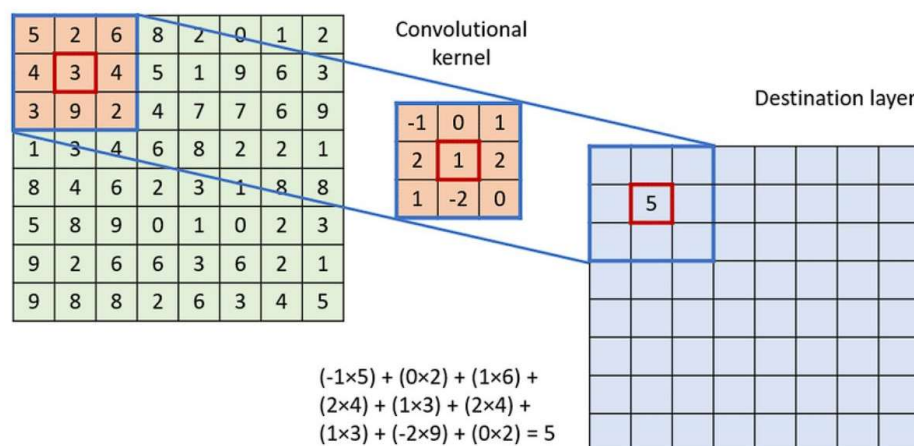
2.2.2.1 Filtragem de Suavização da Imagem

Em uma imagem digital, regiões homogêneas apresentam pouca variação nos níveis de cinza e são associadas a áreas de baixa frequência, enquanto as bordas, que delimitam objetos ou indicam transições de iluminação, correspondem a áreas de alta frequência, por apresentarem variações bruscas de intensidade (DE QUEIROZ e GOMES, 2001).

Para reduzir ruídos e suavizar essas variações abruptas, utilizam-se filtros espaciais, cuja operação altera o valor de um pixel com base nos valores dos pixels em sua vizinhança, geralmente definida por uma região quadrada de tamanho ímpar (MARENGONI e STRINGHINI, 2009).

Essa técnica se baseia na convolução de uma máscara (ou *kernel*) com a imagem, sendo aplicada localmente em cada ponto da matriz de pixels. A operação consiste em sobrepor a máscara sobre uma vizinhança do pixel central, multiplicar cada elemento da máscara pelo valor correspondente do pixel na imagem, somar os resultados e atribuir esse valor ao pixel de saída (GONZALEZ e WOODS, 2018). A Figura 12 ilustra esse processo de forma esquemática.

Figura 12 – Ilustração da operação de convolução entre uma imagem digital e um kernel 3×3.



Fonte: Adaptado de (VISO, 2024).

Dentre os filtros mais comuns estão os da média, gaussiano, mediana e moda, utilizados para suavização e redução de ruídos preservando, em diferentes graus, os contornos das imagens.

O filtro da média de ordem n substitui cada pixel pelo valor da média aritmética dos pixels em sua vizinhança definida por uma máscara (ou *kernel*), suavizando gradualmente as bordas. Quanto maior a máscara, maior o grau de suavização aplicado (DE QUEIROZ e GOMES, 2001).

O filtro Gaussiano é um filtro de suavização do tipo passa-baixa cujos valores de máscara são definidos por uma função Gaussiana bidimensional discreta, centrada na origem e com desvio padrão σ . Diferentemente do filtro da média, ele aplica uma ponderação simétrica, dando maior importância aos pixels mais próximos do centro da máscara (MARENGONI e STRINGHINI, 2009). Isso resulta em uma suavização mais natural, com menor distorção nas bordas da imagem.

O filtro da mediana, por sua vez, substitui o pixel central pelo valor mediano dos pixels na vizinhança. Esse método é altamente eficaz na remoção de ruído impulsivo, e possui a vantagem de preservar bordas melhor que o filtro da média (GONZALEZ e WOODS, 2018).

Por fim, o filtro da moda atribui ao pixel central o valor mais frequente (moda) entre os vizinhos.

2.2.2.2 Limiarização (*Thresholding*)

A técnica de limiarização (ou binarização) é amplamente utilizada na segmentação de imagens devido à simplicidade de sua implementação e à rapidez no processamento computacional (GONZALEZ e WOODS, 2018). Esse método consiste em estabelecer um valor de limiarização T , o qual é comparado individualmente ao valor de cada pixel da imagem. Os pixels que possuem intensidade maior que T recebem o valor 1, enquanto aqueles com intensidade igual ou inferior recebem o valor 0. Como resultado, a imagem original é transformada em uma imagem binária, contendo apenas dois níveis distintos de intensidade: 0 e 1. Matematicamente, isso é representado por:

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) > T \\ 0, & \text{se } f(x, y) \leq T \end{cases} \quad (2.18)$$

Essa técnica é especialmente adequada em situações onde a imagem apresenta grupos distintos de pixels com níveis de cinza bem definidos, como um objeto claro contrastando com um fundo escuro.

A abordagem descrita pela equação (2.18) é conhecida como limiarização global, por utilizar um único valor fixo de T aplicado a toda a imagem. Entretanto, há métodos alternativos, como a limiarização adaptativa, onde o valor de T varia conforme a localização espacial dos pixels, permitindo uma segmentação mais precisa em imagens com variações de iluminação ou contraste.

2.2.2.3 Operações Morfológicas

As operações morfológicas no processamento digital de imagens têm como objetivo extrair componentes estruturais relevantes, úteis para representar e descrever formas de regiões, como bordas, esqueletos e fechos convexos (GONZALEZ e WOODS, 2018). Fundamentadas na teoria dos conjuntos, essas operações são, inicialmente, aplicadas em imagens binárias, onde os objetos que se destacam do fundo podem ser representados apenas por suas coordenadas espaciais (x, y) , ou seja, como subconjuntos do espaço discreto $\mathbf{Z}^2$, onde $\mathbf{Z}$ representa o conjunto dos números inteiros.

Nessa abordagem, são definidos dois subconjuntos principais: um que representa o objeto da imagem e outro correspondente ao elemento estruturante (*structuring element*, SE), cuja função é "varrer" os pixels da imagem, aplicando a operação morfológica em cada posição. As operações mais fundamentais são a erosão e a dilatação.

Na erosão, o elemento estruturante é deslocado sobre a imagem, e em cada posição verifica-se se ele está completamente contido no objeto. Se a estrutura do SE "cabe" dentro da forma do objeto naquela região, o ponto central da máscara é mantido como pixel do objeto; caso contrário, ele é descartado. Essa operação tende a reduzir o tamanho dos objetos, funcionando como um filtro morfológico que remove detalhes finos, ruídos e bordas estreitas (MARENGONI e STRINGHINI, 2009). Assim,

objetos menores que o elemento estruturante são eliminados, tornando o seu tamanho e forma parâmetros críticos no processo.

Já na dilatação, o SE também percorre a imagem, mas a lógica é oposta: se qualquer parte do elemento estruturante sobreposto à imagem coincidir com o objeto, o ponto central da máscara é mantido como parte do objeto. Isso faz com que a operação expanda os objetos, preenchendo pequenos buracos, conectando regiões fragmentadas e suavizando falhas causadas por ruídos ou perdas de dados.

2.2.2.4 *Detecção de Bordas*

Como abordado por Dharampal (2015), a detecção de bordas pode ser definida como o processo de identificar linhas que delimitam e separam regiões distintas em uma imagem digital, marcando os limites entre objetos ou áreas com diferentes características visuais. Essa operação desempenha um importante papel no processamento digital de imagens, sendo usada para segmentação, reconhecimento de objetos e aprimoramento visual (GANESAN e SAJIV, 2017).

A detecção de bordas se baseia na premissa de que variações abruptas de intensidade ou níveis de cinza entre pixels indicam a presença de uma borda. Tais variações são mapeadas por meio da aplicação de máscaras (kernels) que, ao serem convoluídas com a imagem, capturam essas transições. Duas abordagens tradicionais principais são destacadas nesse contexto: a baseada em gradiente e a baseada no Laplaciano (DHARAMPAL, 2015).

Na primeira, as bordas são detectadas utilizando a primeira derivada da imagem, representada pelo vetor gradiente. A magnitude do gradiente fornece uma medida da intensidade da borda, enquanto sua direção revela a orientação da transição. O vetor gradiente de uma imagem bidimensional é expresso por:

$$\nabla f = G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{pmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{pmatrix}. \quad (2.19)$$

A magnitude do gradiente, que representa a força da borda, é calculada por:

$$M(x, y) = \text{magnitude}(\nabla f) = \sqrt{G_x^2 + G_y^2}, \quad (2.20)$$

onde G_x e G_y representam o gradiente nas direções horizontal e vertical, respectivamente.

Para obter esses valores, realizam-se convoluções da imagem com máscaras derivativas (como os operadores de Sobel ou Prewitt) que percorrem a imagem utilizando janelas 3×3 . Em cada posição, a máscara é multiplicada elemento a elemento pela janela da imagem e o somatório desses produtos fornece o valor de G_x ou G_y para o pixel central da janela. O resultado final é uma imagem de intensidade de bordas.

Entre os métodos mais conhecidos baseados no gradiente, destacam-se os operadores de Roberts, Prewitt, Sobel e Canny, cada um utilizando máscaras específicas e apresentando vantagens e limitações (DHARAMPAL, 2015). No estudo realizado por Ganesan e Sajiv (2017), que comparou os principais métodos de detecção de bordas, o detector de Canny apresentou resultados significativamente superiores, especialmente em imagens com presença de ruído, sendo destacado por sua robustez.

Já na segunda abordagem, a segunda derivada da imagem é utilizada como critério para detecção de bordas. Nesse caso, as bordas são identificadas nos pontos de cruzamento por zero da função derivada, onde ocorrem mudanças significativas na intensidade (DHARAMPAL, 2015). No entanto, como a segunda derivada é extremamente sensível a ruídos, é comum a utilização do método conhecido como Laplaciano do Gaussiano (Log), no qual a imagem é previamente suavizada com um filtro gaussiano antes da aplicação do operador Laplaciano. Essa etapa de suavização reduz o impacto de variações indesejadas, preservando as bordas mais relevantes da imagem.

Um ponto relevante a ser considerado é a dificuldade em realizar a detecção de bordas em imagens ruidosas, uma vez que tanto as bordas quanto o ruído compartilham componentes de alta frequência. Isso torna desafiador distinguir entre transições reais de intensidade e variações indesejadas causadas pelo ruído. Por esse motivo, é comum a aplicação prévia de filtros de suavização e técnicas de

limiarização, com o objetivo de reduzir interferências e realçar as bordas significativas antes da utilização dos métodos de detecção propriamente ditos.

2.2.3 Representações e Espaços de Cor

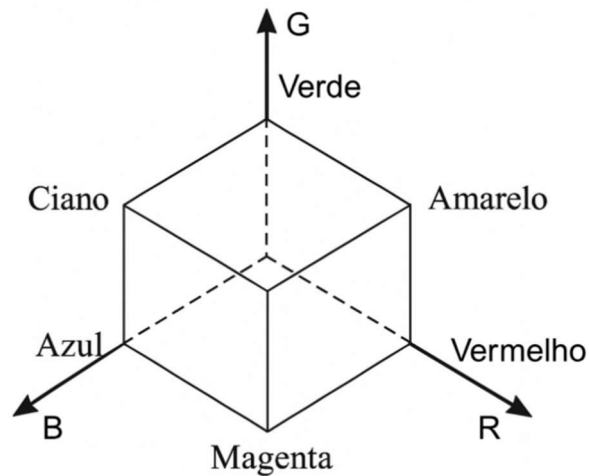
Uma imagem digital é composta por pixels, cada um associado a um valor numérico que representa seu nível de intensidade ou cor. Assim, imagens podem ser interpretadas como representações digitais de objetos do mundo real, que podem ser manipuladas computacionalmente (HEMA e KANNAN, 2019).

Dentro desse contexto, um espaço de cor é uma representação geométrica que organiza as cores em um sistema tridimensional, permitindo que cada cor seja definida por três componentes numéricos que representam suas propriedades visuais (BUSIN, VANDENBROUCKE e MACAIRE, 2008). O modelo RGB é o exemplo mais comum, amplamente utilizado em dispositivos de captura e exibição de imagens. Já o modelo HSV, por separar informações de cor e intensidade, é frequentemente adotado em aplicações de segmentação por cor.

2.2.3.1 Sistema RGB

Segundo Gonzalez e Woods (2018), a percepção de cor pelo olho humano baseia-se em aproximadamente 6 a 7 milhões de cones, divididos em três grupos sensíveis aos comprimentos de onda do vermelho, verde e azul. Essa base fisiológica justifica o uso do modelo RGB (*Red, Green, Blue*), o mais comum em sistemas de aquisição e exibição de imagens. O RGB segue a teoria da tricomacia, assumindo que qualquer cor visível pode ser obtida por uma combinação ponderada dessas três cores primárias (BUSIN, VANDENBROUCKE e MACAIRE, 2008). Nesse modelo, cada pixel da imagem possui três valores, um para cada componente espectral (vermelho, verde e azul), que juntos formam a imagem final. Esses canais podem ser visualizados como três camadas sobrepostas. Por exemplo, quando os valores de todos os canais são zero, o pixel resulta em preto, e quando são máximos (255), resulta em branco (HEMA e KANNAN, 2019). Essa estrutura pode ser representada em um modelo tridimensional, como ilustrado na Figura 13, útil para visualizar a distribuição e a combinação das cores no espaço RGB.

Figura 13 – Representação do sistema RGB.



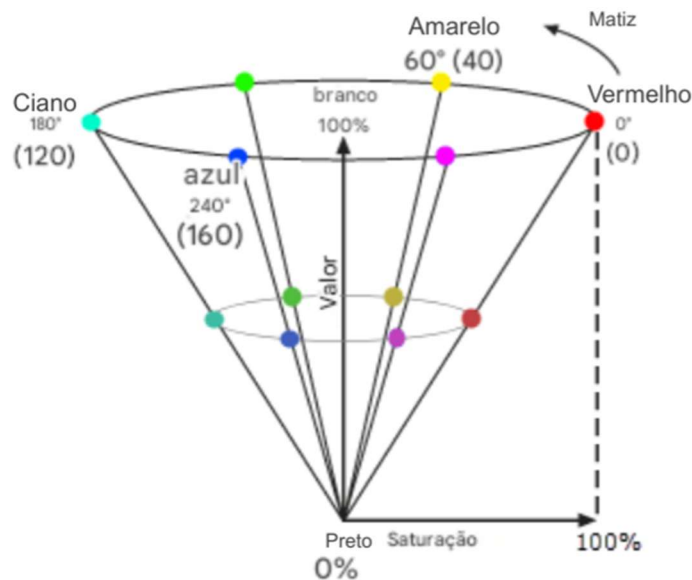
Fonte: Adaptado de (FORSYTH e PONCE, 2002).

2.2.3.2 Sistema HSV

O sistema HSV (*Hue*, *Saturation*, *Value*) foi desenvolvido na década de 1970 por pesquisadores da área de computação gráfica com o objetivo de representar as cores de maneira mais próxima à percepção humana (HASSAN, MING e WAH, 2020). Nesse modelo, *Hue* (H), também conhecido como matiz, representa a cor dominante percebida, *Saturation* (S) indica a pureza ou intensidade da cor, e *Value* (V) está associado ao brilho ou intensidade luminosa do pixel (HEMA e KANNAN, 2019).

O espaço HSV pode ser visualizado como um cilindro geométrico, no qual a angulação define a tonalidade (*Hue*): 0° corresponde ao vermelho, 120° ao verde, 240° ao azul, e 360° fecha novamente no vermelho. A distância do eixo central até a borda do cilindro representa a saturação, sendo que valores mais próximos da borda indicam cores mais puras e vibrantes. Já o eixo vertical central do cilindro corresponde ao valor (*Value*), variando de 0 (preto) na base, onde não há emissão de luz, até 1 (branco) no topo, onde o brilho atinge seu valor máximo (HEMA e KANNAN, 2019). A Figura 14 ilustra a representação desse espaço de cor.

Figura 14 – Representação do sistema HSV.



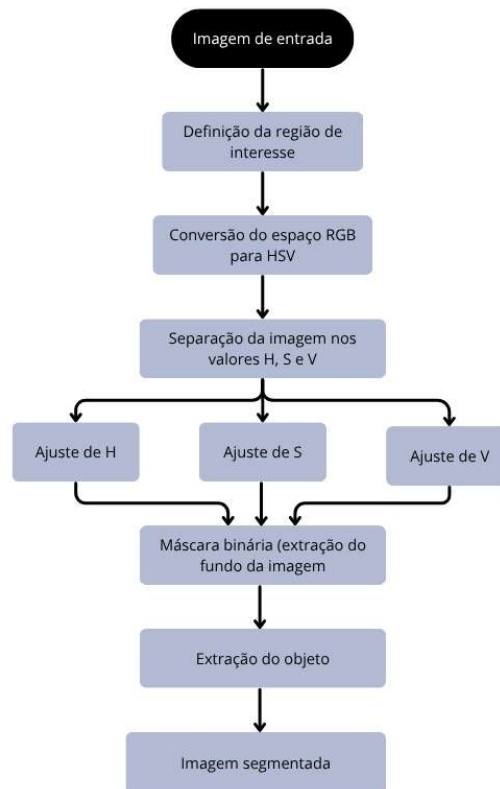
Fonte: Adaptado de (ESRI).

2.2.3.3 Segmentação de Imagens com Base em Cor no Espaço HSV

Assim como abordado em Hema e Kannan (2019), o espaço de cor HSV destaca-se na segmentação de imagens baseada em cor, pois permite definir limites para os parâmetros *Hue* (H), *Saturation* (S) e *Value* (V) com base na tonalidade do objeto a ser identificado. Quando os valores desses parâmetros estão bem ajustados, o objeto de interesse é evidenciado e o fundo pode ser removido por meio de técnicas de limiarização (*thresholding*).

Para aplicar esse tipo de segmentação, é necessário converter a imagem do espaço de cor tradicional RGB para o espaço HSV. A segmentação é então realizada com base nos limites definidos para H, S e V, que delimitam a faixa de cor do objeto desejado. A Figura 15 ilustra esse processo de conversão e ajuste dos parâmetros para extração do objeto.

Figura 15 – Diagrama do processo de segmentação por cor no espaço HSV.



Fonte: Adaptado de (HEMA e KANNAN, 2019).

2.2.4 Comparação entre Métodos Baseados em IA e Tradicionais

A utilização de técnicas baseadas em *deep learning* (aprendizado profundo) na área de visão computacional tem se difundido bastante nos últimos anos, sendo aplicadas nas áreas de classificação de imagens, segmentação e detecção de objetos. Esse avanço é explicado em razão da grande evolução dos recursos computacional e da tecnologia que tornam essas soluções mais acessíveis. No entanto, os métodos tradicionais de CV ainda desempenham um papel relevante.

2.2.4.1 Deep Learning (DL)

Como explicam O' Mahony, Campbell, *et al.* (2019), *deep learning* é uma subárea de *machine learning* (aprendizado de máquina) baseada em redes neurais artificiais. Diferentemente do paradigma tradicional de programação, em que as soluções são construídas por meio de regras explícitas, essa abordagem substitui a análise manual do problema por um processo de treinamento, no qual o sistema recebe um grande

volume de dados de entrada acompanhados de suas respectivas saídas desejadas. Dessa forma, o modelo aprende os padrões e utiliza o aprendizado para aplicação em novos dados.

Em especial, o desenvolvimento das CNNs (*Convolutional Neural Networks*, ou redes neurais convolucionais) na área de visão computacional tem ganhado grande destaque nos últimos anos, sendo responsável por um avanço significativo na capacidade de reconhecimento e interpretação de objetos (O' MAHONY, CAMPBELL, *et al.*, 2019). As CNNs funcionam por meio do uso de *kernels* (matrizes de valores, também chamados de pesos) que são treinados para detectar características específicas nas imagens. Esses *kernels* percorrem a imagem aplicando uma operação de convolução, identificando regiões em que determinada característica está presente.

Dentre as principais vantagens da utilização dessa técnica para CV, destacam-se:

- Altíssima precisão em tarefas como classificação, segmentação e detecção;
- Aprendizado ponta a ponta: a rede aprende a extrair automaticamente as melhores características da imagem;
- Adaptabilidade: as redes podem ser ré-treinadas com novos dados para uma nova aplicação.

Por outro lado, as principais desvantagens são:

- Necessidade de grandes volumes de dados para treinamento;
- Custo computacional elevado;
- DL é menos interpretável (“caixa-preta”);
- Problemas em generalizar para novos dados quando há *overfitting*.

2.2.4.2 Método Tradicional de Detecção

A utilização de algoritmos construídos manualmente, sem o emprego de técnicas de aprendizado de máquina, apresenta importantes vantagens em determinadas aplicações. Em tarefas mais simples, essa abordagem demanda menos tempo de

desenvolvimento, menor esforço computacional e pode alcançar bons níveis de eficiência com menos linhas de código. Como destacam O' Mahony, Campbell, *et al.* (2019), por exemplo, ao classificar duas classes de produtos em uma esteira de montagem (uma pintada de vermelho e outra de azul), uma rede neural profunda poderia ser utilizada, desde que houvesse dados suficientes para o treinamento. No entanto, o mesmo resultado pode ser obtido de forma mais simples e rápida por meio de limiarização de cor.

Além disso, no método tradicional, o engenheiro de CV possui controle total sobre o processo e pode ajustar manualmente os parâmetros do algoritmo, o que é menos viável em redes neurais profundas, devido à sua complexidade e grande número de variáveis. Outro ponto relevante é que algoritmos simples, como os baseados em limiarização de cor e contagem de pixels, são geralmente independentes da classe, ou seja, apresentam desempenho semelhante independentemente do conteúdo da imagem (O' MAHONY, CAMPBELL, *et al.*, 2019).

Apesar de suas vantagens em aplicações simples, os métodos tradicionais de CV apresentam limitações importantes. Eles dependem fortemente da extração manual de características, exigindo conhecimento especializado e ajustes manuais extensivos para cada novo cenário. Além disso, são menos eficazes em ambientes complexos ou com grande variabilidade de iluminação, forma ou textura, onde a robustez e a capacidade de generalização são limitadas em comparação aos métodos baseados em *deep learning* (O' MAHONY, CAMPBELL, *et al.*, 2019).

2.2.5 OpenCV

O uso de bibliotecas no desenvolvimento de aplicações computacionais é uma prática consolidada, especialmente por oferecerem funções e algoritmos prontos para tarefas recorrentes. No campo da visão computacional, destaca-se a biblioteca OpenCV.

O OpenCV é uma biblioteca desenvolvida pela Intel, de código aberto que atende a diversos algoritmos de visão computacional e *machine learning*, podendo ser usado nas linguagens Python, C++ e Java (WAGHMARE e GHATULE, 2022). Além disso, roda em todos os sistemas operacionais (Windows, Linux, Android e Mac) (NETO,

2020). Possui ainda centenas de algoritmos e funções de visão computacional, e é estruturada em módulos (OPENCV, 2025). Os principais módulos disponíveis na arquitetura do OpenCV são listados abaixo:

- **Funcionalidade principal (core):** um módulo compacto que define estruturas de dados básicas, incluindo a matriz multidimensional densa Mat e funções básicas usadas por todos os outros módulos;
- **Processamento de imagem (imgproc):** um módulo de processamento de imagem que inclui filtragem de imagem linear e não-linear, transformações geométricas de imagem (redimensionamento, distorção afim e de perspectiva, remapeamento genérico baseado em tabela), conversão de espaço de cor, histogramas e assim por diante;
- **Leitura e gravação de arquivos de imagem (imgcodecs):** inclui funções para leitura e gravação de arquivos de imagem em vários formatos;
- **E/S de vídeo (videoio):** uma interface fácil de usar para captura de vídeo e codecs de vídeo;
- **GUI de alto nível (highgui):** uma interface fácil de usar para recursos de UI simples;
- **Análise de vídeo (video):** um módulo de análise de vídeo que inclui estimativa de movimento, subtração de fundo e algoritmos de rastreamento de objetos;
- **Calibração de câmera e reconstrução 3D (calib3d):** algoritmos básicos de geometria de múltiplas visualizações, calibração de câmera única e estéreo, estimativa de pose de objeto, algoritmos de correspondência estéreo e elementos de reconstrução 3D;
- **2D Features Framework (features2d):** detectores de características salientes, descritores e correspondentes de descritores;
- **Deteção de objetos (objdetect):** detecção de objetos e instâncias de classes predefinidas (por exemplo, rostos, olhos, canecas, pessoas, carros e assim por diante);
- **Módulo de Rede Neural Profunda (dnn):** Módulo de Rede Neural Profunda;

- **Aprendizado de máquina (ml):** o módulo Aprendizado de máquina inclui um conjunto de classes e funções para classificação estatística, regressão e agrupamento de dados;
- **Fotografia computacional (foto):** técnicas avançadas no processamento de fotos, como redução de ruído e pintura;
- **Costura de imagens (stitching):** funções para costura de imagens e criação de panoramas;
- ... e alguns outros módulos auxiliares, como FLANN e wrappers de teste do Google, vinculações Python e outros.

O módulo de processamento de imagem é mostrado na Figura 16, que contém um grande conjunto de funções, permitindo tarefas como as listadas nos tópicos.

Figura 16 – Modulo de processamento de imagem no OpenCV.

Tópicos

Filtragem de imagem
Transformações de imagens geométricas
Transformações de imagens diversas
Funções de desenho
Conversões de espaço de cores
Mapas de cores no OpenCV
Subdivisão Planar
Histogramas
Análise Estrutural e Descritores de Forma
Análise de movimento e rastreamento de objetos
Detecção de recursos
Detecção de objetos
Segmentação de Imagem
Camada de aceleração de hardware

Fonte: Adaptado de (OPENCV, 2025).

Dessa forma, a ampla variedade de funções disponíveis na biblioteca OpenCV, aliada à sua extensa documentação e facilidade de uso, contribui para sua aplicação em diversas áreas, como monitoramento de fluxo de tráfego, medição de velocidade em diferentes faixas, inovações em robótica, sistemas de supervisão, segurança e conexões de vídeo (WAGHMARE e GHATULE, 2022).

3 DESENVOLVIMENTO DA PLANTA DIDÁTICA *BALL ON PLATE*

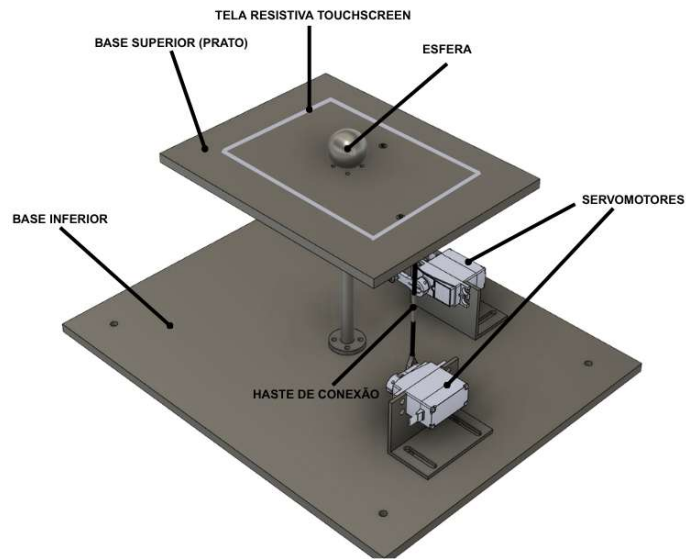
Este capítulo tem como objetivo apresentar de forma detalhada o desenvolvimento da planta didática *Ball on Plate*, abordando desde o princípio de funcionamento e construção do sistema até a modelagem matemática e o controle tradicional.

3.1 Funcionamento do Sistema

O sistema *Ball on Plate* consiste em uma esfera posicionada sobre uma superfície plana (prato ou mesa), cuja inclinação determina o movimento da esfera. Em uma de suas configurações mais comuns, a inclinação é controlada por dois servomotores, que atuam como atuadores do sistema, ajustando o prato em dois eixos ortogonais X e Y. A conexão entre os servomotores e o prato é feita por meio de hastes metálicas articuladas. Segundo Knuplez, Chowdhury e Svecko (2003), trata-se de um sistema eletromecânico bidimensional desenvolvido com o propósito de estudar a dinâmica de sistemas e realizar experimentos laboratoriais, utilizando diferentes abordagens de controle baseadas nas teorias clássica e moderna.

Além disso, o sistema é amplamente utilizado como plataforma didática em cursos de engenharia, proporcionando aos alunos uma aproximação prática dos conceitos teóricos abordados ao longo da formação. No trabalho de Tudić, Šimić e Perić (2022), destaca-se a relevância educacional do sistema, evidenciando que sua implementação permite aos estudantes aplicar ferramentas de modelagem computacional, desenvolver o projeto de controle e realizar a integração entre software e hardware em tempo real, enquanto tentam estabilizar a posição da esfera sobre o prato. A Figura 17 ilustra uma representação da estrutura do sistema *Ball on Plate*.

Figura 17 – Estrutura de um sistema *Ball on Plate*.



Fonte: O autor, 2025.

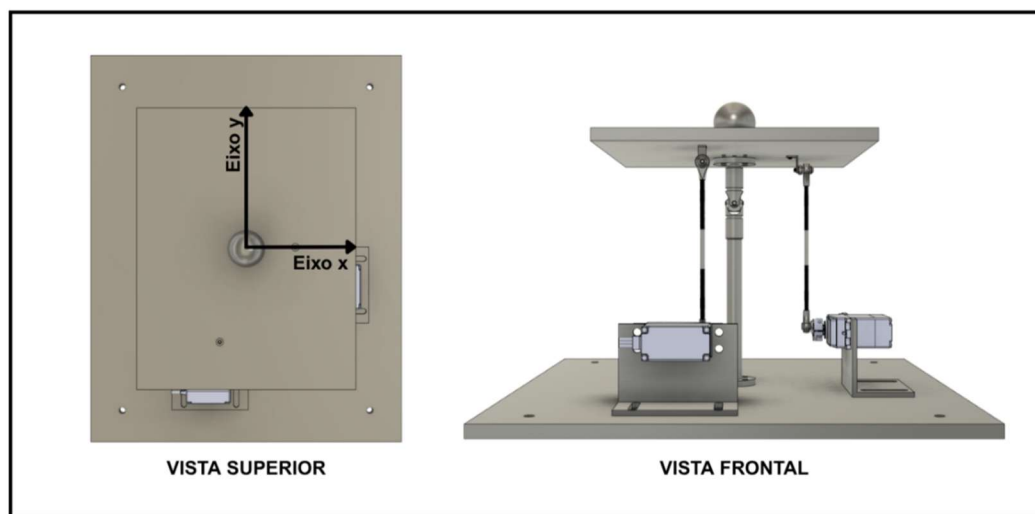
3.1.1 Arquitetura Geral

Neste trabalho, o objetivo é desenvolver um sistema de controle que, a partir de qualquer posição inicial, mantenha a esfera equilibrada sobre a superfície em uma posição fixa, sem a realização de trajetórias específicas (como percursos retangulares e circulares). O funcionamento do sistema segue uma estrutura clássica de controle em malha fechada: uma posição de referência (o centro do prato) é definida como entrada do sistema. Após a liberação da esfera sobre o prato, um sensor de posição é responsável por identificar sua localização atual, expressa pelas coordenadas (x_b, y_b) . A posição medida é comparada com a posição de referência, gerando um sinal de erro, que é processado pelo controlador. Este, por sua vez, calcula a ação de controle necessária para minimizar o erro e reposicionar a esfera conforme desejado. Esse ciclo se repete continuamente, garantindo a atuação em tempo real e o equilíbrio dinâmico da esfera.

Segundo Ham e Taufiq (2015), o sistema *Ball on Plate* pode ser interpretado como a combinação de dois sistemas *Ball and Beam* (bola e haste) dispostos em paralelo, cada um atuando em um eixo distinto, o que lhe confere dois graus de liberdade independentes. Trata-se de um sistema naturalmente instável em malha

aberta e classificado como de múltiplas entradas e saídas (MIMO), com duas entradas (os ângulos de inclinação nos eixos X e Y) e duas saídas (as respectivas posições da esfera nos mesmos eixos). No entanto, o sistema pode ser tratado como dois sistemas de entrada e saídas únicas (SISO) devido a capacidade de separação da dinâmica para cada eixo. A Figura 18 ilustra essa implemetação, mostrando a divisão do sistema completo em dois eixos ortogonais, tratados de forma separada.

Figura 18 – Representação dos eixos ortogonais do sistema *Ball on Plate* em vistas superior e frontal.

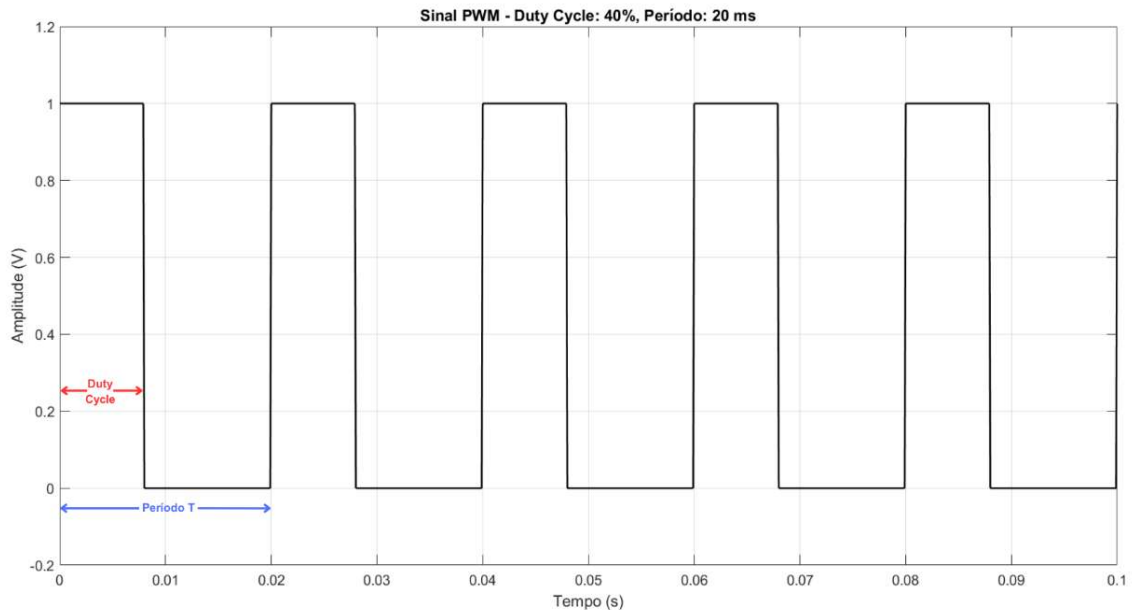


Fonte: O autor, 2025.

3.1.2 Atuadores

Como já mencionado anteriormente, dois servomotores são utilizados como atuadores neste sistema *Ball on Plate*. Esses dispositivos têm a função de converter energia elétrica em energia mecânica, ajustando sua posição angular a partir de sinais de controle do tipo PWM (*Pulse Width Modulation*) enviados por um controlador. O PWM é um sinal periódico com frequência fixa (geralmente 50 Hz para o controle de servomotores), cuja informação de controle é transmitida pela duração do pulso em nível alto (largura de pulso). Essa largura, conhecida como *Duty cycle*, pode ser expressa como porcentagem do período total. A Figura 19 ilustra um sinal PWM, com amplitude unitária, período $T = 20 \text{ mS}$ e *Duty cycle* de 40%.

Figura 19 – Sinal PWM.



Fonte: O autor, 2025.

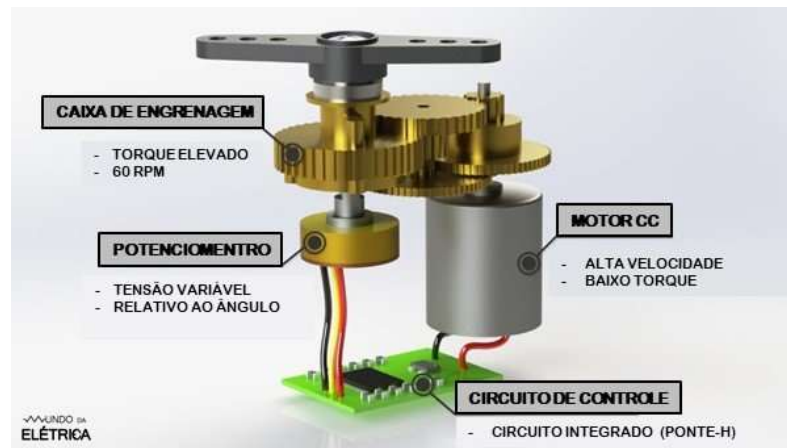
Os servomotores são amplamente empregados em diversas áreas, como na robótica, onde a precisão no controle de inclinação é essencial, e podem operar com rotações limitadas (tipicamente 180°) ou contínuas (360°). Os modelos de 180° são os mais empregados em aplicações de controle de posição, por oferecerem maior precisão (THE ENGINEERING MINDSET, 2022).

A Figura 20 ilustra os principais componentes internos de um servomotor, cuja estrutura típica é composta por:

- Conexões elétricas, geralmente com três fios: um para alimentação, outro para aterramento (referência), e o terceiro para o sinal PWM;
- Um motor de corrente contínua (DC), responsável por gerar o movimento de rotação;
- Um conjunto de engrenagens e rolamentos, utilizado para aumentar o torque na saída e reduzir a velocidade de rotação;
- Uma placa de circuito impresso (driver interno), responsável por interpretar o sinal de entrada, controlar o sentido de rotação e realizar a realimentação de posição;

- E um potenciômetro, acoplado ao eixo de saída, que fornece feedback de posição para o controle em malha fechada.

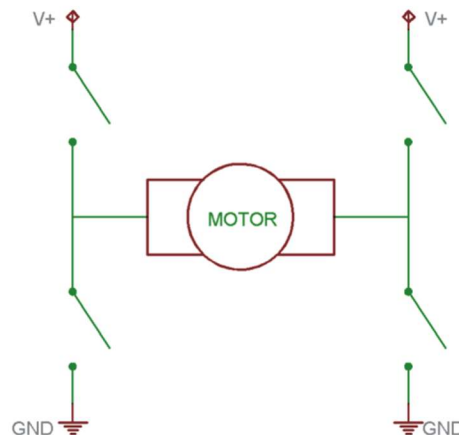
Figura 20 – Componentes internos de um servomotor.



Fonte: (MUNDO DA ELÉTRICA, 2023).

Dessa forma, o motor DC interno gira em alta velocidade, e as engrenagens atuam reduzindo essa velocidade e aumentando o torque na saída do eixo. No eixo de saída, podem ser conectados diversos acessórios conforme a aplicação desejada. O controle da direção de rotação é realizado pela placa de circuito impresso, por meio de uma ponte H que permite a inversão de polaridade, possibilitando o giro em ambos os sentidos. Uma ponte H é composta por 4 chaves, mecânicas ou eletrônicas, posicionadas aos extremos do motor, formando a letra H. Apenas duas chaves opostas diagonais são acionadas, por vez, para girar o motor em determinada direção, fazendo com que a corrente flua do terminal positivo para o negativo (PATSKO, 2006). A Figura 21 ilustra essa representação.

Figura 21 – Representação de uma ponte H.



Fonte: (MUNDO DA ELÉTRICA, 2023).

A grande vantagem dos servomotores está na sua capacidade de operar em malha fechada: a posição do eixo é constantemente monitorada pelo potenciômetro, e seu sinal de realimentação é comparado com o sinal de referência. A partir dessa comparação, o sistema ajusta automaticamente a rotação até que a posição desejada seja atingida com precisão.

3.1.3 Sensores

Um dos principais desafios na implementação do sistema *Ball on Plate* está na definição do método de realimentação, ou seja, na escolha do sensor responsável por detectar a posição da esfera. Essa etapa é fundamental, pois a obtenção precisa e em tempo real da posição da esfera é essencial para o correto funcionamento do sistema de controle. Nesse contexto, duas abordagens são mais utilizadas: a primeira, utilizando uma tela sensível ao toque resistiva para detectar a posição da esfera, e a segunda, utilizando uma câmera para o mesmo objetivo. Pode-se definir a primeira abordagem como o método tradicional de realimentação, e o segundo método como por visão computacional.

De acordo com Ham e Taufiq (2015), a abordagem tradicional utilizando uma tela sensível ao toque tende a ser mais precisa e adequada, especialmente devido à forma como o sensor é integrado diretamente à superfície da mesa, permitindo a leitura da posição da esfera de forma direta e com menor complexidade estrutural. Em

contraste, a utilização de uma câmera exige que esta seja posicionada suspensa acima da plataforma, o que aumenta o volume do sistema.

Já Araujo, Canova, *et al.* (2016) destacam que, embora a utilização de câmeras represente uma alternativa de menor custo, as telas sensíveis ao toque apresentam a vantagem de possibilitar taxas de amostragem mais altas. Isso ocorre porque a leitura da posição da esfera por meio da tela resistiva pode ser feita a uma frequência superior à taxa de atualização das *webcams* comuns, cuja frequência de aquisição é fixa e, em geral, limitada.

Como um dos objetivos centrais deste trabalho é a comparação entre os dois métodos de detecção da posição da esfera, essa análise se torna essencial para a validação das teorias discutidas ao longo do estudo. Para que tal comparação seja possível, é fundamental o conhecimento prévio sobre os materiais e dispositivos empregados na implementação de cada abordagem.

3.1.3.1 Tela Sensível ao Toque Resistiva

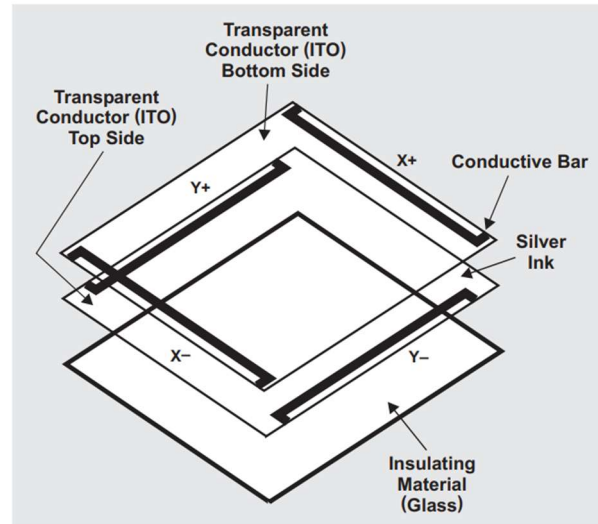
Existem quatro principais tipos de tecnologia de telas sensíveis ao toque, como citado por Bhalla e Bhalla (2010): “Os painéis de toque em si são baseados em quatro tecnologias básicas de tela: resistiva, capacitiva, onda acústica de superfície (*Surface Acoustic Wave*, SAW) e infravermelho (*Infrared*, IR). Cada um desses designs apresenta vantagens e desvantagens distintas”. A tecnologia resistiva destaca-se por sua simplicidade, baixo custo e ampla aplicação em sistemas de controle industrial, pontos de venda e dispositivos médicos.

As telas resistivas podem ser do tipo 4 fios ou 5 fios. As de 5 fios oferecem maior precisão e durabilidade (BHALLA e BHALLA, 2010), porém, para este trabalho, a tela de 4 fios mostrou-se adequada, tanto pelo custo-benefício quanto pela simplicidade de funcionamento.

A tela de 4 fios é composta por um painel de vidro ou acrílico revestido com duas camadas condutivas, normalmente feitas de óxido de índio-estanho (*Indium Tin Oxide*, ITO) (DOWNS, 2005). Essas camadas condutivas são separadas por um espaçador isolante fino, que evita o contato entre elas em repouso. A Figura 22 mostra a estrutura básica da tela resistiva, destacando as camadas condutivas superior e

inferior e os eletrodos para os eixos X ($X+$, $X-$) e Y ($Y+$, $Y-$), nos quais as resistências são distribuídas linearmente ao longo do material.

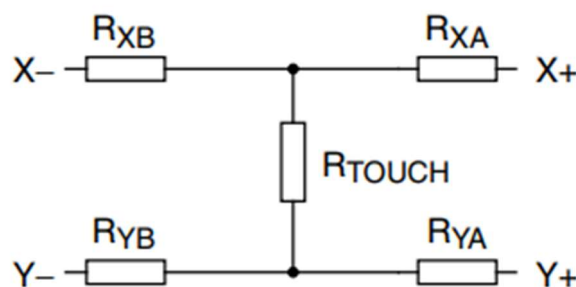
Figura 22 – Composição de uma tela sensível ao toque resistiva de 4 fios.



Fonte: (DOWNS, 2005).

O circuito equivalente da tela é representado na Figura 23, o qual inclui as resistências distribuídas nos dois eixos (R_{XA} , R_{XB} , R_{YA} , R_{YB}), bem como a resistência de contato resultante do toque entre as camadas condutivas (R_{TOUCH}).

Figura 23 – Circuito equivalente da tela sensível ao toque resistiva de 4 fios.



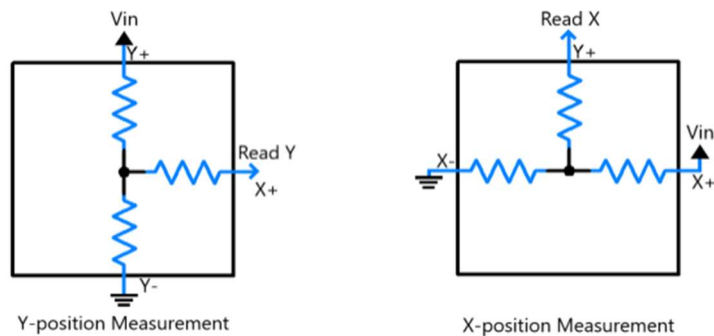
Fonte: (NXP SEMICONDUCTORS, 2008).

A camada superior, mais flexível, é responsável pela condução na direção Y, enquanto a camada inferior, mais rígida, atua na direção X. Cada camada é utilizada

para determinar a posição da esfera em um dos eixos cartesianos. Quando a tela é pressionada, a camada superior se deforma até encostar na inferior, estabelecendo um ponto de contato. Nesse ponto, ocorre a leitura da tensão gerada por um circuito divisor de tensão.

Para medir a posição no eixo Y, aplica-se uma tensão entre os terminais Y+ e Y–, enquanto se realiza a leitura da tensão em um dos terminais X, com as camadas em contato. Analogamente, para obter a coordenada no eixo X, a tensão é aplicada entre os terminais X+ e X–, sendo a leitura realizada em um dos terminais Y. Como se trata de um divisor resistivo linear, a tensão medida é proporcional à posição do toque, permitindo a determinação da localização da esfera sobre a superfície da tela. A Figura 24 ilustra o processo de leitura das coordenadas da esfera por meio da tela resistiva de 4 fios. A Tabela 1 apresenta o diagrama de ligação correspondente, indicando como cada terminal (X+, X–, Y+, Y–) deve ser configurado para permitir a leitura da posição nos eixos X e Y.

Figura 24 – Leitura da posição x e y utilizando divisor de tensão.



Fonte: (ELECTRONICWINGS, 2023).

Tabela 1 – Diagrama de ligação da tela resistiva touchscreen de 4 fios.

Função	X+	Y+	X-	Y-
Leitura posição X	Fonte de tensão	Medição de tensão	Vss	Aberto
Leitura posição Y	Medição de tensão	Fonte de tensão	Aberto	Vss

Fonte: O autor, 2025.

3.1.3.2 Webcam

Para a implementação do método de realimentação baseado em visão computacional, foi utilizada uma webcam como dispositivo de aquisição de imagem. Sua função é capturar, em tempo real, a cena do sistema *Ball on Plate*, permitindo a identificação da posição da esfera sobre o prato.

A escolha por uma webcam se deve à sua ampla disponibilidade, baixo custo e compatibilidade com bibliotecas de processamento de imagem, como o OpenCV. Além disso, sua resolução e taxa de quadros são suficientes para aplicações de controle que não exigem latência extremamente baixa. As imagens obtidas são processadas por meio de algoritmos específicos, que realizam etapas como filtragem, segmentação e detecção de contornos, a fim de estimar com precisão a posição da esfera no plano.

3.1.4 Unidade de Controle

A unidade de controle no sistema *Ball on Plate* corresponde ao dispositivo responsável por executar as principais tarefas do sistema de controle em tempo real, integrando hardware e software. Suas funções incluem: adquirir os dados dos sensores, comparar as informações com os valores de referência, aplicar o algoritmo de controle definido e gerar os sinais de saída que serão enviados aos atuadores do sistema.

No método tradicional de realimentação, a unidade de controle realiza a leitura das variáveis de posição da esfera por meio de uma tela sensível ao toque resistiva, utilizando entradas analógicas. Os sinais adquiridos são então convertidos em coordenadas cartesianas (x , y). Já na abordagem baseada em CV, as coordenadas da esfera são processadas externamente e enviadas diretamente ao controlador em seu formato final.

A partir dos dados de posição adquiridos, o sistema de controle computa um sinal de controle que busca minimizar o erro entre a posição atual e a posição de referência (geralmente o centro do prato). Esse sinal é enviado aos servomotores, que ajustam suas posições angulares para mover a esfera conforme o desejado.

Portanto, a unidade de controle deve dispor de capacidade de processamento (*Central Processing Unit*, CPU), memória, interfaces de entrada e saída digitais e analógicas, bem como saídas PWM para o controle dos servomotores, garantindo a integração eficiente entre sensores, atuadores e algoritmo de controle.

A utilização de dispositivos embarcados para controle em tempo real é amplamente adotada em aplicações de automação, robótica e IoT. Entre as plataformas mais comuns destacam-se Arduino, ESP32, STM32 e Raspberry Pi, cada uma com características específicas quanto a desempenho, custo e recursos integrados.

O Arduino, por exemplo, é amplamente reconhecido por sua simplicidade e ampla comunidade de suporte, sendo ideal para aplicações introdutórias. No entanto, sua capacidade de processamento é limitada, o que pode restringir seu uso em sistemas mais complexos. Por outro lado, o Raspberry Pi oferece excelente capacidade de processamento e suporte a sistemas operacionais completos, porém com maior consumo de energia e custo mais elevado, além de exigir um ambiente mais robusto de desenvolvimento.

Nesse cenário, o ESP32 destaca-se como uma solução intermediária que combina alto desempenho computacional, conectividade Wi-Fi e Bluetooth integradas, baixo custo e facilidade de programação compatível com a IDE do Arduino. Segundo Digi-Key (2023), o ESP32 oferece uma solução mais versátil e potente em comparação ao Arduino, com maior capacidade de processamento e recursos embarcados.

De forma semelhante, Eelectronics (2024) aponta que tanto o ESP32 quanto o STM32 são adequados para aplicações embarcadas exigentes, mas o ESP32 se diferencia por já integrar conectividade sem fio e apresentar menor custo.

Além disso, testes comparativos realizados por Rios, Andrade, *et al.* (2020) evidenciam o bom desempenho do ESP32 frente a outras plataformas, como o ESP8266 e o MKR1000, em critérios como velocidade de processamento, intensidade do sinal Wi-Fi, consumo de energia e presença de sensores internos.

Essas características tornam o ESP32 uma das plataformas mais indicadas para aplicações que exigem controle embarcado em tempo real, destacando-se pelos seguintes tópicos:

- **Custo-benefício:** apresenta preço acessível em relação às funcionalidades oferecidas;
- **Conectividade integrada:** presença de Wi-Fi e Bluetooth embarcados, úteis para comunicação com interfaces externas e visualização de variáveis em tempo real;
- **Desempenho robusto:** capacidade de processamento adequada para executar algoritmos de controle em tempo real;
- **Dimensões reduzidas:** formato compacto, ideal para integração em sistemas com restrições de espaço físico.

3.2 Construção da Estrutura Mecânica

Para a construção da estrutura do sistema, foi utilizado o modelo desenvolvido em Ody (2020), aproveitando-se suas dimensões e o posicionamento dos furos das peças.

Como o objetivo principal do trabalho é realizar o controle do sistema utilizando métodos diferentes de realimentação, a utilização de um modelo previamente consolidado traz a principal vantagem de diminuir o tempo de projeto da planta, uma vez que o posicionamento dos servomotores e do eixo principal já está definido.

Parte das peças de referência foi adquirida pela internet, devido à sua complexidade geométrica e disponibilidade comercial. É o caso da junta universal e das hastes de conexão dos servomotores.

As demais peças foram modeladas no software de modelagem 3D Fusion 360, da Autodesk. Isso inclui as estruturas de suporte dos servomotores, a haste central, o eixo central, e as bases inferior e superior. Em seguida, as peças foram fabricadas por meio de impressão 3D, utilizando o software Repetier-Host, com exceção das bases, que foram confeccionadas através de corte a laser.

A seguir, são apresentadas as peças que compõem a estrutura mecânica do sistema.

3.2.1 Estrutura de Suporte para o Servomotor

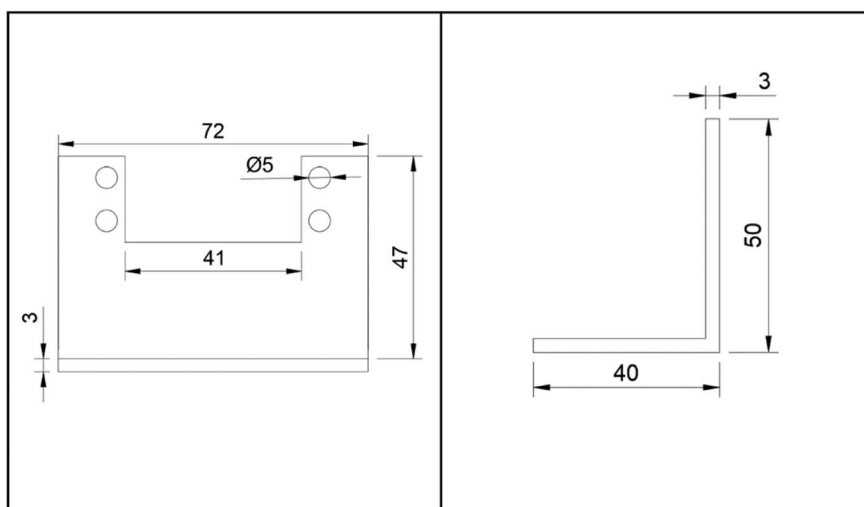
A estrutura de suporte para o servomotor é mostrada na Figura 25. Foram impressas duas unidades dessa peça, uma para cada servomotor. A Figura 26 apresenta as vistas frontal e lateral esquerda da estrutura de suporte, com as respectivas dimensões em milímetros. Observa-se a presença de quatro furos de 5 mm de diâmetro, utilizados para a fixação do servomotor à estrutura.

Figura 25 – Estrutura de suporte para o servomotor.



Fonte: O autor, 2025.

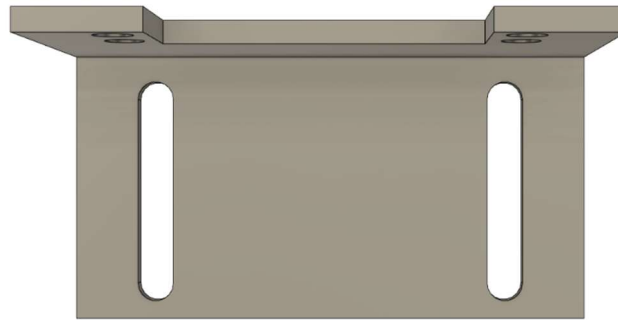
Figura 26 – Estrutura de suporte para o servomotor, vistas frontal e lateral esquerda com dimensões.



Fonte: O autor, 2025.

Conforme ilustrado na Figura 27, a base da estrutura possui uma região vazada com 5 mm de diâmetro, projetada para permitir a fixação do suporte à base do sistema, além de possibilitar o ajuste do posicionamento do servomotor.

Figura 27 – Vista superior da estrutura de suporte para o servomotor.



Fonte: O autor, 2025.

3.2.2 Haste Central

A haste central do sistema, responsável por sustentar o centro do prato superior em relação à base da estrutura, é mostrada na Figura 28. Essa peça foi produzida por meio de impressão 3D, a partir do modelo tridimensional desenvolvido no software Fusion 360.

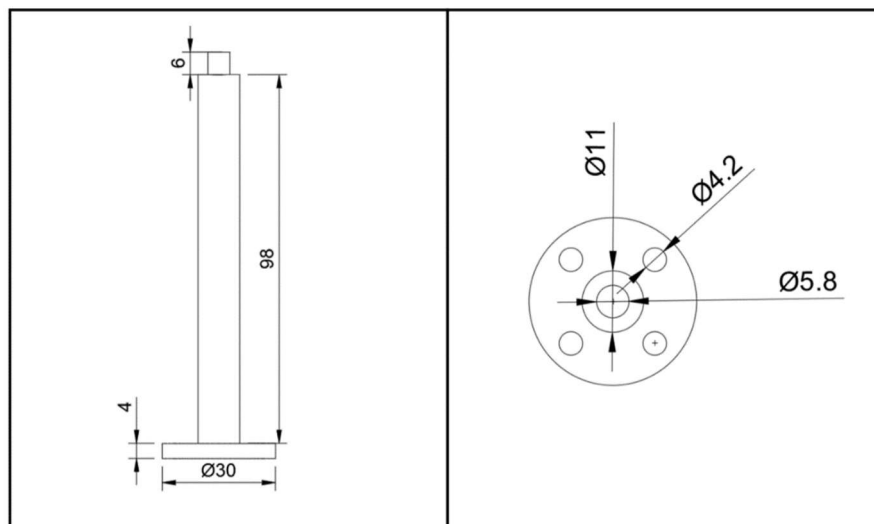
Figura 28 – Haste central do sistema.



Fonte: O autor, 2025.

A base da haste possui 4 furos de 4,20 mm de diâmetro, destinados à sua fixação à base do sistema. O eixo principal da haste possui 11 mm de diâmetro e 98 mm de comprimento. Em sua extremidade superior, há uma prolongação cilíndrica com 6 mm de comprimento e 5,80 mm de diâmetro, projetada para permitir a conexão com uma junta universal, componente responsável por fornecer dois graus de liberdade ao sistema. A Figura 29 mostra as vistas lateral e superior da haste, com suas respectivas dimensões em milímetros.

Figura 29 – Desenho técnico da Haste central do sistema com vistas lateral e superior cotadas.



Fonte: O autor, 2025.

3.2.3 Junta Universal

Uma junta universal, acoplada à haste central na parte inferior, e à um eixo central na parte superior, foi utilizada para conferir ao sistema o deslocamento em dois eixos. A junta universal possui 23 mm de comprimento, 11 mm de diâmetro externo e 6 mm de diâmetro nos furos superior e inferior. A Figura 30 mostra a peça utilizada.

Figura 30 – Junta universal.



Fonte: O autor, 2025.

3.2.4 Eixo Central Superior

O eixo central superior, responsável por fixar o prato superior à haste central por meio da junta universal, é apresentado na Figura 31. Essa peça foi produzida por meio de impressão 3D, com base no modelo desenvolvido no software Fusion 360.

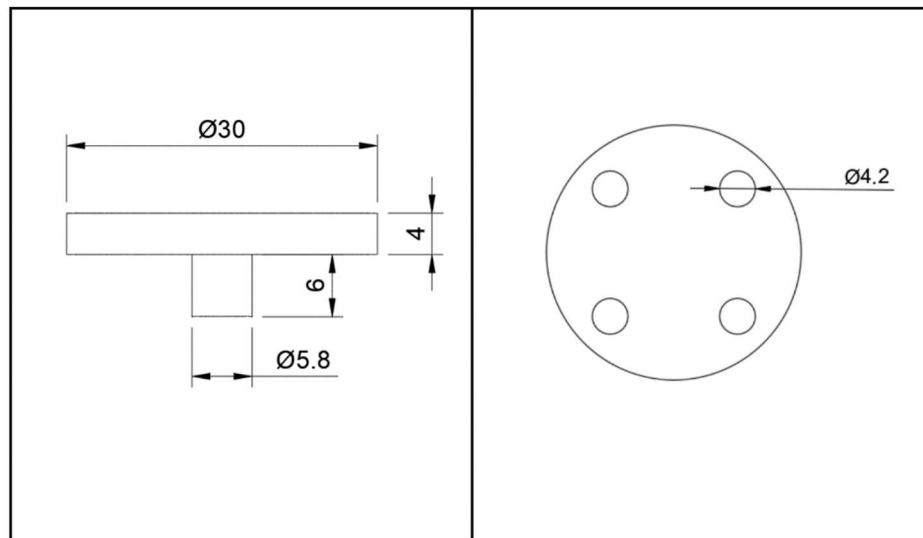
Figura 31 – Eixo central superior.



Fonte: O autor, 2025.

A Figura 32 exibe a peça nas vistas frontal e superior, com suas respectivas dimensões em milímetros. A peça contém 4 furos de 4,2 mm no topo para fixação do prato superior, e uma estrutura cilíndrica de 5,8 mm de diâmetro para encaixe com a junta universal.

Figura 32 – Eixo central superior.



Fonte: O autor, 2025.

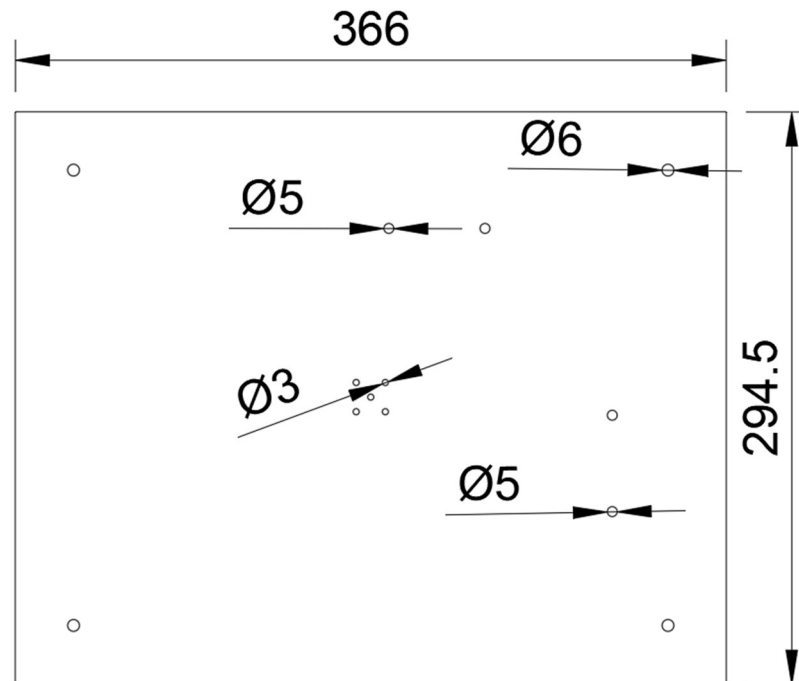
3.2.5 Bases Inferior e Superior

A base inferior foi modelada utilizando o Fusion 360, desenhando-se apenas a face com os furos, uma vez que a peça seria fabricada por corte a laser em uma chapa de acrílico de 4 mm de espessura. O arquivo correspondente foi exportado em formato PDF vetorizado, adequado para o processo de corte.

A Figura 33 apresenta o desenho técnico da face da base inferior, com suas respectivas dimensões em milímetros. A peça possui:

- Quatro furos de 6 mm, mais externos, destinados à fixação dos apoios (pés) da estrutura;
- Quatro furos de 5 mm, que definem o posicionamento das estruturas de suporte dos servomotores;
- Furos de 3 mm, centrais, responsáveis por fixar a base inferior à base da haste central.

Figura 33 – Desenho técnico da face da base inferior, com medidas em milímetros.



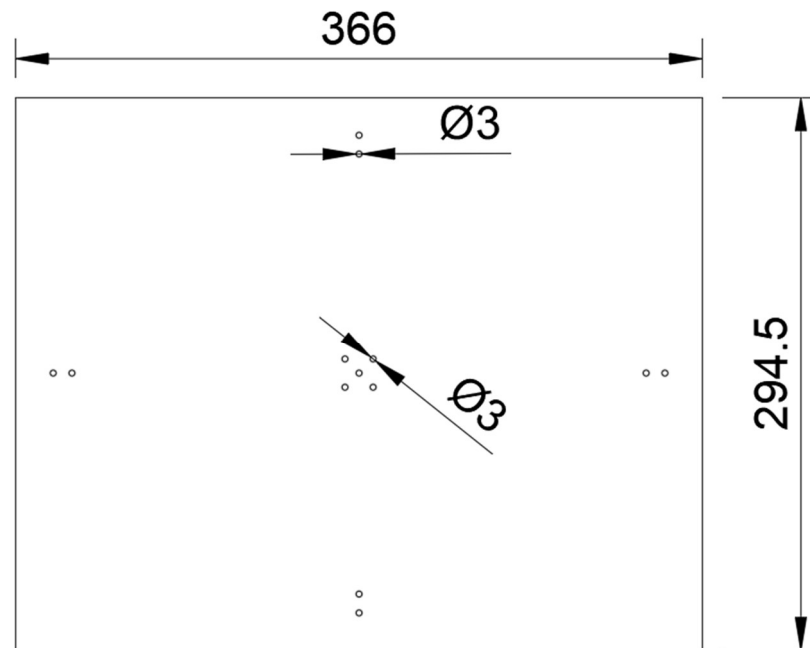
Fonte: O autor, 2025.

O mesmo processo foi adotado para a fabricação da base superior, que também foi confeccionada a partir de uma chapa de acrílico com 4 mm de espessura, por meio de corte a laser.

A Figura 34 apresenta o desenho técnico da face da base superior, com suas respectivas dimensões em milímetros. A peça apresenta furos centrais de 3 mm de diâmetro, responsáveis pela fixação da base superior (prato) ao eixo central do sistema. Além dos furos centrais, a peça apresenta oito furos de 3 mm posicionados no centro de cada lado da base, dois por lado, destinados à conexão entre os servomotores e o prato por meio das hastes de ligação. A existência de dois furos possibilita o ajuste de posição que apresenta o melhor desempenho do sistema.

Neste trabalho, são utilizados apenas dois servomotores, de modo que apenas dois conjuntos de furos (quatro furos ao todo) são efetivamente utilizados para a movimentação do prato.

Figura 34 – Desenho técnico da face da base superior, com medidas em milímetros.



Fonte: O autor, 2025.

3.2.6 Hastes de Conexão

Para conexão dos servomotores com o prato superior, foram utilizadas duas unidades da haste de conexão mostrada na Figura 35. A haste de conexão possui um comprimento longitudinal total ajustável de 86 a 96 mm e diâmetro dos furos nas extremidades de 3 mm.

Figura 35 – Haste de conexão.



Fonte: O autor, 2025.

3.3 Seleção dos Equipamentos e Componentes

Neste tópico, são apresentadas as escolhas dos principais dispositivos utilizados no sistema, incluindo o controlador, os servomotores, a tela resistiva, a câmera utilizada na visão computacional e, por fim, a esfera. Cada componente é descrito com base em suas características técnicas e na justificativa de uso.

3.3.1 ESP32 DevKit V1

Neste trabalho, foi utilizada a placa de desenvolvimento ESP32 DevKit V1, que incorpora o módulo ESP-WROOM-32, fabricado pela Espressif Systems. Este módulo integra o microcontrolador ESP32, caracterizado por possuir arquitetura dual-core Tensilica Xtensa LX6, com frequência de até 240 MHz, 520 KB de SRAM, suporte à memória Flash externa, além de conectividade Wi-Fi 802.11 b/g/n e Bluetooth v4.2 (BLE) integradas (ESPRESSIF SYSTEMS, 2020). O módulo também conta com até 36 pinos GPIO, 16 entradas ADC (12 bits), interface USB, 2 DACs (8 bits) e diversos periféricos de comunicação, como SPI, I2C, UART, I2S e CAN.

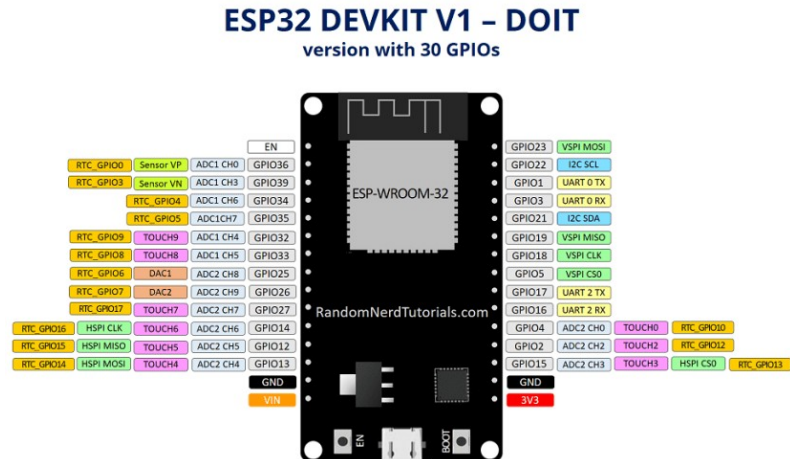
As Figura 36 e Figura 37 apresentam, respectivamente, a placa de desenvolvimento ESP32 DevKit V1 e o esquema de sua pinagem, mostrando os principais pinos de entrada, saída e comunicação utilizados no sistema.

Figura 36 – Placa de desenvolvimento ESP32 DevKit V1.



Fonte: O autor, 2025.

Figura 37 – Diagrama de pinagem da placa ESP32 DevKit V1.



Fonte: (RANDOM NERD TUTORIALS, 2023).

Optou-se pela utilização deste modelo devido à sua disponibilidade, facilidade de uso (por ser compatível com o ambiente de programação Arduino), além de apresentar bom desempenho e excelente custo-benefício.

3.3.2 Servomotor DS3225MG

A Figura 38 apresenta o servomotor DS3225MG, acompanhado de sua estrutura de suporte. Este modelo foi selecionado devido à sua robustez mecânica e alta capacidade de torque, conforme demonstrado em suas especificações técnicas apresentadas na Tabela 2. A Figura 39, ilustra como diferentes larguras de pulso controlam a posição angular do servomotor DS3225MG.

Figura 38 – Servomotor DS3225MG acoplado em sua estrutura de suporte.



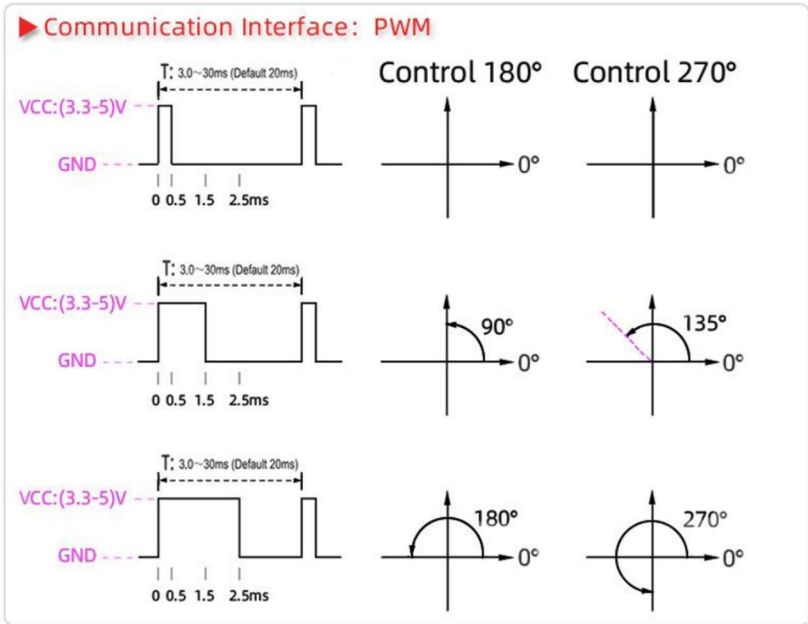
Fonte: O autor, 2025.

Tabela 2 – Especificações técnicas servomotor DS3225MG.

Parâmetro	Especificação
Torque máximo	25 Kg·cm
Tensão de operação	4.8 – 6.8 V
Frequência típica do sinal PWM	50 Hz
Ângulo de operação	180° ou 270° (com sinal PWM entre 500-2500 μ s)
Peso	60g
Dimensões	40 x 20 x 40.5 mm

Fonte: (DONGGUAN CITY DSSERVO TECHNOLOGY CO., LTD., 2023).

Figura 39 – Controle de posição do servomotor DS3225MG por largura de pulso PWM.



Fonte: (DONGGUAN CITY DSSERVO TECHNOLOGY CO., LTD., 2023).

3.3.1 Tela Resistiva de 4 Fios

A tela resistiva, utilizada na implementação do controle tradicional, é mostrada na Figura 40. Optou-se pela tecnologia resistiva de 4 fios pela sua facilidade de implementação. A tela possui 22,5 cm de comprimento e 17,3 cm de altura.

Figura 40 – Tela resistiva de 4 fios.



Fonte: O autor, 2025.

3.3.2 Webcam NB307

Para o método de realimentação baseado em CV, uma webcam NB307 Full HD 1080p é utilizada para adquirir a posição da esfera no prato. A escolha do modelo se deve ao custo-benefício e compatibilidade com o OpenCV. A webcam possui resolução máxima de vídeo de 1920 x 1080 pixels (Full HD), o que garante uma boa definição da imagem capturada, favorecendo a segmentação da esfera. Além disso, a taxa de atualização atinge até 30 quadros por segundo, o que permite acompanhar os movimentos da esfera com fluidez e baixa latência. A Figura 41 mostra o modelo.

Figura 41 – Webcam NB307 Full HD 1080p.



Fonte: O autor, 2025.

3.3.3 Esfera

Optou-se por uma esfera de aço com 20 mm de diâmetro. A seleção desse material justifica-se por apresentar massa (45,0 g) adequada para a detecção na tela resistiva, quando utilizada no método tradicional, além de possuir alta rigidez mecânica e uma superfície lisa e uniforme, o que contribui para a redução do atrito durante o movimento sobre a plataforma. A Figura 42 mostra a esfera utilizada.

Figura 42 – Esfera.



Fonte: O autor, 2025.

Para implementação do método de realimentação baseado em visão computacional, a esfera foi pintada na cor verde através de uma tinta spray metálica. Depois, foi aplicado um verniz fosco incolor para fixação e remoção de brilho. Observou-se que, após a pintura, a esfera ficou menos lisa, porém sua dinâmica ainda permaneceu funcional. A cor verde foi escolhida por ter um alto contraste com o fundo da mesa, além de possuir pouca interferência. Todo o processo de pintura foi pensado para não gerar interferências no processo de detecção. A Figura 43 mostra o resultado.

Figura 43 – Esfera pintada.



Fonte: O autor, 2025.

3.3.4 Circuito de Alimentação

O circuito de alimentação do sistema foi projetado para fornecer uma tensão de 5 V aos terminais dos servomotores, atendendo à demanda de corrente exigida durante sua operação. Segundo o fabricante, os servomotores utilizados consomem até 2,1 A quando alimentados a 5 V e submetidos à condição de travamento (*stall current*), ou seja, no limite de torque.

Contudo, no sistema desenvolvido, os servomotores operam com folga, sem atingir o torque máximo especificado de 25 kg·cm, o que implica um consumo de corrente inferior ao valor de pico. Além dos servomotores, o circuito também fornece alimentação para a placa ESP32, unidade central de controle.

A alimentação do sistema é dividida em três blocos principais, descritos a seguir:

1. **Fonte chaveada CA/CC MS-60-12:** Responsável por converter a tensão da rede elétrica (220 V em corrente alternada, CA) para 12 V em corrente contínua (CC), com capacidade de corrente de até 5 A. Essa tensão é utilizada como entrada para o conversor abaixador de tensão, sendo suficiente para alimentar todos os componentes do sistema;
2. **Conversor buck XL4005:** É um conversor abaixador de tensão, responsável por converter os 12 V da fonte para 5 V CC, tensão adequada para alimentação da ESP32 e dos servomotores. Esse conversor opera com tensões de entrada de 4-35 V CC, possui saída ajustável entre 1-32 V, suporta corrente de até 5 A e potência máxima de 75 W;
3. **Distribuição da tensão:** A tensão de 5 V gerada é distribuída para alimentação dos servomotores e da ESP32.

As Figura 44 e Figura 45 mostram a fonte chaveada e o conversor buck utilizados.

Figura 44 – Fonte chaveada CA/CC MS-60-12.



Fonte: O autor, 2025.

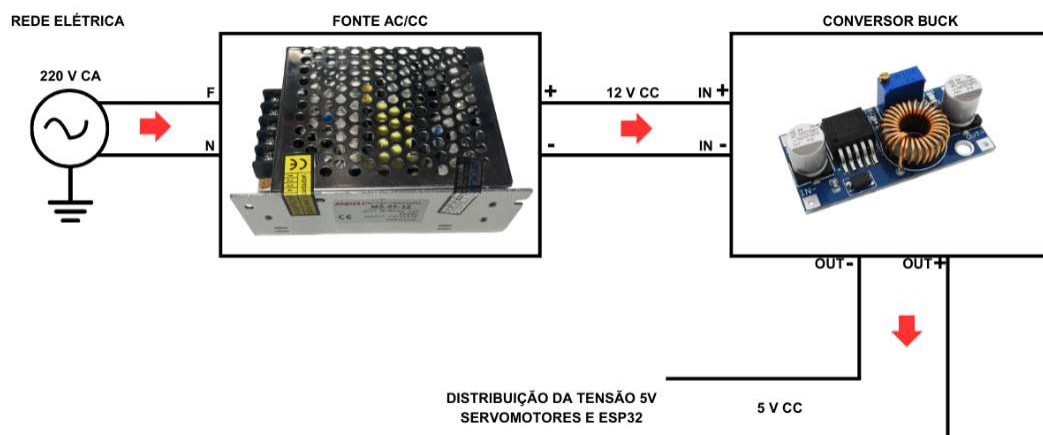
Figura 45 – Conversor buck XL4005.



Fonte: O autor, 2025.

A Figura 46 ilustra o diagrama de conexão do circuito de alimentação do sistema, destacando o fluxo de energia desde a rede elétrica até os componentes alimentados (servomotores e a placa ESP32).

Figura 46 – Diagrama esquemático do circuito de alimentação do sistema.

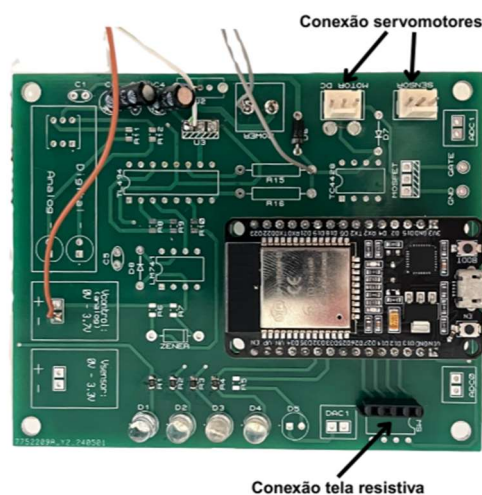


Fonte: O autor, 2025.

3.3.5 Placa para Conexão da ESP32 e Servomotores

A Figura 47 apresenta a placa de circuito impresso utilizada para o projeto, destacando os pontos de conexão dos servomotores e da tela resistiva. Essa placa é alimentada pela fonte chaveada 12 V. O conversor buck, que fornece 5 V aos terminais de conexão dos servomotores e alimenta a ESP32, é conectado nos terminais da placa. Os pinos do microcontrolador estão conectados às respectivas saídas PWM destinadas ao controle dos servomotores, bem como às entradas analógicas (ADC), utilizadas para a leitura das coordenadas da esfera por meio da tela resistiva, no método tradicional de realimentação.

Figura 47 – Placa de circuito impresso utilizada para controle e aquisição de dados do sistema.



Fonte: O autor, 2025.

3.4 Modelagem do Sistema

Duas abordagens podem ser utilizadas para modelar o sistema *Ball on Plate*: a formulação de Newton e a formulação Lagrangiana. Cada um dos métodos tem a sua vantagem e deve ser aplicado de acordo com a análise do sistema.

De acordo com Taylor (2013), a formulação Lagrangiana apresenta duas vantagens principais em relação à formulação Newtoniana. A primeira é que as equações de Lagrange mantêm a mesma forma independentemente do sistema de coordenadas utilizado, ao contrário das equações de Newton. A segunda vantagem é que, em sistemas com vínculos (ou seja, aqueles em que o movimento de partes do

sistema está sujeito a restrições), o método de Lagrange permite eliminar as forças de vínculo da modelagem, como forças normais e de atrito, o que simplifica significativamente o problema. Além disso, as equações de Lagrange partem da análise da energia cinética e potencial, diferentemente de Newton, onde um tratamento vetorial deve ser feito para determinar as direções e resultantes das forças.

Dessa forma, utilizou-se a formulação Lagrangiana para a modelagem do sistema *Ball on Plate*, conforme abordado por Bolívar-Vincenty e Beauchamp-Báez (2014). Esse método baseia-se na definição de uma função escalar chamada função Lagrangiana, que é dada pela diferença entre a energia cinética K e a energia potencial U do sistema, como mostrado na equação abaixo:

$$L = K - U. \quad (3.1)$$

A partir dessa função, aplica-se a equação de Euler-Lagrange, expressa como:

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} = 0, \quad (3.2)$$

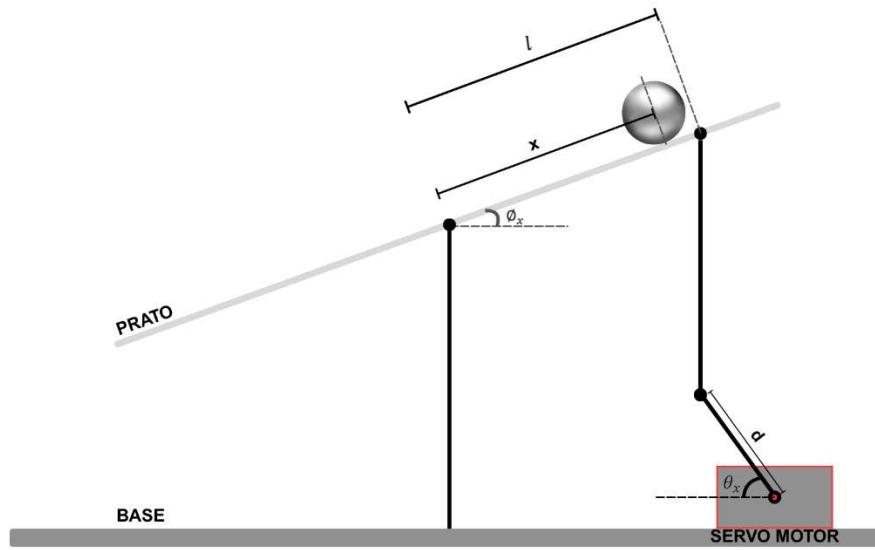
onde o termo q_i é definido como uma coordenada generalizada do sistema. As coordenadas generalizadas são escolhidas de forma a descrever completamente o sistema, podendo incluir posições lineares, ângulos, módulos ou qualquer outra variável adequada à geometria e restrições do sistema. A equação (3.2) é aplicável para qualquer coordenada generalizada, contanto que o sistema seja inercial (TAYLOR, 2013). Para um sistema submetido a forças externas, a equação (3.2) assume a forma:

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} = Q_i, \quad (3.3)$$

onde o termo Q_i é denominado força externa generalizada do sistema.

A posição da esfera no prato é definida como (x_b, y_b) . Para simplificação, será analisado apenas o eixo X do sistema, como mostrado na Figura 48.

Figura 48 – Modelo do *Ball on Plate* simplificado (eixo X).



Fonte: O autor, 2025.

O primeiro passo, na modelagem, é definir as coordenadas generalizadas do sistema, onde serão utilizadas a posição (x) da esfera e o ângulo de inclinação (ϕ_x) do prato:

$$q = (x, \phi_x). \quad (3.4)$$

A energia cinética do sistema é definida como:

$$K = K_b + K_p, \quad (3.5)$$

onde K_b e K_p são a energia cinética da esfera e do prato.

O prato apenas possui energia cinética rotacional definida por:

$$K_p = \frac{1}{2} \cdot J_p \cdot \dot{\phi}_x^2, \quad (3.6)$$

onde J_p e ϕ_x representam, respectivamente, o momento de inércia e o ângulo de inclinação do prato.

Por outro lado, a energia cinética da esfera tem duas componentes: uma referente ao seu movimento linear e outra ao movimento de rotação. Portanto:

$$K_b = \frac{1}{2} \cdot m_b \cdot v_b^2 + \frac{1}{2} \cdot J_b \cdot \omega_b^2, \quad (3.7)$$

em que m_b é a massa da esfera, v_b sua velocidade linear, J_b o momento de inércia, e ω_b a velocidade angular da esfera. Utilizando a relação entre o movimento angular e linear da esfera, tem-se:

$$\omega_b = \frac{v_b}{r} = \frac{\dot{x}_b}{r}; \quad (3.8)$$

$$\omega_b^2 = \frac{\dot{x}_b^2}{r^2}, \quad (3.9)$$

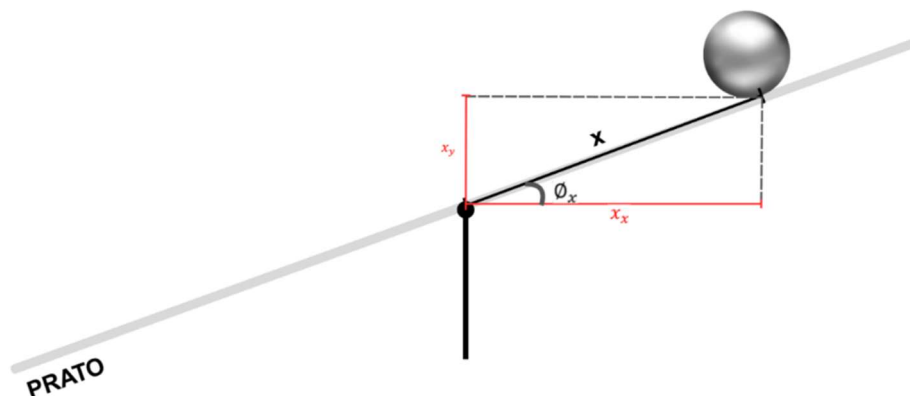
em que r é o raio da esfera.

Para expressar a velocidade da esfera v_b em termos das coordenadas generalizadas, utilizam-se x_x e x_y , que representam, respectivamente, as projeções horizontal e vertical da posição da esfera, conforme ilustrado na Figura 49. Dessa forma, obtêm-se as seguintes expressões:

$$v_b = \dot{x}_b = \dot{x}_x + \dot{x}_y; \quad (3.10)$$

$$v_b^2 = \dot{x}_x^2 + \dot{x}_y^2. \quad (3.11)$$

Figura 49 – Projeções horizontal e vertical da posição da esfera.



Fonte: O autor, 2025.

Com base na Figura 49, observa-se que a projeção horizontal da posição da esfera é dada por:

$$x_x = x \cdot \cos(\phi_x). \quad (3.12)$$

Derivando em relação ao tempo, obtém-se a velocidade da projeção horizontal:

$$\dot{x}_x = \dot{x} \cdot \cos(\phi_x) - x \cdot \sin(\phi_x) \cdot \dot{\phi}_x. \quad (3.13)$$

Elevando ao quadrado para uso na equação (3.11):

$$\dot{x}_x^2 = \dot{x}^2 \cdot \cos^2(\phi_x) - 2 \cdot x \cdot \dot{x} \cdot \cos(\phi_x) \cdot \sin(\phi_x) \cdot \dot{\phi}_x + x^2 \cdot \sin^2(\phi_x) \cdot \dot{\phi}_x^2. \quad (3.14)$$

Executando a mesma análise para a projeção vertical x_y , tem-se:

$$x_y = x \cdot \sin(\phi_x); \quad (3.15)$$

$$\dot{x}_y = \dot{x} \cdot \sin(\phi_x) + x \cdot \cos(\phi_x) \cdot \dot{\phi}_x; \quad (3.16)$$

$$\dot{x}_y^2 = \dot{x}^2 \cdot \sin^2(\phi_x) + 2 \cdot x \cdot \dot{x} \cdot \cos(\phi_x) \cdot \sin(\phi_x) \cdot \dot{\phi}_x + x^2 \cdot \cos^2(\phi_x) \cdot \dot{\phi}_x^2. \quad (3.17)$$

Substituindo as equações (3.14) e (3.17) em (3.11), obtém-se:

$$v_b^2 = \dot{x}^2 + x^2 \cdot \dot{\phi}_x^2. \quad (3.18)$$

Substituindo as equações (3.18) e (3.9) em (3.7):

$$K_b = \frac{1}{2} \cdot m_b \cdot (\dot{x}^2 + x^2 \cdot \dot{\phi}_x^2) + \frac{1}{2} \cdot J_b \cdot \frac{\dot{x}^2}{r^2}. \quad (3.19)$$

Portanto, energia cinética do sistema é definida como:

$$K = \frac{1}{2} \cdot \dot{x}^2 \cdot \left(m_b + \frac{J_b}{r^2} \right) + \frac{1}{2} \cdot m_b \cdot x^2 \cdot \dot{\phi}_x^2 + \frac{1}{2} \cdot J_p \cdot \dot{\phi}_x^2. \quad (3.20)$$

A energia potencial do sistema é dada em função da posição da bolinha em relação ao centro do prato, definindo assim a energia potencial gravitacional do sistema como:

$$U = m_b \cdot g \cdot x \cdot \text{sen}(\phi_x), \quad (3.21)$$

em que g representa a gravidade. Portanto, a função Lagrangiana definida para um eixo do sistema (eixo X), é definida por:

$$L_x = \frac{1}{2} \cdot \dot{x}^2 \cdot \left(m_b + \frac{J_b}{r^2}\right) + \frac{1}{2} \cdot m_b \cdot x^2 \cdot \dot{\phi}_x^2 + \frac{1}{2} \cdot J_p \cdot \dot{\phi}_x^2 - m_b \cdot g \cdot x \cdot \text{sen}(\phi_x). \quad (3.22)$$

De forma análoga, a função Lagrangiana para o eixo Y é dada por:

$$L_y = \frac{1}{2} \cdot \dot{y}^2 \cdot \left(m_b + \frac{J_b}{r^2}\right) + \frac{1}{2} \cdot m_b \cdot y^2 \cdot \dot{\phi}_y^2 + \frac{1}{2} \cdot J_p \cdot \dot{\phi}_y^2 - m_b \cdot g \cdot y \cdot \text{sen}(\phi_y). \quad (3.23)$$

A função Lagrangiana completa do sistema é:

$$L = \frac{1}{2} \cdot (\dot{x}^2 + \dot{y}^2) \cdot \left(m_b + \frac{J_b}{r^2}\right) + \frac{1}{2} \cdot J_p \cdot (\dot{\phi}_x^2 + \dot{\phi}_y^2) + \frac{1}{2} \cdot m_b \cdot (x^2 \cdot \dot{\phi}_x^2 + y^2 \cdot \dot{\phi}_y^2) - m_b \cdot g \cdot x \cdot \text{sen}(\phi_x) - m_b \cdot g \cdot y \cdot \text{sen}(\phi_y). \quad (3.24)$$

Como já mencionado anteriormente, a análise pode ser feita para apenas um dos eixos e a equação será análoga para o segundo eixo. Por isso, na equação (3.22), tem-se que:

$$\frac{\partial L_x}{\partial \dot{x}} = \dot{x} \cdot \left(m_b + \frac{J_b}{r^2}\right); \quad (3.25)$$

$$\frac{d}{dt} \frac{\partial L_x}{\partial \dot{x}} = \ddot{x} \cdot \left(m_b + \frac{J_b}{r^2}\right); \quad (3.26)$$

$$\frac{\partial L_x}{\partial x} = x \cdot m_b \cdot \dot{\phi}_x^2 - m_b \cdot g \cdot \text{sen}(\phi_x). \quad (3.27)$$

Aplicando a equação de Euler Lagrange, descrita pela equação (3.2), para a primeira coordenada generalizada, tem-se:

$$\ddot{x} \cdot \left(m_b + \frac{J_b}{r^2} \right) = x \cdot m_b \cdot \dot{\phi}_x^2 - m_b \cdot g \cdot \sin(\phi_x). \quad (3.28)$$

Aplicando as derivadas parciais para a segunda coordenada generalizada na equação (3.22), obtém-se:

$$\frac{\partial L_x}{\partial \dot{\phi}_x} = (m_b \cdot x^2 + J_p) \cdot \dot{\phi}_x; \quad (3.29)$$

$$\frac{d}{dt} \frac{\partial L_x}{\partial \dot{\phi}_x} = (m_b \cdot x^2 + J_p) \cdot \ddot{\phi}_x + 2 \cdot m_b \cdot x \cdot \dot{x} \cdot \dot{\phi}_x; \quad (3.30)$$

$$\frac{\partial L_x}{\partial \phi_x} = -m_b \cdot g \cdot x \cdot \cos(\phi_x). \quad (3.31)$$

Desenvolvendo a equação de Euler Lagrange para a segunda coordenada generalizada, com a presença de um torque externo (aplicado pelo servomotor), descrita pela equação (3.3), tem-se:

$$(m_b \cdot x^2 + J_p) \cdot \ddot{\phi}_x + 2 \cdot m_b \cdot x \cdot \dot{x} \cdot \dot{\phi}_x + m_b \cdot g \cdot x \cdot \cos(\phi_x) = \tau. \quad (3.32)$$

As equações (3.28) e (3.32) são as equações do movimento do sistema *Ball on Plate* para o primeiro eixo. Analogamente, para o eixo Y, as equações do movimento são descritas por:

$$\ddot{y} \cdot \left(m_b + \frac{J_b}{r^2} \right) = y \cdot m_b \cdot \dot{\phi}_y^2 - m_b \cdot g \cdot \sin(\phi_y); \quad (3.33)$$

$$(m_b \cdot y^2 + J_p) \cdot \ddot{\phi}_y + 2 \cdot m_b \cdot y \cdot \dot{y} \cdot \dot{\phi}_y + m_b \cdot g \cdot y \cdot \cos(\phi_y) = \tau. \quad (3.34)$$

3.4.1 Linearização

As equações do movimento ao longo do eixo X, desenvolvidas anteriormente, são expressas da seguinte forma:

$$\ddot{x} \cdot \left(m_b + \frac{J_b}{r^2} \right) = x \cdot m_b \cdot \dot{\phi}_x^2 - m_b \cdot g \cdot \sin(\phi_x); \quad (3.35)$$

$$(m_b \cdot x^2 + J_p) \cdot \ddot{\phi}_x + 2 \cdot m_b \cdot x \cdot \dot{x} \cdot \dot{\phi}_x + m_b \cdot g \cdot x \cdot \cos(\phi_x) = \tau. \quad (3.36)$$

Observa-se a presença de termos não-lineares na equação, o que impossibilita o uso direto da representação no domínio da frequência, onde técnicas de projeto de controladores, como o Lugar das Raízes (LGR) e o diagrama de Bode, são aplicáveis. Porém, o sistema pode ser linearizado em torno de um ponto de operação para que a análise no domínio da frequência possa ser utilizada. A equação linearizada será uma aproximação do modelo real.

Dessa forma, será seguido o processo de linearização descrito em Ogata (2010), onde a função não-linear será expandida em uma série de Taylor em torno de um ponto de operação, mantendo-se apenas o termo linear (aproximação de primeira ordem). A seguir, apresentam-se as expressões da expansão de Taylor de primeira ordem para funções não-lineares de uma e duas variáveis:

$$f(x) \approx f(x_0) + (df/dx)|_{x=x_0} \cdot (x - x_0); \quad (3.37)$$

$$f(x, y) \approx f(x_0, y_0) + \left(\frac{\partial f}{\partial x} \right) |_{(x_0, y_0)} \cdot (x - x_0) + \left(\frac{\partial f}{\partial y} \right) |_{(x_0, y_0)} \cdot (y - y_0), \quad (3.38)$$

em que x_0 e y_0 são os pontos de operação definidos.

Para a equação (3.35), os termos não-lineares são:

$$f(x, \dot{\phi}_x) = x \cdot \dot{\phi}_x^2; \quad (3.39)$$

e

$$f(\phi_x) = \sin(\phi_x). \quad (3.40)$$

O ponto de operação é definido quando a esfera está em repouso no centro do prato, ou seja:

$$x_0 = 0; \quad (3.41)$$

$$\phi_{x0} = 0; \quad (3.42)$$

$$\dot{\phi}_{x0} = 0. \quad (3.43)$$

Aplicando (3.38) em (3.39), tem-se:

$$f(x, \dot{\phi}_x) \approx f(x_0, \dot{\phi}_{x0}) + \left(\frac{\partial f}{\partial x}\right) |_{(x_0, \dot{\phi}_{x0})} \cdot (x - x_0) + (\partial f / \dot{\phi}_x) |_{(x_0, \dot{\phi}_{x0})} \cdot (\dot{\phi}_x - \dot{\phi}_{x0}); \quad (3.44)$$

$$\left(\frac{\partial f}{\partial x}\right) |_{(x_0, \dot{\phi}_{x0})} = \dot{\phi}_{x0}^2 = 0; \quad (3.45)$$

$$\left(\frac{\partial f}{\partial \dot{\phi}_x}\right) |_{(x_0, \dot{\phi}_{x0})} = 2 \cdot x_0 \cdot \phi_{x0} = 0; \quad (3.46)$$

$$f(x_0, \dot{\phi}_{x0}) = x_0 \cdot \dot{\phi}_{x0}^2 = 0. \quad (3.47)$$

Portanto, a equação (3.39) linearizada é:

$$f(x, \dot{\phi}_x) \approx 0, \quad (3.48)$$

para x e $\dot{\phi}_x$ definidos em torno do ponto de operação.

Aplicando (3.37) em (3.40), obtém-se:

$$f(\phi_x) \approx f(\phi_{x0}) + (df/d\phi_x) |_{\phi_x=\phi_{x0}} \cdot (\phi_x - \phi_{x0}); \quad (3.49)$$

$$\left(\frac{df}{d\phi_x}\right) |_{\phi_x=\phi_{x0}} = \cos(\phi_{x0}) = 1; \quad (3.50)$$

$$(\phi_x - \phi_{x0}) = \phi_x - \phi_{x0} = \phi_x; \quad (3.51)$$

$$\text{sen}(\phi_{x0}) = 0. \quad (3.52)$$

Portanto, a equação (3.40) linearizada é:

$$f(\phi_x) \approx \phi_x, \quad (3.53)$$

para ϕ_x definido em torno do ponto de operação.

Concluindo, substituindo as equações (3.48) e (3.53) na equação (3.35), tem-se:

$$\ddot{x} \cdot \left(m_b + \frac{I_b}{r^2} \right) = -m_b \cdot g \cdot \phi_x. \quad (3.54)$$

A equação (3.54) representa a forma linearizada do sistema e é suficiente para descrever sua dinâmica, estabelecendo uma relação direta entre a posição da esfera e o ângulo de inclinação do prato no eixo X, conforme desenvolvido também em Madhumitha (2021). De forma análoga, para o eixo Y, a equação linearizada que descreve a dinâmica do sistema é definida por:

$$\ddot{y} \cdot \left(m_b + \frac{I_b}{r^2} \right) = -m_b \cdot g \cdot \phi_y. \quad (3.55)$$

3.4.2 Função de Transferência do Sistema em Malha Aberta

Neste trabalho, optou-se por modelar o sistema *Ball on Plate* no domínio da frequência, utilizando funções de transferência. Tal abordagem foi escolhida por proporcionar maior simplicidade e clareza na representação da dinâmica do sistema linearizado. Como a análise foi conduzida separadamente para os eixos X e Y, cada um deles configura um sistema de entrada e saída única (SISO), o que torna o uso de funções de transferência totalmente apropriado. Essa forma de modelagem permite a aplicação direta de técnicas clássicas de análise e projeto de controladores, como o método do lugar das raízes e diagramas de Bode, amplamente difundidos no meio acadêmico e industrial. Além disso, a representação no domínio de Laplace facilita a compreensão do comportamento dinâmico do sistema, incluindo aspectos como estabilidade, resposta transitória e regime permanente, sem envolver a complexidade algébrica inerente ao método de espaço de estados. Outro fator relevante é a excelente integração desse método com ferramentas computacionais como o

MATLAB/Simulink, o que contribui significativamente para a simulação e a implementação eficiente dos controladores projetados.

A função de transferência de malha aberta do sistema pode ser obtida tomando a transformada de Laplace da equação (3.54), de forma a obter-se uma relação direta entre a posição da esfera x e o ângulo de inclinação do prato ϕ_x . Contudo, uma análise mais precisa é definir a posição da esfera em termos da inclinação do servomotor, θ_x .

Pode-se observar, a partir da Figura 48, que:

$$l \cdot \phi_x = d \cdot \theta_x. \quad (3.56)$$

Portanto:

$$\phi_x = \frac{d}{l} \cdot \theta_x. \quad (3.57)$$

Substituindo (3.57) em (3.54), obtém-se:

$$\ddot{x} \cdot \left(m_b + \frac{J_b}{r^2} \right) = - \frac{m_b \cdot g \cdot d}{l} \cdot \theta_x. \quad (3.58)$$

Tomando a transformada de Laplace:

$$\mathcal{L}\{\ddot{x} \cdot \left(m_b + \frac{J_b}{r^2} \right) = - \frac{m_b \cdot g \cdot d}{l} \cdot \theta_x\}; \quad (3.59)$$

$$X(s) \cdot s^2 \cdot \left(m_b + \frac{J_b}{r^2} \right) = - \frac{m_b \cdot g \cdot d}{l} \cdot \theta_x(s). \quad (3.60)$$

Rearranjando a equação acima, tem-se a função de transferência para o eixo X:

$$\frac{X(s)}{\theta_x(s)} = - \frac{m_b \cdot g \cdot d}{l \cdot \left(m_b + \frac{J_b}{r^2} \right) \cdot s^2}. \quad (3.61)$$

O momento de inércia J_b de uma esfera sólida, em relação ao seu centro de massa, é dado por:

$$J_b = \frac{2}{5} \cdot m_b \cdot r^2. \quad (3.62)$$

Substituindo (3.62) em (3.61), simplificando a equação e fazendo a mesma analogia para o eixo Y, obtém-se as funções de transferência do sistema para os dois eixos:

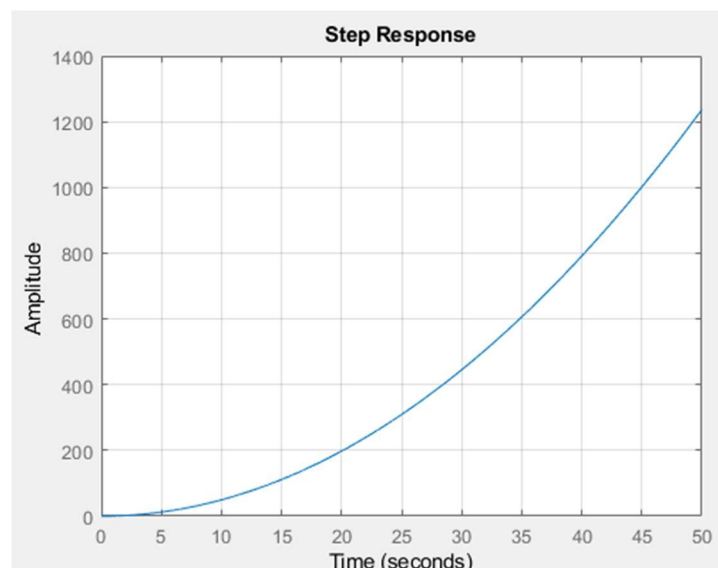
$$\frac{X(s)}{\theta_x(s)} = -\frac{5 \cdot g \cdot d}{7 \cdot l \cdot s^2}, \quad (3.63)$$

$$\frac{Y(s)}{\theta_y(s)} = -\frac{5 \cdot g \cdot d}{7 \cdot l \cdot s^2}. \quad (3.64)$$

3.4.3 Simulação do Sistema em Malha Aberta

Utilizando o software MATLAB, foi realizada a simulação do sistema em malha aberta para o eixo X, considerando os parâmetros $g = 9,81 \text{ m/s}^2$, $d = 24 \text{ mm}$ e $l = 170 \text{ mm}$. Observou-se que, ao aplicar um degrau unitário na entrada, a resposta do sistema cresce continuamente, indicando instabilidade e que resultaria na queda da esfera do prato (Figura 50).

Figura 50 – Simulação em malha aberta (sem controlador) do sistema com entrada em degrau unitário.

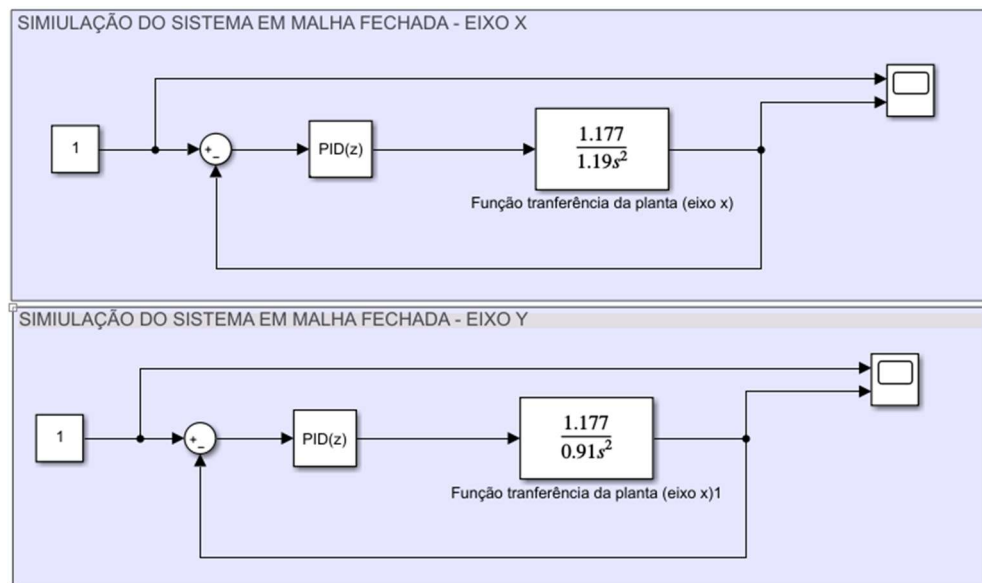


Fonte: O autor, 2025.

3.4.4 Simulação do Sistema em Malha Fechada

Para simulação do sistema em malha fechada, utilizou-se o MATLAB/Simulink, onde foi introduzido um controlador do tipo PID para fechamento de malha. Para calibração dos parâmetros do controlador, foi utilizado o PID Tuner já integrado no *software*. Essa aplicação permite ver em tempo real o tipo e parâmetros da resposta à medida em que os ganhos do controlador PID vão sendo variados. A Figura 51 mostra a simulação para os dois eixos do sistema.

Figura 51 – Simulação do sistema em malha fechada através do MATLAB/Simulink.



Fonte: O autor, 2025.

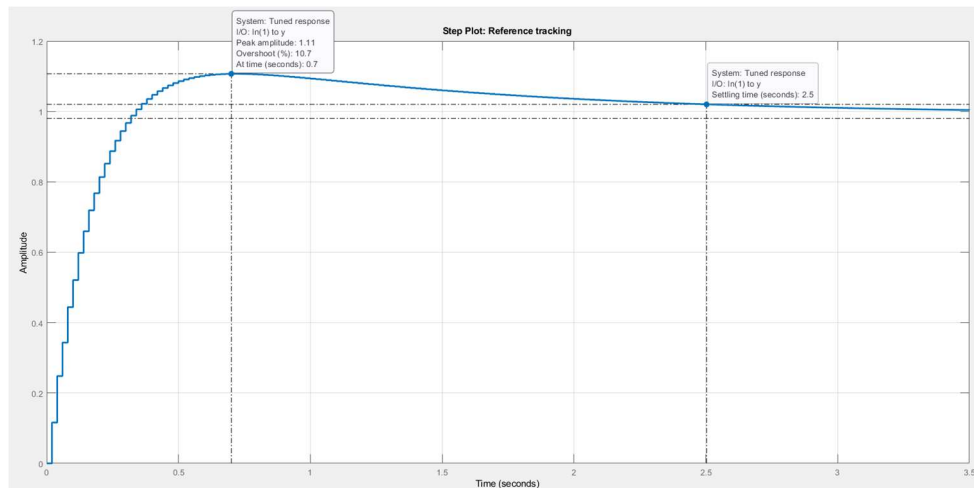
O controlador já foi implementado no domínio de Z, ou seja, em seu equivalente discreto. Utilizou-se o método de Tustin para discretização do integrador (I) e a ação derivativa (D) foi implementada com um filtro de primeira ordem, também discretizado por meio do método trapezoidal, com objetivo de suavizar os efeitos de ruídos de alta frequência. A estrutura resultante do controlador PID é mostrada abaixo:

$$U(z) = K_p + K_i \cdot \frac{T_s}{2} \cdot \frac{z+1}{z-1} + K_d \cdot \frac{N}{1 + N \cdot \frac{T_s}{2} \cdot \frac{z+1}{z-1}}, \quad (3.65)$$

onde $U(z)$ representa a ação de controle, T_s é o período de amostragem, N é o coeficiente do filtro da ação derivativa que aumenta sua agressividade, e K_p , K_i e K_d são, respectivamente, os ganhos proporcional, integrativo e derivativo.

A Figura 52 apresenta a implementação do controlador PID para o eixo X, obtida por meio da ferramenta PID Tuner. Os parâmetros foram ajustados de forma a garantir um tempo de acomodação entre 2 e 3 segundos (utilizando o critério de 2%), com um sobressinal máximo limitado a aproximadamente 10–15%, atendendo aos critérios de desempenho desejados para o sistema.

Figura 52 – Implementação de um controlador utilizando o PID Turner no MATLAB.



Fonte: O autor, 2025.

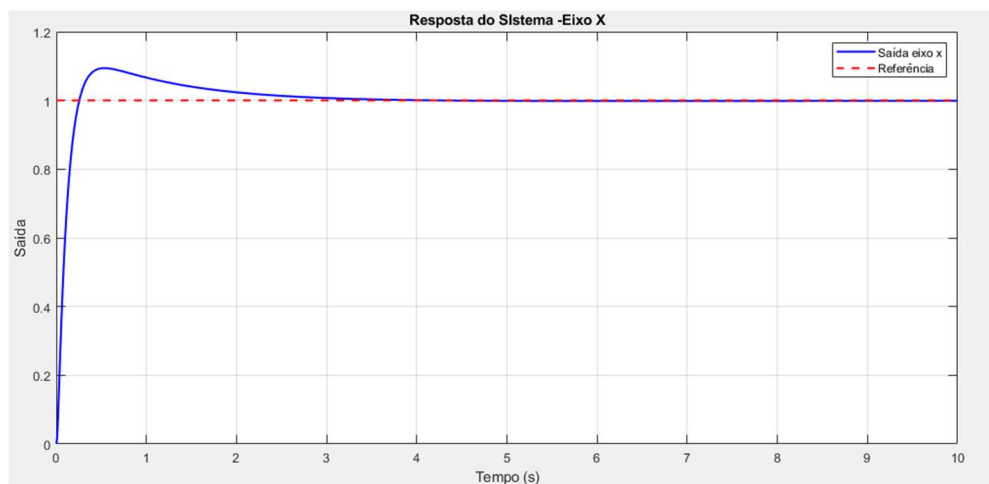
A Tabela 3 mostra os parâmetros utilizados na simulação do sistema de controle em malha fechada para o eixo X, assim como os requisitos de desempenho obtidos na simulação. Como a função de transferência para o eixo Y é similar, os mesmos ganhos foram utilizados nos dois eixos. A Figura 53 mostra o resultado da simulação.

Tabela 3 – Parâmetros do controlador utilizados na simulação em malha fechada e resultados.

Parâmetro	Valor
Ganho proporcional (K_p)	8,332
Ganho integrativo (K_i)	1,404
Ganho derivativo (K_d)	8,233
Fator de filtragem (N)	86,747
Tempo de amostragem (T_s)	20 ms
Tempo de acomodação	2,165 s
Sobressinal	9,4%
Erro em regime permanente	Nulo

Fonte: O autor, 2025.

Figura 53 – Resultado da simulação em malha fechada do sistema com entrada em degrau unitário.



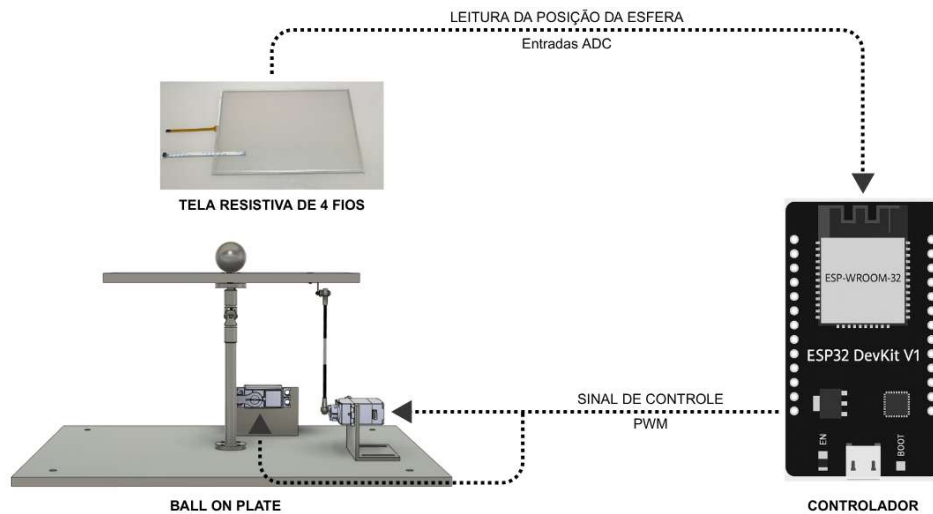
Fonte: O autor, 2025.

3.5 Controle do Sistema por Realimentação Tradicional

A arquitetura de funcionamento do sistema de controle pelo método tradicional de realimentação é mostrada na Figura 54. Inicialmente, a posição da esfera no prato é lida pela ESP32 através de entradas que convertem uma leitura analógica em digital (entradas ADC). As entradas ADC são conectadas aos 4 fios da tela resistiva.

Posteriormente, a posição detectada da esfera é passada para o algoritmo que é executado na unidade de controle, que por sua vez, através de saídas PWM (*Pulse Width Modulation*), controlam a posição angular dos servomotores para ajuste da posição.

Figura 54 – Arquitetura do sistema de controle com realimentação tradicional.



Fonte: O autor, 2025.

3.5.1 Aquisição da Posição da Esfera com Tela Resistiva de 4 Fios

Os pinos destinados à medição analógica dos sinais da tela resistiva são mostrados na Tabela 4. Esses sinais são lidos por meio de entradas ADC com resolução de 12 bits, variando o sinal de 0 a 4095.

Tabela 4 – Diagrama de ligação da Esp32 com a tela resistiva touchscreen de 4 fios.

Terminal da tela resistiva	X+	Y+	X-	Y-
Pino da ESP32	GPIO 13	GPIO 12	GPIO 14	GPIO 27
Número/canal do ADC da ESP32	ADC2 CH4	ADC2 CH5	ADC2 CH6	ADC2 CH7

Fonte: O autor, 2025.

Foi realizada uma etapa de calibração da tela resistiva, na qual foram determinados os valores mínimos e máximos das leituras nos eixos X e Y. Esses valores foram utilizados para escalonar os dados adquiridos em milímetros, de acordo com as dimensões físicas reais da tela (225 mm no eixo X e 173 mm no eixo Y). Esse processo assegura que os dados enviados ao algoritmo de controle representem com precisão a posição da esfera sobre a superfície.

Além disso, foi implementado um filtro passa baixa para a leitura da posição da esfera. Verificou-se que o sensor resistivo apresentava muitos ruídos de alta frequência que comprometeriam o controle, e a implementação de um filtro passa baixa os atenuaria.

Dessa forma, foi utilizado um filtro digital do tipo primeira ordem, com frequência de corte de aproximadamente 5 Hz, considerando a frequência de amostragem de 50 Hz. A equação do filtro digital utilizado para leitura da posição da esfera, discretizado pelo método de Euler, é mostrado a seguir:

$$xf(n) = \alpha \cdot x[n] + (1 - \alpha) \cdot xf[n - 1], \quad (3.66)$$

em que:

- $x[n]$ é a leitura sem filtro da posição;
- $xf(n)$ é o valor filtrado;
- α é o coeficiente de suavização que depende dos parâmetros escolhidos para o filtro.

A expressão de α é mostrada a seguir:

$$\alpha = \frac{2 \cdot \pi \cdot f_c \cdot T_s}{2 \cdot \pi \cdot f_c \cdot T_s + 1}, \quad (3.67)$$

onde f_c representa a frequência de corte e T_s o período de amostragem. O filtro descrito foi implementado separadamente na leitura dos dois eixos do sistema com $\alpha = 0,386$.

3.5.2 Algoritmo de Controle

A Figura 55 mostra o fluxograma do algoritmo de controle, implementado no Arduino IDE, para o sistema *Ball on Plate*. O código está estruturado em duas macroetapas principais: inicialização e loop principal.

A etapa de inicialização é executada apenas uma vez, no início do programa. Já a etapa do loop principal é cíclica, ou seja, suas instruções são repetidas

continuamente ao longo da execução do sistema. As tarefas executadas durante a inicialização são:

- Definição e inicialização das variáveis e constantes;
- Inicialização da comunicação serial com taxa de 115200 bps;
- Configuração dos pinos da mesa resistiva e dos servomotores;
- Configuração dos canais PWM (frequência e resolução);
- Centralização dos servomotores;
- Cálculo dos coeficientes do PID.

A etapa do loop principal está dividida em subetapas menores numeradas e são explicadas abaixo:

1. Verificação do tempo decorrido:

- Calcula o tempo decorrido do código com o objetivo de implementar a amostragem do sistema;
- Quando decorridos 20 ms, o código prossegue para a leitura da posição da esfera.

2. Leitura da posição da esfera:

- Leitura da tensão nos eixos X e Y da mesa resistiva;
- Conversão da leitura bruta para coordenadas em milímetros;
- Aplicação de filtro passa-baixa nas posições;
- Armazenamento das posições filtradas em posX e posY.

3. Cálculo dos erros:

- $\text{erroX} = \text{setpointX} - \text{posX}$;
- $\text{erroY} = \text{setpointY} - \text{posY}$.

4. Controle PID – Eixo X:

- Cálculo das componentes proporcional, integral e derivativa;
- Soma das ações $U_x = P + I + D$;

- Cálculo e limitação do sinal PWM;
- Envio do PWM ao servomotor X.

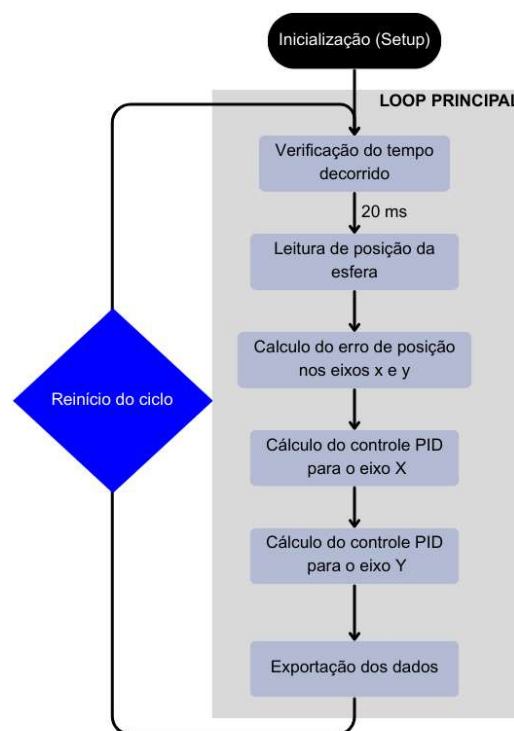
5. Controle PID – Eixo Y:

- Etapa similar ao do eixo X, porém utilizando as variáveis do respectivo eixo.

6. Exportação dos dados via comunicação serial:

- Envio do tempo em milissegundos, das posições atuais x e y da esfera e dos *setpoint* em cada eixo;
- Retorna ao início do *loop* e aguarda novo ciclo.

Figura 55 – Fluxograma do algoritmo de controle implementado para o sistema *Ball on Plate* no método tradicional de realimentação.



Fonte: O autor, 2025.

Os pinos PWM destinados à conexão com os servomotores e suas configurações são mostrados na Tabela 5. O código completo do controle pelo método tradicional pode ser encontrado no GitHub, disponível no Apêndice B.

Tabela 5 – Diagrama de ligação da Esp32 com os servomotores e configurações.

Servomotor	Pino	Canal PWM	Resolução do PWM	Frequência do PWM
Servo X	GPIO 33	0	16 bits	50 Hz
Servo Y	GPIO 15	1	16 bits	50 Hz

Fonte: O autor, 2025.

3.5.3 Definição dos Parâmetros PID do Controlador Utilizados na Planta e Ajuste Fino

A expressão do controlador PID implementado é apresentada na Equação (3.65), a qual considera um filtro na ação derivativa e discretização pelo método de Tustin para a ação integral. O período de amostragem e os ganhos empregados no controlador estão listados na Tabela 6.

Tabela 6 – Parâmetros do controlador utilizados na implementação em malha fechada.

Parâmetro	Valor
Ganho proporcional (K_p)	7,0
Ganho integrativo (K_i)	1,0
Ganho derivativo (K_d)	4,0
Fator de filtragem (N)	15,0
Período de amostragem (T_s)	20 ms

Fonte: O autor, 2025.

Inicialmente, os valores utilizados foram os obtidos na fase de simulação, mostrados na Tabela 3. Porém, esses parâmetros resultaram em um comportamento não satisfatório: o sistema apresentou uma resposta excessivamente agressiva e não foi capaz de controlar a posição da esfera.

Diante disso, optou-se por reiniciar o processo de ajuste, zerando todos os ganhos do controlador. O procedimento de sintonia seguiu a seguinte estratégia:

- **Ajuste do ganho proporcional (K_p):** O ganho proporcional foi aumentado gradualmente até que o sistema passasse a reagir de forma coerente quando

a esfera atingia as extremidades da tela resistiva, inclinando a plataforma de forma adequada;

- **Ajuste do ganho derivativo (K_d) e do parâmetro do filtro (N):** Com o ganho proporcional ajustado, foi aumentando-se gradualmente o ganho K_d e o parâmetro N com o objetivo de intensificar a ação de amortecimento;
- **Ajuste do ganho integrativo (K_i):** Por fim, o ganho integrativo foi incrementado de forma a eliminar o erro em regime permanente, garantindo que a esfera permanecesse no *setpoint* desejado.

Após a realização desses ajustes, foi possível alcançar uma resposta dinâmica satisfatória, e os parâmetros finais foram registrados (Tabela 6).

4 CONTROLE DO SISTEMA *BALL ON PLATE* POR VISÃO COMPUTACIONAL

Este capítulo tem como objetivo apresentar a abordagem de controle baseada em visão computacional. Inicialmente, é feita a descrição do sistema, com foco na aplicação da visão computacional ao sistema *Ball on Plate*. Em seguida, são detalhados os processos desenvolvidos no código de detecção e o fechamento da malha de controle, integrando a posição da esfera como variável de realimentação.

4.1 Descrição do Sistema

A arquitetura de funcionamento do sistema de controle por visão computacional é mostrada na Figura 56. Inicialmente, a imagem é capturada pela webcam e enviada via conexão USB para o computador. O processamento da imagem é feito com o objetivo de detectar, em tempo real, a posição da esfera sobre a superfície. Esse dado é enviado via comunicação serial (USB-UART) para unidade de controle, representada pela ESP32, que executa o algoritmo de controle responsável por ajustar a inclinação da base por meio dos servomotores.

Figura 56 – Arquitetura do sistema de controle por visão computacional no *Ball on Plate*.



Fonte: O autor, 2025.

4.2 Processamento de Imagem para Detecção da Esfera

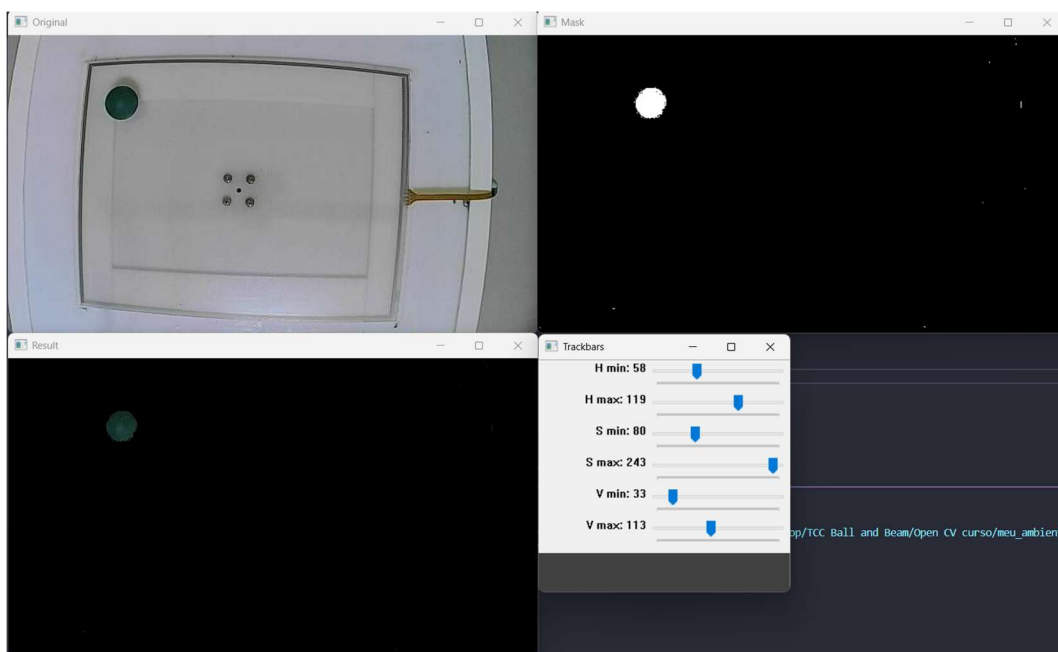
Nesta seção, são descritas as principais etapas do processamento de imagem desenvolvidas para detecção da esfera em tempo real. Inicialmente, são apresentadas duas etapas de calibração, uma voltada à segmentação por cor e outra à delimitação da área de atuação do sistema. Em seguida, é detalhado o algoritmo final de detecção da esfera, que utiliza as configurações definidas nas etapas anteriores.

4.2.1 Calibração de Parâmetros para Segmentação por Cor

O primeiro passo para a detecção da esfera por cor consistiu na definição dos parâmetros matiz, saturação e valor (*Hue*, *Saturation* e *Value*, HSV) que melhor identificassem a tonalidade verde. Para isso, foi desenvolvido um código em Python utilizando a biblioteca OpenCV, responsável por realizar a calibração da segmentação da esfera. Nessa etapa, os limites mínimo e máximo de cada componente HSV foram ajustados manualmente, permitindo observar em tempo real o resultado da segmentação na imagem. O processo foi realizado no ambiente de desenvolvimento Visual Studio Code (VS Code), amplamente utilizado por sua interface intuitiva, suporte a múltiplas linguagens e recursos avançados de depuração.

A Figura 57 ilustra essa etapa de calibração, na qual os valores dos parâmetros HSV foram ajustados até se obter a segmentação desejada, e os valores finais definidos para os parâmetros são apresentados na Tabela 7.

Figura 57 – Etapa de calibração dos parâmetros H, S e V para detecção da esfera.



Fonte: O autor, 2025.

Tabela 7 – Valores dos parâmetros H, S e V utilizados para a detecção da esfera.

Parâmetro	Limite inferior	Limite superior
H (matiz)	58	119
S (saturação)	80	243
V (valor)	33	113

Fonte: O autor, 2025.

4.2.2 Delimitação da Região de Atuação da Esfera

O passo seguinte na implementação do sistema de controle por visão computacional foi a calibração da área de atuação, ou seja, a definição dos limites dentro dos quais a esfera deve ser mantida equilibrada. Essa etapa é fundamental para estabelecer a região de controle, cujo centro corresponde à posição de equilíbrio desejada. Utilizando a mesma linguagem (Python), biblioteca (OpenCV) e ambiente de desenvolvimento (Visual Studio Code), foi desenvolvido um código específico para detectar os limites dessa área com base na imagem do sistema. Inicialmente, buscou-se definir esses limites com base nas dimensões físicas da tela resistiva, utilizada no

método tradicional de realimentação. A Figura 58 mostra o resultado dessa implementação.

Figura 58 – Definição dos limites da região de controle para posicionamento da esfera no centro.



Fonte: O autor, 2025.

O resultado da calibração da área de atuação foi armazenado em um arquivo no formato JSON (JavaScript Object Notation), contendo os limites mínimos e máximos da área de controle. Esse tipo de arquivo é amplamente utilizado para armazenar e transmitir dados de forma estruturada. Nessa aplicação, o uso do arquivo JSON também possibilita a reutilização dos parâmetros no processamento de imagem, além da facilidade de uma nova calibração, caso necessário. Essas informações foram posteriormente utilizadas no algoritmo final de detecção e rastreamento da esfera.

Com os valores ideais de H, S e V definidos e a área de controle delimitada, foi implementado o algoritmo de detecção de cor, que realiza a leitura em tempo real da posição da esfera e se comunica com a ESP32 por meio de interface serial, possibilitando o envio das coordenadas para o sistema de controle.

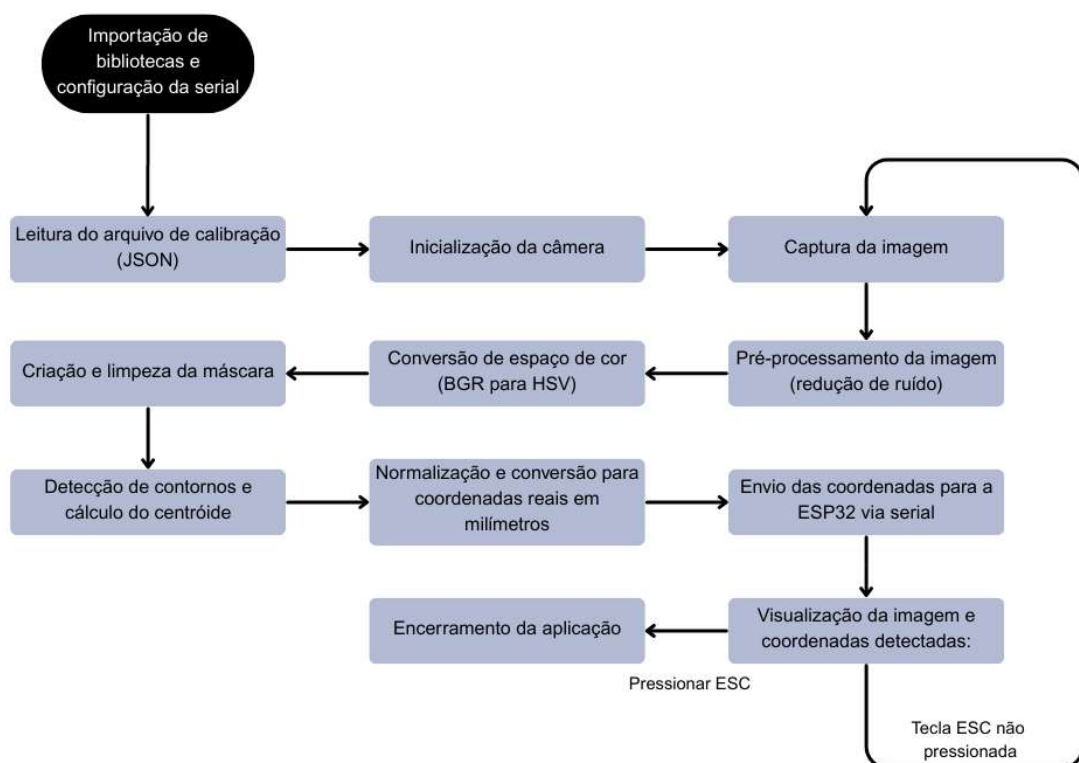
4.2.3 Algoritmo Final de Rastreamento da Esfera

O fluxograma do algoritmo final de rastreamento da esfera, também desenvolvido no VScode com Python, é mostrado na Figura 59. As etapas são descritas abaixo:

1. **Importação de bibliotecas e configuração da serial:** nessa etapa, as bibliotecas necessárias (OpenCV, Numpy, Serial, json e os) são carregadas para processamento de imagem, comunicação com a ESP32 e leitura de arquivo de calibração. Em seguida, estabelece-se comunicação com a porta serial COM3 a 115200 bps;
2. **Leitura do arquivo de calibração (JSON):** O arquivo de calibração é carregado, contendo os valores que delimitam a área calibrada onde a esfera deve permanecer;
3. **Inicialização da câmera:** A webcam é ativada com taxa de aquisição de 30 quadros por segundo;
4. **Captura da imagem:** A imagem é capturada frame a frame para processamento;
5. **Pré-processamento da imagem (redução de ruído):** aplica-se um filtro de suavização vertical para reduzir os ruídos e melhorar a segmentação;
6. **Conversão de espaço de cor (BGR para HSV):** a imagem é convertida do formato padrão da câmera (BGR) para HSV, ideal para detecção de cores;
7. **Criação e limpeza da máscara binária:** Uma máscara é criada com base nos limites HSV previamente estabelecidos. Em seguida, são aplicadas operações morfológicas de abertura e fechamento para eliminar ruídos e preencher falhas;
8. **Deteção de contornos e cálculo do centroide:** localiza-se o maior contorno detectado na máscara e calcula-se seu centroide (com base nos momentos da forma), representando a posição da esfera;
9. **Normalização e conversão para coordenadas reais em milímetros:** As coordenadas do centróide são normalizadas com base nos limites da área calibrada e convertidas para milímetros, considerando as dimensões reais da tela resistiva (225 mm x 173 mm);
10. **Envio das coordenadas para a ESP32 via serial:** as coordenadas reais da esfera são enviadas à ESP32 no formato de texto para que possam ser utilizadas no sistema de controle;

11. **Visualização da imagem e coordenadas detectadas:** A imagem original é exibida com a área de controle destacada (retângulo verde), a posição da esfera marcada (círculo vermelho) e os valores das coordenadas sobrepostos;
12. **Encerramento da aplicação:** se a tecla ESC for pressionada, o loop é encerrado, os recursos são liberados e a conexão com a serial é encerrada. Caso contrário, o programa volta para o loop de captura da imagem.

Figura 59 – Fluxograma do algoritmo final de rastreamento da esfera.



Fonte: O autor, 2025.

O código completo do algoritmo de detecção da esfera pode ser encontrado no GitHub, disponível no Apêndice B.

4.3 Fechamento de Malha

A Figura 60 mostra o fluxograma do algoritmo de controle por visão computacional, implementado no Arduino IDE, para o sistema *Ball on Plate*. O código

está estruturado da mesma forma que no método tradicional de realimentação, com diferença de que, nessa implementação, a posição da esfera é lida através dos dados recebidos na porta COM3 via comunicação serial. Esses dados são enviados pelo algoritmo de detecção da posição da esfera por visão computacional. As tarefas executadas durante a inicialização do sistema são descritas abaixo:

- Definição e inicialização das variáveis e constantes;
- Inicialização da comunicação serial e serial bluetooth com taxa de 115200 bps;
- Configuração dos pinos da mesa resistiva e dos servomotores;
- Configuração dos canais PWM (frequência e resolução);
- Centralização dos servomotores;
- Cálculo dos coeficientes do PID.

A etapa do loop principal está dividida em subetapas menores numeradas e são explicadas abaixo:

1. Verificação do tempo decorrido:

- Calcula o tempo decorrido do código com o objetivo de implementar a amostragem do sistema;
- Quando decorridos 20 ms, o código prossegue para a leitura da posição da esfera via comunicação serial.

2. Leitura da posição da esfera via comunicação serial:

- Verifica a disponibilidade de dados na porta serial;
- Leitura da string enviada pela câmera com os dados das posições nos eixos X e Y;
- Armazenamento das posições em posX e posY.

3. Cálculo dos erros:

- $\text{erroX} = \text{setpointX} - \text{posX}$;
- $\text{erroY} = \text{setpointY} - \text{posY}$.

4. Controle PID – Eixo X:

- Cálculo das componentes proporcional, integral e derivativa;
- Soma das ações $U_x = P + I + D$;
- Cálculo e limitação do sinal PWM;
- Envio do PWM ao servomotor X.

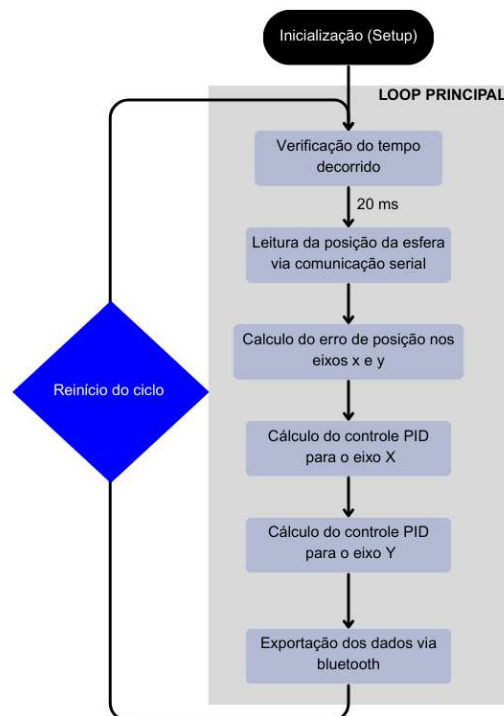
5. Controle PID – Eixo Y:

- Etapa similar ao do eixo X, porém utilizando as variáveis do respectivo eixo.

6. Exportação dos dados via bluetooth:

- Envio do tempo em milissegundos, das posições atuais x e y da esfera e dos *setpoint* em cada eixo;
- Retorna ao início do *loop* e aguarda novo ciclo.

Figura 60 – Fluxograma do algoritmo de controle implementado para o sistema *Ball on Plate* no método de realimentação por visão computacional.



Fonte: O autor, 2025.

Os pinos PWM destinados à conexão com os servomotores e suas configurações são os mesmos utilizados no método tradicional de realimentação, mostrados na Tabela 5. O código completo do controle por visão computacional pode ser encontrado no GitHub, disponível no Apêndice B..

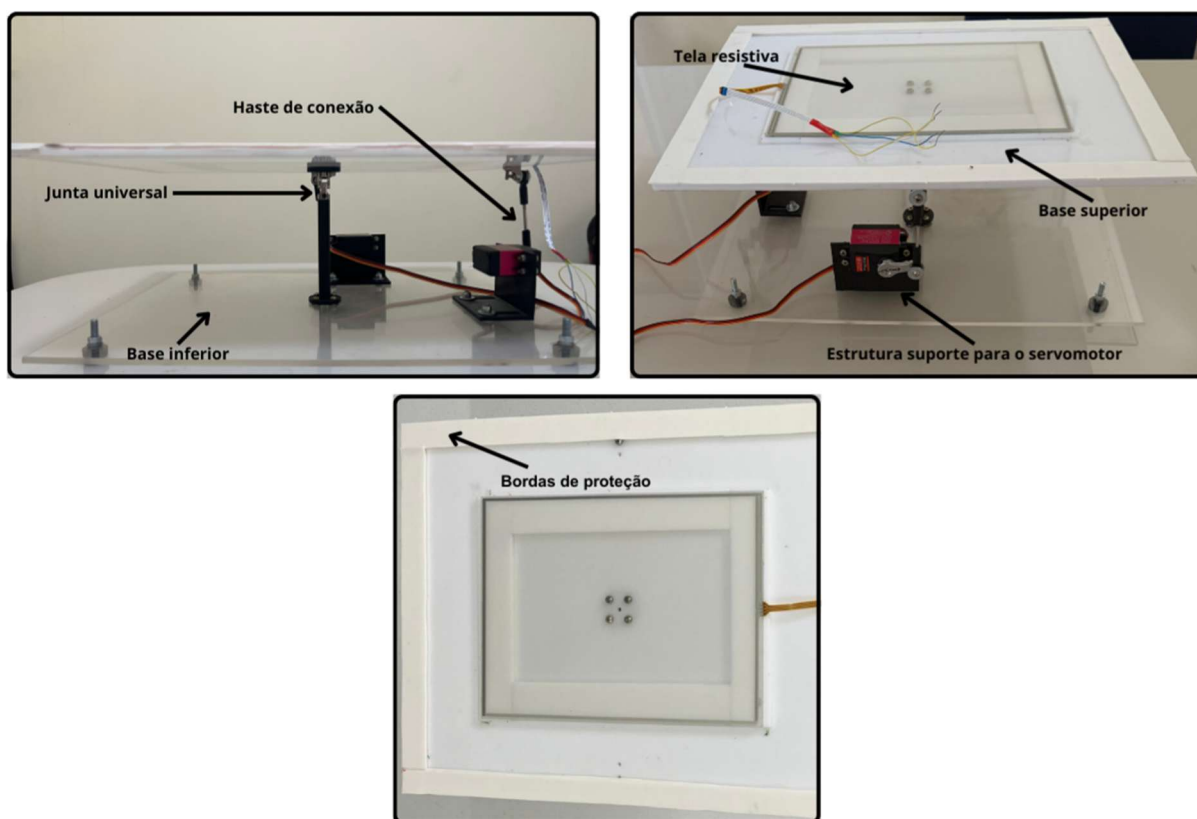
5 RESULTADOS EXPERIMENTAIS E ANÁLISE

Este capítulo apresenta os resultados obtidos ao longo do desenvolvimento deste trabalho e sua respectiva análise, destacando os diferentes métodos de realimentação empregados. No Apêndice A, é disponibilizado um link para visualização dos resultados experimentais.

5.1 Estrutura Mecânica

A Figura 61 apresenta a estrutura mecânica do sistema *Ball on Plate* desenvolvida neste trabalho, exibida em três vistas diferentes. Nela, destacam-se as principais peças e componentes que compõem o sistema, como a tela resistiva, os servomotores, a junta universal, uma borda de proteção feita com emborrachado para a esfera não cair no chão, entre outros.

Figura 61 – Estrutura de suporte para posicionamento da câmera.

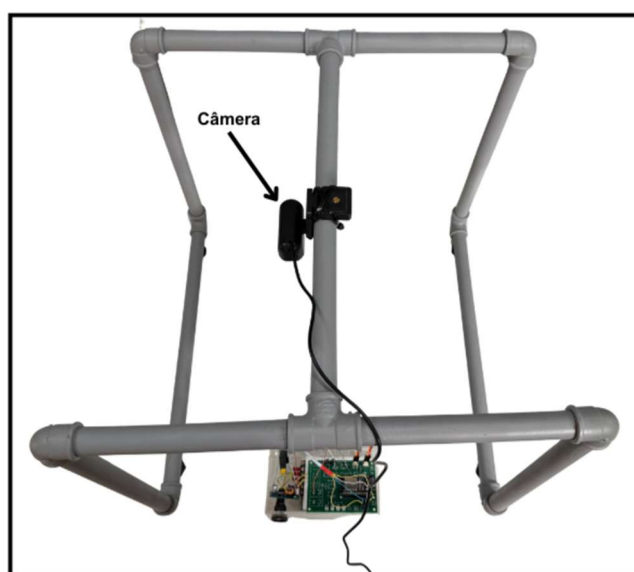


Fonte: O autor, 2025.

5.2 Estrutura Suporte Para Câmera

Foi desenvolvida uma estrutura para suporte da câmera, destinado ao método de realimentação por visão computacional. Essa estrutura é apresentada na Figura 62, destacando-se o posicionamento central da câmera. A estrutura foi desenvolvida com cano PVC, onde foram utilizadas conexões em T e joelhos.

Figura 62 – Estrutura de suporte desenvolvida para fixação e posicionamento da câmera de aquisição de imagens.

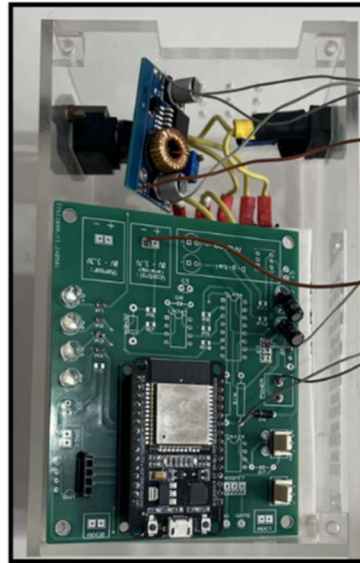


Fonte: O autor, 2025.

5.3 Circuito de Alimentação e Placa de Controle

A Figura 63 apresenta o circuito de alimentação e a placa de controle utilizados no desenvolvimento deste trabalho. Para fins de organização e proteção dos componentes eletrônicos, os elementos foram abrigados em uma caixa de acrílico.

Figura 63 – Circuito de alimentação e placa de controle utilizados no sistema abrigados em uma caixa de acrílico.

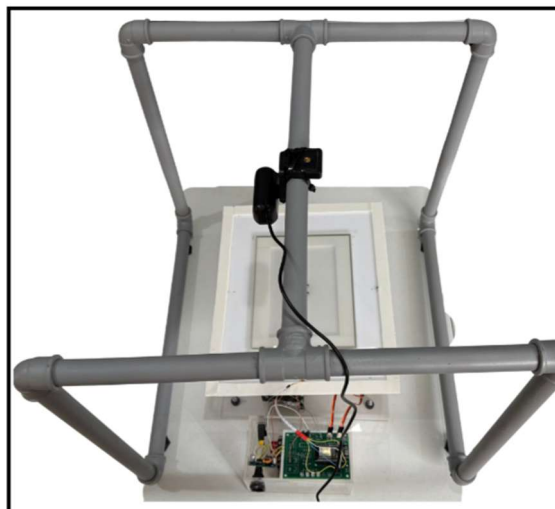


Fonte: O autor, 2025.

5.4 Estrutura Completa

A Figura 64 apresenta a estrutura completa do sistema, com o objetivo de ilustrar a integração entre os diferentes módulos desenvolvidos. É possível observar o conjunto formado pela estrutura mecânica do sistema *Ball on Plate*, a estrutura de sustentação da câmera e a caixa de acrílico que abriga a placa de circuito impresso.

Figura 64 – Estrutura completa do sistema integrando os módulos mecânico, eletrônico e de visão computacional.



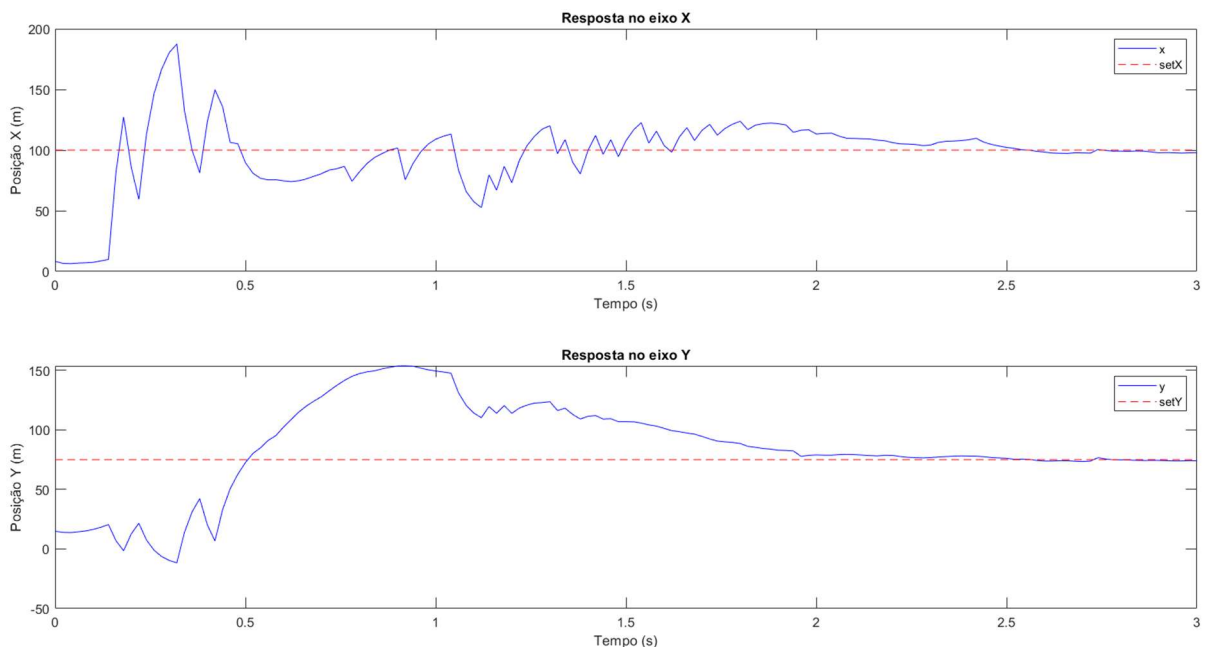
Fonte: O autor, 2025.

5.5 Resultados Obtidos no Método Tradicional de Realimentação

Esta seção apresenta os resultados obtidos com o método tradicional de realimentação, utilizando um controlador PID em malha fechada. Os testes realizados consistiram em colocar a esfera na posição inicial (0,0) no sistema de coordenadas da tela resistiva, e registrar a dinâmica até sua estabilização no centro. Devido à similaridade da dinâmica, os mesmos parâmetros, descritos na Tabela 6, foram utilizados no controle dos dois eixos (X e Y).

A Figura 65 apresenta as respostas nos eixos X e Y do sistema, com a esfera verde, obtidas por meio do MATLAB. Os dados de tempo, posição nos eixos X e Y, bem como os respectivos *setpoints* (setX e setY), foram exportados do sistema e utilizados para análise do desempenho do controle. A posição central da tela resistiva foi escolhida como o *setpoint* do sistema, correspondendo às coordenadas calibradas de 100 mm no eixo X e 75 mm no eixo Y. A Tabela 8 mostra as medidas de desempenho obtidas no sistema de controle, com a esfera verde.

Figura 65 – Resposta dos eixos X e Y no método tradicional de realimentação com a esfera verde.



Fonte: O autor, 2025.

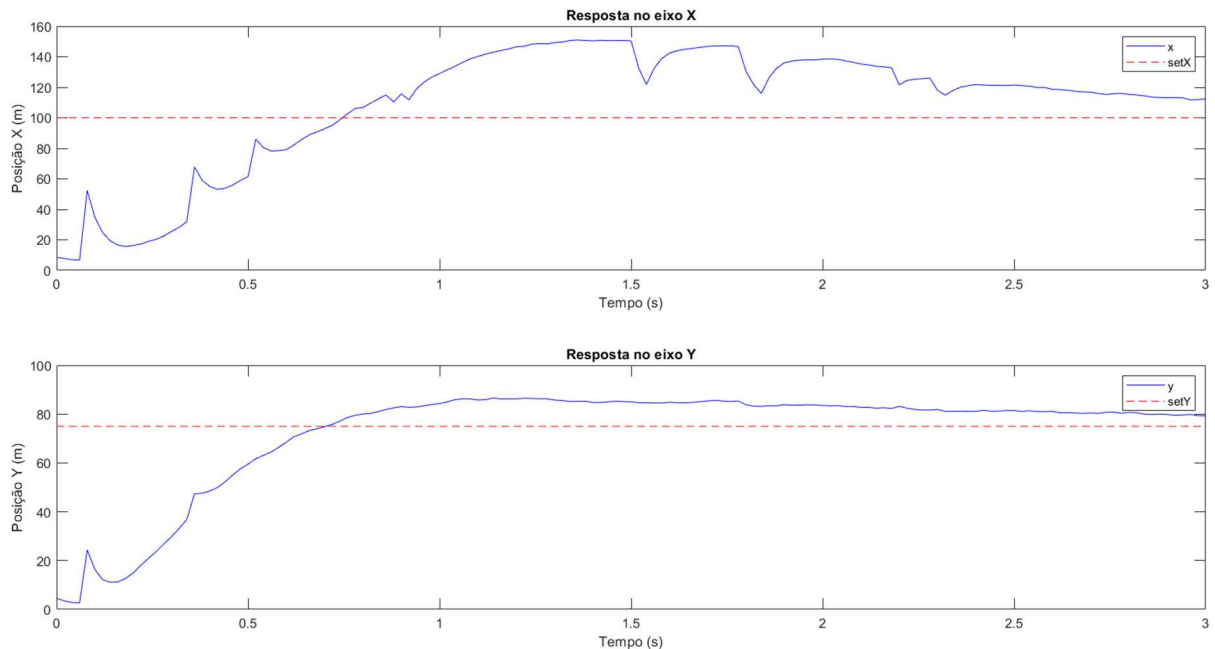
Tabela 8 – Medidas de desempenho obtidas para os eixos X e Y no método tradicional de realimentação com a esfera verde.

Medida de desempenho	Eixo X	Eixo Y
Máximo sobressinal percentual	87,5%	104,9%
Tempo de acomodação	2,46 segundos (5%) 2,73 segundos (2%)	2,00 segundos (5%) 2,28 segundos (2%)
Erro estacionário	Aproximadamente zero	Aproximadamente zero

Fonte: O autor, 2025.

Com o objetivo de analisar a influência de diferentes características físicas da esfera no comportamento do sistema, foi utilizada uma esfera de aço não pintada, com maior diâmetro e massa, substituindo a esfera verde na estrutura de controle. O resultado é mostrado na Figura 66. A Tabela 9 mostra as medidas de desempenho obtidas na implementação com a esfera de aço.

Figura 66 – Resposta dos eixos X e Y no método tradicional de realimentação com a utilização da esfera de aço (esfera prata).



Fonte: O autor, 2025.

Tabela 9 – Medidas de desempenho obtidas para os eixos X e Y no método tradicional de realimentação com a esfera de aço (esfera prata).

Medida de desempenho	Eixo X	Eixo Y
Máximo sobressinal percentual	34,04%	8,90%
Tempo de acomodação	2,59 segundos (5%) 2,85 segundos (2%)	2,01 segundos (5%) 2,61 segundos (2%)
Erro estacionário	12,50	4,5

Fonte: O autor, 2025.

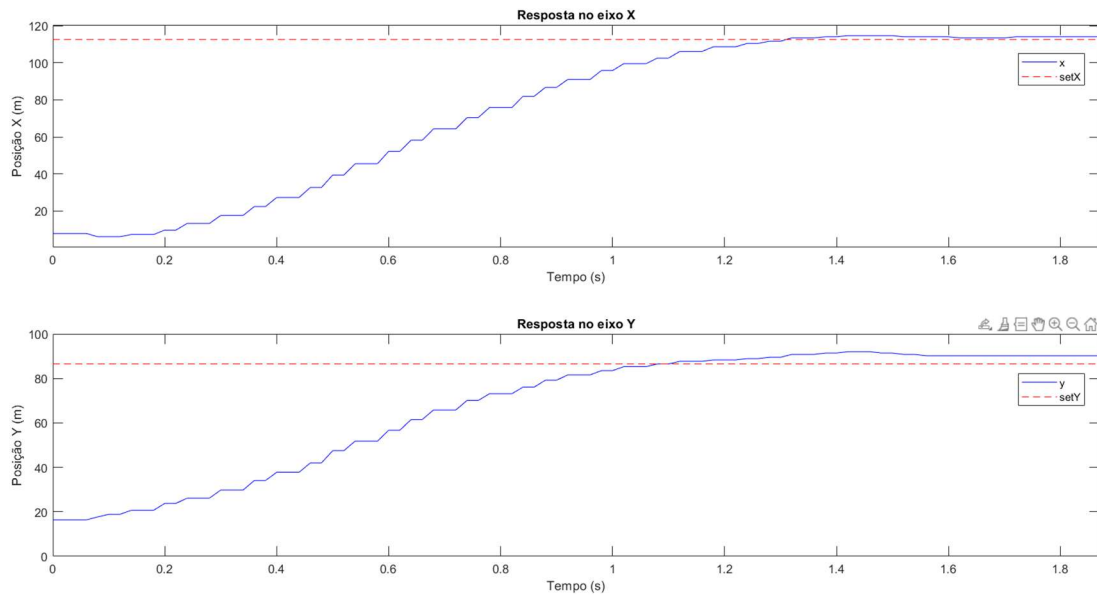
Observa-se que, nas respostas obtidas pelo método tradicional, há presença de ruídos e sobressinais elevados, o que faz com que o sistema atinja a estabilidade apenas entre 2 e 3 segundos. Embora o sistema tenha se mostrado estável, era esperado um comportamento mais suave. Essa característica se deve, principalmente, ao uso da tela resistiva, que introduz ruídos de alta frequência na medição da posição da esfera.

5.6 Resultados Obtidos no Método de Realimentação Por Visão Computacional

Esta seção apresenta os resultados obtidos no método de realimentação por visão computacional. Com o objetivo de efetuar a comparação com o método tradicional, foram utilizados os mesmos parâmetros do controlador adotados anteriormente, descritos na Tabela 6. A posição central da área calibrada no método de visão computacional foi escolhida como o *setpoint* do sistema, correspondendo às coordenadas calibradas de 112,5 mm no eixo X e 86,5 mm no eixo Y (tomando como base a tela resistiva).

A Figura 67 apresenta as respostas nos eixos X e Y do sistema, com a esfera verde. Foi realizado o mesmo teste do método tradicional, onde a esfera foi colocada na posição inicial (0,0) do sistema de coordenadas, e sua dinâmica foi registrada até a estabilização no centro. A Tabela 10 mostra as medidas de desempenho obtidas no sistema de controle, com a esfera verde detectada pela câmera.

Figura 67 – Resposta dos eixos X e Y no método de realimentação por visão computacional com a esfera verde.



Fonte: O autor, 2025.

Tabela 10 – Medidas de desempenho obtidas para os eixos X e Y no método de realimentação por visão computacional com a esfera verde.

Medida de desempenho	Eixo X	Eixo Y
Máximo sobressinal percentual	0,50 %	1,99 %
Tempo de acomodação	1,17 segundos (5%) 1,31 segundos (2%)	1,07 segundos (5%) 1,51 segundos (2%)
Erro estacionário	1,5	3,7

Fonte: O autor, 2025.

Na resposta obtida pelo método de visão computacional, observou-se um comportamento significativamente mais estável. O sistema atingiu a estabilidade em aproximadamente 1 a 1,5 segundos, evidenciando uma resposta mais rápida. Apesar da expectativa de que o atraso inerente ao processamento das imagens pudesse comprometer o desempenho do controle, o sistema apresentou estabilidade, menor sobressinal e maior suavidade na resposta, demonstrando-se mais eficiente e facilmente controlável.

5.7 Análise dos Resultados

Comparando os resultados obtidos nas Seções 5.5 (com a esfera verde) e 5.6, observa-se uma resposta mais suave e controlada no método por visão computacional.

No método tradicional é notável a presença de oscilações acentuadas com um grande sobressinal para os dois eixos: 87,50 % no eixo X e 104,90 % no eixo Y. Já no método por visão, esses números reduzem-se a 0,50 % e 1.99%.

Em relação ao tempo de acomodação, usando os critérios de 5% e 2% em torno do valor em regime permanente, o método por visão computacional se mostrou mais rápido. Enquanto que o sistema estabilizou na faixa de 2-3 segundos no método tradicional, na segunda implementação essa faixa cai para 1-1,5 segundos.

Observa-se ainda na implementação por visão computacional um atraso inicial de resposta do sistema, característico da taxa de atualização da câmera, do processamento da imagem e envio das coordenadas. Mesmo assim, o sistema se mostrou rápido e robusto.

Com relação à precisão do sistema, nas duas implementações um resultado satisfatório foi atingido, apesar de na implementação tradicional o erro ser mais próximo de zero, enquanto que valores pequenos foram obtidos no método por visão computacional: 1,50 (1,33 %) e 3,70 (4,27%).

Comparando os resultados obtidos na Seção 5.5, entre as esferas verde (pintada) e prata (sem pintura, com maior diâmetro e massa), observou-se que a esfera prata apresentou menor sobressinal. Porém, um maior erro em regime permanente foi observado. O tempo de assentamento permaneceu dentro da mesma faixa nas duas implementações.

Foi possível observar, ainda, que a esfera verde, por ser pintada, apresentava maior atrito com a superfície da mesa, o que fazia com que aderisse mais ao plano durante os movimentos. Já a esfera prata, mais lisa e pesada, demonstrou uma dinâmica mais fluida e contínua.

6 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

Este capítulo tem como objetivo apresentar as conclusões obtidas no desenvolvimento do trabalho. Inicialmente, são discutidas as principais dificuldades encontradas durante a implementação do sistema, seguidas das conclusões finais. Por fim, sugestões de melhorias e propostas de continuidade são apresentadas.

6.1 Dificuldades Encontradas

São listadas, a seguir, as principais dificuldades encontradas durante a implementação do sistema:

- **Haste de conexão dos servomotores:** Neste trabalho, utilizou-se a haste de conexão com juntas esféricas nas extremidades (Figura 35). Esse tipo de junta apresenta folgas e movimentos indesejáveis no eixo, de forma que foi tentado mitigar esses movimentos com a inserção de arruelas, mas que não eliminou de fato o problema;
- **Estrutura de suporte para os servomotores:** Observou-se, no funcionamento do sistema, que a estrutura de suporte dos servomotores (impressa em uma impressora 3D) não apresentou rigidez necessária para total fixação na base, de forma que de acordo com o movimento dos servos, a estrutura balança;
- **Interferência de iluminação no método de visão computacional:** Durante a implementação do sistema de controle com visão computacional, observou-se que variações na iluminação do ambiente, sombras e reflexos causavam muita interferência no processo de detecção da esfera base no espaço de cor HSV. Com isso, ocorriam detecções imprecisas que afetavam o sistema de controle. Porém, quando a iluminação era homogênea, o sistema funcionava perfeitamente;
- **Ruído na mesa resistiva:** No método tradicional de realimentação, baseado na mesa resistiva, ruídos elétricos e leituras imprecisas afetaram o sistema. Para mitigar o problema, um filtro passa baixa foi inserido no processo de aquisição da posição, mas não solucionou o problema por completo. Mesmo

sem nenhum toque na superfície da tela, o sistema apresentava leituras inconsistentes. Além disso, verificou-se que as leituras tinham certos limites, de forma que não detectavam os extremos da mesa como coordenada nula (0,0) e coordenada máxima (273, 173, dimensões da tela resistiva). Para solucionar esse problema, um processo de escalonamento das coordenadas foi feito;

- **Calibração e mapeamento das coordenadas:** A calibração da tela resistiva e da área de detecção no método por visão, foi uma etapa crucial na implementação do controle do sistema. Pequenas variações no posicionamento da câmera exigiam a recalibração do sistema.

Mesmo com as principais dificuldades encontradas, o sistema mostrou-se funcional e satisfatório nas duas implementações.

6.2 Conclusões Finais

Conclui-se que, apesar de ser uma implementação mais complexa, o método por visão computacional se mostrou mais robusto e estável, comparado à implementação tradicional. Os ruídos, provenientes da leitura da posição na mesa resistiva, provocam grandes oscilações e sobressinal, o que é indesejado ao sistema, e além disso, impõem um atraso na sua estabilização. O método por visão, apesar de ter o atraso relacionado à captura da imagem e seu processamento, mostrou-se satisfatório no período de amostragem implementado (20 ms).

A comparação entre as esferas prata e verde permitiu identificar como suas propriedades físicas influenciam diretamente a dinâmica do sistema. A esfera prata, por possuir maior massa, e consequentemente maior inércia, tende a resistir a movimentos rápidos da mesa, o que explica a diminuição das oscilações e do sobressinal do sistema. Porém, o mesmo fato faz com que pequenas correções na posição não sejam realizadas, resultando em um maior erro em regime permanente. Além disso, os ruídos, provenientes da leitura da posição com a tela resistiva, não provocam grande instabilidade como na utilização da esfera verde, pois esses ruídos não são suficientes para mover a esfera de posição, em razão da sua maior inércia.

Com relação à textura das esferas, a esfera verde, por ser pintada, apresentou dinâmica menos fluida do que a esfera lisa, sem estar pintada. Isso ocorre em razão de que a superfície pintada fica mais rugosa, o que faz com que ela se agarre mais à superfície. Já a esfera prata, sem pintura e de superfície lisa, apresenta movimento mais contínuo e natural.

Por fim, é importante destacar que, em todas as implementações realizadas, o sistema foi capaz de controlar eficientemente a posição da esfera, mesmo diante de perturbações externas, como toques manuais aplicados intencionalmente. O sistema demonstrou capacidade de recuperar a trajetória e restabelecer a estabilidade no ponto de referência, evidenciando a eficácia da estratégia de controle empregada.

6.3 Pontos de Melhoria

Visando eliminar as dificuldades encontradas no sistema, os seguintes pontos de melhoria são apresentados:

- Substituir as hastes de conexão dos servomotores por um modelo que não possua juntas esféricas nas extremidades, removendo assim movimentos indesejáveis no sistema, tornando-o mais rígido;
- Substituir as estruturas de suporte dos servomotores por um modelo com material mais rígido (metal ou acrílico espesso), a fim de aumentar a estabilidade mecânica;
- Promover um ponto de iluminação difuso na estrutura de suporte da câmera, para que no método de realimentação por visão computacional, o sistema não tenha interferências indesejáveis na detecção;
- Adotar estratégias para melhorar a leitura da posição da esfera utilizando a mesa resistiva, seja por meio do projeto de um filtro digital mais robusto, ou ainda pela substituição da tela resistiva de 4 fios por um modelo de 5 fios ou com tecnologia capacitiva, que oferecem maior precisão, estabilidade e menor sensibilidade a ruídos;
- Melhorar a calibração do sistema de coordenadas no método de detecção por visão computacional, implementando a detecção automática de marcadores

coloridos nas bordas da área calibrada. Com isso, a calibração poderá ser realizada em tempo real, de forma automática e mais robusta, reduzindo a necessidade de ajustes manuais e aumentando a confiabilidade do sistema frente a pequenas variações no posicionamento da câmera.

6.4 Propostas de Continuidade

Algumas propostas de continuidade para o trabalho são listadas abaixo, com o objetivo de aprimorar o desempenho do sistema e testar novas aplicações.

- **Implementação de realização de trajetórias:** Desenvolver o sistema para realização de trajetórias circulares e retangulares;
- **Implementação de controle adaptativo:** Substituir o controlador PID tradicional (com ganhos estáticos) por um controlador adaptativo capaz de ajustar os parâmetros automaticamente conforme a dinâmica da esfera ou mudanças no sistema (troca da esfera, variação da massa ou atrito);
- **Integração de controle preditivo baseado em modelo (MPC):** Utilizar técnicas de controle preditivo, como MPC (*Model Predictive Control*), para antecipar a trajetória da esfera e melhorar a resposta do sistema em termos de suavidade e precisão;
- **Utilização de redes neurais para identificação da posição da esfera:** Utilizar modelos de aprendizado de máquina para detecção da posição da esfera e comparar com o modelo desenvolvido neste trabalho, que utiliza métodos tradicionais de detecção. Uma alternativa promissora é o YOLO (*You Only Look Once*), baseado em redes neurais convolucionais, que permite detecção em tempo real com alta precisão, mesmo em cenários com variações de iluminação ou interferências visuais;
- **Desenvolvimento de um sistema supervisor:** Desenvolver um sistema supervisor capaz de mostrar a posição da esfera em tempo real, e possibilitar ajustes de parâmetros.

REFERÊNCIAS

- ALI, Momina L.; ZHANG, Zhou. The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. **Computers**, 2024.
- ARAUJO, Douglas H. C. D. et al. **Modelagem e implementação de um sistema Ball and Plate**. Revista Ciência e Tecnologia. Campinas – São José: [s.n.]. 2016. p. 85–91.
- ÅSTRÖM, Karl J.; MURRAY, Richard. **Feedback systems: an introduction for scientists and engineers**.
- BHALLA, Mudit R.; BHALLA, Anand V. Comparative Study of Various Touchscreen Technologies. **International Journal of Computer Applications**, Nova York, v. 6, n. 8, p. 12–18, Setembro 2010. ISSN 0975 – 8887.
- BOLÍVAR-VINCENTY, Carlos G.; BEAUCHAMP-BÁEZ, Gerson. **Modelling the Ball-and-Beam System from Newtonian Mechanics and from Lagrange Methods**. Twelfth Latin American and Caribbean Conference for Engineering and Technology. Guayaquil, Equador: [s.n.]. 2014.
- BUSIN, Laurent; VANDENBROUCKE, Nicolas; MACAIRE, Ludovic. Color spaces and image segmentation. **Advances in imaging and electron physics**, v. 151, n. 1, p. 1–162, 2008.
- CAMARGO, Fausto; DAROS, Thuinie. **A sala de aula inovadora-estratégias pedagógicas para fomentar o aprendizado ativo**.
- DE QUEIROZ, José E. R.; GOMES, Herman M. Introdução ao Processamento Digital de Imagens. **Rita**, v. VIII, n. 1, 2001. Disponível em: <https://saulo.arisa.com.br/aulas/cg/PDI_RitaTutorial.pdf>.
- DE REZENDE FREITAS, Elias J.; FORTES, Laís S. Aprendizagem ativa aplicada à Engenharia: um estudo sobre a percepção do aprendizado. **Revista de Ensino de Engenharia**, v. 39, 2020.
- DHARAMPAL, Vikram M. Methods of image edge detection: A review. **Journal of Electrical and Electronic Systems**, 4, n. 2, 2015.
- DIGI-KEY. Comparing Microcontrollers: What Brain Should I Go With? **Digi-Key**, 2023. Disponível em: <<https://www.digikey.com/en/maker/projects/comparing-microcontrollers-what-brain-should-i-go-with/02d2dcb1a0d441f5a11fc9956559b226>>. Acesso em: 4 junho 2025.
- DONGGUAN CITY DSSERVO TECHNOLOGY CO., LTD. DS3225(Red) – Product datasheet. **Dsservo**, 2023. Disponível em: <<https://www.dsservo.com/>>. Acesso em: 01 junho 2025.
- DORF, R. C.; BISHOP, R. H. **Sistemas de Controle Moderno**. 8ª. ed.
- DOWNS, Rick. Using resistive touch screens for human/machine interface. **Analog Applications Journal**, Dallas, v. 3Q 2005, p. 10–13, Julho–Setembro 2005.
- EELECTRONICS. ESP32 vs STM32 vs ESP8266 vs RP2040: A Comparative Overview. **Medium**, 2024. Disponível em: <<https://eelectronics.medium.com/esp32-vs-stm32-vs-esp8266-vs-rp2040-a-comparative-overview-81c22c81d04d>>. Acesso em: 04 junho 2025.
- ELECTRONICWINGS. 4 Wire Resistive Touch Screen Interfacing with ESP32. **ElectronicWings**, 2023. Disponível em: <<https://www.electronicwings.com/esp32/4-wire-resistive-touch-screen-interfacing-with-esp32>>. Acesso em: 02 jun. 2025.
- ESFANDIARI, Ramin S.; LU, Bei. **Modeling and analysis of dynamic systems**.

ESPRESSIF SYSTEMS. ESP32-WROOM-32 Datasheet, 2020. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>. Acesso em: 24 junho 2025.

ESRI. Função Conversão do Modelo de Cores. **ArcGIS Online – Documentação**. Disponível em: <<https://doc.arcgis.com/pt-br/arcgis-online/analyze/color-model-conversion-function.htm>>.

FORSYTH, David A.; PONCE, Jean. **Computer vision: a modern approach**.

FRANKLIN, Gene F.; POWELL, J. D.; EMAMI-NAEINI, Abbas. **Feedback Control of Dynamic Systems**. 6^a. ed.

GANESAN, P.; SAJIV, G. **A comprehensive study of edge detection for image processing applications**. 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). Chennai: IEEE. 2017. p. 1–7.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital Image Processing**. 4^a. ed.

HAM, Chan; TAUFIQ, Mohsin M. **Development of a Ball and Plate System**. ASEE Annual Conference and Exposition. Seattle: [s.n.]. 2015. p. 1-13.

HASSAN, Nazirah; MING, Kong W.; WAH, Choo K. **A comparative study on hsv-based and deep learning-based object detection algorithms for pedestrian traffic light signal recognition**. 2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS). [S.l.]: IEEE. 2020. p. 71-76.

HEMA, D.; KANNAN, S. Interactive color image segmentation using HSV color space. **Science and Technology Journal**, v. 7, n. 1, p. 37-41, 2019.

KNUPLEZ, Andrej; CHOWDHURY, Amor; SVEČKO, Rajko. **Modeling and control design for the ball and plate system**. IEEE International Conference on Industrial Technology (ICIT). Maribor: [s.n.]. 2003. p. 1064–1067.

KOKOTSAKI, Dimitra; MENZIES, Victoria; WIGGINS, Andy. Project-based learning: A review of the literature. **Improving Schools**, v. 19, p. 267-277, 2016.

MADHUMITHA, G. E. A. Design and development of a ball-plate balancing system with a smart phone human-machine interface. **Journal of Physics: Conference Series**, v. 1969, p. 012057, 2021.

MARENGONI, Maurício; STRINGHINI, Denise. Tutorial: Introdução à Visão Computacional usando OpenCV. **Revista de Informática Teórica e Aplicada**, v. 16, n. 1, 2009.

MASSON, Terezinha J. E. A. **Metodologia de ensino: aprendizagem baseada em projetos (PBL)**. Anais do XL Congresso Brasileiro de Educação em Engenharia (COBENGE). Belém: [s.n.]. 2012.

MUNDO DA ELÉTRICA. O que é servo motor e como funciona? **Mundo da Elétrica**, 2023. Disponível em: <<https://www.mundodaeletrica.com.br/o-que-e-servo-motor-e-como-funciona/>>. Acesso em: 31 maio 2025.

NETO, Euripedes P. F. **Visão computacional para identificação de cores em tempo real com opencv e python**. [S.l.]. 2020.

NISE, Norman S. **Engenharia de Sistemas de Controle**. 6^a. ed.

NXP SEMICONDUCTORS. AN10675 - Touch screen control using TSC. **NXP Semiconductors**, 2008. Disponível em: <<https://www.nxp.com/docs/en/application-note/AN10675.pdf>>. Acesso em: 02 jun. 2025.

O' MAHONY, Niall et al. Deep Learning vs. Traditional Computer Vision. **arXiv preprint arXiv:1910.13796**, outubro 2019. Disponível em: <<https://arxiv.org/pdf/1910.13796/1000>>. Acesso em: 10 junho 2025.

ODY, Lucas L. **Projeto e desenvolvimento de uma plataforma ball and plate para o estudo de sistemas de controle**. [S.l.]. 2020.

OGATA, Katsuhiko. **Engenharia de controle moderno**. 5ª. ed.

OPENCV. OpenCV: Open Source Computer Vision Library. **OpenCV**, 2025. Disponível em: <<https://docs.opencv.org/4.x/index.html>>. Acesso em: 11 junho 2025.

PATSKO, Luís F. Tutorial Montagem da ponte H. **Maxwell Bohr-Instrumentação Eletrônica**, 2006.

RANDOM NERD TUTORIALS. Introdução à placa de desenvolvimento ESP32. **Random Nerd Tutorials**, 2023. Disponível em: <<https://randomnerdtutorials.com/getting-started-with-esp32/#esp32-devkit-doit>>. Acesso em: 24 junho 2025.

RANE, Nitin. YOLO and Faster R-CNN object detection for smart Industry 4.0 and Industry 5.0: applications, challenges, and opportunities. **SSRN**, [S.l.], 2023. Acesso em: 14 maio 2025.

RIOS, Artur D. A. et al. Comparativo entre plataformas de desenvolvimento de sistemas microcontrolados para aplicações em Internet das Coisas, 2020.

ROCHA, Nathan P. **Desenvolvimento de uma Planta Didática Baseada em Cubesat**. [S.l.]. 2024.

SZELISKI, Richard. **Computer Vision: Algorithms and Applications**.

TAYLOR, John R. **Mecânica clássica**.

THE ENGINEERING MINDSET. What is a Servo Motor and What Does It Do? **The Engineering Mindset**, 2022. Disponível em: <<https://theengineeringmindset.com/what-is-a-servo-motor-and-what-does-it-do/>>. Acesso em: 29 maio 2025.

TORRES, Geraldo L., 2023.

TUDIĆ, Marko; ŠIMIĆ, Zdenko; PERIĆ, Niko. Design and Implementation of a Ball-Plate Control System and Python Script for Educational Purposes. **Sensors**, 22, 2022. 1875. Disponível em: <<https://www.mdpi.com/1424-8220/22/5/1875>>. Acesso em: 14 junho 2025.

VISO. Convolution Operations in Deep Learning. **VISO.ai**, 2024. Disponível em: <<https://viso.ai/deep-learning/convolution-operations/>>. Acesso em: 09 junho 2025.

WAGHMARE, S. S.; GHATULE, M. G. **Computer Vision based Object Detection using OpenCV**. International Conference on Emerging Smart Computing and Informatics (ESCI). Pune: [s.n.]. 2022.

ZHAO, Xia E. A. A review of convolutional neural networks in computer vision. **Artificial Intelligence Review**, [S.l.], 57, 23 março 2024. 99. Acesso em: 14 maio 2025.

APÊNDICES

APÊNDICE A – LINK PARA OS VÍDEOS DE DEMONSTRAÇÃO DOS RESULTADOS EXPERIMENTAIS

https://drive.google.com/drive/folders/169bD_gwNfkt9nHY-2zh-FkV9nbDCSBX2

APÊNDICE B – LINK DO GITHUB COM OS CÓDIGOS DO PROJETO

https://github.com/Gabrielhxavier/ball_and_plate--TCC-Gabriel-Xavier-