



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

MATHEUS CORDEIRO VIANA

**CONTROLE DE ATITUDE DE CUBESAT COM ATUADOR DE UM EIXO**

Recife  
2023

MATHEUS CORDEIRO VIANA

**CONTROLE DE ATITUDE DE CUBESAT COM ATUADOR DE UM EIXO**

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 Engenharia de Controle e Automação.

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

Recife  
2023

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

Viana, Matheus Cordeiro.

Controle de atitude de cubesat com atuador de um eixo / Matheus Cordeiro  
Viana. - Recife, 2023.  
74 : il., tab.

Orientador(a): Rafael Cavalcanti Neto

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de  
Pernambuco, Centro de Tecnologia e Geociências, Engenharia de Controle e  
Automação - Bacharelado, 2023.

1. Atuador de um eixo. 2. Controle de Atitude. 3. Controle PD. 4. Cubesat.  
5. Nano satélite. I. Neto, Rafael Cavalcanti . (Orientação). II. Título.

620 CDD (22.ed.)

MATHEUS CORDEIRO VIANA

**CONTROLE DE ATITUDE DE CUBESAT COM ATUADOR DE UM EIXO**

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 Engenharia de Controle e Automação.

Aprovado em: 28/09/2023

**BANCA EXAMINADORA**

---

Prof. Dr. Rafael Cavalcanti Neto  
Universidade Federal de Pernambuco

---

Prof. Dr. Fabrício Bradaschia  
Universidade Federal de Pernambuco

---

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

## **AGRADECIMENTOS**

Primeiramente, quero agradecer a Deus por ter me dado saúde, inteligência e determinação para que eu pudesse chegar onde estou.

Quero agradecer também aos meus pais, Paulo Germânio Viana de Lima e Andréa Cordeiro Viana, e aos meus irmãos, Caio César Cordeiro Viana e Kaique Cordeiro Viana, que são minha base, minha fortaleza e fonte de inspiração e motivação para continuar seguindo em frente e alcançar os meus objetivos.

Agradeço também aos meus colegas de curso, que sempre estiveram comigo nestes últimos anos, sempre me incentivando e me ajudando. Com eles pude dividir muitas histórias, risadas e bons momentos. Sem vocês, essa jornada teria sido infinitamente mais difícil.

Por fim, gostaria de agradecer aos meus professores que me guiaram da melhor forma possível, transmitindo conhecimento e sendo sempre bastante solícitos e dispostos a ajudar. Este trabalho só é possível graças aos aprendizados que obtive com cada um de vocês.

“A menos que modifiquemos nossa forma de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo”. Albert Einstein

## RESUMO

Os *CubeSats* são satélites miniaturizados e de formato cúbico, que apresentam diversas vantagens em termos de custo e versatilidade para determinadas missões espaciais, se comparados com satélites maiores e mais robustos. Tendo isso em vista, vem se tornando cada vez mais importante o aperfeiçoamento das técnicas de controle de apontamento destes nano satélites, uma vez que isso influencia diretamente a sua capacidade de comunicação, observação e monitoramento terrestre. Este trabalho tem como intuito, desenvolver um protótipo de *CubeSat* e dispô-lo em uma estrutura na qual ele fique suspenso e possa rotacionar livremente em torno de um único eixo. Os dados de posição angular do satélite serão obtidos por meio de um sensor giroscópio, permitindo a validação teórica do experimento através do software *MatLab*. Neste trabalho, será abordado o controle de atitude em um único eixo, utilizando um atuador e implementando o controle PD do sistema através de um microcontrolador. Com isso, o protótipo desenvolvido permitirá, a partir de testes e análises, a validação prática do sistema proposto. Em suma, espera-se que os resultados obtidos neste trabalho contribuam para a divulgação da tecnologia aeroespacial, o auxílio no estudo de controle no ambiente acadêmico, especialmente no projeto de extensão “Asa Branca” da Universidade Federal de Pernambuco e em disciplinas envolvendo práticas de controle, permitindo o aperfeiçoamento no desenvolvimento de *CubeSats* e outros tipos de nano satélites.

**Palavras-chave:** Atuador de um eixo. Controle de Atitude. Controle PD. *CubeSat*. Nano Satélite.

## ABSTRACT

CubeSats are miniaturized cubic-shaped satellites that offer various advantages in terms of cost and versatility for specific space missions, when compared to larger and more robust satellites. Considering this, the enhancement of pointing control techniques for these nanosatellites has become increasingly important, as it directly influences their capacity for communication, observation, and terrestrial monitoring. The aim of this study is to develop a CubeSat prototype and place it within a structure that allows it to be suspended and freely rotate around a single axis. Satellite angular position data will be acquired through a gyroscope sensor, enabling theoretical validation of the experiment using the Matlab software. This work addresses attitude control along a single axis, utilizing an actuator and implementing PID control of the system via a microcontroller. Consequently, the developed prototype will allow practical validation of the proposed system through tests and analyses. In essence, the results obtained in this study are anticipated to contribute to the dissemination of aerospace technology, aid in the academic study of control, especially in the 'Asa Branca' extension project at the Federal University of Pernambuco and in courses involving control practice, allowing for the improvement in the development of CubeSats and other types of nano satellites.

**Keywords:** Attitude Control. CubeSat. Nano Satellite. PID Control. Single-Axis Actuator.



## LISTA DE FIGURAS

|                                                                                                                                                                      |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 – Exemplo de <i>CubeSat</i> modelo 1U. ....                                                                                                                 | 19 |
| Figura 2 – Lançamento de <i>CubeSats</i> a partir da Estação Espacial Internacional. ....                                                                            | 20 |
| Figura 3 – Componentes e módulos que compõem um <i>CubeSat</i> . ....                                                                                                | 22 |
| Figura 4 – Representação dos eixos <i>Roll</i> , <i>Pitch</i> e <i>Yaw</i> em um avião. ....                                                                         | 23 |
| Figura 5 – Representação dos eixos <i>Roll</i> , <i>Pitch</i> e <i>Yaw</i> em um <i>CubeSat</i> . ....                                                               | 24 |
| Figura 6 – Exemplo de roda de reação responsável por movimentar o <i>CubeSat</i> . ....                                                                              | 25 |
| Figura 7 - Digramas de corpo livre da roda de reação e do <i>CubeSat</i> . ....                                                                                      | 26 |
| Figura 8 – Arquitetura elétrica do <i>CubeSat</i> . ....                                                                                                             | 30 |
| Figura 9 – Esquema de controle em malha fechada do <i>CubeSat</i> . ....                                                                                             | 30 |
| Figura 10 – Chassis do <i>CubeSat</i> presente no Módulo estrutural. ....                                                                                            | 32 |
| Figura 11 – Baterias de íon de lítio em série que compõem o Módulo de energia. ....                                                                                  | 33 |
| Figura 12 – Módulo Bluetooth HC-06 presente no Módulo de Comunicação. ....                                                                                           | 34 |
| Figura 13 – Giroscópio e acelerômetro MPU6050 responsável por fornecer a orientação do <i>CubeSat</i> . ....                                                         | 35 |
| Figura 14 – Motor CC <i>Coreless</i> 1020A utilizado no módulo de controle. ....                                                                                     | 36 |
| Figura 15 – Roda de reação responsável pelo movimento do <i>CubeSat</i> . ....                                                                                       | 36 |
| Figura 16 – Microcontrolador Arduino Uno utilizado no módulo computacional. ....                                                                                     | 37 |
| Figura 17 – <i>Shield</i> para Arduino contendo <i>driver</i> com Ponte H L293D. ....                                                                                | 38 |
| Figura 18 – Estrutura de testes para validação experimental do funcionamento do controle de atitude do <i>CubeSat</i> . ....                                         | 39 |
| Figura 19 – Disposição dos componentes do <i>CubeSat</i> baseado na arquitetura. ....                                                                                | 41 |
| Figura 20 – Montagem do <i>CubeSat</i> feita em três prateleiras. ....                                                                                               | 41 |
| Figura 21 – Interface Gráfica. ....                                                                                                                                  | 42 |
| Figura 22 - Exemplo de curva de Posição Angular obtida a partir do Software de Validação. ....                                                                       | 43 |
| Figura 23 – Fluxograma representativo da lógica de estabelecimento da comunicação <i>Bluetooth</i> com o microcontrolador e inicialização da Interface Gráfica. .... | 44 |
| Figura 24 – Fluxograma representativo da lógica de envio de <i>SetPoint</i> para o microcontrolador. ....                                                            | 44 |

|                                                                                                                                            |    |
|--------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 25 – Fluxograma representativo da lógica responsável por ler dados microcontrolador, gerar arquivo CSV e plotar o gráfico.....      | 45 |
| Figura 26 – Curva de resposta considerando aplicação de PWM com ciclo de trabalho de 50%. ....                                             | 48 |
| Figura 27 – Curva de resposta considerando aplicação de PWM com ciclo de trabalho de 37,5%. ....                                           | 48 |
| Figura 28 – Curva de resposta considerando aplicação de PWM com ciclo de trabalho de 25%. ....                                             | 48 |
| Figura 29 – Comparação entre respostas ao degrau de ciclo de trabalho de 50% do PWM para planta modelada e para a planta experimental..... | 49 |
| Figura 30 – Validação do modelo apresentado na Equação (3) para o cenário avaliado no teste 4. ....                                        | 50 |
| Figura 31 – Validação do modelo apresentado na Equação (3) para o cenário avaliado no teste 5. ....                                        | 51 |
| Figura 32 – Diagrama de blocos do sistema em malha fechada com controlador unitário. ....                                                  | 53 |
| Figura 33 - Resposta a um degrau (mudança da referência de 0° para 10°) para o sistema em malha fechada com controlador unitário. ....     | 54 |
| Figura 34 – LGR quando usando um controlador do tipo P (tempo contínuo) – Ensaio 1. ....                                                   | 54 |
| Figura 35 – LGR quando usando um controlador do tipo P (tempo discreto) – Ensaio 1. ....                                                   | 55 |
| Figura 36 – LGR quando usando um controlador do tipo PD (tempo discreto) – Ensaio 1. ....                                                  | 55 |
| Figura 37 – Resposta ao degrau para o sistema em malha fechada considerando um controlador tipo PD – Ensaio 2.....                         | 56 |
| Figura 38 – Diagrama do sistema em malha fechada, representado no <i>Simulink</i> .....                                                    | 56 |
| Figura 39 – Resposta a um degrau (mudança da referência de 0° para 10°) para o sistema em malha fechada com controlador PD. ....           | 57 |
| Figura 40 – Implementação do Controlador no Microcontrolador.....                                                                          | 58 |
| Figura 41 – Curva de resposta do <i>CubeSat</i> a um <i>SetPoint</i> de -30 graus.....                                                     | 59 |
| Figura 42 – Dados da Interface Gráfica para um <i>SetPoint</i> de -30 graus no <i>CubeSat</i> . ....                                       | 60 |
| Figura 43 – Curva de resposta do <i>CubeSat</i> a um <i>SetPoint</i> de 30 graus.....                                                      | 60 |

|                                                                                                                       |    |
|-----------------------------------------------------------------------------------------------------------------------|----|
| Figura 44 – Dados da Interface Gráfica para um <i>SetPoint</i> de 30 graus no <i>CubeSat</i> .<br>.....               | 61 |
| Figura 45 – Curva de resposta do <i>CubeSat</i> a um <i>SetPoint</i> de -60 graus.....                                | 61 |
| Figura 46 – Dados da Interface Gráfica para um <i>SetPoint</i> de -60 graus no <i>CubeSat</i> .<br>.....              | 62 |
| Figura 47 – Curva de resposta do <i>CubeSat</i> a um <i>SetPoint</i> de 60 graus.....                                 | 62 |
| Figura 48 – Dados da Interface Gráfica para um <i>SetPoint</i> de 60 graus no <i>CubeSat</i> .<br>.....               | 63 |
| Figura 49 – Curva de resposta do <i>CubeSat</i> a um <i>SetPoint</i> de -90 graus.....                                | 63 |
| Figura 50 – Dados da Interface Gráfica para um <i>SetPoint</i> de -90 graus no <i>CubeSat</i> .<br>.....              | 64 |
| Figura 51 – Curva de resposta do <i>CubeSat</i> a um <i>SetPoint</i> de 90 graus.....                                 | 64 |
| Figura 52 – Dados da Interface Gráfica para um <i>SetPoint</i> de 90 graus no <i>CubeSat</i> .<br>.....               | 65 |
| Figura 53 – Comportamento do sistema quando sujeito a uma perturbação. O sistema deve operar em regime em -90°. ..... | 67 |
| Figura 54 - Dados da Interface Gráfica do sistema para o teste de Perturbação.....                                    | 67 |

## LISTA DE TABELAS

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| Tabela 1 – Requisitos de controle definidos neste TCC.....                            | 53 |
| Tabela 2 - <i>Links</i> para vídeos do funcionamento dos 6 testes experimentais. .... | 66 |
| Tabela 3 – <i>Links</i> para vídeo do funcionamento do teste de perturbação. ....     | 68 |

## LISTA DE ABREVIATURAS E SIGLAS

|         |                                                     |
|---------|-----------------------------------------------------|
| 1U      | Uma Unidade                                         |
| 2U      | Duas Unidades                                       |
| 3D      | Três Dimensões                                      |
| 6U      | Seis Unidades                                       |
| AD      | Analógico-Digital                                   |
| ADCS    | <i>Attitude Determination and Control Subsystem</i> |
| CC      | Corrente Contínua                                   |
| CSV     | <i>Comma-Separated Values</i>                       |
| D       | Derivativo                                          |
| DA      | Digital-Analógico                                   |
| LGR     | Lugar Geométrico das Raízes                         |
| LEO     | <i>Low Earth Orbit</i>                              |
| IHM     | Interface Homem Máquina                             |
| HC-06   | <i>Host Controller 06</i>                           |
| I       | Integral                                            |
| I2C     | <i>Inter-Integrated Circuit</i>                     |
| IDE     | <i>Integrated Development Enviroment</i>            |
| INPE    | Instituto Nacional de Pesquisas Espaciais           |
| MIMO    | <i>Multiple Input Multiple Output</i>               |
| MPU6050 | <i>Microprocessor Unit 6050</i>                     |
| P       | Proporcional                                        |
| PD      | Proporcional Derivativo                             |
| PID     | Proporcional Integral Derivativo                    |
| PLA     | Ácido Poliláctico                                   |
| PVC     | Policloreto de Vinila                               |
| PWM     | <i>Pulse Width Modulation</i>                       |
| RS-232  | <i>Recommended Standard 232</i>                     |
| VCC     | Tensão Contínua                                     |

## LISTA DE SÍMBOLOS

|     |                     |
|-----|---------------------|
| A   | Àmpere              |
| Kg  | Quilograma          |
| mA  | Miliampere          |
| V   | Volt                |
| °C  | Graus Celsius       |
| GHz | Giga Hertz          |
| Cm  | Centímetro          |
| rpm | Rotações Por Minuto |
| ms  | Milissegundos       |

## SUMÁRIO

|          |                                                                |           |
|----------|----------------------------------------------------------------|-----------|
| <b>1</b> | <b>INTRODUÇÃO .....</b>                                        | <b>14</b> |
| 1.1      | JUSTIFICATIVA.....                                             | 14        |
| 1.2      | OBJETIVOS .....                                                | 15        |
| 1.3      | METODOLOGIA.....                                               | 15        |
| 1.4      | ORGANIZAÇÃO DO TRABALHO.....                                   | 16        |
| <b>2</b> | <b>FUNDAMENTAÇÃO TEÓRICA .....</b>                             | <b>18</b> |
| 2.1      | O <i>CUBESAT</i> .....                                         | 18        |
| 2.2      | MÓDULOS DE FUNCIONAMENTO DE UM <i>CUBESAT</i> .....            | 20        |
| 2.3      | CONTROLE DE ATITUDE .....                                      | 22        |
| 2.3.1    | Atitude .....                                                  | 23        |
| 2.3.2    | Controle de Estabilidade.....                                  | 24        |
| 2.3.3    | Controle de Apontamento.....                                   | 25        |
| <b>3</b> | <b>DESENVOLVIMENTO DE PLATAFORMA DE TESTES .....</b>           | <b>28</b> |
| 3.1      | ENGENHARIA DE REQUISITOS .....                                 | 28        |
| 3.2      | ARQUITETURA DE MONTAGEM DO <i>CUBESAT</i> .....                | 29        |
| 3.2.1    | Módulo Estrutural.....                                         | 31        |
| 3.2.2    | Módulo de Energia.....                                         | 32        |
| 3.2.3    | Módulo de Comunicação .....                                    | 33        |
| 3.2.4    | Módulo de Controle .....                                       | 34        |
| 3.2.5    | Módulo de Computação.....                                      | 37        |
| 3.3      | ESTRUTURA DE TESTES .....                                      | 39        |
| 3.4      | ESTRUTURA MECÂNICA DO <i>CUBESAT</i> .....                     | 40        |
| 3.5      | <i>SOFTWARE</i> DE VALIDAÇÃO .....                             | 42        |
| <b>4</b> | <b>PROJETO E VALIDAÇÃO DO SISTEMA DE CONTROLE.....</b>         | <b>46</b> |
| 4.1      | MODELAGEM MATEMÁTICA DA PLANTA .....                           | 46        |
| 4.2      | VALIDAÇÃO DA PLANTA .....                                      | 50        |
| 4.3      | CONTROLE DA PLANTA.....                                        | 52        |
| 4.3.1    | Definição dos Requisitos de Controle.....                      | 52        |
| 4.3.2    | Projeto de Controle.....                                       | 53        |
| 4.4      | VALIDAÇÃO EXPERIMENTAL UTILIZANDO A PLATAFORMA DE TESTES<br>58 |           |
| 4.5      | AVALIAÇÃO DE DIFERENTES CENÁRIOS .....                         | 66        |
| 4.6      | DIFICULDADES ENFRENTADAS.....                                  | 68        |
| <b>5</b> | <b>CONCLUSÕES E PROPOSTAS DE CONTINUIDADE .....</b>            | <b>70</b> |
|          | <b>REFERÊNCIAS .....</b>                                       | <b>71</b> |

## 1 INTRODUÇÃO

Com o grande avanço tecnológico do setor aeroespacial nos últimos anos, a exploração espacial tem atingido proporções antes nunca vistas. Esta crescente se deve, dentre outras coisas, ao desenvolvimento dos nano satélites que, como o próprio nome sugere, são satélites em miniatura que têm ganhado cada vez mais notoriedade devido a sua grande versatilidade, capacidade de realizar diversos tipos de missões, assim como o seu baixo custo de produção (FRACKIEWICZ, 2023).

Dentre os nano satélites, uma categoria de grande destaque são os *CubeSats*, pequenos satélites em formato de cubo com diversas aplicabilidades que vão desde pesquisas científicas até aplicações comerciais, a exemplo, no ramo das telecomunicações, defesa e proteção de dados e sensoriamento remoto da Terra (CASTELLAN, 2022). Entretanto, um dos principais desafios enfrentados pelos *CubeSats* é o controle do seu apontamento de forma precisa e eficiente, de tal modo que ele possa posicionar-se e orientar-se de maneira adequada no espaço. Em geral, esses pequenos satélites operam em baixas altitudes, onde estão sujeitos a arrastos atmosféricos e perturbações gravitacionais, tornando o seu controle ainda mais desafiador (OBSAT, 2022).

Uma abordagem simplificada, porém, bastante funcional para implementação do controle de apontamento de um *CubeSat*, é a utilização de um atuador de um eixo desenvolvido exclusivamente para este tipo de satélite (SOUZA, 2002). Essa forma simplificada de realizar o controle de apontamento é viável devido ao fato de o *CubeSat* ser um satélite de dimensões bastante reduzidas, além de que ela apresenta vantagens com relação a soluções baseadas em três atuadores, tais como a redução da complexidade, menor consumo de energia, facilidade de implementação, assim como a adequação as limitações de espaço do *CubeSat*.

### 1.1 Justificativa

O controle de atitude de satélites tem papel fundamental no sucesso de missões aeroespaciais. Devido a diversos fatores como o custo-benefício, por exemplo, os *CubeSats* têm despertado cada vez mais interesse da comunidade científica



(CASTELLAN, 2022). Com esta crescente, é de extrema importância desenvolver técnicas que permitam controlar de forma precisa e eficaz a orientação destes nano satélites.

Do ponto de vista do sistema de controle, o controle de atitude com atuador de um eixo é bastante desafiador, sendo necessário implementar algoritmos de controle que atendam aos requisitos e restrições específicas do sistema, como a massa do *CubeSat*, a alimentação do motor e dos controladores, o volume e a inércia da estrutura, assim como as características de erro, máximo sobressinal e velocidade de atuação do sistema. Além disso, é fundamental garantir que o sistema funcione de forma estável, com controle de apontamento e velocidade precisos e minimizando perturbações e/ou oscilações indesejadas durante a realização das manobras espaciais.

É esperado que este trabalho possa contribuir para a popularização da tecnologia aeroespacial, trazendo para a graduação algumas temáticas no estudo desta área. Este projeto será capaz de trazer consigo uma grande contribuição acadêmica, uma vez que a partir dele, alunos poderão estudar e aplicar técnicas de controle na prática, de uma maneira didática e intuitiva, assim como servir de base para elaboração de projetos mais avançados, como o controle de atitude de 3 eixos (CGEE, 2018).

## **1.2 Objetivos**

Este trabalho tem como principais objetivos, estudar e desenvolver um sistema que realize o controle de atitude de um *CubeSat* através de um atuador de um eixo, além de realizar a montagem de um protótipo e validação experimental do controle de atitude avaliado.

## **1.3 Metodologia**

O trabalho foi desenvolvido obedecendo a seguinte metodologia:

- Avaliar uma arquitetura viável para a montagem física de uma plataforma de testes;

- Montar um protótipo de *CubeSat*;
- Desenvolver um software de validação capaz de enviar comandos e receber informações do nano satélite;
- Modelar matematicamente a dinâmica da planta que representa o sistema;
- Avaliar os requisitos de controle de apontamento do *CubeSat*;
- Modelar via simulação, um controlador para o protótipo;
- Implementar o controle simulado na planta física;
- Validar o controle do *CubeSat* experimentalmente através da plataforma de testes.

#### 1.4 Organização do Trabalho

Este trabalho é dividido em 5 Capítulos, e cada um deles está listado e resumidamente explicado a seguir:

- **1 - INTRODUÇÃO:** a introdução apresenta um panorama geral sobre o que são os *CubeSats*, falando um pouco das suas aplicações e os desafios apresentados para que seja feito o seu controle de apontamento.
- **2 - FUNDAMENTAÇÃO TEÓRICA:** neste capítulo são apresentados os conceitos considerados fundamentais para o entendimento do trabalho. Neste capítulo é explorado mais a fundo o conceito de *CubeSat*, tratando de temas como tamanhos, peso, vantagens e desvantagens deste tipo de satélite. Também são abordados os módulos de funcionamento que compõem um *CubeSat*, assim como o conceito de controle de atitude, tópico deste capítulo que aborda separadamente os conceitos de atitude, controle de estabilidade e controle de apontamento.
- **3 - DESENVOLVIMENTO DE PLATAFORMA DE TESTES:** este capítulo apresenta como e o que foi utilizado para o desenvolvimento do protótipo de *CubeSat*, a começar pelos requisitos de projeto, até os materiais utilizados para confecção de toda a estrutura da planta física, mostrando como foi feita a estrutura mecânica, os módulos de funcionamento, a estrutura de testes e o software que foi utilizado para validação dos testes.

- **4 - PROJETO E VALIDAÇÃO DO SISTEMA DE CONTROLE:** neste capítulo é feito todo o desenvolvimento teórico e são apresentados os resultados obtidos com plataforma desenvolvida no Capítulo 3.
- **5 - CONCLUSÕES E PROPOSTAS DE CONTINUIDADE:** este capítulo expõe as conclusões a respeito do projeto, aqui é feita uma análise crítica a respeito do sistema de controle desenvolvido para planta física, concluindo se ele foi ou não foi satisfatório com relação aos requisitos de projeto.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos básicos fundamentais para o entendimento posterior de como o controle de atitude de um *CubeSat* é desenvolvido.

Ao final deste capítulo, espera-se que o leitor possua uma visão crítica e contextualizada do assunto. Aqui, são abordados os principais fundamentos de um *CubeSat*, desde a sua definição e aplicações, até a sua construção, composta por vários módulos de funcionamento, explicando de forma detalhada a sua estrutura. Além disso, será explorado o conceito de controle de atitude, abordando separadamente as definições de atitude, controle de apontamento e controle de estabilidade.

### 2.1 O *CubeSat*

Os *CubeSats* são satélites em miniatura classificados como nano satélites (satélites cuja massa está entre 1 e 10 kg). O menor tipo de *CubeSat* possui um formato cúbico, com arestas de 10 cm e uma massa de aproximadamente 1,33 kg por unidade. Estas especificações correspondem a unidade padrão deste tipo de satélite, denominada de 1U (exemplificado na Figura 1). A depender das aplicações e objetivos para o qual está sendo construído, o *CubeSat* pode variar de tamanho e possuir uma maior capacidade. Alguns exemplos de modelos que podem ser utilizados, com tamanhos múltiplos do 1U, são o 2U, o 3U e o 6U (LOFF, 2018). Apesar de possuírem um tamanho reduzido, estes satélites podem realizar missões tão complexas quanto satélites mais robustos e de tamanho maior.

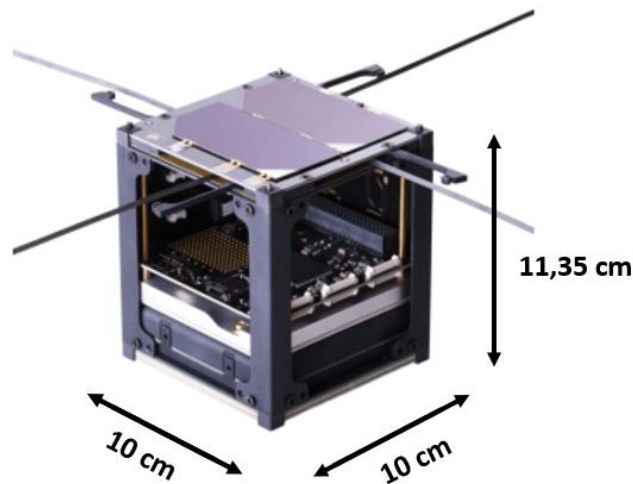
A ideia de *CubeSat* surgiu em 1999, na Universidade da Califórnia, pelos professores *Bob Twiggs* e *Jordi Puig-Suari*. A intenção deles era propor um modelo de satélite que pudesse ser desenvolvido em ambiente acadêmico, dando aos estudantes a oportunidade de experienciar o processo de projeto, construção e operação de um satélite semelhante aos convencionais (CASTELLAN, 2022).

Dentre as principais vantagens de um *CubeSat*, estão o baixo custo de desenvolvimento e a facilidade de acesso ao espaço, pois por ser compacto e leve, o

seu lançamento é mais simples e mais barato. Isto fez com que este satélite atingisse grandes proporções, que vão desde missões espaciais até aplicações militares e civis.

Contudo, pelo seu tamanho reduzido, este nano satélite apresenta desvantagens. Entre as principais restrições destacam-se: grande limitação na carga útil e de comunicação; e uma vida útil limitada (FRACKIEWICZ, 2023).

Figura 1 – Exemplo de *CubeSat* modelo 1U.



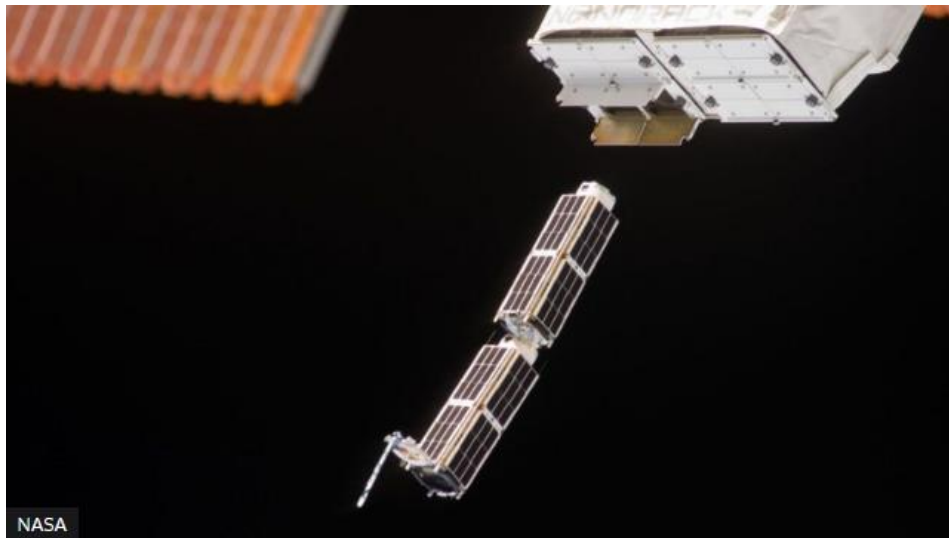
Fonte: (AMSAT-UK, 2018).

Por serem satélites de dimensões bastante reduzidas, os *CubeSats* são lançados e colocados em órbita como carga secundária de um ou mais satélites de maior porte que constituem a carga primária. Por essa razão, os *CubeSats* são obrigados a aceitar a órbita determinada pelo satélite lançador (carga primária). Quando um *CubeSat* está para ser lançado, a órbita do satélite lançador deve ser considerada, havendo um compromisso entre aceitar uma órbita não ideal ou esperar até que haja um lançamento com uma órbita mais adequada, algo que muitas vezes pode demorar bastante tempo (CGEE, 2017).

A grande maioria dos *CubeSats* são lançados em órbita baixa, também conhecida pela sigla LEO, abreviação de *Low Earth Orbit*. Esta órbita varia de 160 a 2000 km de altitude e o período orbital pode variar de 90 a 115 minutos para esses casos (CGEE, 2017). Parâmetros orbitais como a inclinação e a altitude do nano satélite em relação a Terra, influenciam na duração de recebimento de luz solar,

impactando diretamente no carregamento de seus painéis solares por órbita (ENNIS, 2017).

Figura 2 – Lançamento de *CubeSats* a partir da Estação Espacial Internacional.



Fonte: (HARFORD, 2019).

## 2.2 Módulos de funcionamento de um *CubeSat*

Visando o adequado funcionamento de um *CubeSat*, são necessários vários módulos, cada um desempenhando uma função específica dentro do nano satélite. Em geral este tipo de satélite possui 7 módulos, mas este número pode variar a depender das aplicações específicas destinadas a ele. A seguir, uma lista com uma breve descrição de cada um dos módulos de um *CubeSat*.

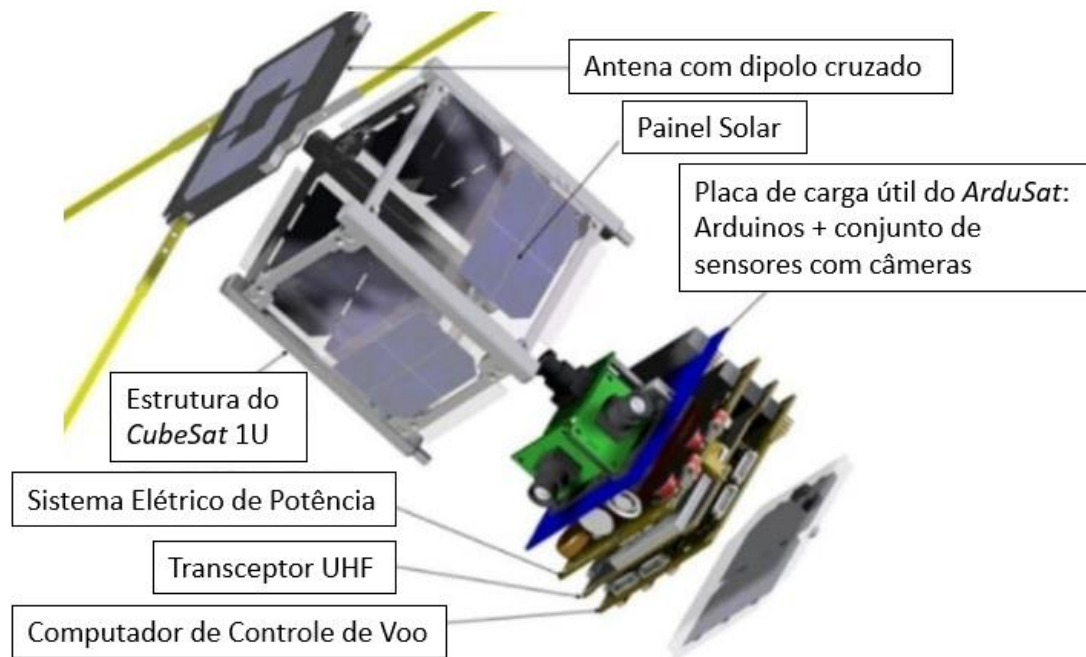
- **Módulo Estrutural:** Este módulo é responsável pela proteção e suporte dos componentes presentes no satélite. Possui um formato cúbico ou de prisma retangular (depende do padrão de tamanho utilizado) e é construído com materiais leves, porém resistentes. Em outras palavras, o módulo estrutural é a “lataria” do *CubeSat*.
- **Módulo de Energia:** Este módulo é o responsável pela alimentação dos componentes eletrônicos presentes no nano satélite, em geral ele é composto por painéis solares (responsáveis pela captação e geração da energia), baterias

(responsáveis pelo armazenamento desta energia gerada) e por conversores estáticos de potência.

- **Módulo de Comunicação:** Este módulo é o encarregado de estabelecer a comunicação entre o satélite, no espaço, e a estação de operação, presente na Terra. É através dele que são enviadas para a Terra, as imagens e os sinais capturados, assim como os comandos de apontamento enviados da estação para o *CubeSat*. Usualmente, este módulo é composto por uma antena de comunicação e um transceptor de rádio.
- **Módulo de Controle:** Este módulo inclui os sensores de orientação, a exemplo do giroscópio e acelerômetro, sensores de temperatura e magnetômetros. Nele também estão presentes os atuadores, como as rodas de reação e *magnetorquers*, componentes responsáveis pelo controle de atitude do *CubeSat*.
- **Módulo de Computação:** Este módulo inclui o computador de bordo (microcontrolador), responsável por toda a programação do *software* de controle, memória e interface de comunicação do nano satélite.
- **Módulo de Carga Útil:** Este módulo inclui todo equipamento específico destinado a realizar a missão (objetivo de funcionamento) do *CubeSat*. Nele podem estar presentes instrumentos científicos de medição, câmeras e sensores.
- **Módulo de Controle Térmico:** Este módulo é o responsável por manter a temperatura adequada dos componentes presentes no *CubeSat*. Geralmente é composto por isolantes térmicos, dissipadores de calor e sistemas de aquecimento.

Na Figura 3, é possível observar os componentes que formam um *CubeSat*, assim como alguns dos módulos descritos acima.

Figura 3 – Componentes e módulos que compõem um *CubeSat*.



Fonte: (FSADU, 2021).

## 2.3 Controle de Atitude

O controle de atitude é indispensável em missões espaciais, visto que é através dele que satélites, como o *CubeSat*, são capazes de controlar sua orientação e se manterem estáveis no espaço. Além disso, este controle é fundamental para que os módulos e componentes presentes no satélite funcionem de maneira adequada.

Entende-se por controle de atitude de veículos espaciais, a alteração proposital dos parâmetros do movimento do centro de massa do veículo e do movimento em torno deste centro. Este controle pode incluir a escolha de uma trajetória ótima, a estabilização de sua posição e a sua orientação com base em uma direção especificada previamente (FABRI, 1997).

Para compreender melhor como funciona o controle de atitude, é importante entender sobre os conceitos de atitude, controle de estabilidade e controle de apontamento, temas abordados nos subtópicos a seguir.



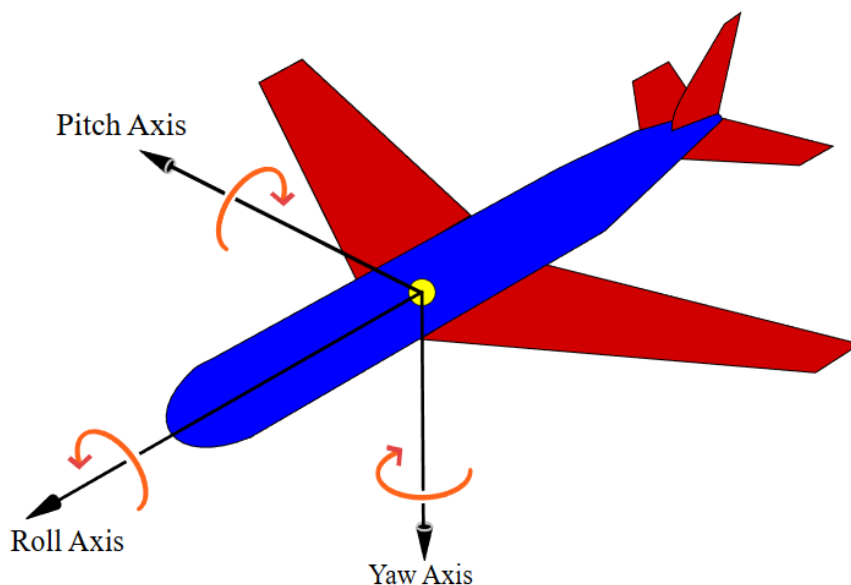
### 2.3.1 Atitude

No contexto aeroespacial, a atitude diz respeito a orientação de um objeto em relação a um sistema fixo de coordenadas. A determinação da atitude de um objeto permite expressar um sistema de coordenadas local, presente no corpo do veículo em função de um sistema de coordenadas externo (FABRI, 1997). Em geral, a atitude de um veículo é composta por 3 elementos referentes aos 3 graus de liberdade que o mesmo possui para rotação em torno dos eixos coordenados, são eles:

- *Roll*: Refere-se à rotação em torno do eixo longitudinal do veículo.
- *Pitch*: Refere-se à rotação em torno do eixo transversal do veículo.
- *Yaw*: Refere-se à rotação em torno do eixo vertical do veículo.

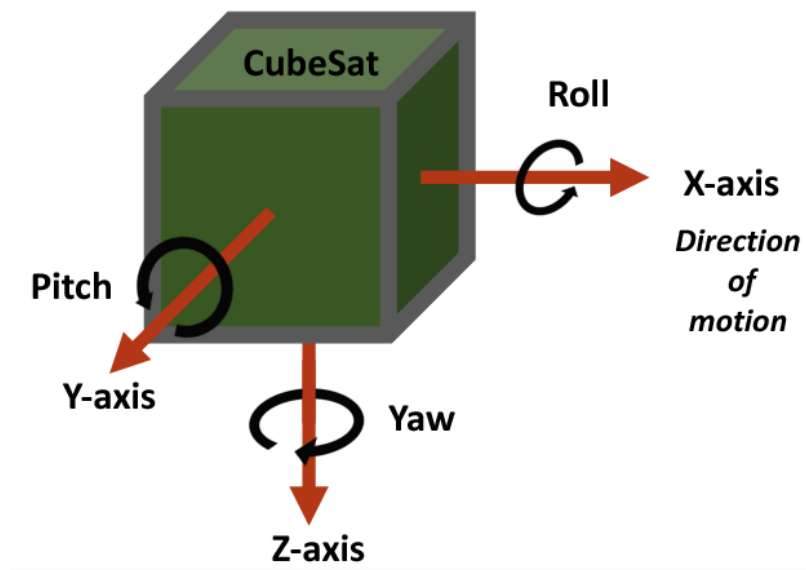
Para fins didáticos, a Figura 4 mostra a disposição dos eixos *Roll*, *Pitch* e *Yaw* em uma ilustração de um avião. Esta representação vale para quaisquer veículos espaciais, incluindo o *CubeSat*, como mostra a Figura 5.

Figura 4 – Representação dos eixos *Roll*, *Pitch* e *Yaw* em um avião.



Fonte: (WIKIPEDIA, 2023).

Figura 5 – Representação dos eixos *Roll*, *Pitch* e *Yaw* em um *CubeSat*.



Fonte: (LISS, ELKAIM, *et al.*, 2018).

### 2.3.2 Controle de Estabilidade

O controle de estabilidade é um elemento crucial na engenharia aeroespacial, pois ele desempenha um papel importante no que diz respeito a manter a atitude de um veículo espacial estável. Quando um *CubeSat* é lançado no espaço, pode começar a rotacionar descontroladamente, com ângulos variando continuamente no sistema de coordenadas. Por conta da conservação do momento angular, é necessário que haja um controle de estabilidade para que o nano satélite fique com ângulos fixos.

O controle de estabilidade, determina a capacidade que um veículo espacial possui, de se manter em uma atitude estável em voo. É de extrema importância que este controle seja bem elaborado, pois a estabilidade do veículo é uma característica unanimemente desejada em praticamente qualquer missão que envolva voos dentro ou fora da atmosfera terrestre.

Tal tarefa, exige sensores capazes de medir a posição do objeto e as perturbações que o envolvem, como os giroscópios e *magnetorquers*, por exemplo. Com base nestas medições e na implementação de um bom algoritmo de controle nos computadores de bordo, é possível ajustar adequadamente a rotação dos eixos do veículo (*Roll*, *Pitch* e *Yaw*), de modo a manter sua atitude estável.

### 2.3.3 Controle de Apontamento

O controle de apontamento é aquele que determina a capacidade que o veículo espacial tem de direcionar-se com precisão e com relativa velocidade, à posição desejada. Por exemplo, caso um satélite esteja apontado para o Brasil e, por alguma razão, deseje tirar fotos do Chile (supondo que ele possui uma câmera), é necessário mudar sua atitude, rotacionando o mesmo de forma que sua câmera fique direcionada ao Chile. O mesmo vale para o apontamento de antenas de comunicação para diferentes estações presentes na Terra e para o apontamento dos painéis solares para o Sol.

Assim como no controle de estabilidade, o controle de apontamento envolve um bom algoritmo capaz de ajustar precisamente os eixos *Roll*, *Pitch* e *Yaw*, para que o veículo espacial manobre de forma correta. A diferença entre os controles de estabilidade e de apontamento, encontra-se no fato de que nas manobras realizadas para controle de apontamento, o satélite adquire uma dada atitude a partir de outra inicial, enquanto que na estabilização é esperado que o satélite realize somente a manutenção de sua posição, impassível a ação de torques perturbadores (FABRI, 1997).

Para movimentar o satélite no espaço de modo a controlar o seu apontamento, são utilizadas Rodas de Reação, como a mostrada na Figura 6. Essas rodas de reação conseguem aumentar o momento de inércia dos atuadores, viabilizando os controles de apontamento e de estabilidade.

Figura 6 – Exemplo de roda de reação responsável por movimentar o *CubeSat*.



Fonte: (ROCKETLAB, 2023).

O princípio básico por trás do movimento do satélite através da rotação da roda de reação é fundamentado na conservação do momento angular. O momento angular total do sistema composto pelo *CubeSat* e pela roda de reação deve ser equivalente à soma dos momentos angulares individuais de cada componente, vide equação (1).

$$L = I_s \cdot \omega_s + I_r \cdot \omega_r \quad (1)$$

Onde:

$L$  = Momento Angular Total;

$I_s$  = Momento de Inércia do Satélite;

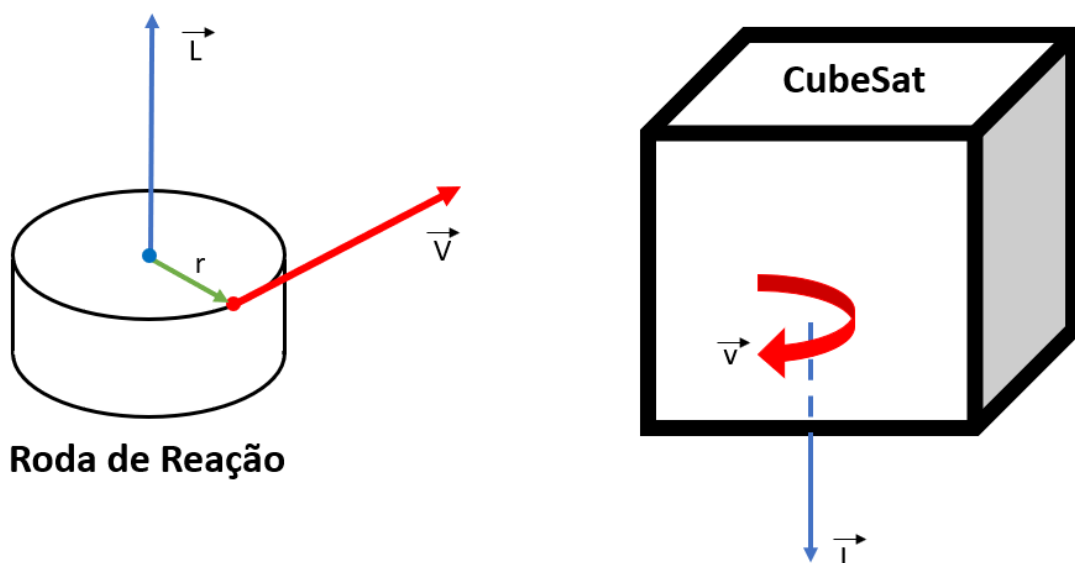
$\omega_s$  = Velocidade Angular do Satélite;

$I_r$  = Momento de Inércia da Roda de Reação;

$\omega_r$  = Velocidade Angular da Roda de Reação.

Em outras palavras, quando a velocidade da roda aumenta, a velocidade do satélite diminui. Isso resulta em um movimento do satélite na direção oposta à rotação da roda, uma vez que o momento angular total deve ser mantido constante. Tal situação é ilustrada na através de um diagrama de corpo livre da roda de reação e do *CubeSat*.

Figura 7 - Diagramas de corpo livre da roda de reação e do *CubeSat*.



Fonte: De autoria própria.

Conforme apresentado na Equação (1), a mudança de velocidade angular do satélite vai acabar dependendo da velocidade angular da roda de reação e do momento de inércia desses dois elementos. Assim, a partir do giro da roda de reação, torna-se possível realizar o controle de apontamento do *CubeSat*.

É importante destacar que o satélite possui dimensões e massa consideravelmente maiores do que a roda de reação, resultando em um momento de inércia superior. Consequentemente, o valor absoluto da velocidade angular do satélite será menor do que o da roda de reação.

Este conceito é fundamental para a compreensão do controle de atitude e movimento de satélites. Controlar a velocidade da roda de reação permite ajustar a orientação e posição do satélite no espaço, sendo um princípio crucial para diversas aplicações espaciais.

A fundamentação teórica apresentada fornece a base necessária para o entendimento de como é feito o desenvolvimento de uma plataforma de testes e planta física de um *CubeSat*, tema do próximo capítulo.

### 3 DESENVOLVIMENTO DE PLATAFORMA DE TESTES

Neste capítulo, é apresentado o desenvolvimento de toda a planta projetada para os testes de controle de atitude. Primeiramente, foi feita uma pesquisa sobre projetos de *CubeSats* para servir de referência. O *CubeSat* da competição *CubeDesign* do INPE foi adotado como modelo base, levando em consideração os requisitos desta competição.

Com base neste modelo, foi projetada a estrutura completa do nano satélite, determinando a posição de cada componente e a arquitetura do sistema. Cada módulo necessário para o funcionamento do *CubeSat* foi cuidadosamente identificado e dimensionado. Em seguida, foram escolhidos quais seriam os dispositivos e materiais utilizados em cada um destes módulos, desde o material que compõe a parte estrutural, até os sensores, atuadores e o microcontrolador.

Além do próprio satélite, também fazem parte da plataforma de testes, a estrutura de suporte, responsável pela suspensão do cubo, e o *software* utilizado para a validação do sistema, que tem o papel de mostrar se a planta obedece aos requisitos de projeto pré-estabelecidos. Esse *software* também tem como objetivo viabilizar a avaliação do desempenho do sistema de controle implementado.

#### 3.1 Engenharia de Requisitos

Para a construção do *CubeSat*, utilizou-se como base os requisitos da competição *CubeDesign*, do INPE. O *CubeDesign* é uma competição de nano satélites criada com o intuito de ressaltar a importância das atividades aeroespaciais, ganhando assim, mais aproximação com a sociedade. Dentro desta competição, existe a categoria *CubeSat*, onde as equipes envolvidas têm a oportunidade de desenvolverem atividades de modelagem e simulação de sistemas espaciais (INPE, 2021).

Um dos objetivos da categoria *CubeSat*, é o ADCS, sigla para “*Attitude Determination and Control Subsystem*” (Subsistema de controle e determinação de atitude), que deve obedecer a alguns pré-requisitos. Dentre eles, são utilizados na construção da plataforma de testes deste trabalho os seguintes requisitos:

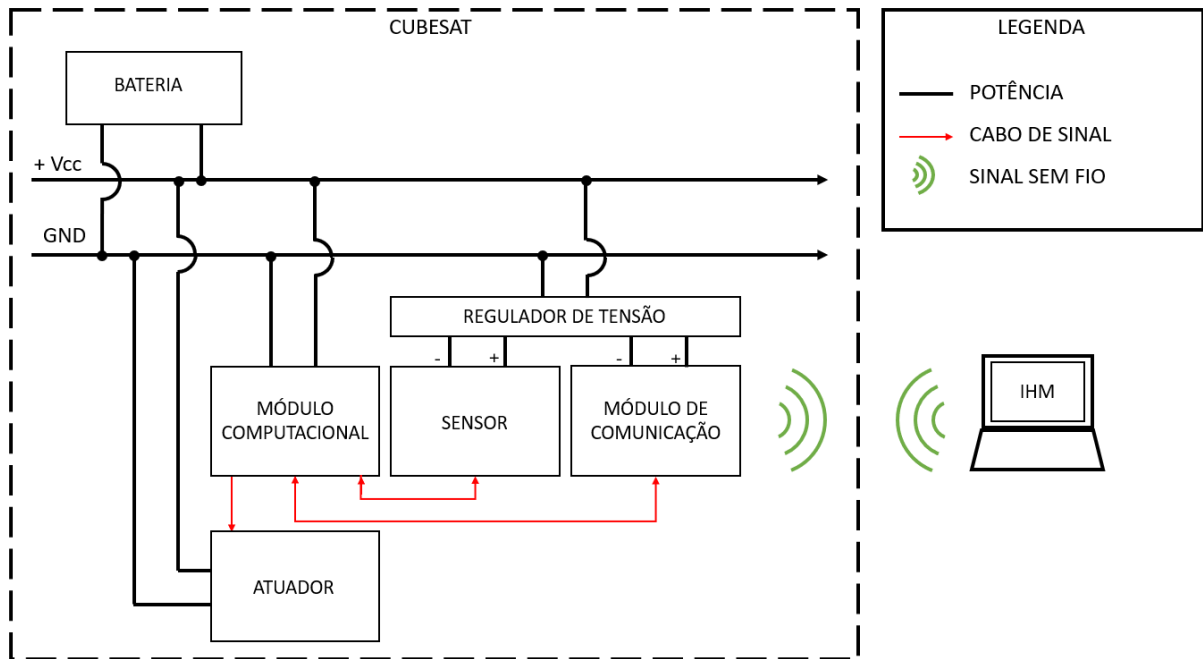
- O *CubeSat* deve ser padrão 1U (10 cm x 10 cm x 11,35 cm);
- A massa do *CubeSat* não deve ultrapassar 1,3 kg;
- O *CubeSat* deve ser alimentado somente pelas baterias.
- O subsistema ADCS deve ser capaz de determinar sua atitude parametrizada em ângulos de Euler, com erro na faixa de  $\pm 1^\circ$  (0,0174533 rad);
- O sistema de controle deve ser capaz de manter a velocidade do *CubeSat* dentro da faixa de  $\pm 0,1^\circ/s$  (0,0174533 rad/s);

### 3.2 Arquitetura de Montagem do *CubeSat*

A arquitetura de montagem do *CubeSat* é de fundamental importância no que diz respeito a parte organizacional do projeto da plataforma de testes. O intuito dela é fornecer uma visão global do sistema implementado através de um diagrama, contendo cada um dos módulos e equipamentos utilizados, como eles se conectam dentro do nano satélite e suas funções.

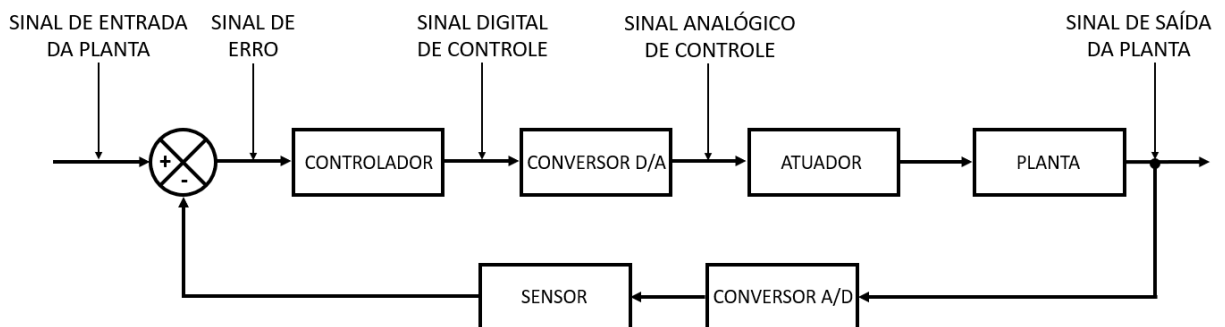
Primeiramente, é necessário alimentar o sistema com uma fonte de energia, que neste caso são duas baterias de corrente contínua em série (ver Subseção 3.2.2 para mais detalhes). Este sistema, irá alimentar o Módulo de Computação, composto por um Microcontrolador e um driver Ponte-H, este último sendo responsável por controlar o motor (ver Subseção 3.2.5 para mais detalhes). Em seguida, tem-se o atuador (motor CC), que é alimentado diretamente pelo módulo de energia e controlado através do módulo computacional.

Uma vez que se deseja fazer o controle em malha fechada do *CubeSat*, faz-se necessário estabelecer uma realimentação no sistema. Para tanto, é utilizado um sensor cuja alimentação também é provida pelo módulo de energia, passando antes por um regulador de tensão. Para este fim, é utilizado um giroscópio de nove eixos, capaz de obter a orientação angular do *CubeSat* (ver Subseção 3.2.4 para mais detalhes). Por fim, tem-se o módulo de comunicação, que para o presente trabalho foi implementado através de um módulo Bluetooth, também alimentado pelas baterias através do regulador. O diagrama representativo da arquitetura de montagem do *CubeSat* é mostrado na Figura 8.

Figura 8 – Arquitetura elétrica do *CubeSat*.

Fonte: De autoria própria.

Com base no esquema acima, é possível construir um diagrama representativo da malha de controle deste sistema, permitindo um melhor entendimento da arquitetura do nano satélite e fornecendo a base para a modelagem matemática da plataforma, abordada no Capítulo 4. A Figura 9, mostra o esquema da malha de controle do sistema implementado.

Figura 9 – Esquema de controle em malha fechada do *CubeSat*.

Fonte: De autoria própria.

De maneira geral, o sinal de entrada do sistema de controle consiste em um ângulo, definido pelo usuário. O erro do sistema de controle é calculado como a diferença entre o valor desse ângulo de referência e a leitura do ângulo atual,



proveniente do sensor. A inércia do *CubeSat* é modelada por meio de uma função de transferência (representação da planta). O microcontrolador engloba toda a lógica de programação de controle, incluindo a leitura dos dados do sensor e emissão de comandos para o atuador, tudo com o objetivo de conduzir o sistema à resposta desejada. tudo com o objetivo de conduzir o sistema à resposta desejada.

Dentro do microcontrolador, a função de controle do sistema é implementada para gerar o sinal de controle apropriado. Esse sinal de controle é então empregado para influenciar o comportamento do sistema, com a finalidade de minimizar o erro entre o ângulo de referência e o ângulo medido pelo sensor. O processo de controle ocorre iterativamente, ajustando continuamente o sinal de controle para alcançar a resposta desejada e manter o sistema estável e alinhado de acordo com os parâmetros estabelecidos.

### **3.2.1 Módulo Estrutural**

A estrutura ou chassi do *CubeSat* é responsável pela sustentação e suporte de todos os componentes que o compõem, ou seja, os módulos de energia, de comunicação, de controle e de computação. Além disso, este módulo também tem a função de proteger tudo o que está dentro do *CubeSat*.

Para a construção desta estrutura, utilizou-se uma impressora 3D e material PLA (Ácido Polilático) com preenchimento de 30%, pois essa tecnologia possibilita a fabricação de peças personalizadas conforme as especificações do projeto. As partes que constituem a estrutura, como as hastes laterais de suporte, a tampa inferior e a tampa superior, foram impressas individualmente. A tampa inferior foi construída com uma abertura para a conexão do motor à estrutura. Esse motor é utilizado para acoplar à roda de reação na estrutura do *CubeSat*. Cada uma dessas peças foi projetada para ser fixada através de parafusos, proporcionando uma montagem fácil, robusta e precisa. Depois de montada, a estrutura possui dimensões 10 x 10 x 11,35 cm. A Figura 10 apresenta o módulo estrutural.

Figura 10 – Chassis do *CubeSat* presente no Módulo estrutural.



Fonte: De autoria própria.

### **3.2.2 Módulo de Energia**

O módulo de energia deve ser cuidadosamente escolhido e testado antes de ser implementado de maneira definitiva, isto porque a alimentação do sistema pode acabar gerando diversos problemas, se mal especificada. Nesse módulo, são incorporadas as baterias, e também podem ser incluídos conversores e reguladores de tensão. Dado que um nano satélite é composto por vários componentes que dependem das baterias como fonte de energia, é possível que instabilidades no funcionamento possam ocorrer.

Para este projeto, o módulo de energia é composto de duas baterias de íon de lítio de 3,7 V e 9900 mA, conectadas em série, como mostra a Figura 11, resultando em uma tensão de aproximadamente 7,4 V. Esta tensão alimenta diretamente o módulo computacional e o motor conectado a ele. Contudo, os módulos de comunicação (módulo Bluetooth) e o sensor (giroscópio) não suportam tensões superiores a 6 V. Em decorrência disso, eles são alimentados através de um regulador de tensão presente no módulo computacional. Esse regulador ajusta a tensão de entrada de 7,4 V da bateria para uma saída de 5 V, que é a tensão nominal de alimentação do módulo Bluetooth e do giroscópio.

Figura 11 – Baterias de íon de lítio em série que compõem o Módulo de energia.



Fonte: De autoria própria.

### 3.2.3 Módulo de Comunicação

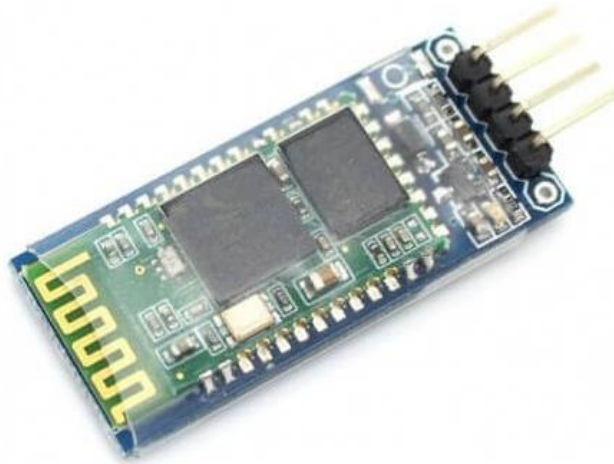
É por meio deste módulo que são enviados comandos para o *CubeSat*, sejam eles de apontamento, ou até mesmo para executar alguma ação, como tirar uma foto de um lugar específico da Terra, por exemplo. Como o próprio nome sugere, este é o módulo responsável pela comunicação entre o nano satélite e a central de comandos. Além de receber comandos da Terra para executar determinada ação no espaço, ele também envia dados do espaço para a Terra, que vão desde as informações coletadas, até diagnósticos no que diz respeito a saúde do satélite.

Para estabelecer a comunicação entre o *CubeSat* e o software de monitoração e comando, utilizou-se um módulo Bluetooth, mais especificamente o módulo HC-06, mostrado na Figura 12, que utiliza o protocolo RS-232 encapsulado no Bluetooth, permitindo uma comunicação bidirecional sem fio.

É importante ressaltar, que a comunicação Bluetooth só funciona a curtas distâncias, o que não atenderia caso o cubo fosse lançado para o espaço. Entretanto, para fins de teste e validação de controle, este tipo de comunicação não só atende, como também é uma das melhores para o experimento em questão (se tratando de comunicação sem fio), isto porque ela é de fácil configuração, possui um baixo

consumo de energia, é capaz de estabelecer uma conexão confiável entre o satélite e o software de monitoração e comando (computador), assim como possui um baixo custo de aquisição.

Figura 12 – Módulo Bluetooth HC-06 presente no Módulo de Comunicação.



Fonte: (XUKYO, 2021).

O módulo Bluetooth HC-06 possui um range de tensão de operação que vai de +3,3V até +6V, uma corrente de operação de 40 mA, range de temperatura de operação que vai de -20 °C até +55 °C e sua banda de transmissão e recebimento de dados varia de 2,40 GHz até 2,48 GHz.

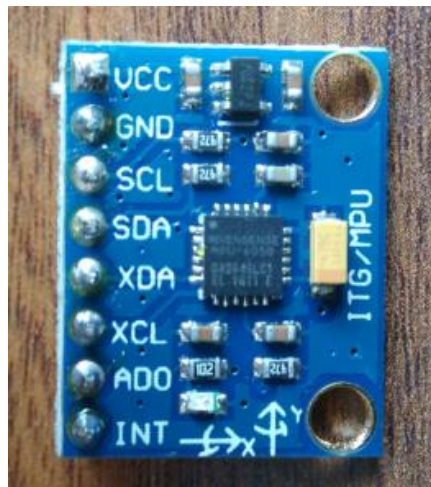
#### **3.2.4 Módulo de Controle**

Este módulo é responsável por controlar a planta. Nesse módulo estão presentes os sensores de posição, velocidade, temperatura e os atuadores. Os sensores são encarregados de fornecer a realimentação do sistema, ou seja, a posição angular do *CubeSat*, sua velocidade de rotação ou translação, assim como a temperatura do sistema. Os dados captados por eles são enviados ao módulo de computação, que os processa e envia um comando para os atuadores, os quais são responsáveis por implementar, de fato, a ação de controle. No geral, esses atuadores costumam ser as rodas de reação e os *magnetorquers*. Estes dispositivos são capazes de alterar a

posição do nano satélite de maneira significativa, permitindo um controle mais preciso e ajustes na sua orientação.

Para este projeto, utilizou-se como sensor de orientação, o giroscópio acelerômetro MPU6050 (vide Figura 13), que é capaz de medir a aceleração e a posição nos 3 eixos, assim como também a temperatura.

Figura 13 – Giroscópio e acelerômetro MPU6050 responsável por fornecer a orientação do *CubeSat*.



Fonte: (GUIMARÃES, 2018).

Este dispositivo se comunica através do protocolo I2C, sendo capaz de transmitir e receber dados. É importante destacar, que os pinos deste sensor possuem um limite máximo de tensão 3,46 V. A única exceção é o Vcc, que pode ser ligado em uma tensão de 5 V, pois sua placa possui um regulador de tensão de 5 V para 3,3 V (GUIMARÃES, 2018). Sua calibração é feita de forma automática através do código implementado no microcontrolador, onde a posição de ângulo igual a zero graus, é calibrada para ser a posição inicial do sensor ao carregar o código.

Como atuador, utilizou-se um motor CC *coreless* 1020A, mostrado na

Figura 14, acoplado a uma roda de reação. Este motor possui tensão nominal de 3,7 V, rotor de 1 mm e pode chegar a uma velocidade de até 40000 rpm. A roda de reação possui altura de 1,8 cm e diâmetro de 4,8 cm e foi fabricada utilizando impressão 3D e material PLA com preenchimento de 100%, de tal modo que o motor pudesse ser encaixado nela. A roda de reação mostrada na Figura 15, foi feita sob medida através de uma impressora 3D, de tal forma que o motor utilizado encaixasse

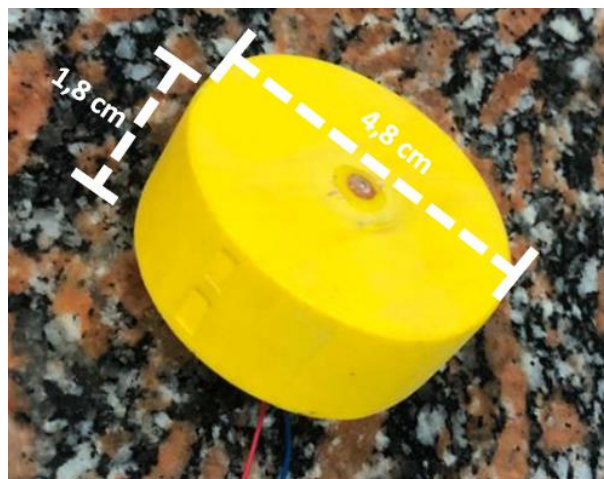
perfeitamente na parte inferior da mesma com o mínimo de folga possível, facilitando a realização do controle.

Figura 14 – Motor CC *Coreless* 1020A utilizado no módulo de controle.



Fonte: De autoria própria.

Figura 15 – Roda de reação responsável pelo movimento do *CubeSat*.



Fonte: De autoria própria.

### 3.2.5 Módulo de Computação

Nesse módulo estão presentes o microcontrolador e a memória de dados do *CubeSat*. Aqui é realizada toda a parte de comandos, tratamento de dados, configuração da comunicação e implementação da lógica de controle.

Para este projeto, utilizou-se o Arduino Uno como plataforma de desenvolvimento (vide Figura 16). A seleção do Arduino Uno como plataforma de desenvolvimento foi motivada por diversos fatores, entre eles estão incluídos: a facilidade de sua programação em linguagem C; sua versatilidade que permite a conexão com diversos periféricos e sensores na formatação de *shields*; a facilidade de aquisição; e existência de uma grande comunidade de usuários, o que facilita o acesso ao suporte e obtenção de recursos de programação.

Figura 16 – Microcontrolador Arduino Uno utilizado no módulo computacional.



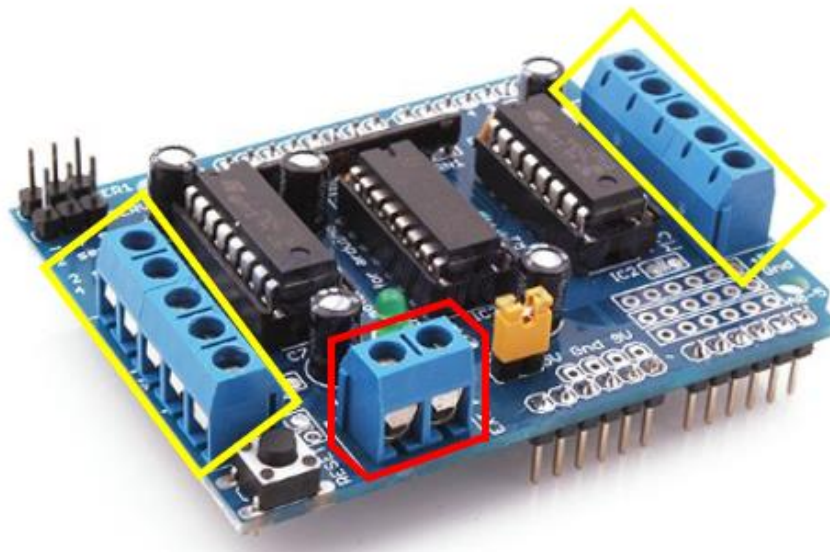
Fonte: (RAVIKIRAN, 2022).

Para controlar o atuador (motor acoplado a roda de reação), foi utilizado um *shield* para Arduino que contém uma ponte H L293D (vide Figura 17). Esse *shield*, é acoplado diretamente ao Arduino, sendo comandado conforme a programação implementada no mesmo, e tem o papel de regular o nível de tensão que chega ao motor, assim como a polaridade desta tensão. Em outras palavras, ele é capaz de controlar a velocidade e o sentido de giro do motor.



A razão pela escolha desse *shield*, se deve principalmente ao fato dele ser integrável de forma direta ao Arduino. De fato, para o uso dele, basta encaixá-lo fisicamente no Arduino Uno, o que contribui não só para facilidade na implementação do sistema de controle, mas também otimiza o espaço disponível dentro do *CubeSat*.

Figura 17 – *Shield* para Arduino contendo *driver* com Ponte H L293D.



Fonte: (ARDUINOECIA, 2014).

É possível observar nas laterais da Figura 17, que o driver L293D possui 4 saídas, destacadas em amarelo, ou seja, ele é capaz de controlar até 4 motores. Entretanto, para o desenvolvimento deste trabalho, só é necessária uma roda de reação para fazer o controle de atitude em um eixo, portanto, utilizou-se somente a saída M1 desse *shield*.

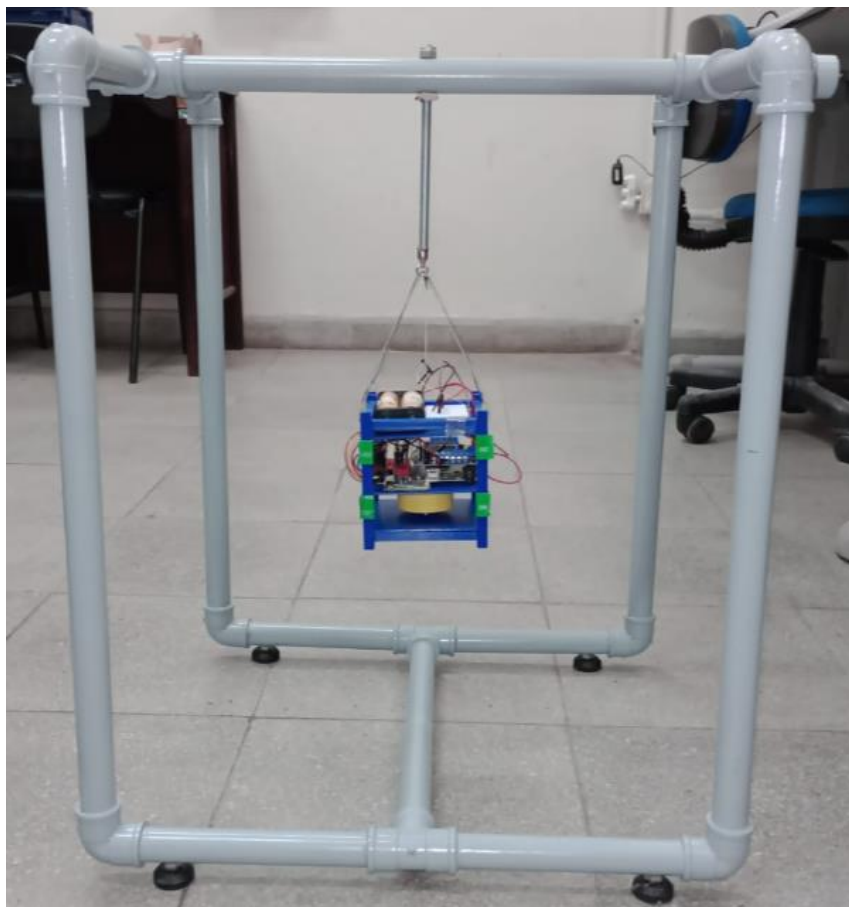
A alimentação desse módulo é provida pela bateria presente módulo de energia. O *shield* que contém a ponte-H L293D é alimentado pelas baterias através da entrada presente na parte superior da Figura 17, destacada em vermelho. Ao fornecer energia para o *shield*, o Arduino é automaticamente alimentado. Isso ocorre porque o barramento de alimentação do Arduino e do seu *shield* são eletricamente conectados.



### 3.3 Estrutura de Testes

A estrutura de testes é composta basicamente por 2 itens, que são uma estrutura responsável por suspender o *CubeSat*, e o próprio *CubeSat*. Para a estrutura responsável por suspender o cubo, utilizou-se de canos PVC, fixadores, fios e uma haste. Com os canos PVC foi montado a parte de suporte, acoplando os fixadores na base da estrutura, uma vez que é muito importante que ela não se mova, pois caso isso aconteça, influenciará na leitura de posição do satélite pelos sensores, comprometendo o controle do sistema. Na parte superior central, existe uma haste cuja ponta pode rotacionar livremente em um eixo (eixo *Yaw*). Na ponta desta haste, são colocados dois arames que se conectam ao cubo, permitindo que o mesmo fique bem seguro e suspenso, podendo rotacionar de forma estável. A Figura 18 apresenta a estrutura de testes completamente montada.

Figura 18 – Estrutura de testes para validação experimental do funcionamento do controle de atitude do *CubeSat*.



Fonte: De autoria própria.

### 3.4 Estrutura Mecânica do *CubeSat*

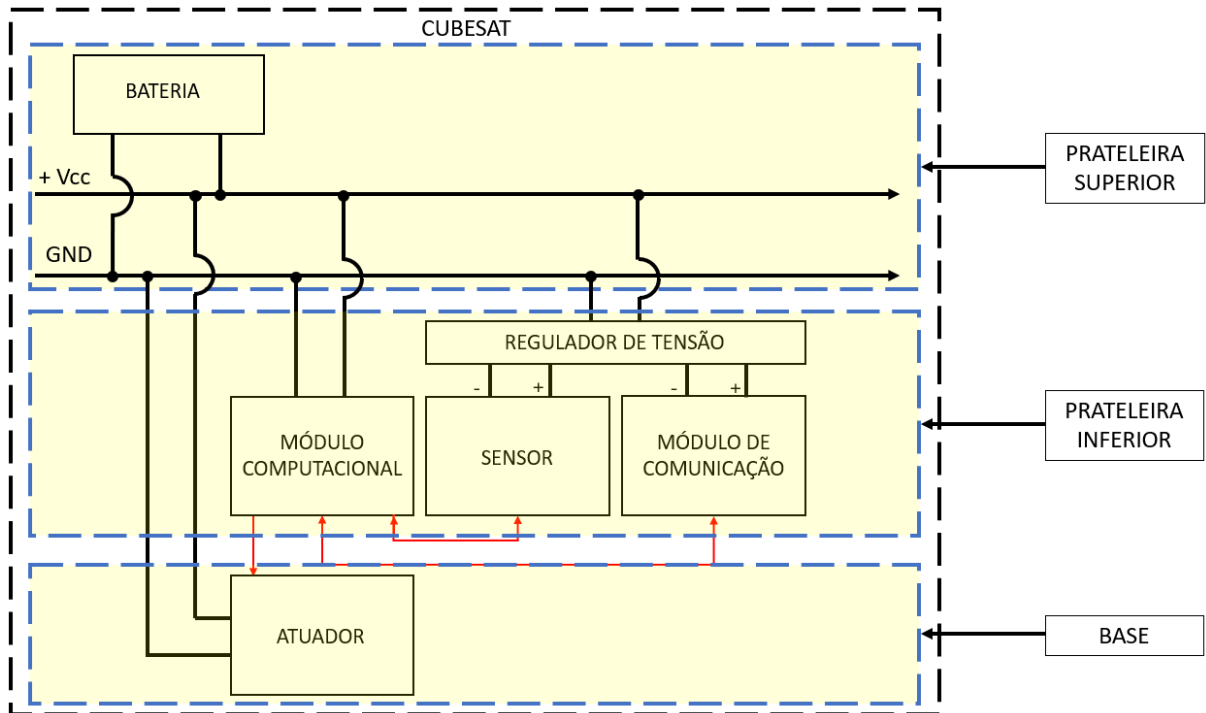
Para montar o *CubeSat*, o primeiro passo é fazer o módulo estrutural, ou seja, toda a parte de sustentação e suporte dos outros módulos, como descrito na Subseção 3.2.1 e mostrado na Figura 10.

A estrutura do *CubeSat* montada neste trabalho conta com três divisões que podem ser utilizadas para apoiar seus módulos. Essas divisões são: prateleira superior (divisão mais superior); prateleira inferior (divisão central); e base (divisão inferior). Com relação a disposições dos módulos do *CubeSat*, esses foram dispostos da seguinte forma:

- Na base foi acoplado o motor e a roda de reação do *CubeSat*;
- Na prateleira inferior foram posicionados o módulo de computação (Arduino e ponte H), o sensor giroscópio MPU6050, o módulo de comunicação (módulo Bluetooth) e uma pequena *protoboard*, para auxiliar nas ligações elétricas dos componentes;
- Na prateleira superior foi posicionado o módulo de energia (baterias).

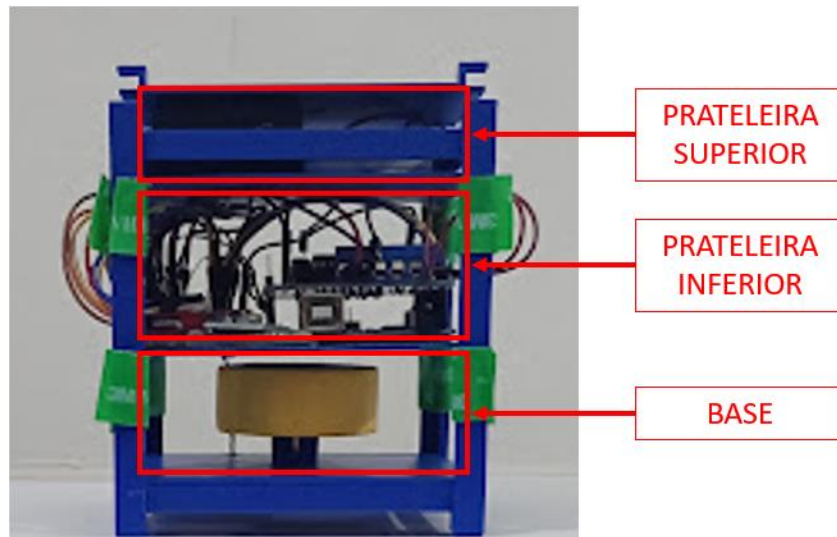
Essa disposição dos componentes do *CubeSat*, foi feita baseada na arquitetura mostrada na Figura 8, pensando principalmente na distância que os componentes estão uns dos outros. A Figura 19 mostra como os componentes foram dispostos fisicamente no *CubeSat* tendo como base a arquitetura do mesmo, enquanto a Figura 20 exibe uma foto do *CubeSat* completamente montado.

Figura 19 – Disposição dos componentes do *CubeSat* baseado na arquitetura



Fonte: De autoria própria.

Figura 20 – Montagem do *CubeSat* feita em três prateleiras.



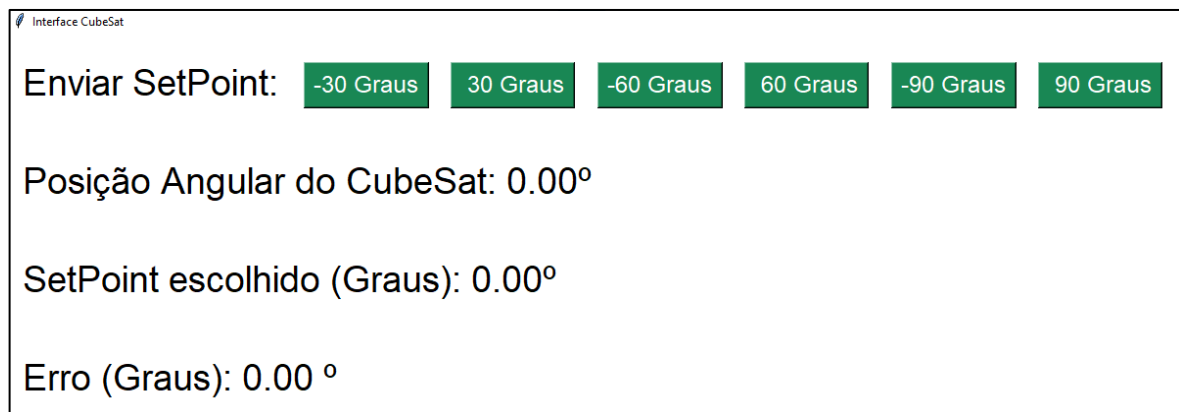
Fonte: De autoria própria.

### 3.5 Software de Validação

Com a plataforma física pronta para os testes, faz-se necessário o uso de um *software* para coletar os dados do sistema. Dados como a posição angular na qual se encontra o *CubeSat* e o erro entre o *SetPoint* requisitado e o seu valor real, são importantes para a validação do sistema implementado.

Pensando nisso, foi implementado um sistema capaz de não só extrair estes dados, como também enviar os comandos de *SetPoint* para o nano satélite através de uma interface gráfica, mostrada na Figura 21.

Figura 21 – Interface Gráfica

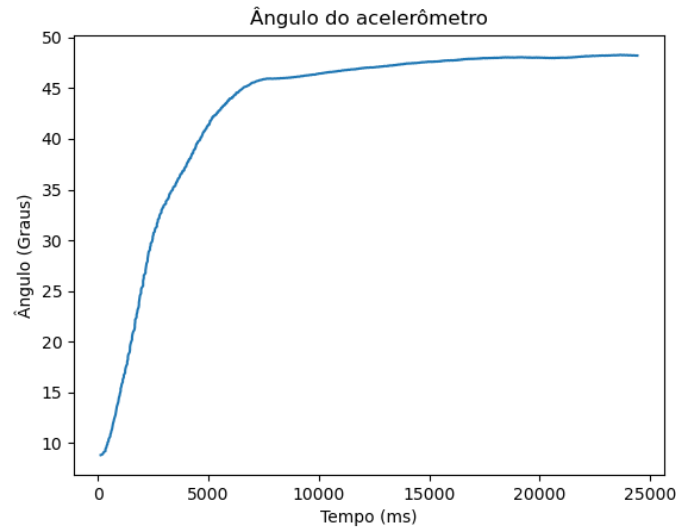


Fonte: De autoria própria.

Esta interface foi desenvolvida em Python utilizando a IDE *pycharm* e a biblioteca “*tkinter*” e, apesar de simples, possui todas as funcionalidades necessárias para validação da plataforma de testes. Nela, existem algumas opções de *SetPoint* de ângulo, onde o usuário pode clicar no botão com o ângulo correspondente ao que ele deseja que o *CubeSat* faça o apontamento. Além disso, é possível acompanhar em tempo real a posição angular do nano satélite fornecida pelo sensor acelerômetro, o *SetPoint* escolhido e o erro entre o *SetPoint* e a posição real.

Ao enviar comandos de *SetPoint* de ângulo, o *software* é capaz de gerar automaticamente uma curva característica do comportamento do sistema, com o eixo X contendo os valores de tempo em milissegundos e o eixo Y contendo os valores de posição angular em graus. A Figura 22 mostra um exemplo de curva obtida através do *software* de validação.

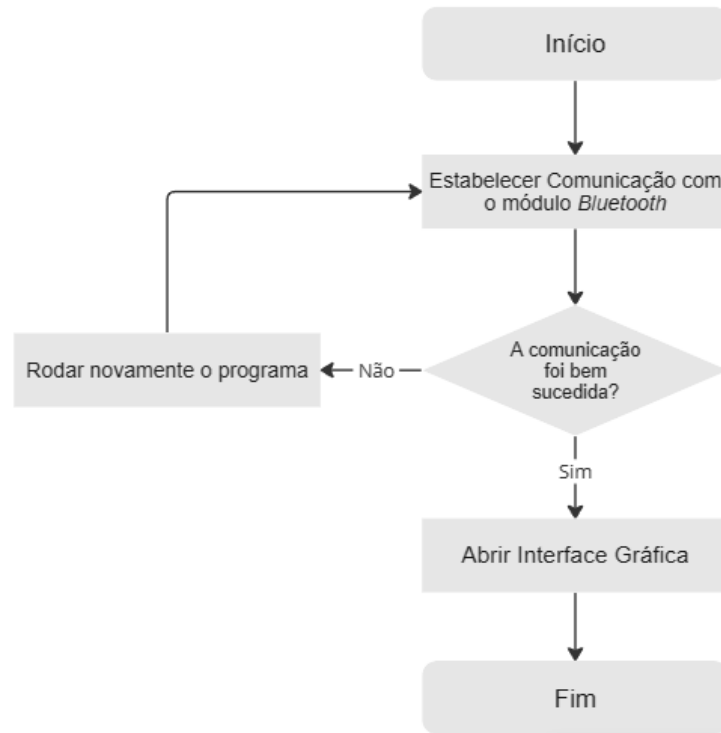
Figura 22 - Exemplo de curva de Posição Angular obtida a partir do Software de Validação



Fonte: De autoria própria.

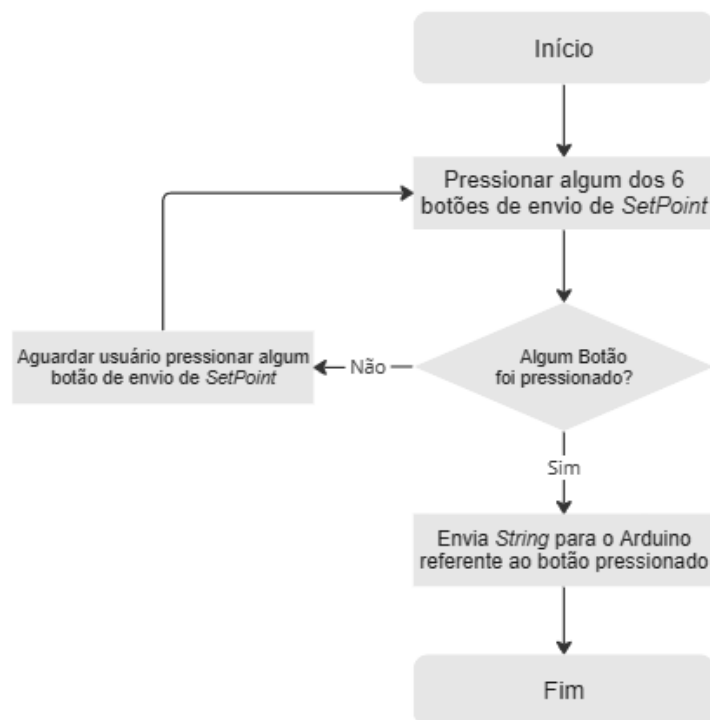
Uma outra funcionalidade do *software* desenvolvido é a sua capacidade de gerar um arquivo CSV com os dados de tempo em milissegundos em uma coluna e os de posição angular em outra. Isso é de extrema importância, uma vez que esses dados em CSV podem ser importados para *softwares* como o *MatLab* e, a partir daí, ser feita a modelagem e o projeto de controle do sistema (tema do próximo capítulo). A lógica de funcionamento do código implementado é apresentada em três partes principais a partir dos fluxogramas da Figura 23, Figura 24 e Figura 25.

Figura 23 – Fluxograma representativo da lógica de estabelecimento da comunicação *Bluetooth* com o microcontrolador e inicialização da Interface Gráfica.



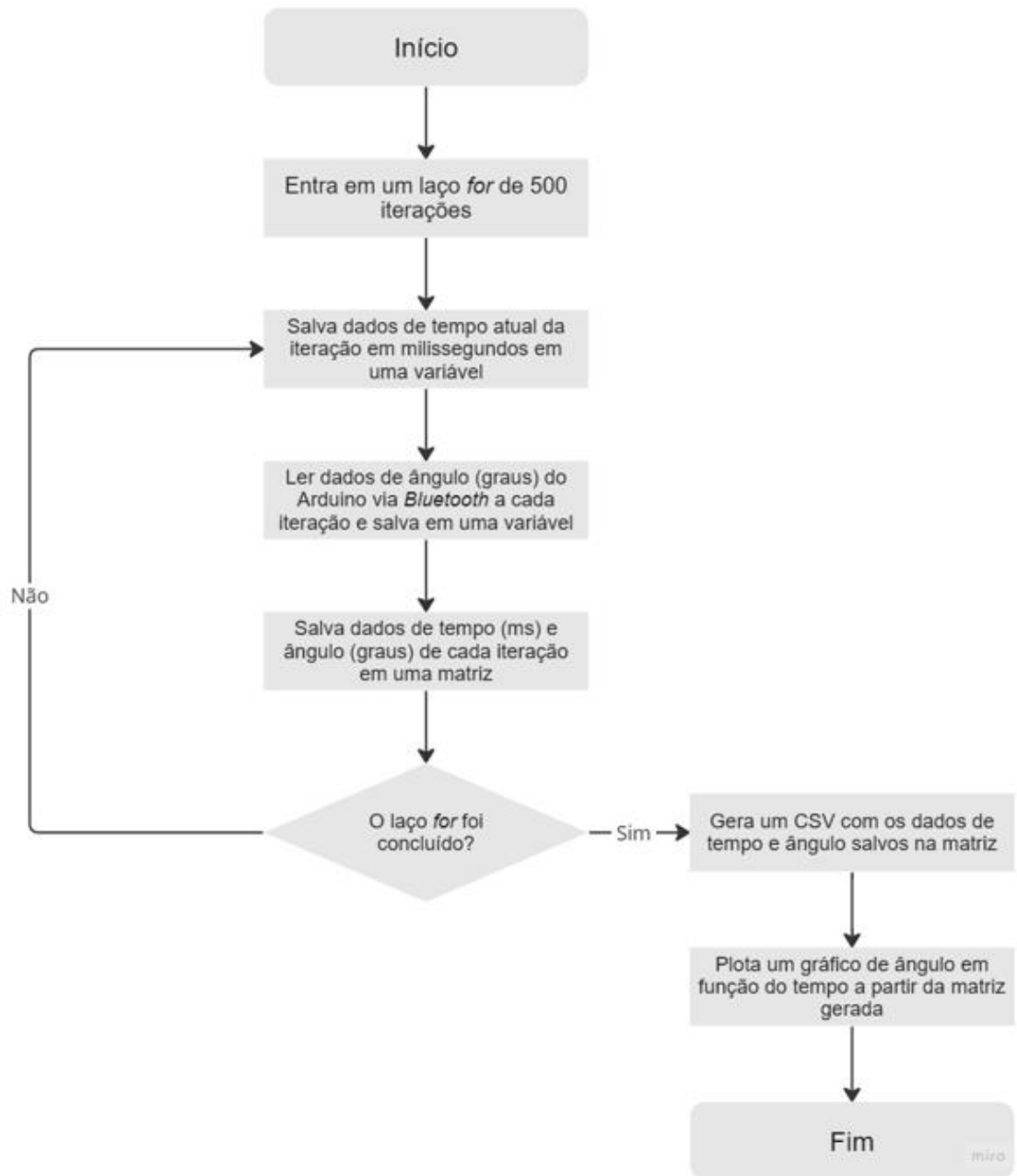
Fonte: De autoria própria.

Figura 24 – Fluxograma representativo da lógica de envio de *SetPoint* para o microcontrolador.



Fonte: De autoria própria.

Figura 25 – Fluxograma representativo da lógica responsável por ler dados microcontrolador, gerar arquivo CSV e plotar o gráfico.



Fonte: De autoria própria.

## 4 PROJETO E VALIDAÇÃO DO SISTEMA DE CONTROLE

Neste capítulo, apresenta-se a implementação e a validação do sistema de controle do *CubeSat*. Com o auxílio da plataforma de testes desenvolvida, é feita a modelagem e validação da planta, obtendo a função matemática que representa o *CubeSat*. Em seguida, é necessário validar o modelo obtido para a planta, abordagem feita na Subseção 4.2.

Com base nisso, o próximo passo é desenvolver o controle do sistema. Primeiramente, foram definidos os requisitos de controle e em seguida, foi projetado o controlador. O *Matlab* foi utilizado para sintonia do controlador.

Com o sistema de controle implementado, foi feita a validação experimental do mesmo utilizando a plataforma de testes. Os gráficos e os arquivos CSV gerados a partir do *software* de validação, são tratados e analisados com o auxílio do *Matlab*, permitindo a validação experimental do projeto. Além disso, foram feitas avaliações de diferentes cenários, como por exemplo, o comportamento do sistema quando submetido a perturbações.

### 4.1 Modelagem Matemática da Planta

Praticamente todo e qualquer dispositivo ou equipamento que possua uma dinâmica associada a ele, pode ter esta dinâmica representada de forma aproximada por equações diferenciais. O primeiro passo para construção de um modelo matemático de uma planta, é escolher qual tipo de abordagem será adotada, ou seja, se o sistema será abordado como Caixa-Branca, Caixa-Cinza ou Caixa-Preta. (AGUIRRE, 2014).

Para o *CubeSat* montado, não foi feita nenhuma análise prévia quanto ao seu formato, sua massa exata e o seu momento de inércia. Contudo, na literatura, diversos trabalhos já trazem uma abordagem analítica para definição da função de transferência desse sistema, como, por exemplo, em (WERTZ, 1978). Segundo (MESQUITA, 2017), um modelo simplificado, de segunda ordem, da dinâmica de um eixo de satélite é dado por:



$$G(s) = \frac{1}{Js^2} \quad (2)$$

em que  $J$  é o momento de inércia do satélite em relação ao eixo avaliado. Tendo em vista a equação acima, optou-se por modelar o *CubeSat* através de uma abordagem do tipo Caixa-Cinza, em que experimentos são utilizados para estimar os parâmetros do modelo teórico (neste caso, o parâmetro  $J$ ).

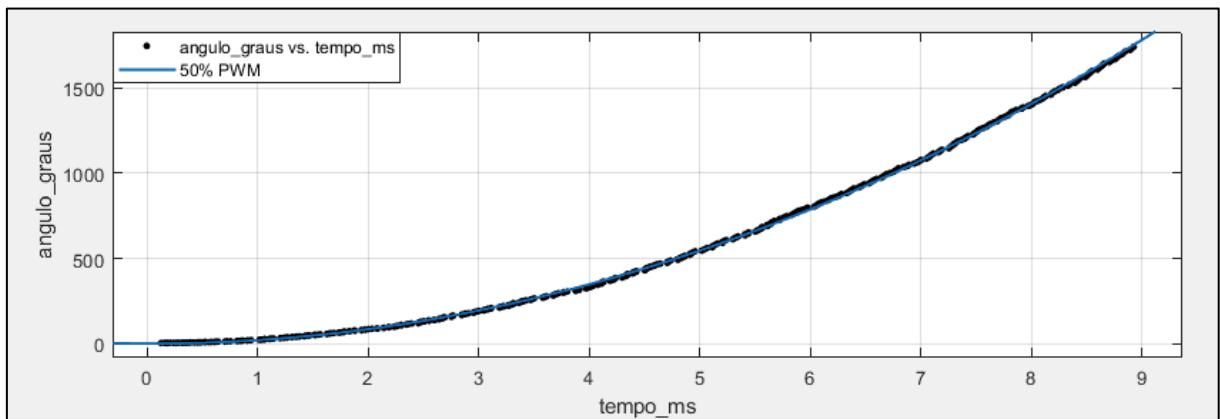
Como o sistema é projetado para atender a uma posição angular desejada a partir de um *SetPoint*, sendo a roda de reação o atuador capaz de produzir o movimento do cubo, os primeiros testes consistem em aplicar um degrau de velocidade no motor e medir a posição angular do nano satélite ao longo do tempo, ou seja, medir como a posição angular varia quando o motor é sujeito a um degrau.

No microcontrolador, a atuação do motor é definida a partir do valor configurado no seu PWM, que varia de 0 a 255, sendo 0 o 0% do PWM e 255 o 100% do PWM. Foram feitos 5 testes aplicando diferentes níveis de PWM no motor, são eles:

- Teste 1: PWM em 255 (100%);
- Teste 2: PWM em 191 (75%);
- Teste 3: PWM em 127 (50%);
- Teste 4: PWM em 91 (37,5%);
- Teste 5: PWM em 63 (25%).

Para os testes 1 e 2, o sistema apresentou fortes vibrações devido à alta velocidade da roda de reação, o que acabou causando instabilidades na comunicação *bluetooth*, impedindo a leitura de ângulo pelo acelerômetro. Em virtude disto, optou-se por utilizar como padrão a velocidade máxima do motor como sendo aquela com o PWM em 127 (50% do mesmo). Para o teste 3, aplicando um degrau de 50% do PWM no motor, obteve-se uma lista de pontos de posição angular do *CubeSat* em função do tempo em um arquivo CSV utilizando o *software* de validação. Estes pontos foram importados para o ambiente do Matlab e utilizados no aplicativo *Curve Fitting Tool*, presente no mesmo. A curva obtida pode ser vista na Figura 26.

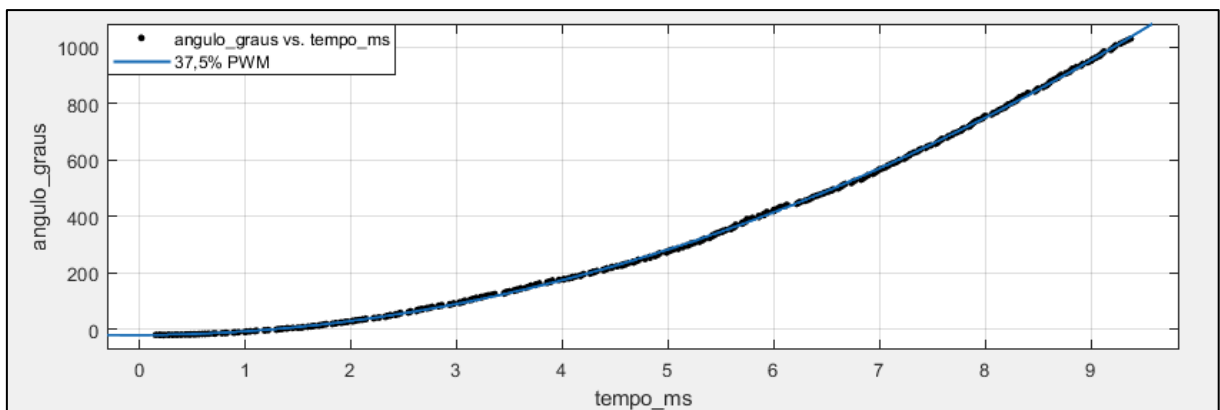
Figura 26 – Curva de resposta considerando aplicação de PWM com ciclo de trabalho de 50%.



Fonte: De autoria própria.

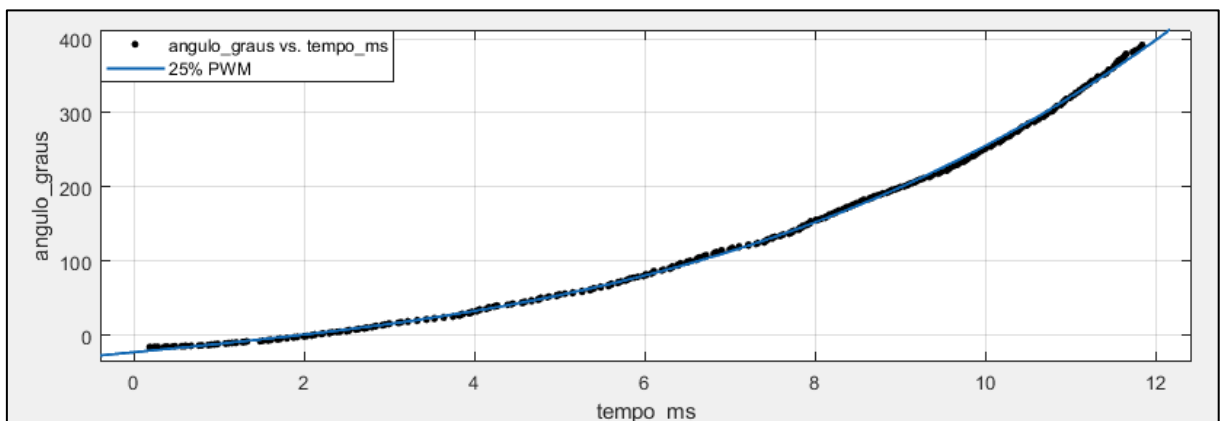
As curvas de posição angular obtidas quando aplicados os valores de 37,5% (teste 4) e 25% (teste 5) do PWM podem ser vistas na Figura 27 e na Figura 28, respectivamente.

Figura 27 – Curva de resposta considerando aplicação de PWM com ciclo de trabalho de 37,5%.



Fonte: De autoria própria.

Figura 28 – Curva de resposta considerando aplicação de PWM com ciclo de trabalho de 25%.



Fonte: De autoria própria.

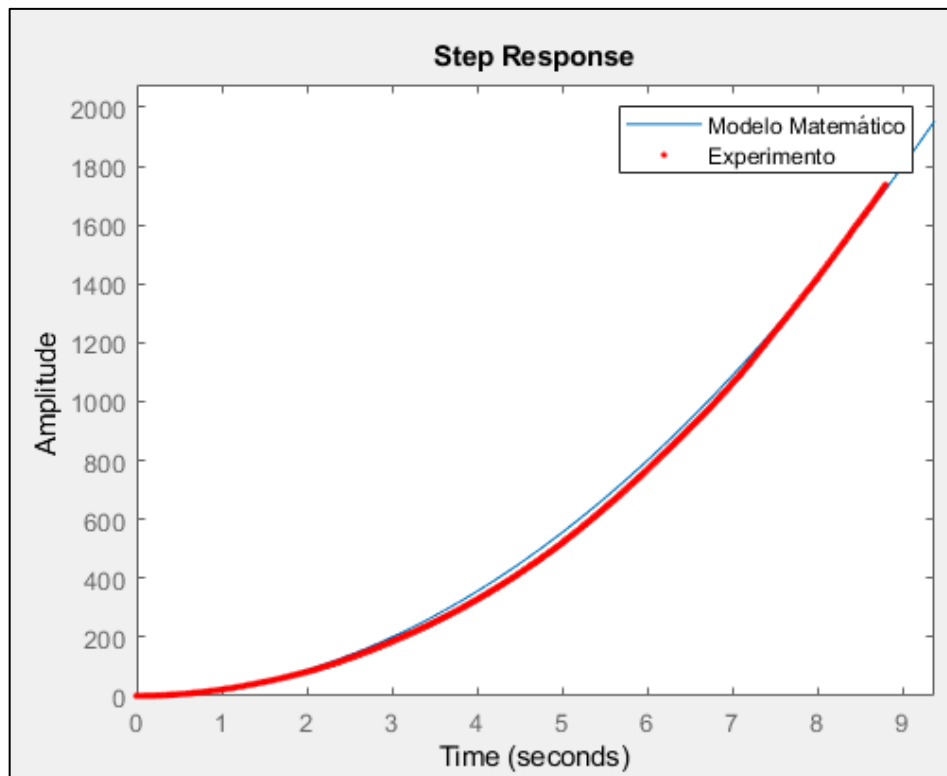
O próximo passo, é estimar, com base nos ensaios experimentais, os parâmetros não conhecidos da função de transferência da planta (Equação (2)). Para isso, utilizou-se como base os pontos de posição angular obtidos no teste 3 como saída para uma entrada degrau de 127 (que equivale a ciclo de trabalho de 50% do PWM). Para o desenvolvimento deste trabalho, será considerada uma ação de controle normalizada. Isso significa que, a ação de controle será um número entre 0 e 1 e será utilizado um escalonamento de 127 para colocar essa ação de controle na base de 0 a 127, faixa de atuação do PWM que será utilizada neste trabalho.

Após a sintonia do parâmetro desconhecido, verificou-se que a planta do sistema é dada por:

$$G(s) = \underbrace{127}_{\text{Normalização do PWM}} \cdot \frac{0,35}{s^2} \quad (3)$$

A validação dessa planta é apresentada na Figura 29, onde são apresentadas a resposta ao degrau da planta experimental e a resposta ao degrau normalizado para o modelo.

Figura 29 – Comparação entre respostas ao degrau de ciclo de trabalho de 50% do PWM para planta modelada e para a planta experimental.



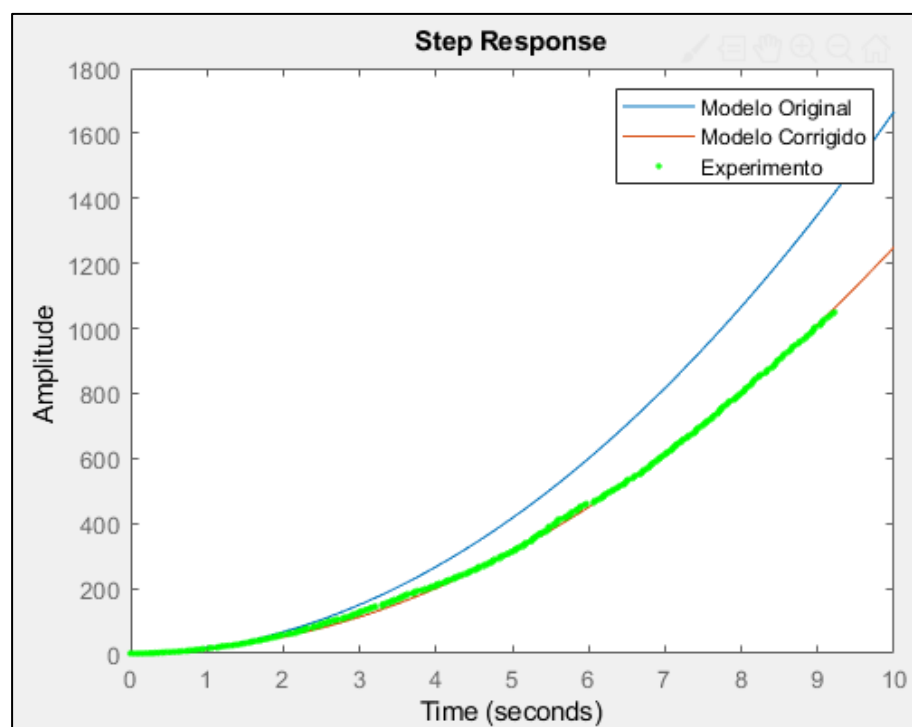
Fonte: De autoria própria.

## 4.2 Validação da Planta

Validar a dinâmica da planta obtida é de extrema importância para a continuidade do projeto de controle, uma vez que esse será feito com base na função de transferência da planta, tornando imprescindível que a representação matemática do sistema seja o mais fiel possível à realidade. Apesar da planta já ter sido validada para o ponto de operação do teste 3, é importante avaliar se ela também é válida para os outros dois pontos de operação obtidos na Subseção 4.1 (referentes aos testes 4 e 5). Ao fazer isso, será possível avaliar se a planta é linear ou não-linear.

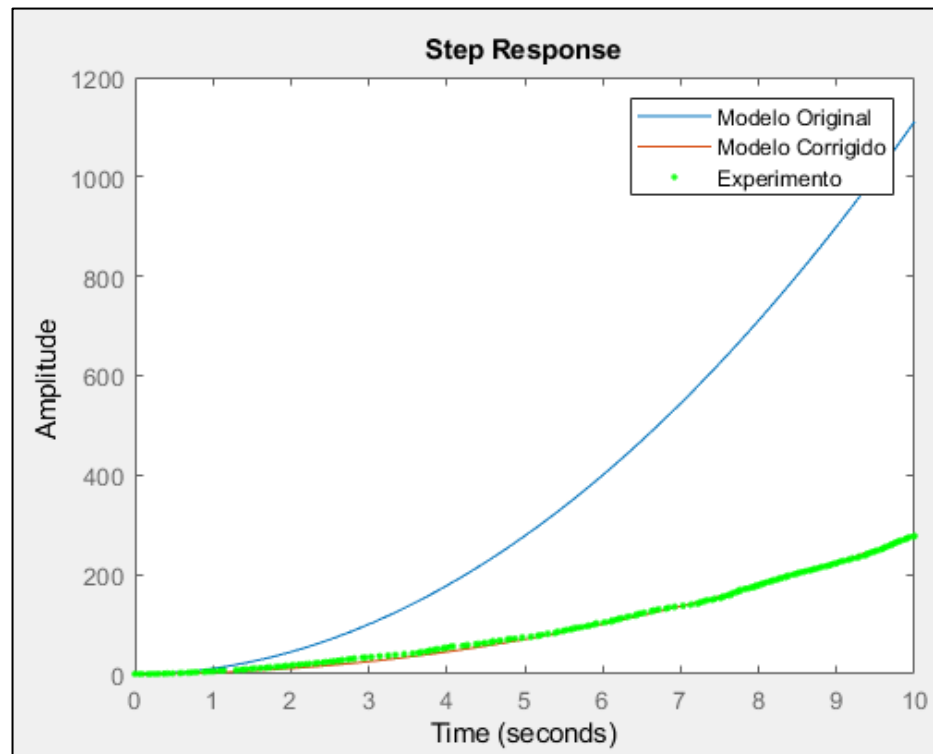
Para essa validação, é importante que o leitor perceba que o ciclo de trabalho dos testes 4 e 5 representam, respectivamente, 75% e 50% do ciclo de trabalho do teste 3. Como o modelo está normalizado, isso significa que para obter um resultado similar ao do teste 4, seria necessário aplicar uma ação de controle de 0,75 na planta modelada pela Equação (3). Analogamente, seria necessário aplicar uma ação de controle de 0,5 na planta modelada pela Equação (3) para obter um resultado similar ao apresentado no teste 5. Nesse contexto, a Figura 30 e a Figura 31 apresentam as respostas ao degrau obtidas experimentalmente e para o modelo, considerando os testes 4 e 5, respectivamente.

Figura 30 – Validação do modelo apresentado na Equação (3) para o cenário avaliado no teste 4.



Fonte: De autoria própria.

Figura 31 – Validação do modelo apresentado na Equação (3) para o cenário avaliado no teste 5.



Fonte: De autoria própria.

As análises da Figura 30 e da Figura 31 mostram que, conforme diminui-se a amplitude do degrau aplicado na entrada da planta, menos compatível fica a resposta ao degrau obtida no experimento em relação ao modelo original (obtido a partir do teste 3). Ao variar o ganho estático do modelo original, observou-se que era possível chegar nas curvas experimentais para os dois cenários de teste, contudo, usando fatores de correção diferentes para cada um dos casos. Isso é uma evidência de que a planta possui uma natureza não-linear da planta. Nesse caso em específico, os polos do modelo estão corretamente modelados, apenas o ganho estático da planta que é não-linear. É importante perceber que essa natureza não-linear já era esperada, visto que o atuador do sistema é um motor CC que já tem características não-lineares.

Por conta da natureza não-linear do sistema, a literatura clássica sugere linearizar a planta em torno de um ponto de operação em que se pretende trabalhar. Porém, como a diferença mais significativa entre os três pontos de operação avaliados se deu em termos do ganho estático da função de transferência, optou-se por fazer uma escolha mais conservadora. Assim, optou-se por manter o uso da função de transferência que apresentava maior ganho estático como modelo da planta (Equação (3)).

### 4.3 Controle da Planta

Este tópico é um dos pontos focais deste trabalho, pois aqui é abordado o desenvolvimento do controlador da planta. Primeiramente, é necessário definir os requisitos de controle do sistema, ou seja, o máximo sobressinal, o erro máximo e o tempo de acomodação máximo permitidos. Com isso, o próximo passo é fazer o projeto do controlador, definindo os ganhos  $K_p$ ,  $K_i$  e  $K_d$  do mesmo.

#### 4.3.1 Definição dos Requisitos de Controle

Definir os requisitos de controle é de fundamental importância ao realizar um projeto de controle, pois são estes requisitos que definem o objetivo final a ser alcançado, dando um panorama geral sobre o sistema, ou seja, se ele está com um bom controle e se atende de forma satisfatória aos comandos solicitados.

Dentre os principais requisitos a serem definidos estão, o erro com relação ao *SetPoint* em regime permanente, o máximo sobressinal permitido e o máximo tempo de acomodação do sistema. Para defini-los, é necessário primeiro entender quais são as aplicações e as necessidades deste projeto. No caso de *CubeSats*, o parâmetro mais crítico é o erro no apontamento final, onde a tolerância para erros com relação ao *SetPoint* pode variar bastante a depender do tipo de missão que ele realizará. Missões de observação da Terra, por exemplo, podem exigir mais precisão de apontamento, já em missões de comunicação este erro pode ter uma tolerância um pouco maior. No geral, o máximo erro absoluto permitido pode variar bastante, indo desde alguns poucos graus até frações de um grau (FABRI, 1997).

Para o nano satélite desenvolvido para este trabalho, adotou-se como tolerável um erro de apontamento angular dentro de uma faixa de  $\pm 5^\circ$  com relação ao *SetPoint*. Já para os requisitos de máximo sobressinal e tempo de acomodação, apesar de não serem uma preocupação tão grande para este projeto, adotou-se como máximo sobressinal aceitável, um valor dentro de uma faixa  $\pm 20^\circ$  e um tempo de acomodação de até 20 segundos. Estes valores foram definidos principalmente com base nas limitações de leitura e calibração do acelerômetro MPU 6050 e nas limitações e atrasos de leitura dos dados por parte do módulo *bluetooth*. A Tabela 1 traz de forma compacta todos estes requisitos definidos.

Tabela 1 – Requisitos de controle definidos neste TCC.

| Parâmetro           | Tolerância    |
|---------------------|---------------|
| Erro                | $\pm 5^\circ$ |
| Máximo Sobressinal  | 25%           |
| Tempo de Acomodação | 8 segundos    |

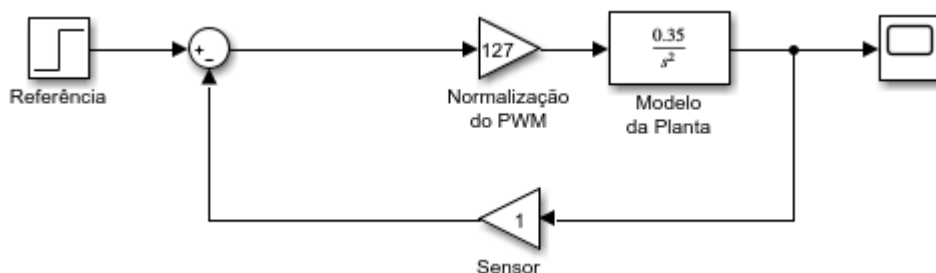
Fonte: De autoria própria.

#### 4.3.2 Projeto de Controle

O projeto de controle foi iniciado com uma avaliação do sistema de controle em malha fechada, considerando inicialmente controlador unitário. Para isso, é necessário definir a função de transferência que representa o bloco sensor, presente na realimentação do sistema. Sabe-se que os dados fornecidos pelo usuário e interpretados pelo microcontrolador para que seja feito o controle, devem ser ângulos em unidade de graus.

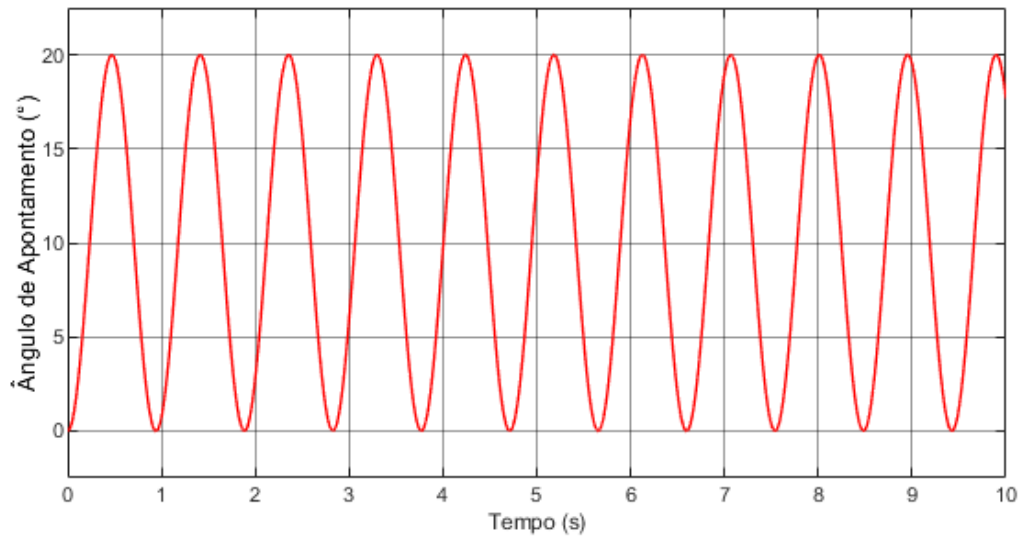
O programa desenvolvido no módulo computacional utiliza uma biblioteca chamada de “*MPU6050\_light.h*”, que é capaz de fazer todo o tratamento de dados fornecidos pelo acelerômetro MPU 6050, e fornecer para o usuário a posição angular em diretamente em graus através da função “*mpu.getAngleZ()*”. Isto implica em dizer que a função matemática que representa o sensor, é dada simplesmente por  $H(z) = 1$ , já que a realimentação de ângulo, recebida pelo controlador através do sensor, vem diretamente na unidade correta (graus) e não precisa ser tratado. Assim, considerando a Equação (3) da planta do sistema, realizou-se a simulação mostrada na Figura 32 do sistema em malha fechada no *Simulink*, com realimentação unitária. A curva obtida nesta simulação para uma mudança da referência em degrau de  $0^\circ$  para  $10^\circ$  pode ser vista na Figura 33.

Figura 32 – Diagrama de blocos do sistema em malha fechada com controlador unitário.



Fonte: De autoria própria.

Figura 33 - Resposta a um degrau (mudança da referência de  $0^\circ$  para  $10^\circ$ ) para o sistema em malha fechada com controlador unitário.

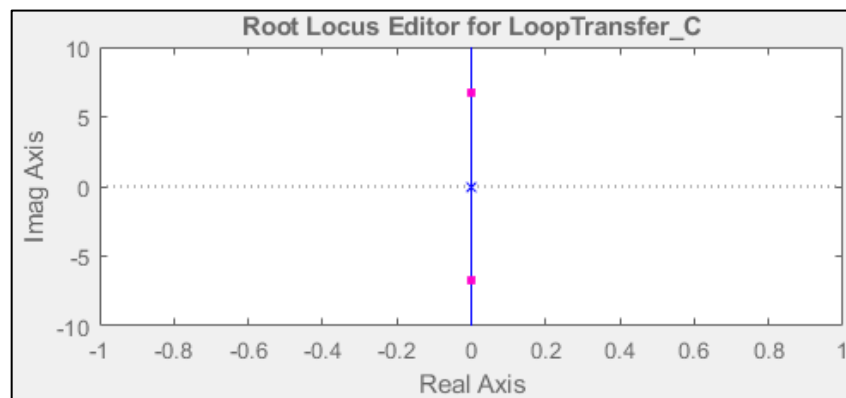


Fonte: De autoria própria.

A partir da análise do resultado obtido na Figura 33, percebe-se que a planta apresenta uma oscilação sustentada, com sobressinal de 100%. Em decorrência disso, pode-se inferir que será necessário o uso de um controlador do tipo PD para realizar o controle da planta.

Com o auxílio da ferramenta “*rltool*” (também conhecida como *sisotool*) do MatLab, iniciou-se a avaliação do uso de um controlador puramente proporcional para esse sistema. Ao observar o LGR (Lugar Geométrico das Raízes), exibido na Figura 34, observa-se que, independentemente do ganho proporcional escolhido, esse sistema possuirá polos complexos conjugados alocados sobre o eixo imaginário, indicando que a oscilação sustentada se manterá para qualquer ganho proporcional selecionado (o sistema seria marginalmente estável).

Figura 34 – LGR quando usando um controlador do tipo P (tempo contínuo) – Ensaio 1.



Fonte: De autoria própria.

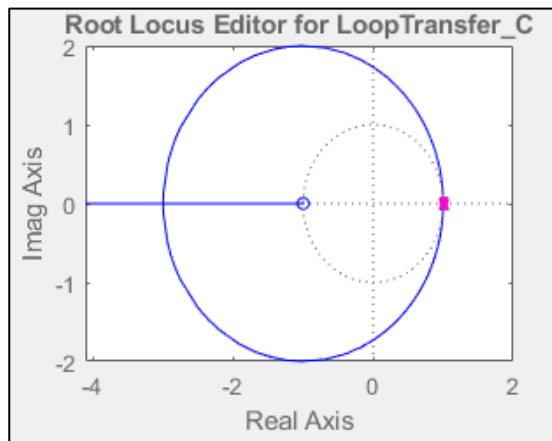


Para viabilizar o projeto do controlador diretamente do domínio discreto, utilizou-se a função “c2d” do *Matlab* para discretizar a planta. Nesse processo, foi utilizado o método ZOH (*zero order holder*), dado que seu modelo matemático se assemelha ao modelo matemático de um modulador PWM (SIMONE BUSO, 2006). Ao fazer isso, obteve-se a seguinte função de transferência:

$$G(z) = 0,01076 \cdot \frac{(z + 1)}{(z - 1)^2} \quad (4)$$

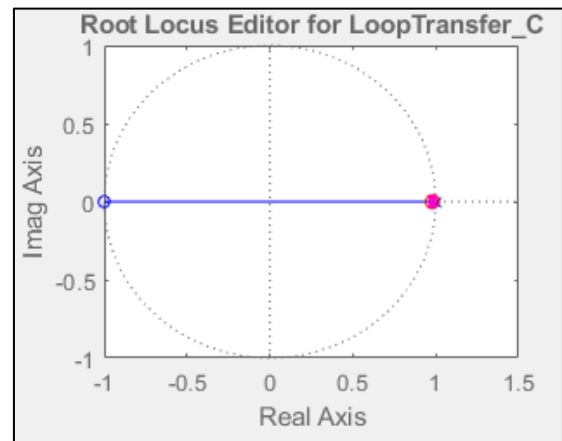
cujo LGR é exibido na Figura 35.

Figura 35 – LGR quando usando um controlador do tipo P (tempo discreto) – Ensaio 1.



Fonte: De autoria própria.

Figura 36 – LGR quando usando um controlador do tipo PD (tempo discreto) – Ensaio 1.



Fonte: De autoria própria.

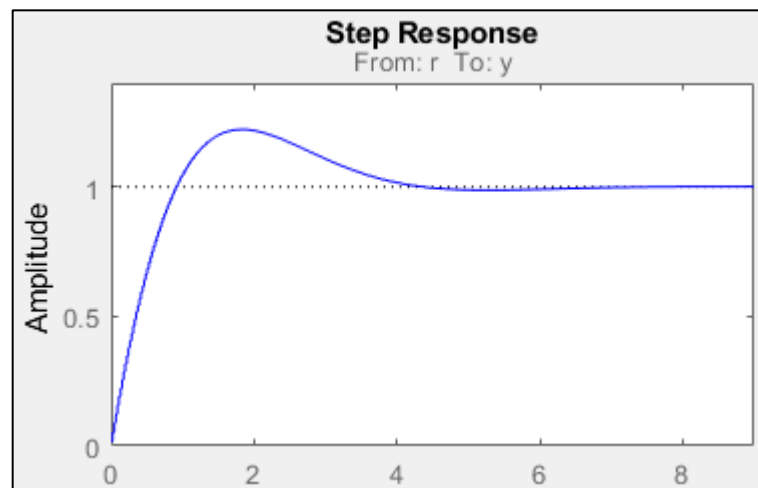
Da Figura 35, observa-se que o controlador proporcional não satisfaz os critérios de projeto que diz respeito ao erro de *SetPoint*, transitório e máximo sobressinal (o sistema fica instável ao considerar a planta discreta). Portanto, optou-se por adicionar um zero em  $z = 0.98$ , de modo a fazer o LGR ficar inteiramente contido na região de estabilidade definida pela circunferência de raio unitário (visto que o sistema está sendo avaliado no domínio discreto). O LGR obtido ao adicionar o zero descrito acima é exibido na Figura 36.

Contudo, ao adicionar o zero, o sistema passou a ter mais zeros do que polos, tornando o controlador em não-causal, logo, não implementável. Para resolver isso, é adicionado um polo em  $z = 0$ , que equivale a adicionar um polo no infinito. Considerando que a dinâmica mais lenta é predominante sobre a dinâmica mais rápida do sistema, esse polo não alterará a resposta do sistema, porém, irá torna-lo

implementável. Essa configuração utilizada é equivalente a um “derivador implementável”.

Além do zero e do polo adicionados, o ganho do controlador foi sintonizado em 1,7 para selecionar o tempo de resposta do sistema. A resposta ao degrau obtida no *rltool* para esse controlador é exibida na Figura 37.

Figura 37 – Resposta ao degrau para o sistema em malha fechada considerando um controlador tipo PD – Ensaio 2.



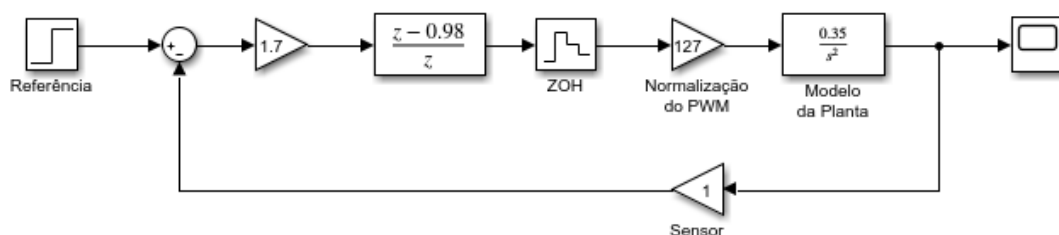
Fonte: De autoria própria.

A partir da análise da Figura 37 e Figura 39, percebe-se uma melhora significativa no transitório do sistema (como esperado ao adicionar um ganho derivativo), diminuindo as oscilações, o tempo de acomodação e o máximo sobressinal. A função de transferência do controlador é dada por:

$$C(z) = 1,7 \cdot \frac{(z - 0,98)}{z} \quad (5)$$

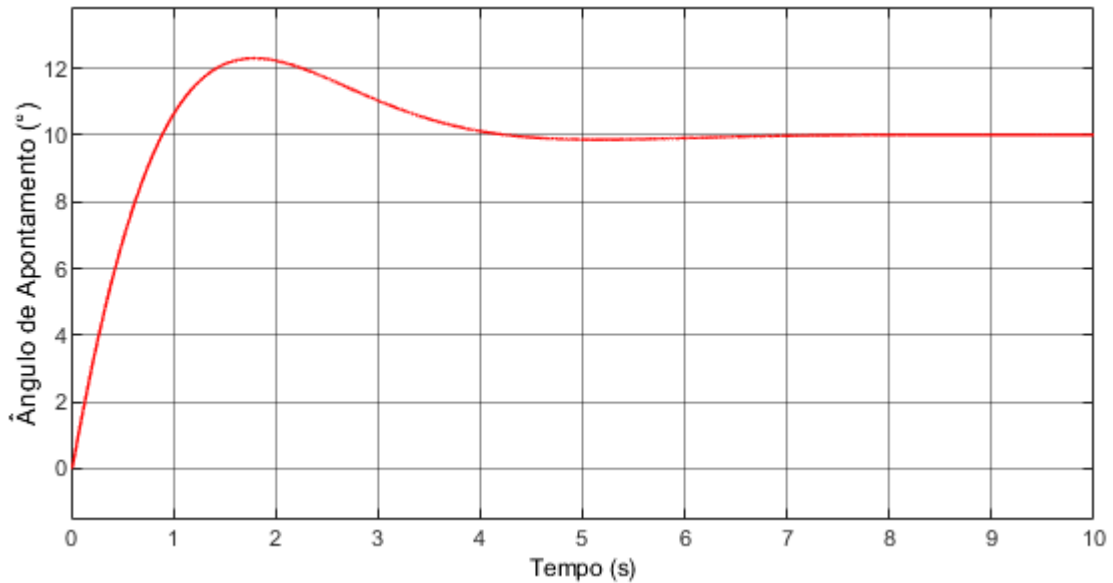
Para avaliar a resposta do sistema controlado, foi introduzido o controlador no diagrama do sistema em malha fechada feito no *Simulink*, conforme mostrado na Figura 38. A resposta obtida para o diagrama descrito acima é mostrada na Figura 39.

Figura 38 – Diagrama do sistema em malha fechada, representado no *Simulink*.



Fonte: De autoria própria.

Figura 39 – Resposta a um degrau (mudança da referência de 0° para 10°) para o sistema em malha fechada com controlador PD.



Fonte: De autoria própria.

Com o controlador definido, o próximo passo é obter a equação diferença deste controlador para então implementá-la no microcontrolador da plataforma de testes. Para se obter a equação diferença, primeiramente é preciso representar a função de transferência do controlador em termos de atrasos, para torná-lo implementável. Dessa forma, a função de transferência do controlador pode ser representada como:

$$C(z) = \frac{U(z)}{E(z)} = 1,7 \cdot \frac{(z - 0,98)}{z} \cdot \frac{z^{-1}}{z^{-1}} \quad (6)$$

ou seja:

$$\frac{U(z^{-1})}{E(z^{-1})} = 1,7 \cdot \frac{(1 - 0,98z^{-1})}{1} \quad (7)$$

A partir da Equação (7), pode-se manipular a função de transferência do controlador para isolar o termo  $U(z^{-1})$ . Ao fazer isso, obtém-se a seguinte equação:

$$U(z^{-1}) = 1,7 \cdot E(z^{-1}) - 0,98 \cdot E(z^{-1}) \cdot z^{-1} \quad (8)$$

Esta manipulação é feita para que seja encontrada a equação diferença da saída do controlador. Com isso, basta aplicar a transformada Z inversa na equação (8) para se obter a expressão final da equação diferença, que é dada por:

$$u[n] = 1,7 \cdot e[n] - 0,98 \cdot e[n - 1] \quad (9)$$

onde  $e[n - k]$  representa o valor do erro na  $k$  – ésima iteração anterior.

A Figura 40 mostra uma parte retirada do código em C onde foi implementado o controlador.

Figura 40 – Implementação do Controlador no Microcontrolador

```

105 // Calcula o erro
106 double error = setpoint - angleZ;
107
108 // Calcula o controle total
109 double control = 1.7*error - 0.98*prevError;
110
111 // Armazena o erro do ciclo atual para o próximo ciclo
112 prevError = error;

```

Fonte: De autoria própria.

Com o controlador pronto, o próximo passo é fazer os testes de apontamento utilizando o *software* de validação e a plataforma de testes, de modo a comparar os resultados reais obtidos com o que é esperado a partir da simulação.

#### 4.4 Validação Experimental utilizando a plataforma de testes

Na interface homem máquina desenvolvida pelo autor, existem 6 possibilidades diferentes de *Setpoint* de ângulos, conforme mostrado na Figura 21. Com isto, foram feitos 7 testes de apontamento, 6 para cada um dos *SetPoints* diferentes e 1 último envolvendo uma perturbação, que será abordado com mais detalhes na Subseção 4.5.

Assim, com a plataforma de testes pronta, o código final implementado no microcontrolador e o *software* de validação feito, iniciaram-se os testes. Os testes foram feitos na seguinte ordem:

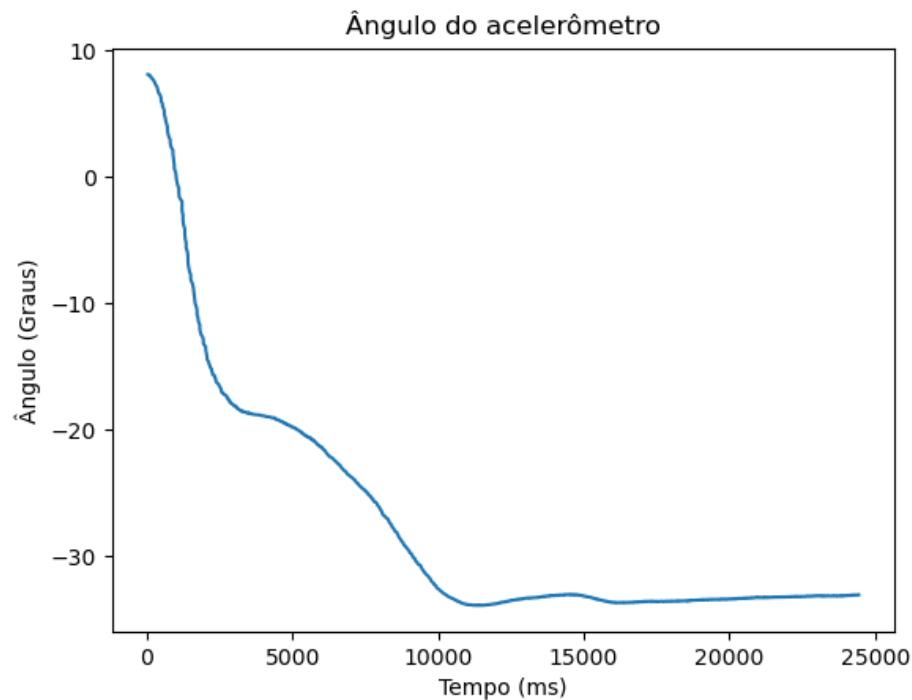
1. Mudança de 0 grau para -30 graus;
2. Mudança de 0 grau para 30 graus;
3. Mudança de 0 grau para -60 graus;
4. Mudança de 0 grau para 60 graus;

5. Mudança de 0 grau para -90 graus;
6. Mudança de 0 grau para 90 graus.

Todos esses 6 testes foram feitos com o *CubeSat* inicialmente na posição aproximada de zero graus, ou seja, partindo do zero até o *SetPoint* escolhido. O valor aproximado de zero, se deve pelo fato do acelerômetro quase nunca indicar um ângulo exatamente igual a zero na posição inicial, mas sim o valor mais próximo alcançado na situação.

Para o primeiro teste, com *SetPoint* de -30 graus, a Interface Gráfica forneceu a curva mostrada na Figura 41. Os dados finais fornecidos pela Interface são mostrados na Figura 42.

Figura 41 – Curva de resposta do *CubeSat* a um *SetPoint* de -30 graus.



Fonte: De autoria própria.

Figura 42 – Dados da Interface Gráfica para um *SetPoint* de -30 graus no *CubeSat*.

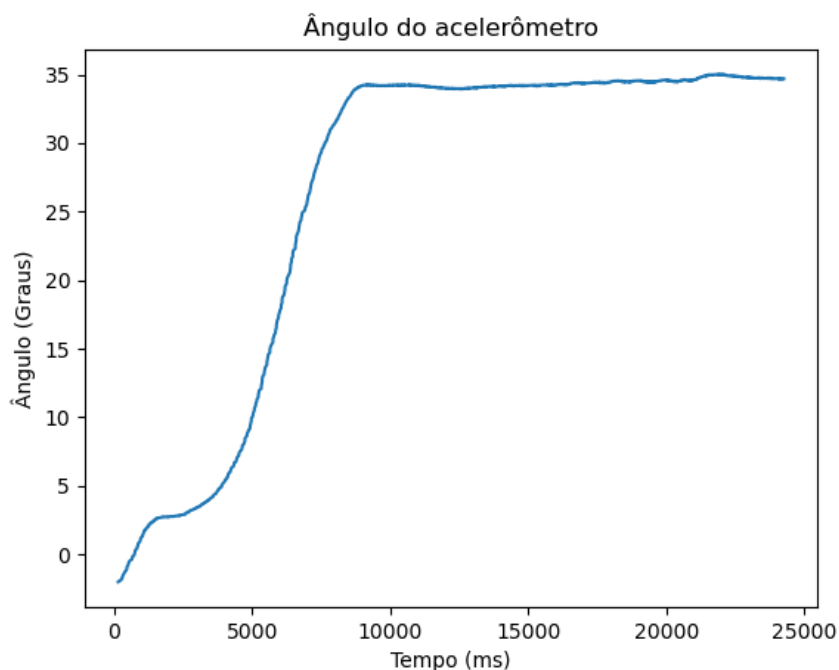
|                                              |           |          |           |          |           |          |
|----------------------------------------------|-----------|----------|-----------|----------|-----------|----------|
| Enviar SetPoint:                             | -30 Graus | 30 Graus | -60 Graus | 60 Graus | -90 Graus | 90 Graus |
| Posição Angular do CubeSat (Graus): -33.07 ° |           |          |           |          |           |          |
| SetPoint (Graus): -30.00 °                   |           |          |           |          |           |          |
| Erro (Graus): 3.07 °                         |           |          |           |          |           |          |

Fonte: De autoria própria.

Com base na Figura 41, percebe-se que o sistema apresentou um pequeno sobressinal e teve um tempo de acomodação de cerca de 10 segundos. Já a Figura 42, indica que o sistema teve um erro absoluto de aproximadamente 3 graus.

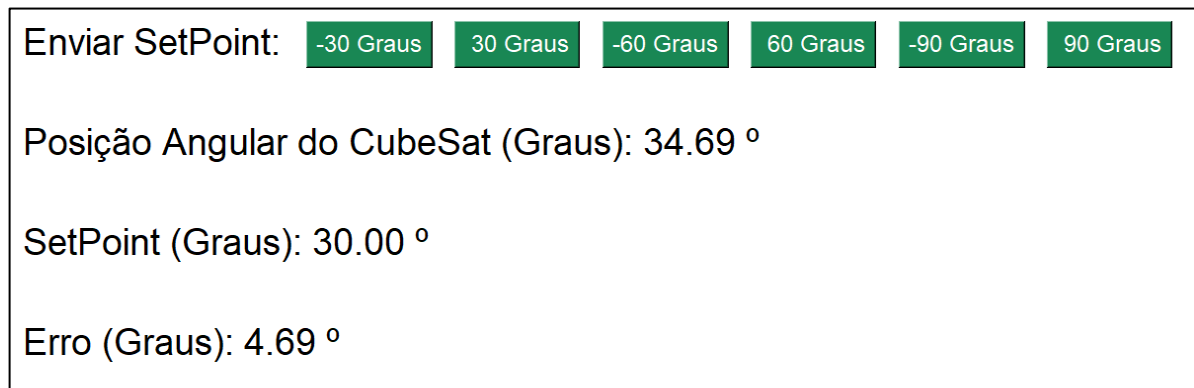
Para o segundo teste, com *SetPoint* de 30 graus, a Interface Gráfica forneceu a curva mostrada na Figura 43. Os dados finais fornecidos pela Interface são mostrados na Figura 44.

Figura 43 – Curva de resposta do *CubeSat* a um *SetPoint* de 30 graus.



Fonte: De autoria própria.

Figura 44 – Dados da Interface Gráfica para um *SetPoint* de 30 graus no *CubeSat*.

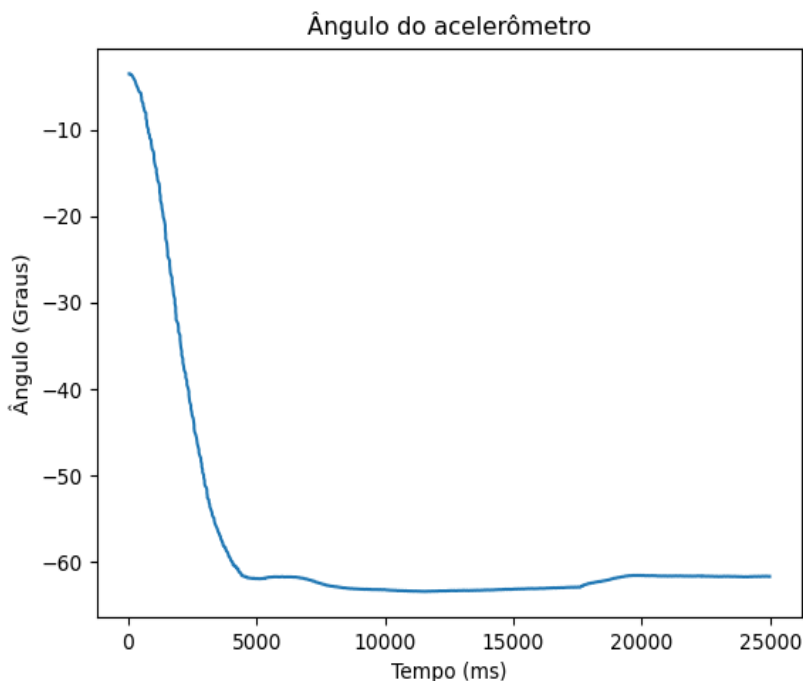


Fonte: De autoria própria.

Com base na Figura 43, percebe-se que o sistema quase não apresentou sobressinal com relação ao *SetPoint* final e teve um tempo de acomodação um pouco menor que 10 segundos. Já a Figura 44, indica que o sistema teve um erro absoluto de quase 5 graus.

Para o terceiro teste, com *SetPoint* de -60 graus, a Interface Gráfica forneceu a curva mostrada na Figura 45. Os dados finais fornecidos pela Interface são mostrados na Figura 46.

Figura 45 – Curva de resposta do *CubeSat* a um *SetPoint* de -60 graus.



Fonte: De autoria própria.

Figura 46 – Dados da Interface Gráfica para um *SetPoint* de -60 graus no *CubeSat*.

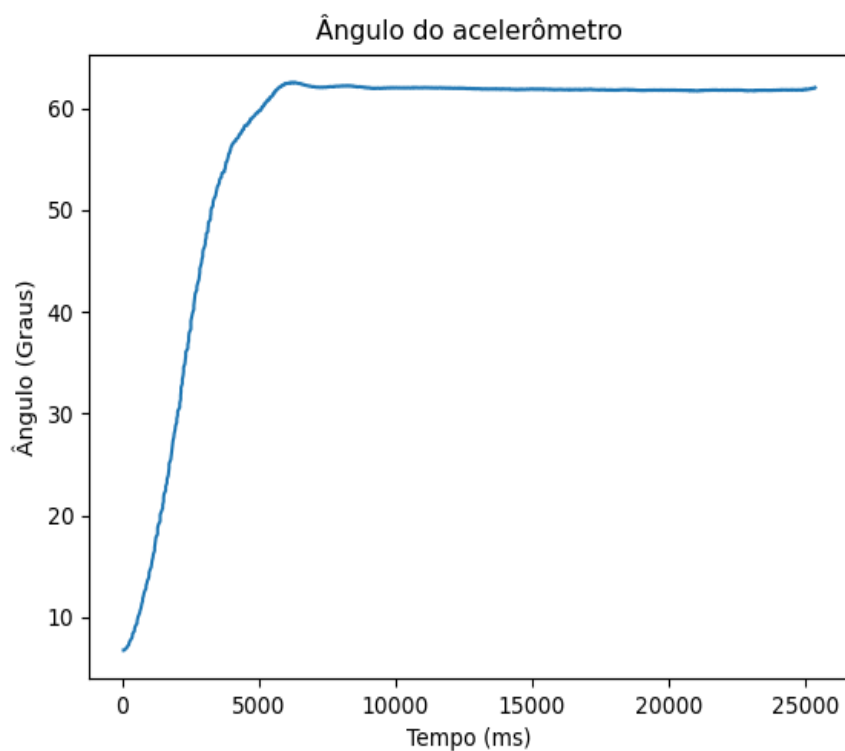
|                                              |           |          |           |          |           |          |
|----------------------------------------------|-----------|----------|-----------|----------|-----------|----------|
| Enviar SetPoint:                             | -30 Graus | 30 Graus | -60 Graus | 60 Graus | -90 Graus | 90 Graus |
| Posição Angular do CubeSat (Graus): -61.69 ° |           |          |           |          |           |          |
| SetPoint (Graus): -60.00 °                   |           |          |           |          |           |          |
| Erro (Graus): 1.69 °                         |           |          |           |          |           |          |

Fonte: De autoria própria.

Com base na Figura 45, percebe-se que o sistema quase não apresentou sobressinal com relação ao *SetPoint* final e teve um tempo de acomodação inferior a 5 segundos. Já a Figura 46, indica que o sistema teve um erro absoluto inferior a 2 graus.

Para o quarto teste, com *SetPoint* de 60 graus, a Interface Gráfica forneceu a curva mostrada na Figura 47. Os dados finais fornecidos pela Interface são mostrados na Figura 48.

Figura 47 – Curva de resposta do *CubeSat* a um *SetPoint* de 60 graus.



Fonte: De autoria própria.



Figura 48 – Dados da Interface Gráfica para um *SetPoint* de 60 graus no *CubeSat*.

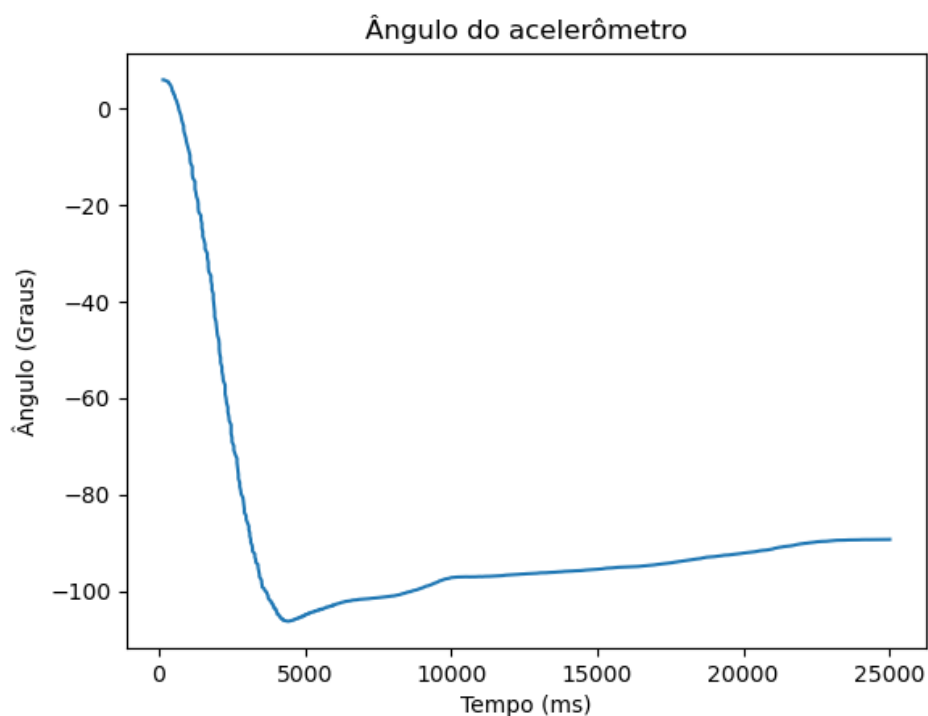
|                                             |           |          |           |          |           |          |
|---------------------------------------------|-----------|----------|-----------|----------|-----------|----------|
| Enviar SetPoint:                            | -30 Graus | 30 Graus | -60 Graus | 60 Graus | -90 Graus | 90 Graus |
| Posição Angular do CubeSat (Graus): 61.99 ° |           |          |           |          |           |          |
| SetPoint (Graus): 60.00 °                   |           |          |           |          |           |          |
| Erro (Graus): 1.99 °                        |           |          |           |          |           |          |

Fonte: De autoria própria.

Com base na Figura 47, percebe-se que o sistema apresentou um sobressinal bastante pequeno com relação ao *SetPoint* final e teve um tempo de acomodação de cerca de 5 segundos. Já a Figura 48, indica que o sistema teve um erro absoluto de cerca de 2 graus.

Para o quinto teste, com *SetPoint* de -90 graus, a Interface Gráfica forneceu a curva mostrada na Figura 49. Os dados finais fornecidos pela Interface são mostrados na Figura 50.

Figura 49 – Curva de resposta do *CubeSat* a um *SetPoint* de -90 graus.



Fonte: De autoria própria.

Figura 50 – Dados da Interface Gráfica para um *SetPoint* de -90 graus no *CubeSat*.

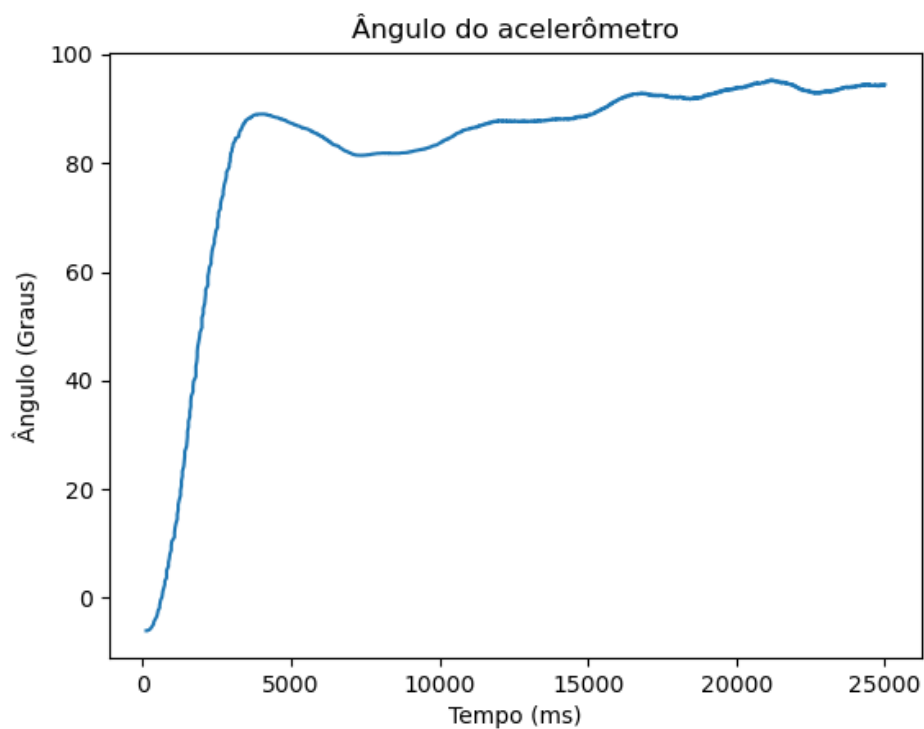
|                                              |           |          |           |          |           |          |
|----------------------------------------------|-----------|----------|-----------|----------|-----------|----------|
| Enviar SetPoint:                             | -30 Graus | 30 Graus | -60 Graus | 60 Graus | -90 Graus | 90 Graus |
| Posição Angular do CubeSat (Graus): -89.24 ° |           |          |           |          |           |          |
| SetPoint (Graus): -90.00 °                   |           |          |           |          |           |          |
| Erro (Graus): 0.76 °                         |           |          |           |          |           |          |

Fonte: De autoria própria.

Com base na Figura 49, percebe-se que o sistema apresentou um sobressinal considerável, com uma diferença de cerca de 20 graus com relação ao *SetPoint* final e teve um tempo de acomodação entre 10 e 15 segundos. Já a Figura 50, indica que o sistema teve um erro absoluto inferior a 1 grau.

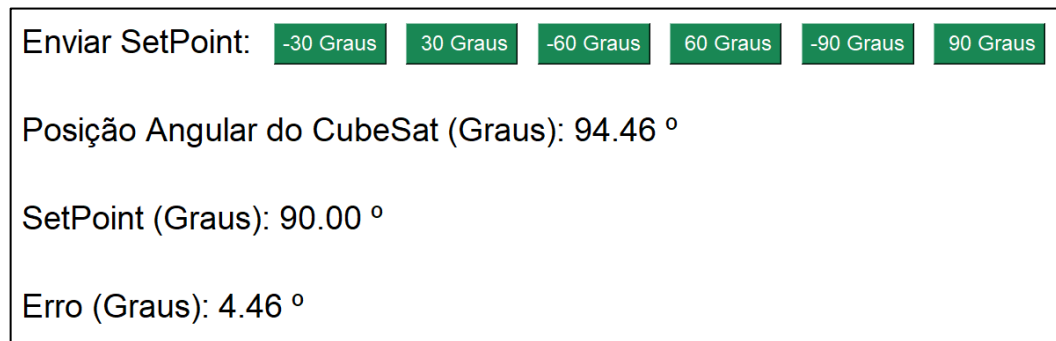
Para o sexto teste, com *SetPoint* de 90 graus, a Interface Gráfica forneceu a curva mostrada na Figura 51. Os dados finais fornecidos pela Interface são mostrados na Figura 52.

Figura 51 – Curva de resposta do *CubeSat* a um *SetPoint* de 90 graus.



Fonte: De autoria própria.

Figura 52 – Dados da Interface Gráfica para um *SetPoint* de 90 graus no *CubeSat*.



Fonte: De autoria própria.

Com base na Figura 51, percebe-se que o sistema apresentou um pequeno sobressinal com relação ao *SetPoint* final e teve um tempo de acomodação de cerca de 15 segundos. Já a Figura 52, indica que o sistema teve um erro absoluto de cerca de 4 graus.

Primeiramente, é importante ressaltar que o sistema apresentou repetibilidade para mesmos valores de *setpoint*, retificando a confiabilidade dos ensaios realizados. A partir da análise destes 6 testes, observa-se que os critérios de erro em regime permanente e sobressinal foram atendidos em todos os casos. Todavia, alguns testes apresentaram tempo de assentamento maior do que o especificado. Isso se deve porque o motor CC tem uma zona morta, que acaba afetando o sistema quando este precisa fazer deslocamentos menores, visto que uma menor ação de controle seria necessária. Além disso, na Subseção 4.2, foi indicado que quando aplicadas ações de controle mais baixas que a máxima, o ganho estático do sistema diminuía. Como consequência, é naturalmente esperado que o tempo de assentamento aumente.

Para os apontamentos de -60 e 60 graus, por exemplo, percebeu-se que o tempo de acomodação foi o menor dentre os testes, ambos apresentando um tempo de cerca de 5 segundos (próximo ao esperado de simulação), assim como erros de *SetPoint* não superiores a 2 graus. Já para os apontamentos de -30 e 30 graus, o tempo de acomodação foi um pouco maior, girando em torno 10 segundos para ambos, assim como também o erro, que variou de 3 a 5 graus. Para os apontamentos de -90 e 90 graus, houve um tempo de acomodação em torno de 5 a 10 segundos. O apontamento de -90 graus, apesar de ter apresentado o menor erro de todos (inferior a 1 grau), teve um sobressinal elevado (cerca de 20 graus), já o de 90 graus, teve um erro bem maior (cerca de 4 graus), porém um sobressinal pequeno.

Estes resultados mostram que o sistema se comportou melhor para ângulos intermediários em valores absolutos, como 60 graus, tanto para erros como para tempo de acomodação e máximo sobressinal. Para ângulos menores em módulo, como 30 graus, era esperado que o sistema tivesse um tempo de acomodação maior, pois como o *SetPoint* possui um valor mais baixo, o sinal de controle também, implicando em mais tempo para o motor romper a inércia imposta pela roda de reação e começar a girar. Para ângulos maiores em módulo, como 90 graus, era esperado maiores valores de sobressinal, visto que o motor recebe um alto sinal de controle, fazendo o *CubeSat* girar em alta velocidade, acarretando também em um maior tempo de acomodação, que foi o que de fato aconteceu para estes casos.

Vídeos apresentando o funcionamento desses 6 testes estão disponíveis nos *links* apresentados na Tabela 2.

Tabela 2 - *Links* para vídeos do funcionamento dos 6 testes experimentais.

| Teste                            | Link                                                                                                                    |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Mudança de 0 grau para -30 graus | <a href="https://youtube.com/shorts/aENkOHZkae4?feature=share">https://youtube.com/shorts/aENkOHZkae4?feature=share</a> |
| Mudança de 0 grau para 30 graus  | <a href="https://youtube.com/shorts/t8SFsUK71v0?feature=share">https://youtube.com/shorts/t8SFsUK71v0?feature=share</a> |
| Mudança de 0 grau para -60 graus | <a href="https://youtube.com/shorts/tJ3N_Z0Du04?feature=share">https://youtube.com/shorts/tJ3N_Z0Du04?feature=share</a> |
| Mudança de 0 grau para 60 graus  | <a href="https://youtube.com/shorts/2KJylwVBqCQ">https://youtube.com/shorts/2KJylwVBqCQ</a>                             |
| Mudança de 0 grau para -90 graus | <a href="https://youtube.com/shorts/W3Vv1saKJ_U">https://youtube.com/shorts/W3Vv1saKJ_U</a>                             |
| Mudança de 0 grau para 90 graus  | <a href="https://youtube.com/shorts/W_LugbJJia0">https://youtube.com/shorts/W_LugbJJia0</a>                             |

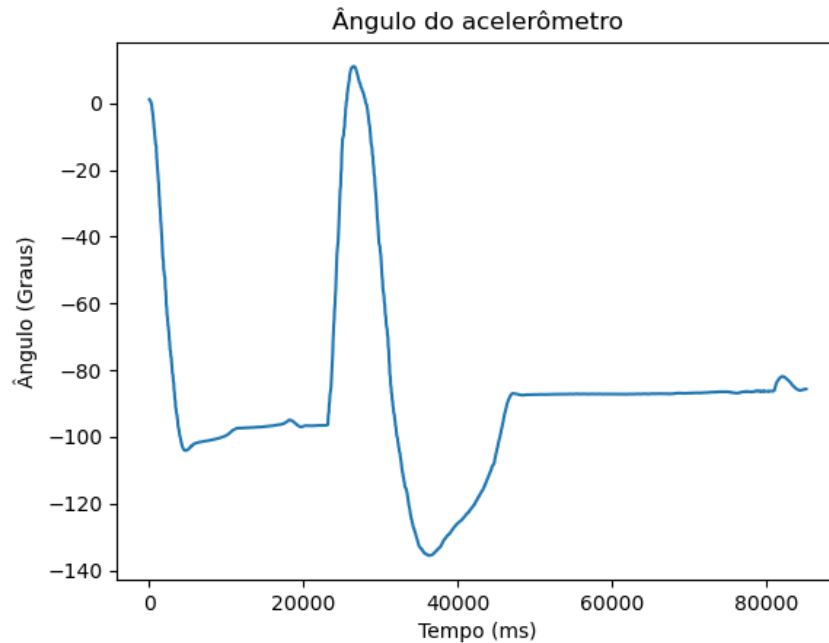
Fonte: De autoria própria.

#### 4.5 Avaliação de diferentes cenários

Para avaliar como o sistema se comporta em diferentes situações, ou seja, a algo que não seja simplesmente obedecer a um comando de *SetPoint* enviado pelo usuário, foi feito um teste aplicando uma perturbação no *CubeSat*. O teste consistiu em enviar um *SetPoint* de ângulo de -90 graus, esperar o sistema atingir o regime e então, rotacionar o *CubeSat* com a mão em torno do eixo *yaw* para outra posição angular, ou seja, aplicar uma perturbação. A escolha pelo *SetPoint* de -90 graus, se deve ao fato de este ter sido o teste que apresentou o menor erro de regime permanente (vide Figura 50).

A curva fornecida pela Interface Gráfica para este teste pode ser vista na Figura 53. Os dados finais fornecidos pela Interface são mostrados na Figura 54.

Figura 53 – Comportamento do sistema quando sujeito a uma perturbação. O sistema deve operar em regime em  $-90^\circ$ .



Fonte: De autoria própria.

Figura 54 - Dados da Interface Gráfica do sistema para o teste de Perturbação.

|                                              |           |          |           |          |           |          |
|----------------------------------------------|-----------|----------|-----------|----------|-----------|----------|
| Enviar SetPoint:                             | -30 Graus | 30 Graus | -60 Graus | 60 Graus | -90 Graus | 90 Graus |
| Posição Angular do CubeSat (Graus): -85.66 ° |           |          |           |          |           |          |
| SetPoint (Graus): -90.00 °                   |           |          |           |          |           |          |
| Erro (Graus): 4.34 °                         |           |          |           |          |           |          |

Fonte: De autoria própria.

Analisando a Figura 53, percebe-se que o sistema teve um sobressinal considerável e um tempo de acomodação entre 10 e 20 segundos. Ao movimentá-lo com a mão, o *CubeSat* foi levado a uma posição angular próxima de zero graus, mas

rapidamente reagiu a esta perturbação, de modo a retornar ao -90 graus. Entretanto, percebe-se que ao retornar para o estado de regime após a perturbação, o satélite apresentou um sobressinal bastante elevado, chegando a quase 50 graus de diferença, que ocorreu devido à alta rotação imposta pelo motor durante a perturbação, como forma de reagir a mesma, resultando em um movimento muito rápido do *CubeSat*, ultrapassando bastante o *SetPoint*. Apesar disso, o sistema conseguiu voltar a condição de regime após cerca de 20 segundos, com erro inferior a 5 graus (vide Figura 54).

Com base na Tabela 1, que informa os requisitos de controle, percebe-se que o sistema não atendeu aos requisitos de máximo sobressinal e de tempo de acomodação após a perturbação, contudo, esteve dentro da margem de erro de *SetPoint* exigida, que é o principal critério a ser cumprido, levando a conclusão de que mesmo o controle não tendo sido feito considerando possíveis perturbações, o sistema apresentou uma resposta consideravelmente boa.

O vídeo apresentando o funcionamento do teste de perturbação está disponível no *link* apresentado na Tabela 3.

Tabela 3 – *Links* para vídeo do funcionamento do teste de perturbação.

| Teste                | Link                                                                    |
|----------------------|-------------------------------------------------------------------------|
| Teste de perturbação | <a href="https://youtu.be/YwhvCtYsNHk">https://youtu.be/YwhvCtYsNHk</a> |

Fonte: De autoria própria.

#### 4.6 Dificuldades enfrentadas

O processo de montar um protótipo de *CubeSat*, uma estrutura de testes e um *software* de validação do sistema, é uma tarefa que exige bastante trabalho, dedicação e paciência, visto que diversos problemas ocorrem neste processo, tanto de *hardware*, quanto de *software*.

Tendo em vista que a planta foi montada com recursos limitados, o *hardware* do sistema é feito de componentes mais baratos e, por consequência, mais frágeis. Dentre os principais problemas enfrentados, estão os relacionados ao sensor giroscópio MPU6050, que recorrentemente parava de funcionar e era incapaz de fazer as leituras de posição angular do sistema, tendo que ser recalibrado e até mesmo

substituído nestes casos. Outro problema enfrentado, foi na montagem e disposição dos componentes eletrônicos dentro do *CubeSat*, uma vez que o Arduino e a Ponte H ocupam bastante espaço, tornando mais difícil a interligação e inserção do sensor, módulo de comunicação e a fiação dos componentes.

Além das dificuldades supracitadas, houveram problemas com a roda de reação, que ficou relativamente pesada, causando um aumento da zona morta do motor e dificultando a modelagem da planta e o projeto do controlador. Apesar de todas estas dificuldades, o sistema teve um bom comportamento e atendeu aos requisitos de controle.

## 5 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

É possível concluir que o sistema físico do *CubeSat* projetado, apresentou um comportamento satisfatório no que diz respeito ao seu controle de atitude. Os resultados obtidos na Subseção 4.4, mostram que o sistema atendeu a todos os requisitos de controle estabelecidos nos testes realizados, enfatizando que o planta física se comportou de forma compatível com aquilo que foi teorizado e simulado. Além disso, a partir dos resultados obtidos na Subseção 4.5, percebe-se que o sistema reage de forma consideravelmente boa quando submetido a perturbações que venham a afetar sua posição angular no eixo *yaw*. Isto mostra que a forma como todo o sistema (dinâmica da planta e controlador) foi modelado, é uma alternativa simples, contudo, viável para a modelagem de controle de atitude de *CubeSats* com um atuador no eixo *yaw*.

Dada a gama de aplicações e projetos em dispositivos que exigem o controle de atitude, a abordagem apresentada aqui pode ser utilizada para o controle de atitude em um eixo em outros modelos de *Cubesats*, como o 2U ou o 6U, por exemplo, além de outros tipos de satélites dos mais variados tamanhos.

Sugere-se como trabalho futuro e proposta de continuidade, realizar com base no que foi feito, uma nova montagem do *hardware* como plataforma de testes, utilizando componentes mais robustos para sensor de posição giroscópio, como o STM32, de modo a obter medições de posição angular mais precisas e uma roda de reação mais leve, que cause uma diminuição significativa na zona morta do motor e menos vibrações na planta. Além disso, sugere-se desenvolver um módulo computacional personalizado, substituindo o Arduino e a Ponte H por um microcontrolador desenvolvido especificamente para o projeto, otimizando o consumo de energia e o espaço físico no interior do *CubeSat*. Outro ponto a ser aprimorado diz respeito a estrutura de testes responsável por suspender o nano satélite, que permitiu que o mesmo fizesse um movimento ligeiramente pendular, prejudicando medições mais precisas de seu apontamento. Para corrigir isso, sugere-se a inclusão de uma haste na parte inferior da estrutura, prendendo o *CubeSat* também pela parte de baixo.

Considerando que este projeto forneceu a base para a modelagem de uma planta de *CubeSat* e o projeto do seu controle de atitude uniaxial, é possível afirmar que este trabalho é o primeiro passo rumo a um objetivo maior: uma futura missão espacial com o lançamento de um nano satélite.



## REFERÊNCIAS

- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Medição de resistência de aterramento e de potenciais na superfície do solo em sistemas de aterramento. **ABNT NBR 15749**, p. 49, set. 2009.
- AGUIRRE, Luis A. **Introdução à Identificação de Sistemas: Técnicas Lineares e Não Lineares: Teoria e Aplicação**. 4. ed.
- AMSAT-UK. Bulgarian CubeSat EnduroSat on ISS awaiting deployment. **AMSAT-UK**, 2018. Disponível em: <<https://amsat-uk.org/2018/05/30/bulgaria-endurosat/>>. Acesso em: 02 Junho 2023.
- ARDUINOECIA. Controlando motores DC com o Arduino Motor Shield L293D. **www.arduinoecia.com.br**, 2014. Disponível em: <<https://www.arduinoecia.com.br/arduino-motor-shield-l293d/>>. Acesso em: 19 Agosto 2023.
- CASTELLAN, Ygor. CubeSats: A grande Tecnologia Dos Pequenos Satélites. **Site AEROJR**, 2022. Disponível em: <<https://aerojr.com/blog/cubesats-a-grande-tecnologia-dos-pequenos-satelites/>>. Acesso em: 26 Maio 2023.
- CGEE. Observatório de Ciência, Tecnologia e Inovação - CGEE. **www.cgee.org.br**, 2017. Disponível em: <[https://www.cgee.org.br/documents/10195/734063/3069\\_Documento+Estrat%C3%A9gio+para+o+Setor+Espacial+-+n%C3%BAmero+1.pdf/27e2e351-6341-468b-93c8-be7ba3f0630e?version=2.0](https://www.cgee.org.br/documents/10195/734063/3069_Documento+Estrat%C3%A9gio+para+o+Setor+Espacial+-+n%C3%BAmero+1.pdf/27e2e351-6341-468b-93c8-be7ba3f0630e?version=2.0)>. Acesso em: 19 Agosto 2023.
- CGEE. CubeSats - CGEE. **www.cgee.org.br**, 2018. Disponível em: <[https://www.cgee.org.br/documents/10195/734063/3111\\_CGEE\\_resumoexecutivo\\_CubeSats\\_web.pdf/d25668f5-4d98-411c-b79a-e7f6f369d7bd?version=2.3](https://www.cgee.org.br/documents/10195/734063/3111_CGEE_resumoexecutivo_CubeSats_web.pdf/d25668f5-4d98-411c-b79a-e7f6f369d7bd?version=2.3)>. Acesso em: 17 ago. 2023.
- DIMCEV, V.; HANDJISKI, B.; SEKERINSKA, R. Alternative fall-of-potential method for grounding grids impedance measurements and inductive coupling between leads. **Electromagnetic Compatibility, 2003. EMC '03. 2003 IEEE International Symposium on**, v. 1, p. 74-77, May 2003.
- ENNIS, Stephen. CubeSat Networks: Balancing Power with Satellite-to-Ground Data Throughput, Dublin, Maio 2017. 18.
- FABRI, Sérgio M. Determinação de Atitude de Satélites Artificiais Estabilizados por Rotação com o uso do GPS. **Tese (Doutorado em Ciências Geodésicas) - Universidade Federal do Paraná**, Curitiba, 13 Novembro 1997. 2-3.
- FRACKIEWICZ, Marcin. Nanosatélites CubeSat para exploração lunar. **ts2.space**, 2023. Disponível em: <<https://ts2.space/pt/nanosatelites-cubesat-para-exploracao-lunar/>>. Acesso em: 17 Agosto 2023.
- FRACKIEWICZ, Marcin. Serviços de lançamento do CubeSat: desafios e oportunidades. **ts2.space**, 2023. Disponível em: <<https://ts2.space/pt/servicos-de-lancamento-do-cubesat-desafios-e-oportunidades/>>. Acesso em: 17 ago. 2023.
- FSADU. UFMA desenvolve o primeiro nanossatélite do NE. **Fundação sousândrade**, 2021. Disponível em: <<https://www.fsadu.org.br/acontece-na-fsadu/ufma-desenvolve-o-primeiro-nanosatelite-do-ne/>>. Acesso em: 03 Junho 2023.
- GUIMARÃES, Fábio. Acelerômetro MPU6050 com o Arduino. **Site do Mundo Projetado**, 2018. Disponível em: <<https://mundoprojetado.com.br/acelerometro-mpu6050-arduino/>>. Acesso em: 23 jun. 2023.

HARFORD, Tim. CubeSat: Como um estranho bicho de pelúcia ajudou a revolucionar a indústria de satélites. **bbcnews**, 2019. Disponível em: <<https://www.bbc.com/portuguese/curiosidades-49057778>>. Acesso em: 19 Agosto 2023.

INPE. CubeDesign Virtual 2021. **Site do gov INPE**, 2021. Disponível em: <<http://www.inpe.br/cubedesign/2021/CubeSat/regulamento.php>>. Acesso em: 30 Maio 2023.

KORASLI, C. Ground resistance measurement with alternative fall-of-potential method. **Power Delivery, IEEE Transactions on**, v. 20, n. 2, p. 1657-1661, Apr 2005.

KORASLI, C. Ground Resistance Measurement with Fall-of-Potential Method Using Capacitive Test Probes. **Transmission & Distribution Construction, Operation and Live-Line Maintenance, 2006. ESMO 2006. IEEE 11th International Conference on**, 15-19 Oct. 2006.

LISS, J. et al. External Verification of CubeSat Using AprilTag Visual Fiducials and an Extended Kalman Filter. **SEMANTIC SCHOLAR**, 2018.

LOFF, Sarah. CubeSats Overview. **Site da Nasa**, 2018. Disponível em: <[https://www.nasa.gov/mission\\_pages/cubesats/overview](https://www.nasa.gov/mission_pages/cubesats/overview)>. Acesso em: 26 Maio 2023.

MA, Jinxi; DAWALIBI, F.P. Extended analysis of ground impedance measurement using the fall-of-potential method. **Power Delivery, IEEE Transactions on**, v. 17, n. 4, p. 881-885, Oct. 2002.

MESQUITA, Brehme D. R. D. ANÁLISE DE ESTIMAÇÃO E CONTROLE DE ATITUDE EM MODO DE OPERAÇÃO NOMINAL DO CONASAT. **Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle**, São José dos Campos, 26 Abril 2017.

MESQUITA, Brehme D. R. D. Análise de Estimação e Controle de Atitude em Modo de Operação Nominal do Conasat por Filtro SDRE e Controle PID. **Dissertação (Mestrado em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais**, São José dos Campos, 26 Abril 2017.

OBSAT. Pequenos satélites: Grandes possibilidades. **obsat.org.br**, 31 Dezembro 2022. Disponível em: <<https://obsat.org.br/ebook/download.php?file=eBook-PequenosSatelites-Dez-2022.pdf>>. Acesso em: 17 Agosto 2023.

PAUL, Clayton R. **Eletromagnetismo para Engenheiros: Com Aplicações**.

PEIXOTO, PEDRO J. D. D. Controle Preditivo Baseado em Modelo Aplicado Satélites Rígido-Flexíveis. **Dissertação de Mestrado**, Brasília, 26 Março 2018. 1-2.

RAVIKIRAN. Beginners Guide to What Is Arduino? - Simplilearn.com. **www.simplilearn.com**, 2022. Disponível em: <<https://www.simplilearn.com/tutorials/programming-tutorial/arduino-beginners-guide>>. Acesso em: 19 Agosto 2023.

ROCKETLAB. Rocket Lab. **rocketlabusa**, 2023. Disponível em: <<https://www.rocketlabusa.com/space-systems/satellite-components/reaction-wheels/>>. Acesso em: 28 Junho 2023.

SIMONE BUSO, Paolo M. **Digital Control in Power Electronics**.

SOUZA, Petrônio N. D. Subsistema de Controle de Atitude - INPE. **inpe**, 2002. Disponível em: <[http://mtc-m21c.sid.inpe.br/col/sid.inpe.br/mtc-m21c/2019/08.22.14.06/doc/170\\_Subsistema%20de%20Controle%20de%20Atitude\\_P2.2\\_v1\\_2002.pdf](http://mtc-m21c.sid.inpe.br/col/sid.inpe.br/mtc-m21c/2019/08.22.14.06/doc/170_Subsistema%20de%20Controle%20de%20Atitude_P2.2_v1_2002.pdf)>. Acesso em: 17 Agosto 2023.

SPARKFUN. What is an Arduino? - SparkFun Learn. **learn.sparkfun.com**. Disponível em: <<https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>>. Acesso em: 19 ago. 2023.

SUNDE, Erling D. **Earth conduction effects in transmission systems.**

WANG, Cheng-gang et al. Grounding resistance measurement using fall-of-potential method with potential probe located in opposite direction to the current probe. **Power Delivery, IEEE Transactions on**, v. 13, n. 4, p. 1128-1135, Oct. 1998.

WERTZ, J. R. **Spacecraft attitude determination and control.**

WIKIPEDIA. Aircraft principal axes. **wikipedia**, 2023. Disponível em: <[https://en.wikipedia.org/wiki/Aircraft\\_principal\\_axes](https://en.wikipedia.org/wiki/Aircraft_principal_axes)>. Acesso em: 04 Junho 2023.

XUKYO. Arduino e o módulo Bluetooth HC-06 - AranaCorp. **www.aranacorp.com**, 2021. Disponível em: <<https://www.aranacorp.com/pt/arduino-e-o-modulo-bluetooth-hc-06/>>. Acesso em: 19 Agosto 2023.