



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

JONAS DA SILVA COSTA NETO

CONTROLE DE ALTITUDE E ATITUDE DE UM QUADROPTERO AUTÔNOMO

Recife
2025

JONAS DA SILVA COSTA NETO

CONTROLE DE ALTITUDE E ATITUDE DE UM QUADROPTERO AUTÔNOMO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em 2025.

Orientador(a): Prof. Dr. Rafael Cavalcanti Neto

Coorientador(a): Prof. Gustavo Azevedo Teixeira

Recife
2025

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

Costa Neto, Jonas da Silva.

Controle de altitude e atitude de um quadcoptero autônomo / Jonas da Silva
Costa Neto. - Recife, 2025.

90 p. : il.

Orientador(a): Rafael Cavalcanti Neto

Coorientador(a): Gustavo Azevedo Teixeira

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

Inclui referências.

1. Quadcoptero. 2. Controle PID. 3. Algoritmo Genético. 4. ESP32. 5.
VANT. I. Cavalcanti Neto, Rafael . (Orientação). II. Azevedo Teixeira, Gustavo .
(Coorientação). IV. Título.

620 CDD (22.ed.)

JONAS DA SILVA COSTA NETO

CONTROLE DE ALTITUDE E ATITUDE DE UM QUADROPTERO AUTÔNOMO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em 2025.

Aprovado em: 15/08/2025

BANCA EXAMINADORA

Prof. Dr. Rafael Cavalcanti Neto (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Aline Victória Cavalcanti Pereira
Universidade Federal de Pernambuco

Prof. Dr. Valdemar Moreira Cavalcante Junior
Universidade Federal de Pernambuco

Dedico esse trabalho a todos da minha família e amigos que se envolveram diretamente e indiretamente, e se hoje me perguntarem se ele voa finalmente posso responder que sim, ele voa.

AGRADECIMENTOS

Agradeço primeiramente a meus pais por todo suporte incondicional durante a faculdade e o ensino médio onde minha mãe durante três anos me levava de Escada para Recife também ao meu Pai em especial, pela idealização e realização de uma das bancadas desenvolvidas para este trabalho, sem o suporte dos dois nunca teria chegado até aqui. Ao meu tio Junior, minha gratidão por sua valiosa contribuição na construção de uma das bancadas de teste. Aos meus amigos, pelo apoio constante, e um agradecimento especial a Wellington Gabriel, cuja ajuda foi fundamental no desenvolvimento da parte prática deste projeto. Expresso também minha sincera gratidão ao meu orientador, Prof. Dr. Rafael Cavalcanti Neto, e ao meu coorientador, Prof. Gustavo Azevedo Teixeira, pela fundamental orientação e suporte técnico. Por fim, estendo meus agradecimentos a todos os membros do projeto de extensão Asa Branca por todo o suporte e companheirismo.

O treinamento de usuários consiste em parte do processo de educação, em base repetitiva, compreende ações e/ou estratégias para desenvolver determinadas habilidades ou habilidades específicas do usuário por desconhecer situações específicas de uso da biblioteca e seus recursos informacionais, que envolvem o conjunto de meios necessários para tal (DIAS; PIRES, 2004, p. 36).

RESUMO

O presente Trabalho de Conclusão de Curso foi motivado pelo interesse em construir um drone competitivo para representar o Projeto de Extensão Asa Branca, na RoboCup, buscando desenvolver um sistema de controle embarcado robusto e eficiente utilizando componentes de baixo custo. O trabalho propôs o desenvolvimento de um sistema embarcado autônomo para o controle de atitude (roll, pitch e yaw) e altitude de um quadcoptero utilizando um ESP32. Inicialmente, os controladores PID foram sintonizados em ambiente de simulação com um algoritmo genético para otimização dos parâmetros, visando desempenho em tempo de acomodação, sobressinal e erro estacionário. Embora os parâmetros otimizados em simulação não tenham se mostrado eficazes no ambiente físico devido a simplificações do modelo, os resultados experimentais em bancadas de teste demonstraram a capacidade do sistema real em atingir boa estabilidade, erro em regime permanente próximo a nulo e sobressinal mínimo para os eixos de controle de atitude e altitude, mesmo sob distúrbios, validando a eficácia dos controladores implementados no hardware.

Palavras-chave: quadcoptero; controle PID; algoritmo genético; ESP32; VANT.

ABSTRACT

The present Undergraduate Thesis was motivated by the interest in building a competitive drone to represent the "Asa Branca" Extension Project in RoboCup, seeking to develop a robust and efficient embedded control system using low-cost components. The work proposed the development of an autonomous embedded system for attitude (roll, pitch, and yaw) and altitude control of a quadcopter using an ESP32. Initially, PID controllers were tuned in a simulation environment with a genetic algorithm for parameter optimization, aiming for performance in settling time, overshoot, and steady-state error. Although the parameters optimized in simulation did not prove effective in the physical environment due to model simplifications, experimental results on test benches demonstrated the real system's ability to achieve good stability, near-zero steady-state error, and minimal overshoot for the attitude and altitude control axes, even under disturbances, validating the effectiveness of the hardware-implemented controllers.

Keywords: quadcopter; PID control; genetic algorithm; ESP32; UAV.

LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura de um quadcoptero.....	17
Figura 2 - Quadcoptero Subindo.	19
Figura 3 - Movimento de Pitch.....	19
Figura 4 - Movimento de Roll.	20
Figura 5 - Movimento de Yaw.....	20
Figura 6 - Sistema SISO x MIMO	22
Figura 7 - Estrutura simples de um PID.	23
Figura 8 - Filtro Passa Baixa Ideal.	26
Figura 9 - ESP 32 Utilizado.	32
Figura 10 - BNO Utilizado.	37
Figura 11 – MPL Utilizado.	40
Figura 12 - Sensor Ultrassônico.	42
Figura 13 - Motor em Corte.	44
Figura 14 - <i>Frame</i> do drone utilizado.	45
Figura 15 - Motor Utilizado.	45
Figura 16 - Hélice Utilizada.	46
Figura 17 - ESC Utilizado.	46
Figura 18 - Bateria Utilizada.....	47
Figura 19 - Bancada de Teste do Motor.....	48
Figura 20 - Curva Força do Motor x Linearização.	49
Figura 21 - Bancada de Altitude.	50
Figura 22 - Modelo 3D do suporte desenvolvido.	51
Figura 23 - Suporte impresso	51
Figura 24 - Drone durante teste na bancada de altitude.	52
Figura 25 - Bancada de teste de atitude.	53
Figura 26 - Modelo 3D desenvolvido.....	54
Figura 27 - Drone na bancada de atitude.....	54
Figura 28 - Malha Simplificada de controle.	59
Figura 29 - Modelo do quadcoptero simplificado.....	60

Figura 30 - Interface do cálculo da matriz de inercia.....	61
Figura 31 - Bancada de medição de torque rotacional planejada	63
Figura 32 - Simulação de Movimento Rotacional.....	64
Figura 33 - Simulação de Movimento Vertical.....	64
Figura 34 - Exportação da função de custo.....	65
Figura 35 - Cálculo da função de custo.....	67
Figura 36 - Execução do Algoritmo Genético.....	67
Figura 37 - Resposta em simulação de roll e pitch.....	68
Figura 38 - Resposta em simulação de yaw.....	69
Figura 39 - Resposta em simulação do eixo Z.....	70
Figura 40 - Teste de Roll.....	72
Figura 41 - Roll x Pitch.....	73
Figura 42 - Roll na bancada.....	73
Figura 43 - Teste Roll com distúrbios.....	74
Figura 44 - Roll x Pitch com distúrbios.....	75
Figura 45 - Teste de Pitch.....	76
Figura 46 - Pitch x Roll.....	76
Figura 47 - Pitch na bancada.....	77
Figura 48 - Teste Pitch com distúrbios.....	77
Figura 49 - Pitch x Roll com distúrbios.....	78
Figura 50 - Teste de Yaw.....	79
Figura 51 - Teste Yaw com distúrbios.....	80
Figura 52 - Teste de altitude.....	81

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog-to-Digital Converter</i>
BLDC	<i>Brushless Direct Current</i>
CPU	<i>Central Processing Unit</i>
ESC	<i>Electronic Speed Controller</i>
FIR	<i>Finite Impulse Response</i>
IIR	<i>Infinite Impulse Response</i>
IMU	<i>Inertial Measurement Unit</i>
ISA	<i>International Standard Atmosphere</i>
I2C	<i>Inter-Integrated Circuit</i>
ITSE	<i>Integral of Time-Weighted Squared Error</i>
LTI	<i>Linear Time-Invariant</i>
MDF	<i>Medium Density Fiberboard</i>
PID	Proporcional-Integral-Derivativo
PWM	<i>Pulse Width Modulation</i>
SCL	<i>Serial Clock Line</i>
SDA	<i>Serial Data Line</i>
VANT	Veículo Aéreo Não Tripulado
VTOL	<i>Vertical Take-Off and Landing</i>

LISTA DE SÍMBOLOS

f_x	Forças geradas pelos respectivos motores
m_x	Torques rotacionais gerados pelo efeito giroscópico
d	Comprimento de um "braço"
τ_z	Equação dinâmica da altitude
τ_ϕ	Equação dinâmica de <i>roll</i>
τ_θ	Equação dinâmica de <i>pitch</i>
τ_ψ	Equação dinâmica de <i>yaw</i>
ϕ	Ângulo de <i>roll</i>
θ	Ângulo de <i>pitch</i>
ψ	Ângulo de <i>yaw</i>
z	Altitude
J	Matriz de inércia do quadcoptero
J_{xx}	Momento de inércia em relação ao eixo x
J_{yy}	Momento de inércia em relação ao eixo y
J_z	Momento de inércia em relação ao eixo z
$J_{componente}$	Momento de inércia do componente
J_{CM}	Momento de inércia do componente em relação ao seu centro de massa
m	Massa do componente
r_c	Raio do corpo central (hub)
h_c	Altura do corpo central (hub)
r_b	Raio dos braços
l_b	Comprimento dos braços
m_m	Massa dos motores
h_m	Altura dos motores
a_e	Largura dos ESCs
b_e	Comprimento dos ESCs
m_e	Massa dos ESCs
d_e	Distância dos ESCs ao centro de massa do sistema
$motor_x$	Comando individual para cada motor (onde x = 1, 2, 3, 4)

PID_z	Componente de controle da altitude
PID_ϕ	Componente de <i>roll</i>
PID_θ	Componente de <i>pitch</i>
PID_ψ	Componente de <i>yaw</i>
$throttle$	Valor base de potência (PWM)
$u(t)$	Sinal de controle (PID)
$e(t)$	Erro (diferença entre referência e saída)
$r(t)$	Referência desejada
$y(t)$	Saída real do sistema
K_p	Ganho proporcional
$u_p(t)$	Ação proporcional
K_I	Ganho integral
$u_I(t)$	Ação integral
K_D	Ganho derivativo
$u_D(t)$	Ação derivativa
$H(e^{j\omega})$	Resposta em frequência de um filtro passa-baixa
ω	Frequência
ω_c	Frequência de corte
$x[n]$	Sinal de entrada (filtragem)
$y[n]$	Sinal filtrado (saída)
α	Coefficiente de suavização
N	Número de amostras (largura da janela do filtro de média móvel)
t	Tempo
C	Função de custo
Ω_H	Velocidade de rotação da hélices

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	14
1.2	METODOLOGIA	15
1.3	ORGANIZAÇÃO DO TRABALHO	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	QUADROPTERO VISÃO GERAL E CARACTERÍSTICAS	17
2.1.1	Princípios de Funcionamento	18
2.2	SISTEMAS DE CONTROLE	21
2.2.1	Sistemas Single Input Single Output (SISO) e Multiple Input Multiple Output (MIMO)	22
2.2.2	Controlador PID	23
2.2.3	Filtros Digitais	24
2.2.3.1	<i>Filtro Passa-Baixa</i>	25
2.2.3.2	<i>Filtro Média-Móvel</i>	27
2.3	ALGORITMO GENÉTICO	28
3	CONSTRUÇÃO DO DRONE E DAS BANCADAS DE TESTES	31
3.1	TRANSDUTORES E ATUADORES PARA APLICAÇÃO EM QUADROPTEROS	31
3.1.1	Unidade de controle	32
3.1.2	IMU Inertial Measurement Unit	33
3.1.2.1	<i>Acelerômetro</i>	34
3.1.2.2	<i>Giroscópio</i>	34
3.1.2.3	<i>Magnetômetro</i>	35
3.1.2.4	<i>BNO-055</i>	36
3.1.3	Altímetro	39
3.1.3.1	<i>MPL</i>	39

3.1.4	Ultrassônico.....	41
3.1.4.1	HC-S04.....	41
3.1.5	Electronic Speed Controller (ESC)	42
3.1.6	Motor Brushless.....	43
3.2	MONTAGEM DE QUADCOPTERO.....	44
3.2.1	Frame	44
3.2.2	Sistema de propulsão	45
3.2.3	Fonte de alimentação	47
3.3	BANCADAS	47
3.3.1	Bancada de modelagem do motor.....	48
3.3.2	Bancada para testes de controle de altitude.....	49
3.3.3	Bancada para testes de controle de atitude (roll, pitch e yaw)	52
4	MODELAGEM E CONTROLE DE QUADCÓPTERO BASEADO EM	
	ALGORITMO GENETICO.....	56
4.1	MODELAGEM DINÂMICA DO QUADCOPTERO.....	56
4.1.1	Estratégia de Controle Distribuído nos Motores	57
4.1.2	Representação do Quadcoptero em ambiente Simulink/Matlab	59
4.2	PROPOSTA DE METODOLOGIA DE PROJETO DE CONTROLADORES	
	BASEADA EM GA APLICADA EM UM QUADCOPTERO	64
4.2.1	Etapa 1 – Seleção dos modelos simulados	65
4.2.2	Etapa 2 – Definição da função de custo	65
4.2.3	Etapa 3 – Configuração e execução do algoritmo genético.....	67
4.3	VALIDAÇÃO DA METODOLOGIA PROPOSTA.....	68
4.3.1	Controle de Roll e Pitch	68
4.3.2	Controle de Yaw	68
4.3.3	Controle de Altitude (Eixo Z).....	69
4.3.4	Conclusão Preliminar.....	70
5	RESULTADOS EXPERIMENTAIS.....	71
5.1	RESULTADOS EXPERIMENTAIS	71

6	CONCLUSÃO E PROPOSTAS DE MELHORIAS	82
6.1	CONCLUSÃO	82
6.2	PROPOSTAS DE MELHORIA.....	82
7	REFERÊNCIAS.....	84

1 INTRODUÇÃO

O crescente interesse em robótica tem impulsionado avanços significativos nos Veículos Aéreos Não Tripulados (VANTs), também conhecidos como drones. Esses sistemas, notavelmente os quadcopteros, tornaram-se uma solução viável para diversas aplicações, desde o uso militar e civil até a pesquisa e recreação [1]. Além disso, avanços recentes em tecnologias de fabricação e eletrônica resultaram em uma significativa redução de custos e no aumento da capacidade de carga, tornando-os mais acessíveis e funcionais para um espectro ainda maior de usuários e aplicações [2].

Um dos principais desafios desses sistemas é garantir a estabilidade e o controle preciso de sua atitude (*Roll*, *Pitch* e *Yaw*) e posição (X, Y e Z) durante o voo. Em contextos onde não há intervenção humana direta (drones autônomos), a importância de sistemas embarcados robustos torna-se ainda mais evidente.

O desenvolvimento deste trabalho foi motivado inicialmente pelo interesse em construir um drone competitivo para representar o Projeto de Extensão Asa Branca, na RoboCup. Diante dessa motivação prática, buscou-se desenvolver um sistema de controle embarcado robusto e eficiente, capaz de manter a estabilidade de atitude e a altitude de um quadcoptero utilizando componentes de baixo custo.

1.1 Objetivos

O principal objetivo deste trabalho é projetar e implementar um sistema embarcado autônomo para o controle de atitude (roll, pitch e yaw) e altitude de um quadcoptero, com um foco particular em sua aplicação em competições de robótica, a RoboCup. A escolha da plataforma de baixo custo, o microcontrolador ESP32, visa não apenas a viabilidade do projeto, mas também sua replicação para fins de pesquisa e educação, promovendo a democratização da tecnologia.

Para alcançar este objetivo geral, foram definidos os seguintes objetivos específicos:

- **Modelagem Matemática do Quadcoptero:** Desenvolver uma modelagem matemática que represente a dinâmica do quadcoptero. Esta etapa é crucial

para a simulação e o ajuste de controladores em um ambiente virtual, o que economiza tempo e recursos antes da implementação física.

- **Integração de Sensores e Atuadores:** Integrar um conjunto de sensores e atuadores, incluindo a Unidade de Medida Inercial (IMU) BNO055 e o sensor de distância ultrassônico HC-SR04. O uso do sensor ultrassônico é especialmente importante, pois a competição ocorre em um ambiente fechado, onde a precisão de medição de altitude em baixas distâncias é fundamental para a navegação segura do drone.
- **Implementação dos Controladores:** Implementar um conjunto de controladores de atitude e altitude, sendo o controlador PID (Proporcional-Integral-Derivativo) a principal escolha devido à sua eficácia e simplicidade.
- **Validação do Sistema:** Validar o sistema de controle tanto em ambiente de simulação quanto em bancadas de teste específicas. A validação em bancada é fundamental, pois permite a análise individualizada do comportamento de cada eixo (roll, pitch e yaw), reduzindo os riscos associados aos voos livres.
- **Desenvolvimento da Interface Web Embarcada:** Criar uma interface web embarcada para o monitoramento de variáveis em tempo real e a configuração de parâmetros do sistema. Esta interface é essencial para a fase de testes e depuração, facilitando o ajuste fino do comportamento de voo do quadcoptero.
- **Avaliação de Desempenho:** Avaliar o desempenho do sistema de controle com base em critérios como estabilidade (capacidade de retornar ao ponto de equilíbrio após perturbações), erro estacionário (desvio entre o valor desejado e o de saída em regime permanente) e tempo de resposta (rapidez com que o sistema reage a uma mudança de referência).

1.2 Metodologia

A metodologia adotada neste trabalho combinou a base teórica com o ciclo de desenvolvimento prático, integrando modelagem, simulação, programação embarcada e testes em hardware. A modelagem do quadcoptero, fundamentada em artigos científicos, foi realizada no domínio do tempo, empregando equações

simplificadas para representar os eixos controlados, o que possibilitou a análise preliminar do comportamento dinâmico do sistema.

As simulações foram conduzidas no ambiente MATLAB/Simulink, onde controladores PID foram aplicados aos canais de roll, pitch, yaw e altitude. A escolha do controlador PID se deu por sua estrutura relativamente simples, eficácia comprovada em diversas aplicações e facilidade de implementação e ajuste. Para a otimização dos parâmetros desses controladores, foi utilizado um algoritmo genético (GA), uma técnica de otimização meta-heurística particularmente adequada para sistemas complexos e não lineares. O GA buscou a minimização do erro e a ausência de sobressinal, garantindo um comportamento de voo estável.

Após a fase de simulação, a validação do sistema foi realizada em bancadas de testes específicas. Essa abordagem permitiu a análise individualizada de cada eixo, avaliando a resposta de motores, ESCs e sensores em um ambiente controlado e seguro. Por fim, o firmware responsável por orquestrar a leitura de sensores, a aplicação dos controladores e o acionamento dos atuadores foi desenvolvido em C++ na IDE do Arduino para o microcontrolador ESP32.

1.3 Organização do trabalho

O presente trabalho de conclusão de curso está organizado da seguinte forma:

- No Capítulo 2, apresenta-se a fundamentação teórica necessária para a compreensão dos conceitos e notações utilizados ao longo do trabalho;
- No Capítulo 3 é apresentado os itens utilizados na montagem do drone e as bancadas construídas para teste e validação do controle.
- No Capítulo 4, são apresentados o desenvolvimento do sistema de controle via algoritmo genético e sua validação, por meio de simulações;
- No Capítulo 5, é discutido os testes e resultados experimentais.
- No Capítulo 6, é dito as conclusões finais do projeto e propostas de possíveis melhorias para trabalhos futuros.

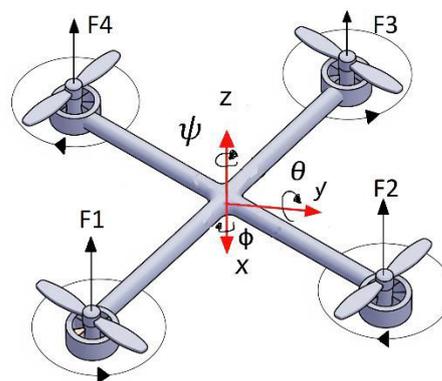
2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo explora os fundamentos teóricos e tecnológicos essenciais para o desenvolvimento do quadcoptero autônomo. São abordados os controladores PID e os filtros digitais (*Finite impulse response* FIR e *Infinite Impulse Response* IIR), os princípios de funcionamento de um quadcoptero (como funciona os seus graus de liberdade), uma descrição dos tipos de sistemas de controle e algoritmo genético.

2.1 Quadcoptero Visão Geral e Características

O quadcoptero, também conhecido como *quadcopter*, é uma aeronave de decolagem e pouso vertical (VTOL – *Vertical Take-Off and Landing*) que pertence à classe dos veículos aéreos não tripulados (VANTs ou UAVs). Sua arquitetura é composta por quatro rotores dispostos em uma configuração simétrica no plano horizontal, geralmente em forma de cruz (“+”) ou em “X”. Dois dos rotores giram no sentido horário e os outros dois no sentido anti-horário, o que permite a estabilização da aeronave sem a necessidade de um rotor de cauda, como ocorre em helicópteros convencionais.

Figura 1 - Estrutura de um quadcoptero.



Fonte: O Autor (2025).

Apesar dessas vantagens, os quadcopteros são dinamicamente instáveis. Isso significa que, sem um sistema de controle ativo, não conseguem manter o equilíbrio por conta própria. Dessa forma, são necessários algoritmos de controle em tempo real, como PID, *Linear-Quadratic Regulator* (LQR) ou técnicas não lineares, para estabilizar a atitude e controlar a posição da aeronave durante o voo.

A estrutura relativamente simples de um quadcoptero permite que seu comportamento seja descrito a partir da aplicação das leis de Newton-Euler. Essa modelagem possibilita tanto a simulação do sistema quanto o desenvolvimento de controladores robustos que considerem os acoplamentos entre os eixos, as não linearidades e as perturbações externas.

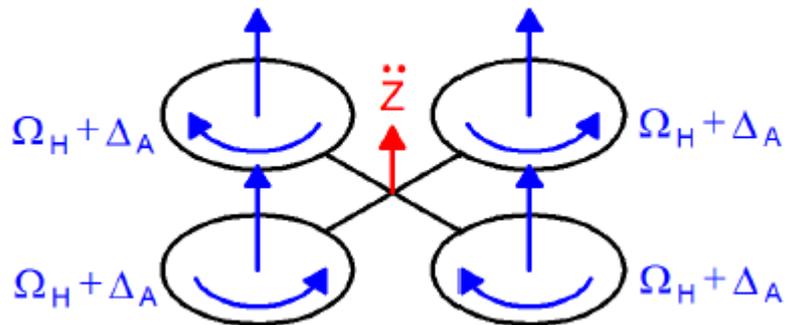
2.1.1 Princípios de Funcionamento

O funcionamento de um quadcoptero baseia-se no controle da velocidade angular dos quatro rotores, que atuam como os únicos elementos responsáveis pela geração de forças de sustentação e torques de controle. Ao contrário de aeronaves convencionais, o quadcoptero não possui superfícies móveis de controle, como lemes ou ailerons. Toda a dinâmica de voo, incluindo subida, descida, rotação e deslocamentos, é obtida exclusivamente pela variação da rotação dos motores *brushless*, controlados individualmente por seus respectivos ESCs.

Para entender como essa variação de velocidades influencia o comportamento da aeronave, é necessário considerar os quatro movimentos fundamentais que um quadcoptero pode realizar:

- **Subida e descida (controle de altitude):** ocorre quando os quatro motores giram com a mesma velocidade angular. Se a força total gerada pelos rotores excede o peso do sistema, o drone sobe; se for menor, ele desce [3].

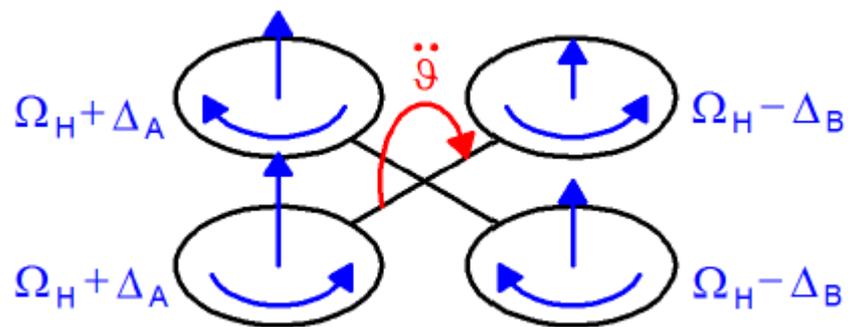
Figura 2 - Quadcoptero Subindo.



Fonte: Figura alterada do [3].

- **Pitch (inclinação frontal/traseira):** é obtido ao aumentar a velocidade dos motores traseiros e diminuir a dos motores frontais (ou vice-versa). Isso cria um torque em torno do eixo y, inclinando o quadcoptero para frente ou para trás [3].

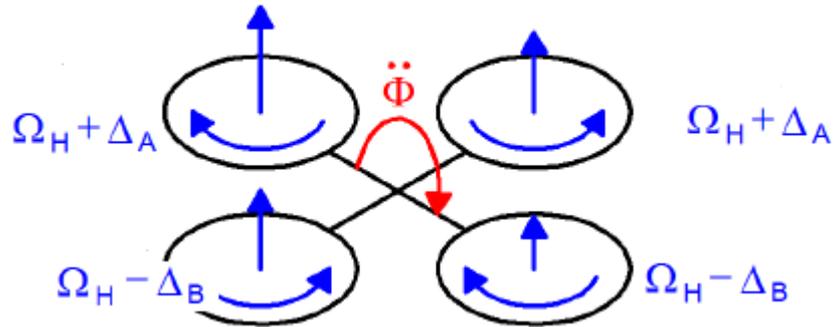
Figura 3 - Movimento de Pitch.



Fonte: Figura alterada do [3].

- **Roll (inclinação lateral):** resulta da variação de velocidade entre os motores da esquerda e da direita, gerando torque em torno do eixo x, o que inclina a aeronave lateralmente [3].

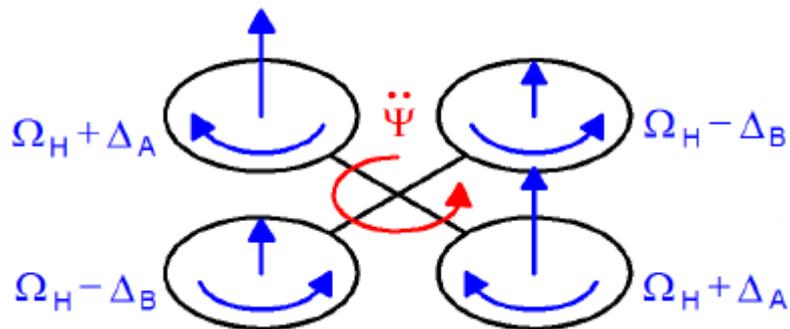
Figura 4 - Movimento de Roll.



Fonte: Figura alterada do [3].

- **Yaw (rotação em torno do eixo vertical):** é controlado explorando os torques de reação dos motores. Ao aumentar a rotação dos dois motores que giram no sentido horário e diminuir a dos que giram no sentido anti-horário, ou vice-versa, cria-se um torque resultante em torno do eixo z, fazendo a aeronave girar sobre si mesma [3].

Figura 5 - Movimento de Yaw.



Fonte: Figura alterada do [3].

Em conjunto, esses movimentos possibilitam ao quadcoptero deslocar-se no espaço tridimensional. Por exemplo, ao inclinar-se para frente (*pitch* negativo) e aplicar maior propulsão total, a componente da força de sustentação se inclina em relação ao solo, gerando um vetor resultante com componente horizontal, o que move o drone na direção desejada. Este princípio de inclinação direcional é fundamental para que o quadcoptero possa realizar tarefas de navegação e rastreamento de

trajetórias. Para realizar esses movimentos e fazer as leituras das variáveis de estado (*roll*, *pitch*, *yaw* e altitude) são utilizados os transdutores e atuadores presentes na seção 3.1.

2.2 Sistemas de Controle

Sistemas de controle são essenciais em quase toda tecnologia moderna, permitindo que máquinas e processos operem de forma autônoma, eficiente e segura. Em sua essência, um sistema de controle busca gerenciar o comportamento de um processo ou dispositivo, garantindo que ele atinja e mantenha um estado desejado, mesmo diante de perturbações ou mudanças nas condições. Desde a temperatura em um forno industrial até a trajetória de um foguete espacial, a capacidade de controlar variáveis críticas é fundamental para o sucesso e a estabilidade de inúmeras aplicações.

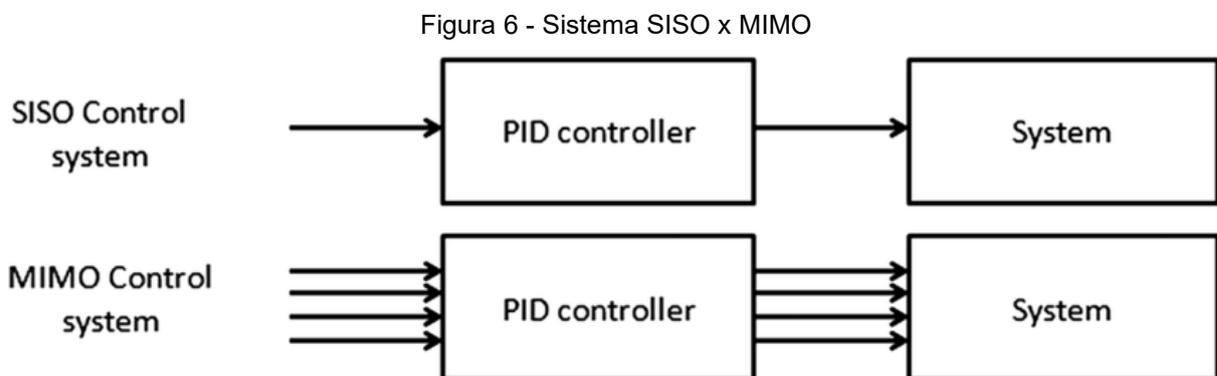
A importância dos sistemas de controle pode ser observada em diversos setores. Na indústria, eles são empregados para automatizar linhas de produção, controlar a qualidade de produtos, e otimizar o consumo de energia, resultando em maior produtividade e redução de custos. Em veículos automotores, sistemas de controle avançados gerenciam desde a injeção de combustível e a transmissão até a frenagem ABS e o controle de estabilidade, aumentando a segurança e o conforto. Na medicina, bombas de infusão controlam a dosagem de medicamentos com precisão, enquanto robôs cirúrgicos oferecem movimentos finos e estáveis. Até mesmo em nosso cotidiano, sistemas de controle estão presentes em dispositivos como termostatos residenciais, elevadores e máquinas de lavar, que ajustam seu funcionamento para atingir metas predefinidas.

Esta seção aprofundará os conceitos fundamentais dos sistemas de controle, explorando tipos específicos de controladores e técnicas de filtragem digital que são essenciais para o desenvolvimento de um quadcoptero autônomo.

2.2.1 Sistemas Single Input Single Output (SISO) e Multiple Input Multiple Output (MIMO)

Os sistemas de controle são classificados, entre outros critérios, de acordo com o número de entradas (sinais de controle) e saídas (variáveis controladas). Quando o sistema possui uma única entrada e uma única saída, ele é chamado de SISO. Esses sistemas são amplamente abordados na teoria clássica de controle devido à sua simplicidade e aplicabilidade em diversos processos industriais [4].

Já os sistemas com múltiplas entradas e múltiplas saídas são denominados MIMO. Esses sistemas são típicos de processos mais complexos, onde diferentes sinais de controle influenciam simultaneamente várias variáveis de saída, e vice-versa [4], uma visualização simples desse tipo de sistema é visto na Figura 6. Um exemplo clássico de sistema MIMO é o controle de atitude e altitude de um quadcoptero, como o que será abordado neste trabalho. Nesse tipo de sistema, os sinais de controle aplicados aos quatro motores afetam simultaneamente os ângulos de *roll*, *pitch*, *yaw* e a altitude da aeronave.



Fonte: Retirado do [5].

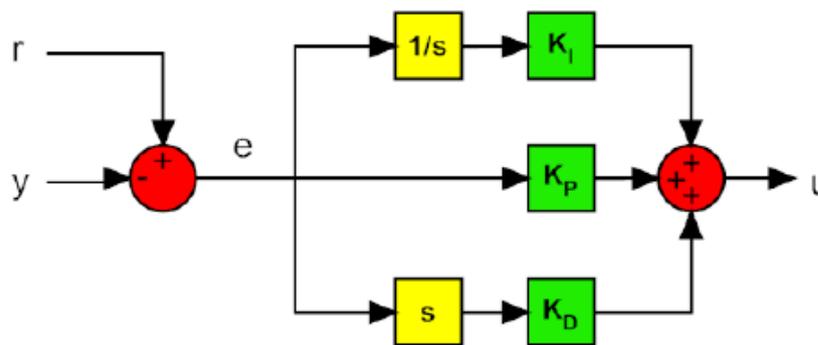
Apesar de sua natureza MIMO, será mostrado ao longo deste trabalho que é possível decompor o sistema do quadcoptero em vários subsistemas SISO, facilitando o projeto de controladores independentes para cada variável. Essa abordagem modular permite aplicar técnicas clássicas de controle, como o PID, de maneira mais simples e eficiente, mesmo em sistemas, originalmente, com multivariáveis.

2.2.2 Controlador PID

O controlador Proporcional-Integral-Derivativo (PID) é um dos algoritmos de controle mais amplamente utilizados na engenharia, especialmente em sistemas de malha fechada. Sua popularidade advém da estrutura relativamente simples, do bom desempenho em diversas aplicações e da facilidade de implementação e ajuste [4].

A estrutura clássica de um controlador PID baseia-se no cálculo do sinal de controle $u(t)$ a partir do erro $e(t)$, que representa a diferença entre a referência desejada $r(t)$ e a saída real do sistema $y(t)$. Esse conceito é representado esquematicamente na Figura 7.

Figura 7 - Estrutura simples de um PID.



Fonte: Retirado do [3].

O sinal de controle é composto por três ações fundamentais:

Ação Proporcional (P): Responsável pela resposta imediata ao erro atual. Essa ação aplica um ganho proporcional K_p ao erro $e(t)$ o que pode ser expresso por:

$$u_p(t) = K_p * e(t). \quad (2.4)$$

Aumentos em K_p tendem a reduzir o tempo de resposta, mas podem introduzir oscilações ou instabilidade se o valor for excessivo.

Ação Integral (I): Atua sobre a acumulação do erro ao longo do tempo, corrigindo desvios residuais e eliminando o erro em regime permanente. É expressa como:

$$u_I(t) = K_I \int_0^t e(t) dt, \quad (2.5)$$

Embora seja eficaz para eliminar erros estacionários, o excesso de ação integral pode causar aumento do tempo de acomodação e do sobressinal.

Ação Derivativa (D): Age com base na taxa de variação do erro, antecipando tendências e contribuindo para a estabilização da resposta. Sua expressão é dada por:

$$u_D(t) = K_D \frac{de(t)}{dt}, \quad (2.6)$$

essa ação é útil para amortecer oscilações e melhorar o tempo de estabilização, sendo especialmente vantajosa em sistemas com resposta dinâmica rápida.

A combinação dessas três ações resulta na equação geral do controlador PID:

$$u(t) = K_p * e(t) + K_I \int_0^t e(t) dt + K_D \frac{de(t)}{dt}. \quad (2.7)$$

Essa estrutura permite flexibilidade no ajuste do comportamento dinâmico do sistema, influenciando diretamente parâmetros como o tempo de subida, o sobressinal, o tempo de acomodação e o erro em regime estacionário. Conforme destaca Ogata, o controlador PID é eficaz mesmo na presença de incertezas paramétricas ou modelos simplificados, sendo amplamente aplicável a uma grande variedade de processos [4].

2.2.3 Filtros Digitais

Filtros digitais são sistemas que atuam sobre sinais amostrados no tempo, modificando suas características espectrais ou temporais de forma a atender critérios específicos de projeto, como a atenuação de ruídos, a extração de informações ou o condicionamento de sinais para controle. Esses filtros desempenham um papel fundamental em aplicações embarcadas, especialmente em sistemas de controle em

tempo real, como o controle de atitude e altitude de drones, onde a presença de ruídos nos sensores pode comprometer significativamente a estabilidade do sistema.

Do ponto de vista teórico, um filtro digital pode ser entendido como um sistema linear e invariável no tempo (*Linear Time Invariant* - LTI), cuja operação é descrita por equações de diferenças. Os filtros são classificados principalmente em duas categorias: os de resposta finita ao impulso (FIR – *Finite Impulse Response*) e os de resposta infinita ao impulso (IIR – *Infinite Impulse Response*). Os filtros FIR possuem uma resposta limitada no tempo e não envolvem realimentação, sendo mais simples de analisar e intrinsecamente estáveis. Já os filtros IIR, por utilizarem realimentação, são capazes de obter desempenho semelhante com menor ordem, mas requerem maior cuidado na análise de estabilidade e na implementação numérica [6].

A construção de filtros digitais eficazes requer a escolha apropriada de coeficientes que satisfaçam as especificações do projeto, como faixa de passagem, faixa de rejeição, ondulações e atenuações. O projeto pode ser realizado tanto no domínio do tempo quanto no domínio da frequência, sendo este último particularmente útil para analisar o efeito dos filtros em sinais com conteúdo harmônico ou ruidoso [6].

No contexto deste trabalho, foram implementadas duas abordagens de filtragem digital: um filtro passa-baixa e um filtro de média móvel, ambos aplicados à suavização dos dados obtidos dos sensores embarcados. Essas técnicas são descritas em mais detalhes nas subseções a seguir.

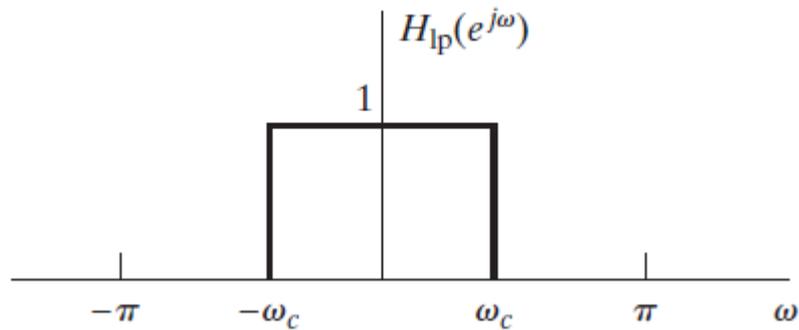
2.2.3.1 Filtro Passa-Baixa

Filtros passa-baixa digitais constituem uma das classes mais importantes de filtros em tempo discreto, sendo amplamente utilizados para eliminar componentes de alta frequência de sinais amostrados. Seu objetivo principal é permitir a passagem de frequências mais baixas, atenuando ruídos e variações rápidas que não são de interesse, o que os torna fundamentais para aplicações em controle e processamento de sinais embarcados.

Do ponto de vista da análise em tempo discreto, um filtro passa-baixa ideal possui uma resposta em frequência $H(e^{j\omega})$ igual a 1 dentro de uma faixa de passagem (tipicamente em torno de $\omega=0$) e nula nas frequências superiores, conforme ilustrado

na Figura 8 [6]. Entretanto, filtros ideais não são realizáveis fisicamente, e por isso utiliza-se aproximações práticas, como os filtros IIR de primeira ordem.

Figura 8 - Filtro Passa Baixa Ideal.



Fonte: Retirado do [6].

Uma implementação comum e eficiente de filtro passa-baixa em sistemas embarcados é o filtro recursivo de primeira ordem, que pertence à classe dos filtros IIR. Ele pode ser descrito pela equação de diferenças:

$$y[n] = \alpha * x[n] + (1 - \alpha) * y[n - 1], \quad (2.8)$$

onde:

- $x[n]$ é o sinal de entrada,
- $y[n]$ é o sinal filtrado (saída),
- α é o coeficiente de suavização, com $0 < \alpha < 1$.

Essa estrutura simples oferece boa capacidade de suavização com baixa complexidade computacional, tornando-se adequada para sistemas com recursos limitados, como o ESP32. O valor de α determina o grau de atenuação das componentes de alta frequência: quanto menor o valor de α , maior a suavização e menor a resposta dinâmica do filtro.

Conforme discutido por [6], o comportamento de tais sistemas LTI pode ser analisado tanto no domínio do tempo quanto no da frequência, sendo sua resposta diretamente relacionada à posição de polos e zeros no plano-z. No caso do filtro recursivo de primeira ordem, o sistema possui um único polo real e estável dentro do círculo unitário, assegurando estabilidade e resposta suavizada.

2.2.3.2 Filtro Média-Móvel

O filtro de média móvel é uma técnica de suavização simples e eficiente, classificada como um filtro FIR. Sua função consiste em calcular a média aritmética de um número fixo de amostras passadas do sinal de entrada, reduzindo a presença de ruídos de alta frequência e variações rápidas. Por sua baixa complexidade computacional e facilidade de implementação, é amplamente utilizado em sistemas embarcados de tempo real.

Sua equação na forma discreta é dada por:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n - k] \quad (2.9)$$

onde:

- $x[n]$ é o sinal de entrada,
- $y[n]$ é o sinal filtrado (saída),
- N é o número de amostras consideradas (largura da janela).

Esse filtro possui resposta finita ao impulso e não utiliza realimentação, o que o torna intrinsecamente estável. Além disso, por manter uma resposta de fase linear, ele evita distorções temporais no sinal processado, o que é desejável em aplicações de controle e medição sensível [6].

No contexto deste trabalho, o filtro de média móvel foi implementado especificamente para suavizar os dados provenientes do sensor ultrassônico HC-SR04, que apresentou flutuações perceptíveis nas leituras de distância, principalmente em razão das vibrações estruturais do drone e de reflexões ambientais, além desses fatores, destaca-se também a limitação intrínseca do conversor analógico-digital (ADC - *Analog-to-Digital Converter*) do microcontrolador ESP32, cuja resolução efetiva e linearidade são inferiores às de conversores dedicados, o que pode introduzir ruídos adicionais nas medições. A aplicação do filtro permitiu obter uma estimativa mais estável da altitude, contribuindo diretamente para a melhoria do desempenho do sistema de controle. Para mais detalhes da implementação visualizar em [7].

2.3 Algoritmo Genético

Algoritmos Genéticos (GA - *Genetic Algorithm*) são algoritmos de busca inspirados nos mecanismos da seleção natural e da genética biológica. Eles operam com base no princípio da “*sobrevivência do mais apto*” entre estruturas de dados, frequentemente representadas como *strings* ou cromossomos, combinando essa lógica com uma troca de informações estruturada e aleatória. Essa abordagem permite que os GAs explorem eficientemente informações históricas para identificar novos pontos de busca com desempenho aprimorado [8].

Desenvolvidos por John Holland e seus colaboradores na Universidade de *Michigan* [8], os GA foram concebidos tanto para explicar os processos adaptativos naturais quanto para projetar sistemas artificiais que preservem esses mecanismos essenciais.

A robustez é um tema central na pesquisa sobre GAs, representando o equilíbrio entre eficiência e eficácia necessário para a sobrevivência em diversos ambientes. Esses algoritmos se destacam por sua simplicidade computacional aliada a uma poderosa capacidade de busca por melhorias. Além disso, não são fundamentalmente limitados por suposições restritivas sobre o espaço de busca, como continuidade, existência de derivadas ou unimodalidade. Isso os torna especialmente adequados para problemas complexos e ruidosos, com múltiplos ótimos locais e descontinuidades, onde métodos tradicionais frequentemente falham [17].

Os algoritmos genéticos diferem das técnicas tradicionais de otimização em quatro aspectos fundamentais:

1. **Trabalham com uma codificação do conjunto de parâmetros:** em vez de manipular diretamente os parâmetros do problema, os GAs operam sobre representações codificadas desses parâmetros, geralmente como *strings* de comprimento finito.
2. **Buscam a partir de uma população de pontos:** diferentemente da abordagem sequencial tradicional, os GAs mantêm uma população de *strings* que evolui em paralelo, o que reduz a probabilidade de convergir prematuramente para ótimos locais.

3. **Utilizam apenas os valores da função objetivo (*payoff*):** não exigem derivadas ou informações auxiliares; apenas os valores da função objetivo associados a cada *string* individual são suficientes para orientar a busca.
4. **Empregam regras de transição probabilísticas:** ao contrário de métodos determinísticos, os GAs utilizam operadores probabilísticos que direcionam a busca para regiões promissoras do espaço de soluções, sem depender exclusivamente do acaso.

Um algoritmo genético simples opera iterativamente por meio de três operadores principais, aplicados em cada geração:

- **Reprodução:** *Strings* são copiadas para a próxima geração com uma probabilidade proporcional aos seus valores de *fitness* (função objetivo). *Strings* mais aptas têm maior chance de gerar descendentes, simulando a seleção natural.
- **Crossover (cruzamento):** pares de *strings* selecionados para reprodução são recombinados. Um ponto de corte é escolhido aleatoriamente ao longo do comprimento da *string*, e segmentos dos pais são trocados, produzindo descendentes que mesclam características bem-sucedidas.
- **Mutação:** consiste em alterações aleatórias e ocasionais (de baixa probabilidade) em posições específicas da *string*. A mutação age como uma “política de seguro” contra a perda prematura de material genético útil, promovendo diversidade e permitindo a exploração de regiões do espaço de busca que talvez não seriam alcançadas apenas por reprodução e cruzamento.

No presente trabalho, essa abordagem foi utilizada com o objetivo de ajustar os parâmetros do controlador responsável pela estabilidade e navegação de um quadcoptero, considerando as exigências impostas por sua dinâmica não linear e pelas incertezas do ambiente, ainda que os algoritmos genéticos ofereçam vantagens como robustez e boa capacidade de exploração do espaço de busca, outros métodos de otimização também têm sido explorados na literatura para essa mesma finalidade. É o caso do algoritmo de Otimização por Enxame de Partículas com múltiplos objetivos (MOPSO - *Multi-Objective Particle Swarm Optimization*), que permite a obtenção de soluções de compromisso entre diferentes critérios de desempenho, conforme apresentado por [1], por outro lado, em [9] utilizaram algoritmos genéticos

para a sintonia de PIDs com enfoque em estabilidade e rejeição a distúrbios. Essas abordagens refletem a diversidade de técnicas disponíveis para o projeto de controladores em veículos autônomos, sendo a escolha do método mais apropriado dependente das características específicas do sistema e dos objetivos de controle.

3 CONSTRUÇÃO DO DRONE E DAS BANCADAS DE TESTES

3.1 Transdutores e Atuadores para Aplicação em Quadcopteros

Transdutores são dispositivos capazes de converter uma forma de energia física em outra, geralmente transformando grandezas mecânicas, térmicas ou ópticas em sinais elétricos, ou vice-versa [10]. Quando utilizados para monitoramento de variáveis ambientais ou do próprio sistema, esses dispositivos são chamados comumente de sensores. Em sistemas embarcados aplicados a veículos autônomos, como drones, os transdutores são essenciais para a percepção do ambiente e para a retroalimentação do sistema de controle.

Neste trabalho, foram utilizados cinco transdutores principais, divididos conforme sua função no sistema: para a estimativa da atitude (*roll*, *pitch* e *yaw*), utilizou-se um módulo inercial com acelerômetro, giroscópio e magnetômetro (BNO055); para a altitude, empregaram-se dois sensores complementares: um sensor ultrassônico (HC-SR04), responsável por medir a distância ao solo em baixas altitudes, e um altímetro barométrico (MPL3115A2), capaz de estimar a altura com base na pressão atmosférica.

Atuadores são dispositivos responsáveis por converter sinais elétricos em ações físicas, possibilitando a interação ativa de um sistema embarcado com o ambiente. Em sistemas de controle, os atuadores representam o estágio final da cadeia de processamento, sendo acionados conforme os comandos gerados pelo controlador para alterar o estado do sistema físico.

No contexto deste trabalho, os atuadores são compostos pelos motores elétricos *brushless* (BLDC - *Brushless Direct Current*) e seus respectivos controladores eletrônicos de velocidade (ESCs – *Electronic Speed Controllers*). Os ESCs recebem sinais de controle modulados em PWM gerados pelo microcontrolador e, a partir desses sinais, regulam a tensão e a frequência aplicadas aos enrolamentos do motor. Isso permite controlar com precisão a velocidade de rotação dos motores, que por sua vez determina a força de empuxo gerada pelas hélices do drone.

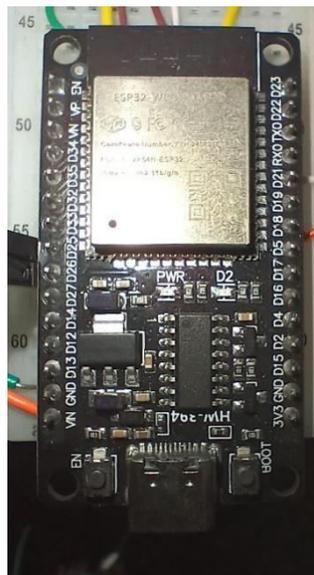
Cada sensor possui características distintas em termos de alcance, precisão, sensibilidade a ruído e frequência de resposta. A seguir, são apresentados as

descrições e os princípios de funcionamento dos transdutores e atuadores utilizados no projeto.

3.1.1 Unidade de controle

O controle central do quadcoptero foi realizado por meio de um ESP32, um microcontrolador altamente integrado com dois núcleos de 32 bits e clock de até 240 MHz. O ESP32, visível na Figura 9, destaca-se por sua combinação de alto desempenho de processamento, baixo consumo de energia e ampla variedade de interfaces de entrada/saída, sendo especialmente adequado para aplicações embarcadas em sistemas autônomos e conectados [11].

Figura 9 - ESP 32 Utilizado.



Fonte: O Autor (2025).

Uma das principais motivações para a escolha do ESP32 neste trabalho foi sua conectividade nativa, pois o chip incorpora suporte completo a Wi-Fi 802.11b/g/n e Bluetooth v4.2 (incluindo BLE), eliminando a necessidade de módulos externos para comunicação sem fio. Essa característica facilitou, tanto o monitoramento remoto de

variáveis, em tempo real, quanto a configuração de parâmetros por meio de uma interface web desenvolvida especificamente para o projeto.

O microcontrolador é responsável por adquirir os sinais dos sensores embarcados, processá-los e calcular, em tempo real, as ações de controle. Os algoritmos PID otimizados são executados diretamente no ESP32, que então envia sinais PWM aos ESCs para acionar os motores.

Os sensores embarcados incluem uma IMU, utilizada para estimar os ângulos de *roll*, *pitch* e *yaw*, e um sensor de distância ultrassônico, responsável pela medição da altitude relativa ao solo. A integração entre sensores, controlador e atuadores ocorre em malha fechada e com frequência máxima disponível dos sensores utilizados.

A arquitetura versátil do ESP32, combinada com sua conectividade sem fio, viabilizou o desenvolvimento de um sistema de controle embarcado eficiente, compacto e de fácil interação com o usuário.

3.1.2 IMU *Inertial Measurement Unit*

A Unidade de Medição Inercial (IMU - *Inertial Measurement Unit*) é um dispositivo composto por sensores capazes de medir acelerações lineares e velocidades angulares em três eixos. Em geral, integra um acelerômetro, um giroscópio e, frequentemente, um magnetômetro, permitindo estimar a orientação e o movimento de um corpo no espaço tridimensional.

Com o avanço da tecnologia *Micro-Electro-Mechanical Systems* (MEMS), IMUs modernas tornaram-se compactas, acessíveis e amplamente utilizadas em aplicações embarcadas, como drones. Esses sensores fornecem dados em alta frequência e são fundamentais para algoritmos de controle de atitude. No entanto, por estarem sujeitos a ruídos e deriva, requerem técnicas de filtragem e calibração para garantir maior precisão [12].

3.1.2.1 Acelerômetro

O acelerômetro é um transdutor capaz de medir a aceleração linear de um corpo em relação a um referencial inercial. Em sistemas embarcados, são geralmente utilizados acelerômetros triaxiais, que fornecem medições nas direções x, y e z. Esses sensores permitem não apenas a detecção de movimentos lineares, mas também a estimativa da inclinação da plataforma com relação à gravidade, especialmente em situações de repouso ou baixa velocidade.

A tecnologia mais comum em aplicações embarcadas é baseada em sistemas micro eletromecânicos, que operam por meio da deflexão de microestruturas internas sensíveis à força inercial. A equação que representa o modelo da saída de um acelerômetro simplificado, considerando as principais fontes que afetam a medição, é dada por:

$$a_m = a_r - a_g + b_a + n_a, \quad (3.1)$$

nessa equação, conforme apresentado em [12]:

- a_m representa a aceleração medida pelo sensor;
- a_r é a aceleração linear real do corpo em movimento;
- a_g é o vetor da aceleração gravitacional;
- b_a representa o viés (*bias*) do sensor, geralmente causado por imperfeições internas;
- n_a é o termo de ruído aleatório.

Em sistemas embarcados como o de um drone, o acelerômetro fornece informações essenciais para o controle de atitude, especialmente nos eixos de *roll* e *pitch*. No entanto, devido à presença de ruídos e vibrações, os dados brutos precisam ser tratados por filtros digitais antes de serem utilizados nos algoritmos de controle [12].

3.1.2.2 Giroscópio

O giroscópio é um transdutor utilizado para medir a velocidade angular de um corpo em torno de seus eixos de atuação. Em sistemas eletrônicos, os giroscópios

são comumente desenvolvidos como MEMS, fornecendo a velocidade angular através de sinais analógicos ou digitais [13]. Esses sensores são cruciais para o controle de atitude, complementando as medições do acelerômetro, pois permitem estimar a taxa de rotação do quadcoptero.

O modelo da saída de um giroscópio, considerando as principais fontes de erro de forma simplificada, pode ser representado por:

$$\omega_m = \omega_r + b_\omega + n_\omega, \quad (3.2)$$

nessa equação, conforme apresentado em [12]:

- ω_m representa a velocidade angular medida pelo sensor;
- ω_r é a velocidade angular real do corpo;
- b_ω representa o viés do giroscópio, que pode variar com o tempo e a temperatura;
- n_ω é o termo de ruído aleatório.

É importante notar que os giroscópios MEMS, ao contrário dos acelerômetros, geralmente possuem um viés e um ganho que variam com o tempo, além de serem fortemente dependentes da temperatura [12]. A modelagem do viés e do ganho como ruído branco integrado é utilizada para compensar pequenas variações nos parâmetros, enquanto a calibração pode minimizar o efeito da temperatura.

Em aplicações como o controle de quadcopteros, os dados do giroscópio são essenciais para determinar as taxas de rotação de *roll*, *pitch* e *yaw*. No entanto, esses dados também são suscetíveis a ruídos, incluindo vibrações, o que exige a aplicação de filtros digitais para garantir a precisão e a estabilidade dos algoritmos de controle [13].

3.1.2.3 Magnetômetro

O magnetômetro é um transdutor utilizado para medir o campo magnético ambiente, incluindo o campo magnético da Terra. Em aplicações de sistemas de controle, como em quadcopteros, ele é fundamental para determinar a orientação em relação ao Norte magnético e, conseqüentemente, o ângulo de *yaw* (guinada) da aeronave.

Apesar de existirem diversos tipos, os magnetômetros MEMS são frequentemente empregados em VANTs devido ao seu tamanho reduzido, baixo custo e alta disponibilidade. Embora esses sensores sejam geralmente compensados por temperatura, minimizando erros intrínsecos de ganho e viés, eles são bastante suscetíveis a campos magnéticos externos, que podem distorcer o campo da Terra e, assim, comprometer a precisão da detecção do Norte verdadeiro.

O modelo da saída de um magnetômetro, de forma simplificada, pode ser representado por:

$$b_m = b_r + b_b + n_b. \quad (3.3)$$

Nessa equação, conforme apresentado em [12]:

- b_m representa o campo magnético medido pelo sensor;
- b_r é o campo magnético real do ambiente;
- b_b representa o viés do sensor, que pode variar dependendo da posição e do tempo devido a campos magnéticos espúrios;
- n_b é o termo de ruído aleatório.

Em um quadcoptero, as medições do magnetômetro são essenciais para a estimativa de atitude, especialmente para o eixo de *yaw*. No entanto, a presença de interferências magnéticas do ambiente ou de outros componentes eletrônicos do próprio drone (como motores e ESCs) pode introduzir erros significativos, exigindo técnicas de filtragem e processamento de dados para mitigar seus efeitos [13].

3.1.2.4 BNO-055

O BNO055, visto na Figura 10, é um *System in Package* (SiP) inteligente, desenvolvido pela Bosch Sensortec, que integra um acelerômetro triaxial de 14 bits, um giroscópio triaxial de 16 bits (com uma faixa de ± 2000 graus por segundo) e um sensor geomagnético triaxial. Além desses sensores, o chip incorpora um microcontrolador *Cortex M0+* de 32 bits, que executa o software de fusão de sensores da Bosch Sensortec. Para integração com outros sistemas, o BNO055 é equipado com interfaces digitais bidirecionais I²C e UART [14].

Figura 10 - BNO Utilizado.



Fonte: O Autor (2025).

O BNO055 oferece diversos modos de operação, que podem ser configurados para atender a diferentes necessidades da aplicação. Esses modos são divididos nas seguintes categorias:

- **Modo de Configuração (*CONFIGMODE*):** Permite configurar o BNO055, reiniciando todos os dados de saída para zero e interrompendo a fusão de sensores. Este é o modo padrão ao ligar ou reiniciar o dispositivo.
- **Modos Não-Fusão (*Non-Fusion Modes*):** Fornecem acesso individual ou combinado aos dados brutos dos sensores, sem processamento de fusão. O BNO055 se comporta como um sensor autônomo, e os outros sensores não utilizados são suspensos para reduzir o consumo de energia. Nesse modo ele pode ser utilizado nos seguintes modos:
 - ***ACCONLY*:** Apenas dados do acelerômetro.
 - ***MAGONLY*:** Apenas dados do magnetômetro.
 - ***GYROONLY*:** Apenas dados do giroscópio.
 - ***ACCMAG*:** Dados do acelerômetro e magnetômetro.
 - ***ACCGYRO*:** Dados do acelerômetro e giroscópio.
 - ***MAGGYRO*:** Dados do magnetômetro e giroscópio.
 - ***AMG (ACC-MAG-GYRO)*:** Todos os três sensores (acelerômetro, magnetômetro e giroscópio) estão ativos.

- **Modos de Fusão (*Fusion Modes*):** Calculam a orientação do dispositivo no espaço, distinguindo entre orientação não-absoluta (relativa) e absoluta (com relação à Terra e seu campo magnético). Todos os modos de fusão fornecem a orientação como dados de quatérnios ou ângulos de Euler. Nesses modos, é possível separar a aceleração linear da gravidade. As configurações dos sensores são controladas pelo BNO055 com valores padrão. Nesse modo ele pode ser utilizado nos seguintes modos:
 - **IMU:** Calcula a orientação relativa a partir de dados do acelerômetro e giroscópio, com alta taxa de saída de dados. Taxa de saída de dados de fusão: 100Hz.
 - **COMPASS:** Mede o campo magnético da Terra para calcular a direção geográfica, utilizando o campo magnético e a gravidade. Taxa de saída de dados de fusão: 20Hz.
 - **M4G (*Magnet for Gyroscope*):** Similar ao modo IMU, mas usa o magnetômetro para detectar rotação em vez do giroscópio, resultando em menor consumo de energia e sem os efeitos de *drift* inerentes ao giroscópio. Taxa de saída de dados de fusão: 50Hz.
 - **NDOF (*Nine Degrees of Freedom*):** Este modo é um modo de fusão de 9 graus de liberdade, onde os dados de orientação absoluta são calculados a partir do acelerômetro, giroscópio e magnetômetro. Combina os três sensores para um cálculo rápido, alta taxa de saída de dados e alta robustez contra distorções do campo magnético. A calibração rápida do magnetômetro é ativada para maior precisão. Taxa de saída de dados de fusão: 100Hz.
 - **NDOF_FMC_OFF (*Nine Degrees of Freedom with Fast Magnetometer Calibration OFF*):** É idêntico ao modo NDOF, mas com a Calibração Rápida do Magnetômetro (*Fast Magnetometer Calibration*) desativada. Taxa de saída de dados de fusão: 100Hz.

No contexto deste trabalho, o chip BNO055 foi utilizado em nosso quadcoptero operando no modo NDOF, pois a fusão interna feita pelo dispositivo se mostrou melhor do que calcular no microcontrolador devido à alta precisão e baixo drift se tornou o melhor modo para essa aplicação.

3.1.3 Altímetro

Um altímetro é um transdutor utilizado para medir a altitude, ou seja, a altura de um ponto em relação a um nível de referência, geralmente o nível do mar. Dentre os diversos tipos existentes, os altímetros barométricos são os mais comuns em aplicações embarcadas, como em aeronaves e drones. Esses dispositivos operam com base na variação da pressão atmosférica, que diminui de forma previsível conforme a altitude aumenta.

A relação entre pressão e altitude é descrita por modelos atmosféricos padrão, como o ISA (*International Standard Atmosphere*), permitindo a conversão direta da pressão absoluta medida pelo sensor em uma estimativa da altitude. Essa medição é fundamental para o controle vertical de veículos aéreos, especialmente em ambientes externos, onde sensores ópticos ou ultrassônicos apresentam limitações de alcance ou confiabilidade.

Embora os altímetros barométricos apresentem maior alcance de medição, sua resolução e taxa de atualização tendem a ser inferiores quando comparados a sensores de distância de curto alcance, como os ultrassônicos. Além disso, estão sujeitos a variações ambientais, como mudanças de temperatura e pressão atmosférica local, que podem introduzir erros no cálculo da altitude se não forem devidamente compensadas [15].

Neste projeto, um altímetro barométrico foi utilizado de forma complementar ao sensor ultrassônico HC-SR04. Enquanto o ultrassônico apresenta boa precisão em baixas altitudes (até aproximadamente 4 metros), o altímetro barométrico estende a capacidade de medição para altitudes superiores, mesmo com menor precisão em ambientes internos. A combinação dos dois sensores permite um sistema de controle de altitude mais robusto e adaptável a diferentes cenários operacionais.

3.1.3.1 MPL

O MPL3115A2, visto na Figura 11, é um sensor barométrico de pressão absoluta desenvolvido pela *NXP Semiconductors*, projetado especificamente para aplicações de medição de altitude e pressão atmosférica. Baseado na tecnologia MEMS, o dispositivo integra um conversor analógico-digital de 24 bits, interface digital I²C e um

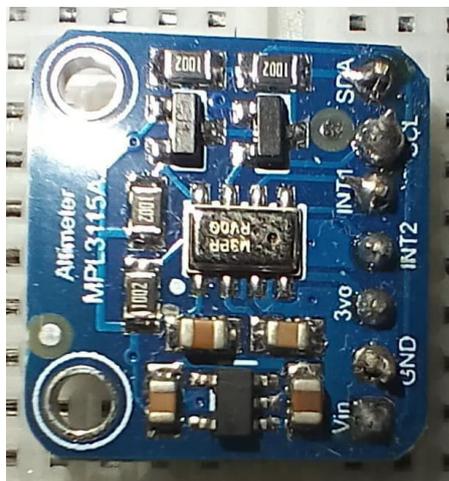
processador interno que realiza os cálculos de altitude diretamente a partir da pressão absoluta medida [15].

Este sensor opera em uma faixa de pressão de 50 a 110 kPa, o que permite estimar altitudes desde o nível do mar até aproximadamente 10.000 metros. A precisão da medição de altitude pode chegar a cerca de $\pm 0,3$ metros em condições ideais, embora fatores como temperatura e variações ambientais possam afetar essa precisão. O dispositivo também possui um modo de medição barométrico e um modo de medição altimétrica, selecionáveis via registradores internos.

Além disso, o MPL3115A2 conta com um sensor de temperatura interno, cuja leitura pode ser utilizada para compensações térmicas na estimativa da altitude. A comunicação com o microcontrolador é realizada por meio do protocolo I²C, permitindo fácil integração em sistemas embarcados de controle e navegação.

As características do MPL3115A2 o tornam apropriado para aplicações em que se requerem medições de altitude com boa faixa dinâmica, mesmo que com resposta temporal moderada e sensibilidade a variações ambientais.

Figura 11 – MPL Utilizado.



Fonte: O Autor (2025).

3.1.4 Ultrassônico

Sensores ultrassônicos utilizam ondas sonoras de alta frequência, inaudíveis ao ouvido humano, para determinar a distância até um objeto. O princípio de funcionamento baseia-se na emissão de um pulso sonoro e na medição do tempo que o eco leva para retornar ao sensor. A distância é calculada multiplicando a velocidade do som pela metade do tempo de viagem da onda. A velocidade do som no ar pode variar de acordo com condições atmosféricas como temperatura, umidade e pressão [16].

Esses sensores possuem um "cone de detecção" cuja largura pode mudar com a distância. A capacidade de detectar um objeto também depende da sua orientação, pois superfícies não planas podem desviar o eco, impedindo seu retorno ao sensor. Embora sejam eficazes em muitas aplicações, os sensores ultrassônicos têm limitações, como a dependência da velocidade de medição em relação à distância (quanto maior a distância, mais lenta a leitura) e a suscetibilidade a ecos indesejados em ambientes confinados, que podem gerar leituras falsas e imprecisas [16].

3.1.4.1 HC-S04

O HC-SR04, visto na Figura 12, é um sensor ultrassônico popular, conhecido por seu baixo custo e facilidade de uso. Ele opera com alimentação de +5V DC e detecta distâncias de 2 a 400 cm com resolução de 0.3 cm. O sensor possui quatro pinos: VCC e GND para alimentação, TRIG, que habilita a emissão de um pulso sonoro de 40 kHz, e ECHO para medir o tempo de retorno do eco [16].

Embora tenha sua dinâmica lenta como as atividades previstas pelo drone será em um ambiente fechado se faz necessário a precisão disponível por ele e a sua dinâmica é compensada no controle.

Figura 12 - Sensor Ultrassônico.



Fonte: O Autor (2025).

3.1.5 Electronic Speed Controller (ESC)

O ESC é um componente essencial em sistemas de controle de motores elétricos, como drones e veículos controlados por controle remoto (CR). Sua função principal é regular a velocidade, direção e frenagem do motor, ajustando a potência fornecida pela bateria. Atuando como uma interface entre a fonte de energia e o motor, o ESC assegura uma aceleração e desaceleração suaves, ao mesmo tempo que previne flutuações de tensão e corrente que poderiam danificar o sistema [17].

ESCs recebem sinais de controladores de voo, microcontroladores ou receptores remotos, que são tipicamente pulsos PWM variando de 1000 μ s (0%) a 2000 μ s (100%). O ESC interpreta esses sinais para determinar a velocidade e potência necessárias do motor. Para motores brushless DC (BLDC), o ESC converte a energia DC da bateria em uma saída AC trifásica, controlando as fases do motor através de MOSFETs organizados em circuitos de ponte H ou ponte trifásica. ESCs avançados também incluem controle de frenagem (inclusive frenagem regenerativa) e direção [17].

ESCs podem ser classificados com base na metodologia de controle e compatibilidade com o motor. Existem ESCs para motores com escovas e ESCs para

motores sem escovas. Os ESCs para motores BLDC utilizam microcontroladores para chavear a potência eletronicamente entre as fases do motor, sendo altamente eficientes e duráveis. Podem ser sensorizados, usando sensores Hall para determinar a posição do rotor, ou sem sensor, estimando a posição através da força contra eletromotriz (*back-Eletromotive Force - EMF*) [17].

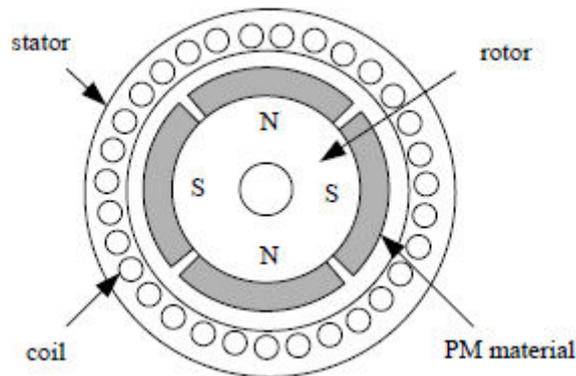
ESCs modernos frequentemente incorporam recursos de segurança, como proteção contra sobrecorrente, desligamento térmico e corte de baixa tensão, aumentando a confiabilidade e a vida útil do sistema [17], o ESC utilizado nesse projeto pode ser visto na Figura 17.

3.1.6 Motor Brushless

Um Motor *Brushless* DC é um tipo de motor elétrico que se diferencia dos motores DC convencionais por não possuir escovas mecânicas e comutadores para a inversão de corrente em suas bobinas. Em vez disso, a comutação, que é o processo de inversão do fluxo de corrente para manter a rotação, é realizada eletronicamente. Essa característica elimina o atrito mecânico, o ruído e as faíscas gerados pelas escovas, o que contribui para uma maior eficiência, vida útil prolongada e menor necessidade de manutenção [18].

A estrutura básica de um motor BLDC consiste em um estator, que é a parte fixa do motor e contém as bobinas da armadura, e um rotor, que é a parte móvel e é composto por ímãs permanentes. A Figura 13 ilustra essa configuração, mostrando o estator com as bobinas e o rotor com os ímãs permanentes. Os ímãs permanentes do rotor interagem com o campo magnético gerado pelas correntes nos enrolamentos do estator, produzindo o torque que impulsiona o motor. Para controlar a rotação, sensores de posição (como sensores Hall) detectam a orientação do rotor. Essa informação é utilizada por um circuito eletrônico de controle que energiza as bobinas do estator na sequência correta para gerar um campo magnético girante que segue o rotor, mantendo-o em movimento contínuo e sincronizado [18], o motor utilizado nesse projeto foi o da Figura 15.

Figura 13 - Motor em Corte.



Fonte: Retirado do [18].

3.2 Montagem de Quadcoptero

Esta montagem envolveu a integração mecânica e eletrônica de diversos componentes, tais como estrutura (*frame*), motores, ESCs, sensores e unidade de processamento. Nesta seção são descritos os principais elementos utilizados no protótipo, destacando suas características técnicas, critérios de escolha e a forma como foram integrados ao sistema.

3.2.1 Frame

A estrutura utilizada foi o DJI F450, visível, um frame em configuração “X” com quatro braços equidistantes e centro rígido em material composto. Esse modelo é amplamente adotado em aplicações experimentais e didáticas por sua leveza, boa resistência mecânica e facilidade de montagem. Sua geometria garante simetria em relação aos eixos de rotação, favorecendo o balanceamento do sistema e a modelagem dinâmica. A Figura 14 apresenta o frame utilizado no projeto.

Figura 14 - *Frame* do drone utilizado.



Fonte: Imagem retirado de [19].

3.2.2 Sistema de propulsão

O sistema de propulsão foi composto por quatro motores *brushless* de 1000 KV, responsáveis por fornecer o empuxo necessário ao voo, visto na Figura 15. A especificação de 1000 KV indica que, para cada volt aplicado, o motor gira aproximadamente 1000 RPM sem carga.

Figura 15 - Motor Utilizado.



Fonte: O Autor (2025).

Cada motor foi acoplado a uma hélice do tipo 1045, em que o número 10 indica o diâmetro de 10 polegadas e o número 45 representa o ângulo de inclinação de 4,5 polegadas por rotação. Essa hélice oferece um bom equilíbrio entre empuxo e

estabilidade, sendo compatível com os motores e o peso total do quadcoptero, visível na Figura 16.

Figura 16 - Hélice Utilizada.



Fonte: O Autor (2025).

Para o acionamento dos motores, utilizaram-se ESCs com corrente nominal de 30 A, suficientes para suportar os picos de corrente durante manobras bruscas sem risco de superaquecimento, pode ser visto na Figura 17. Os ESCs foram conectados diretamente à bateria e comandados por sinais PWM gerados pelo microcontrolador.

Figura 17 - ESC Utilizado.



Fonte: O Autor (2025).

3.2.3 Fonte de alimentação

A alimentação do sistema foi realizada por meio de uma bateria de íons de lítio (Li-íon) de 11,1 V (3S) e capacidade de 3000 mAh, visível na Figura 18. Essa bateria oferece densidade energética suficiente para suportar os testes em bancada com segurança, além de apresentar peso reduzido e fácil recarga. A configuração 3S indica a presença de três células em série, totalizando uma tensão nominal adequada para os motores e demais circuitos eletrônicos.

Figura 18 - Bateria Utilizada.



Fonte: O Autor (2025).

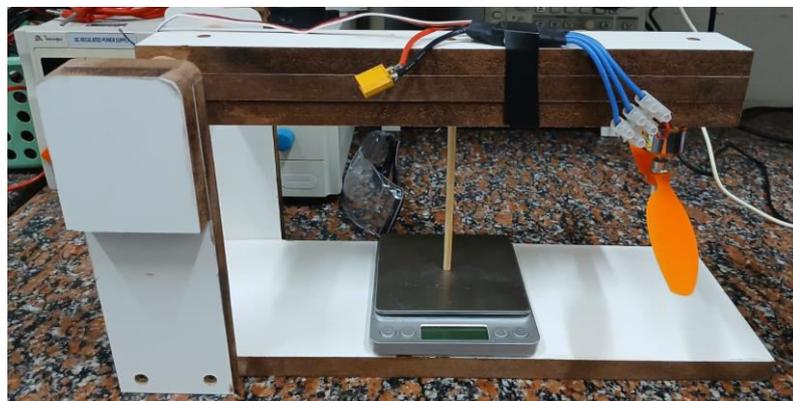
3.3 Bancadas

No desenvolvimento desse trabalho se viu a necessidade de bancadas de teste para o drone devido a sua complexidade, inicialmente foi projetado uma bancada para modelar o motor para a simulação, depois uma que isolasse o eixo z e a terceira que faz o movimento de *roll*, *pitch* e *yaw*. Essas bancadas foi um trabalho em conjunto de diversas pessoas tanta na construção física das mesmas, quanto na idealização delas.

3.3.1 Bancada de modelagem do motor

Antes de iniciar a modelagem no ambiente Simulink/Matlab, foi necessário caracterizar experimentalmente o comportamento do sistema motor-hélice utilizado no quadcoptero. Para isso, construiu-se uma bancada de testes específica, que pode ser visto na Figura 19. Nessa bancada, o motor brushless foi fixado de forma que sua força de empuxo atuasse perpendicularmente a uma haste rígida, cujo extremo permanecia apoiado sobre uma balança digital.

Figura 19 - Bancada de Teste do Motor.



Fonte: O Autor (2025).

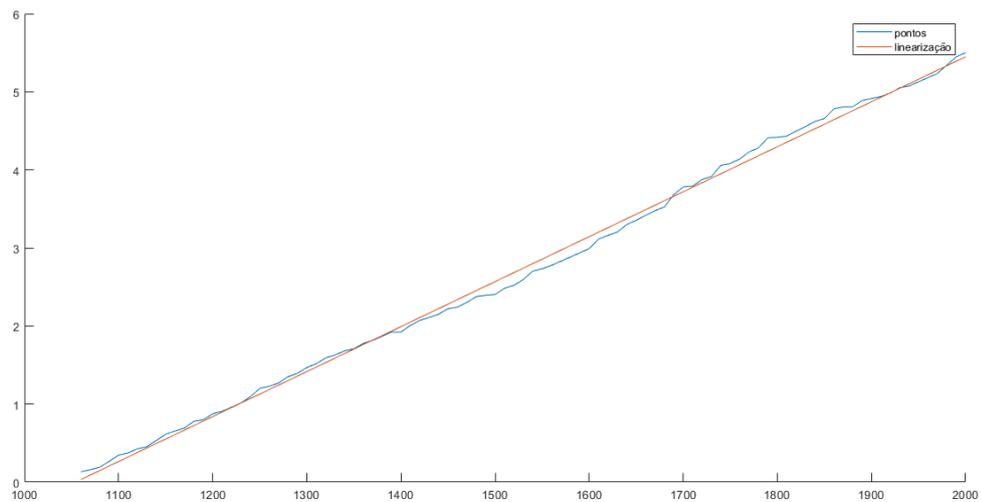
A bancada foi construída partir de tabuas de MDF (*Medium Density Fiberboard*) um fuso e dois rolamentos, os rolamentos ficam posicionados nas duas paredes laterais e servem para anular o atrito do fuso com a madeira e impedindo que haja escorregamento do fuso na madeira central.

Durante os testes, sinais PWM de 1000μ a 2000μ foram aplicados ao motor de 10μ em 10μ , e a massa registrada pela balança foi convertida em força (em Newtons) considerando a aceleração da gravidade local. A força de empuxo resultante foi associada ao sinal PWM correspondente, possibilitando a obtenção de um conjunto de dados experimentais relacionando entrada (PWM) e saída (força). É possível ver um teste dessa bancada em [20].

Com base nesses dados, foi elaborado um código no ambiente Matlab para ajuste de uma função matemática que descrevesse esse comportamento. Assumindo

um modelo de primeira ordem, foi realizada uma regressão linear que resultou em uma função PWM × Força, visto na Figura 20. Essa função serviu de base para a modelagem dos atuadores na simulação do quadcoptero no Simulink, garantindo maior fidelidade entre o modelo computacional e o sistema real.

Figura 20 - Curva Força do Motor x Linearização.



Fonte: O Autor (2025)

3.3.2 Bancada para testes de controle de altitude

A segunda bancada construída teve como finalidade possibilitar o teste do controle no eixo vertical (eixo Z) de forma isolada. A estrutura, ilustrada na Figura 21, foi montada a partir de tubos de PVC e consiste em um sistema de guias verticais, no qual o drone pode se mover apenas para cima e para baixo, sem liberdade para rotacionar nos eixos de *roll*, *pitch* ou *yaw*.

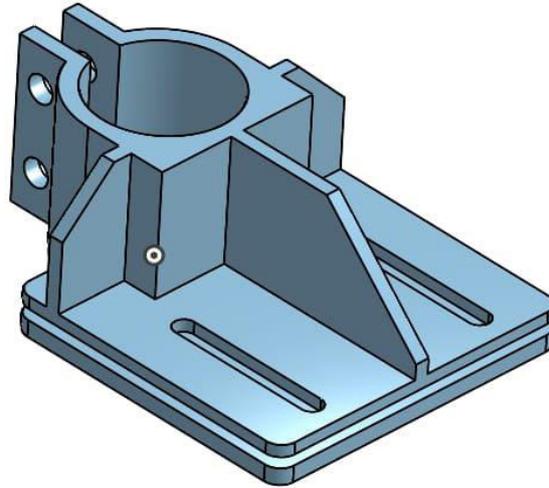
Figura 21 - Bancada de Altura.



Fonte: O Autor (2025).

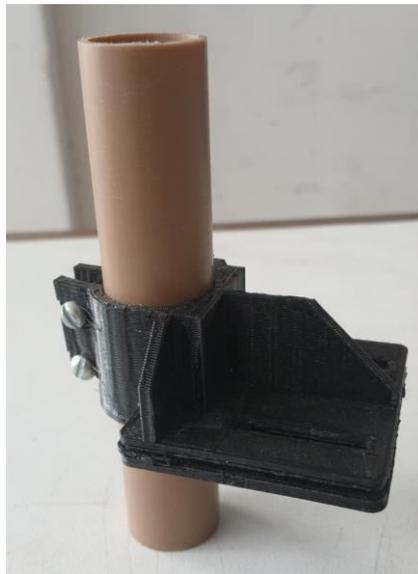
A montagem baseia-se em dois tubos paralelos fixados verticalmente a uma base retangular. No centro dessa estrutura, um tubo de diâmetro maior é acoplado ao corpo do drone por meio de suportes personalizados, desenvolvidos e impressos em 3D para garantir firmeza e alinhamento, visível nas Figuras 22 e 23. Esse tubo maior desliza entre os tubos guias com folga suficiente para permitir o movimento vertical, mas restringe completamente os demais graus de liberdade.

Figura 22 - Modelo 3D do suporte desenvolvido.



Fonte: O Autor (2025).

Figura 23 - Suporte impresso



Fonte: O Autor (2025).

A Figura 24 mostra o drone em operação na bancada, já com os motores acionados e controlado pelo algoritmo PID sintonizado. A estrutura garantiu segurança durante os testes e permitiu a observação do comportamento do sistema em malha fechada sob condições reais.

Figura 24 - Drone durante teste na bancada de altitude.



Fonte: O Autor (2025).

Essa abordagem se mostrou eficaz para a validação do controle de altitude, permitindo comparar os resultados experimentais com os obtidos nas simulações, além de reduzir significativamente os riscos associados a voos livres em ambiente interno. É possível ver um teste dessa bancada em [21].

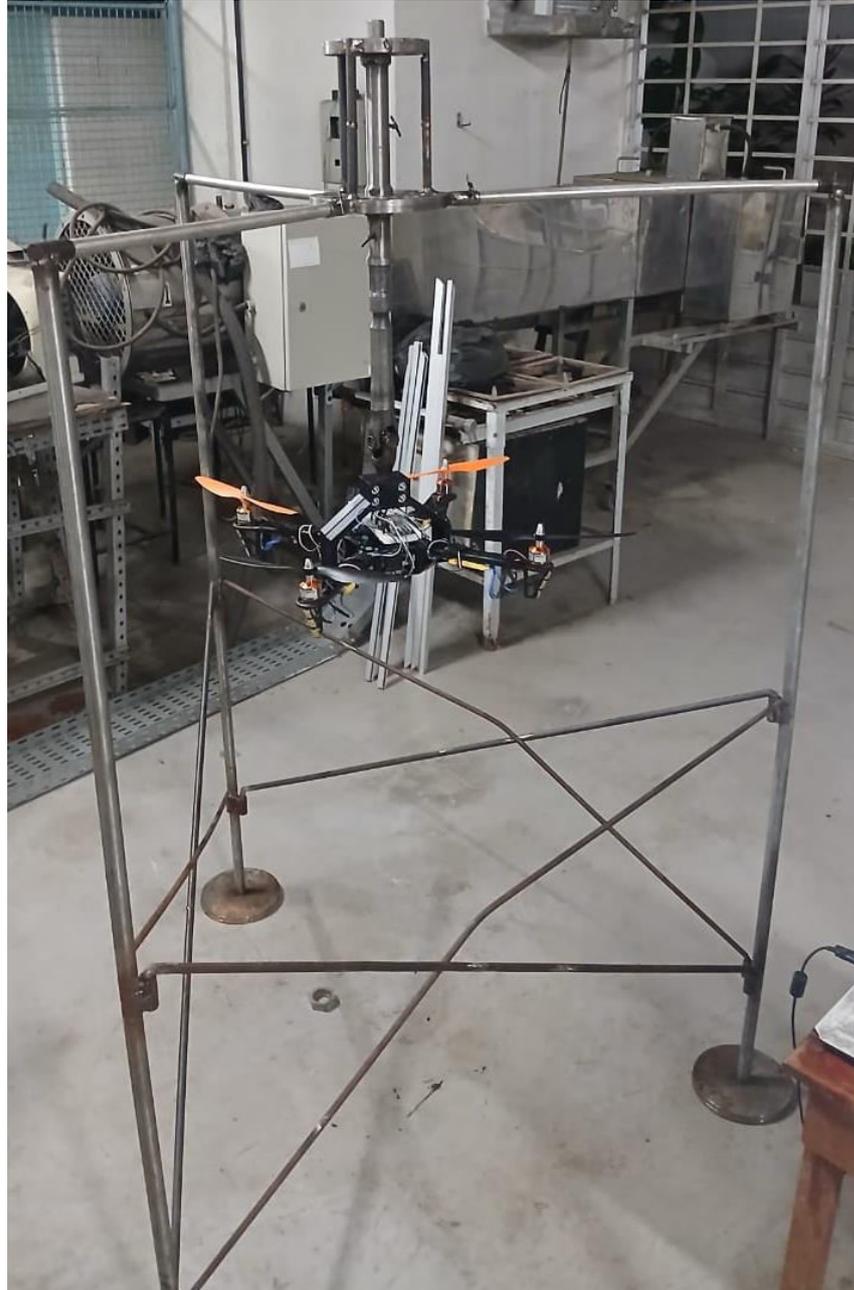
3.3.3 Bancada para testes de controle de atitude (*roll*, *pitch* e *yaw*)

A terceira bancada teve como finalidade testar o sistema de controle de atitude da aeronave, permitindo rotação livre nos eixos x , y e z , correspondentes aos ângulos de *roll*, *pitch* e *yaw*, respectivamente. Nessa configuração, o deslocamento vertical do drone é completamente impedido, permitindo concentrar a análise nas variáveis angulares.

A estrutura principal é composta por uma armação metálica do tipo tripé conforme pode ser observado na Figura 25, construída a partir de uma junta homocinética automotiva (um eixo cardã) e dois rolamentos. O cardã é o que possibilita as rotações de *roll* e *pitch*, e para o movimento de *yaw* foi utilizado 2 rolamentos (1 rolamento apenas já garante o movimento o segundo foi utilizado pois para evitar o movimento do eixo durante testes foi necessário a construção de um suporte mecânico na “ponta” do eixo por isso existe dois círculos metálicos na parte superior da estrutura). O quadcoptero foi fixado por meio de um suporte impresso em 3D visto nas Figuras 26

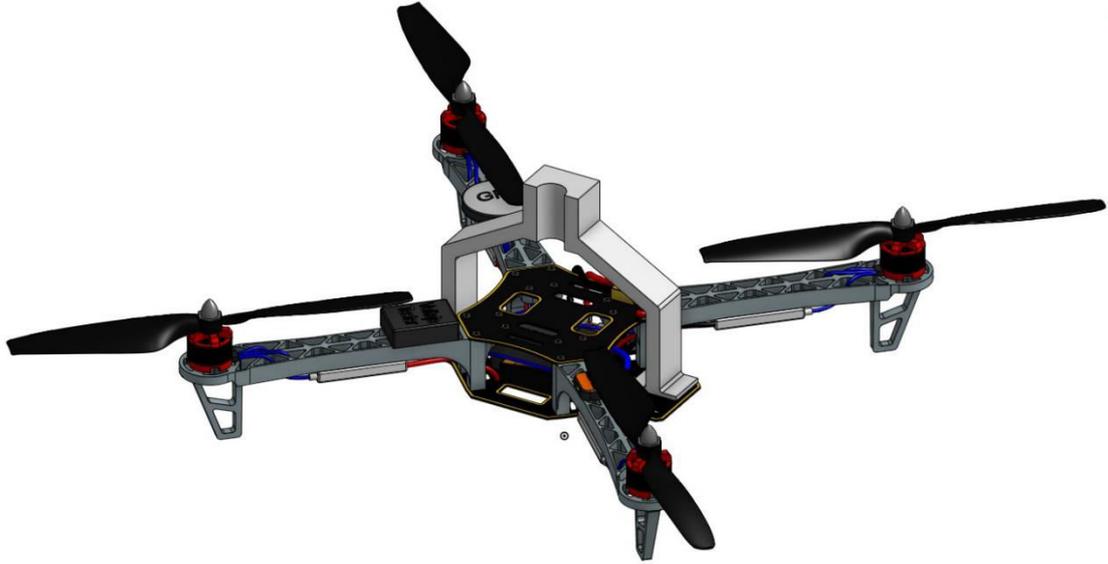
e 27, que o conecta a ponta inferior do eixo cardã por pressão onde duas peças da Figura 26 são colocadas e fixadas via 4 parafusos e porcas.

Figura 25 - Bancada de teste de atitude.



Fonte: O Autor (2025).

Figura 26 - Modelo 3D desenvolvido



Fonte: O Autor (2025).

Figura 27 - Drone na bancada de atitude.



Fonte: O Autor (2025).

Essas juntas permitem o movimento rotacional controlado nos eixos de interesse, mantendo o drone suspenso e impedindo deslocamentos lineares. A configuração adotada possibilitou a realização de testes completos de controle de atitude com segurança, estabilidade e baixo risco operacional.

Ainda que os eixos de rotação não coincidam exatamente com o centro de massa do drone, uma vez que a articulação ocorre externamente, a bancada mostrou-se altamente funcional. Sua aplicação foi essencial para validar os controladores sintonizados em simulação, permitindo testes confiáveis em todos os três graus de liberdade rotacionais.

Vale ressaltar que um dos eixos de rotação do cardã (nos experimentos realizados foi o eixo x) tinha muito mais atrito que o outro, isso implica que o efeito pêndulo é muito mais predominante em um dos movimentos do que o outro e será bem visível na seção de resultados experimentais. É possível ver um teste dessa bancada em [22].

4 MODELAGEM E CONTROLE DE QUADCÓPTERO BASEADO EM ALGORITMO GENETICO

4.1 Modelagem Dinâmica do Quadcoptero

As equações dinâmicas completas do quadcoptero, derivadas da técnica de Newton-Euler, descrevem a variação dos estados de posição linear (X, Y, Z), velocidade linear (U, V, W), posição angular (*Roll, Pitch, Yaw*) e velocidade angular (p, q, r) ao longo do tempo. Neste trabalho foi utilizado uma combinação e simplificação das equações presentes em diversos trabalhos sendo eles: [1–3], [9], [12], [25-28]:

$$\tau_z = f_1 + f_2 + f_3 + f_4, \quad (3.1)$$

$$\tau_\phi = \frac{d}{\sqrt{2}}(-f_1 - f_2 + f_3 + f_4), \quad (3.2)$$

$$\tau_\theta = \frac{d}{\sqrt{2}}(f_1 - f_2 - f_3 + f_4), \quad (3.3)$$

$$\tau_\psi = m_1 - m_2 + m_3 - m_4. \quad (3.4)$$

Onde f_x são as forças geradas pelos respectivos motores e m_x são os torques rotacionais gerados pelo efeito giroscópicos, d é o comprimento de um “braço”. E as equações que regem o movimento são as seguintes:

$$\ddot{\phi} = \left(\frac{J_{yy} - J_{zz}}{J_{xx}} \right) * \dot{\theta} * \dot{\psi} + \frac{\tau_\phi}{J_{xx}}, \quad (3.5)$$

$$\ddot{\theta} = \left(\frac{J_{zz} - J_{xx}}{J_{yy}} \right) * \dot{\phi} * \dot{\psi} + \frac{\tau_\theta}{J_{yy}}, \quad (3.6)$$

$$\ddot{\psi} = \left(\frac{J_{xx} - J_{yy}}{J_{zz}} \right) * \dot{\phi} * \dot{\theta} + \frac{\tau_\psi}{J_{zz}}, \quad (3.7)$$

$$\ddot{z} = \cos \theta * \cos \phi * \frac{\tau_z}{m} - g. \quad (3.8)$$

Onde, a matriz de inercia do quadcoptero (J), assumindo que ele é simétrico, é representado por:

$$J = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}. \quad (3.9)$$

E o restante dos símbolos são:

- $\ddot{\phi}$ é a aceleração rotacional no eixo x;
- $\ddot{\theta}$ é a aceleração rotacional no eixo y;
- $\ddot{\psi}$ é a aceleração rotacional no eixo z;
- \ddot{z} é a aceleração linear no eixo z.

4.1.1 Estratégia de Controle Distribuído nos Motores

Após a obtenção dos sinais de controle correspondentes aos quatro graus de liberdade principais, altitude (z), rotação em *roll* (ϕ), *pitch* (θ) e *yaw* (ψ), é necessário convertê-los em comandos individuais para cada motor. Essa conversão deve considerar tanto a configuração geométrica da aeronave quanto os sentidos de rotação de cada hélice. A equação usada neste trabalho segue a forma clássica para quadcopteros em configuração em “X”, com os motores dispostos conforme descrito anteriormente.

A estrutura de controle utilizada para o cálculo do sinal PWM de cada motor é expressa da seguinte forma:

$$motor_1 = PIDz - PID\phi + PID\theta + PID\psi + throttle, \quad (3.10)$$

$$motor_2 = PIDz - PID\phi - PID\theta - PID\psi + throttle, \quad (3.11)$$

$$motor_3 = PIDz + PID\phi - PID\theta + PID\psi + throttle, \quad (3.12)$$

$$motor_4 = PIDz + PID\phi + PID\theta - PID\psi + throttle. \quad (3.13)$$

Nesta estrutura, cada termo tem a seguinte função:

- $PIDz$: componente de controle da altitude, comum a todos os motores;
- $PID\phi$: componente de *roll*, que afeta simetricamente os lados esquerdo e direito;
- $PID\theta$: componente de *pitch*, que afeta os pares dianteiros e traseiros;
- $PID\psi$: componente de *yaw*, que depende do torque de reação associado ao sentido de rotação de cada motor;
- *throttle*: valor base de potência (PWM) para auxiliar numa melhor resposta individual de cada controle.

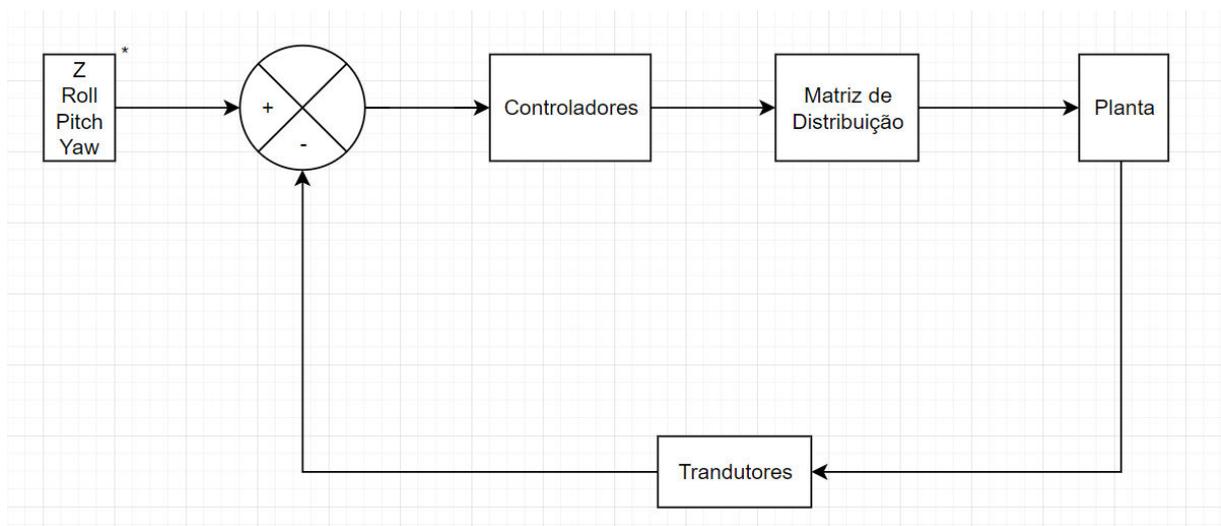
A lógica por trás dessa combinação é permitir que cada grau de liberdade seja controlado independentemente por variações adequadas nos motores. Essa estratégia de controle vetorial é amplamente empregada em controladores embarcados de quadcopteros por sua simplicidade computacional e eficiência prática.

Para uma representação mais compacta e visualização do sistema de controle em malha fechada, as equações de comando dos motores (3.10 a 3.13) podem ser expressas na forma matricial, desconsiderando o termo constante de *throttle* por ser idêntico a todos os motores:

$$\begin{bmatrix} Motor_1 \\ Motor_2 \\ Motor_3 \\ Motor_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} PIDz \\ PID\phi \\ PID\theta \\ PID\psi_4 \end{bmatrix}. \quad (3.14)$$

A Figura 19 ilustra a malha de controle simplificada que implementa essa estratégia. Nela, o vetor de *setpoints* para altitude, *roll*, *pitch* e *yaw* é comparado com as medições dos transdutores, gerando um vetor de erros. Esses erros são processados pelos controladores PID individuais, cujas saídas são combinadas através da matriz de distribuição de controle (como mostrado na Equação 3.14) para gerar os comandos para cada um dos quatro motores. O *feedback* do sistema é providenciado pelos transdutores, que medem a resposta do quadcoptero e realimentam a malha, garantindo a estabilidade e o rastreamento dos *setpoints* desejados.

Figura 28 - Malha Simplificada de controle.



Fonte: O Autor (2025).

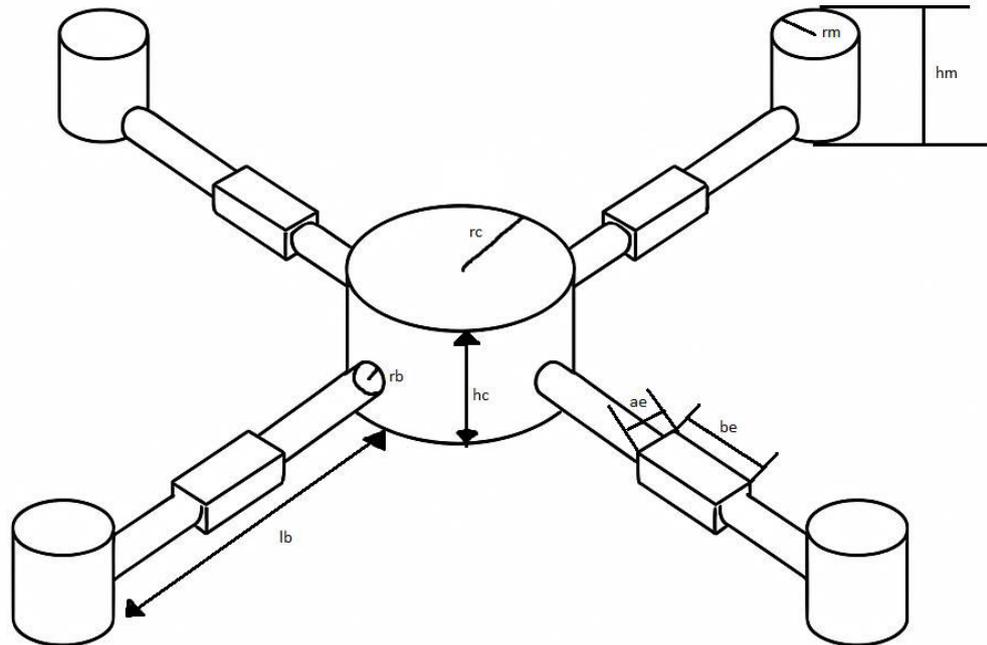
4.1.2 Representação do Quadcoptero em ambiente Simulink/Matlab

Para representar adequadamente a dinâmica rotacional do quadcoptero no ambiente Simulink, foi necessário determinar a sua matriz de inércia J . Essa matriz descreve a distribuição de massa em torno dos eixos principais do sistema e influencia diretamente os momentos angulares resultantes.

O modelo físico foi aproximado por uma composição de sólidos geométricos simples, conforme ilustrado na Figura 29. Cada componente estrutural foi representado como segue:

- **Corpo central (hub):** cilindro de raio r_c , altura h_c e massa m_c ;
- **Braços:** cilindros de raio r_b , comprimento l_b e massa m_b dispostos em configuração simétrica em “X”;
- **Motores:** cilindros de massa m_m , altura h_m e raio r_m , posicionados nas extremidades dos braços;
- **ESCs:** paralelepípedos de largura a_e , comprimento b_e e massa m_e , localizados a uma distância d_e do centro de massa do sistema, ao longo dos braços.

Figura 29 - Modelo do quadcoptero simplificado.



Fonte: O Autor (2025).

A matriz de inércia total do sistema foi calculada por meio da soma das contribuições individuais de cada componente. Para isso, considerou-se o momento de inércia de cada sólido em relação ao seu próprio centro de massa e aplicou-se o teorema dos eixos paralelos (ou teorema de Steiner) para referenciar os momentos ao centro de massa do quadcoptero:

$$J_{componente} = J_{CM} + md^2, \quad (3.15)$$

em que:

- J_{CM} é o momento de inércia do componente em relação ao seu centro de massa;
- m é a massa do componente;
- d é a distância entre o centro de massa do componente e o centro de massa do sistema.

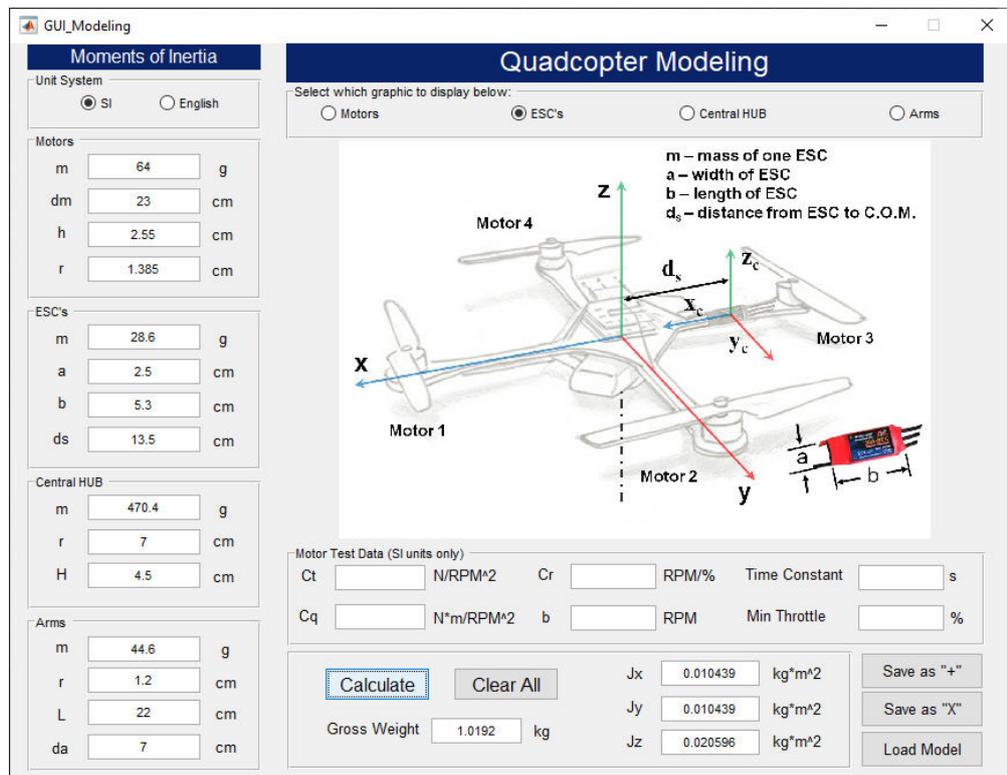
Devido à simetria estrutural e distribuição uniforme dos braços, motores e ESCs, a matriz de inércia resultante assumiu a forma diagonal:

$$J = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}, \quad (3.16)$$

em que J_{xx} e J_{yy} apresentam valores iguais, dado que os eixos x e y são simétricos em relação à geometria adotada.

O cálculo final da matriz foi realizado por meio de uma interface gráfica desenvolvida no ambiente MATLAB feita por terceiros, disponível em [23], que permitiu a entrada dos parâmetros físicos dos componentes e automatizou o processo de somatório dos momentos de inércia. A Figura 30 apresenta a interface gráfica utilizada para estimar a inércia total do sistema.

Figura 30 - Interface do cálculo da matriz de inércia.



Fonte: O Autor (2025).

Ao dar continuidade ao processo de modelagem, foram elaboradas duas simulações distintas no ambiente Simulink, com o objetivo de implementar as equações dinâmicas apresentadas no capítulo anterior. A primeira simulação representa a dinâmica rotacional do quadcoptero, abrangendo os ângulos de *roll*, *pitch*

e *yaw*, enquanto a segunda trata exclusivamente do eixo vertical (eixo Z), associado ao controle de altitude.

Para a implementação das equações diferenciais, utilizaram-se blocos do tipo *Function* (função definida pelo usuário), nos quais foi inserida diretamente a formulação matemática deduzida para cada caso. Os termos derivados foram integrados utilizando blocos padrão de integração do Simulink (*Integrator*), o que permitiu a obtenção das variáveis de estado em função do tempo.

Vale destacar que, embora a força de empuxo dos motores tenha sido caracterizada experimentalmente por meio da bancada de testes, o momento de torção gerado pelas hélices, fundamental para a modelagem precisa do movimento de *yaw*, não foi medido diretamente. A construção de uma bancada específica para essa finalidade foi inicialmente planejada, mas, por limitações de tempo e escopo, acabou não sendo implementada nesta etapa do trabalho. Como consequência, a simulação da dinâmica de *yaw* apresentada nesta seção não representa com exatidão o comportamento físico do sistema real, servindo apenas como uma aproximação teórica com fins didáticos, seria algo semelhante a Figura 31.

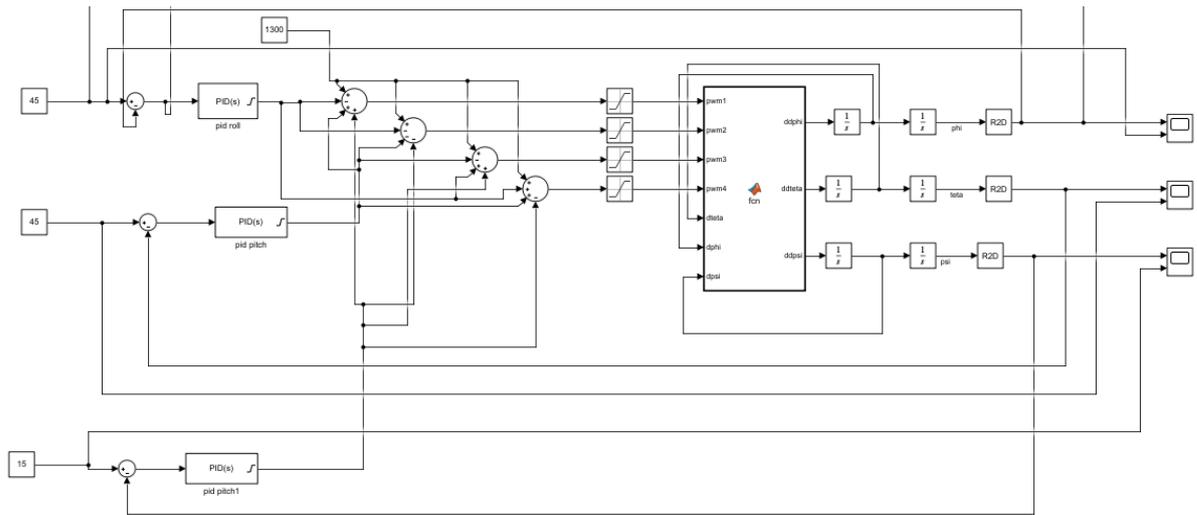
Figura 31 - Bancada de medição de torque rotacional planejada



Fonte: Retirado de [23]

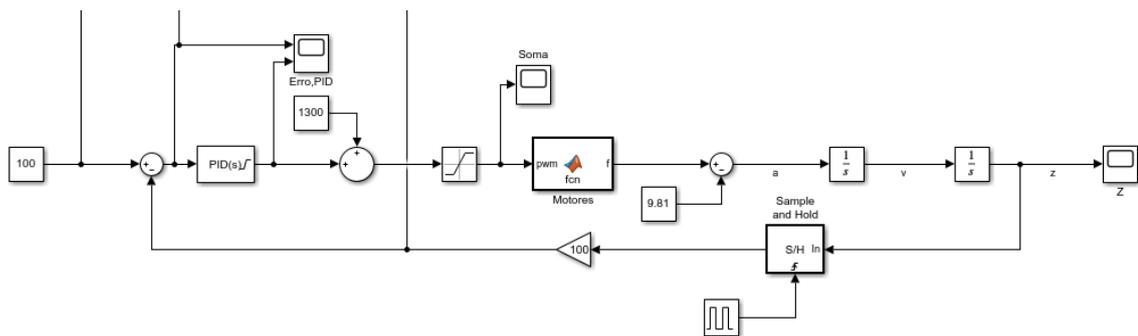
As simulações foram organizadas de modo modular, permitindo a visualização clara do fluxo de dados e facilitando ajustes futuros nos parâmetros do sistema. A Figura 32 apresenta a estrutura geral da simulação da dinâmica rotacional (*roll*, *pitch* e *yaw*) nela é possível ver os somatórios de controle que vão para os motores específicos e o bloco de função onde está implementado as equações descritas no início da seção, enquanto a Figura 33 ilustra o modelo dedicado à dinâmica vertical que por ter apenas um controle é visualmente mais simples porém segue o mesmo padrão a exceção é que como no controle do eixo z todos os motores recebem o mesmo comando isso foi passado para dentro do bloco de função diminuindo a poluição visual.

Figura 32 - Simulação de Movimento Rotacional.



Fonte: O Autor (2025).

Figura 33 - Simulação de Movimento Vertical.



Fonte: O Autor (2025).

Essa abordagem permitiu validar numericamente a consistência das equações modeladas e estabelecer uma base confiável para o projeto dos controladores descritos na seção seguinte.

4.2 Proposta de Metodologia de Projeto de Controladores baseada em GA aplicada em um quadcoptero

Nesta seção, propõe-se uma metodologia para o ajuste automático dos parâmetros dos controladores PID do quadcoptero, utilizando algoritmos genéticos

implementados no ambiente MATLAB/Simulink. A técnica foi aplicada tanto à malha de controle da dinâmica rotacional (*roll* e *pitch*) quanto à malha de controle da altitude (eixo Z), com o objetivo de melhorar o desempenho do sistema em relação a tempo de acomodação, sobressinal e erro estacionário, de forma automatizada e sistemática.

4.2.1 Etapa 1 – Seleção dos modelos simulados

As simulações desenvolvidas na seção 3.1.2 foram utilizadas como base para o processo de sintonia. Dois modelos distintos foram considerados:

- **Modelo 1:** Controle da atitude;
- **Modelo 2:** Controle da altitude;

Cada modelo contava com um controlador PID com três parâmetros a serem ajustados: K_p , K_i e K_d . Para isso foi criada uma função no Matlab que roda a simulação no Simulink e sua saída é a função de custo calculada internamente no ambiente Simulink como visto na Figura 35, o código implementado está visível na Figura 34.

Figura 34 - Exportação da função de custo.

```

1  function cost = tuning(k)
2
3  -   assignin('base','k',k);
4  -   sim('Quadcopter_phi_teta.slx');
5
6  -   cost= itse(length(itse));
7  -   end
8

```

Fonte: O Autor (2025).

4.2.2 Etapa 2 – Definição da função de custo

Índices de desempenho como IAE, ISE, ITAE e ITSE são amplamente adotados como funções de custo no projeto e otimização de controladores PID, tanto em métodos clássicos quanto em técnicas meta-heurísticas como algoritmos genéticos

[24]. O ITSE (*Integral Time Squared Error*), em especial, penaliza o erro quadrático ao longo do tempo, priorizando comportamentos estáveis e compactos. Em situações em que o *overshoot* é indesejável, como no controle de quadcopteros, pode-se incrementar o critério com fatores adicionais, como foi adotado neste trabalho.

O ITSE é definido como a integral do erro quadrático ponderado pelo tempo, conforme a seguinte equação:

$$ITSE = \int_0^T t * e^2(t) dt, \quad (3.17)$$

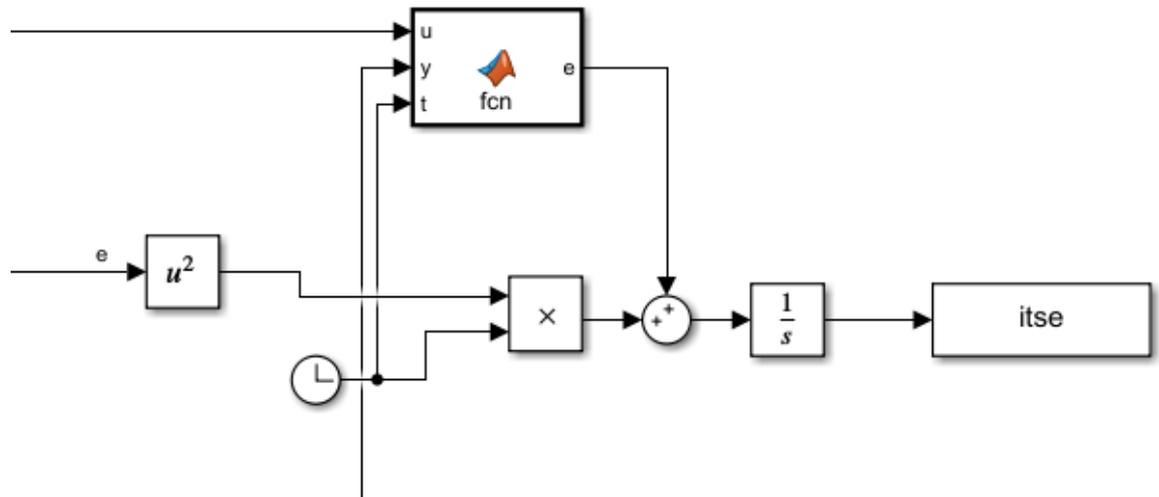
onde $e(t)$ é o erro do sistema e t é o tempo. Este critério é particularmente útil para minimizar tanto o erro de regime permanente quanto o tempo de acomodação do sistema, pois a ponderação pelo tempo amplifica a contribuição de erros prolongados.

A função de custo foi baseada no critério ITSE, com uma penalização adicional aplicada em caso de ocorrência de sobressinal. A equação empregada foi:

$$C = \begin{cases} ITSE + ITSE * 20 & \text{se houver sobressinal} \\ ITSE & \text{caso contrario} \end{cases}. \quad (3.18)$$

Essa formulação tem como objetivo principal eliminar ou minimizar a ocorrência de sobressinal, o que é particularmente desejável para essa aplicação. A função de custo foi calculada diretamente dentro do modelo Simulink como visto na Figura 35, e o valor final exportado para o MATLAB para ser utilizado na avaliação da qualidade de cada solução gerada pelo algoritmo.

Figura 35 – Cálculo da função de custo.



Fonte: O Autor (2025).

4.2.3 Etapa 3 – Configuração e execução do algoritmo genético

A função *ga()* do MATLAB foi utilizada para implementar o algoritmo genético. Foram definidos os parâmetros de número de gerações, tamanho da população e função de visualização gráfica da convergência, os valores finais de cada variável foram escolhidos de maneira empírica avaliando a resposta final gerada em simulação. O número de variáveis foi fixado em três, correspondendo aos ganhos do controlador PID. A Figura 36 apresenta o trecho de código MATLAB utilizado para configurar e executar o algoritmo genético.

Figura 36 - Execução do Algoritmo Genético.

```

1 - no_var = 3;
2 - lb = [0 0 0];
3 - ub = [10 10 10];
4
5 - ga_opt = gaoptimset('Display','off','Generations',50,'PopulationSize',50,'PlotFcns',@gaplotbestf);
6 - obj_fn = @(k) tuning(k);
7
8 - [k,best] = ga(obj_fn,no_var,[],[],[],[],lb,ub,[],ga_opt);

```

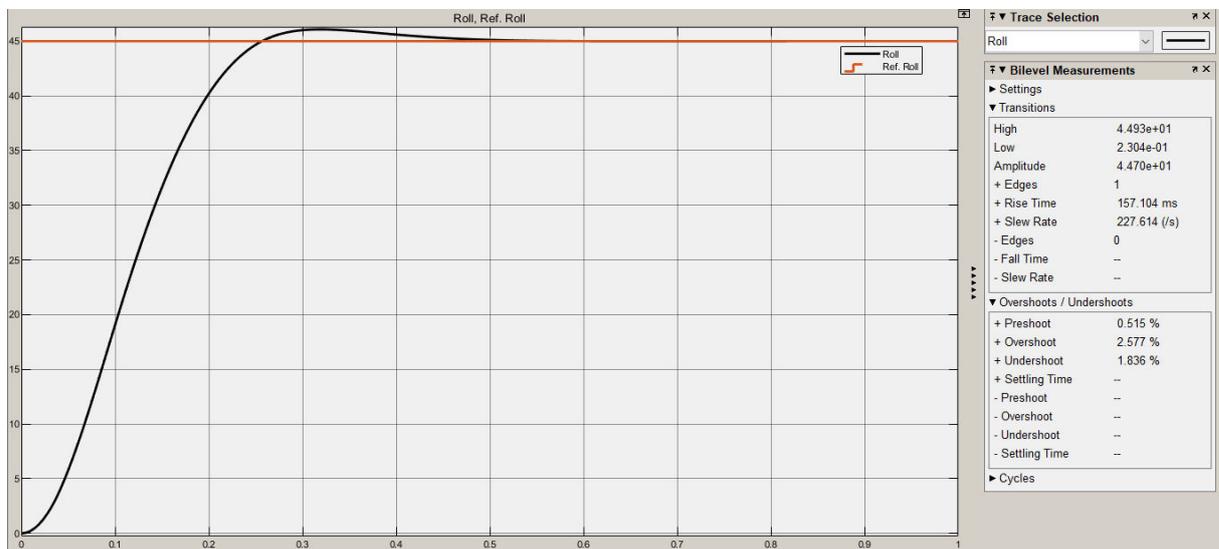
Fonte: O Autor (2025).

4.3 Validação da metodologia Proposta

4.3.1 Controle de Roll e Pitch

A Figura 37 exibe a resposta ao degrau de *roll* que é igual a resposta ao degrau de *pitch* em virtude da simetria do modelo adotado. Observa-se tempo de subida de aproximadamente 0,25 s, seguido de um sobressinal de baixa magnitude que é compensado antes de 0,50 s, resultando em tempo de acomodação inferior a 0,5 s. O erro de regime permanente é nulo. Esse comportamento confirma a efetividade da penalização do sobressinal introduzida na função de custo (Seção 3.2). Vale ressaltar que em um sistema real, pequenas assimetrias estruturais, atrasos de atuadores e ruídos nos sensores poderão provocar diferenças sutis entre os eixos x e y; contudo, tais imperfeições não são consideradas no modelo ideal adotado.

Figura 37 – Resposta em simulação de roll e pitch.



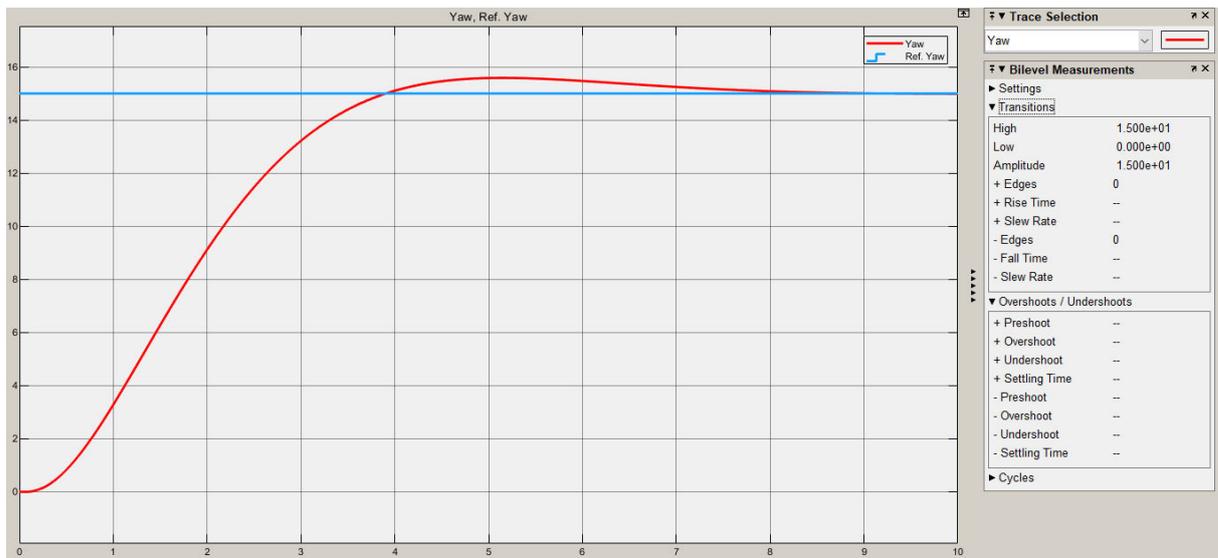
Fonte: O Autor (2025).

4.3.2 Controle de Yaw

A resposta simulada do eixo *yaw* é apresentada na Figura 38. Embora o modelo não contemple o momento de torção gerado pelas hélices, a simulação indica

ausência de sobressinal e tempo de acomodação em torno de 4,5 s. Essa dinâmica mais lenta decorre da simplificação do modelo e evidencia a necessidade de caracterização experimental desse eixo para futuros estudos.

Figura 38 - Resposta em simulação de yaw.

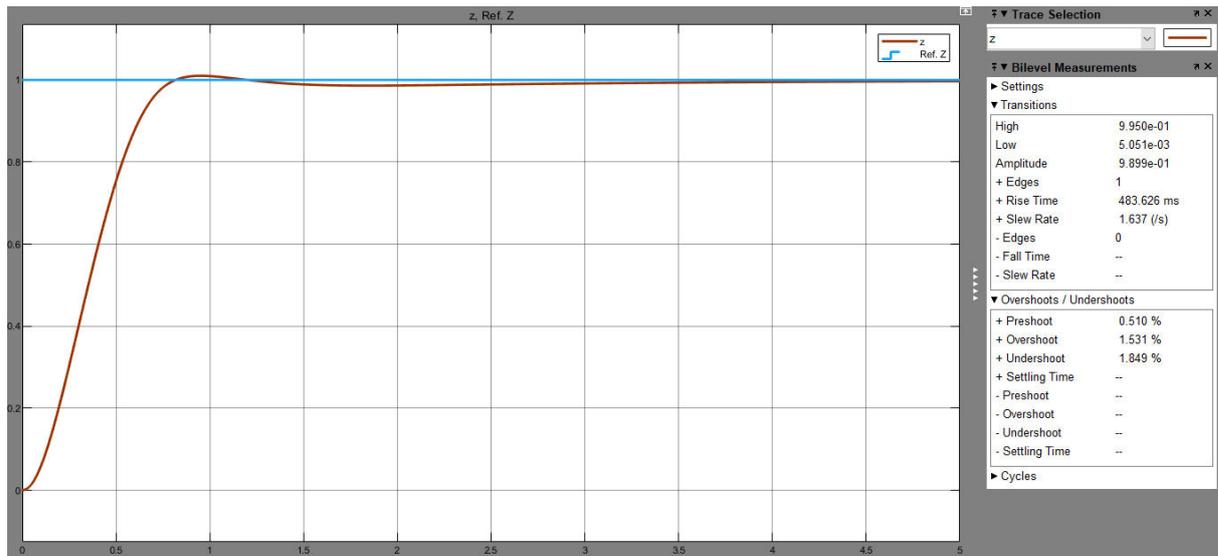


Fonte: O Autor (2025).

4.3.3 Controle de Altitude (Eixo Z)

A Figura 39 ilustra a resposta do controle de altitude. O setpoint é atingido em cerca de 0,8 s, havendo um sobressinal discreto imediatamente compensado. O tempo de acomodação fica abaixo de 2,0 s e o erro de regime permanente é desprezível, evidenciando estabilidade e precisão satisfatórias.

Figura 39 - Resposta em simulação do eixo Z.



Fonte: O Autor (2025).

4.3.4 Conclusão Preliminar

Os resultados simulados demonstram que a metodologia de sintonia automática de controladores PID, baseada em algoritmo genético, atendeu aos requisitos de desempenho no ambiente ideal. Todas as variáveis controladas apresentaram erro de regime permanente nulo, sobressinal mínimo e tempos de acomodação compatíveis com as metas de projeto.

Ressalta-se, entretanto, que o modelo empregado é livre de incertezas e perturbações inerentes à aplicação prática. Assim, é plausível que o desempenho em bancada apresente variações decorrentes de não linearidades, atritos, atrasos de processamento e ruídos de medição. Esses aspectos serão analisados no Capítulo 5, onde se compararão os resultados experimentais com os valores obtidos em simulação.

5 RESULTADOS EXPERIMENTAIS

Após a modelagem e a simulação do sistema de controle do quadcoptero no ambiente MATLAB/Simulink, torna-se necessário validar experimentalmente os resultados obtidos, a fim de verificar a efetividade dos controladores projetados e identificar eventuais limitações do modelo teórico.

Este capítulo apresenta a transição do ambiente simulado para o ambiente real, onde os resultados experimentais são analisados e comparados com os dados simulados, permitindo uma reflexão crítica sobre a fidelidade da modelagem e a robustez dos controladores implementados.

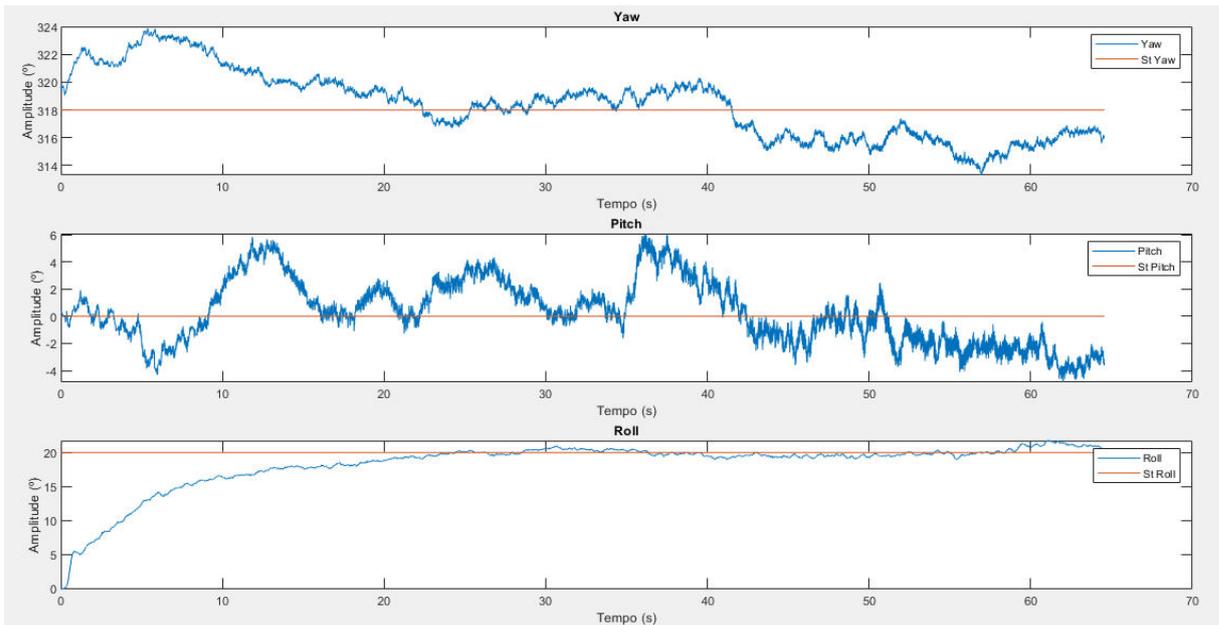
5.1 Resultados experimentais

Os resultados experimentais apresentados nesta seção foram obtidos a partir dos testes realizados nas bancadas de controle de atitude e altitude, descritas anteriormente. As figuras a seguir ilustram o comportamento do quadcoptero em malha fechada, tanto em condições de rastreamento de *setpoints* quanto sob a aplicação de distúrbios externos (os distúrbios em testes foram empurrões, puxadas, cutucadas no drone com o intuito de simular verificar a robustez do controle). É importante ressaltar que a sintonização do controlador PID para os testes em bancada foi realizada manualmente, uma vez que os parâmetros obtidos via algoritmo genético na simulação não estabilizaram a planta no ambiente real.

A Figura 40, mostra o desempenho do controle de *roll*, *pitch* e *yaw*. No teste de *roll* (Figura 40), observa-se que o sistema busca e mantém o *setpoint* de aproximadamente 20 graus. Embora haja flutuações, a capacidade de rastreamento é evidente, também foi possível manter o comportamento superamortecido desejado inicialmente sem grande sobressinal e semelhante as simulações, a grande diferença do ambiente simulado para o teste real é a escala de tempo porém é preciso levar em consideração que a bancada de testes construídas não é uma representação fiel do movimento do drone descrito nas equações utilizadas. A Figura 41, complementa essa análise, evidenciando que, enquanto o *roll* busca seu *setpoint*, o *pitch* se mantém próximo a zero, indicando a capacidade de controle independente dos eixos e como

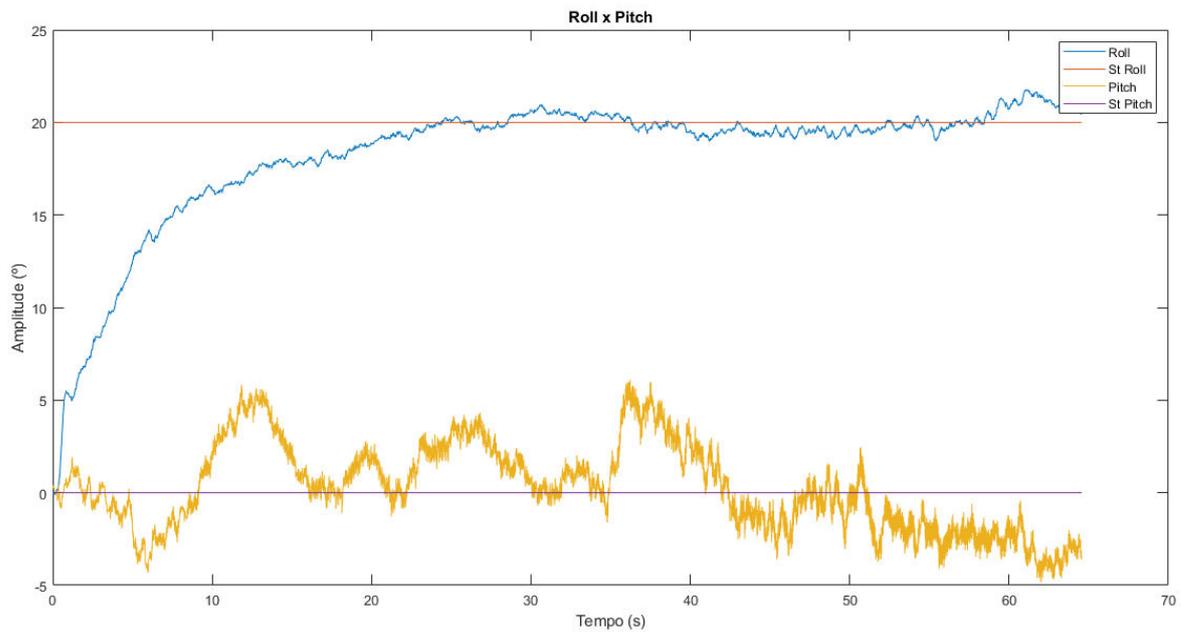
dito no capítulo 3.3.3 aqui é possível ver um pouco a diferença na resposta entre *roll* e *pitch* causada pelo atrito na junta onde a resposta no eixo x aparenta ser mais estável enquanto a no eixo y é mais serrilhada. A Figura 42, oferece uma representação visual do drone inclinado em *roll* durante um teste, confirmando a operação física do sistema.

Figura 40 - Teste de Roll.



Fonte: O Autor (2025).

Figura 41 - Roll x Pitch.



Fonte: O Autor (2025).

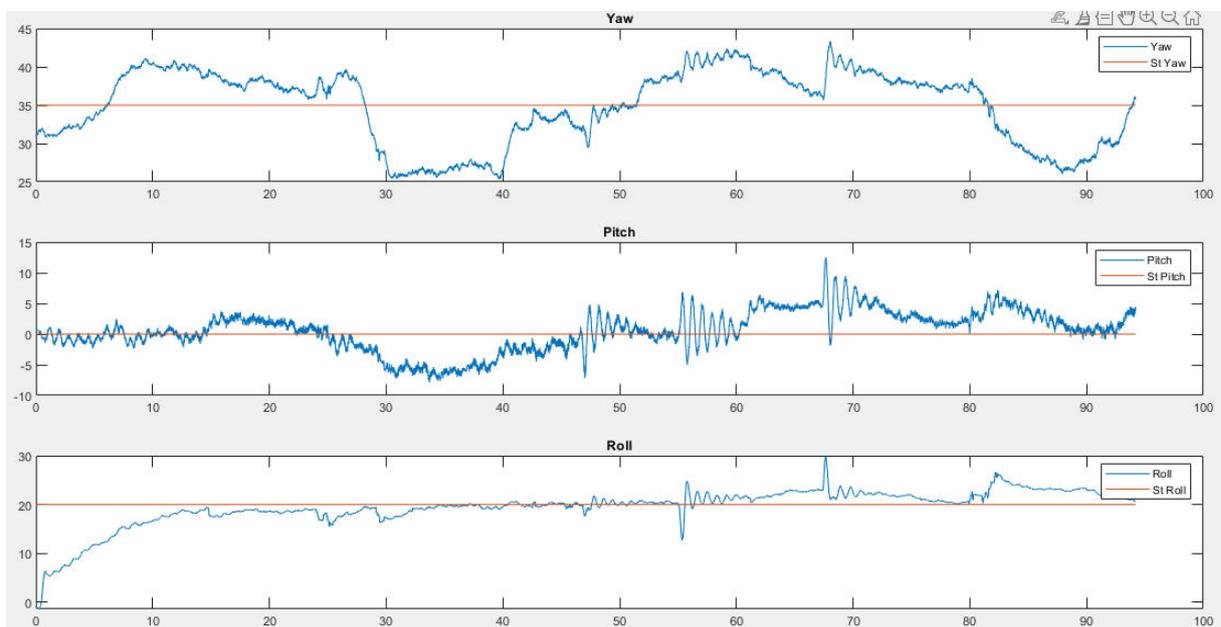
Figura 42 - Roll na bancada.



Fonte: O Autor (2025).

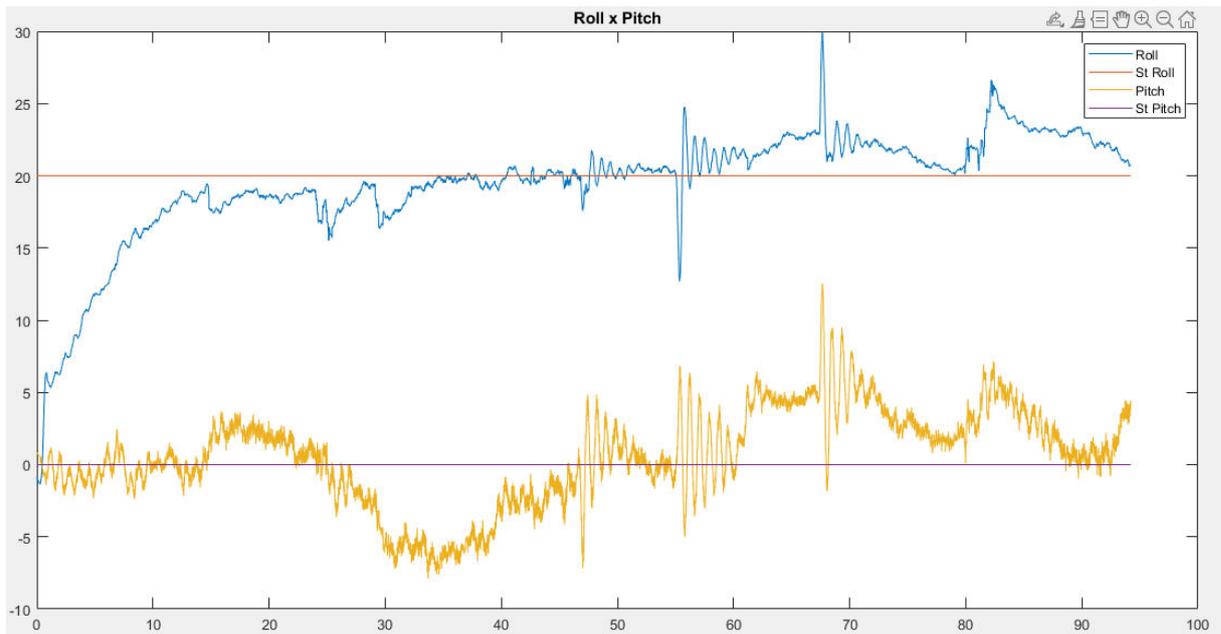
As Figuras 43 e 44 demonstram a robustez do controlador de *roll*, *pitch* e *yaw* diante de perturbações. Nesses gráficos, são visíveis os momentos em que distúrbios foram aplicados, causando picos e oscilações nas respostas. No entanto, o sistema demonstra capacidade de recuperação, retornando aos *setpoints* com rapidez. Isso mostra que o sistema tem alta capacidade de retornar ao ponto de equilíbrio mesmo após grande interferência externas, nesse caso os distúrbios aplicados foram “tapas” e “puxões” no quadcoptero.

Figura 43 - Teste Roll com distúrbios.



Fonte: O Autor (2025).

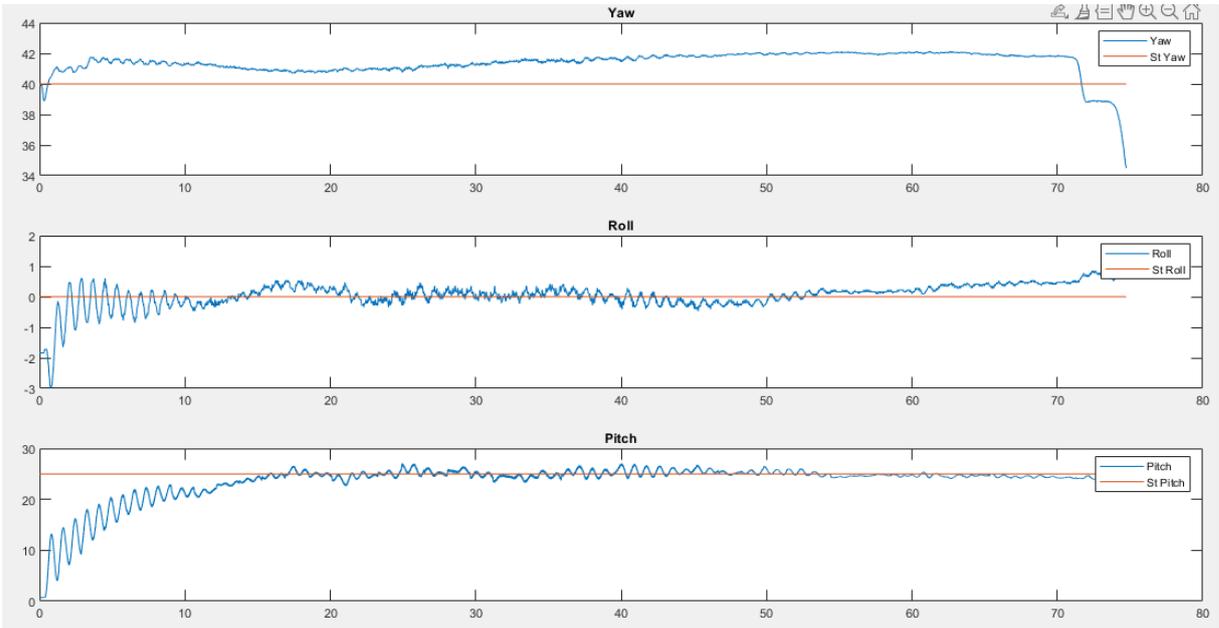
Figura 44 - Roll x Pitch com distúrbios.



Fonte: O Autor (2025).

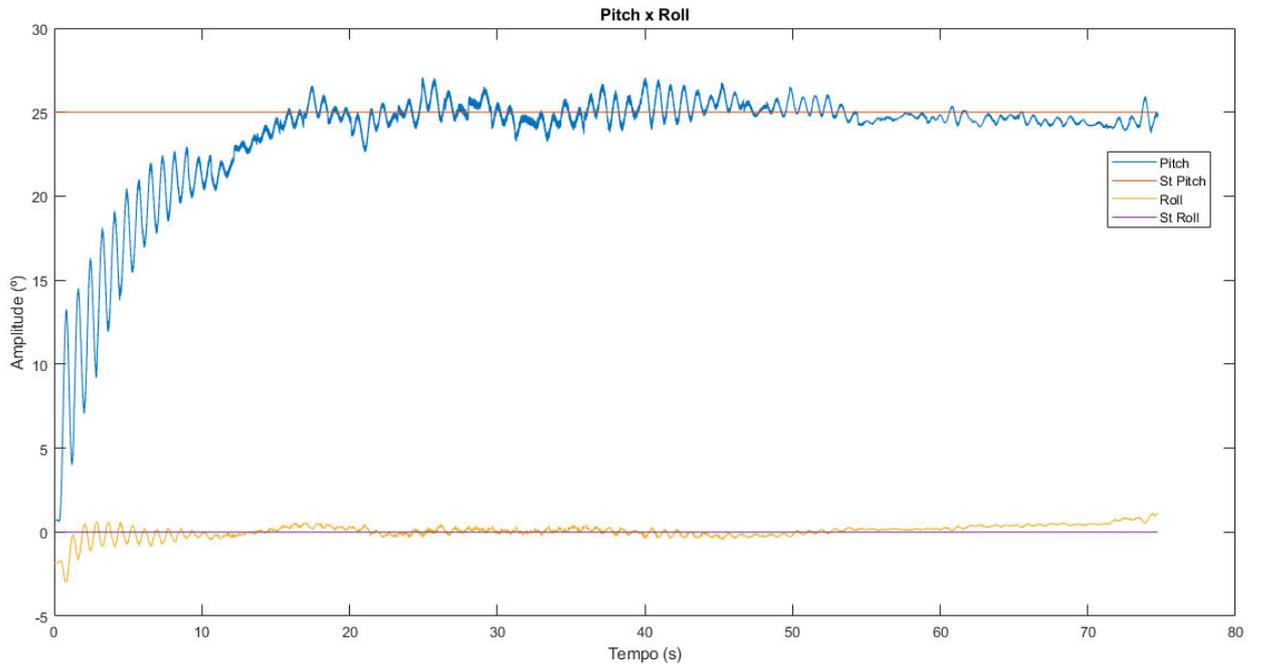
O controle de *pitch* é detalhado nas Figuras 45 e 46. Similarmente ao *roll*, o sistema de *pitch* (Figura 45) atinge e mantém o *setpoint* de 25 graus, porém com oscilações mais acentuadas. Essa maior magnitude nas oscilações no eixo de *pitch* pode ser atribuída à sua maior "liberdade" (menor atrito) na bancada de testes, o que gera um efeito de pêndulo mais pronunciado em comparação com o eixo de rol, como mencionado em sessões passadas. A Figura 46 reforça a independência dos controles, mostrando o *pitch* buscando seu *setpoint* enquanto o *roll* permanece próximo a zero. A Figura 47, corrobora a efetividade do teste em bancada ao ilustrar o drone inclinado em *pitch*. As Figuras 48 e 49, repetem a análise de robustez para o *pitch*, mostrando a recuperação do sistema após a aplicação de distúrbios maiores do que o teste anterior, mesmo com as oscilações mencionadas a estabilidade foi mantida e rapidamente retorna ao setpoint.

Figura 45 - Teste de Pitch.



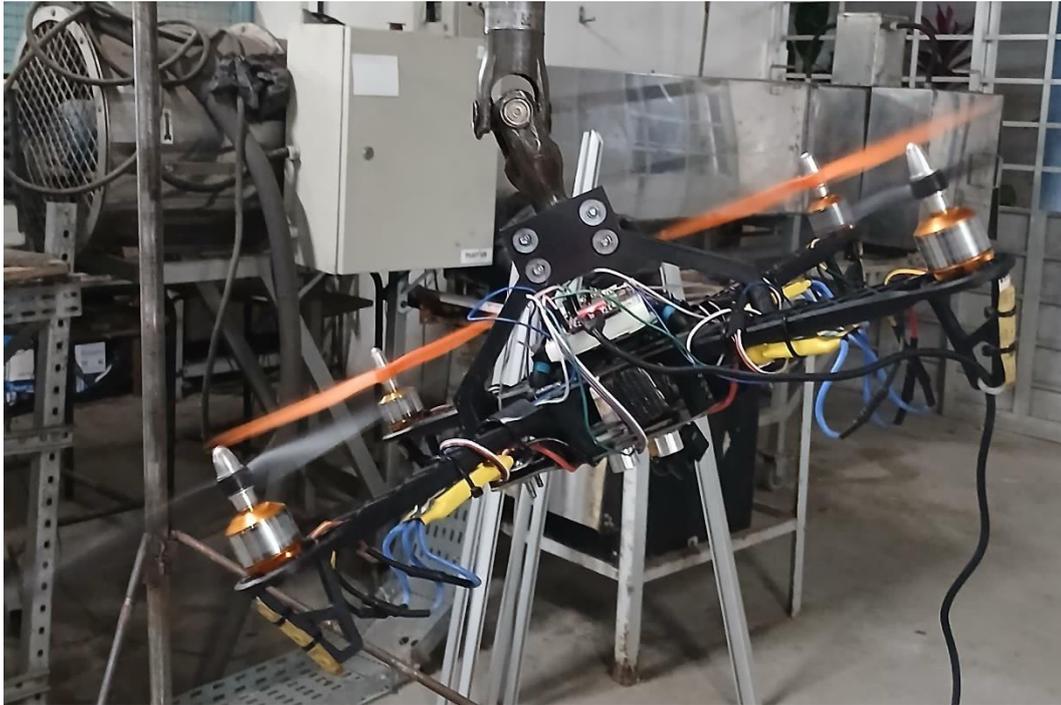
Fonte: O Autor (2025).

Figura 46 - Pitch x Roll.



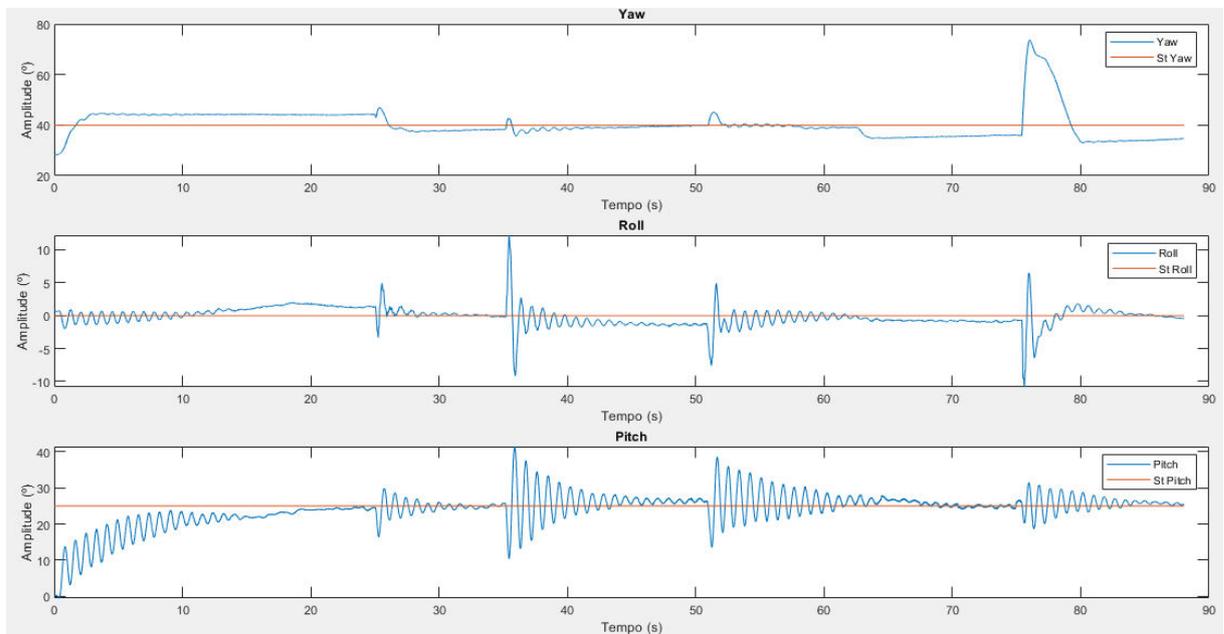
Fonte: O Autor (2025).

Figura 47 - Pitch na bancada.



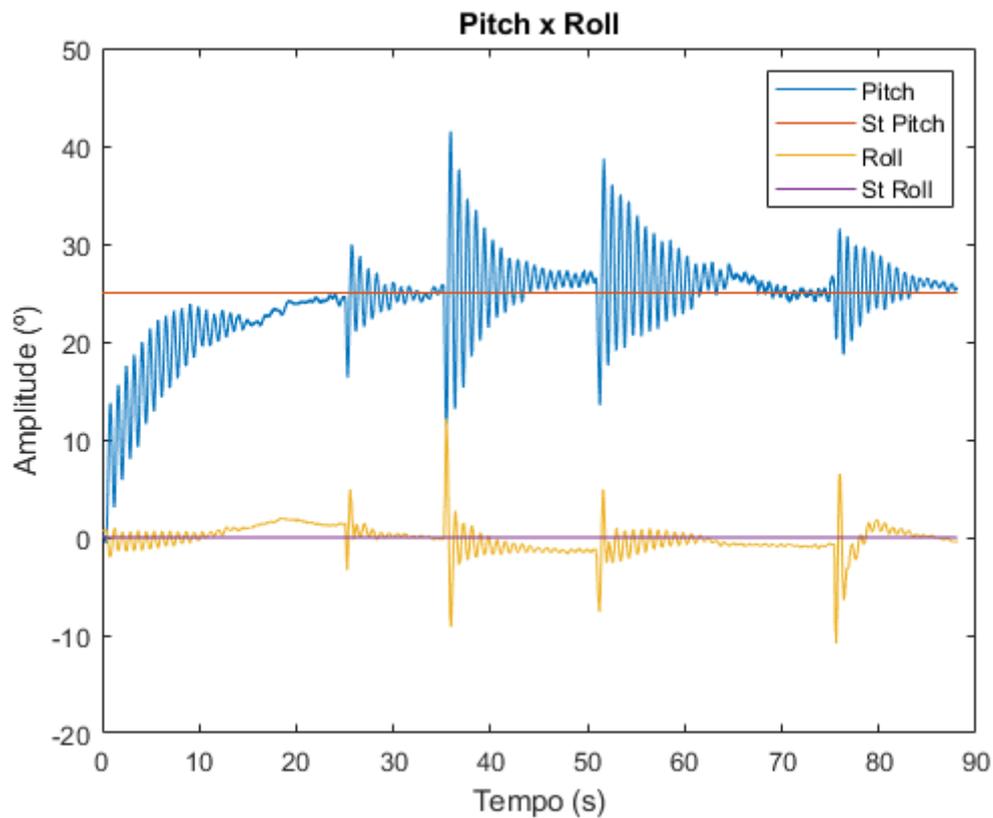
Fonte: O Autor (2025).

Figura 48 - Teste Pitch com distúrbios



Fonte: O Autor (2025).

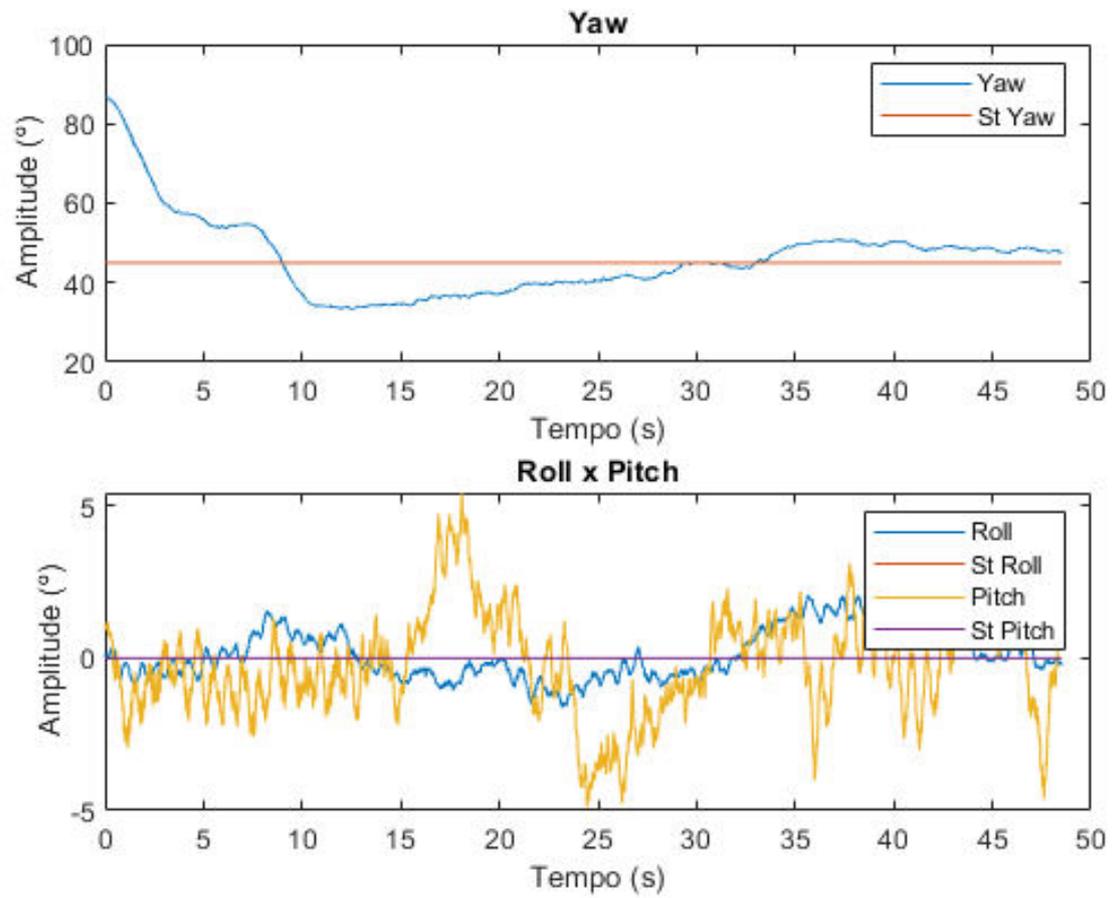
Figura 49 - Pitch x Roll com distúrbios.



Fonte: O Autor (2025).

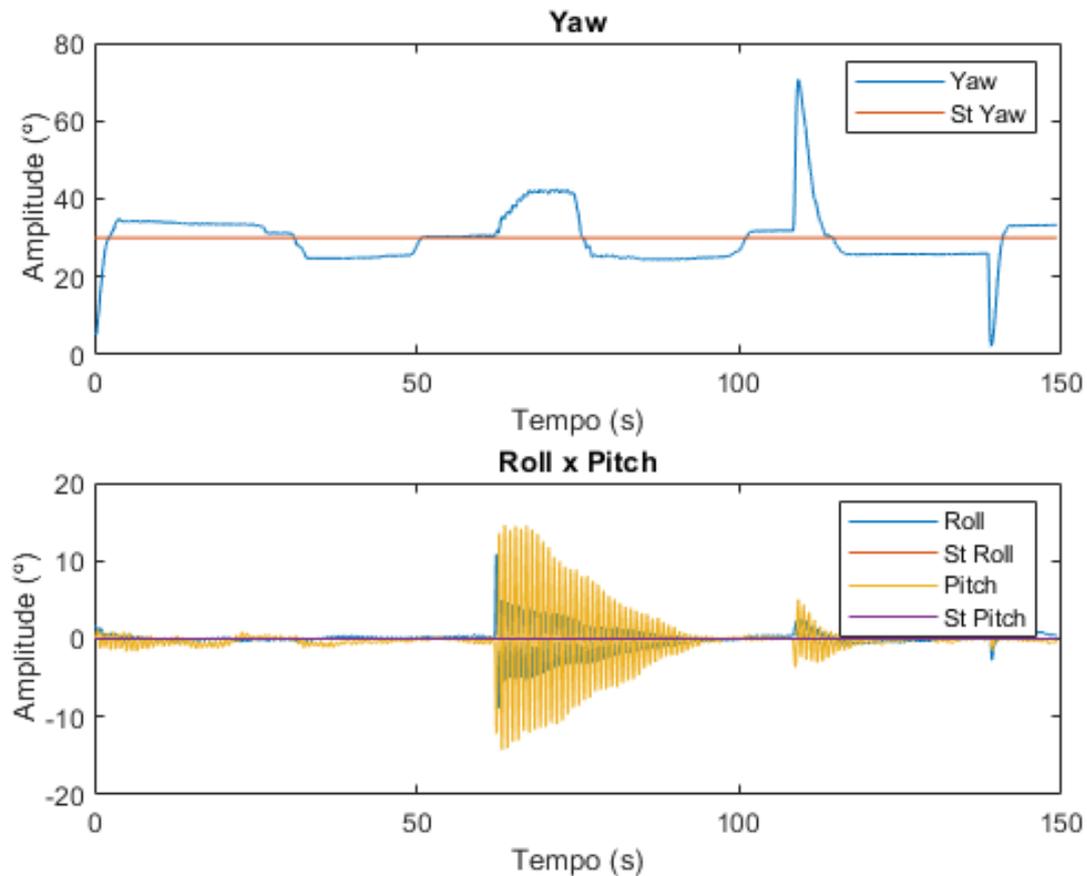
Para o controle de *yaw*, as Figuras 50 e 51, fornecem os resultados de teste similares porem diferente das respostas de *roll* e *pitch* ela não tem uma resposta superamortecido isso foi necessário pois durante outros testes foi notado que o controle de *yaw* durante as manobras pode causar instabilidade, para evitar isso parte da sintonia da malha de *yaw* foi avaliado durante inclinações do drone. A Figura 51 demonstra que, mesmo sob distúrbios, o controle de *yaw* é mantido, com o sistema apresentando a capacidade de retornar ao *setpoint* ou à sua trajetória de controle, vale ressaltar que o primeiro distúrbio não é no eixo de *yaw* direto e por isso o pico em *roll* e *pitch* é muito maior, e nele é possível ver que o controle orientação demora mais a atuar quando os ângulos de rotação não estão próximo aos seus devidos *setpoints*, em contrapartida o segundo distúrbio aplicado no eixo z é rapidamente corrigido.

Figura 50 - Teste de Yaw.



Fonte: O Autor (2025).

Figura 51 - Teste Yaw com distúrbios.

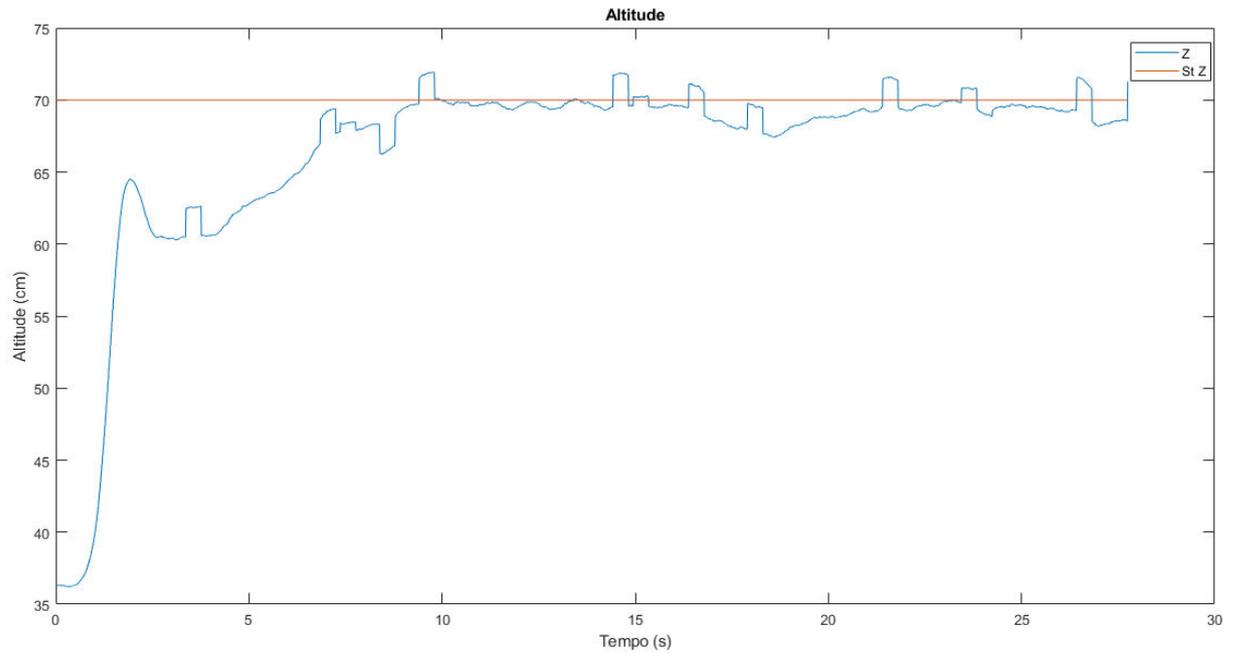


Fonte: O Autor (2025).

Finalmente, a Figura 52, apresenta o desempenho do controle do eixo Z. O gráfico mostra que o quadcoptero é capaz de atingir e manter uma altitude de aproximadamente 70 cm, com pequenas variações. Os resultados indicam um controle de altitude satisfatório, ainda que com pequenas flutuações. Nesse teste é onde o valor de *throttle* mais impacta pois é ele que remove o atraso inicial e ajuda a criar um melhor ajuste fino para o controle de altitude, porém caso ele seja muito alto inicialmente os controles de rotação ficam muito sensíveis dificultando a estabilidade da planta. Outro ponto que é visível no gráfico é que após a rápida subida do drone a uma pequena “queda” parte da culpa desse comportamento está no sensor ultrassônico que possui uma dinâmica lenta em comparação com a IMU, medidas de software foram tomadas para amenizar esse efeito (filtros no termo derivativo do PID) o que teve impactos positivos porém não o suficiente para contrapor eliminar essa

falha em degraus grandes. Ainda sim a precisão desse sensor foi o que tornou capaz de ter uma resposta tão próxima ao setpoint com variações que não ultrapassam 2cm.

Figura 52 - Teste de altitude.



Fonte: O Autor (2025).

6 CONCLUSÃO E PROPOSTAS DE MELHORIAS

Este Trabalho de Conclusão de Curso propôs o desenvolvimento de um sistema embarcado autônomo para o controle de atitude (*roll*, *pitch* e *yaw*) e altitude de um quadcoptero, utilizando um ESP32. Inicialmente, a modelagem e a sintonização dos controladores PID foram realizadas, em ambiente de simulação, empregando um algoritmo genético para otimização dos parâmetros. Contudo, observou-se que os parâmetros PID, ajustados em simulação não se mostraram eficazes no ambiente físico, exigindo uma sintonização manual para os testes experimentais. Este desvio pode ser atribuído às simplificações inerentes ao modelo simulado, que não capturou integralmente as não linearidades, atritos e perturbações presentes no sistema real. No entanto, é importante ressaltar que simulações mais robustas, com a inclusão de dados mais abrangentes e a utilização de equações dinâmicas mais complexas, poderiam representar de forma mais fidedigna o comportamento do modelo físico, aprimorando a transferência dos parâmetros otimizados para a aplicação real.

Apesar dos desafios na transição da simulação para o ambiente físico, o controle do quadcoptero real foi um sucesso. Os resultados experimentais demonstraram que o sistema foi capaz de atingir as metas preestabelecidas para o projeto: o quadcoptero apresentou boa estabilidade, erro em regime permanente próximo a nulo e sobressinal praticamente nulo para os eixos de controle de atitude e altitude, mesmo sob a aplicação de distúrbios. Isso valida a abordagem geral e a eficácia dos controladores implementados no hardware, atendendo aos objetivos principais do trabalho.

6.1 Propostas de Melhoria

Com base nos resultados obtidos e nas lições aprendidas durante o desenvolvimento deste projeto, as seguintes propostas de melhoria são sugeridas para trabalhos futuros:

- **Simulação mais completa:** Desenvolver um modelo de simulação mais abrangente e preciso, incorporando equações dinâmicas mais complexas, caracterização experimental mais detalhada dos torques das hélices (especialmente para o controle de *yaw*), e considerando fatores como atrito,

não linearidades dos atuadores e ruídos nos sensores. Isso permitiria uma sintonização de controladores mais eficaz em ambiente virtual, com maior transferibilidade para o sistema físico.

- **Otimização do código embarcado:** Realizar uma otimização aprofundada do código implementado no ESP32. Isso inclui aprimoramento dos algoritmos de filtragem para lidar com os ruídos observados nos sensores, otimização da frequência de amostragem e processamento dos dados, e revisão da estrutura do firmware para garantir maior eficiência computacional e menor latência.
- **Utilização de hardwares com maior precisão e velocidade:** Explorar a aplicação de sensores com maior precisão e taxa de atualização, como IMUs com menor deriva e altímetros barométricos com maior resolução. Além disso, considerar microcontroladores com maior capacidade de processamento para lidar com algoritmos de controle mais complexos e taxas de *loop* mais elevadas, caso seja necessário para aplicações mais exigentes.
- **Controle de posição como projeto futuro:** Como um projeto futuro e um próximo passo lógico, propõe-se a implementação do controle de posição (eixos X e Y). Isso exigiria a integração de sensores adicionais, como GPS ou sistemas de visão computacional, e o desenvolvimento de algoritmos de controle de nível superior que orquestram as malhas de atitude e altitude para navegar o quadcoptero no espaço tridimensional.

7 REFERÊNCIAS

1. GOMEZ, V. et al. Pareto Optimal PID Tuning for Px4-Based Unmanned Aerial Vehicles by Using a Multi-Objective Particle Swarm Optimization Algorithm, 4 Junho 2020.
2. SABATINO, F. Quadrotor control: modeling, nonlinear control design, and simulation, Stockholm, 25 Junho 2015.
3. BRESCIANI, T. Modelling, Identification and Control of a Quadrotor Helicopter, Lund, Outubro 2008.
4. OGATA, K. **ENGENHARIA DE CONTROLE MODERNO**. 5ª. ed. São Paulo: Pearson Education do Brasil LTDA, 2014.
5. HALIM, H.; ISMAIL, I. Tree physiology optimization on SISO and MIMO PID control tuning, 29 Junho 2018.
6. OPPENHEIM, A. V.; SCHAFER, R. W. **Discrete-Time Signal Processing**. 3ª. ed. Upper Saddle River: Prentice Hall, 2010.
7. COSTA NETO, J. **Github**. Disponível em: <https://github.com/JonasCN/Drone_TCC>.
8. GOLDBERG, D. E. **GENETIC ALGORITHMS in Search, Optimization & Machine Learning**. Alabama: Addison-Wesley Publishing Company, 1989.
9. BOLANDI, H. et al. Attitude Control of a Quadrotor with Optimized PID. **Scientific Research**, Teerã, 9 Dezembro 2012.
10. **Metropole Digital**. Disponível em: <<https://materialpublic.imd.ufrn.br/curso/disciplina/1/52/1/9>>. Acesso em: 04 Junho 2025.
11. SYSTEMS, E., Shanghai, 2025. ISSN ESP32 Series Datasheet Version 4.9. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 17 Junho 2025.
12. FRESK, E. Towards Aerial Robotic Workers, Luleå, 2015.
13. ARAUJO FILHO, P. F. D.; NETO, R. C. Modelagem e Controle Digital de Quadricóptero, Recife, 28 janeiro 2016.
14. BOSCH. Smart Sensor:BNO055. **Bosch Sensortec**. Disponível em: <<https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bno055-ds000.pdf>>. Acesso em: 18 Junho 2025.
15. SEMICONDUCTOR, F. **sparkfun**, 2012. Disponível em: <<https://cdn.sparkfun.com/datasheets/Sensors/Pressure/MPL3115A2.pdf>>. Acesso em: 21 Junho 2025.
16. MORGAN, E. J. HCSR04 Datasheet (PDF) - List of Unclassified Manufacturers. **alldatasheet**, 2014. Disponível em: <<https://www.alldatasheet.com/datasheet-pdf/view/1132204/ETC2/HCSR04.html>>. Acesso em: 19 Junho 2025.

17. SINGH, U. A Comprehensive Guide to Electronic Speed Controllers (ESC). **MECHTEX**. Disponível em: <<https://mechtex.com/blog/a-comprehensive-guide-to-electronic-speed-controllers>>. Acesso em: 18 Junho 2025.
18. XIA, C.-L. **PERMANENT MAGNET BRUSHLESS DC MTOR DRIVES AND CONTROLS**. 1ª. ed. Tianjin: WILEY, 2012.
19. **Mercado Livre**. Disponível em: <https://produto.mercadolivre.com.br/MLB-756649028-frame-flame-whell-hj-450-dji-f450-para-drone-quacopter-_JM>. Acesso em: 09 jun. 2025.
20. COSTA NETO, J. S. Bancada para modelagem do motor. **YouTube**, 2025. Disponível em: <<https://www.youtube.com/watch?v=UlhMeilvV2s>>.
21. COSTA NETO, J. N. Bancada de Altitude. **YouTube**, 2025. Disponível em: <<https://www.youtube.com/watch?v=RKJ16kJUYgU>>.
22. COSTA NETO, J. S. Bancada de teste de atitude do drone. **YouTube**, 2025. Disponível em: <<https://www.youtube.com/watch?v=gEFPOxs0tO8&t=417s>>.
23. HARTMAN, D. et al. **GitHub**, 2014. Disponível em: <<https://github.com/dch33/Quad-Sim>>. Acesso em: 4 Junho 2025.
24. ABDOALNASIR, A.; MIHALIS, P.; ANASTASIOS, D. Fast Tuning of the PID Controller in An HVAC System Using the Big Bang–Big Crunch Algorithm and FPGA Technology, 28 Setembro 2018.
25. BURKANSHAW, L. K. TOWARDS A LOW-COST QUADROTOR RESEARCH, MONTEREY, CALIFORNIA, Março 2010. 86.
26. IAN POUNDS, P. E. Design, Construction and Control of a Large Quadrotor Micro Air Vehicle, Camberra, Setembro 2007.
27. CÁRDENAS R., C. A. et al. Quadrotor Modeling and a PID Control Approach, 12 Abril 2020.
28. KIM, A. R. et al. Dynamic Modeling and Simulation of A Quadcopter with Motor Dynamics, Lawrence, 9 Janeiro 2017.